



Gyanmanjari
Innovative University

Course Syllabus
Gyanmanjari College of Computer Application
Semester-5 (BCA)

Subject: Front-End development using React and Angular – BCAWT10319

Type of course: Major Core

Prerequisite: Basic knowledge of HTML, CSS, and JavaScript (variables, functions, arrays, and DOM events). Familiarity with VS Code and a web browser is recommended.

Rationale:

This course introduces React.js and Angular — two leading front-end frameworks widely used in the software industry. Starting from the absolute basics and moving in a structured, step-by-step manner, the course covers all essential concepts that a beginner needs to understand and build modern web applications. Each topic is designed to be approachable for students who have only basic JavaScript knowledge, while still being thorough enough to form the foundation for a comprehensive textbook and question bank. On completion, students will be able to build simple but complete web applications using both frameworks and will be well-prepared for further learning in full-stack development.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks
CI	T	P		C	SEE		CCE		
			Theory		Practical	MSE	LWA	ALA	
3	0	2	4	75	25	30	20	50	200

Legends: CI-Class Room Instructions; T – Tutorial; P - Practical; C – Credit; SEE - Semester End Evaluation; MSE- Mid Semester Examination; LWA - Lab Work Assessment; V – Viva voce; CCE-Continuous and Comprehensive Evaluation; ALA- Active Learning Activities.

3 Credits * 25 Marks = 75 Marks (each credit carries 25 Marks) Theory

1 Credits * 25 Marks = 25 Marks (each credit carries 25 Marks) Practical

SEE 100 Marks will be converted in to 50 Marks

CCE 100 Marks will be converted in to 50 Marks

It is compulsory to pass in each individual component.



Course Content:

Sr. No	Course Content	Hrs	% Weightage
1	<p>Introduction to Web Frameworks & React Basics Introduction to Front-End Frameworks, React Introduction – Features, ReactJS Life Cycle, Applications, React Setup & Project - Installing Node.js and npm, creating a React app using Create React App, React JS ReactDOM, Understanding JSX, React Components, Props in React, State in React with useState, React Events - React Event Handlers ReactJS Rendering Elements, Rendering Lists with .map(), Conditional Rendering, React Forms</p>	9	20%
2	<p>React Hooks & Component Communication Introduction to React Hooks and their rules. useState for managing state, React useEffect Hook, useRef for DOM access. Context API for shared state. Types of React Components - Functional Components & Class Component, Pure Components, Styling components with CSS and className. Component lifecycle overview, React DevTools for debugging. Handling Events in React :onClick, onChange, onSubmit event handlers, HTML DOM vs React DOM</p>	9	20%
3	<p>React Router & Introduction to State Management Understanding Single Page Applications, Setting Up React Router, Defining Routes and Navigation, URL Parameters and Dynamic Routes, Programmatic Navigation and 404 Pages, Introduction to Global State Management, Redux — Core Concepts, Redux Toolkit — Modern Redux, When to Use Which State Solution, React Router — Advanced Patterns, Redux Toolkit — Async Actions with createAsyncThunk, Redux DevTools — Debugging State Changes, React Performance Optimisation</p>	9	20%
4	<p>Introduction to Angular & Core Concepts Introduction to Angular and its features, React vs Angular, Understanding Decorators Deeply, TypeScript — The Language of Angular, Angular CLI — Your Development Tool, Angular Components, Data Binding — Connecting Component and Template, Structural Directives — Changing the DOM, Angular Pipes — Transforming Display Data, Angular Modules (NgModule), Services and Dependency Injection, Angular Pipes — Creating Custom Pipes, Angular Forms Comparison — Template vs Reactive in Depth</p>	9	20%
5	<p>Angular Forms, Routing&HTTP</p>	9	20%



	<p>Two Approaches to Forms in Angular, Template-Driven Forms, Reactive Forms, Angular Router, Route Guards — Protecting Routes, Angular HttpClient, Observables — The Core of Angular Async, Angular Lifecycle Hooks, Building and Deploying Angular Applications, RxJS Operators — Power Tools for Observables, Angular Interceptors — Modifying All HTTP Requests, Lazy Loading — Loading Modules On Demand</p>		
--	---	--	--

Continuous Assessment:

