



**24SDCS01A/24SDCS01E
FRONT END DEVELOPMENT
FRAMEWORKS
WORKBOOK**

**Front End Development Frameworks Team
Department of Computer Science & Information
KLEF (Deemed to be University), Green Fields, Vaddeswaram, Andhra Pradesh**

24SDCS01A/24SDCS01E
**FRONT END DEVELOPMENT
FRAMEWORKS
WORKBOOK**

| | |
|---------------------|--|
| STUDENT NAME | |
| STUDENT ID | |
| YEAR | |
| SEMESTER | |
| SECTION | |
| FACULTY NAME | |

KL University Vision and Mission

Vision :

To be a globally renowned university.

Mission :

To impart quality higher education and to undertake research and extension with emphasis on application and innovation that cater to the emerging societal needs through all-round development of the students of all sections enabling them to be globally competitive and socially responsible citizens with intrinsic values.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 3 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Department of Computer Science & Information

| Technology Existing Vision & Mission of the Department | Proposed Vision & Mission of the Department |
|--|--|
| <p>Vision: To Promote Teaching and Learning that includes the latest Tool-based skill-driven learning in Information Technology which provides a strong practical base for the Graduate Professionals.</p> <p>Mission: To become a Centre of Excellence in Information Technology with a strong Research environment that produces top-class competent Professionals to the real IT world.</p> | <p>Vision: To be a leader in Information Technology education and research, driving innovation in emerging technologies and empowering students to develop ethical, sustainable, and impactful IT solutions for a digitally connected world.</p> <p>Mission:</p> <ul style="list-style-type: none">➤ To provide high-quality, tool-based hands-on education that integrates theory with industry-driven learning.➤ To foster a research ecosystem that promotes innovation and practical applications in IT.➤ To equip students with leadership skills, entrepreneurship mindset, and interdisciplinary adaptability.➤ To instill ethical and responsible computing practices in emerging domains.➤ To prepare graduates to excel in the IT industry with technical expertise and continuous learning. |

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 4 of 81 |

Program Educational Objectives

| Existing Program Educational Objectives | Proposed Program Educational Objectives |
|--|--|
| ► Practice engineering in a broad range of industrial, societal and real-world applications. | ► Demonstrate proficiency in emerging tools, methodologies, and emerging technologies, enabling them to contribute effectively to industry-driven projects and real-world problem-solving. |
| ► Pursue advanced education, research and development, and other creative and innovative efforts in science, engineering, and technology, as well as other professional careers. | ► Engage in research and innovation, applying analytical and critical thinking skills to develop novel solutions that address contemporary challenges. |
| ► Conduct themselves in a responsible, professional, and ethical manner. | ► Exhibit leadership qualities, an entrepreneurial mindset, and the ability to adapt to interdisciplinary environments, fostering career growth and business ventures in the IT sector. |
| ► Participate as leaders in their fields of expertise and in activities that support service and economic development throughout the world. | ► Uphold ethical computing practices, social responsibility, and professional integrity while continuously updating their knowledge to stay relevant in the evolving landscape. |
| ► An ability to Identify, Design, and Analyze complex computer systems, Implement and Interpret the results from those systems. | ► To develop expertise in tool-based solutions, applying theoretical knowledge to industry-driven challenges in software development, data analytics, and emerging technologies. |
| ► An ability to select and apply current techniques, skills, and tools necessary for computing practice and integrate IT-based solutions into the user environment effectively. | ► To foster innovation through research-driven problem-solving while upholding ethical and responsible computing practices to address IT challenges. |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

SYLLABUS:

Module 1: Git, Version Control, and Introduction to HTML/CSS

Introduction to Git and Version Control Systems, Installing Git and Initial Setup, Git Basics (git init, git add, git commit), Understanding Git Workflow, Branching and Merging, Working with Remote Repositories (git clone, git pull, git push), Resolving Merge Conflicts, Git Logs and Viewing History, Git Reset, Revert, and Checkout, Creating and Managing Branches, Basic Git Commands for Daily Use, Introduction to HTML5 structure, Semantic HTML5 tags, Forms and input types, Media elements (audio, video, images), CSS syntax and selectors, Introduction to responsive web design concepts.

