

Institute/ School Name	School of Engineering and Technology		
Department Name	Department of Computer Science & Engineering		
Program Name	Bachelor of Engineering (Computer Science & Engineering): B.E (CSE)		
Course Code	24CS014	Course Name	Front-end Frameworks and Libraries
L-T-P (Per Week)	2-0-2	Course Credits	03
Academic Year	2025-26	Semester/Batch	4 <sup>th</sup> /2024-2028
Pre-requisites (if any)	None		
NHEQF Level	5	SDGs	4,9
Course Coordinator	Dr. Mohit		

### **1. Scope and Objective of the Course:**

This course covers the fundamentals and practical aspects of modern front-end web development using popular frameworks such as React, Angular, and Vue. It introduces core JavaScript concepts and advanced asynchronous programming using Promises, async/await, and API integration through Fetch and Axios. The course emphasizes React's component-based architecture, state management, hooks, forms, and lifecycle methods for building dynamic user interfaces. It includes TypeScript to enhance code robustness, type safety, and maintainability through structured programming and DOM manipulation. Hands-on projects, such as a sustainable weather application, a quiz app, and a form validator, help students apply concepts to real-world scenarios and develop industry-relevant skills.

### **2. Programme Outcomes (POs):**

At the end of the programme, students will be able to achieve knowledge about the following:	
PO 1	<b>Engineering knowledge:</b> Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO 2	<b>Problem analysis:</b> Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO 3	<b>Design/development of solutions:</b> Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO 4	<b>Conduct investigations of complex problems:</b> Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO 5	<b>Modern tool usage:</b> Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
PO 6	<b>The engineer and society:</b> Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO 7	<b>Environment and sustainability:</b> Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO 8	<b>Ethics:</b> Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO 9	<b>Individual and teamwork:</b> Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	<b>Communication:</b> Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	<b>Project management and finance:</b> Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	<b>Life-long learning:</b> Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**3. Course Learning Outcomes (CLO):**

After completing the course, the students will be able to:

- CLO1:** Understand and compare modern front-end frameworks, such as React, Angular, and Vue, and analyze advanced JavaScript concepts, including Promises, async/await, and API-based data fetching, to strengthen foundational front-end development skills.
- CLO2:** Develop interactive and modular web applications using React components, JSX, and Props to enhance employability by applying reusable and scalable user interface design principles.
- CLO3:** Evaluate and implement efficient state management and lifecycle methods in React using both class-based components and Hooks (useState, useEffect) to enhance problem-solving and implementation skills for dynamic data rendering.
- CLO4:** Apply TypeScript concepts such as typing, interfaces, classes, generics, and DOM manipulation to build robust, maintainable, and type-safe front-end solutions that support industry-ready employability.
- CLO5:** Design and develop practical web solutions, such as form validators and API-integrated applications, with an entrepreneurship mindset by addressing environmental and sustainability themes in software design.

**4. CLO-PO Mapping Matrix:**

Course Learning Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	NHEQF Level Descriptor
CLO1	-	-	-	-	H	-	-	-	-	-	M	-	Q1, Q2
CLO2	-	-	H	-	H	-	-	-	M	M	L	-	Q5
CLO3	-	M	H	-	H	-	-	-	-	-	-	-	Q4
CLO4	M	-	H	-	H	-	-	-	-	-	-	M	Q3
CLO5	-	-	H	-	H	-	H	M	M	M	M	H	Q5, Q6

**5. ERISE Grid Mapping:**

Feature Enablement	Level (1-5, 5 being highest)
Entrepreneurship	3
Research/Innovation	1
Skills	5
Employability	4

**6. Recommended Books (Reference Books/Text Books):**

- B01:** Duckett, J., JavaScript and jQuery: Interactive front-end web development (1<sup>st</sup> Edition). Wiley Publishing.
- B02:** Anthony, A., Nathaniel, M., & Ari, L., Fullstack React: The complete guide to ReactJS and friends (1<sup>st</sup> Edition). Fullstack.io.
- B03:** Cherny, B., Programming TypeScript: Making your JavaScript applications scale (1<sup>st</sup> Edition). O'Reilly Media.

