AC:22/2/2025
Item No:1.1.7

SIES (Nerul) College of Arts, Science & Commerce
RISE WITH EDUCATION

SIES (Nerul) College of Arts, Science and Commerce (Autonomous)

B.Sc.(Information Technology)

Sr. No.	Heading	Particulars
1.	Title of the course	B. Sc.(Information Technology)
2.	Minimum percentage for admission	45%
3.	Passing Marks	40%
4.	Semesters	V
5.	Level	UG
6.	Pattern	3-4 years & 6-8 semesters Choice Based Grading System
7.	Status	New
8.	To be implemented from	From Academic year 2025-26 in a progressive manner

Date: 22nd February, 2025.

Signature:
Koel RC
Dr. Koel Roychoudhury

AC Chairperson

SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS) ★
NERUL
NAVI
MUMBAI
400 706.

Dr. Meghna Bhatia

Head of the Department

SIES (Nerul) College of Arts, Science & Commerce
RISE WITH EDUCATION

SIES (Nerul) College of Arts, Science and Commerce (Autonomous)

(Affiliated to University of Mumbai)

RE-ACCREDITED GRADE “A” BY NAAC (3rd CYCLE)

BOARD OF STUDIES
SYLLABUS FOR
B.Sc (Information Technology)

(WITH EFFECT FROM THE ACADEMIC YEAR 2025-2026)

OBJECTIVES OF THE PROGRAMME:

- To strengthen the fundamentals and basics of Information Technology and to boost technical development skills.
- To develop the ability to apply the knowledge acquired through analysis of algorithms, mathematical and statistical techniques
- To develop skills to work efficiently in designing the solution to various software and hardware problems by implementing various tools.
- To inculcate professional ethics, managerial skills to encourage entrepreneurship and promote teamwork to manage diverse projects.
- To enhance employability skills and provide scope for higher education and research in the field of information technology.

SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS) ★
NERUL
NAVI
MUMBAI
400 706.

B.Sc. Information Technology Programme

(To be implemented from Academic Year- 2025-26)

Semester V			
Course Code	**Course Type**	**Course Title**	**Credit**
U25IT5MJ01	Major (Core)	Artificial Intelligence	3
U25IT5MJP01		Artificial Intelligence Practical	1
U25IT5MJ02	Major (Core)	Data Storage Techniques	3
U25IT5MJP02		Data Storage Techniques Practical	1
U25IT5MJ03	Major (Core)	Cryptography in Ancient India	2
U25IT5E01	Elective -I	Information and Network Security	3
U25IT5EP01		Information and Network Security	1
U25IT5E02	Elective -II	Java Script and Allied Technologies -I	3
U25IT5EP02		Java Script and Allied Technologies -I Practical	1
U25IT5MI01	Minor	Decision Making Techniques	3
U25IT5MIP01		Decision Making Techniques Practical	1
U25CC5CEP01	CEP	Community Engagement Project	4
Total			**22**

SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS) ★ NERUL NAVI MUMBAI 400 706.

Major (Credit 4(3+1))

Artificial Intelligence

COURSE CODE: U25IT5MJ01 **COURSE CREDIT: 03**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

- To study the distinction between optimal reasoning Vs. human like reasoning
- To understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.
- To get an idea on different knowledge representation techniques.
- To understand the applications of AI, namely game playing, theorem proving
- To realize problems under uncertainty and acquire machine learning algorithms

Course Outcomes:

After successful completion of this course, students would be able to :

1. Articulate the historical development and current trends in Artificial Intelligence, demonstrating a comprehensive understanding of its foundations and principles.
2. Demonstrate proficiency in implementing and analyzing various search algorithms, utilizing both uninformed and informed strategies to solve complex problems efficiently.
3. Apply adversarial search techniques to decision-making in competitive environments, including games,and effectively manage uncertainty and partial observability.
4. Demonstrate competency in logical reasoning and inference, utilizing propositional and first-order logic to represent and solve real-world problems in AI applications.
5. Gain practical experience in planning algorithms and generative AI techniques, enabling them to design and implement AI systems capable of planning actions and generating novel content autonomously.

Unit	Topic	Hours
I	**Introduction to AI**: Definition and Scope,Intelligent Agents,Problem-Solving Agent **Search Strategies**: Uninformed Search: BFS, DFS, Uniform Cost Search, Iterative Deepening DFS, Bidirectional Search,Informed Search: Greedy Best-First Search, A* Search, Heuristic Functions, **Beyond Classical Search**: Hill Climbing, Simulated Annealing, Local Search, Searching with Non-Deterministic Actions, Searching with Partial Observations,Online Search Agents and Unknown Environments	15
II	**Adversarial Search**:Games and Optimal Decisions,Minimax Algorithm,Alpha-Beta Pruning,Imperfect Real-Time Decisions **Constraint Satisfaction Problems (CSP):**Constraint Propagation,Backtracking Search,Local Search for CSPs,Problem Structures in CSPs **Propositional Logic:Knowledge-Based Agents** and The Wumpus World,Propositional Theorem Proving: Inference, Proof by Resolution, Horn and Definite Clauses,Forward and Backward Chaining,Model Checking and Agent-Based Logic	15
III	**First-Order Logic**: Syntax and Semantics, Using First-Order Logic for AI,Knowledge Engineering in FOL **Inference in First-Order Logic**: Propositional vs. First-Order Inference, Unification and Lifting,Forward and Backward Chaining,Resolution Methods **Ethics of AI:** Benefits and risks of artificial intelligence,ethical challenges of AI,AI code of ethics,Risks of AI	15

Books and References:

Text Books

1. Artificial Intelligence: A Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

References

1. Artificial Intelligence, 3rd Edn., E. Rich and K. Knight (TMH)

2. Artificial Intelligence, 3rd Edn., Patrick Henny Winston, Pearson Education.

3. Artificial Intelligence, Shivani Goel, Pearson Education.

4. Artificial Intelligence and Expert systems – Patterson, Pearson Education

Artificial Intelligence Practicals

COURSE CODE: U25IT5MJP01 **COURSE CREDIT: 01**

1 credit - 15 lectures **1 lecture is 120 minutes**

Course Objectives:

- To study the distinction between optimal reasoning Vs. human like reasoning
- To understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.
- To get an idea on different knowledge representation techniques.
- To understand the applications of AI, namely game playing, theorem proving
- To realize problems under uncertainty and acquire machine learning algorithms

Course Outcomes:

After successful completion of this course, students would be able to :

1. Articulate the historical development and current trends in Artificial Intelligence, demonstrating a comprehensive understanding of its foundations and principles.
2. Demonstrate proficiency in implementing and analyzing various search algorithms, utilizing both uninformed and informed strategies to solve complex problems efficiently.
3. Apply adversarial search techniques to decision-making in competitive environments, including games,and effectively manage uncertainty and partial observability.
4. Demonstrate competency in logical reasoning and inference, utilizing propositional and first-order logic to represent and solve real-world problems in AI applications.
5. Gain practical experience in planning algorithms and generative AI techniques, enabling them to design and implement AI systems capable of planning actions and generating novel content autonomously.

Sr.No.	List of Practicals
1.	**Write programs for the following:** a. Implement depth first search algorithm. b. Implement breadth first search algorithm.
2	**Write programs for the following:** a. Implement A* algorithm.
3	**Write programs for the following:** a. Implement hill climbing problem.
4	**Write programs for the following:** Simulate tic – tac – toe game using min-max algorithm.

5	**Write program for the following:** Design an application to simulate number puzzle problem.
6	**Write program for the following:** Solve constraint satisfaction problem.
7	**Write programs for the following:** Simulate 4-Queen / N-Queen problem
8	**Write programs for the following:** Water Jug Problem
9	**Write programs for the following:** a. Derive the expressions based on Associative Law. b. Derive the expressions based on Distributive Law.
10	Write programs for the following: Solve tower of Hanoi problem
11	**Write programs for the following:** a. Shuffle deck of cards.
12	**Write program for the following:** a. Derive the predicate. (for e.g.: Sachin is batsman, batsman is cricketer) - > Sachin is Cricketer - > dhoni is wicketkeeper - > Ronaldo is a footballer and footballer is not cricketer
13	Write a program which contains three predicates: male, female, parent. Make rules for following family relations: father, mother, grandfather, grandmother, brother, sister, uncle,aunt, nephew and niece, cousin. Question: i. Draw Family Tree. ii. Define: Clauses, Facts,Predicates and Rules with conjunction and disjunction
14	**Write program for the following:** a.Build a simple rule-based expert system using Python (e.g., a medical diagnosis system)
15	**Write program for the following:** a.Build a simple rule-based logic using Python to suggest career path

Major (Credit 4(3+1))

Data Storage Techniques

COURSE CODE: U25IT5MJ02 **COURSE CREDIT: 03**

1 credit :15 lectures **1 lecture is 60 minutes**

Course Objectives:

- Explore advanced storage techniques in relational and non-relational databases.
- Understand distributed and scalable data storage systems.
- Learn emerging trends and future-oriented data storage technologies.

Course Outcomes:

After successful completion of this course, students would be able to :

1. Develop a thorough understanding of relational database storage methods.
2. Learn indexing and optimization strategies for enhanced database performance.
3. Explore transaction management and concurrency control techniques.
4. Understand NoSQL databases and their practical applications.
5. Study distributed storage techniques, including sharding, replication, and consistency, along with big data file systems and modern trends in data storage.

Unit	Topic	Hours
I	**Relational Database Storage Architecture** Physical storage mechanisms: Pages, extents, tablespaces, Row-oriented vs. column-oriented storageIndexing mechanisms: B-trees, B+ trees, and hash-based indexes. **Storage Optimization Techniques** Data partitioning: Horizontal and vertical partitioning.Denormalization strategies: Balancing performance and redundancy. **Transactions and Concurrency Control** ACID properties in-depth.Multi-version Concurrency Control (MVCC).Locking mechanisms and deadlock resolution.	15
II	**Introduction to NoSQL Databases** Overview of NoSQL: Key-value stores, document databases, column-family stores, and graph databases.Comparative analysis of NoSQL vs. relational databases.Use cases for NoSQL systems: MongoDB, Cassandra, Neo4j. **Distributed Data Storage Concepts** Fundamentals of distributed systems: Sharding and replication.CAP theorem: Consistency, Availability, Partition Tolerance trade-offs.Distributed databases: Cassandra, DynamoDB. **Distributed File Systems**	15

	Hadoop Distributed File System (HDFS).Google File System (GFS).Characteristics and challenges of distributed file systems.	
III	**Modern Big Data Storage** Overview of big data storage systems: Apache Hadoop,Data serialization formats: Parquet, Avro, and Protobuf.Real-time storage and streaming systems: Apache Kafka **Cloud Storage Systems** Object storage, block storage, and file storage in cloud platforms.Data lakes vs. data warehouses.Case studies of cloud storage systems: Google Cloud Storage	15

Books and References:

Textbooks:

1. "Database Systems: The Complete Book" by Hector Garcia-Molina et al.
2. "Designing Data-Intensive Applications" by Martin Kleppmann.
3. “Fundamentals of Database Systems” by Elmasri R, & Navathe S(2017), 7th Edition,Pearson Education
4. Database System Concepts by Silberschatz,A ,Korth , H.F & Sudarshan,S,(2011),6th Edition,Tata McGraw-Hill Education.
5. Practical MongoDB,Architecting,Developing and Administering MongoDB by Shankuntala Edward, Navin Sabharwal,Apress

Supplementary Materials:

1. MongoDB and Cassandra documentation.
2. Online resources for Hadoop, Spark, and cloud platforms.

Data Storage Techniques Practical

COURSE CODE: U25IT5MJP02 **COURSE CREDIT: 01**

1 credit - 15 lectures **1 lecture is 120 minutes**

Course Objectives:

- Explore advanced storage techniques in relational and non-relational databases.
- Understand distributed and scalable data storage systems.
- Learn emerging trends and future-oriented data storage technologies.