Module 2: Advanced HTML5, CSS3, and Core JavaScript

Advanced HTML5 elements (section, aside, nav, header, footer), HTML5 form enhancements (placeholder, required, autofocus, pattern), CSS positioning (static, relative, absolute, fixed, sticky), CSS media types and feature queries, CSS transitions and keyframe animations, Introduction to JavaScript, JavaScript syntax and data types, operators, functions and arrow functions, conditional statements and loops, objects, arrays, set and maps, DOM manipulation, event handling, ES6+ features (let, const, destructuring, spread/rest operators), promises and async/await, fetch API, form validation, error handling, localStorage and sessionStorage, modular JavaScript, classes and objects, Built In Objects, working with JSON.

Module 3: React Fundamentals and UI Development

Introduction to React, JSX syntax and expressions, functional components, props and state management, event handling in React, conditional rendering, list rendering and keys, useEffect and useState hooks, component lifecycle overview, forms and controlled components, React Router for navigation, lifting state up, component communication, structure of React apps, reusable components and atomic design principles, state management using Redux Toolkit, UI libraries like Material UI and Tailwind CSS, deployment using Vite.

Module 4: Advanced React and Working with Node.js/Express

Context API for global state management, hooks for reusable logic, lazy loading and code splitting for performance, testing with React Testing Library, introduction to Node.js (event loop, non-blocking I/O), setting up a Node.js project with npm, building a simple server with core modules (http), introduction to Express.js (setup, routing), creating RESTful endpoints (GET, POST), middleware basics, input validation and sanitization.

Module 5: Development with Next.js and MongoDB

Introduction to Next.js and differences with React, file-based routing, static generation (SSG) and server-side rendering (SSR), dynamic routing and API routes, getStaticProps and getServerSideProps, image optimization and built-in support for SEO, middleware and authentication using NextAuth.js, integration with MongoDB with all the insert, delete & update operations.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 6 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

List of Experiments

| | |
|---|----|
| Exercise 1: Project Initialization and First Commit..... | 10 |
| █ Scenario:..... | 10 |
| 🔧 Tasks:..... | 10 |
| 📋 Expected Output:..... | 10 |
| Viva: | 13 |
| Exercise 2: Branching and Feature Integration..... | 15 |
| █ Scenario:..... | 15 |
| 🔧 Tasks:..... | 15 |
| 📋 Expected Output:..... | 15 |
| Viva: | 18 |
| Exercise 3: Responsive Web Page and Git History Exploration..... | 20 |
| █ Scenario:..... | 20 |
| 🔧 Tasks:..... | 20 |
| 📋 Expected Output:..... | 20 |
| Viva: | 24 |
| Exercise 4: HTML5 and CSS Positioning with Responsive Design..... | 26 |
| █ Scenario:..... | 26 |
| 🔧 Tasks:..... | 26 |
| 📋 Expected Output:..... | 26 |
| Viva: | 29 |
| Exercise 5: JavaScript DOM Manipulation and Event Handling for Form Validation..... | 32 |
| █ Scenario:..... | 32 |
| 🔧 Tasks:..... | 32 |
| 📋 Expected Output:..... | 32 |
| Viva: | 36 |
| Exercise 6: ES6+ Features and Asynchronous JavaScript (Weather App)..... | 39 |
| █ Scenario:..... | 39 |
| 🔧 Tasks:..... | 39 |
| 📋 Expected Output:..... | 39 |
| Viva: | 43 |
| Exercise 7: Modular JavaScript and Working with JSON (Online Bookstore)..... | 46 |
| █ Scenario:..... | 46 |

