Progressive Education Society's

Modern College of Arts, Science & Commerce (Autonomous)

Ganeshkhind, Pune – 411016

NATIONAL EDUCATION POLICY - 2020 (NEP-2020)

Basic and Honors Degree Program in

Bachelor of Computer Applications (Science): B.C.A. (Science)

(Faculty of Science & Technology)

Syllabus for T.Y. B.C.A. 5th and 6th Semesters

To be implemented from Academic Year 2026-2027

Semester -V T.Y. B.C.A (Science)							
Course Code	**Course Type**	**Course Title**	**Credits**		**Evaluation**		
			TH	**PR**	**CIA**	**CSE**	**Total**
24BCA35101	MC	Advanced Java	2	-	20	30	50
24BCA35102	MC	Lab on Advanced Java	-	2	20	30	50
24BCA35103	MC	Basics of ReactJS and NextJS	4	-	40	60	100
24BCA35104	MC	Operating System	2	-	20	30	50
24BCA35105	DSE	Go Programming	2	-	20	30	50
24BCA35106	DSE	Lab on Go Programming	-	2	20	30	50
OR (Choose any one DSE Theory with practical)							
24BCA35107	DSE	Object Oriented Software Engineering	2	-	20	30	50
24BCA35108	DSE	Lab on Object Oriented Software Engineering	-	2	20	30	50
24BCA35211	Minor	Artificial Intelligence	4	-	40	60	100
24BCA35407	VSEC	Lab on Basics of ReactJS and NextJS	-	2	20	30	50
24FP35604	FP	FP: Project Based Learning-III	-	2	20	30	50
Total			**14**	**8**	**220**	**330**	**550**
Total Credits: [14 (TH) + 8 (PR)] = 22							
TH: Theory **PR:** Practical **CIA:** Continuous Internal Assessment **CSE:** College Semester Examination							

Semester -VI T.Y. B.C.A (Science)							
Course Code	**Course Type**	**Course Title**	**Credits**		**Evaluation**		
			TH	**PR**	**CIA**	**CSE**	**Total**
24BCA36101	Major	Android Programming	2	-	20	30	50
24BCA36102	Major	Lab on Android Programming	-	2	20	30	50
24BCA36103	Major	React Native	2	-	20	30	50
24BCA36104	Major	Lab on React Native	-	2	20	30	50
24BCA36105	Major	Cloud Computing	2	-	20	30	50
24BCA36106	DSE	Software Testing Tools	2	-	20	30	50
24BCA36107	DSE	Lab on Software Testing Tools	-	2	20	30	50
OR (Choose any one Minor Elective [T + P])							
24BCA36108	DSE	Django Web Framework	2	-	20	30	50
24BCA36109	DSE	Lab on Django Web Framework	-	2	20	30	50
24BCA36212	Minor	Machine Learning	4	-	40	60	100
24OJT36605	OJT	OJT: Internship-I	-	4	40	60	100
Total			**12**	**10**	**220**	**330**	**550**
Total Credits: [12 (TH) + 10 (PR)] = 22							
TH: Theory **PR:** Practical **CIA:** Continuous Internal Assessment **CSE:** College Semester Examination							

T.Y. B.C.A. SEMESTER V	
Subject Code: 24BCA35101 **Subject Name: Advanced Java**	
Credits: 02 **Theory: 02 Hrs./Week**	**Continuous Internal Assessment: 20 Marks** **College Semester Examination: 30 Marks**

Course Objectives:

- To understand collection classes and interfaces.
- To know the process of application development using Graphical User Interface (GUI).
- To acquire knowledge about handling databases using Java.
- To study web components for developing web applications.

Course Outcomes:

On completion of the course, student will be able to–

- Design end to end applications using object-oriented constructs.
- Apply collection classes for storing java objects.
- Use Java APIs for program development.
- Handle abnormal termination of a program using exception handling.

Course Contents	
Unit-I: Collection	**06 Hrs.**
Introducing to Collections framework, List - ArrayList, LinkedList, Set - HashSet, TreeSet, Map - HashMap and TreeMap, Interfaces such as Comparator, Iterator, ListIterator, Enumeration, Exception handling- try, catch, finally, throw and throws,Inputs Outputs.	
Unit-II: Multithreading	**06 Hrs.**
Threads and Life cycle of thread, creating threads - Thread class, Runnable interface, Thread priorities, Running multiple threads, Synchronization and interthread communication, Thread Methods, Thread Scheduler, ThreadGroupclass	
Unit-III: JDBC Programming	**06 Hrs.**
The role of jdbc, Connectivity with database, JDBC Statements – Statement, Prepared Statement, Callable Statement, Scrollable and updatable result sets, Metadata – DatabaseMetadata, ResultSetMetadata (Database: PostgreSQL)	
Unit-IV : Servlets and JSP	**06 Hrs**
Servlets: Introduction to Servlets, Life cycle of a servlet, Tomcat configuration	

(Note: Only for Lab Demonstration), Handing get and post request (HTTP), Handling a data from HTML to a servlet, Session tracking – Cookies and Http Session, Life cycle of a JSP, Implicit Objects, Scripting elements – Declarations, Expressions, Scriplets, Comments, JSP Directives – Page Directive, include directive, Mixing Scriplets and HTML.	
Unit-V : Spring	**06 Hrs**
Introduction to the Spring Ecosystem, Setting up the Development Environment, Steps to create Spring application, Creating the First Spring Boot Application, Anatomy of a Spring Boot Application.	

Reference Books:

1) Core Java Volume I - Fundamentals by Cay S. Horstmann, 11th Edition, Prentice Hall, ISBN 978-0-13-516630-7

2) The Complete Reference by Herbert Shildt, 11th Edition, McGraw Hill Education, ISBN 978-260-44023-2

3) Java Beginners Guide by Herbert Shildt, 8th Edition, McGraw-Hill Education ISBN 978-1- 260-44021-8

4) Core Java Volume II – Fundamentals by Cay S. Horstmann, 11th Edition, Prentice Hall, ISBN 978-013-516631-4

5) Java 2 Programming Black Book by Steven Holzner, DreamTech Press, ISBN 978-93-5119-953-4

T.Y. B.C.A. SEMESTER V	
Subject Code: 24BCA35102 **Subject Name: Lab on Advanced Java**	
Credits: 02 **Practical: 04 Hrs./Week**	**Continuous Internal Assessment: 20 Marks** **College Semester Examination: 30 Marks**

Course Objectives:

- To understand collection classes and interfaces.
- To know the process of application development using Graphical User Interface (GUI).
- To acquire knowledge about handling databases using Java.
- To study web components for developing web applications.

