PPSU

P P SAVANI UNIVERSITY

SCHOOL OF ENGINEERING

M.SC. COMPUTER SCIENCE

SYLLABUS BOOK

AY 2026-27

INSTITUTE VISION
To emerge as an Institute of Excellence by imparting value-based education aided with Research, Innovation and Entrepreneurial skills.

INSTITUTE MISSION	
1	To impart the holistic engineering education of highest quality & prepare socially responsible professionals with entrepreneurial skills.
2	To prepare value-aided engineering professionals to meet up global industry requirements by imparting cutting edge professional education.
3	To inculcate the attitude of research and innovation among the stake holders through experiential and project-based teaching-learning pedagogy.
4	To acquire global talent pool by providing world class amenities for teaching, learning & research.

Graduates will demonstrate ability to:

PEO No	PROGRAMME EDUCATIONAL OBJECTIVES
PEO 1	Lead complex software development projects and deliver enterprise grade solutions by applying advanced computing knowledge and strategic project management skills.
PEO 2	Excel in senior technical roles such as software architects, project managers, or IT consultants by demonstrating strong analytical, leadership, and communication capabilities in industry or entrepreneurship.
PEO 3	Contribute to technological innovation through lifelong learning, applied research, or doctoral studies, continuously adapting to advancements in computing and the software industry.
PO No	**PROGRAMME OUTCOMES**
PO 1	**Advanced Computing Knowledge:** Apply advanced knowledge of computing principles, algorithms, mathematics, and domain-specific concepts to model and solve complex computing problems.
PO 2	**Problem Analysis & Solution Design:** Analyse complex technical problems and propose efficient, scalable, and innovative computing solutions using appropriate methodologies and technologies.
PO 3	**Software Development & System Implementation:** Design, develop, test, deploy, and maintain reliable software applications and platforms using modern development practices, tools, and quality standards.
PO 4	**Research, Innovation & Investigation:** Conduct systematic research, experimentation, and critical analysis to develop innovative solutions and contribute meaningfully to knowledge creation in computing.
PO 5	**Data Analytics & Emerging Technologies:** Apply data analytics, artificial intelligence, machine learning, cloud computing, and cybersecurity to address real-world challenges effectively and responsibly.
PO 6	**Modern Tools & Interdisciplinary Applications:** Leverage advanced computing tools and frameworks to address interdisciplinary problems spanning healthcare, finance, smart systems, and business domains.
PO 7	**Ethics, Communication & Leadership:** Exhibit professional ethics, effective communication, collaborative teamwork, and leadership while upholding social

	responsibility in professional and research settings.
PO 8	**Lifelong Learning & Adaptability:** Commit to continuous learning, adapt to technological advancements, and pursue ongoing professional development, research growth, and innovation throughout the career.

PSO No	PROGRAMME SPECIFIC OUTCOMES (PSO)
PSO 1	Graduates will develop comprehensive technical project reports and effectively present and demonstrate computing solutions to diverse stakeholders and user groups.
PSO 2	Graduates will apply advanced problem-solving skills and computing knowledge to analyse, formulate, and deliver solutions to complex real-world challenges.
PSO 3	Graduates will emerge as technically competent professionals, researchers, and entrepreneurs, capable of excelling in competitive examinations and advancing through higher studies.

Credit Guidelines (General)

One (1) credit represents 30 hours of learning, comprising an appropriate combination of Theory, Practical, Tutorial, Project, Group Discussion, Case Study, Field Visit, Presentation, Assignment, Term Work, Self-Learning, and other curriculum-approved learning activities.

CO-PO Mapping Guidelines

Mapping Level	%age Mapping	Indicator
0	0	No Mapping
1	0-33	Low Level (Slightly Mapped)
2	33-66	Medium Level (Moderately Mapped)
3	>66	High Level (Strongly Mapped)

Syllabus Book

P P Savani University

School of Engineering

Effective From: 2026-27

Authored by: P P Savani University

PPSU P P SAVANI UNIVERSITY NAAC ACCREDITED A+ GRADE

Institute of Computer Science & Application

CONTENT

PPSU
P P SAVANI UNIVERSITY
NAAC ACCREDITED A+

Institute of
Computer Science &
Application

FIRST YEAR M.SC (2026-27)

P P SAVANI UNIVERSITY

INSTITUTE OF COMPUTER SCIENCE & APPLICATION

TEACHING & EXAMINATION SCHEME FOR FIRST YEAR MSc(CS) PROGRAMME AY: 2026-27

Sem	Course Code	Course Title	Offered By	Teaching Scheme										Examination Scheme								
				Direct Contact Hours			Indirect Contact Hours					Total	Credit	Theory	Practical	Tutorial	Project	GD/CS/FV/PPT	Assignment	Term Work	Self Learning	Total
				Theory	Practical	Tutorial	Project	GD/CS/FV/PPT	Assignment	Term Work	Self Learning			CE/ESE	CE/ESE	CE/ESE	CE/ESE	CE/ESE	CE/ESE	CE/ESE	CE/ESE	CE/ESE
1	ICACS7020	Python Programming	CS	30	45	0	0	0	30	0	15	120	04	30 50	30 50	0	0	0	20 00	0	20 00	100 100
	ICACA7010	Programming Concepts with Java	CA	45	60	0	0	0	15	0	0	120	04	30 50	30 50	0	0	0	40 00	0	0	100 100
	ICACA7020	SQL & NoSQL Database Management System	CA	30	45	0	0	15	30	0	0	120	04	30 50	30 50	0	0	20 00	20 00	0	0	100 100
	ICASH7010	Computational Mathematics	S & H	45	0	30	0	0	30	0	15	120	04	30 50	0	30 50	0	0	20 00	0	20 00	100 100
	CFLS7110	Professional & Academic Communication	CFLS	45	0	0	0	0	45	0	0	90	03	100 00	0	0	0	0	0	0	0	100 00
											Total	**570**	**19**									**900**
2	ICACS7010	Information & Network Security	CS	30	45	0	30	0	0	0	15	120	4	30 50	30 50	0	20 00	0	0	0	20 00	100 100
	ICACS7030	Web Development with Python	CS	30	45	0	30	0	0	0	15	120	04	30 50	30 50	0	20 00	0	0	0	20 00	100 100
	ICACA7030	Java Web Application Development	CA	30	45	0	15	0	30	0	0	120	04	30 50	30 50	0	20 00	0	20 00	0	0	100 100
	ICACA7040	Software Quality & Assurance	CA	45	30	0	0	15	30	0	0	120	04	30 50	30 50	0	0	20 00	20 00	0	0	100 100
	ICACS7040	Prompt Engineering & Generative AI	CS	30	45	0	0	0	30	0	15	120	04	30 50	30 50	0	0	0	20 00	0	20 00	100 100
	ICACS7810	Minor Project/Social Impact Project	CS	0	0	0	90	30	0	30	0	150	05	0	0	0	50 50	00 50	0	50 00	0	100 100
											Total	**750**	**25**									**1200**

P P Savani University

School of Engineering

Offered by: Department of Computer Science

COURSE DETAILS:

Course Code	ICACS7020
Course Name	Python Programming
Prerequisite Course/s	None

TEACHING SCHEME:

Credit	Total Learning Hours	Direct Contact Hours			Indirect Contact Hours				
		Theory	Practical	Tutorial	Project	CS	Assignment	Term Work	Self-Learning
04	120	30	45	0	0	0	30	0	15

GD = Group Discussion | CS = Case Study | FV = Field Visit | PPT = Presentation

EXAMINATION SCHEME:

Theory		Practical		Tutorial		Project		CS		Assignment		Term Work		Self-Learning		Total
CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	
30	50	30	50	0	0	0	0	0	0	20	00	0	0	20	00	200

CE: Continuous Evaluation, ESE: End Semester Exam

OBJECTIVE OF THE COURSE:

To help learners to:

- Understand the fundamentals of Python programming and computational problem-solving techniques
- Develop programs using decision-making and looping constructs and modular programming concepts.
- Apply Python data structures and built-in functions for solving real-world problems.
- Design and implement object-oriented applications with proper exception handling mechanisms.
- Utilize file processing and standard Python libraries for developing practical applications.

COURSE CONTENT:

Module No.	Content	Hours	Weightage in%
1	**Introduction to Python Programming:** History and Features of Python, Installation and Configuration, Python IDEs, Program Structure, Variables, Data Types, Input and Output Functions, Operators, Expressions, Type Conversion and Basic Problem Solving.	05	16
2	**Decision Making and Iterative Control Structures:** Conditional Statements (if, if-else, nested if), Looping Statements (for, while), Nested Loops, Break, Continue, Pass Statements, Pattern Generation and Problem-Solving Applications.	05	17
3	**Functions and Modular Programming:** User Defined Functions, Function Arguments, Return Values,	05	17

	Scope and Lifetime of Variables, Recursive Functions, Anonymous Functions (Lambda), Modules and Packages.		
4	**Python Data Structures:** Strings, Lists, Tuples, Sets, Dictionaries, List Comprehensions, String Manipulation, Searching and Sorting Operations, Applications of Python Collections.	05	17
5	**Object Oriented Programming and Exception Handling:** Classes and Objects, Constructors, Instance and Class Variables, Encapsulation, Inheritance, Polymorphism, Method Overriding, Exception Handling, User Defined Exceptions.	05	17
6	**File Handling and Python Libraries:** Text File Handling, CSV File Processing, Regular Expressions, Introduction to NumPy, Pandas and Matplotlib, Mini Application Development.	05	16
	TOTAL	**30**	**100%**

LIST OF PRACTICAL / TUTORIAL:

Sr. No	Practical / Tutorial	Hours
1	Installation of Python Environment, IDE Configuration, and Execution of Basic Python Programs	02
2	Development of Programs using Variables, Data Types, Operators and Expressions	02
3	Programs based on Conditional Statements and Menu-Driven Applications	03
4	Programs using Iterative Statements (for, while) and Pattern Generation	03
5	Development of Applications using User-Defined Functions, Recursion and Lambda Functions	04
6	String Processing Applications (Searching, Validation, Formatting, Palindrome, Frequency Analysis)	03
7	Implementation of List, Tuple, Set and Dictionary Based Applications	04
8	Searching and Sorting Applications using Python Collections	02
9	Development of Modular Programs using Python Modules and Packages	02
10	Design and Implementation of Applications using Classes and Objects	04
11	Programs demonstrating Inheritance, Polymorphism and Method Overriding	04
12	Exception Handling and User Defined Exception Applications	02
13	File Handling Applications using Text, CSV and JSON Files	04
14	Data Analysis and Visualization using NumPy, Pandas and Matplotlib	04
15	Mini Project: Development of a Real-World Python Application	02
	TOTAL	**45**

SELF-LEARNING ONLINE RESOURCES:

Sr. No	Link / Description	Self-Learning Hours
1	https://onlinecourses.nptel.ac.in/noc25_cs56/preview – Video lectures covering Python fundamentals, problem solving, functions, data structures and applications.	03
2	https://swayam.gov.in/ Self-paced video courses on Python programming and computational thinking offered by Indian institutions.	03
3	https://www.coursera.org/specializations/python - Video-based learning on Python programming, data structures, file handling and web applications.	03
4	https://www.edx.org/learn/python/massachusetts-institute-of-technology-	03

	introduction-to-computer-science-and-programming-using-python - Introductory video lectures on Python programming and problem solving.	
5	https://learn.microsoft.com/training/browse/?products=python Guided video modules and hands-on learning resources for Python programming.	03