Course Outcomes:

After successful completion of this course, students would be able to :

1. Develop a thorough understanding of relational database storage methods.
2. Learn indexing and optimization strategies for enhanced database performance.
3. Explore transaction management and concurrency control techniques.
4. Understand NoSQL databases and their practical applications.
5. Study distributed storage techniques, including sharding, replication, and consistency, along with big data file systems and modern trends in data storage.

Sr.No.	List of Practicals
1.	**Schema Design and Normalization:**Design a normalized database schema for a real-world application. a. Create a schema for a Student Management System/Hospital Management System or any case . b. Normalize the schema to 3NF/BCNF and denormalize specific sections.
2.	**Indexing and Query Optimization:**Test the impact of indexing on query performance. a. Create clustered and non-clustered indexes. b. Use EXPLAIN in MySQL to analyze query execution plans.
3.	**Data Partitioning:** Implement and evaluate horizontal and vertical partitioning. a. Partition a large dataset based on time intervals (horizontal). b. Split table columns based on access frequency (vertical).
4.	**NoSQL Database Setup and CRUD Operations** Install and configure a NoSQL database for document storage. a. Install MongoDB. b. Create a database, drop database and perform basic CRUD operations.

5.	**Schema Design in NoSQL Databases:**Design a schema for a document or key-value database. a. Model a database for an e-commerce platform. b. Perform data insertion and queries.
6.	**Perform MongoDB Queries on the sample document -I** a. Download restaurants.json file b. Perform basic queries with selection and projection operations.
7.	**Perform MongoDB Queries on the sample document -II** a. Download movies.json file. b. Perform queries with indexing, sorting,regular expressions
8.	**Implementing Aggregation** a. Write a MongoDB query to use sum, avg, min and max expression. b. Write a MongoDB query to use push and addToSet expression. c. Write a MongoDB query to use the first and last expression.
9.	**Replication, Backup and Restore** a. Write a MongoDB query to create Replica of existing database. b. Write a MongoDB query to create a backup of an existing database. c. Write a MongoDB query to restore database from the backup.
10.	**Sharding in Distributed NoSQL Databases :** Implement sharding to scale data storage. a. Configure sharding in MongoDB. b. Distribute data across shards and analyze query performance.
11.	**CAP Theorem Analysis :** Demonstrate the trade-offs between consistency, availability, and partition tolerance. a. Configure a distributed database to simulate partition tolerance. b. Test eventual consistency under network partition.
12.	**Working with Distributed File Systems:**Store and retrieve large files using a distributed file system. a. Upload files to Hadoop Distributed File System (HDFS). b. Analyze block distribution and replication.
13.	**Java and MongoDB** Connecting Java with MongoDB and inserting, retrieving, updating and deleting.
14.	**Big Data Storage with Apache :** Load and process large datasets. a. Use RDDs for transformations and queries.
15.	**Cloud Storage Solutions:** Store and retrieve data using cloud storage. a. Configure Google Cloud Storage.

Major (Credit 2)

Cryptography in Ancient India

COURSE CODE: U25IT5MJ03 **COURSE CREDIT: 02**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

- To understand the historical context and significance of cryptography in Ancient India, exploring its role in governance, military strategy, and society.
- To learn and analyze the various classical cryptographic techniques used in Ancient India, including systems like Mlecchita Vikalpa and KaTaPaYaDi.
- To compare Indian cryptographic methods with those from other ancient civilizations such as Greece, Rome, and Egypt to appreciate their uniqueness and cultural relevance.

Course Outcomes:

After successful completion of this course, students would be able to :

1. Students will gain an in-depth understanding of cryptographic practices in Ancient India and their applications in various domains.
2. Students will be able to identify and explain the cryptographic systems used in India, such as the KaTaPaYaDi System and Kautilya's ciphers.
3. Students will develop the ability to compare and contrast Indian cryptography with those from other ancient civilizations, recognizing their contributions to modern cryptography principles.

Unit	Topic	Hours
I	**Cryptography in ancient India**: Definition and Purpose of Cryptography,Historical Context of Cryptography in India,Role of Cryptography in Ancient Indian Society,Role of Cryptography in Ancient Indian Governance and Military Strategy **Classical Indian Cryptographic Techniques:** Mlecchita Vikalpa, KaTaPaYaDi System, Kautilya's Arthashastra, Muladeviya and Kautiliyam Ciphers, Chari and Chakra Ciphers.	15
II	**Cryptography in Ancient India** Ramshalaka, Bhutasamkhya System, KAutilyama, Muladivaya, Arysbhata's Substitution Code, Gaudhayojya.	15

	Comparison with Cryptographic Techniques from Other Civilizations: Ancient Greek Cryptography, Ancient Rome Cryptography, Ancient Egypt Cryptography, Ancient Hebrew Cryptography, Ancient Arab Cryptography.	

Books and References:

1. *Cryptography in Ancient India* by Pramod Ghogare
2. C.P. Bauer, Secret history : The story of cryptography, CRC Press, 5, 2013.
3. S. Singh, The code book, Doubleday, New York, 9, 1999.
4. Sanada, M. J., Segal, M. J., Chanchanbi, K., & Singh, K. M. (2019). *Chronology of cryptology*. *JETIR*, 6(6)

SCHEME OF EXAMINATION

The scheme of examination shall be divided into two parts:

- **Internal assessment 40% i.e. 20 marks**
- **Semester end examination 60% i.e. 30 marks**

(A) Internal Assessment 20 marks

Description	**Marks**
Assignment	10
Class participation and Attendance	10

(B) External Assessment 30 marks

Individual or Group Project

Design a project integrating ancient Indian cryptographic principles with modern IT technologies. For example:

- Developing a basic cipher using ancient Indian principles.
- Creating a digital platform for encrypting Sanskrit or Vedic texts using modern encryption techniques.

Elective -I (Credit 4(3+1))

Information and Network Security

COURSE CODE: U25IT5E01 **COURSE CREDIT: 03**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

- To provide students with knowledge of basic concepts of computer security including network security and cryptography.
- Understand network security threats, security services, and countermeasures.
- Understand vulnerability analysis of network security.
- Acquire background on hash functions; authentication; firewalls; intrusion detection techniques.

Course Outcomes:

After successful completion of this course, students would be able to :

1. Understand the principles and practices of cryptographic techniques.
2. Understand a variety of generic security threats and vulnerabilities, and identify & analyze particular security problems for a given application.
3. Understand various protocols for network security to protect against the threats in a network.

Unit	Topic	Hours
Unit I	**Introduction:** Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms **Classical Encryption Techniques:** Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Steganography, Block Cipher Principles, The Data Encryption Standard, The Strength of DES, AES (round details not expected), Multiple Encryption and Triple DES, Block Cipher Modes of Operation, Stream Ciphers **Public-Key Cryptography and RSA:** Principles of Public-Key Cryptosystems, The RSA Algorithm	15
Unit II	**Key Management:** Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange **Message Authentication and Hash Functions**: Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm, HMAC **Digital Signatures and Authentication:** Digital Signatures, Authentication Protocols, Digital Signature Standard **Authentication Applications:** Kerberos, X.509 Authentication, Public-Key Infrastructure	15

Unit III	**Electronic Mail Security:** Pretty Good Privacy, S/MIME **IP Security:** Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management **Web Security:** Web Security Considerations, Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction **Intrusion:** Intruders, Intrusion Techniques, Intrusion Detection **Malicious Software:** Viruses and Related Threats, Virus Countermeasures, DDOS **Firewalls:** Firewall Design Principles, Types of Firewalls	15

Books and References:

1)Cryptography and Network Security, Atul Kahate, Tata McGraw-Hill, 2013.

2) Cryptography and Network, Behrouz A Fourouzan, Debdeep Mukhopadhyay, 2nd Edition,TMH,2011.

3)Cryptography And Network Security Principles And Practice, Fourth or Fifth Edition, William Stallings, Pearson

Information and Network Security Practical

COURSE CODE: U25IT5EP01 **COURSE CREDIT: 01**

1 credit - 15 lectures **1 lecture is 120 minutes**

Sr.No.	List of Practicals
1	Implement Caesar Cipher
2	Implement Monoalphabetic Cipher
3	Implement Vernam Cipher
4	Implement Playfair Cipher with a key entered by the user.
5	Implement Affine Cipher with equation c=3x+12
6	Implement Rail fence technique
7	Implement Simple Columnar Transposition technique
8	Implement DES algorithm
9	Implement AES algorithm
10	Implement an RSA algorithm to perform encryption / decryption of a given string.
11	Write a program to implement the Diffie-Hellman Key Agreement algorithm to generate symmetric keys.
12	Write a program to implement the MD5 algorithm to compute the message digest.
13	Write a program to calculate HMAC-SHA1 Signature
14	Write a program to implement SSL.
15	Configure Windows Firewall to block: - A port - An Program - A website

SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS) ★ NERUL NAVI MUMBAI 400 706.

Elective -II (Credit 4(3+1))

Java Script and Allied Technologies-I

COURSE CODE: U25IT5E02 **COURSE CREDIT: 03**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

- To introduce core JavaScript and Node.js concepts for building dynamic web applications.
- To explain asynchronous programming, error handling, and debugging techniques.
- To provide an understanding of backend development with Node.js and MongoDB.

Course Outcomes:

After successful completion of this course, students would be able to :

1. Demonstrate proficiency in JavaScript programming and DOM manipulation.
2. Build server-side applications using Node.js and connect to databases.
3. Apply best practices for error handling, debugging, and asynchronous code.