Course Outcomes:

On completion of the course, student will be able to–

- Design end to end applications using object-oriented constructs.
- Apply collection classes for storing java objects.
- Use Java APIs for program development.
- Handle abnormal termination of a program using exception handling.

Lab Course Contents

Sr. No	Assignment
1	Collection
2	Multithreading
3	JDBC
4	Servlet
5	JSP
6	Mini project combined JDBC, Servlet and JSP

T.Y. B.C.A. SEMESTER V	
Subject Code: 24BCA35103	
Subject Name: Basics of ReactJS and NextJS	
Credits: 04	**Continuous Internal Assessment: 40 Marks**
Theory: 04 Hrs./Week	**College Semester Examination: 60 Marks**

Course Objectives:

- Build strong fundamentals in ReactJS and Next.js.
- Set up and manage modern frontend development environments.
- Understand how Angular fits into the modern frontend ecosystem.

Course Outcomes:

On completion of the course, student will be able to

- Develop interactive UIs using React components and hooks.
- Implement routing, forms, and state handling in React.
- Build modern web apps using Next.js routing, API routes, and data fetching.
- Understand the basic structure and concepts of Angular.

Course Contents

Unit I: Introduction to Modern Frontend Development	**10 Hrs**
Evolution of frontend frameworks, Basics of modern Web architecture, Concept of SPAs & MPAs, Overview of React, Overview of Next.js, Light introduction to Angular & where it's used, Setting up environment: Node.js, NPM, React, Next.js, Angular CLI.	
Unit II: React Fundamentals	**10 Hrs**
JSX and Virtual DOM, Functional & Class Components, Props and State, Reusable components, Component architecture patterns, Basic project structure.	
Unit III: React Hooks, Forms & Routing	**10 Hrs**
Hooks: useState, useEffect, useContext, Event handling, Controlled components, Form validation basics, React Router: BrowserRouter, Route, Link : Nested routing & dynamic routing.	
Unit IV: Next.js Basics	**10 Hrs**
Why Next.js?, File-based routing system, Pages & Components structure, Linking between pages, Layouts, Static vs Server-side Rendering, Image optimization basics	

Unit V: Next.js Advanced Concepts	10 Hrs
Data fetching: getStaticProps, getServerSideProps, getStaticPaths, API Routes (creating backend-like endpoints), Environment variables, Deployment basics (Vercel overview), Combining React + Next.js project architecture	
Unit VI: Basics of Angular (Concept-Level Only)	**10 Hrs**
(This unit is intentionally light, only for awareness) Angular architecture, Role of Modules, Components & Templates, Angular CLI overview, Basic Data Binding types, Concept of Directives (Structural & Attribute), Dependency Injection (overview only)	
Reference Books: • React: The Comprehensive Guide — Sebastian Springer • Learn Angular in 24 Hours — Lakshmi Kamala • Next.js Official Documentation	

T.Y. B.C.A. SEMESTER V	
Subject Code: 24BCA35104 **Subject Name: Operating System**	
Credits: 02 **Theory: 02 Hrs./Week**	**Continuous Internal Assessment: 20 Marks** **College Semester Examination: 30 Marks**

Course Objectives:

- To study algorithms for CPU-scheduling, Process Creation and Termination.
- To understand the notion of a Multithreading and Inter-Process Communication.
- To learn Critical-Section problems and Classical Process-Synchronization problems.
- To know the Deadlock Concept, different methods for Preventing or Avoiding Deadlocks and techniques for Memory Management.
- To learn and understand File System, Directory Structure, File Allocation Methods and Disk Scheduling Algorithms.

Course Outcomes:

On completion of the course, student will be able to–

- Describe algorithms for Process, Memory and Disk Scheduling
- Apply technique for Inter-Process Communication and Multithreading.
- Implement concept of Critical-Section
- Compare and contrast Deadlock Avoidance and Prevention.
- Use functions for File System Management

Course Contents

Unit I Process Scheduling and Multithreaded Programming	**07 Hrs.**
Process Scheduling – Scheduling queues, Schedulers, Context switch, Operations on Process, Inter-process Communication, Basic Concept – CPU-I/O burst cycle, Scheduling Types, Scheduling Criteria, CPU Scheduling Algorithms – FCFS, SJF, Priority scheduling, Round- robin scheduling, Threads (Introduction)	
Unit II Process Synchronization	**04 Hrs.**
Critical Section Problem, Semaphores, Classic Problems of Synchronization – the Bounded Buffer problem, reader- writer problem, Dining Philosopher problem.	
Unit III Deadlock	**05 Hrs.**
Deadlock Characterization – Necessary Conditions, Resource Allocation Graph Deadlock Prevention, Deadlock Avoidance - Safe state, Resource-Allocation-Graph	

Algorithm, Banker's, Algorithm, Deadlock Detection, Recovery from Deadlock – Process Termination, Resource Pre-emption	
Unit IV Memory Management	**06 Hrs.**
Logical Versus Physical Memory, Dynamic Loading and Dynamic Linking, Overlays, Swapping, Memory allocation Types, Fragmentation, Paging, Memory Protection, Segmentation – Basic concept, Virtual Memory Management – Demand paging, Page replacement – FIFO, Optimal, LRU, MRU, LFU, MFU	
Unit V File System	**05Hrs.**
File Concept, File Attribute, File Operations, File Types, File Structure, Access Methods - Sequential Access Method, Direct Access Method, Other Access, Methods, Directory overview, Single level directory, Two level directory, Tree structure directory, Acyclic graph directory, General graph directory, File System Structure and Implementation - Partitions and Mounting, Virtual, File Systems, Allocation Methods - Contiguous allocation, Linked allocation, Indexed allocation	
Unit VI Disk Scheduling	**03 Hrs.**
System Model, Deadlock Characterization – Necessary Conditions, Resource Allocation Graph Overview, Disk Structure, Disk Scheduling, FCFS Scheduling, SSTF Scheduling, C-SCAN, C-LOOK	
Case Study – Window OS	