COURSE EVALUATION & ASSESSMENT RUBRICS:

Component	CE/ESE	Evaluation Methodology / Rubrics	Marks	Passing Marks
Theory	CE	Two Internal Tests of 30 marks each shall be conducted during the semester. The average of both tests will be converted to 30 marks.	30	00
	ESE	Comprehensive written examination covering all modules of the course. Duration: 2 Hours	50	25
Practical	CE	Performance of practical which should be evaluated out of 10 marks for each practical and average of the same will be converted to 30 marks.	30	00
	ESE	Quiz / Practical Performance – 25 Marks External Viva Voce – 25 Marks	50	25
Assignment	CE	Assignment Evaluation Rubrics: Problem Understanding & Analysis – 05 Marks Program Design & Logic Development – 05 Marks Coding Standards & Documentation – 05 Marks Result Validation & Presentation – 05 Marks	20	00
Self Learning	CE	40 MCQ questions based on self-study resources and online learning | 0.5 marks per question | Total = 20 marks	20	00
		TOTAL MARKS	**200**	**100**

TEXT BOOK(S):

Title	Author/s	Publication
Python Programming Using Problem Solving Approach	Reema Thareja	Oxford University Press

REFERENCE BOOK(S):

Title	Author/s	Publication
Core Python Programming	Dr. R. Nageswara Rao	Dreamtech Press
Let Us Python	Yashavant Kanetkar	BPB Publications

COURSE OUTCOME(S):

After the completion of the course, the students will be able to:

ICACS7020	Python Programming
CO 1	Explain Python programming fundamentals, data types, operators, and control structures for solving computational problems.
CO 2	Develop modular programs using functions, recursion, modules, and packages.
CO 3	Apply Python data structures such as strings, lists, tuples, sets, and dictionaries to solve real-world problems.
CO 4	Design and implement object-oriented applications using classes, inheritance, polymorphism, and exception handling mechanisms.

CO 5	Develop Python applications involving file processing and standard libraries such as NumPy, Pandas, and Matplotlib

MAPPING OF CO WITH PO:

ICACS7020	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	1	0	1	0	0	0
CO 2	2	3	2	0	2	0	0	0
CO 3	2	3	3	1	2	0	0	1
CO 4	2	2	3	2	2	0	0	0
CO 5	2	2	3	2	3	1	0	0

MAPPING OF CO WITH PSO:

ICACS7020	PSO1	PSO2	PSO3
CO 1	3	1	0
CO 2	3	2	0
CO 3	2	3	1
CO 4	2	3	2
CO 5	2	3	3

LEVEL OF REVISED BLOOM'S TAXONOMY IN ASSESSMENT

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Introduction to Python Programming	1,2
2	Decision Making and Iterative Statements	2,3
3	Functions and Modular Programming	3
4	Python Data Structures	3,4
5	Object-Oriented Programming and Exception Handling	4,5
6	File Handling and Python Libraries	5,6

P P Savani University

School of Engineering

Offered by: Department of Computer Application

COURSE DETAILS:

Course Code	ICACA7010
Course Name	Programming Concepts with Java
Prerequisite Course/s	None

TEACHING SCHEME:

Credit	Total Learning Hours	Direct Contact Hours			Indirect Contact Hours				
		Theory	Practical	Tutorial	Project	CS	Assignment	Term Work	Self-Learning
04	120	45	60	0	0	0	15	0	0

GD = Group Discussion | CS = Case Study | FV = Field Visit | PPT = Presentation

EXAMINATION SCHEME:

Theory		Practical		Tutorial		Project		CS		Assignment		Term Work		Self-Learning		Total
CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	
30	50	30	50	0	0	0	0	0	00	40	00	0	0	0	0	200

CE: Continuous Evaluation, ESE: End Semester Exam

OBJECTIVE OF THE COURSE:

To help learners to:

- Understand the fundamental concepts of programming and Java application development.
- Apply programming constructs and object-oriented principles to solve real-world problems.
- Develop Java applications using arrays, methods, classes, objects, and exception handling techniques.
- Apply object-oriented concepts and collections to build efficient Java applications.
- Prepare technical documentation and develop lifelong learning skills.

COURSE CONTENT:

Module No.	Content	Hours	Weightage in%
1	**Introduction to Programming and Java:** Introduction to Programming, Algorithms, Flowcharts, History of Java, Features of Java, JDK, JRE, JVM, Byte Code, Program Structure, Variables, Data Types, Operators and Input/Output Operations. Decision Making and Looping: Conditional Statements (if, if-else, nested if, switch), Looping Statements (for, while, do-while), Pattern Programs and Problem-Solving using Loops, User Input using Scanner class.	06	16
2	**Arrays, String & Exception Handling:** Array (1D & 2D), ArrayList and HashMap, String Handling, Collections: String Class, StringBuffer, StringBuilder, Exception Handling (try, catch, finally, throw), User defined Exception handling	06	16

3	**Object-Oriented Programming Concepts:** Encapsulation, Classes, Objects, Constructors, this keyword, Polymorphism- Method Overloading, Static Members, Access Specifier, finalize( ) method, Garbage collection.	06	16
4	**Inheritance, Interfaces & Packages:** Using super creating multilevel Hierarchy, Polymorphism-method overriding, Dynamic method dispatch, abstract classes, using final with Inheritance, Using Package: Defining package, finding package and CLASSPATH, Access protection, importing package, Interface: Defining Interface, Implementing Interface, Variables in Interface.	08	18
5	**Multithreaded Programming:** Java Thread Model, Life cycle of thread, The Thread Class & Implementing Runnable interface, thread methods, using isAlive() & join() methods, thread priority, thread exceptions, Synchronization, Inter thread communication & deadlocks	07	16
6	**GUI Programming with JavaFX & Database Connectivity using JDBC:** Introduction to GUI Programming: AWT vs Swing vs JavaFX, JavaFX Architecture: Stage, Scene, Node hierarchy, Application class lifecycle: init(), start(), stop(), Layout Panes: BorderPane, GridPane, HBox, VBox, StackPane, UI Controls: Label, Button, TextField, TextArea, CheckBox, RadioButton, ComboBox, ListView, TableView, Alert/Dialog boxes, Introduction to JDBC: JDBC Architecture, JDBC Drivers, Setting up JDBC: Connecting Java with MySQL/SQLite, Class.forName(), DriverManager.getConnection(), Statement vs PreparedStatement , Executing Queries: executeQuery(), executeUpdate(), executeBatch(), ResultSet: navigating and extracting data, Form-based CRUD operations: Insert, Update, Delete, Search using PreparedStatement	12	18
	TOTAL	**45**	**100%**

LIST OF PRACTICAL / TUTORIAL:

Sr. No	Practical / Tutorial	Hours
1	Write a Java program to demonstrate variables, data types, operators, and basic input/output operations using the Scanner class.	02
2	Programs using conditional statements, looping constructs and pattern generation	02
3	Write a Java program to perform basic operations (insertion, searching, sorting, traversal) on 1D and 2D arrays.	04
4	Write a Java program to demonstrate the use of ArrayList and HashMap for storing and manipulating a collection of records.	04
5	Write a Java program to demonstrate String, StringBuffer, and StringBuilder operations and compare their performance/mutability.	04
6	Write a Java program to implement exception handling using try, catch, finally, and throw, including a user-defined exception class.	04
7	Write a Java program to demonstrate encapsulation, constructors, and the use of the this keyword through a class-object model.	04
8	Write a Java program to demonstrate method overloading (compile-time polymorphism) and the use of static members and access specifiers.	04
9	Write a Java program to demonstrate multilevel inheritance using the super	04

	keyword and method overriding (runtime polymorphism) with dynamic method dispatch.	
10	Write a Java program to demonstrate abstract classes and interfaces, including implementation of multiple interfaces in a class.	04
11	Write a Java program to create and use a user-defined package, demonstrating access protection and the import mechanism.	04
12	Write a Java program to implement multithreading using both the Thread class and the Runnable interface, demonstrating thread lifecycle and priority.	04
13	Write a Java program to demonstrate thread synchronization and inter-thread communication to solve a producer-consumer type problem.	04
14	Design a JavaFX application with various layout panes (BorderPane, GridPane, HBox/VBox) and UI controls (Label, Button, TextField, ComboBox, TableView) with event handling.	06
15	Write a Java program to connect to a MySQL/SQLite database using JDBC and perform CRUD (Create, Read, Update, Delete) operations through a JavaFX form, using PreparedStatement and displaying results in a TableView.	06
	TOTAL	**60**

COURSE EVALUATION & ASSESSMENT RUBRICS:

Component	CE/ESE	Evaluation Methodology / Rubrics	Marks	Passing Marks
Theory	CE	Two Internal Tests of 30 marks each shall be conducted during the semester. The average of both tests will be converted to 30 marks.	30	00
	ESE	Comprehensive written examination covering all modules of the course. Duration: 2 Hours	50	25
Practical	CE	Performance of practical which should be evaluated out of 10 marks for each practical and average of the same will be converted to 30 marks	30	00
	ESE	Quiz / Practical Performance – 25Marks External Viva Voce – 25 Marks	50	25
Assignment	CE	Assignment questions can involve 1. Legacy Code Autopsy where poorly-written, undocumented Java program needs to be reverse engineered and structured & well documented code must be prepared, report includes before/after class diagrams and a flaw log. 2. Students research a real-world concurrency problem (e.g., banking transactions, ticket booking system, producer-consumer) and write a theoretical report explaining race conditions, deadlocks, and synchronization strategies. 3. SQL Injection Attack & Defense related report writing Assignment Evaluation Rubrics: Problem Understanding & Theoretical Research- 8 Marks Design & Problem-Solving Approach – 8 Marks Implementation & Code Quality – 10 Marks Testing, Evidence & Edge-Case Handling – 8 Marks Documentation & Reflection – 6 Marks	40	00
		TOTAL MARKS	**200**	**100**

TEXT BOOK(S):

Title	Author/s	Publication
Programming with Java	E. Balagurusamy	McGraw-Hill Education
Java: The Complete Reference	Herbert Schildt	McGraw-Hill Education

REFERENCE BOOK(S):

Title	Author/s	Publication
Head First Java	Kathy Sierra & Bert Bates	O'Reilly Media
Core Java Volume I – Fundamentals	Cay S. Horstmann	Pearson Education
Java How to Program	Paul Deitel & Harvey Deitel	Pearson Education
Beginning Java Programming	Barry Burd	Wiley

COURSE OUTCOME(S):

After the completion of the course, the students will be able to:

ICACA7010	Programming Concepts with Java
CO 1	Explain Java programming fundamentals, data types, operators, and control structures.
CO 2	Implement array, string, and collection structures along with exception handling mechanisms to build robust, error-resilient programs.
CO 3	Apply object-oriented programming concepts to develop reusable software solutions.
CO 4	Design multi-class Java applications using inheritance, interfaces, and packages to promote code reusability, modularity, and abstraction.
CO 5	Develop concurrent and database-driven GUI applications by integrating multithreading concepts with JavaFX and JDBC to build functional, real-world desktop solutions.