Unit	Topic	Hours
I	**Unit 1: JavaScript Fundamentals** Introduction to Programming and JavaScript Basics: Overview of Programming Concepts, Introduction to JavaScript Syntax Variables, Data Types, and Operators: Declaring Variables (var, let, const), Understanding Primitive Data Types and Operators Control Structures and Functions: Using Conditional Statements, Loops, Function Declarations, and Arrow Functions DOM Manipulation and Events: Accessing and Modifying the DOM, Handling Basic Events	15
II	**Unit 2: Core Node.js Concepts** Introduction to Node.js: Overview, Environment Setup, and Understanding Non-blocking I/O Modules, npm, and Express Basics: Working with Built-in Modules, Package Management, and Setting Up Express Building a Basic HTTP Server and REST API: Creating a Server with the http Module, Introduction to REST Principles and CRUD Operations Working with Databases: Introduction to MongoDB and Performing CRUD Operations using Mongoose.	15

SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS) ★
NERUL
NAVI
MUMBAI
400 706.

III	**Unit 3: Advanced Node.js and Full-Stack Integration** Advanced Asynchronous Programming: Deep Dive into Promises, Async/Await, and Managing Concurrency Real-time Applications and Security: Building a Simple Chat App with WebSockets, and Implementing Security Best Practices Testing and Deployment: Unit Testing with Mocha/Chai and Deploying Node.js Applications (Heroku/AWS)	15

Books and References:

1. Haverbeke, Marijn. Eloquent JavaScript. 3rd ed., No Starch Press, 2018.
2. Lassoff, Mark. JavaScript and Node.js Fundamentals. 2nd ed., 2020.
3. Holmes, Simon. Node.js for Beginners. CreateSpace Independent Publishing Platform, 2017.
4. Mead, Andrew. Learning Node.js Development. 2nd ed., Packt Publishing, 2017.
5. Buckler, Craig. Node.js: Novice to Ninja. 1st ed., SitePoint, 2018.
6. Clements, David Mark. Node.js 14 Cookbook. Packt Publishing, 2020.

Java Script and Allied Technologies-I Practicals

COURSE CODE: U25IT5EP02 **COURSE CREDIT: 01**

1 credit - 15 lectures **1 lecture is 120 minutes**

Course Objective:

- To apply JavaScript and Node.js skills in building interactive and dynamic web applications.
- To develop hands-on experience with tools like Express.js, MongoDB, and WebSockets.
- To gain experience in testing, deploying, and debugging applications locally.

Course Outcome:

1. Create interactive web pages and backend services using JavaScript and Node.js.
2. Implement real-time communication and database operations in applications.
3. Test, debug, and deploy web applications on local environments.

Sr. No	List of Practicals
1	Create a "Hello World" webpage using basic HTML and JavaScript.
2	Practice declaring variables using var, let, and const, and perform arithmetic operations.
3	Implement control structures (if-else, switch) and write functions, including arrow functions.
4	Manipulate the DOM by selecting elements and modifying their content and styles dynamically.
5	Develop a simple interactive webpage (e.g., a calculator or to-do list) integrating variables, functions, and DOM manipulation.
6	Set up the Node.js environment and create a basic HTTP server that responds to client requests.
7	Build a simple web server using Express.js, setting up routes to handle different HTTP requests.
8	Develop a REST API with CRUD operations using Express.js.
9	Connect a Node.js application to MongoDB and perform basic database operations (Insert, Read, Update, Delete).
10	Use Express middleware for logging and error handling; implement dynamic routing with URL parameters.
11	Fetch data asynchronously using fetch() and display it on a webpage, handling JSON responses.

12	Build a real-time chat application using WebSockets and Socket.io for bi-directional communication.
13	Write unit tests for Node.js applications using Mocha and Chai to ensure functionality.
14	Implement version control with Git by creating a local repository and pushing code to GitHub.
15	Develop a full-stack application integrating a Node.js backend with a frontend framework (e.g., React or Vue).

Minor (Credit 4(3+1))

Decision Making Techniques

COURSE CODE: U25IT5MI01 **COURSE CREDIT: 03**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

- To introduce students to the concept of probability distributions, including discrete and continuous random variables, and to help them understand how to use probability functions to solve real-world problems.
- To familiarize students with the process of hypothesis testing, including key concepts such as null and alternative hypotheses, p-values, and Type I/Type II errors, enabling them to make data-driven decisions.
- To teach students how to analyze time series data, identify its components, and apply forecasting methods like the Least Squares and Moving Average methods for accurate predictions.

Course Outcome:

After successful completion of this course, students would be able to:

1. Apply discrete and continuous probability distributions, including the Binomial and Normal distributions, to calculate probabilities and analyze data.
2. conduct hypothesis tests to draw conclusions about population parameters from sample data.
3. Analyze and forecast time series data, identifying trends, seasonal patterns, and using methods such as Least Squares and Moving Averages to make predictions.

Unit	Details	Lectures
I	**Probability Distribution** Discrete Random Variable, Probability Function, Probability Distribution, Continuous Random Variable, Probability Density Function (PDF), Cumulative Density Function (CDF), Binomial Distribution: Introduction, Probability mass function of Binomial distribution, Mean and Variance of Binomial distribution, Properties of Binomial Distribution, Uses of Binomial Distribution,Normal Distribution:Introduction, Probability density function of Normal distribution, Properties of Normal distribution, Importance of Normal Distribution.	15
II	**Hypothesis Testing** Concept of hypothesis in statistics, Null hypothesis (H0) vs. Alternative hypothesis (H1), Type I and Type II errors, Significance level (α) and	

	power of a test, The p-value interpretation, One-tailed vs. two-tailed tests, Z-tests, One-sample Z-test, Two-sample Z-test for comparing means, Test for Proportion, Test for difference of proportion, T-tests, One-sample t-test, Independent two-sample t-test, Concept of ANOVA, One-way ANOVA, Chi-square goodness-of-fit test.	15
III	**Time Series** Introduction to Time series, Components of Time series, Least Square Method, Moving Average Method, Determination of Seasons. **Decision Theory** Decision making situation, Decision maker, Courses of Action, States of Nature, Pay-off and Pay-off matrix, Decision making under uncertainty, Maximin, Maximax, Minimax regret and Laplace criterion, Simple examples to find optimum decision, Formulation of Pay-off matrix. **Probabilistics (Decision making under Risk)** EMV, EOL, EVPI	15

Books and References:

1. Statistics for Management by *Richard I. Levin and David S. Rubin*, Seventh Edition, Pearson.
2. Fundamental of Applied Statistics by *S. C. Gupta and V. K. Kapoor*, Sultan Chand & Sons.
3. Statistics by *Murray R. Spiegel and Larry J. Stephens*, Third edition, Schauma's Outlines.
4. Fundamentals of Mathematical Statistics by *S. C. Gupta, V. K. Kapoor* 12th Edition. Sultan Chand & Sons.

Decision Making Techniques Practicals

COURSE CODE: U25IT5MIP01 **COURSE CREDIT: 01**

1 credit - 15 lectures **1 lecture is 120 minutes**

Course Objectives:

- To introduce students to the concept of probability distributions, including discrete and continuous random variables, and to help them understand how to use probability functions to solve real-world problems.
- To familiarize students with the process of hypothesis testing, including key concepts such as null and alternative hypotheses, p-values, and Type I/Type II errors, enabling them to make data-driven decisions.
- To teach students how to analyze time series data, identify its components, and apply forecasting methods like the Least Squares and Moving Average methods for accurate predictions.

Course Outcome:

After successful completion of this course, students would be able to:

1. Apply discrete and continuous probability distributions, including the Binomial and Normal distributions, to calculate probabilities and analyze data.
2. conduct hypothesis tests to draw conclusions about population parameters from sample data.
3. Analyze and forecast time series data, identifying trends, seasonal patterns, and using methods such as Least Squares and Moving Averages to make predictions.

Sr.No.	List of Practicals
1.	Examples based on Probability definition.
2	Examples based on elementary Theorems of probability.
3	Implement examples based on Conditional probability.
4	Examples based on PDF and CDF.
5	Implement examples on calculation of probability, mean and variance based on Binomial distribution.
6	Implement examples on calculation of probability based on Normal distribution.
7	Examples based on Z-test.
8	Examples based on T-test.
9	Examples based on Chi-Square test.

10	Examplcs bascd on onc way ANOVA.
11	Examples based on thc Least Square method.
12	Examples based on the Moving Average method.
13	Examples based on Maximin, Maximax, Minimax regret and Laplace criterion.
14	Examples based on EMV, EOL, EVPI.
15	Real life examples on decision making.

Revised Scheme of Examination
Faculty of Science
(Undergraduate Programme)

SCHEME OF EXAMINATION (for 100 marks 3 credits)

The scheme of examination shall be divided into two parts:

- **Internal assessment 40% i.e.40 marks**
- **Semester end examination 60% i.e.60 marks**

(A)Internal Assessment 40 marks

Description	Marks
An internal test of 20 marks Q.1 a. Multiple choice Questions - 05 Marks b. True/False - 05 Marks Q.2. Attempt 2 questions out of 3 questions (5 marks each) - 10 Marks OR **Online MCQ test**	20
Presentation/Case Studies/Assignments	15
Attendance and Class Participation	5
Total	40

B) Semester End examination 60 marks

PAPER PATTERN

Duration: 2 hours	
Total Marks:60	
Q.1 15 marks OR 15 marks (7 and 8 marks)-Unit 1	15
Q.2 15 marks OR 15 marks (7 and 8 marks)-Unit 2	15
Q.3 15 marks OR 15 marks (7 and 8 marks)-Unit 3	15
Q.4 15 marks-attempt any 3 out of 6 (from Unit 1, Unit 2, Unit 3)	15
Total	60

Passing criteria: Minimum 40% in Internal (16 out of 40) and 40% (24 out of 60) in semester-end examination.

SCHEME OF EXAMINATION (for 50 marks ,2 credits)

The scheme of examination shall be divided into two parts:

- **Internal assessment 40% i.e.20 marks**
- **Semester end examination 60% i.c.30 marks**

(A)Internal Assessment 20 marks(Theory)

Description	**Marks**
An internal test of 10 marks Q.1 Multiple choice Questions/True or False - 05 Marks Q.2. Attempt 1 questions out of 2 questions (5 marks each) - 05 Marks OR **Online MCQ test**	10
Presentation/Case Studies/Assignments	05
Attendance and Class Participation	05
Total	20

OR

(A)Internal Assessment 20 marks(Practical)

Description	**Marks**
Practical Question	10
Journal	05
Viva	05
Total	20

B) Semester End examination 30 marks

PAPER PATTERN

Duration: 1 hour	
Total Marks:30	
Q.1 15 marks OR 15 marks (7 and 8 marks)-Unit 1	15
Q.2 15 marks OR 15 marks (7 and 8 marks)-Unit 2	15
Total	30

OR

PAPER PATTERN(1 credit Theory)

Duration: 1 hour	
Total Marks:30	
Q.1 15 marks OR 15 marks (7 and 8 marks)-Unit 1	15
Q.2 15 marks OR 15 marks (7 and 8 marks)-Unit 1	15
Total	30

Passing criteria: Minimum 40% in Internal (08 out of 20) and 40% (12 out of 30) in semester-end examination.