Reference Books:

1) Operating System Concepts, Avi Silberschatz, Peter Galvin, Greg Gagne, 8th Edition, Wiley Asia
2) Operating Systems: Internals and Design Principles, William Stallings, Prentice Hall of India
3) Modern Operating Systems 4th Edition, by Andrew Tanenbaum, Herbert Bos
4) Distributed Operating Systems Concepts and Design, Pradeep K. Sinha, PHI
5) Advanced Concepts in Operating Systems, Mukesh Singhal and Niranjan G Shivaratri, Tata McGraw Hill Inc, 2001

T.Y. B.C.A. SEMESTER V	
Subject Code: 24BCA35105 **Subject Name: Go Programming**	
Credits: 02 **Theory: 02 Hrs./Week**	**Continuous Internal Assessment: 20 Marks** **College Semester Examination: 30 Marks**

Course Objectives:

- To study various programming constructs in GO
- To understand salient features in GO
- To know advance features in GO

Course Outcomes:

On completion of the course, student will be able to–

- Describe the core features and concepts in Go
- Write simple Go programs using functions
- Apply defining methods and Go Interfaces
- Use Go routines and Channels. Explore Go Packages

Course Contents

Unit I Introduction	**10 Hrs.**
Go Runtime and Compilations, Keywords and Identifiers, Constants and Variables, Operators and Expressions, Booleans, Numeric, Characters, Pointers and Addresses, Strings, if-else, switch, for loop, Iterations, Using break and continue, Implementing Arrays and structure, Use of Functions.	
Unit II Methods and Interfaces	**10 Hrs.**
Method Declarations, Functions vs. Methods, Pointer and Value Receivers, Method Values and Expressions, Interface Types and Values, Type Assertions and Type Switches, Method Sets with Interfaces, Embedded Interfaces, Empty Interfaces	
Unit III Goroutine and Channels	**10 Hrs.**
Goroutine Functions and Lambdas, Wait Groups, Channels, Sending and Receiving, Unbuffered and Buffered Channels, Multiplexing with select, Timers and Tickers, Packages and Workspaces, Exporting Package Names, Import Paths and Named Imports, Package Initializations, Blank Imports, Unit Testing with Test Functions, Table Tests and Random Tests, Benchmarking	

Reference Books:

1) Programming in GO , Dr. Dipali Meher, Dr. Kalyani Sambhoo
2) Introducing Go, Caleb Doxey, Oreilly publication
3) Learning Go Programming: Build Scalable Next-Gen Web Application using Golang (English Edition), Shubhangi Agarwal, BPB publication

E-Books:

1) Introducing Go By Caleb Doxey, Released January 2016Publisher(s): O'Reilly Media, Inc. ISBN: 9781491941959https://www.oreilly.com/library/view/introducing-go/9781491941997/ 2) Go Bootcamp by Matt Aimonettihttp://www.golangbootcamp.com/book

T.Y. B.C.A. SEMESTER V	
Subject Code: 24BCA35106 **Subject Name: Lab on Go Programming**	
Credits: 02 **Theory: 04 Hrs./Week**	**Continuous Internal Assessment: 20 Marks** **College Semester Examination: 30 Marks**

Course Objectives:

- To introduce essential programming features in GO
- To become familiar with programming techniques in GO

Course Outcomes:

On completion of the course, student will be able to–

- Write programs using features supported in GO
- Handle errors and utilize Goroutines and Channels

Lab Course Contents	
Sr. No.	**Assignment**
1	Introduction to Go Programming
2	Control Structures
3	Functions
4	Working with data
5	Structures and User Defined Types
6	Methods and Interfaces
7	Pointers and Memory Handling
8	Error Handling
9	Packages and Modular Programming
10	Go routines and channels

T.Y. B.C.A. SEMESTER V	
Subject Code: 24BCA35107	
Subject Name: Object Oriented Software Engineering	
Credits: 02	**Continuous Internal Assessment: 20 Marks**
Theory: 02 Hrs./Week	**College Semester Examination: 30 Marks**

Course Objectives:

- To understand the fundamental concepts, characteristics, and application domains of software and software engineering.
- To study various software development life cycle (SDLC) models and their relevance to modern software development practices.
- To learn the techniques of software requirement gathering, analysis, and design using appropriate tools.
- To understand the importance of software quality assurance, testing strategies, and test planning.
- To prepare students to apply software engineering principles for developing reliable and maintainable software systems.

Course Outcomes:

On completion of the course, student will be able to–

- Explain fundamental concepts, characteristics, and processes of software and software engineering.
- Compare and select suitable software development life cycle models for different types of projects.
- Apply requirement analysis and design tools such as DFDs, data dictionaries, and decision tables to real-world problems.
- Develop effective test plans and strategies to ensure software quality and reliability.

Course Contents

Unit 1: Fundamentals of Software Engineering	**6 Hrs**
Definition and Characteristics of Software – Software Application Domains – Software vs. Program – Need for Software Engineering – Characteristics of Good Software – Introduction to Object Orientation: Classes & Objects, Messages, Attributes & Methods, Encapsulation, Inheritance, Polymorphism – Overview of Object-Oriented Methodologies (Coad & Yourdon).	
Unit II: Software Development Life Cycle (SDLC)	**6 Hrs**
Conventional Models: Build-and-Fix, Waterfall, Prototyping, Iterative Enhancement, Spiral – Object-Oriented and Modern Models: Fountain Model, Rational Unified Process (RUP) – Overview of Agile and Incremental Development Approaches.	
Unit III: Requirement Analysis and Design Engineering	**6 Hrs**
Software Requirements: Stakeholder Identification, Functional & Non-Functional	

