Gandhinagar Institute of Technology

Academic Regulation & Curriculum

for

MTech

in

Software Engineering
Computer Engineering Department

MTech PROGRAMME

***w.e.f.* 2026-27**

Preamble

The Master of Technology (M.Tech.) in Software Engineering is a postgraduate program designed to develop advanced knowledge, technical expertise, and research capabilities in the field of software systems and emerging technologies. The program aims to equip students with a strong foundation in software design, development, testing, maintenance, project management, and advanced computing paradigms. Through a carefully structured curriculum comprising core courses, professional electives, research methodology, seminars, and dissertation work, students gain both theoretical understanding and practical experience to address complex engineering challenges.

The program emphasizes innovation, critical thinking, and research-oriented learning while aligning with current industry requirements and technological advancements. Specialized courses in areas such as Data Science, Distributed Systems, High Performance Computing, Machine Learning, Cloud Computing, and Software Project Management enable students to develop expertise in contemporary domains. The inclusion of research reviews, internal evaluations, and dissertation work fosters analytical skills, independent learning, and scholarly contributions.

The Department of Computer Engineering (SE) is committed to producing competent professionals, researchers, and academicians who can contribute effectively to the software industry, research organizations, and higher education institutions. The program promotes lifelong learning, ethical practices, and professional excellence, preparing graduates to meet the evolving demands of the global technology landscape.

Table of Contents

1.Institutional Vision & Mission

VISION:

To create a flexible and learner-focused engineering ecosystem that builds strong technical skills, practical knowledge, and innovative thinking, preparing graduates to adapt, perform, and lead in a rapidly evolving technological and societal environment.

MISSION:

To provide a holistic learning environment that develops technical skilled, innovative, and socially responsible engineers capable of addressing industrial, technological, and societal needs.

2.Program Objectives

PEO 1:
To provide graduates with advanced knowledge and technical competence in Software Engineering, enabling them to design, develop, manage, and maintain complex software systems using modern tools and technologies.

PEO 2:
To develop analytical, problem-solving, and research skills that enable graduates to investigate emerging challenges and contribute innovative solutions in software engineering and related domains.

PEO 3:
To prepare graduates for successful careers in industry, research organizations, academia, and entrepreneurship through professional expertise, leadership qualities, and effective communication skills.

PEO 4:
To foster lifelong learning and adaptability, enabling graduates to continuously upgrade their knowledge and skills in response to evolving technological advancements and industry requirements.

PEO 5:
To inculcate professional ethics, social responsibility, and commitment to quality in the development and deployment of software systems for the benefit of society.

PEO 6:
To encourage innovation, interdisciplinary collaboration, and scholarly contributions through research, publications, and advanced project work in emerging areas of Software Engineering.

3.**Program Outcomes (POs)**

POs	Title	Description
PO1	Advanced Knowledge	Apply advanced concepts, theories, and methodologies of Software Engineering to analyze and solve complex engineering problems.
PO2	Problem Analysis	Identify, formulate, and critically analyze complex software engineering challenges using research-based knowledge and modern computational techniques.
PO3	Design and Development of Solutions	Design, develop, and evaluate software systems, architectures, and processes that meet specified requirements while considering technical, economic, and societal constraints.
PO4	Research and Innovation	Conduct independent research, investigate emerging technologies, and contribute to innovation through experimentation, analysis, and scholarly publications.
PO5	Modern Tool Usage	Select and apply appropriate software engineering tools, frameworks, platforms, and technologies for the

		development and management of complex software systems.
PO6	Project Management and Leadership	Demonstrate project planning, management, teamwork, leadership, and decision-making skills in multidisciplinary environments.
PO7	Ethics and Professional Responsibility	Apply ethical principles, professional standards, and legal responsibilities in software development, deployment, and maintenance.
PO8	Communication Skills	Communicate effectively through technical reports, research papers, presentations, and professional interactions with diverse stakeholders.
PO9	Lifelong Learning	Recognize the need for continuous learning and engage in self-directed professional development to adapt to technological advancements.
PO10	Societal and Environmental Impact	Evaluate the impact of software solutions on society, business, and the environment, ensuring sustainable and responsible technological development.
PO11	Interdisciplinary Collaboration	Work effectively with professionals from diverse disciplines to develop innovative solutions for real-world problems.

PO12	Independent Research and Dissertation	Plan, execute, and present high-quality research and dissertation work that contributes to the advancement of Software Engineering and related fields.

4. **Program Details**

4.1. Program Structure & Duration

- Duration & Intake:2 Years & 60 Seats
- Semesters: 4
- Mode: Full Time
- Credit Structure: As per NEP / University norms
- Medium of Instruction: English
- Intake: 18
- Fees: The fee structure prescribed by the University from time to time shall be applicable.

4.2 Eligibility Criteria for Admission

Candidates seeking admission to M.Tech must have completed a B.E./B.Tech degree or an equivalent qualification in the relevant branch from a recognized university. Generally, candidates belonging to the General category must secure at least 50% marks in the qualifying examination, while candidates from reserved categories such as SC, ST, SEBC, and EWS of Gujarat require a minimum of 45% marks. Admission is primarily based on a valid GATE score, and candidates without GATE qualification can apply through the Gujarat PGCET examination or on the basis of merit as per ACPC rules.

4.3 Admission Procedure

Admission to the M.Tech program shall be granted on the basis of merit obtained in the qualifying examination and/or valid GATE score and/or Gujarat PGCET score, as prescribed by the Admission Committee for Professional Courses (ACPC) and the University from time to time. Candidates must possess a Bachelor's degree in Engineering/Technology or an equivalent qualification in the relevant discipline from a recognized university. The admission process may also include document verification, centralized counseling, choice filling, and seat allotment as per ACPC norms. The University reserves the right to revise the eligibility criteria, admission procedure, and selection process from time to time.

4.4 Credit Framework

Sl. No.	Course Type	MTech-SE Credit
1.	Dissertation (Ph I + II)	28
2.	Professional Core Courses	16
3.	Professional Elective Courses	16
4.	Internal Review (I + II)	4
5.	Mandatory Learning (RM & IPR)	6
6.	Technical Writing	-
	Total	70

Definition of Credit (as per UGC / AICTE):

- 1 Hour Lecture (L)per Week=1Credit
- 1 Hour Tutorial (T) per Week=1Credit
- 1 Hour Practical (P) per Week=0.5Credits
- 2 Hours Practical (Lab) per Week=1Credit

Range of Credits (as per UGC / AICTE):

- ✔ A total of 70 credits will be necessary for a student to be eligible to get MTech degree.
- ✔ These could be acquired through MOOCs.
- ✔ Any student completing any course through MOOC will have to submit an appropriate certificate to earn the corresponding credit.
- ✔ For any additional information , the student may contact the concerned HODs.

5. Curriculum Structure

			Gandhinagar Institute of Technology (01)									
Program Name: Master of Technology (02)				Program Code: CE (007) Software Engineering								
Semester / Year : First Year Master of Technology - Semeter I				Effective From Academic Year: 2022-23								
Subject Code	Course Type	Subject Name	Credit	Hours per week				Theory (Marks)		Practical (Marks)		Total (Marks)
				L	T	P	Total	University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
10070101	Core I	Advance Software Engineering	4	3	0	2	5	60	40	30	20	150
10070102	Core II	Advance Data Structures & Algorithms	4	3	0	2	5	60	40	30	20	150
10000153	Audit Course	Constitution of India	0	2	0	0	2	0	50	0	0	50
10000151	MLC	Research Methodology and IPR	3	1	1	2	4	0	0	80	20	100
	Program Elective I	-	4	3	0	2	5	60	40	30	20	150
	Program Elective II	-	4	3	0	2	5	60	40	30	20	150
			19				26					750
10070103	Program Elective I	Data Science										
10070104	Program Elective I	Data Warehousing & Data Mining										
10070105	Program Elective I	High Performance Computing										
10070106	Program Elective II	Image Processing										

10070107	Program Elective II	Wireless Sensor Network										
10070108	Program Elective II	Data Security & Access Control										

			Gandhinagar Institute of Technology (01)									
Program Name: Master of Technology (02)							Program Code: CE (007) Software Engineering					
Semester / Year : First Year Master of Technology - Semeter II							Effective From Academic Year: 2022-23					
Subject Code	Course Type	Subject Name	Credit	Hours per week				Theory (Marks)		Practical (Marks)		Total (Marks)
				L	T	P	Total	University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
10000251	Audit Course	Technical Writing	0	2	0	0	2	0	50	0	0	50
10070201	Core III	Software Project Management	4	3	0	2	5	60	40	30	20	150
10070202	Core	Research Review & Seminar	3	0	0	6	6	0	0	0	100	100
10070203	Core IV	Distributed Systems	4	3	0	2	5	60	40	30	20	150
	Program Elective III	-	4	3	0	2	5	60	40	30	20	150
	Program Elective IV	-	4	3	0	2	5	60	40	30	20	150
			19				28					750
10070204	Program Elective III	Machine Learning										
10070205	Program Elective III	Data Visualization										
10070206	Program Elective III	Blockchain Technology										
10070207	Program Elective	Computer Vision										

	IV											
10070208	**Program Elective IV**	**Cloud Computing**										
10070209	**Program Elective IV**	**Natural Language Processing**										

	Gandhinagar Institute of Technology (01)

Mtech SE (Sem - 3)

Semester / Year : Second Year Master of Technology - Semeter III	**Effective From Academic Year: 2026-27**

Subject Code	**Course Type**	**Subject Name**	**Credit**	**Hours per week**				**Theory (Marks)**		**Practical (Marks)**		**Total (Marks)**
				L	**T**	**P**	**Total**	University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
10070301	Internal Review	Internal Review I	2	0	0	4	4	0	0	0	100	100
10070302	Dissertation	Dissertation Phase I	14	0	0	28	28	0	0	100	0	100
			16				32					100

Gandhinagar Institute of Technology (01)

Mtech SE (Sem - 4)

Semester / Year : Second Year Master of Technology - Semeter IV	**Effective From Academic Year: 2026-27**

Subject Code	**Course Type**	**Subject Name**	**Credit**	**Hours per week**				**Theory (Marks)**		**Practical (Marks)**		**Total (Marks)**
				L	**T**	**P**	**Total**	University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
10070401	Internal Review	Internal Review II	2	0	0	4	4	0	0	0	100	100
10070402	Dissertation	Dissertation Phase II	14	0	0	28	28	0	0	100	0	100
			16				32					200

1stSem	750	**3rdSem**	200		
2ndSem	750	**4th Sem**	200	**TotalCredit:**	**70**
				TotalMarks:	**1900**

6. Teaching Learning and Evaluation

6.1 Teaching Methodology

The program shall adopt the following teaching methodologies:

- Classroom Teaching
- Case Studies
- Industry Interaction
- Site Visits
- Workshops
- Field Surveys
- Research Projects
- Research work
- Paper Publication
- AI-based Learning Tools
- Blended and Digital Learning

6.2 Attendance Rules

- Minimum 75% attendance shall be mandatory in each course.
- Students below the prescribed attendance may not be permitted to appear in examinations.
- Additional attendance requirements may apply for practical and research work and paper publication.

6.3 Examination and Evaluation System

6.3.1 Evaluation Components

The evaluation may consist of Continuous Internal Evaluation (CIE) and End Semester Examination (ESE).

Continuous Internal Evaluation (CIE) may include:

- Mid-Semester Examination
- Assignments
- Presentations
- Research work
- Paper Publication

- Class Tests
- AI Quiz & Viva Voce
- Dissertation
- Project Work

6.4 Suggested Weightage

- Continuous Internal Evaluation (CIE): 40%
- End Semester Examination (ESE): 60%

6.5 Passing Standards

- Minimum 50% marks in each component where applicable.
- Minimum 50% aggregate marks for passing the course/program.
- Separate passing may be required in theory and practical examinations.

6.6 Grading System

The program shall follow Gandhinagar University grading norms.

6.6.1 Grade Table

Grade	Grade Points	From Marks	To Marks
AA	10	80	100
AB	9	75	79
BB	8	70	74
BC	7	65	69
CC	6	60	64
CD	0	55	59
DD	-	50	54
DF	-	0	49

6.7 Dissertation / Project Work

Requirements may include:

- Proposal Presentation
- Periodic Review
- Final Report Submission
- Viva Voce Examination

The dissertation shall be evaluated by internal and external examiners.

7. Rules and Regulations

7.1 Rules for Promotion

A student shall be promoted to the next semester as per Gandhinagar University rules, subject to fulfilment of academic requirements, minimum attendance, and completion of required credits.

7.2 Maximum Duration for Completion

The maximum duration for completing the program shall be 2 years from the date of admission (or as prescribed by the University).

7.3 Dissertation Guidelines

7.3.1 Internal Review – I /II

Purpose

Internal Review – I and II is conducted before the formal submission of Dissertation Phase – I and II respectively to evaluate the early-stage research progress.

The review focuses on:

- Research problem identification
- Literature survey quality
- Research gap identification
- Feasibility and relevance of the selected topic
- Research methodology planning
- Implementation progress
- System/model architecture
- Experimental setup
- Preliminary results and analysis
- Performance evaluation

- Challenges faced and solutions adopted
- Future work and completion plan

Student Presentation Requirements

- Proposed research topic
- Problem statement
- Literature review
- Research gap identification
- Objectives of the research
- Proposed methodology/framework
- Expected outcomes
- Implementation status report
- Experimental methodology
- Preliminary results and analysis
- Graphs, tables, or comparative outputs
- Challenges faced and corrective actions
- Updated dissertation draft
- Final work completion plan

Submission Requirements

- Literature survey summary
- Research problem formulation document
- Proposed methodology/framework
- Preliminary research plan
- Timeline for Dissertation Phase – I
- Minimum 10–15 research paper references from reputed journals/conferences
- Implementation status report
- Experimental methodology
- Preliminary results and analysis
- Graphs, tables, or comparative outputs
- Challenges faced and corrective actions
- Updated dissertation draft
- Final work completion plan

7.3.2 Dissertation Phase – I/II

Purpose

Dissertation Phase – I and II focuses on research planning, methodology development, system/model design, and preliminary experimentation.

Students shall work under the guidance of an assigned faculty supervisor.

Activities to be Performed

- Identification of research problem
- Extensive literature review
- Definition of objectives and scope
- Selection of suitable methodology
- Dataset identification and collection
- Initial model/system design
- Tool and technology selection
- Preliminary experimentation and analysis
- Research proposal preparation
- Algorithm/model implementation
- AI model training and testing
- Simulation or deployment
- Experimental analysis
- Comparative performance evaluation
- Result validation
- Optimization and improvements
- Research paper preparation (if applicable)
- Final dissertation report preparation

Seminar and Review

The presentation may include:

- Research significance
- Technical approach
- Proposed framework
- Preliminary results
- Research plan for Phase – II

7.3.3 Final Dissertation Report Structure

The dissertation report shall contain the following chapters:

1. Title Page
2. Certificate
3. Declaration
4. Acknowledgement
5. Abstract
6. Table of Contents
7. List of Figures
8. List of Tables
9. Introduction
10. Literature Review
11. Problem Statement and Research Gap
12. Proposed Methodology / Model
13. System Design and Architecture
14. Implementation and Experimental Setup
15. Results and Analysis
16. Comparison with Existing Methods
17. Conclusion and Future Work
18. References
19. Appendices (if applicable)

7.3.4 Dissertation Submission Requirements

Students must submit:

- Final dissertation report (hard copy and soft copy)
- Plagiarism report as per UGC norms
- Source code/software/project files
- Dataset details (if applicable)
- Research paper/publication details (if any)
- Completion certificate signed by guide
- Presentation slides for viva voce
- Project demonstration (if applicable)

7.4 Discipline Rules

Students shall maintain discipline, professional ethics, and proper conduct within the institution, during industry interactions, examinations, and all academic activities.

Students are expected to:

- Attend classes regularly
- Follow institutional rules and regulations
- Respect faculty members, staff, fellow students, and institutional property
- Maintain academic integrity in assignments, projects, and examinations

Any form of misconduct, ragging, indiscipline, use of unfair means in examinations, damage to institutional property, harassment, or behavior affecting the academic environment shall be treated seriously and may result in disciplinary action including:

- Warning
- Suspension
- Cancellation of Examination
- Withholding of Results
- Expulsion from the Program

Such actions shall be taken as per University rules and UGC regulations.