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 7 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|---|----|
| Tasks:..... | 46 |
| Expected Output:..... | 46 |
| Viva: | 50 |
| Exercise 8: To-Do List Application with React Hooks..... | 52 |
| Scenario: | 52 |
| Tasks:..... | 52 |
| Expected Output: | 52 |
| Exercise 9: Book Explorer with React Router and Props | 58 |
| Scenario: | 58 |
| Tasks:..... | 58 |
| Expected Output: | 58 |
| Viva: | 61 |
| Exercise 10: Feedback Collector with Redux Toolkit | 64 |
| Scenario: | 64 |
| Tasks:..... | 64 |
| Expected Output: | 64 |
| Viva: | 69 |
| Exercise 11: Full-Stack Feedback Form with MongoDB | 72 |
| Scenario: | 72 |
| Tasks:..... | 72 |
| Expected Output: | 72 |
| Viva: | 74 |
| Exercise 12: Authentication with NextAuth.js | 76 |
| Scenario: | 76 |
| Tasks:..... | 76 |
| Expected Output: | 76 |
| Viva: | 79 |

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 8 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

INDEX

| Exercise Number | Date of Evaluation | Implementation (20M) | Output (20M) | Viva Voce (10M) | Total (50M) | Faculty Signature |
|------------------------|---------------------------|-----------------------------|---------------------|------------------------|--------------------|--------------------------|
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |
| 7 | | | | | | |
| 8 | | | | | | |
| 9 | | | | | | |
| 10 | | | | | | |
| 11 | | | | | | |
| 12 | | | | | | |
| 13 | | | | | | |
| 14 | | | | | | |
| 15 | | | | | | |

Exercise 1: Project Initialization and First Commit

Scenario:

You have recently joined a software development team as a frontend developer. Before starting feature development, your team lead has asked you to set up version control for the project and make the first commit with a basic HTML structure.

Tasks:

1. **Install Git** on your system and configure your user information using:
 - git config --global user.name "Your Name"
 - git config --global user.email "you@example.com"
2. **Initialize a local Git repository** in your project folder using git init.
3. **Create a basic index.html file** that includes:
 - Your full name
 - Your title (e.g., Frontend Developer)
 - A short descriptive paragraph about yourself or your role
4. **Track and commit your file** using the following Git commands:
 - git add index.html
 - git commit -m "Initial commit with HTML structure"
5. **Verify the commit history** using git log.

Expected Output:

- A Git repository containing a committed index.html file.
- Screenshot showing the output of the git log command with the initial commit details.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 10 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 11 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 12 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva:

1. What is the purpose of using `git init`, and what does it do internally?
2. Why is it important to configure `user.name` and `user.email` in Git, and how does this affect your commits?
3. Explain the difference between `git add` and `git commit`. What happens in each step?
4. What information does the `git log` command provide? How is it useful in project management?
5. Can you describe the structure and basic elements of a valid HTML file? What tags did you use in your `index.html` file?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 13 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|-----------------------------------|---|
| | Marks Secured: ___out of 50 |
| Evaluator Remark (if Any): | Signature of the Evaluator with Date |

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 14 of 81 |

Exercise 2: Branching and Feature Integration

Scenario:

You are working on a web project that requires the addition of two new sections: an **About** section and a **Contact** form. To follow best practices, you will develop each feature in a separate Git branch and then integrate them into the main branch using proper version control techniques.

Tasks:

1. **Create and switch** to a new branch named about-feature.
 2. In the about-feature branch, **add an About section** to the webpage using appropriate HTML headings and paragraph text.
 3. **Commit your changes** with a meaningful message describing the addition.
 4. Create a second branch named contact-feature from the main branch.
 5. In the contact-feature branch, **add a Contact form** containing fields for Name, Email, and Message.
 6. **Commit your changes** and **merge both branches** (about-feature and contact-feature) into the main branch one by one.
 7. **Resolve any merge conflicts** that arise and verify that the final version of index.html includes both the About section and the Contact form.
-

Expected Output:

- A complete index.html file that contains both the About section and the Contact form.
- Git command log demonstrating:
 - Branch creation and switching
 - Feature commits
 - Successful merges