SCHEME OF PRACTICAL EXAMINATION

(for 50 marks , 1 credit)

Semester end examination 50 marks

PAPER PATTERN

Duration: 2.5 hours	
Total Marks:30	
Q.1 Practical Q1	20
Q.2 Practical Q2	20
Viva	05
Journal	05
Total	50

Passing Standards

The learners to pass a course shall have to obtain a minimum of 40% marks in each head of passing, consisting of Internal Assessment and Semester End Examination. The learners shall obtain a minimum of 40% marks (i.e. 16 out of 40 or 8 Out of 20) in the Internal Assessment and 40% marks in the Semester End Examination (i.e. 24 Out of 60 or 12 Out of 30) separately, to pass the course and a minimum of Grade D, wherever applicable, to pass a particular semester. A learner will be said to have passed the course if the learner passes the Internal Assessment and Semester End Examination together.

AC:22/2/2025
Item No:1.1.7

SIES (Nerul) College of Arts, Science & Commerce
RISE WITH EDUCATION

SIES (Nerul) College of Arts, Science and Commerce (Autonomous)
B.Sc.(Information Technology)

Sr. No.	Heading	Particulars
1.	Title of the course	B. Sc.(Information Technology)
2.	Minimum percentage for admission	45%
3.	Passing Marks	40%
4.	Semesters	VI
5.	Level	UG
6.	Pattern	3-4 years & 6-8 semesters Choice Based Grading System
7.	Status	New
8.	To be implemented from	From Academic year 2025-26 in a progressive manner

Date: 22nd February, 2025.

Signature:

Dr. Koel Roychoudhury

AC Chairperson

SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS)
NERUL
NAVI
MUMBAI
400 706.

Dr. Meghna Bhatia

Head of the Department

SIES (Nerul) College of Arts, Science & Commerce
RISE WITH EDUCATION

SIES (Nerul) College of Arts, Science and Commerce (Autonomous)

(Affiliated to University of Mumbai)

RE-ACCREDITED GRADE "A" BY NAAC (3rd CYCLE)

BOARD OF STUDIES
SYLLABUS FOR
B.Sc (Information Technology)

(WITH EFFECT FROM THE ACADEMIC YEAR 2025-2026)

OBJECTIVES OF THE PROGRAMME:

- To strengthen the fundamentals and basics of Information Technology and to boost technical development skills.
- To develop the ability to apply the knowledge acquired through analysis of algorithms, mathematical and statistical techniques
- To develop skills to work efficiently in designing the solution to various software and hardware problems by implementing various tools.
- To inculcate professional ethics, managerial skills to encourage entrepreneurship and promote teamwork to manage diverse projects.
- To enhance employability skills and provide scope for higher education and research in the field of information technology.

SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS) ★
NERUL
NAVI
MUMBAI
400 706.

B.Sc. Information Technology Programme

(To be implemented from Academic Year- 2025-26)

Semester VI			
Course Code	**Course Type**	**Course Title**	**Credit**
U25IT6MJ01	Major (Core)	Advanced Web Technology using .NET	3
U25IT6MJP01		Advanced Web Technology using .NET Practical	1
U25IT6MJ02	Major (Core)	Mastering JQuery & JSON	3
U25IT6MJP02		Mastering JQuery & JSON Practical	1
U25IT6MJ03	Major (Core)	UI/UX	1
U25IT6MJP03		UI/UX Practical	1
U25IT6E01	Elective -I	Ethical Hacking	3
U25IT6EP01		Ethical Hacking Practical	1
U25IT6E02	Elective -II	Java Script and Allied Technologies -II	3
U25IT6EP02		Java Script and Allied Technologies -II Practical	1
U25IT6MI01	Minor	Mathematics for Machine Learning	3
U25IT6MIP01		Mathematics for Machine Learning Practical	1
U25IT6OJT01	OJT	On Job Training	4
Total			**22**

Major (Credit 4(3+1))

Advanced Web Technology using .NET

COURSE CODE: U25IT6MJ01 **COURSE CREDIT: 03**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

- Introduce C# basics, OOP concepts, and SOLID principles for clean code development.
- Teach how to build dynamic web apps using ASP.NET Core and Entity Framework with state management
- Introduce design patterns, AJAX, and web security for building responsive and secure web apps.

Course Outcomes:

After successful completion of this course, students would be able to :

1. Students will learn core **C# programming** and **SOLID principles** to write maintainable and reusable code.
2. Students will create **ASP.NET Core MVC** applications using **Entity Framework** for database management and handle **state management**.
3. Students will apply **design patterns**, **AJAX**, and **security practices** to develop responsive and secure web applications.

Unit	Topics	Hours
I	**Unit 1: Introduction to .NET, C# Basics & SOLID Principles** **Overview of .NET:**.NET Core vs .NET Framework, CLR (Common Language Runtime) **C# Programming Essentials:**Variables, Data Types, and Operators,Loops, Conditional Logic, and Methods,Introduction to OOP: Classes, Objects, and Inheritance **SOLID Principles & Design Patterns:**What are SOLID Principles?,Simple examples of Factory, Singleton, and Repository Patterns	15
II	**Unit 2: ASP.NET Core, Entity Framework & Data Management** **ASP.NET Core Basics:**Understanding MVC (Model-View-Controller),Working with Razor Pages for dynamic web content,Creating Controllers and Views	15

	Entity Framework Core: Using Entity Framework (EF) Core for database access,Performing CRUD operations (Create, Read, Update, Delete),Code-First approach and handling Migrations **State Management:**Using SessionState, ViewState, and Cookies for data storage across pages,Implementing Authentication with ASP.NET Identity	
III	**Unit 3: Design pattern,Web Security, AJAX & Deployment** **Advanced Design Patterns:**Implementing the Repository Pattern,Introduction to Dependency Injection **Web Security:**Understanding Security Requirements, Introduction to Authentication and Authorization,Implementing Role-Based Access Control (RBAC) **AJAX and Bootstrap:**Using AJAX for partial page updates,Designing responsive web pages with Bootstrap	15

Books and References:

1. Pro ASP.NET Core MVC 2" by Adam Freeman
2. Professional C# and .NET 5" by Christian Nagel
3. Pro C# 9 with .NET 5" by Andrew Troelsen & Philip Japikse
4. ASP.NET Core in Action" by Andrew Lock
5. Entity Framework Core in Action" by Jon P. Smith
6. Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides
7. Web Application Security: Exploitation and Countermeasures for Modern Web Apps" by Andrew Hoffman
8. AJAX and JavaScript: A Beginner's Guide" by Sanjaya Sharma

Advanced Web Technology using .NET Practical

COURSE CODE: U25IT6MJP01 **COURSE CREDIT: 01**

1 credit - 15 lectures **1 lecture is 120 minutes**

Course Objectives:

1. To gain practical experience in building web applications using C#, ASP.NET Core, Entity Framework, and AJAX.

Course Outcome:

By the end of this course, students will be able to:

- Develop dynamic, secure web applications with ASP.NET Core and Entity Framework, and apply design patterns and AJAX for real-time functionality.

Sr. No	Practical
1	Create a C# console application to demonstrate variables, data types, and operators.
2	Write a program using loops and conditional logic.
3	Design a class in C# with attributes and methods. Implement inheritance.
4	Implement the Factory pattern in C#.
5	Create a Singleton class in C#.
6	Implement the Repository pattern in C#.
7	Create a "Hello World" application with ASP.NET Core MVC.
8	Build a CRUD application using Entity Framework.
9	Develop a Razor Pages app to display dynamic content.
10	Implement sessions and cookies for state management.
11	Build a user authentication system using ASP.NET Identity.
12	Create a CRUD application with Entity Framework and AJAX.
13	Implement Dependency Injection in an ASP.NET Core MVC application.
14	Create an application with Role-Based Access Control (RBAC).
15	Build a responsive page using Bootstrap and implement AJAX for dynamic updates.

Major (Credit 4(3+1))

Mastering JQuery & JSON

COURSE CODE: U25IT6MJ02 **COURSE CREDIT: 03**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

1. Understand the Role of jQuery and JSON – Learn the fundamentals of JavaScript, the need for jQuery, and the importance of JSON in web development.
2. Develop Proficiency in jQuery – Gain hands-on experience in selecting, traversing, and manipulating DOM elements using jQuery.
3. Enhance User Interaction with jQuery – Implement event handling, animations, and effects to create dynamic web applications.
4. Work with AJAX and JSON – Learn how to fetch, parse, and manipulate JSON data using AJAX and integrate it with APIs.
5. Apply JSON in Various Contexts – Explore JSON handling in different programming languages, databases, and APIs, including validation and advanced concepts.

Course Outcomes:

After successful completion of this course, students would be able to :

1. Efficient DOM Manipulation – Demonstrate the ability to select, traverse, and modify DOM elements dynamically using jQuery.
2. Interactive Web Applications – Implement event-driven programming and animations to enhance user experience.
3. Seamless Data Handling with JSON – Parse, validate, and manipulate JSON data effectively in JavaScript and other programming languages.
4. API Integration and Data Communication – Utilize AJAX and Fetch API to send and receive JSON data from RESTful APIs.
5. Advanced JSON Implementation – Apply JSON Schema validation, work with NoSQL/SQL databases, and implement authentication with JWT.