MAPPING OF CO WITH PO:

ICACA7010	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	0	0	0	0	0	1
CO 2	3	3	2	0	0	0	0	1
CO 3	3	3	3	2	0	1	1	2
CO 4	2	3	2	2	0	2	1	3
CO 5	2	3	3	2	2	2	2	3

MAPPING OF CO WITH PSO:

ICACA7010	PSO1	PSO2	PSO3
CO 1	1	0	1
CO 2	1	0	1
CO 3	2	1	1
CO 4	2	1	2
CO 5	3	2	3

LEVEL OF REVISED BLOOM'S TAXONOMY IN ASSESSMENT

1: Remember	2: Understand	3: Apply

4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Introduction to Programming and Java	1,2
2	Decision Making and Looping	2,3
3	Methods, Arrays and Classes	2,3,4
4	Object-Oriented Programming Concepts	2,3,4
5	String Handling, Exception Handling and Collections	2,3,4,5
6	File Handling and Debugging	3, 4, 5, 6

P P Savani University

School of Engineering

Offered by: Department of Computer Application

COURSE DETAILS:

Course Code	ICACA7020
Course Name	SQL & NoSQL Database Management System
Prerequisite Course/s	None

TEACHING SCHEME:

Credit	Total Learning Hours	Direct Contact Hours			Indirect Contact Hours				
		Theory	Practical	Tutorial	Project	CS	Assignment	Term Work	Self-Learning
04	120	30	45	0	0	15	30	0	0

GD = Group Discussion | CS = Case Study | FV = Field Visit | PPT = Presentation

EXAMINATION SCHEME:

Theory		Practical		Tutorial		Project		CS		Assignment		Term Work		Self-Learning		Total
CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	
30	50	30	50	0	0	0	0	20	00	20	00	0	0	0	0	200

CE: Continuous Evaluation, ESE: End Semester Exam

OBJECTIVE OF THE COURSE:

To help learners to:

- Understand the architecture and design principles of Relational and NoSQL database systems.
- Develop proficiency in writing advanced SQL queries, schema design, and query optimization techniques.
- Explore NoSQL database types including Document (MongoDB), Key-Value (Redis), Column-Family (Cassandra), and Graph (Neo4j) databases through hands-on learning.
- Apply normalization, transaction management, and concurrency control concepts to design robust, consistent databases.
- Build awareness of database security, cloud databases, and emerging database technologies for real-world application development.

COURSE CONTENT:

Module No.	Content	Hours	Weightage in%
1	**Introduction to Databases & SQL Fundamentals:** Introduction to Database Management Systems (DBMS): Concepts, Purpose, Architecture; Types of Databases – Relational, NoSQL, NewSQL; Database vs File System; Relational Model – Tables, Rows, Columns, Keys (Primary, Foreign, Composite, Candidate); Introduction to SQL – History, Standards (SQL-92, SQL:1999, SQL:2003), DDL, DML, DCL, TCL; SQL Data Types – Numeric, Character, Date/Time, Boolean, Large Object; DDL Commands – CREATE, ALTER, DROP, TRUNCATE, RENAME; DML Commands – INSERT,	06	20

	UPDATE, DELETE, SELECT; Constraints – NOT NULL, UNIQUE, CHECK, DEFAULT, PRIMARY KEY, FOREIGN KEY.		
2	**Advanced SQL & Query Optimization:** Advanced SELECT – WHERE, Comparison and Logical Operators, NULL handling; Aggregate Functions – COUNT, SUM, AVG, MIN, MAX; GROUP BY and HAVING clauses; Joins – INNER JOIN, LEFT/RIGHT/FULL OUTER JOIN, CROSS JOIN, SELF JOIN; Subqueries – Scalar, Row, Table, Correlated; EXISTS, IN, ANY, ALL operators; Set Operations – UNION, UNION ALL, INTERSECT, EXCEPT/MINUS; Views – Creating, Updating, Dropping, Materialized Views; Indexes – Types (B-Tree, Hash, Composite, Covering), Index creation and performance; Query Execution Plans – EXPLAIN/EXPLAIN ANALYZE; Query Optimization strategies.	06	20
3	**Database Design, Normalization & Transactions:** Entity-Relationship (ER) Model – Entities, Attributes, Relationships, ER Diagrams; Enhanced ER (EER) – Generalization, Specialization, Aggregation; Functional Dependencies and Armstrong's Axioms; Normal Forms – 1NF, 2NF, 3NF, BCNF; Higher Normal Forms – 4NF, 5NF; Denormalization – When and Why; Transaction Management – ACID Properties, Transaction States, Commit and Rollback; Concurrency Control – Locking (Shared, Exclusive), Two-Phase Locking (2PL), Deadlocks; Recovery Techniques – Log-Based Recovery, Checkpoints, Shadow Paging.	06	20
4	**Introduction to NoSQL Databases:** NoSQL Overview – Motivation, CAP Theorem, BASE Properties vs ACID; Types of NoSQL Databases – Document, Key-Value, Column-Family (Wide-Column), Graph; Document Stores – MongoDB Architecture, Collections, Documents, BSON Format; MongoDB CRUD Operations – insertOne/Many, find, updateOne/Many, deleteOne/Many; MongoDB Querying – Query Operators, Projection, Sorting, Limit, Skip; MongoDB Aggregation Pipeline – $match, $group, $sort, $lookup, $unwind, $project; Key-Value Stores – Redis Architecture, Data Types (Strings, Lists, Sets, Hashes, Sorted Sets); Column-Family Stores – Apache Cassandra Architecture, Keyspaces, Tables, CQL basics.	06	20
5	**Advanced NoSQL, Graph Databases & Cloud Databases:** Advanced MongoDB – Indexes (Single, Compound, Text); Schema Design Patterns; Graph Databases – Neo4j Architecture, Nodes, Relationships, Properties, Labels; Cypher Query Language – MATCH, CREATE, MERGE, RETURN; Graph Use Cases – Social Networks, Recommendation Engines, Fraud Detection.	04	14
6	**Database Security, Administration & Emerging Trends:** Database Security – Authentication, Authorization, Role-Based Access Control (RBAC); SQL Injection Attacks – Types and Prevention using Prepared Statements; Database Backup & Recovery strategies (Full, Incremental); Big Data & Databases – Introduction to Apache Hive.	02	06
	TOTAL	**30**	**100%**

LIST OF PRACTICAL / TUTORIAL:

Sr. No	Practical / Tutorial	Hours
1	Install and configure MySQL/PostgreSQL; create a database; implement DDL commands (CREATE, ALTER, DROP, TRUNCATE, RENAME) on tables with appropriate data types	04
2	Implement DML operations (INSERT, UPDATE, DELETE, SELECT) and apply constraints (NOT NULL, UNIQUE, CHECK, DEFAULT, PRIMARY KEY, FOREIGN KEY) on a relational schema	04
3	Write SELECT queries using WHERE, comparison/logical operators, NULL handling, and Aggregate Functions (COUNT, SUM, AVG, MIN, MAX) with GROUP BY and HAVING	04
4	Implement all types of Joins (INNER, LEFT, RIGHT, FULL OUTER, CROSS, SELF JOIN) on multi-table datasets	04
5	Write Subqueries (Scalar, Row, Table, Correlated) using EXISTS, IN, ANY, ALL; apply Set Operations (UNION, UNION ALL, INTERSECT, EXCEPT)	02
6	Create and manage Views and Materialized Views; create Indexes (B-Tree, Composite, Covering) and analyze query performance using EXPLAIN/EXPLAIN ANALYZE	04
7	Demonstrate Transaction Management (COMMIT, ROLLBACK, SAVEPOINT) and Concurrency Control (Shared/Exclusive Locks, Two-Phase Locking) with deadlock simulation	04
8	Install MongoDB; perform CRUD operations (insertOne/Many, find, updateOne/Many, deleteOne/Many) and apply Query Operators, Projection, and Sorting	04
9	Implement MongoDB Aggregation Pipeline ($match, $group, $sort, $lookup, $unwind, $project) on a sample dataset	04
10	Perform Redis operations using Key-Value data types (Strings, Lists, Sets, Hashes, Sorted Sets)	04
11	Set up Apache Cassandra; create Keyspaces and Tables; perform basic CRUD operations using CQL	02
12	Install Neo4j; create Nodes, Relationships, and Properties; write Cypher queries (MATCH, CREATE, MERGE, RETURN) for graph traversal	02
13	Demonstrate SQL Injection vulnerability and prevent it using Prepared Statements; implement Role-Based Access Control (RBAC) with user roles and permissions	03
	TOTAL	**45**

COURSE EVALUATION & ASSESSMENT RUBRICS:

Component	CE/ESE	Evaluation Methodology / Rubrics	Marks	Passing Marks
Theory	CE	Two Internal Tests of 30 marks each shall be conducted during the semester. The average of both tests will be converted to 30 marks.	30	0
	ESE	Comprehensive written examination covering all modules of the course. Duration: 2 Hours	50	25
Practical	CE	Performance of practical which should be evaluated out of 10 marks for each practical and average of the same will be converted to 30 marks.	30	0
	ESE	Quiz / Practical Performance – 25Marks External Viva Voce – 25 Marks	50	25
Case Study	CE	Case Study (CS) Evaluation Rubrics Design an ER diagram for a real-world case study	20	0

		and convert it into a normalized relational schema (apply 1NF, 2NF, 3NF, BCNF) Problem Identification & Understanding – 5 Marks Correctness of ER Diagram– 5 Marks Accuracy of Normalization– 5 Marks Presentation Quality – 5 Marks		
Assignment	CE	Assignment Evaluation Rubrics Database Crash Case Study Report, NewSQL Technology Review, NoSQL Database Comparison Matrix, SQL Injection Awareness Report, Database Selection Pitch. Evaluated on: Understanding of Concepts – 5 Marks, Justification & Reasoning – 5 Marks, Comparative Analysis Depth – 5 Marks, Report Structure & Clarity – 5 Marks.	20	0
		TOTAL MARKS	**200**	**100**

TEXT BOOK(S):

Title	Author/s	Publication
SQL, PL/SQL the Programming Language of Oracle	Ivan Bayross	BPB Publications
Database System Concept	Silberscatz, Korth, Sudarshan	McGraw Hill
MongoDB: The Definitive Guide	Shannon Bradshaw, Eoin Brazil & Kristina Chodorow	O'Reilly Media

REFERENCE BOOK(S):

Title	Author/s	Publication
Fundamentals of Database Systems	Ramez Elmsari, Shamkant B Navathe	Pearson Education
Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement	Luc Perkins, Eric Redmond & Jim Wilson	Pragmatic Bookshelf
Cassandra: The Definitive Guide	Jeff Carpenter & Eben Hewitt	O'Reilly Media

COURSE OUTCOME(S):

After the completion of the course, the students will be able to:

ICACA7020	SQL & NoSQL Database Management System
CO 1	Explain DBMS concepts, relational model, and SQL syntax to design basic database schemas.
CO 2	Apply advanced SQL features and indexing techniques to retrieve, manipulate, and optimize data efficiently.
CO 3	Design normalized relational databases using ER modelling and manage transactions with ACID properties.
CO 4	Implement and query NoSQL databases across Document, Key-Value, Column-Family, and Graph models.
CO 5	Evaluate database security, performance tuning, and emerging database technologies.