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 15 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 16 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 17 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva:

1. What is the purpose of using branches in Git?
2. How do you create and switch to a new branch?
3. What happens when two branches are merged in Git?
4. How do you resolve merge conflicts?
5. What is the difference between a fast-forward and a non-fast-forward merge?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 18 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|-----------------------------------|--|
| | Marks Secured: _____ out of 50 |
| Evaluator Remark (if Any): | Signature of the Evaluator with Date |

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 19 of 81 |

Exercise 3: Responsive Web Page and Git History Exploration

Scenario:

Your manager has assigned you the task of enhancing an existing webpage to make it mobile-friendly and to explore the Git version history for auditing purposes. The goal is to ensure the webpage displays well on various screen sizes while maintaining proper version control documentation.

Tasks:

1. **Modify the existing HTML page** to enhance responsiveness:
 - Add a responsive image.
 - Use CSS media queries to ensure the layout adapts to different screen sizes.
2. **Add a separate CSS file:**
 - Create a style.css file and link it to the HTML document for styling.
3. **Track your changes using Git:**
 - Use git add, git commit, and git status as you make updates.
4. **Explore Git history and version control features:**
 - Demonstrate the use of git log, git diff, git checkout, and git reset to manage and explore changes in the repository.

Expected Output:

- A responsive HTML webpage styled with an external CSS file.
- Screenshots showing the webpage in:
 - **Desktop view**
 - **Mobile view**
- Command line outputs demonstrating use of:
 - git diff
 - git checkout
 - git reset

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 20 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 21 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 22 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 23 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva:

1. What are media queries in CSS and how do they help in responsive design?
2. How can you create a responsive image in HTML/CSS?
3. What is the use of git diff, and how does it help in code review?
4. How do git checkout and git reset differ in functionality?
5. Why is exploring Git history important in software development?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 24 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|-----------------------------------|---|
| | Marks Secured: ___out of 50 |
| Evaluator Remark (if Any): | Signature of the Evaluator with Date |

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 25 of 81 |

Exercise 4: HTML5 and CSS Positioning with Responsive Design

Scenario:

You have been hired to develop a responsive promotional webpage for a local bakery. The client wants a modern layout that includes a slideshow of featured products and a user-friendly experience across all devices. The navigation bar should always stay at the top of the screen while scrolling, and the design must adapt well on both desktop and mobile devices.

Tasks:

1. **Create the basic structure** of the webpage using semantic HTML5 elements:
 - o <header>, <nav>, <section>, and <footer>.
2. **Implement CSS positioning**:
 - o Use fixed positioning for the navigation bar to keep it pinned to the top during scrolling.
 - o Use relative, absolute, or other positioning techniques as necessary to achieve layout goals.
3. **Create a featured product slideshow**:
 - o Use CSS transitions or keyframe animations to create an auto-playing or manually controlled slideshow.
4. **Make the layout responsive**:
 - o Apply CSS media queries to adjust styles for different screen sizes (especially for mobile view).
5. **Test and validate**:
 - o Preview the webpage in both mobile and desktop views to ensure functionality and responsiveness.

Expected Output:

- A fully functional HTML5 webpage with:
 - o Fixed navigation bar.
 - o Animated slideshow for featured products.
 - o Layout that adapts to various screen sizes.
- Screenshots of the webpage in:
 - o **Desktop view**.
 - o **Mobile view**.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 26 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 27 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 28 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva:

1. What is the difference between relative, absolute, and fixed positioning in CSS?
2. How do semantic HTML5 elements enhance the structure of a webpage?
3. How would you implement a fixed navigation bar using CSS?
4. What are CSS keyframe animations and how are they used in a slideshow?
5. How can you test responsiveness across different screen sizes?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 29 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 30 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|-----------------------------------|---|
| | Marks Secured: ___out of 50 |
| Evaluator Remark (if Any): | Signature of the Evaluator with Date |

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 31 of 81 |

Exercise 5: JavaScript DOM Manipulation and Event Handling for Form Validation

▀ Scenario:

You have been assigned to develop a user-friendly registration form for a client's website. The form should validate user inputs in real-time and guide the user with interactive feedback. Your solution should use JavaScript for dynamic validation and DOM manipulation to show error or success messages based on the input.