Unit	Topic	Hours
I	**Introduction to jQuery:**Understanding JavaScript and the Need for jQuery,Setting Up jQuery in a Project ,Basics of the Document Object Model (DOM), Writing and Executing jQuery Code **Selecting and Traversing Elements:** CSS Selectors in jQuery Traversing the DOM (Parent, Child, and Sibling Elements) ,Chaining Methods and Filtering	15

	Manipulating the DOM: Changing HTML and Text Content, Modifying CSS and Styles,Adding and Removing Elements Dynamically **Handling Events in jQuery:**Event Listeners: Click, Hover, Keypress, Focus, Blur,Event Delegation and Propagation,Preventing Default Actions and Stopping Event Bubbling **Animations and Effects:** Basic Animation with .animate() Fading and Sliding Effects.Custom Animations and Timing **AJAX and JSON with jQuery:** Introduction to AJAX ,Fetching and Parsing JSON Data ,Working with APIs and Dynamic Content **Writing jQuery Plug-ins:** Understanding the Structure of jQuery Plug-ins, Creating and Using Custom Plug-ins, Adding Options and Callbacks	
II	**Introduction to JSON:** Overview of JSON JSON vs XML: Key Differences Advantages and Applications of JSON **JSON Syntax and Structure:** JSON Data Types (String, Number, Boolean, Null, Object, Array) ,JSON Objects and Arrays, Nested JSON Structures, Formatting and Validating JSON **Working with JSON in JavaScript :** Creating and Parsing JSON in JavaScript , JSON.stringify() and JSON.parse(), Handling JSON Data in JavaScript Applications, Working with Arrays and Objects in JSON **JSON in Different Programming Languages :** JSON Handling in Python (json module) , PHP (json_encode, json_decode)	15
III	**JSON and APIs:** Understanding REST APIs and JSON, Fetch API and AJAX Requests with JSON ,Handling JSON Responses and Errors Sending JSON Data in API Requests **JSON with Databases :** Storing JSON in NoSQL Databases (MongoDB) , JSON in SQL Databases (MySQL), Querying JSON Data from Databases. **JSON Schema and Validation:** Introduction to JSON Schema Defining JSON Structure with Schema, Validating JSON Data **Advanced JSON Concepts:** JSON Web Tokens (JWT) for Authentication. JSONPath for Querying JSON Data , JSONP (JSON with Padding) and Cross-Origin Requests	15

Books and References:

1. Beginning jQuery Jack Franklin Russ Ferguson Apress Second
2. Beginning JSON Ben Smith Apress
3. Head First jQuery by Ryan Benedetti
4. JSON and AJAX with JavaScript for beginners
5. Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON 1st Edition by Lindsay Bassett

Mastering JQuery & JSON Practical

COURSE CODE: U25IT6MJP02 **COURSE CREDIT: 01**

1 credit - 15 lectures **1 lecture is 120 minutes**

Course Objectives:

1. Understand the Role of jQuery and JSON – Learn the fundamentals of JavaScript, the need for jQuery, and the importance of JSON in web development.
2. Develop Proficiency in jQuery – Gain hands-on experience in selecting, traversing, and manipulating DOM elements using jQuery.
3. Enhance User Interaction with jQuery – Implement event handling, animations, and effects to create dynamic web applications.
4. Work with AJAX and JSON – Learn how to fetch, parse, and manipulate JSON data using AJAX and integrate it with APIs.
5. Apply JSON in Various Contexts – Explore JSON handling in different programming languages, databases, and APIs, including validation and advanced concepts.

Course Outcomes:

After successful completion of this course, students would be able to :

1. Efficient DOM Manipulation – Demonstrate the ability to select, traverse, and modify DOM elements dynamically using jQuery.
2. Interactive Web Applications – Implement event-driven programming and animations to enhance user experience.
3. Seamless Data Handling with JSON – Parse, validate, and manipulate JSON data effectively in JavaScript and other programming languages.
4. API Integration and Data Communication – Utilize AJAX and Fetch API to send and receive JSON data from RESTful APIs.
5. Advanced JSON Implementation – Apply JSON Schema validation, work with NoSQL/SQL databases, and implement authentication with JWT.

Sr.No.	Topic	Hours
1.	**Basic jQuery Selectors and Manipulation:**Understand how to select and manipulate elements using jQuery. a. Create an HTML page and apply jQuery to change the background color of a <div>.	15
2	**DOM Traversal and Filtering:** Learn how to navigate and filter elements in the DOM. a. Select parent, child, and sibling elements using .parent(), .children(), and .siblings(). b. Use .find(), .filter(), and .each() to work with multiple elements.	15

3	**Handling Events in jQuery** Implement various event handlers using jQuery. a. Attach click, hover, focus, and blur events to elements. b. Demonstrate event delegation using .on(). c. Prevent default form submission and stop event propagation.	15
4	**jQuery Animations and Effects:** : Implement animations and effects using jQuery. a. Create fade-in, fade-out, and slide effects on elements. b. Use .animate() to create custom animations. c. Implement an image slider using fade effects.	
5	**AJAX and JSON Handling with jQuery:**: Fetch and display dynamic content using AJAX. a. Load JSON data from a file or API using .ajax() or .getJSON(). b. Display API data dynamically on a webpage.	
6	**Creating a Simple jQuery Plug-in**: Learn how to create a custom jQuery plug-in. a. Develop a plug-in that highlights specific elements on the page. b. Add options and default settings to the plug-in and apply the plug-in to multiple elements.	
7	**jQuery Advanced** a. jQuery Animation effects, jQuery Chaining b. jQuery Callback, jQuery Get and Set Contents c. jQuery Insert Content, jQuery Remove Elements and Attribute	
8	**Basic JSON Structure and Syntax: :** Understand JSON data format and create JSON objects. a. Create a JSON file with sample data (e.g., student details, employee records). b. Validate JSON using an online JSON validator. c. Convert JSON to a string and parse it back to an object in JavaScript.	
9	**Parsing and Stringifying JSON in JavaScript :** Convert JSON data to and from JavaScript objects. a. Use JSON.stringify() to convert an object to a JSON string. b. Use JSON.parse() to convert a JSON string back to an object. c. Modify object properties dynamically and display them in the console.	
10	**Working with Nested JSON Data** :Access and manipulate nested JSON objects and arrays. a. Create a JSON object with nested arrays (e.g., product categories with subcategories). b. Use JavaScript to extract and display specific values. c. Modify values within nested JSON structures.	

11.	**Fetching JSON Data Using AJAX and Fetch API : :** Retrieve JSON data from a server. a. Usc fetch() or XMLIIttpRequest to retrieve JSON from a URL. b. Display the fetched data dynamically in an HTML table. c. Handlc crrors and display appropriate messages.	
12	**Handling JSON Data in Different Programming Languages :** Read, write, and manipulate JSON in different languages. a. Write Python code to load and parse a JSON file using the json module. b. Use PHP's json_encode() and json_decode() to handle JSON data.	
13	**Storing and Retrieving JSON Data in Databases**: Store JSON data in NoSQL and SQL databases. a. Store JSON data in MongoDB and retrieve it using queries. b. Use MySQL's JSON data type to insert and retrieve JSON records.	
14	**Validating JSON Using JSON Schema** :Define and validate JSON data structures. a. Create a JSON Schema for validating user details.	
15.	**Using practical 14 perform:** b.Use an online tool or a JavaScript library to validate JSON data. c.Modify the schema to enforce required fields and data types.	

Major (Credit 2(1+1))

UI /UX

COURSE CODE: U25IT6MJ03 **COURSE CREDIT: 01**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

1. To provide knowledge of user centred methods in design
2. To introduce students to key UI/UX principles while addressing specific needs.

Course Outcome :

After successful completion of this course, students would be able to:

1. Create meaningful and useful designs
2. Understand iterative user-centred design of graphical user interfaces
3. Apply the design thinking to different devices and requirements

Unit	Topic	Hours
Unit I	What is User Interface Design (UI) -The Relationship Between UI and UX, Roles in UI/UX, A Brief Historical Overview of Interface Design, Design thinking and Core Stages of Design Thinking – Divergent and Convergent Thinking – Brainstorming and Game Storming – Observational Empathy Introduction to UX Basics- Foundation of UX design, Good and poor design, Understanding Your Users, Designing the Experience Elements of user Experience, Visual Design Principles, Functional Layout, Interaction design, Introduction to the Interface, Navigation Design, User Testing, Developing and Releasing Your Design	15

Books and References:

1. Universal principles of Design - William Lidwell, Kritina Holden, Jill Butler
2. Design of Everyday life – Don Norman 3
3. Universal methods of design – Brus hanignton
4. The Elements of User Experience: User-Centred Design for the Web and Beyond, Second Edition Jesse James Garrett, Pearson Education. 2011.
5. The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, Third Edition Wilbert O. Galitz , Wiley Publishing, 2007.

UI /UX Practical's

COURSE CODE: U25IT6MJP03 **COURSE CREDIT:** 01

1 credit - 15 lectures **1 lecture is 120 minutes**

Course Objectives:

1. To provide hand on of user centred methods in design
2. To key UI/UX principles while addressing specific needs.

Course Outcome:

After successful completion of this course, students would be able to:

1. Create meaningful and useful designs
2. Understand iterative user-centred design of graphical user interfaces
3. Apply the user interfaces principles to different devices and requirements

Sr.No.	**List of Practicals**
1	Identify problems in five interfaces in overall navigation, look and feel of the interface, relevance of the information. Record all findings. (Design tool).
2	Designing a Responsive Webpage layout for an application using HTML and CSS or Figma
3	Design a Webpage exploring various UI Interaction Patterns.
4	Developing an Interface with Proper UI Style Guides.
5	Design and develop wireflow diagram for application using open-source software
6	Exploring various open-source collaborative interface Platform
7	Hands on Design Thinking Process for a new product
8	Suggesting and Brainstorming feature for proposed product
9	Defining the Look and Feel of the new Project.
10	Create a sample pattern library for that product – Mood board, Fonts, Colors based on UI principle
11	Identify a customer problem to solve
12	Conduct end-to-end user research – User Research, Creating Personas, Ideation Process (User Stories, Scenarios), Flow Diagram, Flow Mapping

13	Sketch, design with popular tools and build a prototype.
14	Perform usability testing and identify improvements in above
15	Create prototyping with different interactions – tab, click, hover, delay. for the given user interface and convert created prototype in webpage(s).

Elective 1 (Credit 4(3+1))

Ethical Hacking

COURSE CODE:U25IT6E01 **COURSE CREDIT: 03**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

1. To demonstrate proficiency in reconnaissance and information gathering techniques
2. To identify and assess application security vulnerabilities.
3. To apply ethical hacking methodologies to exploit system vulnerabilities.

Course Outcome:

After successful completion of this course, students would be able to:

1. Demonstrate proficiency in reconnaissance and information gathering techniques
2. Identify and assess application security vulnerabilities.
3. Apply ethical hacking methodologies to exploit system vulnerabilities.