Sr. No	Active Learning Activities	Marks
1	<p>React Profile Card using Components & Props Create a React app that displays a Student Profile Card using components and props. The parent App component stores student data (name, roll number, branch, year) and passes it to a child StudentCard component. Apply basic CSS styling. Upload a screenshot and a short note (5–6 lines) explaining props to the GMIU Web Portal.</p>	10
2	<p>Interactive Counter App using useState Hook Students will build a Counter application in React using the useState Hook. The app must have three buttons — Increment (+1), Decrement (–1), and Reset (back to 0) — and display the current count on screen. The count must not go below 0 (use a simple if condition). Students will take a screenshot of all three button interactions and write 4–5 lines explaining what the useState Hook does and why state is needed. Upload both the screenshot and the explanation on the GMIU Web Portal.</p>	10
3	<p>Multi-Page Website using React Router Build a 3-page portfolio website (Home, Projects, Contact) using React Router with a navigation bar. The Contact page shows a 'Thank You' message on submit using useState. Upload screenshots of all three pages and the zipped project folder to the GMIU Web Portal.</p>	10
4	<p>Angular Student List App with Add & Delete Build an Angular app that displays a student list using *ngFor. Add students via an input field with ngModel (two-way binding) and remove them with a Delete button. No backend needed — use a local array. Upload screenshots and the zipped project folder to the GMIU Web Portal.</p>	10
5	<p>Build a Simple Form (Student Registration) Students will create a basic student registration form using Angular with fields like Name, Email, and Course. They will apply simple form validation and display</p>	10



	the submitted data dynamically on the screen to understand form handling and data binding. Students have to upload their completed work on the GMIU Web Portal.	
	Total	50

Suggested Specification Table with Marks (Theory): 75

Distribution of Theory Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage	25%	40%	15%	10%	5%	5%

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcome:

CO	After learning the course, the students should be able to:
CO1	Understand and Build simple React web pages using JSX, functional components, props, state, and event handlers.
CO2	Analyze and Use React Hooks (useState, useEffect, useRef) to manage state, fetch data from a public API, and pass data between components.
CO3	Build multi-page React applications using React Router and manage shared state using Redux Toolkit.
CO4	Demonstrate Angular applications using components, data binding (interpolation, ngModel), directives (*ngIf, *ngFor), pipes, and services with dependency injection.
CO5	Prepare Angular forms with validation, implement routing with route parameters, fetch data from APIs using HttpClient, and deploy React and Angular apps online.

List of Practical:

Sr. No	Descriptions	Unit No	Hrs
1	Install Node.js and create a React app using Create React App. Explore the folder structure. Build a StudentInfo component that displays your name, branch, and semester.	1	2
2	Create a parent component with an array of 5 subjects (name, code). Pass each subject as props to a child SubjectCard component and render all cards using map() with a key prop.	1	2



3	Build a counter app with Increment, Decrement, and Reset buttons using useState. Display the count on screen. The count should not go below 0.	1	2
4	Fetch users from https://jsonplaceholder.typicode.com/users on page load. Display each user's name and email. Show 'Loading...' while fetching and an error message if the request fails.	2	2
5	Create a ThemeContext with React.createContext(). Wrap the app in a Provider with a 'light'/'dark' theme value. Add a Toggle button that switches themes and applies different background and text colours using useContext.	2	4
6	Create both functional and class components. Apply styling using CSS and className, and compare how both component types work.	3	4
7	Set up a Redux store using configureStore and createSlice with addTodo and removeTodo actions. Build a UI to add and delete tasks using useSelector and useDispatch.	3	4
8	Install Angular CLI and create a new Angular project. Generate a component using CLI and display basic student information using template and TypeScript.	4	4
9	Create a component that demonstrates interpolation, property binding, event binding, and two-way binding. Add an input field and display live user input using [(ngModel)].	5	4
10	Set up Angular Router and create two components (Home and About). Configure routes and navigate between them using routerLink.	5	2
			30

Instructional Method:

The course should be delivered primarily through live coding sessions where the instructor builds examples in front of the class using VS Code. Every concept must be accompanied by a working, runnable example that students can replicate immediately. Teachers may additionally use online sandboxes like StackBlitz or CodeSandbox for quick in-class demonstrations without any setup.

Supplementary resources recommended for students: Official React Documentation (react.dev), Official Angular Documentation (angular.dev), beginner courses on freeCodeCamp and Scrimba, and YouTube channels such as Traversy Media and Codevolution. From the course content, approximately 10% of topics (e.g., Context API, Redux Toolkit, Route Guards) are suitable for flipped classroom instruction where students study the concept through a video before the class discussion.

The internal evaluation will be based on Active Learning Assignments which are practical, focused, and completable by students working independently. Practical and Viva examinations will be conducted at the end of the semester.



Reference Books:

- [1] Learning React: Modern Patterns for Developing React Apps, 2nd Edition (2020), Alex Banks & Eve Porcello, O'Reilly Media
- [2] React Explained: Your Step-by-Step Guide to React (2020), Zac Gordon, self-published (beginner-friendly online book)
- [3] ng-book: The Complete Guide to Angular (2022), Nathan Murray, Felipe Coury, Ari Lerner & Carlos Taborda, Fullstack.io
- [4] Angular: Up and Running: Learning Angular, Step by Step (2018), Shyam Seshadri & Brad Green, O'Reilly Media