MAPPING OF CO WITH PO:

ICACA7020	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	1	0	1	0	0	1
CO 2	2	3	2	1	2	0	0	1
CO 3	3	3	2	1	1	0	1	1
CO 4	2	2	3	1	3	1	0	2
CO 5	1	2	2	2	3	1	3	2

MAPPING OF CO WITH PSO:

ICACA7020	PSO1	PSO2	PSO3
CO 1	2	0	2
CO 2	3	0	2
CO 3	3	0	2
CO 4	3	1	2
CO 5	2	3	2

LEVEL OF REVISED BLOOM'S TAXONOMY IN ASSESSMENT

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Introduction to Databases & SQL Fundamentals	1,2
2	Advanced SQL & Query Optimization	2,3,4
3	Database Design, Normalization & Transactions	3,4,5,6
4	Introduction to NoSQL Databases	1,2
5	Advanced NoSQL, Graph Databases & Cloud Databases	3,4,5,6
6	Database Security, Administration & Emerging Trends	3,4,5

P P Savani University

School of Engineering

Offered by: Department of Science and Humanities

COURSE DETAILS:

Course Code	ICASH7010
Course Name	Computational Mathematics
Prerequisite Course/s	None

TEACHING SCHEME:

Credit	Total Learning Hours	Direct Contact Hours			Indirect Contact Hours				
		Theory	Practical	Tutorial	Project	PPT	Assignment	Term Work	Self-Learning
04	120	45	0	30	0	0	30	0	15

GD = Group Discussion | CS = Case Study | FV = Field Visit | PPT = Presentation

EXAMINATION SCHEME:

Theory		Practical		Tutorial		Project		PPT		Assignment		Term Work		Self-Learning		Total
CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	
30	50	0	0	30	50	0	0	0	0	20	00	0	0	20	00	200

CE: Continuous Evaluation, ESE: End Semester Exam

OBJECTIVE OF THE COURSE:

To help learners to:

- Understand the fundamental concepts of sets, relations, and functions along with their properties and applications.
- Develop logical reasoning skills using propositional logic and truth tables.
- Apply graph theory and tree concepts to model and solve computational problems.
- Utilize numerical methods to obtain approximate solutions to mathematical problems.
- Analyze basic concepts of probability and statistics for decision-making and problem-solving.

COURSE CONTENT:

Module No.	Content	Hours	Weightage in%
1	**Set, Relation, Function & Logic-I:** Sets, Set operations, Introduction of Relations, Relations of Sets, Types of Relations, Properties of Relations, Equivalence Relation.	07	15
2	**Set, Relation, Function & Logic-II:** Partial Ordering, Hasse Diagram, GLB & LUB, Functions, Classification of functions, Types of functions, Propositional Logic, Truth Tables.	08	16
3	**Graph Theory:** Graphs and Graph Models, Graph Terminology and Types of graphs, Representing graphs and Isomorphism, Connectivity, Euler and Hamilton Paths-Circuits, Applications of weighted graphs.	08	18

4	**Trees & Spanning Trees:** Introduction to Trees, Rooted Tree, Properties of tree, Binary tree, Tree Traversal, Spanning Tree, DFS, BFS, Minimum Spanning Tree, Prim's Algorithm, Kruskal's Algorithm.	08	17
5	**Numerical Methods:** Errors and Approximation, Bisection Method, Newton-Raphson Method, Interpolation (basic idea), Trapezoidal Rule (Numerical Integration)	08	18
6	**Probability & Basic Statistics:** Experiments, Sample Space, Events, Conditional Probability, Bayes' Theorem, Random Variables, Expectation and Variance.	06	16
	TOTAL	**45**	**100%**

LIST OF PRACTICAL / TUTORIAL:

Sr. No	Practical / Tutorial	Hours
1	Problems based on Set, Relation & Function-1	02
2	Problems based on Set, Relation & Function-2	04
3	Problems based on Lattices and Hasse Diagrams	04
4	Problems based on Mathematical Logic and Proof	02
5	Problems based on Graph Theory-1	02
6	Problems based on Graph Theory-2	04
7	Problems based on Trees and Spanning Trees	04
8	Problems based on Numerical Methods-1	02
9	Problems based on Numerical Methods-2	04
10	Problems based on Probability & Basic Statistics	02
	TOTAL	**30**

SELF-LEARNING ONLINE RESOURCES:

Sr. No	Link / Description	Self-Learning Hours
1	https://youtu.be/sWsXBY19o8I?si=78RuX3xjzJnG2mgM (In mathematics, graph #theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects. A #graph in this context is made up of vertices (also called nodes or points) which are connected by edges (also called links or lines). A distinction is made between undirected graphs, where edges link two vertices symmetrically, and directed graphs, where edges link two vertices asymmetrically; see Graph (discrete mathematics) for more detailed definitions and for other variations in the types of graph that are commonly considered. Graphs are one of the prime objects of study in discrete mathematics.)	05
2	https://youtu.be/rAaUA7WTdSM?si=vABHxbE64vbeoJUd (Error Analysis, Definition and Relation Between Numerical Operators, Interpolation and Extrapolation, Newton's Binomial Expansion, Newton's Forward & Backward Interpolation Formula and Gauss Forward & Backward Interpolation Formula)	05
3	https://youtu.be/L_OLifCqxCQ?si=LTUjhVtH2-vVzzOX (This video serves as a comprehensive Statistics for Data Science course covering key concepts for industry roles. It explains Descriptive Statistics,	05

	including Mean, Median, and Mode and Data Dispersion. The instructor provides practical Python implementations using libraries like NumPy, Pandas, and SciPy for every topic. Later, the course covers Inferential Statistics and Hypothesis Testing with step-by-step code demonstrations.)	

COURSE EVALUATION & ASSESSMENT RUBRICS:

Component	CE/ESE	Evaluation Methodology / Rubrics	Marks	Passing Marks
Theory	CE	Two Internal Tests of 30 marks each shall be conducted during the semester. The average of both tests will be converted to 30 marks.	30	00
	ESE	Comprehensive written examination covering all modules of the course. Duration: 2 Hours	50	25
Tutorial	CE	Continuous Evaluation consists of the performance of tutorial which will be evaluated out of 10 marks for each tutorial and average of the same will be converted to 30 marks.	30	00
	ESE	Performance – 25Marks External Viva Voce – 25 Marks	50	25
Assignment	CE	Assignment Rubrics Conceptual Understanding – 5 Marks Problem Solving Methodology – 5 Marks Application of Theorems and Techniques – 5 Marks Clarity and Neatness – 5 Marks	20	00
Self Learning	CE	40 MCQ questions based on self-study resources and online learning | 0.5 marks per question | Total = 20 marks	20	00
		TOTAL MARKS	**200**	**100**

TEXT BOOK(S):

Title	Author/s	Publication
Discrete Mathematical Structures	Bernard Kolman, Robert C. Busby, and Sharon Cutler Ross	Pearson Education
Introductory Methods of Numerical Analysis	S. S. Sastry	PHI Learning Pvt. Ltd.
Fundamentals of Mathematical Statistics	S. C. Gupta and V. K. Kapoor	Sultan Chand & Sons

REFERENCE BOOK(S):

Title	Author/s	Publication
Discrete Mathematics and its Applications	Kenneth H. Rosen	McGraw Hill Education
Numerical Methods for Scientific and Engineering Computation	M. K. Jain, S. R. K. Iyengar, and R. K. Jain	New Age International Publishers
An Introduction to Probability Theory and Mathematical Statistics	V. K. Rohatgi and A. K. Md. Ehsanes Saleh	Wiley Student Edition

COURSE OUTCOME(S):

After the completion of the course, the students will be able to:

ICASH7010	Computational Mathematics

CO 1	Apply the concepts of sets, relations, functions, and propositional logic to formulate and solve discrete computational problems.
CO 2	Model, analyze, and solve network and structural problems using core principles and terminologies of Graph Theory.
CO 3	Implement tree structures and optimization algorithms (such as Prim's, Kruskal's, DFS, and BFS) to solve routing and traversal challenges.
CO 4	Utilize numerical methods and approximation techniques (like Bisection, Newton-Raphson, and the Trapezoidal Rule) to solve engineering equations.
CO 5	Analyze data, uncertainty, and random processes using core statistical concepts, conditional probability, and Bayes' Theorem.

MAPPING OF CO WITH PO:

ICASH7010	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	3	1	1	0	0	0	0
CO 2	3	3	2	2	1	0	0	0
CO 3	3	2	3	2	2	1	0	1
CO 4	3	3	2	2	2	0	0	0
CO 5	3	3	2	2	1	1	0	0

MAPPING OF CO WITH PSO:

ICASH7010	PSO1	PSO2	PSO3
CO 1	3	2	1
CO 2	3	3	2
CO 3	3	3	3
CO 4	2	3	2
CO 5	2	3	2

LEVEL OF REVISED BLOOM'S TAXONOMY IN ASSESSMENT

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Set, Relation, Function & Logic-I	2
2	Set, Relation, Function & Logic-II	3
3	Graph Theory	3
4	Trees & Spanning Trees	4
5	Numerical Methods	5
6	Probability & Basic Statistics	3

P P Savani University

School of Engineering

Offered by: Department of Computer Science

COURSE DETAILS:

Course Code	ICACS7010
Course Name	Information & Network Security
Prerequisite Course/s	None

TEACHING SCHEME:

Credit	Total Learning Hours	Direct Contact Hours			Indirect Contact Hours				
		Theory	Practical	Tutorial	Project	CS	Assignment	Term Work	Self-Learning
4	120	30	45	0	30	0	0	0	15

GD = Group Discussion | CS = Case Study | FV = Field Visit | PPT = Presentation

EXAMINATION SCHEME:

Theory		Practical		Tutorial		Project		CS		Assignment		Term Work		Self-Learning		Total
CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	
30	50	30	50	0	0	20	00	0	0	0	0	0	0	20	00	200

CE: Continuous Evaluation, ESE: End Semester Exam

OBJECTIVE OF THE COURSE:

To help learners to:

- Understand the core concepts of information security, classical cryptography, and modern mathematical security principles.
- Evaluate and implement standard symmetric and asymmetric cryptographic algorithms.
- Explore network layer security protocols, web security mechanisms, and firewall architectures.
- Develop awareness about emerging cyber threats, malware, vulnerabilities, and incident handling.
- Formulate robust organizational security policies, compliance blueprints, and practice ethical cyber methodologies.