Students shall also comply with all safety, ethical, and professional standards prescribed by the institution from time to time.

7.5 Anti-Ragging Policy

The program shall strictly follow:

- UGC Anti-Ragging Regulations
- Anti-Ragging Affidavit Requirements
- Institutional Disciplinary Procedures
- Ragging in any form is strictly prohibited.

7.6 Ethical Use of Technology and AI Tools

Students are encouraged to use:

- Various Software and AI Tools
- Data Analytics Tools
- AI-based Learning Platforms
- Valuation Software
- Digital Survey Tools

Use of AI tools shall be ethical and academic integrity must be maintained.

8. Student Support

8.1 Industry Collaboration

The Department of Software Engineering shall actively promote collaboration with software development companies, IT service organizations, cloud service providers, cybersecurity firms, research institutions, startups, consulting organizations, product-based companies, and government agencies to enhance the practical exposure and industry relevance of the M.Tech (Software Engineering) program.

Such collaborations may include internships, industrial training, live projects, expert lectures, workshops, industrial visits, hackathons, consultancy assignments, collaborative research activities, certification programs, skill development initiatives, and placement support.

The institution may establish Memoranda of Understanding (MoUs) with industries, research laboratories, professional bodies, and technology organizations for academic enrichment, technology transfer, innovation activities, startup incubation, and collaborative research aimed at improving employability, entrepreneurship, technical competency, and industry readiness among students.

8.2 Research and Consultancy

The department shall encourage faculty members and students to actively engage in research, consultancy, innovation, and applied studies related to Software Engineering, Software Architecture, Software Testing and Quality Assurance, Software Project Management, Distributed Systems, Cloud Computing, Data Science, High Performance Computing, Artificial Intelligence, Machine Learning, Cybersecurity, and other emerging technologies.

Students may undertake research projects, software development activities, cloud-based application development, performance analysis, software process improvement studies, distributed application design, data analytics projects, and industry-oriented projects as part of their academic and professional training.

The institution may also undertake consultancy assignments and collaborative projects for industries, startups, government organizations, healthcare institutions, financial organizations, educational institutions, and technology firms in areas such as enterprise software development, cloud solutions, software quality assurance, distributed systems, software maintenance, software process optimization, and data-driven applications.

Participation in seminars, conferences, workshops, technical competitions, publications, funded projects, patent filing activities, and collaborative research initiatives shall be encouraged to promote academic excellence, innovation, and research-oriented learning.

8.3 Library and Laboratory Facilities

Adequate library, laboratory, computing, and digital learning facilities shall be made available to support effective teaching, practical training, research, and skill development under the M.Tech (Software Engineering) program.

The department shall facilitate access to relevant books, journals, research publications, e-resources, IEEE and ACM digital libraries, software repositories, cloud platforms, project management tools, software development frameworks, and online learning resources related to Software Engineering and allied domains.

Necessary computing facilities, software development environments, cloud computing platforms, version control systems, testing and quality assurance tools, high-performance computing resources, simulation software, internet access, and other technical resources required for practical and project-based learning shall also be provided to enhance industry-oriented education, research capability, innovation, and professional competency among students.

8.4 Student Support and Mentoring

A structured system for student support, academic mentoring, research guidance, and professional counselling shall be implemented for students enrolled in the M.Tech (Software Engineering) program.

The department shall facilitate academic counselling, dissertation guidance, career mentoring, research support, technical training, certification guidance, and skill development activities to assist students in academic progression and professional growth.

Students may also be provided opportunities for interaction with industry experts, researchers, alumni, and entrepreneurs through workshops, seminars, coding competitions, technical events, training programs, industrial visits, and extracurricular activities aimed at enhancing employability, leadership qualities, communication skills, teamwork, innovation, and overall personality development.

Necessary support mechanisms for addressing academic and personal concerns shall also be made available as per institutional policies.

8.5 Placement and Career Development

The department shall promote placement assistance and career development activities for students through industry interaction, internships, research collaboration, professional networking, technical training programs, and campus engagement initiatives.

Efforts shall be made to establish linkages with software development companies, IT consulting firms, cloud computing organizations, cybersecurity companies, research institutions, startups, multinational corporations, and technology-based enterprises to facilitate internships, live projects, industrial exposure, research opportunities, and employment opportunities for students.

Career development activities such as aptitude training, coding practice sessions, software development workshops, resume preparation, interview preparation, soft skill development, entrepreneurship guidance, certification programs, expert sessions, technical workshops, and placement-oriented training programs may also be organized to enhance professional competency, innovation capability, leadership skills, and employability of students.

9. Program Credibility

9.1 Executive Summary

The M.Tech (Software Engineering) program is designed to develop industry-ready professionals, researchers, and innovators with advanced expertise in Software Engineering, Software Architecture, Software Development, Software Testing, Distributed Systems, Cloud Computing, Data Science, High Performance Computing, and emerging software technologies aligned with current industry and research requirements.

The curriculum integrates strong theoretical foundations with practical implementation, research orientation, industry exposure, and advanced computing skills to prepare students for professional careers, higher studies, research, and entrepreneurship in software and technology-driven domains.

The program is delivered by experienced faculty members, researchers, industry experts, and professionals actively engaged in Software Engineering, Software Development, Data Analytics, Cloud Computing, Cybersecurity, and emerging computing technologies, ensuring that students gain practical knowledge along with academic understanding.

Specialized modules related to Advanced Software Engineering, Data Structures and Algorithms, Software Project Management, Distributed Systems, Cloud Computing, Data Science, Machine Learning, High Performance Computing, Software Quality Assurance, and Research Methodology are delivered through laboratory practice, case studies, seminars, workshops, industry interaction, project-based learning, and research activities.

The program emphasizes innovation, problem-solving, software design, research competency, and professional ethics through dissertation work, live projects, technical training, collaborative learning, and industry engagement activities.

The curriculum is aligned with current technological advancements, software industry requirements, emerging research trends, and national educational guidelines while integrating advanced computing, analytical, managerial, and research skills required in academia, industry, startups, and research organizations.

Graduates of the program shall be academically and professionally prepared for careers in Software Development, Software Architecture, Cloud Computing, Data Science, Project Management, Research and Development, Enterprise Systems, and emerging technology domains, as well as for higher studies and doctoral research.

The program also encourages continuous exposure to modern software development tools, cloud platforms, open-source technologies, software engineering frameworks, research publications, innovation ecosystems, startup culture, and real-world software applications as part of experiential and industry-oriented learning throughout the program duration.

9.2 Scope and Career Opportunities

- Software Development Companies
- Information Technology (IT) Industries
- Cloud Computing Organizations
- Data Science and Analytics Firms
- Cybersecurity Companies
- Research and Development Centers
- Enterprise Software Organizations
- Product-Based Technology Companies
- FinTech Companies
- Healthcare Technology Organizations
- E-Commerce and Digital Service Providers
- Smart Systems and IoT Industries
- Government and Public Sector Organizations
- Startups and Innovation Centers
- Consulting and Technology Service Firms
- Academic and Research Institutions

9.3 Potential Job Roles

- Software Engineer
- Senior Software Developer
- Software Architect
- Software Quality Assurance Engineer
- Software Test Engineer
- Full Stack Developer
- Cloud Solutions Engineer
- DevOps Engineer
- Site Reliability Engineer (SRE)
- Data Scientist
- Machine Learning Engineer
- Distributed Systems Engineer
- Project Manager
- Technical Consultant
- Systems Analyst
- Research Associate
- Technology Consultant
- Enterprise Application Developer
- Product Engineer
- Research and Development Engineer

9.4 Higher Study and Professional Opportunities

Students may pursue:

- Ph.D. in Software Engineering and related disciplines
- Research Fellowships and Academic Research Programs
- Advanced Certifications in Cloud Computing, Software Architecture, Project Management, Data Science, Artificial Intelligence, and Cybersecurity
- International Professional Certifications
- Industry-based Research and Consultancy Opportunities
- Startup and Entrepreneurship Initiatives
- Patent Filing and Research Publication Activities
- Specialized Training Programs in Emerging Technologies
- Academic and Research Careers in Universities and Research Institutions
- Professional Memberships in National and International Technical Bodies

The program promotes lifelong learning, innovation, research excellence, and professional development, enabling graduates to contribute effectively to the evolving global software and technology ecosystem.

10. Collaborative Field Work / Experimental Work

10.1 Software Requirements Analysis and Documentation Exercise

Students shall perform requirement elicitation, analysis, and documentation for a real-world software application. The exercise shall include preparation of Software Requirement Specifications (SRS), use case diagrams, requirement validation, and feasibility analysis.

(Recommended Organizations / Platforms: IEEE Standards Association, Atlassian, IBM Rational, Microsoft)

10.2 Software Design and Architecture Exercise

Students shall design software architectures using object-oriented principles, design patterns, UML modeling, and architectural frameworks. The exercise shall include class diagrams, sequence diagrams, component diagrams, and architectural evaluation.

(Recommended Organizations / Platforms: Enterprise Architect, Visual Paradigm, Lucidchart, IBM)

10.3 Software Development and Implementation Exercise

Students shall develop software applications using modern programming frameworks and software engineering methodologies. The activity shall include coding standards, version control, modular development, and software deployment practices.

(Recommended Organizations / Platforms: GitHub, GitLab, Microsoft, Oracle, Open Source Communities)

10.4 Software Testing and Quality Assurance Exercise

Students shall perform software testing activities including unit testing, integration testing, system testing, regression testing, and performance testing. The exercise shall include defect analysis, quality metrics, and test documentation.

(Recommended Organizations / Platforms: Selenium, JUnit, Apache JMeter, IBM Quality Management)

10.5 Data Science and Analytics Exercise

Students shall perform data preprocessing, analysis, visualization, and predictive modeling using real-world datasets. The exercise shall include statistical analysis, data mining techniques, and interpretation of analytical results.

(Recommended Organizations / Platforms: Kaggle, Tableau, Microsoft Power BI, Python Foundation)

10.6 Cloud Computing and Distributed Systems Laboratory Exercise

Students shall deploy and manage applications in cloud environments and distributed systems. The exercise shall include virtualization, resource allocation, cloud deployment, service orchestration, and performance evaluation.

(Recommended Organizations / Platforms: AWS, Microsoft Azure, Google Cloud Platform)

10.7 High Performance Computing Experimental Work

Students shall implement parallel and distributed computing applications to solve computationally intensive problems. The exercise shall include performance measurement, scalability analysis, and optimization techniques.

(Recommended Organizations / Platforms: NVIDIA, OpenMP, MPI Foundation, Intel)

10.8 Machine Learning and Intelligent Systems Exercise

Students shall develop and evaluate machine learning models for classification, prediction, clustering, and recommendation systems. The activity shall include data preprocessing, model training, evaluation, and deployment.

(Recommended Organizations / Platforms: Scikit-learn, TensorFlow, PyTorch, OpenAI)

10.9 Software Project Management Exercise

Students shall prepare project plans involving effort estimation, scheduling, risk management, resource allocation, budgeting, and quality management using software project management tools and methodologies.

(Recommended Organizations / Platforms: Microsoft Project, Jira, Atlassian, PMI)

10.10 Research Review and Seminar Presentation

Students shall review recent research papers published in reputed journals and conferences related to Software Engineering and emerging technologies. Students shall prepare technical summaries, critical analysis reports, and seminar presentations.

(Recommended Organizations / Platforms: IEEE, ACM, Springer, Elsevier)

10.11 Industry-Oriented Mini Project

Students shall undertake software development projects addressing real-world problems. The project shall include requirement analysis, design, implementation, testing, deployment, and documentation following software engineering best practices.

(Recommended Organizations / Platforms: Industry Partners, Startups, Technology Firms)

10.12 Software Engineering Case Study Analysis

Students shall analyze software engineering case studies involving software failures, project management challenges, software quality issues, and successful software development practices. Students shall prepare analytical reports and recommendations.

(Recommended Organizations / Platforms: IEEE Software, ACM Digital Library, Software Engineering Institute)

10.13 Innovation Challenge / Hackathon Participation

Students shall participate in coding competitions, software development hackathons, innovation challenges, and technology events focused on solving real-world problems using software engineering principles and emerging technologies.

(Recommended Organizations / Platforms: Smart India Hackathon, AICTE, GitHub, NASSCOM)

10.14 Industry Internship / Industrial Training

Students shall undertake internships or industrial training in software development companies, IT organizations, research laboratories, startups, cloud service providers, or technology firms to gain practical exposure to professional software development environments.

(Recommended Organizations / Platforms: IT Industries, Startups, Research Organizations, Technology Firms)

10.15 Dissertation / Research Project

Students shall undertake a dissertation in Software Engineering or related interdisciplinary domains. The dissertation work shall include problem identification, literature review, research methodology, implementation, experimentation, analysis, and final thesis submission under the supervision of a faculty guide.

Students shall present their work through seminars, reviews, dissertation reports, and viva-voce examinations as per University and UGC norms.

(Recommended Organizations / Platforms: IEEE, ACM, Springer, Research Laboratories)

10.16 Portfolio Presentation and Exit Viva

Students shall prepare a professional portfolio consisting of software projects, research outputs, publications, experimental studies, internship work, certifications, and technical achievements completed during the program. Students shall deliver a final presentation and appear for an exit viva before a faculty and industry panel.

---

11. Market Survey

Strategic Feasibility for M.Tech (Software Engineering) in Ahmedabad, Gujarat

11.1 Executive Summary

India is witnessing rapid growth in Software Engineering, Cloud Computing, Data Science, Artificial Intelligence, Cybersecurity, Enterprise Software Development, Digital Transformation, and Distributed Computing due to increasing technological adoption across industries, government initiatives, startup ecosystems, and Industry 4.0 developments.

Gujarat, particularly Ahmedabad and Gandhinagar, is emerging as a major technology and innovation hub through the growth of GIFT City, IT parks, startup ecosystems, smart city initiatives, fintech organizations, digital governance projects, and technology-driven industries.

A market survey and industry interaction conducted among software companies, IT service providers, technology firms, startups, consulting organizations, research institutions, and industry professionals revealed a strong demand for highly skilled software engineering professionals capable of designing, developing, managing, and maintaining complex software systems.

The survey indicated increasing demand for expertise in Software Development, Software Architecture, Cloud Computing, Distributed Systems, Data Science, Software Testing, DevOps, Project Management, Cybersecurity, and emerging technologies.

Industry representatives emphasized the need for graduates possessing strong software engineering fundamentals, practical development experience, cloud-based application skills, software quality assurance expertise, research capabilities, and project management competencies.

The survey findings clearly indicate that introducing an M.Tech (Software Engineering) program will be highly beneficial and industry-relevant considering the increasing demand for advanced software professionals, research opportunities, technological innovation, and employment growth in Gujarat and across India.

11.2 Market Demand and Industry Trends

The demand for Software Engineering professionals continues to grow due to:

- Digital Transformation Initiatives
- Growth of Software Product Development
- Expansion of Cloud Computing Services
- Enterprise Application Development
- Data Science and Analytics Adoption
- Industry 4.0 and Smart Manufacturing
- Cybersecurity Requirements
- FinTech and E-Commerce Growth
- Software Quality and Automation Needs

- Distributed and Scalable Computing Systems
- Research and Innovation in Software Technologies

Emerging domains creating significant opportunities include:

- Software Engineering and Architecture
- Cloud Computing
- Distributed Systems
- Data Science and Analytics
- Machine Learning Applications
- DevOps and Site Reliability Engineering
- Software Quality Assurance
- Enterprise Application Development
- Cybersecurity Engineering
- High Performance Computing
- Research and Development

11.3 Skill Gap Identified by Industry

Industry feedback identified the following gaps among graduates:

- Limited practical exposure to large-scale software development
- Insufficient understanding of software architecture and design principles
- Lack of hands-on experience with cloud platforms and distributed systems
- Limited exposure to software testing and quality assurance tools
- Inadequate project management and team collaboration skills
- Insufficient research and analytical capabilities
- Limited industry-oriented project experience
- Lack of knowledge of modern software development frameworks and DevOps practices
- Limited exposure to emerging technologies and software engineering standards

The industry highlighted the need for professionally trained graduates capable of contributing effectively to software development, software maintenance, software quality assurance, research, and innovation activities with minimal additional training.