Requirements – Requirement Elicitation Techniques (Interviews, Brainstorming, Prototyping) – Software Requirement Specification (SRS): Nature and Organization – Analysis and Design Tools: Data Flow Diagram (DFD), Data Dictionary, Decision Table, Decision Tree, Input/Output Design, Pseudocode.	
Unit IV: Software Testing Strategies and Quality Assurance	**6 Hrs**
Software Quality Concepts – Software Testing Objectives and Principles – Verification and Validation – Testing Levels: Unit, Integration, System, Acceptance (Alpha & Beta) – Testing Approaches: Black-Box, White-Box, Agile Testing – Test Case Design, Test Planning, Defect Management, and Test Reporting.	
Unit V: Software Maintenance, Project Management	**6 Hrs**
Software Maintenance: Definition and importance of maintenance Types of maintenance: Corrective, Adaptive, Perfective, Preventive Challenges in software maintenance Software Project Management: Project planning and scheduling, Effort estimation, Risk management Team organization and roles Configuration Management: Version control basics, Importance of change management, Introduction to tools (e.g., Git – conceptual overview)	

Reference Books:

- Roger S. Pressman & Bruce R. Maxim – *Software Engineering: A Practitioner's Approach*, McGraw Hill Education, 9th Edition, 2020.
- Ian Sommerville – *Software Engineering*, Pearson Education, 10th Edition, 2015.
- Rajib Mall – *Fundamentals of Software Engineering*, PHI Learning, 5th Edition, 2018.
- Pankaj Jalote – *An Integrated Approach to Software Engineering*, Narosa Publishing House, 4th Edition, 2012.

T.Y. B.C.A. SEMESTER V

Subject Code: 24BCA35108

Subject Name: Lab on Object Oriented Software Engineering

Credits: 02	**Continuous Internal Assessment: 20 Marks**
Practical: 04 Hrs./Week	**College Semester Examination: 30 Marks**

Course Objectives:

- To understand and apply various phases of the software development life cycle (SDLC).
- To analyze and document software requirements effectively using standard templates and tools.
- To design software systems using modeling tools and object-oriented concepts.
- To develop and test software using structured and object-oriented testing approaches.
- To utilize bug tracking and reporting tools for software quality improvement.

Course Outcomes:

On completion of the course, student will be able to–

- Demonstrate understanding of software development processes and models.
- Prepare software requirement specifications, data flow diagrams, and design documents using appropriate tools.
- Apply software testing techniques such as black-box and white-box testing to evaluate software quality.
- Use software engineering tools for design, testing, and bug tracking in real-world scenarios.

Lab Course Contents

Sr. No	Assignments
1	Study of SDLC Models
2	Identify System Requirements
3	Prepare SRS Document
4	Draw DFD, UML & Data Dictionary
5	Create Decision Table & Tree
6	Design Input & Output Screens
7	Write Pseudocode & Flowchart
8	Prepare Test Plan & Test Cases
9	Perform Black & White Box Testing
10	Use Bug Tracking Tool

T.Y. B.C.A. SEMESTER V	
Subject Code: 24BCA35211 **Subject Name: Artificial Intelligence**	
Credits: 04 **Practical: 04 Hrs./Week**	**Continuous Internal Assessment: 40 Marks** **College Semester Examination: 60 Marks**

Course Objectives:

- To introduce the fundamental concepts and goals of Artificial Intelligence.
- To develop understanding of search techniques, knowledge representation, and reasoning.
- To explain probabilistic reasoning and decision-making under uncertainty.
- To expose students to learning techniques, NLP, and computer vision.
- To explore modern AI applications and understand ethical and social implications.

Course Outcomes:

On completion of the course, student will be able to–

- Describe the basic concepts, history, and applications of AI.
- Formulate real-world problems as state-space search and apply suitable search algorithms.
- Apply machine learning techniques for predictive and classification problems and understand AI applications such as chatbots, NLP, and computer vision.
- Evaluate the ethical and social impacts of AI technologies.

Course Contents

Unit I: Introduction to Artificial Intelligence	**8Hrs**
Introduction and motivation of AI, Comparison of AI, Machine Learning, and Deep Learning,Applications of AI (Health, Finance, Education, Industry, Government), AI Techniques: Search, Knowledge, and Learning, Intelligent Agents – Structure , types (simple reflex, model-based, goal-based, utility-based, learning agents), Nature of environments (deterministic, stochastic, episodic, dynamic).	
Unit II: Problem Solving and Search	**10 Hrs**
Problem formulation as State Space Search, Production systems and Problem characteristics, Search and Control strategies, Classical AI problems: Water Jug, Missionary-Cannibal, Monkey-Banana, Adversarial Search (Game Playing Agents)	
Unit III: Searching Algorithms	**10Hrs**
Uninformed Search (Blind Search):– Breadth-First Search (BFS), Depth-First Search (DFS), Informed Search (Heuristic Search):– Generate-and-Test, Hill Climbing, Best-First Search, A*, AO*, Means-End Analysis, Constraint Satisfaction Problems (CSPs), Application: Pathfinding or Puzzle Solving using Python	

Unit IV: Knowledge Representation and Reasoning	12 Hrs
Types of Knowledge – Procedural vs. Declarative, Knowledge Representation Approaches – Semantic Networks, Frames, Rules, Logical Representation – Propositional and Predicate Logic, Conversion to Clause Form and Resolution, Planning and Acting in the Real World (AI Agent Example), Probabilistic Reasoning – Bayes' Theorem and Bayesian Networks	
Unit V: Introduction to Machine Learning	**10 Hrs**
Introduction to Machine Learning, Types of Learning – Supervised, Unsupervised, Reinforcement, Learning from Observations – Statistical Learning Methods, Applications of Machine learning, Relation between ML,AI and statistics	
Unit VI: Advanced Topics and Ethical Considerations	**10Hrs**
Basic concept and Applications of-AI-Powered Catbots (SBI card chatbot (ILA)), Natural Language Processing (NLP) , Large Language Models (LLMs), Generative AI, Computer Vision and Image Recognition – Overview and Tools (OpenCV , TensorFlow and Keras, PyTorch,), Activity- Face Detection using OpenCV, Bias, Fairness, and Ethical AI, AI in Society – Challenges, Opportunities, and Future Directions	

Reference Books:

1. Russell, S. and Norvig, P. — *Artificial Intelligence: A Modern Approach*, Pearson.
2. Elaine Rich, Kevin Knight — *Artificial Intelligence*, Tata McGraw Hill.
3. Stuart Russell — *Intelligent Systems and Agents* (IIT Delhi Lecture Notes).
4. Nils J. Nilsson — *Principles of Artificial Intelligence*, Narosa Publishing.
5. Tom Mitchell — *Machine Learning*, McGraw Hill

Online resources:

Related Online Contents (MOOCS, SWAYAM, NPTEL, WEBSITES etc.)
Kaggle AI case studies, Google AI Education Portal.