COURSE CONTENT:

Module No.	Content	Hours	Weightage in%
1	**Introduction & Symmetric Cryptography:** Security attacks, services, and mechanisms; Classical encryption techniques (Substitution & Transposition); Block cipher principles; Data Encryption Standard (AES) & Advanced Encryption Standard (AES); Block cipher modes of operation.	06	20
2	**Asymmetric Cryptography & Key Management:** Principles of Public-Key Cryptosystems; RSA Algorithm; Diffie-Hellman Key Exchange; Elliptic Curve Cryptography (ECC); Message Authentication Codes (MAC); Digital Signatures and Public Key Infrastructure (PKI).	06	20
3	**Network & Web Security Protocols:**	06	20

	Transport Level Security: SSL, TLS, and HTTPS; Electronic Mail Security: PGP, S/MIME; IP Security (IPSec): Architecture, Authentication Header (AH), and Encapsulating Security Payload (ESP).		
4	**System Security & Wireless Defense:** Intruders and Intrusion Detection Systems (IDS/IPS); Malicious Software: Viruses, Worms, Trojans, and Ransomware; Firewall Design Principles, Types, and Configurations; Wireless network security (WPA2, WPA3).	04	14
5	**Hands-on Security Controls & Case Studies:** Penetration testing basics using Kali Linux; Wireshark packet filtering and vulnerability assessment; Analyzing real-world cyber breaches (e.g., Ransomware attacks, phishing simulations); Auditing enterprise network diagrams.	04	14
6	**Security Policies, Laws & Cyber Governance:** Information Security Management Systems (ISMS) & ISO 27001 standard; Legal and ethical aspects: IT Act 2000/2008, GDPR, Data Privacy compliance; Cyber risk assessment and incident response frameworks.	04	12
	TOTAL	**30**	**100%**

LIST OF PRACTICAL / TUTORIAL:

Sr. No	Practical / Tutorial	Hours
1	Implementation of Classical Substitution Ciphers (Caesar, Playfair, and Vigenere Ciphers) using Python or Java to understand character shifting and matrix key blocks.	04
2	Implementation of Polyalphabetic Transposition Techniques (Rail Fence and Row-Column Transposition Ciphers).	04
3	Practical implementation of standard Symmetric Key Cryptography using built-in libraries (DES and AES algorithms) with different padding modes.	04
4	Implementing Asymmetric Key Cryptography using the RSA algorithm (Key Generation, Encryption, and Decryption).	04
5	Simulating Key Exchange over an insecure channel using the Diffie-Hellman Key Exchange protocol.	03
6	Generation and comparison of Message Digests using Hash functions (MD5, SHA-1, and SHA-256) for verifying file integrity.	03
7	Implementation of Digital Signature Standard (DSS/DSA) for non-repudiation and message authentication.	03
8	Network Packet Sniffing and deep-dive protocol analysis using Wireshark (capturing clear-text HTTP/FTP passwords vs. analyzing encrypted TLS/SSH packets).	04
9	Performing network reconnaissance, port scanning, and OS fingerprinting using Nmap and vulnerability scanning using Nessus.	03
10	Installation, configuration, and writing custom signature alert rules using Snort IDS/IPS on a virtual private network.	03
11	Deploying a local Web Server and configuring secure communication endpoints utilizing OpenSSL to generate self-signed X.509 Certificates and enforcing HTTPS.	03
12	Demonstrating email encryption and secure communications using GnuPG (GNU Privacy Guard) / PGP protocols.	02
13	Mini Case Study Presentation: Dissecting an enterprise data breach or ransomware scenario, analyzing structural security failures, and formulating a	02

	corrective defense blueprint.	
14	Mini Project Execution: Building an end-to-end encrypted local chat application or a secure cryptographic file storage system utilizing hybrid encryption.	03
	TOTAL	**45**

SELF-LEARNING ONLINE RESOURCES:

Sr. No	Link / Description	Self-Learning Hours
1	https://nptel.ac.in – Cryptography and Network Security, Introduction to Cyber Security, Network Security baseline modules.	03
2	https://owasp.org – OWASP Top 10 Vulnerabilities Guide, API Security Project, and Secure Coding Checklists.	03
3	https://learn.microsoft.com – SC-900: Microsoft Security, Compliance, and Identity Fundamentals, Azure Network Security practices.	03
4	https://attack.mitre.org – Threat Matrix for Enterprise, Adversary Tactics & Techniques repository, and Case Studies.	03
5	https://www.coursera.org – IBM Cybersecurity Analyst Professional Certificate, Stanford Cryptography Specialization, and SWAYAM NPTEL MOOCs.	03

COURSE EVALUATION & ASSESSMENT RUBRICS:

Component	CE/ESE	Evaluation Methodology / Rubrics	Marks	Passing Marks
Theory	CE	Two Internal Tests of 30 marks each shall be conducted during the semester. The average of both tests will be converted to 30 marks.	30	0
	ESE	Comprehensive written examination covering all modules of the course. Duration: 2 Hours	50	25
Practical	CE	Performance of practical which should be evaluated out of 10 marks for each practical and average of the same will be converted to 30 marks.	30	0
	ESE	Quiz / Practical Performance – 25Marks External Viva Voce – 25 Marks	50	25
Project	CE	Project Evaluation Rubrics: Problem Statement & Requirement Analysis – 5 Marks Design, Architecture & Tool Usage – 5 Marks Implementation & Functionality – 5 Marks Project Demonstration & Viva-Voce – 5 Marks	20	0
Self Learning	CE	40 MCQ questions based on self-study resources and online learning | 0.5 marks per question | Total = 20 marks	20	0
		TOTAL MARKS	**200**	**100**

TEXT BOOK(S):

Title	Author/s	Publication
Cryptography and Network Security: Principles and Practice	William Stallings	Pearson Education

REFERENCE BOOK(S):

Title	Author/s	Publication
Network Security Essentials: Applications and Standards	William Stallings	Pearson
Principles of Information Security	Michael E. Whitman & Herbert J. Mattord	Cengage Learning

COURSE OUTCOME(S):

After the completion of the course, the students will be able to:

ICACS7010	Information & Network Security
CO 1	Explain core network security vulnerabilities, threat vectors, classical substitution-transposition ciphers, and fundamental security design models.
CO 2	Use modern public-key cryptography infrastructures, standard digital signatures, and automated vulnerability scanning tools for network mapping.
CO 3	Apply infrastructure-level defences such as Firewalls, Intrusion Detection Systems (IDS), and cryptographic communication tunnels (SSL/TLS, IPSec) to insulate network ecosystems.
CO 4	Evaluate legal cyber compliances, internal IT security architectures, data privacy frameworks, and the aftermath of historical global threat operations.
CO 5	Design hardened application-level workflows using defence-in-depth secure code designs, data backup strategy lifecycles, and risk mitigation methodologies.

MAPPING OF CO WITH PO:

ICACS7010	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	1	0	2	0	0	1
CO 2	2	2	2	1	3	2	0	1
CO 3	3	3	3	2	3	2	1	2
CO 4	0	1	1	1	2	1	3	2
CO 5	2	3	3	2	3	3	2	3

MAPPING OF CO WITH PSO:

ICACS7010	PSO1	PSO2	PSO3
CO 1	2	1	1
CO 2	3	2	1
CO 3	3	3	2
CO 4	1	2	3
CO 5	3	3	3

LEVEL OF REVISED BLOOM'S TAXONOMY IN ASSESSMENT

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Introduction & Symmetric Cryptography	1,2
2	Asymmetric Cryptography & Key Management	3
3	Network & Web Security Protocols	3

4	System Security & Wireless Defense	5
5	Hands-on Security Controls & Case Studies	4
6	Security Policies, Laws & Cyber Governance	6

P P Savani University

School of Engineering

Offered by: Department of Computer Science

COURSE DETAILS:

Course Code	ICACS7030
Course Name	Web Development with Python
Prerequisite Course/s	None

TEACHING SCHEME:

Credit	Total Learning Hours	Direct Contact Hours			Indirect Contact Hours				
		Theory	Practical	Tutorial	Project	CS	Assignment	Term Work	Self-Learning
04	120	30	45	0	30	0	0	0	15

GD = Group Discussion | CS = Case Study | FV = Field Visit | PPT = Presentation

EXAMINATION SCHEME:

Theory		Practical		Tutorial		Project		CS		Assignment		Term Work		Self-Learning		Total
CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	
30	50	30	50	0	0	20	00	0	0	0	0	0	0	20	00	200

CE: Continuous Evaluation, ESE: End Semester Exam

OBJECTIVE OF THE COURSE:

To help learners to:

- Understand modern web technologies and web application architecture.
- Design responsive and interactive web interfaces using HTML5, CSS3, Bootstrap and JavaScript.
- Develop dynamic web applications using Python and Django framework.
- Integrate databases and implement authentication, authorization and security mechanisms.
- Develop RESTful APIs and deploy web applications on cloud platforms.

COURSE CONTENT:

Module No.	Content	Hours	Weightage in%
1	**Web Fundamentals and Front-End Development: :** Internet and WWW, Client-Server Architecture, Web Browsers and Web Servers, Static and Dynamic Websites, HTML5 Fundamentals, Forms, Tables, Multimedia Elements, CSS3 Styling, Responsive Web Design, Bootstrap Framework, Introduction to JavaScript, DOM Manipulation and Events.	04	13
2	**Introduction to Django Framework:** Python for Web Development, Django Architecture (MVT), Installation and Configuration of Django, Creating Projects and Applications, Django Project Structure, URL Routing, Views, Templates, Static Files and Media Files Management.	06	25
3	**Database Integration using Django ORM: :** Introduction to Relational Databases, MySQL Integration, Models, Migrations, CRUD Operations, QuerySets, Model Relationships, Django Admin Panel, Data Validation and	06	15

	Model Forms.		
4	**Templates, Forms and User Interaction: :** Template Language, Template Inheritance, Dynamic Content Rendering, Form Handling, Form Validation, Bootstrap Integration, User Input Processing, Error Messages and User Experience Design.	04	15
5	**Authentication and Security: :** User Registration, Login and Logout, Password Management, Authorization and Permissions, Sessions and Cookies, CSRF Protection, Security Best Practices, Error Handling and Logging.	04	16
6	**REST API Development: :** REST Architecture, Introduction to Django REST Framework, Serializers, API Views, CRUD APIs, Authentication Mechanisms, JSON Processing, API Testing using Postman, AJAX and Fetch API Integration.	06	16
	TOTAL	**30**	**100%**

LIST OF PRACTICAL / TUTORIAL:

Sr. No	Practical / Tutorial	Hours
1	Installation and configuration of Python, Django and development environment.	03
2	Design responsive web pages using HTML5 and CSS3.	03
3	Develop web pages using Bootstrap components and layouts.	03
4	Implement JavaScript programs for DOM manipulation and form validation.	03
5	Create Django project and application structure.	03
6	Implement URL routing, views and template rendering in Django.	03
7	Create database models and perform migrations using Django ORM.	03
8	Develop CRUD operations using Django ORM and MySQL database.	03
9	Configure and customize Django Admin Panel.	03
10	Develop form-based applications with validation and error handling.	06
11	Implement user registration, login and authentication system.	06
12	Develop and test a REST API using Django REST Framework — implement serializers and CRUD API views, secure them with an authentication mechanism, and validate functionality using Postman with AJAX/Fetch integration on the front end.	06
	TOTAL	**45**

SELF-LEARNING ONLINE RESOURCES:

Sr. No	Link / Description	Self-Learning Hours
1	NPTEL - Python for Data Science and Web Development Courses – https://nptel.ac.in	03
2	Django Official Documentation - Comprehensive guide for Django Framework, Models, Views, Templates, Forms, ORM and Deployment – https://docs.djangoproject.com	03
3	W3Schools - HTML5, CSS3, Bootstrap, JavaScript and Django Tutorials – https://www.w3schools.com	03
4	freeCodeCamp (YouTube & Website) - Full Stack Web Development with Python and Django Projects – https://www.freecodecamp.org	03
5	PythonAnywhere & Render Documentation - Cloud Hosting and Deployment	03

	of Django Applications – https://www.pythonanywhere.com and https://render.com	

COURSE EVALUATION & ASSESSMENT RUBRICS:

Component	CE/ESE	Evaluation Methodology / Rubrics	Marks	Passing Marks
Theory	CE	Two Internal Tests of 30 marks each shall be conducted during the semester. The average of both tests will be converted to 30 marks.	30	00
	ESE	Comprehensive written examination covering all modules of the course. Duration: 2 Hours	50	25
Practical	CE	Performance of practical which should be evaluated out of 10 marks for each practical and average of the same will be converted to 30 marks.	30	00
	ESE	Quiz / Practical Performance – 25Marks External Viva Voce – 25 Marks	50	25
Project	CE	Backend Architecture & Clean Code Structure – 5 Marks Database Design & Integrity (Models/ORM) – 5 Marks Functionality, API Routing & Validation – 5 Marks Security Implementation (Auth/Session/CSRF) – 5 Marks	20	00
Self-Learning	CE	40 MCQ questions based on self-study resources and online learning | 0.5 marks per question | Total = 20 marks	20	00
		TOTAL MARKS	**200**	**100**

TEXT BOOK(S):

Title	Author/s	Publication
Django for Beginners	William S. Vincent	WelcomeToCode Publications
Python Crash Course	Eric Matthes	No Starch Press

REFERENCE BOOK(S):

Title	Author/s	Publication
Django 5 By Example	Antonio Melé	Packt Publishing
Web Development with Django	Ben Shaw	Packt Publishing
REST APIs with Django	William S. Vincent	WelcomeToCode Publications
Django for Professionals	William S. Vincent	WelcomeToCode Publications

COURSE OUTCOME(S):

After the completion of the course, the students will be able to:

ICACS7030	Web Development with Python
CO 1	Explain web technologies, web architecture, front-end development concepts and Django framework fundamentals.
CO 2	Develop responsive and interactive web applications using HTML, CSS, Bootstrap,

	JavaScript and Django.
CO 3	Design and implement database-driven web applications using Django ORM and MySQL database.
CO 4	Implement authentication, authorization, security mechanisms and RESTful APIs in web applications.
CO 5	Deploy web applications, prepare technical documentation and demonstrate complete web-based solutions.

MAPPING OF CO WITH PO:

ICACS7030	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	0	0	2	0	0	0
CO 2	3	3	2	0	3	0	0	0
CO 3	3	3	3	2	3	0	0	0
CO 4	2	3	3	2	3	1	0	2
CO 5	2	3	3	2	3	1	1	2

MAPPING OF CO WITH PSO:

ICACS7030	PSO1	PSO2	PSO3
CO 1	2	2	1
CO 2	2	3	2
CO 3	2	3	2
CO 4	3	3	2
CO 5	3	3	3

LEVEL OF REVISED BLOOM'S TAXONOMY IN ASSESSMENT

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Web Fundamentals and Front-End Development	1,2
2	Introduction to Django Framework	2,3
3	Database Integration using Django ORM	3,4
4	Templates, Forms and User Interaction	3,4
5	Authentication and Security	3,4
6	REST API Development	4,5,6

P P Savani University

School of Engineering

Offered by: Department of Computer Application

COURSE DETAILS:

Course Code	ICACA7030
Course Name	Java Web Application Development
Prerequisite Course/s	None

TEACHING SCHEME:

Credit	Total Learning Hours	Direct Contact Hours			Indirect Contact Hours				
		Theory	Practical	Tutorial	Project	CS	Assignment	Term Work	Self-Learning
04	120	30	45	0	15	0	30	0	0

GD = Group Discussion | CS = Case Study | FV = Field Visit | PPT = Presentation

EXAMINATION SCHEME:

Theory		Practical		Tutorial		Project		CS		Assignment		Term Work		Self-Learning		Total
CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	
30	50	30	50	0	0	20	00	0	0	20	00	0	0	0	0	200

CE: Continuous Evaluation, ESE: End Semester Exam

OBJECTIVE OF THE COURSE:

To help learners to:

- Refresh core Object-Oriented Java syntax and bridge baseline frontend markup with server architectures.
- Manage robust application data pipelines securely using advanced JDBC driver models and design patterns.
- Develop enterprise server-side solutions by mastering the lifecycle, state tracking, and request handling of Java Servlets.
- Build clean, decoupled view models utilizing Java Server Pages (JSP) and standard runtime tag expressions.
- Implement industry-standard enterprise frameworks via Spring Boot auto-configurations to construct scalable RESTful APIs.

COURSE CONTENT:

Module No.	Content	Hours	Weightage in%
1	**Bridge Module: Core Java Refresher & Web Basics:** Quick review of Object-Oriented Programming (OOP) in Java: Classes, Objects, Inheritance, Interface, and Packages. Exception Handling mechanics (try-catch-finally, throw, throws). Foundations of Web Architectures (1-tier, 2-tier, 3-tier). Review of HTML5 Form structures, input components, and HTTP request actions (GET vs. POST methods).	05	15
2	**Database Persistence & Servlets Core:** Introduction to Java Database Connectivity (JDBC): Driver	06	20

	configurations, DriverManager, executing queries via Statement and PreparedStatement, handling ResultSet. Introduction to Java Servlets: Lifecycle, architecture, configuration via @WebServlet annotations. Processing dynamic client data using doGet and doPost methods.		
3	**Session Management & Middleware Filtering:** The necessity of state management over stateless HTTP protocols. Implementing Session Management techniques: Cookies, Hidden Form Fields, URL Rewriting, and HttpSession API. Servlet collaboration patterns: Request forwarding vs. page redirection (RequestDispatcher). Intercepting web traffic using Servlet Filters and Listeners.	04	15
4	**Dynamic Web Views (JSP) & Spring Foundations:** Introduction to Java Server Pages (JSP) and its compilation lifecycle. Dynamic scripting tags: Declarations, Expressions, and Scriptlets. JSP Directives and Standard Actions. Decoupling code using Expression Language (EL) and JSP Standard Tag Library (JSTL) core tags. Limitations of legacy designs and introduction to the Spring Framework Ecosystem.	06	20
5	**Rapid Enterprise Automation: Spring Boot:** Core architecture principles: Inversion of Control (IoC) and Dependency Injection (DI) models via @Autowired annotations. Transitioning to Spring Boot: Starters, application configurations, and structural setup via Spring Initializr. Implementing the modern MVC architectural design using @Controller and handling views.	04	15
6	**Decoupled System Architecture & RESTful APIs:** Introduction to Service-Oriented Architecture (SOA), Microservices, and RESTful guidelines. Building production-tier endpoints using Spring Boot annotations: @RestController, @GetMapping, @PostMapping, @PutMapping, @DeleteMapping, @PathVariable, and @RequestBody. Formatting clean JSON response payloads and managing exceptional status responses.	05	15
	TOTAL	**30**	**100%**

LIST OF PRACTICAL / TUTORIAL:

Sr. No	Practical / Tutorial	Hours
1	Write a core Java program demonstrating class inheritance, custom interface structures, and multi-catch robust exception handling routines.	04
2	Create a static HTML profile application containing complex fields (input types, dropdowns, buttons) that directs submissions toward server targets.	04
3	Connect a Java program to a relational database using JDBC and execute fundamental CRUD statements securely via a reusable.	04
4	Develop a baseline Java Servlet that captures basic parameters submitted from an HTML interface and generates an on-the-fly dynamic greeting layout.	04
5	Build a secure multi-page login authentication flow that tracks user sessions and restricts malicious navigation using explicit HttpSession state handling.	04
6	Create a robust administrative tracking module that utilizes custom Servlet Filters to intercept incoming requests and log activity metrics to the console.	04
7	Convert a hardcoded dynamic text generation dashboard entirely into clean	05

	presentation architectures leveraging JSP layout attributes and core JSTL tag rules.	
8	Configure a baseline Spring engine workspace to demonstrate Dependency Injection behaviors utilizing modular runtime object initialization components.	05
9	Deploy a standalone Spring Boot application implementing a decoupled layout that delivers and handles structured JSON payloads through specialized.	06
10	Mini Project Execution: Synthesize all components by developing a lightweight business catalog management module featuring database integration, controller layouts, and input processing.	05
	TOTAL	**45**

COURSE EVALUATION & ASSESSMENT RUBRICS:

Component	CE/ESE	Evaluation Methodology / Rubrics	Marks	Passing Marks
Theory	CE	Two Internal Tests of 30 marks each shall be conducted during the semester. The average of both tests will be converted to 30 marks.	30	00
	ESE	Comprehensive written examination covering all modules of the course. Duration: 2 Hours	50	25
Practical	CE	Performance of practical which should be evaluated out of 10 marks for each practical and average of the same will be converted to 30 marks.	30	00
	ESE	Quiz / Practical Performance – 25Marks External Viva Voce – 25 Marks	50	25
Project	CE	Project Evaluation Rubrics Backend Architecture & Clean Data Layout – 5 Marks API Standard Routing Validation – 5 Marks Session Security Implementation – 5 Marks Database Integrity Handling – 5 Marks	20	00
Assignment	CE	Continuous Evaluation based on technical case assignments, framework structural configurations, and microservice analysis reports.	20	00
		TOTAL MARKS	**200**	**100**

TEXT BOOK(S):

Title	Author/s	Publication
Java Server Programming Java EE 7 (J2EE 1.7) Black Book	Dreamtech Software Team	Dreamtech Press.
Spring Boot in Action	Craig Walls	Manning Publications

REFERENCE BOOK(S):

Title	Author/s	Publication
Java: The Complete Reference	Herbert Schildt	McGraw Hill
Spring Microservices in Action	John Carnell and Illary Huaylupo Sánchez	Manning Publications
Designing RESTful Web APIs	Mark Masse	O'Reilly Media

COURSE OUTCOME(S):

After the completion of the course, the students will be able to:

ICACA7030	Java Web Application Development
CO 1	Configure reliable database connections using the JDBC framework.
CO 2	Develop dynamic server-side applications using Java Servlets.
CO 3	Build structural Model-View-Controller presentation layers using JSP.
CO 4	Implement dependency injection principles within the Spring engine.
CO 5	Design and deploy scalable, decoupled RESTful APIs using Spring Boot.