11.4 Employment Opportunities

Graduates of the M.Tech (Software Engineering) program shall have employment opportunities in:

- Software Development Companies
- Information Technology Organizations
- Product-Based Technology Companies
- Cloud Computing Companies
- Cybersecurity Organizations
- Research and Development Centers
- FinTech Organizations
- Healthcare Technology Companies
- E-Commerce Enterprises

- Government and Public Sector Organizations
- Consulting Firms
- Startups and Innovation Centers
- Academic and Research Institutions

11.5 Industry Support and Collaboration

Industry stakeholders have expressed strong support for the program through internships, industrial training, live projects, hackathons, technical workshops, guest lectures, expert sessions, collaborative research activities, and placement assistance.

The institution may establish collaborations and MoUs with software companies, cloud service providers, research laboratories, technology firms, professional organizations, and innovation centers to strengthen practical learning, research, and employability.

11.6 Research and Innovation Opportunities

The program shall encourage:

- Research Publications
- Patent Filing Activities
- Software Innovation Projects
- Funded Research Proposals
- National and International Conferences
- Technical Competitions and Hackathons
- Startup and Incubation Initiatives
- Industry-Sponsored Research Projects

Research opportunities shall be available in Software Engineering, Cloud Computing, Distributed Systems, Data Science, Software Testing, Software Architecture, High Performance Computing, Artificial Intelligence Applications, and related emerging domains.

11.7 Conclusion

The market survey and industry feedback clearly indicate a strong demand for professionally trained manpower in Software Engineering, Cloud Computing, Data Science, Distributed Systems, Software Development, and emerging digital technologies.

Considering the rapid expansion of the IT industry, software product development ecosystem, startup culture, digital transformation initiatives, and research opportunities in Gujarat and India, the proposed M.Tech (Software Engineering) program will provide excellent academic, research, and career opportunities while fulfilling the growing demand for advanced software engineering professionals.

The program is academically relevant, research-oriented, industry-focused, and aligned with current and future technological requirements, innovation ecosystems, and UGC guidelines for higher technical education.

12. Review and Revision of Regulations

The University reserves the right to interpret, modify, revise, amend, or update any provision, rule, regulation, curriculum structure, evaluation scheme, academic requirement, course content, credit distribution, research component, dissertation guidelines, or operational procedures related to the **M.Tech (Software Engineering)** program from time to time in accordance with University regulations, UGC/AICTE guidelines, statutory requirements, technological advancements, software industry developments, research trends, and academic needs.

The curriculum and academic regulations shall be reviewed periodically to ensure their relevance to emerging areas of Software Engineering, including Software Architecture, Cloud Computing, Distributed Systems, Data Science, High Performance Computing, Software Quality Assurance, Software Project Management, Artificial Intelligence applications, and other evolving technologies.

In case of any ambiguity, inconsistency, or dispute regarding the interpretation of any provision contained in these Academic Regulations and Curriculum documents, the decision of the University and/or the Competent Authority shall be final and binding on all stakeholders.

Any amendments, additions, revisions, modifications, or updates approved by the appropriate academic, administrative, or statutory bodies of the University shall automatically become applicable to the M.Tech (Software Engineering) program from the date of approval or implementation as notified by the University.

The University may also introduce new courses, modify existing courses, revise credit structures, update research and dissertation requirements, alter evaluation methodologies, or incorporate emerging technologies and industry practices to maintain academic excellence, industry relevance, and compliance with national higher education policies and regulatory frameworks.

Annexure I: Detail Syllabus

GANDHINAGAR INSTITUTE OF TECHNOLOGY

MTech in Software Engineering

(Computer Engineering Department

Semester 1

Subject Code: 10070101	**Subject Title:** Advanced Software Engineering
Pre-requisite: Basic computer knowledge, knowledge of computer architecture and application development	

Course Objective:

- To introduce the fundamental concepts of software engineering.
- To build an understanding on various phases of software development.
- To introduce various software process models.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	**Introduction to software engineering:** Software, Evolving role of software, Three "R"-Reuse, Reengineering and Retooling, Software Process, Software engineering practice, Software Myths.	3	15
2	**Software Process Models:** Waterfall Model, Incremental Process model: Iterative approach, RAD, JAD model, Evolutionary Process Model: Prototype and Spiral Model, Concurrent Development Model,	4	15
3	**Agile Development:** Concept of Agility and Cost of change, Agile Process Model, Extreme programming, Scrum. Agile Modeling and tool set for Agile	7	20
4	**Software Requirement Analysis, Specification Design:** Types of Requirement, Feasibility Study, Requirement Analysis and Design, HIPO Chart, Warnier Orr Diagram, Facilitated Application Specification Technique (FAST), Design Concepts and Principles, User Interface Design.	8	10

5	**Software Quality:** Software and System Quality Management: Overview of ISO 9001, SEI Capability Maturity Model, McCalls Quality Model, Six Sigma, Formal Technical Reviews, Tools and Techniques for Quality Control, Pareto Analysis, Statistical Sampling, Quality Control Charts and the seven Run Rule.	4	10
6	**Software Estimation:** Size Estimation: Function Point (Numericals). Cost Estimation: COCOMO (Numericals), COCOMO-II (Numericals). Earned Value Management.	4	10

7	**DevOps:** Overview, Problem Case Definition, Benefits of Fixing Application Development Challenges, DevOps Adoption Approach through Assessment, Solution Dimensions, What is DevOps?, DevOps Importance and Benefits, DevOps Principles and Practices, 7 C's of DevOps Lifecycle for Business Agility, DevOps and Continuous Testing, How to Choose Right DevOps Tools, Challenges with DevOps Implementation, Must Do Things for DevOps, Mapping My App to DevOps Assessment, Definition, Implementation, Measure and Feedback	4	10
8	**Case Study:** Agile methodology, Software Project Estimation	2	10

Course Outcome: After completion of this course, students will be able to do the following:

- To understand the process of Software Engineering
- To conceptualize the Software Development Life Cycle (SDLC) models.
- Recognize how to ensure the quality of software products, different quality standards, and software review techniques.

List of References:

1. Roger S. Pressman, Software Engineering: A practitioner's approach, McGraw Hill publication, Eighth edition, 2014.
2. Jack T Marchewka, Information Technology Project Management, Wiley India publication.
3. Douglas Bell, Software Engineering for students: A Programming Approach, Pearson publication.
4. Ian Sommerville, Software Engineering, University of Lancaster, Pearson Education, Seventh edition, 2004.
5. S.A. Kelkar, Software Project Management: A concise study, PHI, Third edition, 2012.
6. Walker Royce, Software Project Management : A unified framework, Pearson Education, 1998

Open e-Resource:

1. https://onlinecourses.swayam2.ac.in/cec22_cs24/preview
2. https://onlinecourses.nptel.ac.in/noc22_cs106/preview
3. https://onlinecourses.swayam2.ac.in/cec21_cs21/preview

List of Suggested Experiments:

1. An introduction to software engineering.
2. Development of SRS document, Design document for the selected project.
3. Development of DFD, data dictionary, E-R diagram, structured chart for the project.
4. To study and draw various UML diagrams.
5. To illustrate the use of class diagrams.
6. To draw an activity diagram and use case diagram for ATM and Library Management System.
7. Draw Object Diagram for ATM System.
8. Development of State Transition Diagram.
9. Draw ER Diagram for Hospital Management System

Subject Code: 10070102	**Subject Title:** Advance Data Structures & Algorithms
Pre-requisite: Basic programming Skills, UG Level Knowledge of Data Structures and Algorithms	

Course Objective:

- The student should be able to choose appropriate data structures, understand the ADT/libraries, and use it to design algorithms to solve a specific problem with necessary mathematical abstraction.
- The student should be able to analyze the efficiency and proof of correctness to solve algorithmic problems through advanced paradigms and data structures.
- Introduce students to the advanced methods of designing and analyzing algorithms.
- The student should be able to understand different classes of problems based on their computation difficulties and choose appropriate algorithms to solve a specific problem.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**

1	**Advanced Data Structures:** Dictionaries: Definition, Dictionary Abstract Data Type, Implementation of Dictionaries; Hashing: Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing; Skip Lists: Need for Randomizing Data Structures and Algorithms, Search and Update Operations on Skip Lists, Probabilistic Analysis of Skip Lists, Deterministic Skip Lists; Recent Trends in Hashing	8	25
2	**Trees:** Binary Search Trees, AVL Trees, Red Black Trees, 2-3 Trees, B-Trees, Splay Trees, Recent Trends in Trees	4	10
3	**Text Processing:** Sting Operations, Brute-Force Pattern Matching, The Boyer-Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard Tries,	8	20
	Compressed Tries, Suffix Tries, The Huffman Coding Algorithm, The Longest Common Subsequence Problem (LCS), Applying Dynamic Programming to the LCS Problem.		
4	**Graphs & Algorithms:** Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkasra's), computation of strongly connected components using DFS and BFS, emphasis on correctness proof of the algorithm and time/space analysis, introduction and example of amortized analysis. Paths and Circuits: Euler Graphs, Hamiltonian Paths & Circuits; Cut-sets, Planar Graphs, Isomorphism, Graph Coloring, Covering and Partitioning, Topological sort	8	20
5	**Algorithms Terminology:** Introduction to NP-completeness, Examples and proof of NP-hardness and NP-completeness; Introduction to Approximation algorithms, Randomized Algorithms, Interior Point Method, Advanced Number Theoretic Algorithm, Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures.	8	25

Course Outcome: After completion of this course, student will be able to

- Master a variety of advanced abstract data type (ADT) and data structures and their implementations.
- Develop and analyze algorithms for red-black trees, B-trees, Splay trees and for text processing applications.
- Analyze the complexity and performance of various algorithms.
- Solve the specific set of problems by categorizing them into various classes according to their complexity and suitable data structure.

List of References:

1. Cormen, Leiserson, Rivest, & Stein, “Introduction to Algorithms”, The MIT Press (2009), 3rd Edition.
2. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C++”, 2nd Edition, Pearson, 2004
3. M T Goodrich, Roberto Tamassia, “Algorithm Design”, John Wiley, 2002.
4. Aho, Hopcroft, Ullman, “"The Design and Analysis of Computer Algorithms", Pearson.
5. Kleinberg and Tardos, “Algorithm Design", Pearson.

Open e-Resource:

1. https://www.coursera.org/specializations/algorithms
2. https://visualgo.net/bn
3. https://online.stanford.edu/courses/cs161-design-and-analysis-algorithms

List of Suggested Experiments:

1. Write a program to implement any hashing methods to perform insertion, deletion and search operations for Hash Table Management.
2. Implement a Dictionary as an Abstract Data Type using suitable Data Structure.
3. Write a program which creates Binary Search Tree. And also implement recursive and non-recursive tree traversing methods inorder, preorder and post-order for the BST.
4. Write a program which creates AVL Tree. Implement Insert and Delete Operations in AVL Tree. (Note that each time the tree must be balanced.)
5. Implement a program for String Matching using Boyer-Moore Algorithm on a text file content.
6. Implement a program for String Matching using Knuth-Morris-Pratt Algorithm on a text file content.
7. Implement Dijkstra’s algorithm. Find an application that can best be solved by Dijkstra’s algorithm.
8. Implement a program for strongly connected components by using BFS or DFS.
9. Implement One Dimensional and Two Dimensional Range Searching in any language.
10. Write a program to implement heap and radix sort.

Subject Code: 10070103	**Subject Title:** Data Science
Pre-requisite: Excel, Programming Concepts, Basics of Math’s and Statistics	

Course Objective:

- To provide fundamentals of Data Science with the help of math and statistics.
- To develop skills to handle enormous data need to be analyzed for making better decisions.
- Helps in visualizing data in such a way that it becomes easy for the organizations to decide upon their business strategies.

Teaching Scheme (Hours per week)	**Evaluation Scheme (Marks)**

Lecture	Tutorial	Practical	Credit	Theory		Practical		Total
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	**Introduction to Business Analytics** Why Analytics, Business Analytics: The Science of Data-Driven Decision Making, Descriptive Analytics, Predictive Analytics, Prescriptive Analytics Descriptive, Predictive and Prescriptive Analytics Techniques, Big Data Analytics, Web and Social Media Analytics, Machine Learning Algorithms, Framework for Data-Driven Decision Making Analytics, Capability Building Roadmap for Analytics, Capability Building Challenges in Data-Driven Decision Making and Future	3	10
2	**Statistics** Important statistical concepts used in data science, Difference between population and sample, Types of variables, Measures of central tendency, Measures of variability, Coefficient of variance, Skewness and kurtosis **Inferential Statistics** Normal distribution, Test hypotheses, Central limit theorem, Confidence interval, T-test, Type I and II errors, Student's T distribution	8	20
3	**Introduction to Probability** Introduction to Probability Theory, Probability Theory – Terminology, Fundamental Concepts in Probability – Axioms of Probability, Application of Simple Probability Rules – Association Rule Learning Bayes' Theorem, Random Variables Probability Density Function (PDF) and Cumulative Distribution Function (CDF) of a Continuous Random	11	25
	Variable, Binomial Distribution, Poisson Distribution, Geometric Distribution, Parameters of Continuous Distributions, Uniform Distribution, Exponential Distribution, Chi-Square Distribution, Student's t-Distribution, F-Distribution		
4	**Regression and ANOVA** Regression, ANOVA, R square, Correlation and causation	5	15
5	**Exploratory Data Analysis** Data visualization, Missing value analysis, The correction matrix, Outlier detection analysis	5	15

6	**Supervised Machine Learning** Python Scikit tool, Neural networks, Support vector machine, Logistic and linear regression, Decision tree classifier	5	15

Course Outcome: After completion of this course, student will be able to

- Understand the fundamental concepts of Data Science.
- Describe the various application areas.
- Explain the concepts of Statistics and Probability.
- Solve regression and classification problems.

List of References:

1. Business Analytics: The Science of Data-Driven Decision Making by U. Dinesh Kumar, Wiley India
2. V.K. Jain, Data Science & Analytics, Khanna Book Publishing, New Delhi
3. Data Science For Dummies by Lillian Pierson, Jake Porway
4. Doing Data Science by Cathy O'Neil and Rachel Schutt O'Reilly publication
5. Practical Data Science with Jupyter Author: Prateek Gupta, BPB publications

Open e-Resource:

1. www.analyticsvidhya.com
2. www.kaggle.com
3. https://nptel.ac.in/courses/106106179
4. https://nptel.ac.in/courses/106105186
5. https://nptel.ac.in/courses/106106182

List of Suggested Experiments:

Consider a student dataset with appropriate fields and implement the following in Python or R-

1. Identify the data types and perform descriptive analysis.
2. Implement a method to find out variation in data. For example, the difference between highest and lowest marks in each subject semester wise.
3. Plot the graph showing result of current stock of any five different companies.
4. Plot the graph showing the geographical location of medical stores near you.
5. Plot the graph showing number of male and female students in a particular and/or all the current semesters.
6. Implement a method to treat missing value for gender and missing value for marks.
7. Implement linear regression to predict the result of student of any semester.
8. Implement linear regression to predict the trends on any social media.
9. Implement logistic regression and decision tree to classify the student as average or clever.
10. Implement logistic regression to find the sentiment of a comment as sad or happy.

Subject Code: 10070104	**Subject Title:** Data Warehousing & Data Mining
Pre-requisite: An understanding of basic database concepts such as schema, ER model, structured query	

language, etc.

Course Objective:

The main objectives for offering the course Data Warehousing and Data Mining are:

- To understand the concept of Data Warehousing and mining.
- To have hands on experience on various Data Mining tools.
- To have insight of Date Preprocessing, Classification, Association and Prediction and various data mining related advance topics like text mining, web mining etc.