Unit	**Topic**	**Hours**
Unit I	Introduction to Ethical Hacking: Defining Ethical Hacking Understanding Standing the purpose of Ethical Hacking, Hacking Technology Types, Ethical Hacking Phases, Hacktivism, Hacker Classes, Hackers and Crackers, Skills Required for an Ethical Hacker, Vulnerability Research, Ways to Conduct Ethical Hacking Types of malwares: Worms, Viruses, Trojans, Spyware, Rootkits Types of vulnerabilities: OWASP Top 10: cross-site scripting (XSS), cross site request forgery (CSRF/XSRF), SQL injection, input parameter manipulation, broken authentication, sensitive information disclosure, Broken access control, Security Misconfiguration, Using components with known vulnerabilities, Insufficient Logging and monitoring, CVE Database	15

Unit II	Types of attacks and their common prevention mechanisms: Keystroke Logging, Denial of Service (DoS /DDoS), Waterhole attack, brute force, phishing and fake WAP, Eavesdropping, Man-in-the-middle, Session Hijacking, Clickjacking, Cookie Theft, URL Obfuscation, buffer overflow, DNS poisoning, ARP poisoning, Identity Theft, BOTs and BOTNETs Case-studies on recent attacks – Yahoo, Adult Friend Finder, eBay. Foot printing: Definition, Information Gathering Methodology, Competitive Intelligence, DNS Enumeration, Whois and ARIN Lookups, Types of DNS Records, Traceroute in Foot printing, E-Mail Tracking Social Engineering: Common Types of Attacks Scanning and Enumeration: Port Scanning, Network Scanning, Vulnerability Scanning, CEH Scanning Methodology, Ping Sweep Techniques, Nmap Command Switches, SYN, Stealth, XMAS, NULL, IDLE, FIN Scans, Anonymizers, HTTP Tunnelling Techniques, IP Spoofing Techniques, What Is Enumeration SNMP Enumeration, Null Sessions, Steps Involved in Enumeration	15
Unit III	IDS, Firewalls and Honeypots: Intrusion Detection Systems, Firewalls, Honeypots Physical Security Physical Security ,Physical Security Breach Incidents, Understanding Physical Security ,What Is the Need for Physical Security? Who Is Accountable for Physical Security? Penetration Testing Methodologies, Defining Security Assessments, Overview of Penetration Testing Methodologies , List the Penetration Testing Steps Case Studies on Ethical Hacking.	15

Books and References:

Textbook

CEH official Certfied Ethical Hacking Review Guide, Wiley India Edition

Additional References

1. Gray Hat Hacking - The Ethical Hackers Handbook, Allen Harper, Shon Harris, Jonathan Ness, Chris
2. Penetration Testing: A Hands-on Introduction to Hacking Book by Georgia WeidmanEagle, Gideon Lenkey, and Terron Williams, 3rd Edition, Tata McGraw-Hill.
3. Rafay Baloch, "A Beginners Guide to Ethical Hacking".

Ethical Hacking Practical's

COURSE CODE:U25IT6EP01 **COURSE CREDIT: 01**

1 credit - 15 lectures **1 lecture is 120 minutes**

Course Objectives:

- Understand the terminology and concepts related to ethical hacking and penetration testing.
- Explore various hacking technologies and the skills required to become an ethical hacker.
- Learn the different phases involved in ethical hacking and the methodologies .
- Gain knowledge of common hacking techniques, such as footprinting, scanning, enumeration, and session hijacking.
- Develop proficiency in identifying and exploiting vulnerabilities .

Course Outcome:

After learning the course the students should be able to:.

1. Describe and understand the basics of the ethical hacking
2. Can Perform the footprinting and do basic network commands for getting network information..
3. Demonstrate the techniques for system hacking
4. Characterize the malware and their attacks .

Sr.No.	Practical Name
1.	Understanding and Implementing with DOS prompt / Linux/Mac terminal specific security commands.
2.	Understanding and using Search Engines like Maltego or Shodan for explicit information.
3.	Implement and study the use of network reconnaissance tools like WHOIS, dig, traceroute, nslookup to gather information about networks and domain registers
4.	Implement Password Encryption and Cracking with CrypTool
5.	Use Cain and Abel for cracking Windows account password using Dictionary attack and to decode wireless network passwords.
6.	Implement and Study Linux Network Analysis Commands.
7.	Perform ARP Poisoning in Windows
8.	Implement and Use NMap scanner to perform port scanning of various forms – ACK, SYN, FIN, NULL, XMAS.
9.	Use Wireshark (Sniffer) and Implement Network analysis to capture network traffic on a specific network interface.
10.	Use Nemesy to launch DOS Attack against a target system or network

11.	Simulate persistent cross-site scripting attack.
12.	Session impersonation using Firefox and Tamper Data add-on.
13.	Perform SQL injection attack.
14.	Create a simple keylogger using python
15.	Using Metasploit to exploit (Kali Linux).

Elective 2 (Credit 4(3+1))

Javascript and Allied Technologies-II

COURSE CODE: U25IT6E02 **COURSE CREDIT: 03**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

1. To provide students with a solid understanding of React fundamentals, including components, JSX, state, and props, enabling them to develop dynamic web applications.
2. To teach students the principles of modern web development using React, focusing on best practices, component-based architecture, and the use of hooks for state and lifecycle management.
3. To introduce students to core concepts of front-end development such as event handling, data binding, and conditional rendering, enabling them to build interactive and responsive web applications.

Course Outcomes:

After successful completion of this course, students would be able to:

1. Students will be able to explain the core concepts of React, including the component lifecycle, state management, and the virtual DOM, with an understanding of how these concepts improve web application performance.
2. Students will demonstrate proficiency in using React to build dynamic, stateful, and interactive user interfaces, adhering to modern web development standards and best practices.
3. Students will be able to analyze and debug common issues in React applications, ensuring robust, maintainable, and efficient code.

Unit	Topic	Hours
Unit 1	Introduction to React, Components and Props, State and Event Handling, Conditional Rendering, Lists and Keys, Handling Forms, Lifting State Up, React Lifecycle Methods, Introduction to useState and useEffect, Basic Routing with React Router, Handling Errors, React Context API, React Testing Basics	15
Unit 2	Advanced Hooks, Context API and useContext Hook, React Router Advanced Techniques, Code Splitting and Lazy Loading, Optimizing Performance in React, Styled Components, Error Boundaries and Error Handling, React with APIs, Building Forms in React, Authentication and Authorization, Testing React Applications, State Management with Redux	15
Unit 3	React with Backend (Node.js), React and MongoDB, Deploying a React Application, Building Full-Stack Applications, Authentication with JWT, Building Progressive Web Apps (PWA), Testing Full-Stack Applications	15

Books and References:

1. *Stefanov, Stoyan. React Up & Running. O'Reilly Media, 2016.*
2. *Banks, Alex, and Eve Porcello. Learning React: Modern Patterns for Developing React Apps. O'Reilly Media, 2017.*
3. *Schwarzmüller, Maximilian. React - The Complete Guide. Udemy, 2019.*
4. *Accomazzo, Anthony, Nathaniel Murray, and Ari Lerner. Fullstack React: The Complete Guide to ReactJS and Friends. Fullstack.io, 2017.*
5. *Antonio, Cassio de Sousa. Pro React. Apress, 2019.*
6. *Thomas, Mark Tielens. React in Action. Manning Publications, 2018.*

Javascript and Allied Technologies-II Practicals

COURSE CODE:U25IT6EP02 **COURSE CREDIT: 01**

1 credit - 15 lectures **1 lecture is 120 minutes**

Course Objectives:

1. To enable students to apply React concepts by building practical web applications, including state management, user input handling, and data persistence.
2. To provide hands-on experience with React's ecosystem tools, such as useState, useEffect, and useContext, and teach students how to integrate React with other web technologies like local storage.
3. To help students gain experience in testing, debugging, and deploying React applications, fostering confidence in their ability to work independently on full-fledged React projects.

Course Outcomes:

1. Students will be able to independently develop React applications, from simple UI components to complex, interactive, and stateful applications.
2. Students will be proficient in using React hooks, routing, and component structures to create scalable and maintainable applications.
3. Students will gain the ability to test, debug, and deploy their React projects, ensuring their applications run smoothly in a real-world environment.

Practical No.	List of Practicals
1	Create a "Hello World" React application using create-react-app and render a message to the webpage.
2	Build a simple React component that accepts props and displays them on the page.
3	Create a React component with state and handle a button click event to update the state.
4	Implement conditional rendering based on user input (e.g., show/hide elements based on a button click).
5	Create a list of items in React using .map() and display each item with a unique key.
6	Develop a form in React that takes user input and displays it dynamically on the screen.
7	Implement "Lifting State Up" to share state between sibling components in React.
8	Use React lifecycle methods to fetch data from a local file or API and display it on the screen.

9	Create a React application with routing to switch between different pages using react-router-dom.
10	Use the useState hook to manage the state of a counter and update it on button clicks.
11	Create a React component that uses the useEffect hook to fetch data from a static JSON file or a mock API.
12	Implement form validation and show error messages when the user inputs incorrect data.
13	Build a simple multi-step form in React and manage state across steps.
14	Implement React Context API to manage global state and share data between components without props drilling.
15	Write unit tests for a React component using Jest and React Testing Library to check its functionality.

Minor (Credit 4(3+1))

Mathematics for Machine Learning

COURSE CODE: U25IT6MI01 **COURSE CREDIT: 03**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

1. To understand the fundamental concepts of analytic geometry, including norms, inner products, orthogonality, and orthonormal bases, and their applications in machine learning.
2. To introduce multivariable calculus, focusing on limits, continuity, partial derivatives, and gradient-based methods essential for machine learning algorithms.
3. To explore the mathematical tools used in machine learning, including matrices, Fourier transforms, and Bayesian methods, and their role in optimization and model evaluation.

Course Outcome:

After successful completion of this course, students would be able to:

1. Apply concepts of analytic geometry and vector calculus in the analysis and design of machine learning algorithms.
2. Gain proficiency in using multivariable calculus for optimization tasks such as gradient computation, backpropagation, and automatic differentiation
3. Understand and implement key mathematical techniques, such as matrix operations, Fourier transforms, and error metrics, in solving real-world machine learning problems.

Unit	Details	Lectures
I	**Analytic Geometry** Norms, Inner Products, Lengths and Distances, Angles and Orthogonality, Orthonormal Basis, Orthogonal Complement, Orthogonal Projections, Gram-Schmidt Orthogonalization. **Limit and Continuity of Multivariable Functions** Introduction to multivariable function, examples, Definition of limit, Finding limit of two and three variable function, Definition of Continuity, examples.	15
II	**Vector Calculus** Differentiation of Univariate Functions, Partial Derivatives, Higher order partial derivative, Chain rule, Taylor Series, Gradients of Vector-Valued Functions, Gradients of Matrices, Divergence and Curl, Useful Identities for Computing Gradients, Backpropagation and Automatic	15

	Differentiation, Higher-Order Derivatives, Linearization and Multivariate Taylor Series.	
III	**Application of Math in ML** Matrices in ML, Single value decomposition, Fourier Transform, The infinite Fourier Transform, sine and cosine transform, properties. Equation of line in ML, Bayesian Methods in ML, Sigmoid Function in ML, Entropy in ML,Mean Square Error, Mean Absolute Error, Root mean squared error, R-squared.	15

Books and References:

1. Higher Engineering Mathematics by Dr. B. S. Grewal, Khanna Publications.
2. Mathematics for Machine Learning by Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, Cambridge University Press.
3. Calculus Early Transcendentals by James Stewart.
4. Introduction to the Theory of Fourier Series and Integrals by Harry Dym and Hans Helson.