**7. Other readings and relevant websites:**

Resources	Link of Journals, Magazines, Websites and Research Papers
<b>R1</b>	<a href="https://onlinecourses.swayam2.ac.in/nou24_cs09/preview">https://onlinecourses.swayam2.ac.in/nou24_cs09/preview</a>
<b>R2</b>	<a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript">https://developer.mozilla.org/en-US/docs/Web/JavaScript</a>
<b>R3</b>	<a href="https://react.dev/learn">https://react.dev/learn</a>
<b>R4</b>	<a href="https://codequotient.com">https://codequotient.com</a>
<b>R5</b>	<a href="https://www.typescriptlang.org/docs/">https://www.typescriptlang.org/docs/</a>

  

Resources	Link of Audio-Video resources
<b>V1</b>	<a href="https://www.youtube.com/playlist?list=PLJ5C_6qdAvBEJ6-TBzKoa1Ov21lwDzJfM">https://www.youtube.com/playlist?list=PLJ5C_6qdAvBEJ6-TBzKoa1Ov21lwDzJfM</a>
<b>V2</b>	<a href="https://www.youtube.com/watch?v=W6NZfCO5SIk">https://www.youtube.com/watch?v=W6NZfCO5SIk</a>
<b>V3</b>	<a href="https://www.youtube.com/watch?v=SqcY0GIETPk">https://www.youtube.com/watch?v=SqcY0GIETPk</a>
<b>V4</b>	<a href="https://www.youtube.com/watch?v=30LWjhZzg50">https://www.youtube.com/watch?v=30LWjhZzg50</a>
<b>V5</b>	<a href="https://youtu.be/d56mG7DezGs?si=oZ8PNUr-1GS-5jP4">https://youtu.be/d56mG7DezGs?si=oZ8PNUr-1GS-5jP4</a>

## 8. Recommended Tools and Platforms:

- VS Code, Nodejs, libraries of Nodejs
- Code Quotient (online platform- <https://codequotient.com/>)

## 9. Course Plan:

Lecture Number	Topics	Weightage in ETE (%)	Instructional Resources
1-2	JavaScript Recap: Functions, Arrays & Objects	25	R2, V2, B01
3-4	Introduction to Promises & async/await, Error handling in async/await		R2, R3 R4, V2, V3, B01
5-6	Practice Problems: <ul style="list-style-type: none"> <li>• Flatten a Multidimensional Array, Sort an Array of Object</li> <li>• Write an async function that: <ul style="list-style-type: none"> <li>○ Accepts an array of numbers</li> <li>○ Returns only even numbers after a 1-second delay</li> </ul> </li> </ul>		R2, R3 R4, V2, V3, B01
8-9	Recap of Fetch vs XMLHttpRequest, Making API request using fetch and rendering response.		R2, R4 V2, B01
10-11	Axios: Installing Axios, making API request using axios.		R2, R4, V2, V3, B01
12-13	Practice Problems: <ul style="list-style-type: none"> <li>• Post data on server.</li> <li>• Weather Application using an API.</li> </ul>		R2, R4, V2, V3, B01
14-15	Introduction to Front-End Frameworks: What is front-end framework? Difference between framework and library. Benefits of using a framework. Difference among React, Angular and Vue.		R1, R3, R4, V3, B02
16-17	Intro to React, Virtual DOM, Webpack, Using JSX.		R1, R3, R4, V3
18-19	Practice Problem: Installing React using CRA & Vite		R1, R3, R4, V3, B02
20 - 21	Component-driven approach: Class Component & Function Component, Class Component Life Cycle.		R1, R3, R4, V3, B02
22 - 23	Practice Problems: <ul style="list-style-type: none"> <li>• Convert Functional Component to Class Component</li> <li>• Stopwatch</li> </ul>	30	R1, R3, V3, B02
24-25	Props: Passing data between components, Handling default and Dynamic Props. Using props in a function and class component.		R1, R3, R4, V3, B02
26-27	Practice Problems: <ul style="list-style-type: none"> <li>• Create three components: Parent, Child1, and Child2. <ul style="list-style-type: none"> <li>○ Pass a message from the Parent component to Child1 and Child2 using props.</li> <li>○ Display the message in both Child1 and Child2.</li> </ul> </li> </ul> Create a functional component called StatusBadge that accepts a status prop. Based on the status value, display a badge with different styles: <ul style="list-style-type: none"> <li>○ Active -&gt; Green badge</li> <li>○ Inactive -&gt; Red badge</li> <li>○ Pending -&gt; Yellow badge</li> </ul>		R1, R3, R4, V3, B02
28-29	State Management: Updating and rendering data based on the state in Class Component & using useState in a function component.		R1, R3, R4, V3, B02
30	Forms and Event Handling in React.		R1, R3, R4, V3, B02