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
				Theory		Practical		
Lecture	Tutorial	Practical	Credit	University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	Total
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	Topic	Total Hours	Weight (%)
1	**Overview and concepts Data Warehousing**: What is data warehousing - The building Blocks?, Defining Features – Data warehouses and data marts, Overview of the components, Metadata in the data warehouse, Need for data warehousing, Basic elements of data warehousing, Trends in data warehousing	4	10
2	**OLAP in Data Warehouse:** OLAP (Online analytical processing) definitions, Difference between OLAP and OLTP, Dimensional analysis - What are cubes?, Drill-down and roll-up - slice and dice or rotation, OLAP models, ROLAP versus MOLAP, defining schemas: Stars, snowflakes and fact constellations	4	10
3	**Introduction to data mining (DM):** DM Functionalities, Classification of DM Systems, Issues in DM – KDD Process	4	10
4	**Data Preprocessing:** Why to preprocess data?, Data cleaning: Missing Values, Noisy Data, Data Integration and transformation, Data Reduction: Data cube aggregation, Dimensionality reduction, Data Compression, Numerosity Reduction, Data Mining Primitives, Languages and System Architectures: Task relevant data, Kind of Knowledge to be mined, Discretization and Concept Hierarchy.	6	15
5	**Concept Description and Association Rule Mining:** What is concept description?, Data Generalization and summarization-based characterization, Attribute relevance, class comparisons Association Rule Mining, Market basket analysis- basic concepts, Finding frequent item sets: Apriori algorithm - generating rules	6	20

6	**Classification and Prediction:** What is classification and prediction?, Issues, Classification using Decision trees, Linear and nonlinear regression, Introduction of tools such as DBMiner /WEKA/DTREG DM Tools	6	20
7	**Advance topics:** Introduction of clustering, Spatial mining, Web mining, Text mining	6	15

Course Outcome:

- After completion of this course, students will be able to do the following:
- Students will able to understand important of data mining and its various concepts like data preprocessing, various classification algorithms etc.
- Students will use knowledge of the subject in higher semester for subjects like Advance Data Warehousing and Data Mining concepts.
- Student will be able to develop a reasonably sophisticated data mining application.
- Perform the preprocessing of data and apply mining techniques on it.

List of References:

1. J. Han, M. Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann
2. Paulraj Ponnian, "Data Warehousing Fundamentals", John Willey.
3. M. Kantardzic, "Data mining: Concepts, models, methods and algorithms, John Wiley &Sons Inc.
4. M. Dunham, "Data Mining: Introductory and Advanced Topics", Pearson Education.
5. Pieter Adriaans, Dolf Zantinge, "Data Mining", Pearson Education Asia

Open e-Resource:

1. https://onlinecourses.swayam2.ac.in/cec22_cs06/preview
2. https://onlinecourses.nptel.ac.in/noc22_mg67/preview
3. http://www.dataminingblog.com
4. http://www.kdnuggest.com

List of Suggested Experiments:

Perform/simulate the following suggested practicals using any open source tool like Weka, PowerBI, Orange etc.

1. Perform the basics of Analytical Analysis.
2. Survey of Different Data Mining Tools.
3. Study of open source tools for Data mining.
4. Perform classification.
5. Perform clustering.
6. Learn to use Pivot Tables in Excel 2010 to Organize Data.
7. Study of the Apriori Algorithm (Finding Frequent Item Sets Using Candidate Generation).
8. Decision Tree Learning.
9. Design & create cube by identifying measures & dimensions for star schema.

10. Design & create cube by identifying measures & dimensions for snowflake schema.

Subject Code: 10070105	**Subject Title: High Performance Computing**
Pre-requisite: Linear Algebra and Numerical Methods, Parallel Algorithms	

Course Objective:

- To solve a single problem faster often requires the algorithm to introduce parallelism, and then programming in a language that can express this parallelism.
- Efficient knowledge of machine arithmetic, computer architecture, and memory hierarchies.
- High performance is an important aspect of high performance scientific computing that use of basic tools and techniques to improve your efficiency as a computational scientist.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	**Introduction: Computational Science and Engineering**: Computational Science and Engineering Applications; characteristics and requirements, Review of Computational Complexity, Performance: metrics and measurements, Granularity and Partitioning, Locality: temporal/spatial/stream/kernel, Basic methods for parallel programming, Real-world case studies (drawn from multiscale, multi-discipline applications)	6	15
2	**High-End Computer Systems:** Memory Hierarchies, Multi-core Processors: Homogeneous and Heterogeneous, Shared-memory Symmetric Multiprocessors, Vector Computers, Distributed Memory Computers, Supercomputers and Petascale Systems, Application Accelerators / Reconfigurable Computing, Novel computers: Stream, multithreaded, and purpose-built	6	20
3	**Parallel Algorithms**: Parallel models: ideal and real frameworks, Basic Techniques: Balanced Trees, Pointer Jumping, Divide and Conquer, Partitioning, Regular Algorithms: Matrix operations and Linear Algebra, Irregular Algorithms: Lists, Trees, Graphs, Randomization: Parallel Pseudo- Random Number Generators, Sorting, Monte Carlo techniques	6	20

4	**High-End Computer Systems**: Memory Hierarchies, Multi-core Processors: Homogeneous and Heterogeneous, Shared-memory Symmetric Multiprocessors, Vector Computers, Distributed Memory Computers, Supercomputers and Petascale Systems, Application Accelerators / Reconfigurable Computing, Novel computers: Stream, multithreaded, and purpose-built	4	15
5	**Parallel Programming**: Revealing concurrency in applications, Task and Functional Parallelism, Task Scheduling, Synchronization Methods, Parallel Primitives (collective operations), SPMD Programming (threads, OpenMP, MPI), I/O and File Systems, Parallel Matlabs (Parallel Matlab, Star-P, Matlab MPI), Partitioning Global Address Space (PGAS) languages (UPC, Titanium, Global A	8	20
6	**Achieving Performance:** Measuring performance, Identifying performance bottlenecks, Restructuring applications for deep memory hierarchies, Partitioning applications for heterogeneous resources, using existing libraries, tools, and frameworks	6	10

Course Outcome:

- After completion of this course, student will be able to
- Have an understanding of the various paradigms of high performance computing and their potential for performance and programmability.
- Have an awareness of the modern field of computational science and engineering and of the impact of high performance computing on science and industry.
- Able to reason about the accuracy of mathematical and numerical models of real physical phenomena.
- Illustrate mapping of applications to high-performance computing systems

List of References:

1. Introduction to Parallel Computing, AnanthGrama, Anshul Gupta, George Karypis, and Vipin Kumar, 2nd edition, Addison-Welsey, 2003.
2. G.E. Karniadakis, R.M. Kirby II, Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and their Implementation, Cambridge University Press,2003.
3. Wilkinson and M. Allen, Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, 2/E, Prentice Hall, 2005.
4. M.J. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill, 2004.
5. Using OpenMP: Portable Shared Memory Parallel Programming by Chapman, Jost, and vander Pas.
6. David Culler Jaswinder Pal Singh,"Parallel Computer Architecture: A hardware/Software Approach", Morgan Kaufmann, 1999.

Open e-Resource:

1. https://nptel.ac.in/courses/112105293
2. https://www.ibm.com/in-en/topics/hpc
3. https://aws.amazon.com/hpc

List of Suggested Experiments:

1. To study the basic commands of Linux.
2. To establish Beowulf Cluster using MPI (Message Passing Interface) Library.
3. To implement efficient parallel debugging for MPI, Threads, and Beyond.
4. To execute a sample program using openMP.
5. Installation and Configuration of Alchemi Grid and Running a sample application on Alchemi Grid Environment.
6. To Run two sample programs using GridSim Toolkit.
7. To Implement Cloud Simulation Toolkit with example.
8. To Setup Hadoop platform.
9. Run sample program using Hadoop Framework

Subject Code: 10070106	**Subject Title:** Image Processing
Pre-requisite: Physics and fundamentals of signal processing	

Course Objective:

- To become familiar with digital image fundamentals.
- To get exposed to simple image enhancement techniques in Spatial and Frequency domain.
- To learn image restoration, interpolation, and filtering operations.
- To learn various image processing operations like edge detection, segmentation; color-based processing, region and boundary detection, morphological operations.
- To become familiar with image compression and recognition methods.
- To get exposure to digital image processing in detail with enhanced knowledge in various application areas.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**

1	**Digital image fundamentals:** Examples of fields that use image processing, Light and Electromagnetic spectrum, Components of Image processing system, Image formation and digitization concepts, Neighbors of pixel adjacency connectivity, regions and boundaries, Distance measures.	3	15
2	**Spatial domain Image Enhancements:** Basic gray level transformations, Histogram processing, using arithmetic and Logic operations, smoothing and Sharpening spatial filters.	4	12
3	**Frequency domain Image Enhancements:** Introduction to the Fourier transform and frequency domain concepts, smoothing frequency-domain filters, Sharpening frequency domain filters- Ideal, Butterworth and Gaussian filters.	4	12
4	**Image Restoration:** Various noise models, image restoration using spatial domain filtering, image restoration using frequency domain filtering, Estimating the degradation function, Inverse filtering.	4	18

5	**Colour Image processing:** Colour fundamentals, Colour models, Colour transformation, Smoothing and Sharpening, Colour segmentation.	4	5
6	**Wavelet and Multi-resolution processing:** Image pyramids, Multi-resolution expansion, wavelet transform.	3	8
7	**Image compression:** Introduction, Image compression model, Error-free compression, Lossy compression, image compression standards	4	5
8	**Morphological Image processing:** Preliminaries, Dilation and Erosion, Opening and Closing, Hit-or-Miss Transformation, Basic morphological algorithms for boundary extraction, Region filling, extraction of connected components, thinning and thickening.	4	15
9	**Image segmentation:** Detection of discontinuities, Edge linking and boundary detection, thresholding.	4	5
10	**Image processing applications:** Target detection, object recognition, feature extraction and matching, writer identification system, digital watermarking, medical and Industrial applications, Image Morphing and Image Fusion	2	5

Course Outcome: After completion of this course, student will be able to

- Understand the basic image enhancement techniques in spatial & frequency domains.

- Understand the various kind of noise present in the image and how to restore the noisy image.
- Understand the basic multi-resolution techniques and segmentation methods.
- To apply these concepts for image handling in various fields.

List of References:

1. Digital Image Processing, by Rafel C. Gonzalez and Richard E. Woods, Pearson Education, latest edition.
2. Digital Image Processing by Bhabatosh Chanda and Dwijesh Majumder, PHI
3. Fundamentals of Digital Image Processing by Anil K Jain, PHI
4. Digital Image Processing Using Matlab, Rafel C. Gonzalez and Richard E. Woods, Pearson Education

Open e-Resource:

1. http://www.scilab.org/
2. http://www.imageprocessingplace.com/
3. http://ocw.mit.edu/resources/res-6-008-digital-signal-processing-spring-2011/video-lectures/
4. http://imagingbook.com/
5. https://nptel.ac.in/courses/117105135

List of Suggested Experiments:

Perform the following practicals using any of the tool like MATLAB/OpenCV/Python image processing tools/libraries.

1. To extract different Attributes of an Image.
2. Image Negation.
3. Power Law Transformation.
4. Histogram Mapping and Equalization.
5. Image Smoothening and Sharpening.
6. Edge Detection using Sobel, Prewitt and Roberts Operators.
7. Morphological Operations on Binary Images.
8. Pseudo Coloring.
9. Chain Coding.
10. DCT/IDCT Computation.

Subject Code: 10070107	**Subject Title: Wireless Sensor Networks**
Pre-requisite: Knowledge of Computer Networks, Basic computer languages, Microprocessor, Algorithms & Communication Basics	

Course Objective:

- Understand standards and communication protocols for wireless sensor networks.
- Functionality of the node architecture and use of sensors for various applications.
- Get familiar with architectures, functions and performance of wireless sensor networks systems and platforms

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
				Theory		Practical		
Lecture	Tutorial	Practical	Credit	University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	Total
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	Topic	Total Hours	Weight (%)
1	**Introduction** – challenges and constraints, Applications	1	5
2	**Single Node Architecture** – Hardware components, Energy consumption of sensor nodes, Operating systems and execution environments, examples of sensor nodes **Network Architecture** – Sensor network scenarios, optimization goals, design principles, service interfaces, gateway concepts	7	20
3	**Physical layer** – Introduction, wireless channel and communication fundamentals, physical layer and transceiver design considerations in WSNs	4	10
4	**MAC protocols** – Fundamentals of MAC protocols, Low duty cycle protocols and wakeup concepts, contention-based protocols, schedule-based protocols, The IEEE 802.15.4 MAC protocol **Link Layer Protocol** – Tasks and requirements, Framing, Link Management	8	20
5	**Naming and addressing** – Fundamentals, address and name management, Assignment of MAC address, Distributed assignment of locally unique addresses, content based and geographic addressing.	2	10
6	**Time Synchronization, Localization and positioning** - Time synchronization problem, protocols, properties of localization and positioning procedures, single hop localization, Positioning in multihop environments	4	15
7	**Routing protocols** – Forwarding and routing, MANET protocols, gossiping and agent based unicast forwarding, Energy-efficient unicast, Broadcast and multicast, geographic routing, Mobile nodes	6	15
8	**IoT** – What is IoT, WSN and IoT, Applications, Architecture and reference, routing, Data aggregation, data centric storage	4	10

Course Outcomes:

After completion of course, students would be able to:

- Describe and explain radio standards and communication protocols for wireless sensor networks.
- Explain the function of the node architecture and use of sensors for various applications.
- Be familiar with architectures, functions and performance of wireless sensor networks systems and platforms.

List of References:

1. Holger Karl and Andreas Willig, "Protocols and Architectures for wireless sensor networks, WILEY
2. W. Dargie and C. Poellabauer, "Fundamentals of Wireless Sensor Networks –Theory and Practice", Wiley 2010
3. KazemSohraby, Daniel Minoli and TaiebZnati, "wireless sensor networks -Technology, Protocols, and Applications", Wiley Interscience 2007

Open e-Resource:

1. https://nptel.ac.in/courses/106105160
2. https://www.udemy.com/course/wireless-sensor-networks-electronics-telecommunication-f/

List of Suggested Experiments:

1. Implement a campus sensor communication application for assignment file transfer using Bluetooth.
2. Implement a campus sensor communication application for assignment file transfer using wireless LAN.
3. Implement a campus sensor communication application for assignment file transfer using ZigBee.
4. Implement a campus sensor communication application for assignment file transfer using WiMax/hotspot.
5. Implement TinyOS. Perform analysis of various routing protocols
6. Perform analysis of various routing protocols on sensor network simulator.
7. Perform analysis of various MAC protocol on sensor network simulator and find how affects the energy efficiency.

Subject Code: 10070108	**Subject Title: Data Security & Access Control**
Pre-requisite: Basic of Cryptography and Information Security	

Course Objective:

- The objective of the course is to provide fundamentals of data security.
- Various access control techniques mechanisms.
- Application areas of access control techniques.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	**Introduction to Security:** Need for security, Security approaches, Principles of security, Types of attacks, Encryption Techniques: Plaintext, Cipher text, Substitution & Transposition Techniques, Encryption & Decryption, and Types of attacks, Key range & Size.	5	15
2	**Introduction to Data Security:** Threats, Vulnerabilities and Risks, Types of Data Security, Data Security Strategies, Data Security Trends, Data Security Technologies and frameworks	5	15
3	I**ntroduction and fundamentals of Access Control**: Access Control Policy and Terminology, Access Control models: Mandatory Access Control (MAC), Discretionary Access Control (DAC), Non- Discretionary Access Control, Role-Based Access Control (RBAC), Rule-Based Access Control (RB-RBAC), Capabilities and Limitations of Access Control Mechanisms: Access Control List (ACL) and Limitations, Capability List and Limitations.	8	25
4	**Role-Based Access Control (RBAC) and Limitations**: Core RBAC, Hierarchical RBAC, Statically Constrained RBAC, Dynamically Constrained RBAC, Limitations of RBAC, Comparing RBAC to DAC and MAC Access control policy, Integrating RBAC with enterprise IT infrastructures: RBAC for WFMSs, RBAC for UNIX and JAVA environments.	10	25
5	**Case Studies**: Role-Based Access Control (RBAC) systems, Multi-line Insurance Company, Single Sign On(SSO), Secure Inter-branch Payment Transactions, Secret Splitting, Secure Multiparty Calculation.	8	20

Course Outcome: After completion of this course, student will be able to

- A strong grasp of the basic concepts underlying classical and modern cryptography, and the fundamentals.
- Students are able to design and implement access control rules, to assign privileges and protect data.
- Students learn and practice various access control theories and techniques including mandatory access control, discretionary access control, and role‐based access control.
- They will further be able to assess the strengths and weaknesses of various access control models and to analyze their behavior.