Mathematics for Machine Learning Practicals

COURSE CODE: U25IT6MIP01 **COURSE CREDIT: 01**

1 credit - 15 lectures **1 lecture is 120 minutes**

Course Objectives:

4. To understand the fundamental concepts of analytic geometry, including norms, inner products, orthogonality, and orthonormal bases, and their applications in machine learning.
5. To introduce multivariable calculus, focusing on limits, continuity, partial derivatives, and gradient-based methods essential for machine learning algorithms.
6. To explore the mathematical tools used in machine learning, including matrices, Fourier transforms, and Bayesian methods, and their role in optimization and model evaluation.

Course Outcome:

After successful completion of this course, students would be able to:

1. Apply concepts of analytic geometry and vector calculus in the analysis and design of machine learning algorithms.
2. Gain proficiency in using multivariable calculus for optimization tasks such as gradient computation, backpropagation, and automatic differentiation
3. Understand and implement key mathematical techniques, such as matrix operations, Fourier transforms, and error metrics, in solving real-world machine learning problems.

Sr.No.	List of Practicals
1.	Write a Python function to compute the norm and inner product of two vectors using NumPy.
2	Given two vectors, compute their Euclidean distance.
3	Implement a function to compute the angle between two vectors using the dot product.
4	Verify if a given set of vectors forms an orthonormal basis.
5	Implement the Gram-Schmidt process to obtain an orthonormal basis from a set of linearly independent vectors.
6	Given a function f(x,y), compute its partial derivatives using SymPy.
7	Compute the Taylor series expansion of e^x around x=0 up to the 4th order.
8	Calculate the gradient of a multivariable function using NumPy or SymPy.
9	Compute the gradient of a function using automatic differentiation.

10	Approximate a multivariable function using the first-order Taylor expansion.
11	Implementation of the concept of matrices in python.
12	Implementation of Bayes Theorem in python.
13	Implementation of equation of line in python.
14	Implementation of Fourier series.
15	Implementation of evaluation metrics in python.

On Job Training (Credit 4)

COURSE CODE:U25IT6OJT01 **COURSE CREDIT: 04**

1 credit - 15 lectures **1 lecture is 60 minutes**

Course Objectives:

- .To provide a practical environment where learners can enhance their practical knowledge.
- To provide the learners the opportunity to integrate theory and practice in his/her professional education

Course Outcome:

Upon successful completion of the On-Job Training (OJT), students will be able to:

1. Apply Theoretical Knowledge to Practical Work – Utilize academic concepts in real-world workplace scenarios to enhance professional competence.
2. Develop Industry-Specific and Technical Skills – Gain hands-on experience, problem-solving abilities, and proficiency in tools and technologies relevant to the field.
3. Enhance Communication and Teamwork Abilities – Demonstrate effective verbal, written, and interpersonal communication while collaborating in a professional environment.
4. Adapt to Workplace Culture and Ethics – Understand organizational behavior, professional responsibility, and ethical standards in the industry.
5. Prepare for Career Growth and Entrepreneurship – Gain exposure to industry trends, job opportunities, and entrepreneurial insights for future career prospects.

Introduction:

- On Job training (OJT) is an important component of the UG program that provides students with a unique opportunity to bridge the gap between theoretical knowledge gained in the classroom and practical application in a real-world environment.
- On Job training (OJT) aims to equip students with both technical and non-technical skills that are essential for success in the industry.
- By participating in OJT, students are able to apply the concepts and technologies learned with implementation during their coursework to real-world scenarios. They gain hands-on experience, problem-solving skills, and a deeper understanding of how the industry operates.
- From an organizational perspective, hosting OJT programs allows companies to gain insights

into the curriculum and content of UG Program. They can provide valuable feedback on the relevance of the coursework and industry requirements, enabling academic institutions to continually improve the program's alignment with industry needs. This collaboration between academia and industry fosters a mutually beneficial relationship, ensuring that graduates are well-prepared for the job market.

Enhancing practical Skills through OJT:

- The On the Job Training (OJT) program spans 4-6 weeks, requiring a minimum of 80 to 120 hours of physical presence at the organization.
- OJT covers any subject within the syllabus, allowing students to align their experience with their academic interests.
- OJT bridges the gap between theoretical knowledge and practical application, preparing students for successful careers in their respective area of interest.

Interning organization: Students have the flexibility to pursue their OJT in various types of organizations, including but not limited to:

Ø Hardware/Manufacturing Firms: Learn about hardware design, manufacturing processes, and quality assurance.

Ø Civic Departments: Engage with local civic departments such as ward offices, post offices, police stations, or panchayats to understand their functioning and contribute to their activities.

Ø Research Centre's/University Departments/Colleges: Contribute as research assistants or in similar roles for research projects or initiatives, fostering collaboration between academia and industry.

Ø Small-Scale Industries/Service Providers: Explore opportunities in diverse sectors such as banking, clinics, NGOs, and professional institutions like CA firms or law firms

Ø Industries/Finance and Insurance sector/Management and Media Development Firms: Gain practical experience and to understand their functioning and contribute to their activities.

Ø Any other organization where technologies can be implemented.

OJT mentors: To enhance the learning experience and ensure the quality of the program, each student

SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS) ★
NERUL
NAVI
MUMBAI
400 706.

participating in the OJT will be assigned two mentors: a faculty mentor from the institution and an industry mentor from the organization where the student is interning.

Industry Mentor Role: The industry mentor plays a crucial role in guiding the student during the internship. They ensure that the internee fulfills the requirements of the organization and successfully meets the demands of the assigned project. Through their expertise and experience, industry mentors provide valuable insights into real-world practices and industry expectations.

Faculty Mentor Role: The faculty mentor serves as the overall coordinator of the OJT program. They oversee the entire internship process and evaluate the quality of the OJT in a consistent manner across all students. The faculty mentor ensures that the OJT aligns with the program's objectives and provides valuable learning opportunities. They also facilitate communication between the institution, industry mentor, and student to ensure a fruitful OJT experience. By having both an industry mentor and a faculty mentor, students benefit from a comprehensive guidance system that combines industry expertise and academic support

Submission of documentation for OJT

The student will make two documents as part of the OJT

1. Online diary: This ensures that the student updates daily activity, which could be accessed by both the mentors. Weekly entry can be of 3- 4 sentences giving a very brief account of the learning/activities/interaction taken place.

2. OJT report: A student is expected to make a report based on the OJT he or she has done in an organization. It should contain the following:

It should contain the following:

1. **Certificate**: A certificate from the organization where the OJT was done.

2. **Title**: A suitable title giving the idea about what work the student has performed during the OJT.

Title Page-Sample format of Title page is given in Appendix 1 of this block. Students should follow the given format. Original Copy of the Approved Performa of the Project Proposal

3.**Abstract**

This should be one/two short paragraphs (100-150 words total), summarizing the project work. It is important that this is not just a restatement of the original project outline. A suggested flow is background, project aims and main achievements. From the abstract, a reader should be able to ascertain if the project is of interest to them and, it should present results of which they may wish to know more details.

4.**Acknowledgements**

This should express student's gratitude to those who have helped in the preparation of the project. Table of Contents: The table of contents gives the readers a view of the detailed structure of the report. The students would need to provide section and subsection headings with associated pages.

Chapter 1: Introduction

The introduction has several parts as given below:

Background: A description of the background and context of the project and its relation to work already done in the area. Summarise existing work in the area concerned with the project work.

Objectives: Concise statement of the aims and objectives of the project. Define exactly what is going to be done in the project; the objectives should be about 30 /40 words.

Purpose, Scope and Applicability: The description of Purpose, Scope, and Applicability are given below:

• Purpose: Description of the topic of the project that answers questions on why this project is being done. How the project could improve the system its significance and theoretical framework.

• Scope: A brief overview of the methodology, assumptions and limitations. The students should answer the question: What are the main issues being covered in the project? What are the main functions of the project?

• Applicability: The student should explain the direct and indirect applications of their work. Briefly discuss how this project will serve the computer world and people.

Achievements: Explain what knowledge the student achieved after the completion of the work. What contributions has the project made to the chosen area? Goals achieved - describes the degree to which the findings support the original objectives laid out by the project. The goals may be partially or fully achieved, or exceeded.

Organisation of Report: Summarising the remaining chapters of the project report, in effect, giving the reader an overview of what is to come in the project report.

Chapter 2: Survey of Technologies

In this chapter Survey of Technologies should demonstrate the students awareness and understanding of Available Technologies related to the topic of the project. The student should give the details of all the related technologies that are necessary to complete the project. The should describe the technologies available in the chosen area and present a comparative study of all those,

Available Technologies. Explain why the student selected the one technology for the completion of the objectives of the project.

Chapter 3: Requirements and Analysis

Problem Definition: Define the problem on which the students are working in the project.Provide details of the overall problem and then divide the problem into sub-problems. Define each sub-problem clearly.

Requirements Specification: In this phase the student should define the requirements of the system,independent of how these requirements will be accomplished. The Requirements Specification describes the things in the system and the actions that can be done on these things. Identify the operation and problems of the existing system.

Planning and Scheduling: Planning and scheduling is a complicated part of software development.Planning, for our purposes, can be thought of as determining all the small tasks that must be carried out in order to accomplish the goal. Planning also takes into account rules, known as constraints,which control when certain tasks can or cannot happen. Scheduling can be thought of as determining whether adequate resources are available to carry out the plan. The student should show the Gantt chart and Program Evaluation Review Technique (PERT).

Software and Hardware Requirements: Define the details of all the software and hardware needed for the development and implementation of the project.

- Hardware Requirement: In this section, the equipment, graphics card, numeric co-processor,mouse, disk capacity, RAM capacity etc. necessary to run the software must be noted.

- Software Requirements: In this section, the operating system, the compiler, testing tools, linker, and the libraries etc. necessary to compile, link and install the software must be listed.

Preliminary Product Description: Identify the requirements and objectives of the new system.Define the functions and operation of the application/system the students are developing as project.Conceptual Models: The student should understand the problem domain and produce a model of the system, which describes operations that can be performed on the system, and the allowable sequences of those operations. Conceptual Models could consist of complete Data Flow Diagrams,ER diagrams, Object-oriented diagrams, Systcm Flowcharts ctc.