31-32	<p>Practice Problem: Create a Functional Component named EmployeeForm.</p> <p>Requirements:</p> <ul style="list-style-type: none"> <li>• Use state to store: name, department, salary</li> <li>• Create a form with three inputs and a submit button</li> <li>• Use one change handler for all inputs</li> <li>• On submit:           <ul style="list-style-type: none"> <li>◦ Prevent default behavior</li> <li>◦ Update state and display entered details below the form</li> </ul> </li> <li>• Add a Reset button to clear the form and re-render UI</li> </ul>		R1, R3, R4, V3, B02
33	Rendering data from API's in React components		R1, R3, R4, V3, B02
34-35	Practice Problem: Quiz App with API integration.		R1, R3, R4, V3, B02
36-37	Introduction to React Router DOM, Client-side routing in React, BrowserRouter, Routes and Route, Link and NavLink.		R1, R3, R4, V3, B02
38-39	Nested Routing, Layout Routes using Outlet, Dynamic Routing and Route Parameters, useParams hook, Navigation using useNavigate.		R1, R3, R4, V3, B02
40	Protected Routes (Private Route concept), Redirects and Not Found Route.		R1, R3, R4, V3, B02
41-42	Practice Problem: Create a Landing Page application using React Router DOM that demonstrates client-side routing and navigation.		R1, R3, R4, V3, B02
43 – 44	Introduction to TypeScript. Advantages of TypeScript over JavaScript. Setting up typescript in JavaScript, Converting TypeScript to JavaScript, Basic Types: Type Annotation, Number, String, Boolean, Object Type, Array		R5, R4, V4, B03
45	Functions in TypeScript: Functions, Function Types, Return Type, Optional, Default Parameter, Rest Parameter, Function Overloading.		R5, R4, V4, B03
46-47	<p>Practice Problem: Write a TypeScript program that processes student marks for a class.</p> <p>The program should store basic student information such as student id, name, and active status, using proper type annotations. It should maintain an array of marks scored by the student. Create a function that calculates the total marks and another function that returns the result status ("Pass" or "Fail") based on the total score. The program should also include a function that can accept either a student id or a student name and return a formatted string containing the student's details, demonstrating function overloading. Finally, compile the TypeScript file into JavaScript to run it in the browser or Node.js.</p>	25	R5, R4, V4, B03
48-49	Enums, Generics, Tuples and Advanced Types.		R5, R4, V4, B03
50-51	Classes and Interfaces		R5, R4, V4, B03
52-54	DOM Manipulation in TypeScript: Selecting DOM with strict types.		R5, R4, V4, B03
55-56	DOM Events: Adding event listeners in TypeScript,		R5, R4, V4, B03
57-58	Error Handling in DOM operations.		R5, R4, V4, B03
59-60	Practice Problem: Form Validator		R5, R4, V4, B03

**10. Industry Interventions:**

<https://codequotient.com/#>

**11. Innovative Pedagogies:**

- AI Chatbot Interface (Annexure – I)
- Code Review Workshop: Refactoring Legacy Component (Annexure-II)
- Personal Finance Tracker & Visualization Dashboard (Annexure -III)
- Professional Practice Integration – Collaborative React Development Workflow (Annexure – IV)

**12. Action plan for different types of learners**

Slow Learners	Average Learners	Advanced Learners
Remedial Classes	Practice Assignment	Inter-class competition

**13. Evaluation Scheme & Components:**

Evaluation Component	Type of Component	No. of Assessments	Weightage of Component	Mode of Assessment (Offline/ Online)
Internal Component 1	Formative Assessments (FAs)	01	20%	Online
Internal Component 2	Sessional Tests (STs)	02*	30%	
External Component	End Term Examination	01	50%	
<b>Total</b>			<b>100%</b>	

\* Average of the STs shall be used to determine the final marks.