List of References:

1. Principles of Information Security: Michael E. Whitman, Herbert J. Mattord, CENGAGE Learning, 5th Edition.

2. Role Based Access Control: David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli.
3. Access Control, Authentication, and Public Key Infrastructure by Bill Ballad, Tricia Ballad, Erin Banks

Open e-Resource:

1. Access Control: Securit, Identity Management and Trust Models: https://shazkhan.files.wordpress.com/2010/10/access-control-systems.pdf
2. https://www.wireshark.org/
3. https://nmap.org/
4. https://www.openvas.org/

List of Suggested Experiments:

1. Perform the encryption and decryption operations using any algorithm.
2. Study of X.509 Certificate
3. Implement Digital Signature algorithm.
4. Perform various attacks through DVWA.
5. Perform Network vulnerability scanning through OpenVAS.
6. Perform Automated SQL injection with SqlMAP.
7. Perform Packet tracing through Wireshark & sniffing packets through it.
8. Perform analysis on Mandatory Access Control (MAC) models in cyber security.
9. Perform analysis on Discretionary Access Control (DAC) models in cyber security.
10. Perform analysis on Role-Based Access Control (RBAC) models in cyber security.

Subject Code: 10000151	**Subject Title:** Research Methodology and IPR
Pre-requisite: Nil	

Course Objective:

- To give an overview of the research methodology and explain the technique of defining a research problem.
- To explain the functions of the literature review in research.
- To explain carrying out a literature search, its review, developing theoretical and conceptual frameworks and writing a review.
- To explain various research designs and their characteristics.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
ecture	**itorial**	**actical**	**Credit**	**University Assessment**	**itinuous Assessment**	**University Assessment**	**itinuous Assessment**	**Total**
1	1	2	3	0	0	80	20	100

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations	5	17
2	Effective literature studies approaches, analysis Plagiarism, Research ethics,	4	15
3	Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee	5	17
4	Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.	5	17
5	Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.	5	17
6	New Developments in IPR: Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc. Traditional knowledge Case Studies, IPR and IITs.	5	17

Course Outcome: After completion of this course, student will be able to

- Understand research problem formulation.
- Analyze research related information
- Follow research ethics
- Understand that today's world is controlled by Computer, Information Technology, but tomorrow world will be ruled by ideas, concept, and creativity.
- Understanding that when IPR would take such important place in growth of individuals & nation, it is needless to emphasis the need of information about Intellectual Property Right to be promoted among students in general & engineering in particular.
- Understand that IPR protection provides an incentive to inventors for further research work and investment in R & D, which leads to creation of new and better products, and in turn brings about, economic growth and social benefits.

List of References:

1. Stuart Melville and Wayne Goddard, "Research methodology: an introduction for science & engineering students'"
2. Wayne Goddard and Stuart Melville, "Research Methodology: An Introduction"

3. Ranjit Kumar, 2 nd Edition, “Research Methodology: A Step by Step Guide for beginners”
4. Halbert, “Resisting Intellectual Property”, Taylor & Francis Ltd ,2007.
5. Mayall , “Industrial Design”, McGraw Hill, 1992.
6. Niebel , “Product Design”, McGraw Hill, 1974
7. Asimov, “Introduction to Design”, Prentice Hall, 1962.
8. Robert P. Merges, Peter S. Menell, Mark A. Lemley, “ Intellectual Property in New Technological Age”, 2016.
9. T. Ramappa, “Intellectual Property Rights Under WTO”, S. Chand, 2008

Open e-Resource:

1. NPTEL: https://onlinecourses.nptel.ac.in/noc19_ge21/preview
2. NPTEL: https://onlinecourses.nptel.ac.in/noc22_ge08/preview

Subject Code: 10000153	**Subject Title:** Constitution of India
Pre-requisite:- Nil	

Course Objective:

- To provide basic information to students about the Indian constitution.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
2	0	0	0	0	50	0	0	50

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	**Introduction to Indian Constitution:** Meaning of Indian Constitution, Constituent Assembly, The salient features of Indian Constitution.	3	10
2	**Fundamentals Rights:** Fundamentals Rights and Environmental protection, Fundamentals Duties, The directives Principles of State Policy, Special rights created in the constitution of Dalits, Backward classes, Women and Children and religious and linguistic minorities.	5	20
3	**Union Government:** Union Government, Union Legislative, Lok Sabha and Rajya Sabha (With powers and Functions), Union Executive, President of India (With powers and Functions), Prime Minister of India (With powers and Functions)	5	20

4	**State Government:** State Government, State Legislative, Powers and Functions of the State Legislature, State Executive, Governor of the State (With powers and Functions), The Chief Minister of the State (With powers and Functions)	5	25
5	**Election provisions, Emergency provisions, Amendment of the constitution:** Election Commission of India-composition, powers and functions and electoral process. Types of emergency-grounds, procedure, duration and effects. Amendment of the constitution- meaning, procedure and limitations.	5	25

Course Outcome: After completion of this course, student will be able to

- Understand and explain the features of Indian Constitution.
- Exercise his fundamental rights in proper sense at the same time identifies his responsibilities in national building.
- Analyse the Indian political system, the powers and functions of the Union, State and Local Governments.
- Understand Electoral Process, Emergency provisions and Amendment procedure**.**

List of References:

1. Indian Constitution by D Srinivasan, Himalaya Publishing House.
2. Indian Constitution: C. Jamnadas Publication
3. M.V. Pylee, "Introduction to the Constitution of India",4th Edition, Vikas publication,2005.
4. Durga Das Basu (DD Basu), "Introduction to the constitution of India", (Student Edition), 19th edition, Prentice-Hall EEE, 2008.

Open e-Resource:

1. https://legislative.gov.in/constitution-of-india
2. https://www.india.gov.in/my-government/constitution-india

Semester II

Subject Code: 10070201	**Subject Title:** Software Project Management
Pre-requisite: Knowledge of structure programming language and Application development	

Course Objective:

- To introduce the fundamental concepts of software project management
- To build an understanding on various phases of software project management and maintenance.
- To introduce various techniques for estimating the time, cost and quality

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	**Introduction to Software Project Management** (SPM): Rationale, Software Projects Vs other types of Projects, Contract Management and Technical Project Management, Activities Covered by SPM, Plans, Methods and Methodologies, Categorizing Software Projects, Project Charter, Stakeholders, Setting Objectives, Project Success and Failure, Management Control, Project Management Life Cycle, Traditional versus Modern Project Management Practices	4	15
2	**Project Planning:** Tasks in Project Planning; Work Breakdown Structures (WBS), Planning Methods, Selecting Project Approach, SDLC, Software Processes and Process Models, Choice of Process Models, A Generic Project Model, Software Cost Estimation; COCOMO Model; Budgeting.	6	15
3	**Project Scheduling, Monitoring & Control:** Scheduling Techniques, Program Evaluation and Review Technique (PERT), Gantt Chart, Critical Path Method (CPM), Automated Tools. Project Status Reporting; Project Metrics; Earned Value Analysis (EVA); Project Communication Plan & Techniques; Steps for Process Improvement.	8	20

4	**Project Procurement management:** Relationship between people and Effort: Staffing Level Estimation, Effect of schedule Change on Cost, Degree of Rigor & Task set selector, Project Schedule, Schedule Control, CPM (Numericals), Basic Planning Purchases and Acquisitions, Planning Contracting, Requesting Seller Responses, Selecting Sellers, Out Sourcing: The Beginning of the outsourcing phenomenon, Types of outsourcing relationship, The realities of outsourcing, Managing the outsourcing relationship	6	10

5	**Software Quality:** Software and System Quality Management: Overview of ISO 9001, SEI Capability Maturity Model, McCalls Quality Model, Six Sigma, Pareto Analysis, Statistical Sampling, Quality Control Charts and the seven Run Rule.	5	10
6	**Human Resource Management:** Human Resource Planning, Acquiring the Project Team: Resource Assignment, Loading, And Leveling, Developing the Project Team: Team Structures, Managing the Project Team, Change management: Dealing with Conflict & Resistance Leadership & Ethics.	5	10
7	**Software Risk Management and Reliability issues:** Risk Management: Identify IT Project Risk, Risk Analysis and Assessment, Risk Strategies, Risk Monitoring and Control, Risk Response and Evaluation. Software Reliability: Reliability Metrics, Reliability Growth Modeling.	4	10
8	**Case Study:** Project Management Practices	2	10

Course Outcome: After completion of this course, students will be able to do the following:

- Describe and determine the purpose and importance of a software project and project management practices.
- Estimate and evaluate project cost and schedules and determine risk management approaches.
- Define and evaluate quality assurance measures

List of References:

1. Information Technology Project Management by Jack T Marchewka Wiley India publication.
2. Software Engineering Project Management by Richard H. Thayer Wiley India Publication.
3. S.A. Kelkar, Software Project Management: A concise study, PHI, Third edition, 2012.
4. Bob Hughes and Mike Cotterell, "Software Project Management ", Tata McGraw Hill, 4th edition, 2006
5. Walker Royce, "Software Project Management", Pearson Education, 2005
6. Roger S. Pressman , "Software Engineering – A Practitioner's approach", Tata McGraw Hill, 2009

Open e-Resource:

1. https://onlinecourses.nptel.ac.in/noc22_cs107/preview

List of Suggested Experiments:

1. An introduction to software project management.
2. Development of SRS document, Design document for the selected project.
3. Development of DFD, data dictionary, E-R diagram, structured chart for the project.
4. Estimate project cost and prepare project schedule.
5. Evaluate risk management approaches suitable for the project.
6. Design test suite to ensure software quality.

Subject Code: 10070202	**Subject Title: Research Review & Seminar**
Pre-requisite:	

Course Objective:

- To learn to review and assess scientific literature critically.
- To write and present an overview of the relevant literature for a specific research topic.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
0	0	6	3	0	0	0	100	100

Content:

Research surveys help them to boost their skills and widen their horizon of thinking for the domain. It will act like a beginners guide to undertake the research survey for the students interest domain which will also help for the student during his/her final year dissertation work. The student shall deliver specifications of the research survey in terms of implemented, existing research work and result with finding facts and challenges with proposed solution of domain. In this course the student learns how to perform research surveys and its analysis. Students will have a semester presentation and end semester presentation. Mid semester presentation will include identification of the problem based on the latest literature available. End semester presentation should be done along with the report and the methodology adopted involving scientific research, collection and analysis of data, determining solutions highlighting individuals' contribution.

Course Outcome:

At the end of the course, the student will be able to:

- Identify engineering knowledge and understanding problems by reviewing in a system context.
- Study and develop elicit, analyze, and document skills in the form of a requirements specification.

- Evaluate traceability between requirements, design, and implementation artefacts for the research work.
- Able to write technical reports.
- Develop skills to present and defend their work in front of a technically qualified audience.

Subject Code: 10070203	**Subject Title: Distributed Systems**
Pre-requisite: Operating System, Computer Network, Data and File Structures, and Algorithms	

Course Objective:

- This subject will give an introduction to traditional vs distributed computing systems.
- It will be introduced to various issues in the design of distributed systems.
- Learn how Process queries in real time for distributed systems.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	**Introduction to DS:** Introduction to Distributed Systems, Distributed system models, Design issues in DS.	2	5
2	**Communication in Distributed System**: Inter process communication: Message passing model, Remote procedure call and implementation issues, Point to point and Group communication, Client Server model & its implementation, Socket programming, Case Studies: SUN RPC, DEC RPC	6	15
3	**Synchronization in distributed systems:** Introduction, Temporal ordering of events, Clock synchronization, mutual exclusion, Deadlock in distributed systems, Election algorithms	5	10
4	**Processes and processors in distributed systems:** Threads, system model, processor allocation, scheduling in distributed systems: Load balancing and sharing approach, fault tolerance, Real time distributed systems, Process migration and related issues	5	10
5	**Remote Method Invocation:** Introduction, Java RMI Architecture, API for	2	10

	Java RMI, Client Callback, Stub downloading		
6	**Common Object Request Broker Architecture:** Introduction, Interface, Inter-ORB Protocol, Object server and object client, Naming service, Object service.	5	15

7	**Processes and processors in distributed systems:** Threads, system model, processor allocation, scheduling in distributed systems: Load balancing and sharing approach, fault tolerance, Real time distributed systems, Process migration and related issues.	4	10
8	**Distributed File Systems:** Introduction, features & goal of distributed file system, file models, file accessing models, file sharing semantics, file caching scheme, file replication, fault tolerance, trends in distributed file system, case study- HDFS.	4	15
9	**Distributed Shared Memory:** Introduction, general architecture of DSM systems, design and implementation issues of DSM, granularity, structure of shared memory space, consistency models, replacement strategy, thrashing	3	10

Course Outcome: After completion of this course, student will be able to

- Understand the concepts and issues related to distributed systems and parallel database.
- Design and develop the application for distributed environment.
- Manage performance, reliability and other issues while designing in distributed environment.
- Apply knowledge of distributed systems techniques and methodologies.
- Evaluate the performance of distributed systems.

List of References:

1. Distributed Computing – Principles and Applications, M. L. Liu, Pearson.
2. Distributed Operating Systems Concepts and Design, Pradeep K Sinha, PHI
3. Distributed Systems Concepts and Design, George Colouris, Jean Dollimore, Tim Kindberg, Pearson

Open e-Resource:

1. Coursera: Parallel, Concurrent, and Distributed Programming in Java Specialization, https://www.coursera.org/specializations/pcdp
2. NPTEL: https://onlinecourses.nptel.ac.in/noc21_cs87/preview
3. GITHUB: https://github.com/zhangruochi/Parallel-Concurrent-and-Distributed-Programming-in-Java-Specialization
4. https://hadoop.apache.org/
5. UDEMY: https://www.udemy.com/topic/distributed-computing/

List of Suggested Experiments:

1. Write a Program to implement Concurrent Echo Client Server Application.
2. Write a Program to implement Concurrent Day Time Server Application.
3. Write a program to solve Producer-Consumer Problem using thread.
4. Write a program to implement any simple program using Socket in java.
5. Implement RPC Programming.
6. Implementation of Server that adds given two values by the clients using Java RMI.
7. Write a program to create CORBA based client server application
8. Implementing Bully Election algorithm for synchronization
9. Implementing Ring Election algorithm for synchronization
10. Write a Program to Increment a Counter in Shared Memory

Subject Code: 10070204	**Subject Title:** Machine Learning
Pre-requisite: Familiarity with programming in Python, Linear Algebra, Probability and Statistics.	

Course Objective:

- To understand the basic theory underlying machine learning.
- To be able to formulate machine learning problems corresponding to different applications.
- To understand a range of machine learning algorithms along with their strengths and weaknesses.
- To be able to apply machine learning algorithms to solve problems of moderate complexity.
- To apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	**Introduction:** what is ML; Problems, data, and tools; Visualization, Linear regression; SSE; gradient descent; closed form; normal equations; features engineering, Overfitting and complexity; training, validation, test data	6	15
2	**Supervised Learning** : Basic methods: Distance-based methods, Nearest-Neighbours , Decision Trees, Naive Bayes Linear models: Linear Regression, Logistic Regression, Generalized Linear Models, Support Vector Machines, Nonlinearity and Kernel Methods, Beyond Binary Classification:	10	25

	Multi-class/Structured Outputs, Ranking		
3	**Unsupervised Learning:** Clustering: K-means/Kernel K-means Dimensionality Reduction: PCA and kernel PCA Matrix Factorization and Matrix Completion Generative Models (mixture models and latent factor models)	8	20
4	**Ensemble Methods**: Rationale for ensemble method, methods for constructing an Ensemble classifier, Bias-Variance decomposition, Bagging, Boosting, Random forests, Empirical comparison among Ensemble methods.	6	20
5	**Advanced Topics**: Semi-supervised Learning,Active Learning, Reinforcement Learning, Introduction to Bayesian Learning and Inference, Recent trends in various learning techniques of machine learning and classification methods	6	20

Course Outcomes:

After completion of course, students would be able to:

- Understand Key concepts, tools, and methods for machine learning using real data sets.
- Evaluate a range of machine learning algorithms along with their strength & weaknesses.
- Formulate machine learning problems corresponding to different applications.
- Apply appropriate machine learning techniques to solve problems of moderate complexity.
- Compare the machine learning models based on their accuracy.
- Develop machine learning-based solutions to real-world problems.