AI Course Material — IIT Mandi (CS305) and IIT Patna (MTech in AI).

Suggested Activity / Mini Project Ideas

- AI-based Chatbot using Python
- Pathfinding Game using Search Algorithms
- Disease Diagnosis using Bayesian reasoning
- Face or Object Recognition using OpenCV

T.Y. B.C.A. SEMESTER V	
Subject Code: 24BCA35407	
Subject Name: Lab on Basics of ReactJs and NextJS	
Credits: 02	**Continuous Internal Assessment: 20 Marks**
Practical: 04 Hrs./Week	**College Semester Examination: 30 Marks**

Course Objectives:

- To understand Basics of React and Angular.
- To know the process of application development using different React and Angular Components
- To acquire knowledge about handling Events using React and Angular.

Course Outcomes:

On completion of the course, student will be able to–

- Design end-to-end applications using React.
- Apply collection of React components.
- Create reactive form using React and Angular

Lab Course Contents

Sr. No	Assignments
1	Environment Setup & First Components (React + Next.js)
2	React Components, JSX & Props
3	React State & Hooks (useState, useEffect)
4	React Forms & Validation (Controlled Components)
5	React Router (Multi-Page App + Dynamic Routing)
6	Mini React App (CRUD + Hooks + Routing)
7	Next.js File-based Routing & Dynamic Pages
8	Next.js Data Fetching (SSG + SSR)
9	Next.js Layouts & API Routes
10	Angular CLI Setup & First Component

T.Y. B.C.A. SEMESTER VI	
Subject Code: 24BCA36101	
Subject Name: Android Programming	
Credits: 02	**Continuous Internal Assessment: 20 Marks**
Theory: 02 Hrs./Week	**College Semester Examination: 30 Marks**

Course Objectives:

- To understand the Android Operating System
- To study Android Apps Development Cycle
- To learn to create Android Applications.

Course Outcomes:

On completion of the course, student will be able to–

- Create mobile applications on the Android Platform.
- Design and implement mobile applications involving data storage in SQLite database.
- Use location-based services while developing applications

Course Contents

Unit I: Android & Kotlin Fundamentals	6 Hrs
Modern Android Ecosystem- Evolution to Android 14/15, Android device types: Phones, Foldables, Tablets, Wearables, TV, Role of Jetpack libraries, Android Architecture (Updated)- Linux Kernel, HAL, Android Runtime & ART, Jetpack Architecture Components overview, Kotlin Essentials for Android- Variables, Functions, Null Safety, Data classes, Extensions, Collections, Coroutines intro (launch, suspend), Development Setup, Android Studio (latest), Gradle basics, project structure, Creating your first Kotlin app	
UNIT II — App Components, Navigation & Lifecycle	**6 Hrs**
Core Components- Activities (modern lifecycle), Fragments (when/when not to use), ViewModel & LifecycleOwner, Intents & Navigation- Explicit and Implicit Intents, Passing data using intents, App Links & Deep Links, Navigation graph (Jetpack Navigation component), Permissions & Security- Runtime permissions, Scoped storage basics, Background permission rules (Android 13+)	
UNIT III — UI Development with Jetpack Compose	**6 Hrs**
Compose Fundamentals, Composable functions, Recomposition, State & MutableState, Material 3 components, Layouts & Design- Column, Row, Box, LazyList (LazyColumn / LazyRow), Themes, typography, color schemes, Adaptive UI- Responsive layouts, Foldable	

/ dual-screen basics, Light & dark mode, User Interaction- Buttons, Text fields, Cards, Snackbars, dialogs, Handling gestures	
UNIT IV — Data, Architecture & Networking	**6 Hrs**
MVVM Architecture- ViewModel, LiveData vs StateFlow, Repository pattern, Local Storage- Room Database, Entities, DAO, Queries, DataStore (replacement for SharedPreferences), Networking - Retrofit / Ktor Client, Making API calls, JSON parsing with Kotlinx Serialization, Background Tasks- WorkManager, Coroutines & Flows for async tasks	
UNIT V — Advanced Android Features & Deployment	**6 Hrs**
Google Play Services- Location API & Maps, Places API (basic usage), Firebase Integration, Firebase Authentication, Firestore basics, Firebase Cloud Messaging (Push Notifications), Multimedia & Device Features, CameraX basics, Selecting media from gallery,Sharing content, App Optimization & Deployment - Debugging & profiling tools, App Signing, Release build & publishing overview	

Reference Books:

- Beginning Android4 Application Development, By Wei-Meng Lee WILEY India Edition WROX Publication
- Professional Android 4 Application Development, By Reto Meier WROX Publication
- Head First Android Development: A Brain-Friendly Guide, By David Griffiths and Dawn Griffiths

Reference Material :

Android Developers Official Guides

- Google Codelabs
- Android Programming with Kotlin for Beginners
- Jetpack Compose by Tutorials (Kodeco)

Websites:

1) The official site for Android developers - https://developer.android.com
2) https://www.tutorialspoint.com/android/index.htm
3) https://www.javatpoint.com/android-tutorial

T.Y. B.C.A. SEMESTER VI

Subject Code: 24BCA36102

Subject Name: Lab on Android Programming

Credits: 02	**Continuous Internal Assessment: 20 Marks**
Practical: 04 Hrs./Week	**College Semester Examination: 30 Marks**