Chapter 4: System Design

Describes desired features and operations in detail, including screen layouts, business rules,process diagrams, pseudocode and other documentation.

Basic Modules: The students should follow the divide and conquer theory, so divide the overall problem into morc managcablc parts and dcvelop cach part or module separately. When all modules are ready, thc student should integrate all the modules into one system. In this phase, the student should briefly describe all the modules and the functionality of these modules.

Data Design: Data design will consist of how data is organised, managed and manipulated.

• Schema Design: Define the structure and explanation of schemas used in the project.

• Data Integrity and Constraints: Define and explain all the validity checks and constraints provided to maintain data integrity.

Procedural Design: Procedural design is a systematic way for developing algorithms or procedurals.

• Logic Diagrams: Define the systematic flow of procedure that improves its comprehension and helps the programmer during implementation. e.g., Control Flow Chart, Process Diagrams etc.

• Data Structures: Create and define the data structure used in procedures.

• Algorithms Design: With proper explanations of input data, output data, logic of processes,design and explain the working of algorithms.

User Interface Design: Define user, task, environment analysis and how to map those requirements in order to develop a "User Interface". Describe the external and internal components and the architecture of user interface. Show some rough pictorial views of the user interface and its components.

Security Issues: Discuss Real-time considerations and Security issues related to the project and explain how the student intends avoiding those security problems. What are the security policy plans and architecture?

Test Cases Design: Define test cases, which will provide easy detection of errors and mistakes within a minimum period of time and with the least effort. Explain the different conditions in which the students wish to ensure the correct working of the project.

Chapter 5: Implementation and Testing

Implementation Approaches: Define the plan of implementation, and the standards the students have used in the implementation.

Coding Details and Code Efficiency: Students do not need to include full source code, instead, include only the important codes (algorithms, applets code, forms code etc). The program code should contain comments needed for explaining the work a piece of code does. Comments may be needed to explain why it does it, or why it does a particular way.

The student can explain the function of the code with a shot of the output screen of that program code.

- Code Efficiency: The student should explain how the code is efficient and how the students have handled code optimisation.

Testing Approach: Testing should be according to the scheme presented in the system design chapter and should follow some suitable model – e.g., category partition, state machine-based. Both functional testing and user-acceptance testing are appropriate. Explain the approach of testing.

- Unit Testing: Unit testing deals with testing a unit or module as a whole. This would test the interaction of many functions but, do confine the test within one module.

- Integrated Testing: Brings all the modules together into a special testing environment, then checks for errors, bugs and interoperability. It deals with tests for the entire application. Application limits and features are tested here.

Modifications and Improvements: Once the students finish the testing they are bound to be faced with bugs, errors and they will need to modify your source code to improve the system. Define what modification are implemented in the system and how it improved the system.

Chapter 6: Results and Discussion

Test Reports: Explain the test results and reports based on the test cases, which should show that the project is capable of facing any problematic situation and that it works fine in different conditions.Take the different sample inputs and show the outputs.

User Documentation: Define the working of the software; explain its different functions,components with screenshots. The user document should provide all the details of the product in such a way that any user reading the manual, is able to understand the working and functionality of the document.

Chaptcr 7: Conclusions

Conclusion: The conclusions can be summarised in a fairly short chapter (2 or 3 pages). This chapter brings together many of the points that would have been made in the other chapters.

Limitations of thc Systcm: Explain thc limitations encountered during the testing of the project that the students were not able to modify. List the criticisms accepted during the demonstrations of the project.

Future Scope of the Project describes two things: firstly, new areas of investigation prompted by developments in this project, and secondly, parts of the current work that was not completed due to time constraints and/or problems encountered.

Interaction between mentors: To ensure the smooth conduct of the OJT a meet-up involving the intern, industry mentor, and the faculty mentor will be scheduled as a review. The meeting ensures the synergy between all stakeholders of the OJT.

OJT workload for the faculty: Every student is provided with a faculty member as a mentor. So, a faculty mentor will have a few students under him/her. A faculty mentor is the overall in charge of the OJT of the student. He/she constantly monitors the progress of the OJT by regularly overseeing the diary, interacting with the industry mentor, and guiding on the report writing etc.

Appendix-II

(Proforma for the certificate for internship in official letter head)

This is to certify that Mr./Ms.................... from...................... College has worked as an intern towards the partial fulfilment of _________degree in the academic year ____and has not been submitted for any other examination and does not form part of any other course undergone by the candidate.

The particulars of internship are given below:

Internship starting date: _________________

Internship ending date: ______________

Actual number of days worked: ___________

Tentative number of hours worked: ________________ Hours

Broad area of work: ________________

A small description of work done by the intern during the period:

Signature:

Seal of the organization)

Designation:

Contact details:

Email:

SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS)
NERUL
NAVI
MUMBAI
400 706

Evaluation

1. Presentation of the project implemented using any of the technologies studied during the program.
2. Project documentation.

Rubrics for 100 marks

1. 60 marks for external evaluation
2. 40 marks Internal evaluation

Revised Scheme of Examination
Faculty of Science
(Undergraduate Programme)

SCHEME OF EXAMINATION (for 100 marks 3 credits)

The scheme of examination shall be divided into two parts:

- **Internal assessment 40% i.e.40 marks**
- **Semester end examination 60% i.e.60 marks**

(A)Internal Assessment 40 marks

Description	**Marks**
An internal test of 20 marks Q.1 a. Multiple choice Questions - 05 Marks b. True/False - 05 Marks Q.2. Attempt 2 questions out of 3 questions (5 marks each) - 10 Marks OR **Online MCQ test**	20
Presentation/Case Studies/Assignments	15
Attendance and Class Participation	5
Total	40

B) Semester End examination 60 marks

PAPER PATTERN

Duration: 2 hours	
Total Marks:60	
Q.1 15 marks OR 15 marks (7 and 8 marks)-Unit 1	15
Q.2 15 marks OR 15 marks (7 and 8 marks)-Unit 2	15
Q.3 15 marks OR 15 marks (7 and 8 marks)-Unit 3	15
Q.4 15 marks-attempt any 3 out of 6 (from Unit 1, Unit 2, Unit 3)	15
Total	60

Passing criteria: Minimum 40% in Internal (16 out of 40) and 40% (24 out of 60) in semester-end examination.

SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS) ★
NERUL
NAVI
MUMBAI
400 706.

PAPER PATTERN

Duration: 1 hour	
Total Marks:30	
Q.1 15 marks OR 15 marks (7 and 8 marks)-Unit 1	15
Q.2 15 marks OR 15 marks (7 and 8 marks)-Unit 2	15
Total	30

OR

PAPER PATTERN(1 credit Theory)

Duration: 1 hour	
Total Marks:30	
Q.1 15 marks OR 15 marks (7 and 8 marks)-Unit 1	15
Q.2 15 marks OR 15 marks (7 and 8 marks)-Unit 1	15
Total	30

Passing criteria: Minimum 40% in Internal (08 out of 20) and 40% (12 out of 30) in semester-end examination.

SCHEME OF PRACTICAL EXAMINATION
(for 50 marks , 1 credit)

Semester end examination 50 marks

PAPER PATTERN

Duration: 2.5 hours	
Total Marks:30	
Q.1 Practical Q1	20
Q.2 Practical Q2	20
Viva	05
Journal	05
Total	50

Passing Standards

The learners to pass a course shall have to obtain a minimum of 40% marks in each head of passing, consisting of Internal Assessment and Semester End Examination. The learners shall obtain a minimum of 40% marks (i.e. 16 out of 40 or 8 Out of 20) in the Internal Assessment and 40% marks in the Semester End Examination (i.e. 24 Out of 60 or 12 Out of 30) separately, to pass the course and a minimum of Grade D, wherever applicable, to pass a particular semester. A learner will be said to have passed the course if the learner passes the Internal Assessment and Semester End Examination together.

AC: 30/07/2024
Item No.: 1.2.3

SIES (Nerul) College of Arts, Science & Commerce
RISE WITH EDUCATION

CELEBRATING 25 YEARS 1998-2023

SIES (Nerul) College of Arts, Science and Commerce (Autonomous)
M.Sc.(Information Technology)

Sr. No.	Heading	Particulars
1	Title of the course	M. Sc.(Information Technology) Part II(Sem III and IV)
2	Eligibility for admission	B.Sc(IT,CS,AI,DS), B.E(IT,CS,Electronics,DS) BCA, B.Sc(Physics), B.Sc(Maths), B.Sc(Stats), B.Sc(Electronics).
3	Passing Marks	40%
4	No. of Years / Semesters	02 Semesters per Year
5	Level	PG
6	Pattern	Semester
7	To be implemented from	From Academic year 2024-25 in a progressive manner

Date: 30th July, 2024.

Signature:

SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS)
NERUL
NAVI
MUMBAI
400 706.

Dr. Koel Roychoudhury
AC Chairperson

Dr. Anu Thomas
Head of the Department

SIES (Nerul) College of Arts, Science and Commerce (Autonomous)
(Affiliated to University of Mumbai)
RE-ACCREDITED GRADE "A" BY NAAC (3rd CYCLE)

BOARD OF STUDIES
SYLLABUS FOR
M.Sc (Information Technology)

(WITH EFFECT FROM THE ACADEMIC YEAR 2024-2025)

OBJECTIVES OF THE PROGRAMME:

- Ability to apply the knowledge of Information Technology with recent trends aligned with research and industry.
- Ability to apply IT in the field of Computational Research, Soft Computing, Big Data Analytics, Data Science, Image Processing, Artificial Intelligence, Networking and Cloud Computing.
- Ability to provide socially acceptable technical solutions in the domains of Information Security, Machine Learning, Infrastructure Services as specializations.
- Ability to apply the knowledge of Intellectual Property Rights and Cyber Forensics and various standards in interest of National Security and Integrity along with IT Industry.
- Ability to write effective project reports, research publications and content

SCHEME OF EXAMINATION (for 50 marks ,2 credits)

The scheme of examination shall be divided into two parts:

- **Internal assessment 40% i.e.20 marks**
- **Semester end examination 60% i.e.30 marks**

(A)Internal Assessment 20 marks(Theory)

Description	**Marks**
An internal test of 10 marks Q.1 Multiple choice Questions/True or False - 05 Marks Q.2. Attempt 1 questions out of 2 questions (5 marks each) - 05 Marks OR **Online MCQ test**	10
Presentation/Case Studies/Assignments	05
Attendance and Class Participation	05
Total	20

OR

(A)Internal Assessment 20 marks(Practical)

Description	**Marks**
Practical Question	10
Journal	05
Viva	05
Total	20

B) Semester End examination 30 marks