List of References:

1. Tom M. Mitchell, “Machine Learning”, McGraw-Hill Education (INDIAN EDITION), 2013.
2. Christopher M. Bishop, "Pattern Recognition and Machine Learning", by Springer, 2007
3. Amanda Casari, Alice Zheng, “Feature Engineering for Machine Learning”, O’Reilly, 2018.
4. Andreas Muller, "Introduction to Machine Learning with Python: A Guide for Data Scientists", Shroff/O'Reilly; First edition (2016).
5. Aurelien Geron Hands-on Machine Learning with Scikit-Learn, Keras and Tensorflow 2.0O’Reilly.

Open e-Resource:

1. https://onlinecourses.nptel.ac.in/noc22_cs29/preview
2. https://www.udemy.com/course/machinelearning/
3. https://www.kaggle.com/learn/intro-to-machine-learning

List of Suggested Experiments:

Refer the given below dataset for performing experiment

https://www.kaggle.com/datasets
https://github.com/topics/dataset
UCI Machine Learning Repository

1. Basic exercises on Python Machine Learning Packages such as Numpy, Pandas and matplotlib.
2. Write a program to compute the Covariance, Correlation between a pair of attributes. Extend the program to compute the Covariance Matrix and Correlation Matrix.
3. Write a program to fit the points with a hyper plane using Linear Regression. Calculate sum of residual error.
4. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
5. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set.
6. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
7. Write a program to implement feature reduction using Principle Component Analysis.
8. Write a program to implement Support Vector Machine and estimate it test performance.
9. Write programs to implement ADALINE and MADALINE for given learning task.
10. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.
11. Write a program to implement K means clustering algorithm. Select your own dataset to test the program. Demonstrate the nature of output with varying value of K.

Subject Code: 10070205	**Subject Title:** DATA VISUALIZATION
Pre-requisite: Knowledge of Programming Language, DBMS, JavaScript and HTML5	

Course Objective:

- To develop charts, maps, tables, and other visual representations of data.
- To create compelling, interactive dashboards to combine several visualizations into a cohesive and functional whole.
- To utilize advanced Tableau features including parameters, data blending, custom SQL, huge datasets, custom date hierarchies, and others.
- To use data visualizations, dashboards to support relevant communication for diverse audiences

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**

1	**Introduction to Data Visualization:** Acquiring and Visualizing Data, Simultaneous acquisition and visualization, Applications of Data Visualization, Keys factors of Data Visualization (Control of Presentation, Faster and Better JavaScript processing, Rise of HTML5, Lowering the implementation Bar) Exploring the Visual Data Spectrum: charting Primitives (Data Points, Line Charts, Bar Charts, Pie Charts, Area Charts), VizHub, Exploring advanced Visualizations (Candlestick Charts, Bubble Charts, Surface Charts, Map Charts, Infographics). Making use of HTML5 CANVAS, Integrating SVG	5	10
2	**Basics of Data Visualization – Tables:** Reading Data from Standard text files (.txt, .csv, XML), Displaying JSON content Outputting Basic Table Data (Building a table, Using Semantic Table, Configuring the columns), Assuring Maximum readability (Styling your table, Increasing readability, Adding dynamic Highlighting), Including computations, Using data tables library, relating data table to a chart	5	15
3	**Visualizing data Programmatically:** Creating HTML5 CANVAS Charts (HTML5 Canvas basics, Linear interpolations, A Simple Column Chart, Animations), Starting with Google charts (Google Charts API Basics, A Basic bar chart, A basic Pie chart, Working with Chart Animations).	9	25

4	**Introduction to D3.js:** Getting setup with D3, Making selections, changing selection's attribute, Loading and filtering External data : Building a graphic that uses all of the population distribution data, Data formats you can use with D3,Making a Face with D3, Creating a server to upload your data, D3's function for loading data, Dealing with Asynchronous requests, Loading and formatting Large Data Sets	6	15
5	**Advanced Data Visualization:** Making charts interactive and Animated: Data joins, updates and exits, interactive buttons, Updating charts, Adding transactions, using keys Adding a Play Button: wrapping the update phase in a function, Adding a Play button to the page, Making the Play button go, Allow the user to interrupt the play, sequence	4	15
6	**Information Dashboard Design:** Introduction, Dashboard design issues and assessment of needs, Considerations for designing dashboard-visual perception, Achieving eloquence, Advantages of Graphics _Library of Graphs, Designing Bullet Graphs, Designing Sparklines, Dashboard Display Media, Critical Design Practices, Putting it all together - Unveiling the dashboard.	7	20

Course Outcomes:

After completion of course, students would be able to:

- Explore various data visualization techniques in order to provide new insight.
- Apply appropriate data visualization techniques to provide trends/insights for the given dataset.
- Apply visualization tools / techniques for various data analysis tasks.
- Given the application context for given data set, Design the information Dashboard for access information based on user criteria.

List of References:

1. Jon Raasch, Graham Murray, Vadim Ogievetsky, Joseph Lowery, “JavaScript and jQuery for Data Analysis and Visualization”, WROX
2. Ritchie S. King, Visual story telling with D3” Pearson
3. Ben Fry, "Visualizing data: Exploring and explaining data with the processing environment", O'Reilly, 2008.
4. A Julie Steele and Noah Iliinsky, Designing Data Visualizations: Representing Informational Relationships, O’Relly
5. Andy Kirk, Data Visualization: A Successful Design Process, PAKT
6. Scott Murray, Interactive Data Visualization for Web, O’Relly
7. Nathan Yau, "Data Points: Visualization that means something", Wiley, 2013.

Open e-Resource:

1. https://www.tutorialrepublic.com/html-tutorial/html5-canvas.php HTML5 (Canvas and SVG tags)
2. https://d3js.org/ (D3.js)
3. https://canvasjs.com/ (Canvas.js)

List of Suggested Experiments:

1. Setup Environment for All the Tools
2. Develop the following Program Using HTML5 CANVAS and SVG TAG
   a. Develop the Different basic Graphical Shapes using HTM5 CANVAS

   a. Develop the Different Advanced Graphical Shapes using HTM5 CANVAS
   b. Develop the Different basic Graphical Shapes using HTM5 SVG
   c. Develop the Different Advanced Graphical Shapes using HTM5 SVG
2. Develop Following Program Using HTML5 and JavaScript
   a. Develop the simple bar chart usingTML5 CANVAS
   a. Read the data .txt file and draw Data Table
   b. Read the data .txt file and draw Simple Bar Chart
   c. Read the data .csv file and draw Data Table
   d. Read the data .csv file and draw Column Bar Chart
   e. Read the data XML file and draw Data Table
   f. Read the data XML file and draw Simple Chart
   g. Read JSON Data and draw Data Table
   h. Read JSON Data and draw Simple Chart
2. Develop Following Program Using HTML5 and D3.js and Canvas.js
   a. Showing the data as a column chart (simple)
   a. Showing the data as a stacked column chart
   b. Showing the Data as a column chart for four age group
   c. Showing the data as a Line chart (single, fewer and multiple lines)
   d. Showing the data as a Pie Chart (single and multiple pie)
   e. Showing the data as a Bar Chart (Simple and multiple)
2. Develop Following Program Using HTML5 and Google Chats API and Map API
   a. Using Google Charts API Basics draw charts like a Bar chart
   a. Using Google Charts API Basics draw charts like a Line chart
   b. Using Google Charts API Basics draw PieChart.
   c. Using Google Charts API Basics draw Donut Chart.
   d. Using Google Charts API Basics draw Candle Chart.
   e. Using Google Charts API Basics draw other types of Chart.

f. Using Google API read JSON file and create Google Map.

Subject Code: 10070206	**Subject Title:** Blockchain
Pre-requisite: Knowledge of Cryptography and System Security	

Course Objective:

- To understand the concept of Blockchain and its relevance with cryptography.
- To acquire knowledge of various techniques used in Blockchain.
- To apply the Blockchain concept in real life applications.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	**Introduction:** Cryptographic Hash Function, Properties of a hash function, Hash pointer and Merkle tree, Digital Signature, Public Key Cryptography, A basic cryptocurrency, Introduction to Origin of Blockchain: What Is Blockchain, Public Ledgers, Bitcoin, Smart Contracts, Block in a Blockchain, Transactions, Distributed Consensus, Public vs Private Blockchain, Understanding Crypto currency to Blockchain, Permissioned Model of Blockchain, Overview of Security aspects of Blockchain	4	15
2	**Cryptocurrency:** Introduction, Bitcoin and the Cryptocurrency, Evolution of Currency, Birth of Bitcoin, Cryptocurrency Basics, What Is Cryptocurrency, Characteristics of Cryptocurrency, Cryptocurrency Wallets, Types of Cryptocurrency, Altcoins, Tokens, Popular Coins and Tokens, Cryptocurrency Usage	6	15

3	**Public Blockchain System:** Introduction, Public Blockchain, Blockchain Layers, Popular Public Blockchains, The Bitcoin Blockchain, Need for Bitcoin and Its Value, common Terminologies, Bitcoin Mining, Proof of Work (PoW), Proof of Stake (PoS) and Hashcash in Bitcoin, Bitcoin Script, Transaction in the Bitcoin Network, Ethereum Blockchain, Introduction to Smart Contracts, Code Is Law and Ethereum Classic, Ethereum Components, How mining Works in Ethereum, Merkle Patricia Tree, Architecture of Ethereum, Workflow of Ethereum, Comparison of Bitcoin and Ethereum	7	25
4	**Private Blockchain System:** Introduction, Key Characteristics of Private Blockchain, Private Blockchain and Permissioned Blockchain, Why We Need Private Blockchain, Private Blockchain Examples, Private Blockchain and Open Source, E-commerce Site Example, Problems with the Centralized Server, Private Blockchain Concepts in an E-commerce, Scenario, Various Commands (Instructions) in E-commerce, Blockchain and Smart Contract in Private Environment; Consortium Blockchain Introduction, Key characteristics of Consortium Blockchain, Hyperledger platform, Overview of Ripple, Overview of Corda	7	25
5	**Enterprise Application of Blockchain:** Cross border payments, Know Your Customer (KYC), Food Security, Mortgage over Blockchain, Blockchain enabled Trade, We Trade — Trade Finance Network, Supply Chain Financing, Identity on Blockchain	6	10
6	**Privacy, Security Issues in Blockchain:** Pseudo-anonymity vs. anonymity, Zcash and Zk-SNARKS for anonymity preservation, attacks on Blockchains – such as Sybil attacks, selfish mining, 51% attacks - -advent of algorand, and Sharding based consensus algorithms to prevent these attcks.	6	10

Course Outcome: After completion of this course, student will be able to

- Acquire the basic knowledge of Blockchain technology.
- Analyze various algorithms used in Blockchain.
- Introduction about cryptocurrency and various regulations.
- Aware about privacy and security issues in Blockchain.
- Design and understand various applications using Blockchain.

List of References:

1. S. Shukla, M. Dhawan, S. Sharma, S. Venkatesan, -Blockchain Technology: Cryptocurrency and Applications, Oxford University Press, 2019.
2. Andreas Antonopoulos, “Mastering Bitcoin: Unlocking Digital Cryptocurrencies”, O’Reilly, 2014.
3. Daniel Drescher, “Blockchain Basics”, First Edition, Apress, 2017

Open e-Resource:

1. https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/ crse0401.html
2. https://www.edx.org/course/bitcoin-and-cryptocurrencies?
3. https://www.udemy.com/course/understanding-blockchain-technology/
4. https://www.hyperledger.org/projects/fabric
5. https://www.packtpub.com/big-data-and-business-intelligence/ hands-blockchain-hyperledger

List of Suggested Experiments:

1. Implement SHA-1.
2. Create different types of Blockchain bitcoin wallet.
3. Implement Merkle tree hash algorithm
4. Set up an environment for Ethereum on Windows and Linux.
5. Create smart contracts using solidity.
6. Create and deploy a blockchain network using Hyperledger Fabric SDK for Java Set up and initialize the channel, install and instantiate chain code, and perform invoke and query on your blockchain network.
7. Deploy an asset-transfer app using blockchain. Learn app development within a Hyperledger Fabric network.
8. Use blockchain to track fitness club rewards. Build a web app that uses Hyperledger Fabric to track and trace member rewards.
9. Build the Environment for Corda.

Subject Code: 10070207	**Subject Title: Computer Vision**
Pre-requisite: Linear Algebra, Vector Calculus, Data Structures and Programming.	

Course Objective:

- To introduce the student with the foundation of image formation, measurement, and analysis.
- Understand the computer vision algorithms, methods and concepts which will enable the student to implement computer vision systems with emphasis on applications and problem solving.
- Grasp the principles of state-of-the-art deep neural networks. Lab exercises will familiarize the student with typical software development tools.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	**Overview**: Computer imaging systems, image representation, Image analysis, pre-processing and Binary image analysis.	4	05
2	**Edge detection**: Edge detection performance, Hough transform, corner detection.	5	15
3	**Segmentation**: contours and edges of multiple objects in an image, Morphological filtering, Fourier transform, object labeling and counting	5	10
4	**Feature extraction**: shape, histogram, color, spectral, texture, using CV IP tools, Feature analysis, feature vectors, distance /similarity measures, boundary descriptors, region descriptors, data preprocessing	6	20
5	**Pattern Analysis**: Clustering: K-Means, K-Medoids, Mixture of Gaussians Classification: Discriminant Function, Supervised, Un-supervised, Semi- supervised Classifiers: Bayes, KNN, ANN models; Dimensionality Reduction: PCA, LDA, ICA, and Non-parametric methods.	5	15
6	**Recent trends in Activity Recognition**, computational photography, Face detection, Face recognition, Surveillance, foreground-background separation, combining views from multiple cameras, identifying road signs	5	15
7	Basics **Neural Network and Deep Learning Methods**: perceptron, back propagation, CNN and RNN.	6	20

Course Outcome: After completion of this course, student will be able to

- Review image processing techniques for computer vision.
- Understand shape and region analysis.
- Understand hough transform and its applications to detect lines, circles, ellipses.
- Develop the practical skills necessary to build computer vision applications.
- Study some applications of computer vision algorithms.

List of References:

1. Computer Vision: Algorithms and Applications by Richard Szeliski.
2. ComputerVision –A modern approach, by D.Forsythand J.Ponce, PrenticeHall Robot Vision,by B.K.P.Horn, McGraw-Hill.

3. Feature Extraction & Image Processing for Computer Vision by Mark Nixon and Alberto S. Aquado, Third Edition, Academic Press.
4. Computer Vision: Models, Learning, and Inference, by Simon J. D. Prince, Cambridge University Press.
5. Dictionary of Computer Vision and Image Processing, by Fisher et al.
6. Three-Dimensional Computer Vision, by Olivier Faugeras, The MIT Press.

Open e-Resource:

1. https://nptel.ac.in/courses/106105216
2. https://nptel.ac.in/courses/106106182

List of Suggested Experiments:

Perform the following practicals using any of the MATLAB/OpenCV/Python image processing tools/libraries.

1. To perform variants of linear filters on an image.
2. To perform a median filter on an image.
3. To perform all morphological filter operations on the image.
4. To perform edge detection on an image using Sobel, Prewitt, Roberts and Canny Method.
5. To perform Hough transform on an image.
6. Generate histogram and perform histogram equalization of an image.
7. To perform clustering using any technique on the image.
8. To perform various pattern classification and analysis techniques on images.
9. Implement The Finite Words Classification System Using Backpropagation Algorithm
10. Use case implementation using CNN.