MAPPING OF CO WITH PO:

ICACA7030	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	2	0	1	0	1	2
CO 2	3	2	3	0	1	0	0	2
CO 3	2	1	3	0	2	0	0	2
CO 4	3	2	3	1	3	1	0	3
CO 5	3	3	3	2	3	2	1	3

MAPPING OF CO WITH PSO:

ICACA7030	PSO1	PSO2	PSO3
CO 1	3	0	2
CO 2	3	0	2
CO 3	3	0	2
CO 4	3	1	3
CO 5	3	2	3

LEVEL OF REVISED BLOOM'S TAXONOMY IN ASSESSMENT

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Bridge Module: Core Java Refresher & Web Basics	1,2,3
2	Database Persistence & Servlets Core	2,3,4
3	Session Management & Middleware Filtering	3,4
4	Dynamic Web Views (JSP) & Spring Foundations	3,4
5	Rapid Enterprise Automation: Spring Boot	3,4,6
6	Building RESTful Web Services & APIs	3,4,5,6

P P Savani University

School of Engineering

Offered by: Department of Computer Engineering

COURSE DETAILS:

Course Code	ICACA7040
Course Name	Software Quality & Assurance
Prerequisite Course/s	None

TEACHING SCHEME:

Credit	Total Learning Hours	Direct Contact Hours			Indirect Contact Hours				
		Theory	Practical	Tutorial	Project	PPT	Assignment	Term Work	Self-Learning
04	120	45	30	0	0	15	30	0	0

GD = Group Discussion | CS = Case Study | FV = Field Visit | PPT = Presentation

EXAMINATION SCHEME:

Theory		Practical		Tutorial		Project		PPT		Assignment		Term Work		Self-Learning		Total
CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	
30	50	30	50	0	0	0	0	20	00	20	00	0	0	0	0	200

CE: Continuous Evaluation, ESE: End Semester Exam

OBJECTIVE OF THE COURSE:

To help learners to:

- Understand the concepts, principles, and standards of software quality assurance.
- Apply software verification, validation, and testing techniques throughout the software development life cycle.
- Design effective test cases using black-box and white-box testing techniques.
- Utilize automated testing tools, quality metrics, and defect management practices.
- Implement software quality standards, reviews, audits, and process improvement models in software projects.

COURSE CONTENT:

Module No.	Content	Hours	Weightage in%
1	**Introduction to Software Quality Assurance:** Software Quality Concepts, Quality Factors, Software Quality Attributes, Software Quality Assurance (SQA), Software Quality Control (SQC), Verification vs Validation, Software Development Life Cycle and Quality, Cost of Quality, Software Quality Standards (ISO 9001, ISO/IEC 25010), Capability Maturity Model (CMMI).	08	18
2	**Software Reviews and Testing Fundamentals:** Technical Reviews, Walkthroughs, Inspections, Audits, Static Testing, Dynamic Testing, Software Testing Principles, Test Process, Levels of Testing (Unit, Integration, System, Acceptance), Regression Testing, Smoke Testing, Alpha and	08	18

	Beta Testing.		
3	**Black Box Testing Techniques:** Test Case Design, Equivalence Class Partitioning, Boundary Value Analysis, Decision Table Testing, Cause-Effect Graphing, State Transition Testing, Use Case Testing, Error Guessing, Test Data Preparation.	08	18
4	**White Box Testing Techniques:** Basis Path Testing, Control Flow Graph, Cyclomatic Complexity, Statement Coverage, Branch Coverage, Condition Coverage, Path Coverage, Loop Testing, Data Flow Testing, Mutation Testing.	08	18
5	**Software Quality Metrics and Test Management:** Software Metrics, Product Metrics, Process Metrics, Reliability Metrics, Defect Life Cycle, Defect Tracking, Test Planning, Test Strategy, Test Documentation (IEEE Standards), Risk-Based Testing, Configuration Management, Test Management Tools.	08	18
6	**Automation Testing and Software Quality Standards:** Introduction to Test Automation, Selenium IDE/WebDriver Basics, JUnit/TestNG Overview, Continuous Testing, CI/CD Concepts, Quality Models (McCall, Boehm, ISO 25010), Six Sigma Basics, Software Process Improvement, Industrial Case Studies.	05	10
	TOTAL	**45**	**100%**

LIST OF PRACTICAL / TUTORIAL:

Sr. No	Practical / Tutorial	Hours
1	Preparation of Software Requirement Specification (SRS) for a case study	02
2	Identification of Functional and Non-functional Requirements	02
3	Preparation of Test Plan and Test Strategy	02
4	Design of Test Cases using Equivalence Partitioning and Boundary Value Analysis	02
5	Decision Table and Cause-Effect Graph Testing	02
6	State Transition Testing and Use Case Testing	02
7	Control Flow Graph Construction and Cyclomatic Complexity Calculation	02
8	White Box Testing using Statement, Branch and Path Coverage	02
9	Defect Reporting and Bug Tracking using Bug Tracking Tool (Bugzilla/JIRA Demo)	02
10	Preparation of Test Summary Report and Software Metrics Calculation	02
11	Automation Testing using Selenium IDE/WebDriver	02
12	Mini Project: Complete Testing of a Web/Desktop Application including Documentation	08
	TOTAL	**30**

COURSE EVALUATION & ASSESSMENT RUBRICS:

Component	CE/ESE	Evaluation Methodology / Rubrics	Marks	Passing Marks
Theory	CE	Two Internal Tests of 30 marks each shall be conducted during the semester. The average of both tests will be converted to 30 marks.	30	00
	ESE	Comprehensive written examination covering all modules of the course. Duration: 2 Hours	50	25

Practical	CE	Performance of practical which should be evaluated out of 10 marks for each practical and average of the same will be converted to 30 marks.	30	00
	ESE	Quiz / Practical Performance – 25 Marks External Viva Voce – 25 Marks	50	25
PPT	CE	Presentation Evaluation Rubrics: Problem Identification & Requirement Analysis – 05 Marks Test Planning, Quality Analysis & Solution Approach – 05 Marks Technical Documentation & Report Preparation – 05 Marks Presentation, Demonstration & Viva Voce – 05 Marks	20	00
Assignment	CE	40 MCQ questions based on software quality assurance concepts, testing methodologies, software quality standards, testing tools, and self-learning resources. 0.5 marks per question (Total = 20 Marks).	20	00
		TOTAL MARKS	**200**	**100**

TEXT BOOK(S):

Title	Author/s	Publication
Software Quality Assurance: Principles and Practice	Nina S. Godbole	Narosa Publishing House
Software Testing: Principles and Practices	Srinivasan Desikan & Gopalaswamy Ramesh	Pearson Education

REFERENCE BOOK(S):

Title	Author/s	Publication
Software Engineering: A Practitioner's Approach	Roger S. Pressman & Bruce Maxim	McGraw-Hill
Effective Methods for Software Testing	William E. Perry	Wiley
Foundations of Software Testing	Dorothy Graham, Erik van Veenendaal, Isabel Evans & Rex Black	Cengage Learning
Software Testing	Ron Patton	Pearson Education

COURSE OUTCOME(S):

After the completion of the course, the students will be able to:

ICACA7040	Software Quality & Assurance
CO 1	Explain software quality concepts, quality models, software development life cycle, and software quality assurance processes.
CO 2	Apply software review techniques and testing methodologies at different levels of software development.
CO 3	Design effective test cases using black-box and white-box testing techniques for software validation.
CO 4	Analyze software quality using metrics, defect management, test planning, and documentation practices.
CO 5	Utilize automation testing tools and software quality standards to improve software reliability and process quality.

MAPPING OF CO WITH PO:

ICACA7040	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	1	0	1	0	1	1
CO 2	2	3	2	1	1	1	1	1
CO 3	2	3	3	2	1	2	1	1
CO 4	2	3	2	3	2	2	2	1
CO 5	2	2	3	2	3	3	2	2

MAPPING OF CO WITH PSO:

ICACA7040	PSO1	PSO2	PSO3
CO 1	1	2	1
CO 2	2	3	1
CO 3	2	3	2
CO 4	3	2	2
CO 5	3	3	3

LEVEL OF REVISED BLOOM'S TAXONOMY IN ASSESSMENT

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Introduction to Software Quality Assurance	1,2
2	Software Reviews and Testing Fundamentals	2,3
3	Black Box Testing Techniques	3,4
4	White Box Testing Techniques	3,4
5	Software Quality Metrics and Test Management	4,5
6	Automation Testing and Software Quality Standards	5,6

P P Savani University

School of Engineering

Offered by: Department of Computer Science

COURSE DETAILS:

Course Code	ICACS7040
Course Name	Prompt Engineering & Generative AI
Prerequisite Course/s	None

TEACHING SCHEME:

Credit	Total Learning Hours	Direct Contact Hours			Indirect Contact Hours				
		Theory	Practical	Tutorial	Project	CS	Assignment	Term Work	Self-Learning
04	120	30	45	0	0	0	30	0	15

GD = Group Discussion | CS = Case Study | FV = Field Visit | PPT = Presentation

EXAMINATION SCHEME:

Theory		Practical		Tutorial		Project		CS		Assignment		Term Work		Self-Learning		Total
CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	
30	50	30	50	0	0	0	0	0	0	20	00	0	0	20	00	200

CE: Continuous Evaluation, ESE: End Semester Exam

OBJECTIVE OF THE COURSE:

To help learners to:

- Understand the fundamental concepts, architectures, and evolution of Generative AI and Large Language Models.
- Develop proficiency in designing, structuring, and refining prompts for diverse generative AI tools and tasks.
- Explore text, image, audio, video, and code generation tools and their underlying prompting techniques.
- Apply prompt engineering strategies to solve real-world academic, professional, and creative problems.
- Build awareness of ethical, responsible, and safe use of generative AI tools in academic and professional contexts.

COURSE CONTENT:

Module No.	Content	Hours	Weightage in%
1	**Introduction to Generative AI & Prompt Engineering:** What is Generative AI; Evolution from rule-based AI to Large Language Models; How LLMs work (tokens, embeddings, transformers - conceptual overview); Overview of leading platforms (ChatGPT, Gemini, Claude); Introduction to Prompt Engineering; Why prompts matter; Real-world use cases across industries.	06	20
2	**Prompt Design Principles & Techniques:** Anatomy of a good prompt (context, instruction, persona,	06	20

	format, constraints); Zero-shot, one-shot, and few-shot prompting; Chain-of-Thought prompting; Role-based and persona prompting; Iterative refinement and prompt debugging; Common pitfalls and prompt anti-patterns.		
3	**Prompting for Text, Code & Productivity Tools:** Prompting for writing, summarization, and research with ChatGPT/Gemini/Claude; Prompting for code generation, debugging, and explanation with AI coding assistants; Prompts for presentations, reports, and emails; AI productivity tools (Notion AI, Microsoft Copilot); Prompt templates and reusable prompt libraries	06	20
4	**Multimodal Generative AI: Image, Audio & Video:** AI image generation tools (DALL-E, MidJourney, Adobe Firefly, Stable Diffusion); Prompting techniques for image generation (style, composition, lighting, negative prompts); AI audio and voice generation tools; AI video generation tools; Use cases in design, marketing, and content creation.	04	14
5	**Advanced Prompting & AI Agent Workflows:** Retrieval-Augmented Generation (RAG) basics; Introduction to AI agents and autonomous workflows; Prompt chaining and multi-step reasoning; Combining tools (AI + browsing + code execution); Building simple automations using AI tools; Evaluating and comparing AI-generated outputs.	04	14
6	**Ethics, Limitations & Future of Generative AI:** Hallucinations and factual reliability; Bias, fairness, and data privacy in generative models; Copyright, plagiarism, and academic integrity concerns; Deepfakes and misuse risks; Responsible and ethical prompting practices; Emerging trends in Generative AI and career opportunities.	04	12
	TOTAL	**30**	**100%**