Course Objectives:

- To understand the Android Operating System
- To study Android Apps Development
- To learn to create Android Applications

Course Outcomes:

On completion of the course, student will be able to–

- Create mobile applications on the Android Platform.
- Design and implement mobile applications involving data storage in SQLite database.
- Use location-based services while developing applications

Lab Course Contents

Sr. No	Assignments
1	First App: “Hello Android”
2	Basic UI Components
3	Counter App with State
4	Navigation Between Two Screens
5	Simple List Using LazyColumn
6	Add Items to a List (Basic Todo UI)
7	Basic Storage Using DataStore
8	Fetch Data From Public API
9	Simple Notifications
10	Google Map (View Only)

T.Y. B.C.A. SEMESTER VI	
Subject Code: 24BCA36103 **Subject Name: React Native**	
Credits: 02 **Theory: 02 Hrs./Week**	**Continuous Internal Assessment: 20 Marks** **College Semester Examination: 30 Marks**

Course Objectives:

- To introduce the fundamentals and ecosystem of React Native.
- To design user interfaces using React Native components and navigation systems.

Course Outcomes:

On completion of the course, student will be able to–

- Understand the basics, features, and structure of React Native
- Configure and run React Native apps using CLI and Expo.
- Build UI and implement navigation using Stack, Drawer, and Tab navigators.
- Use styles, Flexbox layouts, and state management effectively.
- Manage data through API calls, local storage, and validated forms.

Course Contents	
Unit 1: Fundamentals of React Native	**4 Hrs**
Introduction to React Native, Features and Use Cases of React Native, Difference between React vs React Native, Setting up React Native Development Ecosystem	
Unit II: React Native Environment Setup	**4 Hrs**
React Native Internals, Introduction to Expo and its Features, Working with Expo CLI, Navigating the Project Structure	
Unit III: UI Components & Navigation	**6 Hrs**
Introduction to Basic Components (Preview module), React Navigation Components, Implementing Stack, Drawer, and Tab Navigation , Understanding Android SDK Emulator , Working with Android Emulator, Introduction to iOS Simulator , Configuring iOS Simulator	
Unit IV: Styling, Layouts & State Management	**5 Hrs**
Introduction to Stylesheets (Preview module), Flexbox for Layouts ,Styled-Components , State Management using useState / useReducer , Passing Data via Props , Context API for State Management , useRef & Custom Hooks	
Unit V: Data Handling & Lists	**6 Hrs**

API Calling using Axios, Handling TextInput and Forms, Validating Forms with Formik and Yup, FlatList and SectionList, Pull-to-Refresh & Infinite Scrolling, Using AsyncStorage for Persistent Data	
Unit VI : Debugging, Testing & Deployment	**5 Hrs**
React DevTools (Preview module), Handling Errors and Debugging, Creating Android APK Files, Handling App Updates in React Native	

- Reference Books:
- React and React Native by Mikhail Sakhniuk & Adam Boduch
- Fullstack React Native published by CB India

T.Y. B.C.A. SEMESTER VI	
Subject Code: 24BCA36104	
Subject Name: Lab on React Native	
Credits: 02	**Continuous Internal Assessment: 20 Marks**
Practical: 04 Hrs./Week	**College Semester Examination: 30 Marks**

Course Objectives:

- Understand and Set Up the React Native Ecosystem.
- Implement Navigation and Basic Components.
- Design and style mobile app interfaces using Stylesheets, Layouts, Flexbox, and Styled-Components.

Course Outcomes:

On completion of the course, student will be able to–

- Understand the fundamentals, features, and use cases of React Native.
- Design and build user interfaces using core React Native components and navigation techniques.
- Apply styles, layouts, and state management using hooks, props, and Context API.

Lab Course Contents

Sr. No	Assignments
1	React Native Components
2	Working with User Interface Layouts
3	Implementing Navigation in React Native
4	Styling Components Using Styled-Components
5	Using State and Props
6	Using Context API for State Management
7	Handling Forms and Input Validation
8	API Integration using Axios
9	Using AsyncStorage for Local Data Storage
10	Debugging and Error Handling in React Native

T.Y. B.C.A. SEMESTER VI	
Subject Code: 24BCA36105	
Subject Name: Cloud Computing	
Credits: 02	**Continuous Internal Assessment: 20 Marks**
Theory: 02 Hrs./Week	**College Semester Examination: 30 Marks**

Course Objectives:

- To introduce students to the fundamental concepts and architecture of cloud computing.
- To understand virtualization, containerization, and cloud service delivery models.
- To explore cloud platforms, APIs, and tools used for cloud application development.
- To study emerging trends and security aspects of cloud computing.

Course Outcomes:

On completion of the course, student will be able to–

- Explain the concept of cloud computing, characteristics, and service models (IaaS, PaaS, SaaS).
- Understand virtualization and container technologies like Docker and Kubernetes.
- Explain how to use cloud development environments and APIs for building applications in the cloud.
- Recognize security, privacy, and compliance issues in cloud environments.

Course Contents	
Unit I – Cloud Computing Concepts and Service Models	**7 Hrs**
Overview and characteristics of Cloud Computing, Cloud service models: IaaS, PaaS, SaaS Deployment models: Public, Private, Hybrid, and Community Cloud, Cloud providers: AWS, Microsoft Azure, Google Cloud Platform, Case study: Hosting a website using cloud infrastructure	
Unit II – Virtualization and Containerization	**8 Hrs**
Concept and need for virtualization, Types of virtualization: Server, Storage, and Network Hypervisors: Type I and Type II, Introduction to Containerization: Docker basics, Images, and Containers, Overview of Kubernetes for container orchestration	
Unit III – Cloud Platforms and APIs	**7 Hrs**
Introduction to cloud development environments and APIs, Cloud SDKs and REST APIs Accessing cloud storage and databases, Serverless computing introduction (AWS Lambda example), Hands-on demo (theoretical): Deploying code to cloud via API	
Unit IV – Cloud Security and Compliance	**8 Hrs**

Security challenges in the cloud, Identity and Access Management (IAM) Data protection, encryption, and privacy, Cloud compliance standards (GDPR, ISO 27001, SOC 2), Disaster recovery and backup in cloud environments	

Suggested Practical Activities / Mini-Projects (Optional)

- Create a free-tier AWS or Azure account and launch a virtual machine.
- Host a static website using cloud storage service.
- Explore Docker container creation and image deployment.
- Demonstrate serverless function using AWS Lambda.
- Compare performance and cost across different service models.