Subject Code: 10070208	**Subject Title: Cloud Computing**
Pre-requisite: Operating System, Networking	

Course Objective:

- This course gives students an insight into the basics of cloud computing along with virtualization, cloud computing is one of the fastest growing domains from a while now.
- It will provide the students basic understanding about cloud and virtualization along with it how one can migrate over it.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents

Sr. No	Topic	Total Hours	Weight (%)
1	**Cloud Computing Overview:** Origins of Cloud computing – Cloud components - Essential characteristics – On-demand self-service, Broad network access, Location independent resource pooling, Rapid elasticity, measured service, Comparing cloud providers with traditional IT service providers, Roots of cloud computing.	4	15
2	**Cloud Insights:** Architectural influences – High-performance computing, Utility and Enterprise grid computing, Cloud scenarios – Benefits: scalability, simplicity, vendors, security, Limitations – Sensitive information - Application development- security level of third party - security benefits, Regularity issues: Government policies.	6	20
3	**Cloud Architecture- Layers and Models:** Layers in cloud architecture, Software as a Service (SaaS), features of SaaS and benefits, Platform as a Service (PaaS), features of PaaS and benefits, Infrastructure as a Service (IaaS), features of IaaS and benefits, Service providers, challenges and risks in cloud adoption. Cloud deployment model: Public clouds – Private clouds – Community clouds - Hybrid clouds - Advantages of Cloud computing.	6	20
4	**Cloud Simulators- CloudSim and GreenCloud:** Introduction to Simulator, understanding CloudSim simulator, CloudSim Architecture (User code, CloudSim, GridSim, SimJava) Understanding Working platform for CloudSim, Introduction to GreenCloud	8	20
5	**Introduction to VMWare Simulator:** Basics of VMWare, advantages of VMware virtualization, using Vmware workstation, creating virtual machines- understanding virtual machines, create a new virtual machine on local host, cloning virtual machines, virtualize a physical machine, starting and stopping a virtual machine.	8	15
6	**Introduction to Hybrid Cloud:** Hybrid cloud management, Managing the Hybrid workloads, Development and Deployment in Hybrid cloud.	4	10

Course Outcomes:

After completion of course, students would be able to:

- Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure & compare the advantages and disadvantages of various cloud computing platforms.
- Deploy applications over commercial cloud computing infrastructures such as Amazon Web Services, Windows Azure, and Google AppEngine.
- Analyze the performance, scalability, and availability of the underlying cloud technologies and software.
- Explain recent research results in cloud computing and identify their pros and cons.
- Solve a real-world problem using cloud computing through group collaboration.

List of References:

1. Rajkumar Buyya, James Broberg, Andrzej M Goscinski, Cloud Computing: Principles and Paradigms,Wiley publication.
2. Toby Velte, Anthony Velte, Cloud Computing: A Practical Approach, McGraw-Hill Osborne Media.
3. George Reese, Cloud Application Architectures: Building Applications and Infrastructure in the
4. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online - Michael Miller - Que 2008

Open e-Resource:

1. https://archive.nptel.ac.in/courses/106/105/106105167/
2. https://www.coursera.org/learn/introduction-to-cloud

List of Suggested Experiments:

1. Study and implementation of Infrastructure as a Service. Study of Cloud Computing & Architecture.
2. Implement phishing attack on cloud.
3. Write a case study of incidence reporting in case of breach of cloud security.
4. Use the services offered by Azure, AWS and GOOGLE. Compare them.
5. Use open-source tool to evaluate performance of cloud platform.
6. Prepare a case study of security policy or service level agreement is signed by cloud service provider.
7. Prepare a case study of facebook, twitter data which is stored on cloud. Write a program to inject malware in this data.
8. Implement attribute-based encryption algorithm for cloud.
9. Implement compartmentalization techniques, the provider can use to prevent access into virtual container of one customer by other customers.
10. Implement identity management mechanism in cloud.

Subject Code: 10070209	**Subject Title:** Natural Language Processing
Pre-requisite: Linear algebra, Probability and Statistics, Programming in any high level language, preferably python or Matlab	

Course Objective:

- To read, understand, and decode human words in a valuable manner.
- To familiarize the concepts and techniques of Natural language Processing.
- To Recognize Speech and parsing with grammar.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
3	0	2	4	60	40	30	20	150

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**
1	**Introduction to NLP:** What is NLP? Why NLP is Difficult? History of NLP, Advantages of NLP, Disadvantages of NLP, Components of NLP, Origins and challenges of NLP,Applications of NLP, How to build an NLP pipeline? Phases of NLP, NLP APIs, NLP Libraries	4	15
2	**Language Modeling and Part of Speech Tagging:** Unigram Language Model, Bigram, Trigram, N-gram, Advanced smoothing for language modeling, Empirical Comparison of Smoothing Techniques, Applications of Language Modeling, Natural Language Generation, Parts of Speech Tagging, Morphology, Named Entity Recognition	10	25
3	**Word Level Analysis:** Words and Word Forms: Bag of words, skip-gram, Continuous Bag-Of-Words, Embedding representations for words Lexical Semantics, Word Sense Disambiguation, Knowledge Based and Supervised Word Sense Disambiguation, Smoothing, Interpolation and Backoff – Word Classes	6	15
4	**Text Analysis, Summarization and Extraction:** Sentiment Mining, Text Classification, Text Summarization, Information Extraction, Named Entity Recognition, Relation Extraction, Question Answering in Multilingual Setting; NLP in Information Retrieval, Cross-Lingual IR	8	20
5	**Machine Translation and Advanced Algorithms:** Need of MT, Problems of Machine Translation, MT Approaches, Direct Machine, Translations, Rule- Based Machine Translation, Knowledge Based MT System, Statistical Machine Translation (SMT), Encoder-decoder architecture, Algorithms used by neural networks, Rule-Based System for NLP, Machine Learning for NLP Problems, Deep learning for NLP.	8	25

Course Outcomes:

After completion of course, students would be able to:

- Understand comprehend the key concepts of NLP and identify the NLP challenges and issues
- Develop Language Modelling for various text corpora across the different languages
- Illustrate computational methods to understand language phenomena of word sense disambiguation
- Design and develop applications for text or information extraction/summarization/classification.
- Apply different Machine translation techniques for translating a source to target language(s)

List of References:

1. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition Jurafsky, David, and James H. Martin, PEARSON
2. Foundations of Statistical Natural Language Processing, Manning, Christopher D., and Hinrich Schütze, Cambridge, MA: MIT Press

3. Natural Language Understanding, James Allen. The Benjamin/Cummings Publishing Company Inc.
4. Natural Language Processing with Python – Analyzing Text with the Natural Language ToolkitSteven Bird, Ewan Klein, and Edward Loper.

Open e-Resource:

1. https://www.kaggle.com/learn/natural-language-processing
2. https://www.javatpoint.com/nlp
3. https://onlinecourses.nptel.ac.in/noc19_cs56/preview
4. https://www.coursera.org/specializations/natural-language-processing

List of Suggested Experiments:

1. To study Preprocessing of text (Tokenization, Filtration, Script Validation, Stop Word Removal, Stemming)
2. To tag a given text with basic Language features.
3. To implement a rule-based system to tackle morphology/syntax of a language.
4. To implement the N-gram model.
5. To implement POS tagging.
6. To implement chunking.
7. To study Named Entity Recognition.
8. To compare and contrast the use of different statistical approaches for different types of NLP applications.

Subject Code: 10000251	**Subject Title: Technical Writing**
Pre-requisite: Basic Language Skills	

Course Objective:

- Technical Writing focuses on planning and preparation of word choices and sentences and focuses on vocabulary building and helps them to be an effective communicator.
- Students will be able to produce a set of documents related to writing in the workplace and will be able to improve their ability to write clearly and accurately.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
2	0	0	0	0	50	0	0	50

Subject Contents			
Sr. No	**Topic**	**Total Hours**	**Weight (%)**

1	Planning and Preparation, Word Order, Breaking up long sentences, Structuring	4	10
2	Paragraphs and Sentences, Being Concise and Removing Redundancy, Avoiding Ambiguity and Vagueness	4	10
3	Sections of a paper – Title, abstract, introduction, etc. review of the literature, writing - methods, results, discussion, conclusions and final check	6	20
4	Organization of technical report: Title, Authors, Affiliation, Abstract,	4	20
5	Three pass reading. Effective communication– Teacher/Student perspective. Listening – Hearing Vs. Listening. Notes Taking. Elevator Pitch. Technical writing.	4	20
6	Useful phrases and punctuation, in-text citation and bibliography– MLA/APA styles	4	20

Course Outcome: After completion of this course, student will be able to

- Understand that how to improve your writing skills and level of readability.
- Learn about what to write in each section.
- learn various skills necessary that are necessary for doing research.
- Ensure the good quality of paper at very first-time submission.

List of References:

1. Adrian Wallwork, English for Writing Research Papers, Springer New York Dordrecht Heidelberg London, 2011
2. Day R, How to Write and Publish a Scientific Paper, Cambridge University Press ,2006
3. Raman, Meenakshi and Sangeeta Sharma, "Technical Communication: Principle and Practice", Oxford University Press.
4. Goldbort R. Writing for Science, Yale University Press, 2006

Open e-Resource:

1. **https://onlinecourses.nptel.ac.in/**
2. **https://nptel.ac.in/courses**

List of Suggested Experiments: NA

Semester 3

Subject Code:	Internal Review I

Course Objective:

1. To guide students in finalizing the research problem and scope.
2. To assess the quality of literature review and research gap identification.
3. To ensure the research methodology is clearly defined.
4. To provide constructive feedback before proceeding to Dissertation Phase–I implementation.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
0	0	4	2	0	0	0	100	100

Internal Review – I ,is conducted before the formal submission of Dissertation Phase–I. The review is carried out by the faculty guide/supervisor to monitor the early-stage research progress.

Students are required to present:

- Proposed research topic in SE
- Problem statement
- Literature review
- Research gap identification
- Objectives of the research
- Proposed methodology or framework
- Expected outcomes

The guide evaluates the research feasibility, technical depth, and relevance of the selected topic.

Review Components

Students must submit and present:

- Topic approval document
- Literature survey summary
- Research problem formulation
- Proposed methodology
- Preliminary research plan
- Timeline for Dissertation Phase–I

Subject Code:	Dissertation Phase I
Pre-requisite: Research Methodology and IPR, Advanced Software Engineering, Advanced Data Structures & Algorithms, Software Project Management, Distributed Systems, Data Science, Cloud Computing, or equivalent courses.	

Course Objective:

1. To enable students to identify a research problem in Software Engineering or related domains.
2. To develop skills in literature survey, problem formulation, and research methodology.
3. To encourage students to analyze existing software engineering approaches and identify research gaps.
4. To design a novel framework, model, methodology, or software solution.
5. To prepare students for independent research, experimentation, and technical reporting.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
0	0	28	14	0	0	100	0	100

Course Description

Students shall select a research topic in Software Engineering or related interdisciplinary areas such as:

- Software Architecture
- Software Testing and Quality Assurance
- Software Project Management
- Distributed Systems
- Cloud Computing
- Data Science
- Machine Learning Applications
- DevOps and Software Automation
- Cyber security
- High Performance Computing
- Software Process Improvement
- Enterprise Software Systems

Students shall work under the supervision of a faculty guide and conduct a comprehensive literature review to identify the research gap and formulate the research problem.

The work in Phase-I should include:

- Identification of research problem

- Literature survey
- Definition of objectives and scope
- Selection of methodology
- Initial design of system/model
- Preliminary experimentation (if possible)

Students are required to present their research proposal and progress in seminars and reviews conducted by the department.

Progress Report (Submission Requirements)

Students must submit a progress report containing:

- Research problem statement
- Literature survey and related work
- Research gap identification
- Objectives of the research
- Proposed methodology
- Expected outcomes

Students must present their work before the department review committee.

Assessment Components

Evaluation may be based on:

- Problem definition and novelty
- Literature survey quality
- Research methodology
- Presentation and seminar
- Progress report documentation

Semester 4

Subject Code: 11390401	Internal Review II

Course Objective:

1. To evaluate the implementation progress of the dissertation work.
2. To assess experimental setup, security analysis, and preliminary results.
3. To guide improvements in methodology and experimentation.
4. To ensure readiness for final Dissertation Phase–II submission.
5. To improve research documentation and presentation quality.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
0	0	4	2	0	0	0	100	100

Course Description

Internal Review – II is conducted after completion of Dissertation Phase–I and before Dissertation Phase–II submission.

This review evaluates the progress of implementation, experiments, and initial results.

The review is conducted by the research guide/supervisor to ensure that the student is progressing toward the successful completion of the dissertation.

Students must present:

- Implementation progress
- System architecture or model design
- Experimental setup
- Preliminary results
- Security evaluation or performance analysis
- Planned improvements and future work

Review Components

Students must submit and present:

- Implementation status report
- Experimental setup and methodology
- Preliminary results and analysis
- Challenges faced and solutions
- Plan for final dissertation completion

Subject Code: 11390402	Dissertation Phase II
Pre-requisite: Successful completion of Dissertation Phase-I	

Course Objective:

1. To implement the proposed software framework, model, or system developed in Phase–I.
2. To perform experimentation, validation, and performance evaluation.
3. To analyze results and compare them with existing approaches.
4. To prepare a dissertation thesis and technical documentation.
5. To develop skills in research presentation, defense, publication, and professional communication.

Teaching Scheme (Hours per week)				**Evaluation Scheme (Marks)**				
				Theory		**Practical**		
Lecture	**Tutorial**	**Practical**	**Credit**	**University Assessment**	**Continuous Assessment**	**University Assessment**	**Continuous Assessment**	**Total**
0	0	28	14	0	0	100	0	100

Course Description

This phase represents the final stage of the M.Tech dissertation work. Students will implement the proposed solution developed in Phase-I and conduct extensive experiments and analysis.

The dissertation work may involve:

- Software framework development
- Software architecture implementation
- Cloud-based application development
- Distributed system implementation
- Software quality and testing studies
- Data-driven software applications
- Performance evaluation and optimization
- Comparative study with existing technique

Students must prepare a complete dissertation report and present their research findings in an open seminar and viva-voce examination.

Dissertation Report Structure

The final thesis should include:

- Introduction
- Literature Review

- Problem Statement and Research Gap
- Proposed Methodology / Model
- Implementation and Experimental Setup
- Results and Analysis
- Comparison with Existing Methods
- Conclusion and Future Work
- References

Assessment Components

Evaluation will be based on:

- Implementation and technical quality
- Experimental validation and analysis
- Dissertation report
- Research contribution
- Seminar presentation and viva-voce

ANNEXURE III

1. Executive Summary

A structured industry validation survey was conducted among 25 senior technology leaders, HR professionals, and engineering managers across IT, BFSI, manufacturing, healthcare, and public-sector organizations in Gujarat and pan-India. The purpose was to assess demand for formally trained M.Tech. graduates in Software Engineering and to validate the need for the proposed programme at the University.

The findings are unambiguous and strongly positive. India currently faces a critical shortage of advanced software engineering talent — professionals capable of architecting, building, and governing large-scale enterprise systems. NASSCOM's 2025 Technology & Talent Report estimates India requires over 1.4 million additional software professionals with advanced engineering competencies by 2027, a gap that undergraduate programmes alone cannot fill. Simultaneously, the World Economic Forum's Future of Jobs Report 2025 identifies Software and Application Developers as one of the five largest growth occupations globally, with roles in cloud-native development, DevSecOps, and software architecture growing at 26% through 2030.

Key Survey Findings at a Glance

Key Indicator	Result
Confirm critical Software Engineering talent gap in the current market	100%
Would actively recruit M.Tech. Software Engineering graduates	88%
Open to formal industry partnership with the University	80%
Average rating — recommend programme launch	4.5 / 5
Overall recommendation score (launch now / with adjustments)	96%

Every survey respondent confirmed a significant talent gap in the current market. Twenty-two out of twenty-five organizations expressed willingness to recruit graduates from a formally structured university M.Tech. programme aligned with UGC's Curriculum and Credit Framework for PG Programmes (June 2024) and NEP 2020. Twenty organizations expressed interest in participating as industry partners. The survey validates the University's decision to launch this programme and provides strong industry backing for an immediate start.