LIST OF PRACTICAL / TUTORIAL:

Sr. No	Practical / Tutorial	Hours
1	Comparative evaluation of LLM architectures (GPT, Gemini, Claude, open-source models) on tokenization, context window, and reasoning benchmarks	03
2	Designing zero-shot, one-shot, and few-shot prompts with controlled variable testing to empirically compare output quality and consistency	03
3	Implementing Chain-of-Thought, Tree-of-Thought, and Self-Consistency prompting strategies on multi-step reasoning and mathematical problems	03
4	Constructing and stress-testing role-based and persona-conditioned prompts across adversarial and edge-case scenarios	03
5	Building a reusable, parameterized prompt template library with version control and systematic prompt debugging logs	03
6	AI-assisted technical report writing, summarization, and citation-aware research synthesis using ChatGPT, Gemini, and Claude	03
7	Prompt engineering for AI coding assistants: code generation, refactoring, vulnerability detection, and automated test-case generation	03
8	Designing structured prompts for AI productivity tools (Notion AI, Microsoft Copilot) integrated into a real document/workflow pipeline	01
9	Advanced image-generation prompt engineering: style transfer, compositional control, negative prompting, and parameter tuning (DALL-E, MidJourney, Stable Diffusion)	03

10	Multimodal prompting for AI audio/voice and video generation tools with iterative refinement and quantitative output evaluation	02
11	Implementing a basic Retrieval-Augmented Generation (RAG) pipeline: embeddings, vector search, and grounded prompt construction	04
12	Designing and orchestrating multi-step prompt chains and a simple autonomous AI agent workflow combining reasoning, tool-use, and code execution	04
13	Building an LLM output evaluation framework: defining metrics (faithfulness, relevance, hallucination rate) and benchmarking multiple models on a common task	03
14	Critical analysis of bias, hallucination, and data-privacy risks in generative AI outputs through structured red-teaming exercises	03
15	Capstone Mini Project: Designing, implementing, and presenting a multi-step, prompt-driven Generative AI solution to a real-world academic or industry problem	04
	TOTAL	**45**

SELF-LEARNING ONLINE RESOURCES:

Sr. No	Link / Description	Self-Learning Hours
1	https://www.youtube.com/watch?v=YhRfgYH_AoU – Prompt Engineering Full Course: Beginner to Pro	05
2	https://www.youtube.com/watch?v=zAHvJAEyhZ0 - Complete Prompt Engineering Course for Beginners https://www.youtube.com/watch?v=QnZYrD5ia7c - Prompt Engineering for Beginners: Full Workshop	05
3	https://www.youtube.com/watch?v=qJ8f90aU8qw - Master Prompt Engineering: 7 Essential Tips	05

COURSE EVALUATION & ASSESSMENT RUBRICS:

Component	CE/ESE	Evaluation Methodology / Rubrics	Marks	Passing Marks
Theory	CE	Two Internal Tests of 30 marks each shall be conducted during the semester. The average of both tests will be converted to 30 marks.	30	00
	ESE	Comprehensive written examination covering all modules of the course. Duration: 2 Hours	50	25
Practical	CE	Performance of practical which should be evaluated out of 10 marks for each practical and average of the same will be converted to 30 marks.	30	00
	ESE	Quiz / Practical Performance – 25Marks External Viva Voce – 25 Marks	50	25
Assignment	CE	Prompting Portfolio on real world problems- 15 Marks Regular Completion- 05 Marks	20	00
Self Learning	CE	40 MCQ questions based on self-study resources and online learning | 0.5 marks per question | Total = 20 marks	20	00
		TOTAL MARKS	**200**	**100**

TEXT BOOK(S):

Title	Author/s	Publication
Prompt Engineering for Generative AI: Future-Proof Inputs for Reliable AI Outputs	James Phoenix & Mike Taylor	O'Reilly Media

REFERENCE BOOK(S):

Title	Author/s	Publication
Prompt Engineering for LLMs: The Art and Science of Building Large Language Model-Based Applications	John Berryman & Albert Ziegler	O'Reilly Media
Prompt Engineering in the Enterprise: Competitive Advantages through Generative AI and Large Language Models	Daniel Koch, Andreas Kohne & Nils Brechbühler	Springer

COURSE OUTCOME(S):

After the completion of the course, the students will be able to:

ICACS7040	Prompt Engineering & Generative AI
CO 1	Explain the fundamental concepts of Generative AI, Large Language Models, and the principles of prompt engineering.
CO 2	Design and apply effective prompts using techniques such as few-shot prompting and Chain-of-Thought across ChatGPT, Gemini, and Claude for academic and professional tasks.
CO 3	Apply prompting techniques for multimodal generative AI tools, including text, code, image, audio, and video generation platforms.
CO 4	Evaluate ethical, legal, and societal concerns of generative AI, including hallucinations, bias, copyright, and deepfakes, in professional practice.
CO 5	Design multi-step, prompt-driven solutions and AI agent workflows for real-world academic and professional problems.

MAPPING OF CO WITH PO:

ICACS7040	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	1	0	2	0	0	0
CO 2	1	2	2	0	3	0	0	0
CO 3	2	3	3	2	3	2	0	1
CO 4	0	1	1	0	0	3	3	0
CO 5	2	2	3	2	3	2	1	2

MAPPING OF CO WITH PSO:

ICACS7040	PSO1	PSO2	PSO3
CO 1	1	3	1
CO 2	2	3	2
CO 3	3	3	1
CO 4	1	2	3
CO 5	3	3	2

LEVEL OF REVISED BLOOM'S TAXONOMY IN ASSESSMENT

1: Remember	2: Understand	3: Apply

4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Introduction to Generative AI & Prompt Engineering	1,2
2	Prompt Design Principles & Techniques	3
3	Prompting for Text, Code & Productivity Tools	3
4	Multimodal Generative AI: Image, Audio & Video	5
5	Advanced Prompting & AI Agent Workflows	4
6	Ethics, Limitations & Future of Generative AI	6

P P Savani University

School of Engineering

Offered by: Department of Computer Science

COURSE DETAILS:

Course Code	ICACS7810
Course Name	Minor Project/Social Impact Project
Prerequisite Course/s	None

TEACHING SCHEME:

Credit	Total Learning Hours	Direct Contact Hours			Indirect Contact Hours				
		Theory	Practical	Tutorial	Project	PPT	Assignment	Term Work	Self-Learning
05	150	0	0	0	90	30	0	30	0

GD = Group Discussion | CS = Case Study | FV = Field Visit | PPT = Presentation

EXAMINATION SCHEME:

Theory		Practical		Tutorial		Project		PPT		Assignment		Term Work		Self-Learning		Total
CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	CE	ESE	
0	0	0	0	0	0	50	50	00	50	0	0	50	00	0	0	200

CE: Continuous Evaluation, ESE: End Semester Exam

OBJECTIVE OF THE COURSE:

To help learners to:

- Enable students to identify a real-world problem and formulate it into a feasible engineering/research project or industry-based assignment.
- Develop the ability to carry out literature survey, requirement analysis, and feasibility study relevant to the chosen project.
- Apply design thinking, planning, and project management skills to design and develop the proposed solution.
- Inculcate professional, ethical, and on-the-job working practices through industry exposure and team-based execution of the Minor Project.
- Build skills in technical documentation, presentation, and effective communication of the project work and outcomes.

COURSE CONTENT:

Module No.	Content	Hours	Weightage in%
1	**Problem Identification & Literature Survey:** Selection of project in consultation with Mentor/industry mentor; problem statement formulation; literature review of existing processes; feasibility study (technical, economic, operational); finalization of scope and objectives; submission of project synopsis.		13
2	**Requirement Analysis & System Design:** Detailed requirement gathering (functional & non-functional); selection of tools, technologies, and platform; system/process		20

	architecture and design (block diagrams, DFD/flowcharts as applicable); preparation of project plan with timelines and resource allocation; design review with guide.		
3	**Development / Implementation:** Module-wise development/execution of the proposed solution, prototype, model, or assigned industrial work; coding/fabrication/process execution as applicable; integration of modules/components; use of relevant tools, software, and equipment.		23
4	**Testing, Validation & Quality Check:** Unit and integration testing of the developed system/process; validation against requirements; performance evaluation and result analysis, Comparison with existing solutions, where applicable.		17
5	**Group Discussion & Presentation:** Group discussion on project findings and industry practices; case study of similar systems/products; field visit to relevant industry/organization (where applicable); structured PPT presentation of project progress before guide and peers; peer review and feedback session.		10
6	**Documentation, Demonstration & Final Evaluation:** Preparation of final project report ; final demonstration of the working project/deliverable; viva-voce and presentation before the evaluation panel.		17
	TOTAL	**0**	**100%**

COURSE EVALUATION & ASSESSMENT RUBRICS:

Component	CE/ESE	Evaluation Methodology / Rubrics	Marks	Passing Marks
Project	CE	Internal evaluation of project progress through periodic review meetings with guide based on synopsis, design, and implementation status. Average of both reviews converted to 50 marks.	50	00
	ESE	Final project evaluation through demonstration of working project/prototype and comprehensive viva-voce covering all phases of work.	50	25
PPT	ESE	Literature Review & Objective -15 Methodology & Execution -15 Report Quality & Structure -10 Presentation & Viva- 10	50	25
Term Work	CE	Regularity & Guide Feedback Log – 10 Marks Documentation & Project Report – 10 Marks Originality, Innovation & Problem-Solving Approach – 10 Marks Working Model/Output, Demonstration & Overall Execution – 20 Marks	50	00
		TOTAL MARKS	**200**	**100**

COURSE OUTCOME(S):

After the completion of the course, the students will be able to:

ICACS7810	Minor Project/Social Impact Project
CO 1	Identify a real-world problem and formulate a feasible project problem statement through literature survey and feasibility study.
CO 2	Design and plan the proposed solution/system, including requirement analysis, architecture, and project scheduling.
CO 3	Develop, implement, and integrate the proposed solution/prototype using appropriate tools, technologies, and engineering practices.
CO 4	Test, validate, and evaluate the performance of the developed system/process against the defined requirements.
CO 5	Demonstrate professional, ethical, and effective communication skills through project documentation, presentation, and viva-voce.

MAPPING OF CO WITH PO:

ICACS7810	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	1	0	2	0	0	0
CO 2	3	3	3	2	0	0	2	0
CO 3	3	2	3	0	3	2	2	2
CO 4	2	2	3	3	0	0	0	0
CO 5	0	0	0	0	0	0	3	3

MAPPING OF CO WITH PSO:

ICACS7810	PSO1	PSO2	PSO3
CO 1	2	1	3
CO 2	3	2	2
CO 3	3	3	2
CO 4	2	2	3
CO 5	1	1	2

LEVEL OF REVISED BLOOM'S TAXONOMY IN ASSESSMENT

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Problem Identification & Literature Survey:	1,2
2	Requirement Analysis & System Design	3
3	Development / Implementation	3
4	Testing, Validation & Quality Check	5
5	Case Study / Presentation	4
6	Documentation, Demonstration & Final Evaluation	6