**14. Details of Evaluation Components:**

Evaluation Component	Description	Syllabus Covered	Timeline of Examination	Weightage (%)
Internal Component 1	FA	Lab Assessment	Will be intimated in due course	20%
Internal Component 2	ST 01	Up to 40% (Lectures 1-24)		30%
	ST 02	40% - 80% (Lectures 25-48)		
External Component	End Term Examination*	100%		
<b>Total</b>				100%

\* Minimum 75% attendance is required to become eligible for appearing in the End Semester Examination.

**15. Format of Evaluation Components:**

Type of Assessment	Total Marks	Assignments / Interactions	1 Mark MCQ	2 Marks MCQ	5 Marks Coding	10 Marks Coding
Formative Assessments	20	20	-	-	-	-
Sessional Tests	40	-	10	5	2	1
End Term Examination	60	-	10	10	2	2

**16. Revision (if any):**

Academic Year of Previous Version	2024-2025	Percentage of Revision	11%
<b>Topics:</b>			
<ul style="list-style-type: none"><li>Introduction to React Router DOM, Client-side routing in React, BrowserRouter, Routes and Route, Link and NavLink. (Added)</li><li>Nested Routing, Layout Routes using Outlet, Dynamic Routing and Route Parameters, useParams hook, Navigation using useNavigate. (Added)</li><li>Protected Routes (Private Route concept), Redirects and Not Found Route. (Added)</li></ul>			

**17. This Document is.**

Designation	Name	Signature
Prepared by Course Coordinator	Dr. Mohit	<i>Mohit</i>
Verified by Assistant Dean	Dr. Hakam Singh	<i>Hakam Singh</i>
Date	12-01-2026	

## Annexure-I

S. No	Topics
1.	AI Chatbot Interface

**Objective:** students will prototype a conversational AI interface using React, integrating with a language model API to handle user queries on topics like coding help or trivia.

**Key Concepts Covered:** API Integrations, React Portals for modals, useReducer for complex state, Error Handling with Try-Catch.

**Outcome:**

- Students will learn how to create real-time conversational interfaces in React, integrate AI APIs, manage chat state, and handle dynamic responses with loading states.
- This challenge encourages rapid prototyping, and troubleshooting API integrations across different components.
- They will practice key React concepts like useReducer and portals, while gaining experience building collaborative AI tools under time pressure.

## Annexure-II

S. No	Topics
1	Code Review Workshop: Refactoring Legacy Components

**Description:**

Description: Students receive a "legacy" React codebase with outdated patterns (class components, prop drilling). They refactor to hooks in guided sessions, then review peers'.

**Outcome:**

Learners develop critical code review skills, understand evolution of React patterns, and practice maintainable refactoring—vital for industry roles involving codebase modernization.

## Annexure-III

S. No	Topics
1	Personal Finance Tracker & Visualization Dashboard

**Objective:** Students will create a web application that allows users to track their income, expenses, and savings. The application will use React for the front-end, integrate forms for data input, and visualize the financial data in different suitable react component.

**Key Concepts Covered:** React Component, State Management, Event Handling, Conditional Rendering

**Outcome:**

- Students will learn how to create interactive and dynamic React applications, handle user input, manage state, and implement form validation and error handling.
- They will practice using key React concepts like components, props, state, and event handling, while also gaining practical experience in creating useful, real-world applications.
- This project also encourages problem-solving and troubleshooting, especially when managing the flow of data across different components.

## Annexure-IV

S. No	Topics
1	Professional Practice Integration – Collaborative React Development Workflow

**Objective:**  
Expose students to industry-standard frontend development workflows by collaboratively building and managing a React application using professional tools and practices.

**Key Concepts Covered:**  
Component-based development in React, version control with Git, collaborative coding practices, branching strategies, code merging, and basic project documentation.

**Implementation Steps:**

Divide students into small teams to collaboratively develop different React components of a single application. Use GitHub for repository management, branching, and pull requests.  
Assign roles such as component owner, reviewer, and integrator.  
Perform code reviews and resolve merge conflicts collaboratively.  
Maintain a README and basic technical documentation for the project.

**Outcome:**

- Students will develop collaborative and professional coding skills essential for real-world frontend teams.
- They will learn how React projects are managed and maintained in a team-based environment.
- Learners will gain practical experience with version control, peer review, and code integration.