Reference Books:

1. Arshdeep Bahga and Vijay Madisetti, *Cloud Computing: A Hands-On Approach*, Universities Press.
2. Thomas Erl, *Cloud Computing: Concepts, Technology & Architecture*, Pearson Education.
3. Barrie Sosinsky, *Cloud Computing Bible*, Wiley India.
4. Rajkumar Buyya et al., *Mastering Cloud Computing*, McGraw Hill Education.
5. Online resources: AWS Educate, Microsoft Learn, Google Cloud Skills Boost.

T.Y. B.C.A. SEMESTER VI	
Subject Code: 24BCA36106	
Subject Name: Software Testing Tools	
Credits: 02	**Continuous Internal Assessment: 20 Marks**
Theory: 02 Hrs./Week	**College Semester Examination: 30 Marks**

Course Objectives:

- To understand the importance and role of automated testing tools in the software development lifecycle.
- To gain practical exposure to various functional and non-functional software testing tools.
- To develop skills for creating, executing, and managing test cases using open-source and commercial tools.
- To analyze test results and prepare effective test reports using automation frameworks.

Course Outcomes:

On completion of the course, student will be able to–

- Understand the concepts and benefits of software testing tools and automation frameworks.
- Apply appropriate testing tools for functional, regression, performance, and security testing.
- Design, execute, and manage test cases using automation tools.
- Analyze and interpret test results to improve software quality.
- Compare and evaluate different software testing tools based on project requirements.

Course Contents	
Unit 1: Introduction to Software Testing Tools	**10 Hrs**
Review of software testing fundamentals, Need for automation in testing, Categories of software testing tools, Overview of open-source and commercial tools (Selenium, JMeter, Postman, Appium, Bugzilla, TestLink),Installation and environment setup	
Unit II: Functional and Regression Testing Tools	**7 Hrs**
Introduction to Selenium IDE, WebDriver, and Grid. Creating and executing test scripts,Locators and XPath, Test automation using Selenium, Introduction to test case management tools (TestLink / Jira)	
Unit III: Performance and API Testing Tools	**7 Hrs**
Introduction to performance testing, Working with Apache JMeter : Test plan, thread group, listeners, Load and stress testing, Introduction to API Testing using Postman ,Integrating reports and analyzing results	
Unit IV : Defect Tracking and Reporting Tools	**6 Hrs**

Defect lifecycle and bug tracking, Working with Bugzilla / Jira for defect reporting, Generating test summary and defect reports, Comparison of various tools, Introduction to Continuous Testing in CI/CD pipeline (Jenkins overview)	

Reference Books:

1. Srinivasan Desikan & Gopalaswamy Ramesh – *Software Testing: Principles and Practices*, Pearson Education, 2nd Edition, 2015.
2. Kshirasagar Naik & Priyadarshi Tripathy – *Software Testing and Quality Assurance: Theory and Practice*, Wiley India, 2nd Edition, 2013.
3. Elfriede Dustin – *Effective Software Testing: 50 Specific Ways to Improve Your Testing*, Pearson Education, 2003.
4. Unmesh Gundecha – *Selenium Testing Tools Cookbook*, Packt Publishing, 2nd Edition, 2015.

T.Y. B.C.A. SEMESTER VI

Subject Code: 24BCA36107

Subject Name: Lab on Software Testing Tools

Credits: 02	**Continuous Internal Assessment: 20 Marks**
Practical: 04 Hrs./Week	**College Semester Examination: 30 Marks**

Course Objectives:

- To understand the role and importance of software testing tools in improving software quality.
- To learn the installation, configuration, and operation of various software testing tools.
- To develop the ability to design, execute, and manage automated test cases.
- To analyze and interpret test results for better defect management and reporting.

Course Outcomes:

On completion of the course, student will be able to–

- Understand and demonstrate the use of various software testing tools and their features.
- Apply automation tools like Selenium and JMeter for functional and performance testing.
- Design, execute, and manage test cases using test management tools such as TestLink or Jira.
- Analyze test reports and summarize software quality through automation frameworks.

Lab Course Contents

Sr. No	Assignments
1	Installation and setup
2	Selenium IDE recording
3	Selenium WebDriver
4	Selenium Grid
5	Test Management using TestLink / Jira
6	Test Report Generation
7	Data-Driven Testing using Selenium
8	Automation Framework – Keyword or Hybrid Framework Basics
9	API Testing using Postman / REST Assured
10	Performance and Load Testing using Apache JMeter

T.Y. B.C.A. SEMESTER VI	
Subject Code: 24BCA36108	
Subject Name: Django Web Framework	
Credits: 02	**Continuous Internal Assessment: 20 Marks**
Theory: 02 Hrs./Week	**College Semester Examination: 30 Marks**

Course Objectives:

- To Learn Django Web Framework and Create Django apps and handle web pages with views
- To Work with static files and PostgreSQL database
- To Deploy Django project to the web

Course Outcomes:

On completion of the course, student will be able to–

- Set up virtual environment and install Django
- Connect pages using Django URLs
- Work with databases using Django models
- Work with static files and PostgreSQL database