2. Regulatory and Policy Framework

The proposed M.Tech. in Software Engineering is designed in full conformity with applicable national regulatory instruments.

UGC (University Grants Commission): Statutory body responsible for maintaining standards of Higher Education in India. UGC approval is mandatory for degree validity.

AICTE (All India Council for Technical Education): Mandatory for all technical engineering programmes; responsible for planning and development of technical and management education in India.

UGC PG Credit Framework (June 2024): Provides for flexible one-year and two-year PG programme structures, exit provisions at Year 1, multi-disciplinary elective baskets, and creditization of work experience under the National Credit Framework (NCrF).

NEP 2020: Encourages postgraduate specialization in advanced computing disciplines including software engineering, system design, cybersecurity, and emerging technology integration.

For M.Tech. eligibility, candidates who have completed a 4-year undergraduate programme or a 3-year undergraduate programme and a 2-year PG programme in STEM subjects are eligible for admission in M.E./M.Tech. in allied areas. IIT Bombay's and IIT Madras's M.Tech. Software Engineering and Computer Science programmes — aligned with NEP 2020 — exemplify best practice, offering flexible basket-based courses including programme core courses (PCC), programme elective courses (PEC), open electives, and mandatory industrial internship or research activity. The University's M.Tech. Software Engineering curriculum mirrors this structure to ensure national comparability and industry relevance.

3. Respondent Profile

The 25 respondents were drawn from technology-intensive organizations across five sectors spanning Ahmedabad, Gandhinagar, Surat, and national tech hubs including Bengaluru, Pune, and Hyderabad. Respondent designations included Chief Technology Officers, VP Engineering, Software Architecture Heads, Senior Engineering Managers, HR & Talent Acquisition Heads, and R&D Directors — ensuring responses reflect both technical requirements and strategic hiring decisions.

Sector	No. of Respondents	% of Sample
Information Technology & Software Products	10	40%
Banking, Financial Services & Insurance (BFSI)	5	20%
Manufacturing & Industry 4.0 / ERP Systems	4	16%
Healthcare & Life Sciences (HealthTech)	3	12%
Public Sector / Smart City / GovTech	3	12%
TOTAL	**25**	**100%**

4. Findings — Q1: Software Engineering Talent Gap in Fresh Graduates

Respondents were asked to identify the most critical knowledge and competency gaps they observe when hiring fresh engineering graduates for software engineering roles. Multiple selections were permitted. The responses reveal a consistent and structurally deep pattern across all sectors surveyed.

India produces over 1.5 million engineering graduates annually, yet industry leaders consistently report that fewer than 25% are considered 'deployment-ready' for senior software engineering roles without significant upskilling.

NASSCOM 2025 estimates a shortage of over 1.4 million advanced software engineering professionals by 2027 — spanning software architecture, cloud-native development, DevSecOps, and large-scale system design.

The demand for Software Architects and Senior Engineers with postgraduate-level depth grew by 68% year-on-year between 2023 and 2025, driven by enterprise digital transformation programmes across BFSI, manufacturing, and public sectors.

Average salaries for M.Tech.-level Software Engineers are 40–55% higher than B.Tech. counterparts in equivalent roles, reflecting the premium that industry places on advanced engineering competency.

Knowledge and Competency Gaps Identified by Employers

Knowledge / Competency Gap Identified	Respondents	Percentage
No experience with large-scale software architecture & system design	25/25	100%
Inability to implement DevSecOps / CI-CD pipelines in production	24/25	96%
No hands-on cloud-native development (AWS / Azure / GCP)	23/25	92%
Weak software quality assurance, testing & test automation skills	22/25	88%
No understanding of microservices & distributed systems	21/25	84%
Lack of software project management & agile engineering practices	20/25	80%
No knowledge of enterprise security & secure coding practices	19/25	76%

Knowledge / Competency Gap Identified	Respondents	Percentage
Cannot work with legacy system modernization & re-engineering	18/25	72%
Weak understanding of software product management & product lifecycle	15/25	60%
Poor research & technical documentation skills	13/25	52%

Selected Employer Comments — Talent Gap

> *"Fresh B.Tech. graduates can write code, but they cannot design systems. When we ask them to architect a multi-tenant SaaS platform or reason about failure modes at scale, the gap is immediately visible. An M.Tech. focused on software engineering depth would solve our most persistent hiring problem."*
>
> **— CTO, Ahmedabad IT Products Company**

> *"Our BFSI platform handles millions of transactions daily. We need engineers who understand both the technical architecture and the regulatory compliance layer — DPDP Act, RBI IT guidelines, SOC 2. Fresh graduates come with neither. That combined literacy is what we cannot find in the market."*
>
> **— VP Engineering, BFSI Sector, Mumbai**

> *"DevSecOps is no longer an optional capability — it is mandatory for every software release. We are running 3-month internal bootcamps just to bring fresh hires to production readiness. A university that embeds this into an M.Tech. would save us six months of internal training per hire."*
>
> **— Head of Engineering, Technology Firm, Bengaluru**

5. Findings — Q2: Advanced Software Engineering Readiness

Respondents were asked how their demand for advanced software engineering professionals — capable of system design, cloud-native development, and DevSecOps — has changed since 2022, and whether they are finding enough qualified candidates in the Gujarat and national market.

Change in Demand for Advanced Software Engineers (Since 2022)

How has demand for advanced software engineers changed since 2022?	Count	%
Significantly increased	17	68%
Moderately increased	6	24%
No change	2	8%
Decreased	0	0%

Availability of Qualified Software Engineering Professionals

Are you finding enough qualified senior software engineers?	Count	%
No — serious shortage at senior / architect level	19	76%
Partially — some available but require extensive upskilling	5	20%
Yes — adequate supply	1	4%

6. Findings — Q3: Most Valued Curriculum Subject Areas

Respondents were asked which subject areas they would most value in M.Tech. Software Engineering graduates — directly informing curriculum design under the UGC PG Credit Framework 2024. The results strongly support a curriculum structured around software architecture, cloud-native engineering, DevSecOps, quality engineering, and software project management, with domain-specific electives for BFSI, HealthTech, and Industry 4.0.

Subject Area / Curriculum Module	Respondents	Percentage
Software Architecture & Large-Scale System Design	25/25	100%
DevSecOps, CI/CD Pipelines & Release Engineering	24/25	96%
Cloud-Native Development (AWS / Azure / GCP)	23/25	92%
Microservices, Distributed Systems & API Design	22/25	88%
Software Quality Engineering & Test Automation	21/25	84%
Enterprise Security, Secure Coding & Compliance (DPDP / ISO 27001)	20/25	80%
Agile Engineering, Scrum & Software Project Management	19/25	76%
Database Engineering, Data Modelling & NoSQL Systems	17/25	68%
AI/ML Integration in Software Systems	16/25	64%
Software Product Management & Product-Led Engineering	14/25	56%
Domain Engineering (FinTech / HealthTech / Industry 4.0)	13/25	52%
Research Methods & Technical Writing for Engineers	11/25	44%

These findings directly support the University's proposed four-semester curriculum structure distributing credits across software engineering core theory, applied engineering practicals, domain-specific electives, research methodology, and a capstone industry project or thesis.

7. Findings — Q4: Recruitment Intent

Respondents were asked to rate their likelihood of actively recruiting from a University M.Tech. Software Engineering programme producing graduates trained in software architecture, DevSecOps, cloud-native development, quality engineering, and enterprise security. Industry data confirms that M.Tech.-qualified software engineers command a 40–55% salary premium over B.Tech. counterparts, reflecting deep demand for graduate-level engineering depth.

Recruitment Intent Ratings

Rating	Count	Percentage	Interpretation
5 — Definitely would recruit	13	52%	High intent
4 — Likely to recruit	9	36%	Positive intent
3 — Possibly would recruit	3	12%	Neutral
2 — Unlikely	0	0%	—
1 — Definitely would not	0	0%	—
Average Rating: 4.4 / 5.0			**Strong positive**

Target Roles for M.Tech. Software Engineering Graduates

Target Role	Respondents	Percentage
Software Architect / Solutions Architect	24/25	96%
Senior Software Engineer / Lead Engineer	23/25	92%
DevSecOps / Platform Engineer	22/25	88%
Cloud Engineer / Cloud Solutions Developer	20/25	80%
Software Quality Assurance / Test Automation Lead	18/25	72%
Engineering Manager / Technical Programme Manager	16/25	64%
Security Engineer / Application Security Specialist	15/25	60%
Software Product Manager (Technical Track)	12/25	48%

8. Findings — Q5: Industry Partnership Interest

Respondents were asked whether they would engage formally with the University as industry partners covering guest faculty, lab sessions, capstone project hosting, internships, recruitment letters of intent, and curriculum advisory input. The enthusiasm reflects industry's recognition that academia-industry collaboration is essential to producing work-ready, senior software engineers at the pace the market demands.

Partnership Engagement Mode	Respondents	Percentage
Guest faculty / expert lecture & workshop sessions	23/25	92%
Industry project / capstone problem host	22/25	88%
Structured 6-month internship provider	21/25	84%
Capstone pitch panelist & industry evaluator	20/25	80%
Letter of Intent for recruitment	18/25	72%
Curriculum advisory board membership	16/25	64%
Cloud lab / sandbox environment access for students	13/25	52%
Co-branding / sponsorship of hackathons & student competitions	10/25	40%

Selected Partner Quotes — Industry Partnership

> *"We are ready to provide real-world software engineering challenges as capstone problem statements. Our legacy modernization and microservices migration projects are perfect thesis material — students who work on these problems graduate with experience that money cannot buy."*
>
> **— Head of Engineering, BFSI Sector**

> *"If M.Tech. students complete a structured six-month internship on live production systems, they understand CI/CD pressures, incident management, and release cycles first-hand. We can absorb four to five interns per year from a well-structured programme and convert the best to full-time hires."*
>
> **— VP Engineering, Ahmedabad IT Products Firm**

> *"We will issue a formal Letter of Intent without hesitation. If the programme is structured around real engineering depth — system design, security, DevSecOps — and not just coding tutorials, we want to be a founding partner from Batch 1."*
>
> **— CTO, Technology Services Company, Gandhinagar**

9. Overall Programme Recommendation Rating

Respondents were asked to rate overall whether they would recommend that the University proceed with launching the M.Tech. in Software Engineering programme.

Overall Rating	Count	%	Meaning
5 — Strongly recommend launching now	15	60%	Launch immediately
4 — Recommend with minor conditions	9	36%	Launch with adjustments
3 — Neutral / more information needed	1	4%	Cautious
2 — Do not recommend	0	0%	—
1 — Strongly against	0	0%	—
Average Rating: 4.6 / 5.0			**Strongly Positive**

10. Conclusions & Recommendations

The industry validation survey of 25 organizations produces five clear and actionable conclusions specific to the M.Tech. Software Engineering programme:

01

The software engineering talent gap is structural, acute, and universally confirmed.

Every organization surveyed identified one or more critical gaps in fresh graduates. The inability to design large-scale systems, implement DevSecOps pipelines, and develop cloud-native applications are the top three deficiencies — precisely the competencies the University's proposed M.Tech. curriculum is designed to deliver.

02

The recruitment market is ready to hire from the programme from Day 1.

An average recruitment intent score of 4.4/5.0 with 88% of organizations rating 4 or 5 demonstrates that the placement pipeline for the first batch is already in formation. M.Tech.-qualified software engineers command a 40–55% salary premium over B.Tech. counterparts, creating an exceptional graduate outcome environment that will attract high-quality applicants.

03

Software Architecture and DevSecOps are the two highest-priority curriculum areas.

100% and 96% of respondents respectively identified Software Architecture & System Design and DevSecOps & CI/CD Engineering as the most critical curriculum areas. This validates embedding these competencies as programme core courses across Semesters I and II, with cloud-native development and distributed systems forming the applied engineering backbone of the curriculum.

04

Industry partnership interest is exceptional and immediately actionable.

84% of organizations are open to providing structured internships, 72% are prepared to issue Letters of Intent for recruitment, and 64% are willing to serve on the curriculum advisory board. This gives the programme the credibility infrastructure it needs at launch and ensures the 6-month industry internship in Semester III is operationalized from Day 1 with real engineering organizations.

05

The overall recommendation is to proceed with the programme launch without delay.

96% of respondents gave a rating of 4 or 5. A 4.6/5.0 average recommendation score is one of the strongest industry validation signals a new academic programme can receive.

NEP 2020 and the UGC PG Credit Framework 2024 provide complete regulatory authorization. Market conditions, talent deficit data, and employer commitment all point to an immediate launch of the M.Tech. Software Engineering programme.

11. References & Citations

All citations embedded in the body of this Annexure correspond to the following authoritative sources. These references were verified as of April 2026 and are provided to substantiate the market demand analysis, regulatory compliance positioning, and curriculum validation findings presented herein.

[1] Structured Industry Validation Survey — M.Tech. Software Engineering, University (Internal Document), February–April 2026. Survey administered to 25 senior technology leaders across Gujarat and pan-India.

[2] NASSCOM Technology & Talent Report 2025. 'India requires over 1.4 million additional software professionals with advanced engineering competencies by 2027.' Available at: https://nasscom.in

[3] World Economic Forum. Future of Jobs Report 2025. Software and Application Developers identified among five largest growth occupations globally, 26% growth projected 2025–2030. Available at: https://www.weforum.org/publications/the-future-of-jobs-report-2025/

[4] UGC. Curriculum and Credit Framework for Post-graduate Programmes (June 2024). University Grants Commission, Ministry of Education, Government of India. Available at: https://www.ugc.gov.in

[5] AICTE (All India Council for Technical Education). Approval requirements for technical and management education programmes in India. Available at: https://www.aicte-india.org/

[6] Careers360. 'UGC aligns PG programme frameworks with NEP 2020; features, credit requirements.' June 16, 2024.

[7] IIT Bombay. M.Tech. Computer Science & Engineering — Programme Structure and Curriculum (2024–25). NEP 2020-aligned basket-based course structure.

[8] IIT Madras. M.Tech. Software Engineering — Curriculum Overview and Elective Framework (2024).

[9] UGC. Curriculum and Credit Framework for Post-graduate Programmes (June 2024), p. 8. Eligibility criteria for M.E./M.Tech. programmes under NEP 2020.

[10] LinkedIn Talent Insights India 2025. 'Software Architect and Senior Software Engineer roles grew 68% year-on-year between 2023–2025 driven by enterprise digital transformation.' Available at: https://www.linkedin.com/business/talent/blog/talent-strategy/india-hiring-trends

[11] NASSCOM & Zinnov. 'Engineering Talent in India: Closing the Depth Gap.' 2025. Fewer than 25% of fresh engineering graduates are deployment-ready for senior software roles without upskilling.

[12] Indeed India Hiring Tracker 2025. DevSecOps and Cloud Engineering roles among fastest growing software engineering positions. Available at: https://www.indeed.com/career/software-engineer/salaries/in/india

[13] Glassdoor India Salary Report 2025. M.Tech.-qualified Software Engineers earn 40–55% premium over B.Tech. counterparts in equivalent roles. Available at: https://www.glassdoor.co.in

[14] Ministry of Electronics and Information Technology (MeitY). India's Digital Economy Vision 2025 — Workforce Skill Imperatives. Demand for cloud-native and enterprise software architects cited as critical.

[15] Pondicherry University. M.Tech. Computer Science & Engineering — Regulations, Curriculum & Syllabus under NEP 2020 / UGC PG Credit Framework 2024. Effective 2025–26.

[16] Reserve Bank of India (RBI). IT Governance Framework for Regulated Entities 2024. Compliance-aware software engineering competencies required in BFSI technology teams.

[17] Digital Personal Data Protection Act (DPDP), Government of India, 2023. Secure software development and data engineering compliance requirements for technology organizations.

[18] Dheya Career Mentors & NASSCOM. 'Software Engineering Career Outlook India 2026: Skills, Salaries & Roadmap.' Available at: https://www.dheya.com/insights/software-engineering-career-india-2026