Course Contents	
Unit 1: Introduction to Introduction to Django	**4 Hrs**
Installing Django, Creating the first project in Django, Difference between ASGI and WSGI Server, Creating a Django app and Understanding files Module.	
Unit II: Templates	**6 Hrs**
Returning HTTP response from Django and understanding URLS, Understanding the namespace of URLs and request parameters in views, Creating Dynamic URLs in django, Class-based views vs. function-based views.	
Unit III: Template Language	**5 Hrs**
Django template language, Template inheritance, Adding CSS. Mini project using the Django template engine.	
Unit IV: Django Models(I)	**5 Hrs**
Installing Databases, Connecting with different DB like mysql and postgres, SQLite, Defining models and fields, Parameters in Django fields, Relationships between models.	
Unit V: Django Models(II)	**5 Hrs**
What is ORM and what benefits of using ORM, Creating a CRUD operation using Django	

shell Null vs Blank, Create vs Save vs get _or_create, How to use faker to insert fake data, Master django query searching filtering lookups, ordering, Overwrite save update method , Slug field in django Bulk create in django.	
Unit VI (Forms)	**5 Hrs**
Meta classes in Django model, Understanding Soft delete and Object Manager in Django, Django HTML forms, Django Model forms, Adding validation on files in Django	
Reference Books: • Web framework for Python Django Book : compiled by Suvash Sedhain • django.pdf - https://app.readthedocs.org/projects/django/downloads/pdf/2.2.x/	

T.Y. B.C.A. SEMESTER VI

Subject Code: 24BCA36109

Subject Name: Lab on Django Web Framework

Credits: 02	**Continuous Internal Assessment: 20 Marks**
Practical: 04 Hrs./Week	**College Semester Examination: 30 Marks**

Course Objectives:

- To Learn Django Web Framework and Create Django apps and handle web pages with views
- To Work with static files and PostgreSQL database
- To Deploy Django project to the web

Course Outcomes:

On completion of the course, student will be able to–

- Set up virtual environment and install Django
- Connect pages using Django URLs
- Work with databases using Django models
- Work with static files and PostgreSQL database
- Deploy your Django project to the web

Reference

https://edurev.in/v/103834/Django-Tutorial-for-Beginners-1-Installing-Django#course_10142

Lab Course Contents

Sr. No	Assignments
1	Installation of Python, setting up DJango along with database system
2	Write a Python GUI program to import Tkinter package and create a window, Create simple login form,Create sample webpage
3	Write a program to create login and registration page using MVC architecture in DJango Framework
4	Write a program to create a sample page in Django by integrating Bootstrap.
5	Write a program to create a Login and Registration page using Django architecture.
6	Write a program to define models and perform database migrations in Django.
7	Write a program to create an application which fetches data from the database and also saves objects in the database.
8	Write a program to implement update and delete operations on database records in Django.
9	Write a program to work with static and media files in Django.
10	Write a program to implement form handling and validations in Django.

T.Y. B.C.A. SEMESTER VI	
Subject Code: 24BCA36212	
Subject Name: Machine Learning	
Credits: 04	**Continuous Internal Assessment: 40 Marks**
Theory: 04 Hrs./Week	**College Semester Examination: 60 Marks**

Course Objectives:

- Introduce fundamental concepts of machine learning and its applications.
- Understand various types of learning and model evaluation techniques.
- Explore popular machine learning algorithms and their real-world applications.

Course Outcomes:

On completion of the course, student will be able to–

- Understand fundamental concepts, methods, and applications of machine learning.
- Apply machine learning techniques to solve real-world problems.
- **Evaluate** models and **analyze** their performance.
- Understand ethical considerations and challenges in implementing machine learning systems.

Course Contents

Unit 1: Introduction to Machine Learning	**10 Hrs.**
Definition, Importance, and Applications of Machine Learning, Types of Machine Learning: Supervised, Unsupervised, Semi-supervised, Reinforcement Learning, Basic Terminologies: Dataset, Features, Labels, Training, Testing, Steps in the Machine Learning Pipeline. Essential Libraries and Tools: Jupyter notebbok, NumPy, SciPy, matplotlib, pandas.	
Unit II: Supervised Learning	**12 Hrs.**
Classification & Regression, Generalization, Overfitting and Under fitting, Regression: Linear Regression, Multiple Linear Regression, Logistic Regression, Maximum Likelihood Estimation Classification: Decision Trees, Ensembles of Decision Trees, Support Vector Machines (SVM), k-Nearest Neighbors, Naïve Bayes Classifiers, Random Forest Model, Model Evaluation Metrics: Accuracy, Precision, Recall, F1-Score.	
Unit III: Unsupervised Learning	**12 Hrs.**
Types of Unsupervised learning, Challenges in Unsupervised learning, Pre-processing and Scaling, Clustering: K-Means, Hierarchical Clustering, Agglomerative Clustering,	

Comparing and Evaluating Clustering Algorithms, Dimensionality Reduction: Principal Component Analysis (PCA, Applications of Unsupervised Learning.	
Unit IV: Reinforcement Learning	**8 Hrs.**
Concepts of Agents, Environment, Rewards, and Policy, Q-Learning: The Q Function, An Algorithm for Learning Q, An Illustrative Example, Convergence, Updating Sequence, Deep Q-Learning Basics, Nondeterministic Rewards and Actions, Generalizing from Examples, Real-World Applications.	
Unit V: Representing Data and Engineering Features	**10 Hrs.**
Categorical variables, One-Hot-Encoding (Dummy variables), Binning, Discretization, Linear models, and Trees, Interactions and Polynomials, Univariate Nonlinear Transformation, Automatic feature Selection, Univariate statistics, Model based feature selection, Iterative feature Selection,	
Unit VI: Tools and Applications	**8 Hrs.**
Introduction to Machine Learning Libraries: scikit-learn, TensorFlow, PyTorch, Case Studies: Spam Filtering, Recommendation Systems, Image Recognition, Ethical Considerations and Challenges in Machine Learning	

Reference Books:

1. Introduction to Machine Learning “by Abhishek Gupta”.
2. Machine Learning: Fundamentals and Applications “by P. Ranjan and B. Joshi”.
3. Machine Learning “by Saikat Dutt, Subramanian Chandramouli, and Amit Kumar Das”.
4. Machine Learning and Big Data “by Rajiv Chopra”.
5. Data Mining and Machine Learning “by V. K. Jain”.
6. Machine Learning “by Tom M. Mitchell”.
7. Pattern Recognition and Machine Learning “by Christopher M. Bishop”.
8. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow “by Aurélien Géron”.
9. Introduction to Machine Learning with python “by Andreas C. Muller & Sarah Guido”.