🔧 Tasks:

1. Create an HTML form with the following fields:

- Full Name
- Email
- Password
- Phone Number

Ensure required fields use proper HTML5 validation attributes like required, pattern, and type.

2. Use JavaScript to add event listeners for each input field to:

- Check if the input is empty.
- Validate the email format using a regular expression.
- Assess password strength (minimum length, use of special characters, etc.).
- Validate the phone number format.

3. Implement dynamic feedback using DOM manipulation:

- Display inline error messages for each invalid field.
- Change the border color or background color to indicate valid/invalid inputs.
- Show a success message when all validations pass.

4. Handle form submission:

- Prevent submission if any field is invalid.
- Display a success message dynamically when the form is submitted with valid inputs.

▀ Expected Output:

- A registration form with real-time validation.
- DOM-based error messages and visual cues for valid/invalid inputs.
- A success message shown only when the form is correctly filled out.
- Screenshots demonstrating:
 - Validation errors.
 - Correct form submission with the success message.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 32 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 33 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 34 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 35 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva:

1. How does JavaScript interact with the DOM?
2. What are event listeners and how are they used for form validation?
3. How would you use regular expressions to validate email input?
4. What is the role of preventDefault() in form submission?
5. How can you dynamically show success or error messages in a form?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 36 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 37 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|----------------------------|--------------------------------------|
| Evaluator Remark (if Any): | Marks Secured: ___out of 50 |
| | Signature of the Evaluator with Date |

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 38 of 81 |

Exercise 6: ES6+ Features and Asynchronous JavaScript (Weather App)

情境:

You are building a weather application for a client that provides users with real-time weather information based on their city input. The app must retrieve data from a public REST API and use modern JavaScript features like fetch, Promises, and async/await for asynchronous processing. The app should also remember the user's last searched city using localStorage to enhance user experience.

任务:

1. **Design a simple user interface:**
 - An input field for the user to enter a city name.
 - A button to initiate the weather data fetch process.
2. **Use the fetch API** to retrieve weather information (e.g., temperature, weather description) from a public API based on the entered city name.
3. **Handle asynchronous operations** using:
 - Promises or async/await syntax.
 - Proper error handling for failed API calls or invalid user inputs.
4. **Display the retrieved data dynamically:**
 - Update the DOM to show the current weather (temperature, description, etc.) in a user-friendly layout.
5. **Implement localStorage functionality:**
 - Save the last searched city in localStorage.
 - On page load, check localStorage and automatically fetch and display weather for the last searched city, if available.

预期输出:

- A functional weather application interface.
- Weather data (e.g., temperature, condition) displayed dynamically based on user input.
- Error message shown for invalid inputs or failed API responses.
- City name retained using localStorage and used on next page load.
- Screenshots showing:
 - UI before and after fetching data.
 - LocalStorage-based weather data retrieval on reload.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 39 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|-----------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 40 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|-----------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 41 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 42 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva:

1. What is the difference between Promises and async/await in JavaScript?
2. How does fetch() work, and what kind of data does it return?
3. What is localStorage and how can it be used in a weather app?
4. How do you handle errors when making API calls?
5. Why is it important to update the DOM dynamically in a weather application?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 43 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 44 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|-----------------------------------|---|
| | Marks Secured: ___out of 50 |
| Evaluator Remark (if Any): | Signature of the Evaluator with Date |

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 45 of 81 |

Exercise 7: Modular JavaScript and Working with JSON (Online Bookstore)

Scenario:

You are developing an online bookstore application that allows users to browse available books, add them to a shopping cart, and view or update their cart contents. To ensure clean, maintainable code, the application must use modular JavaScript with ES6 import/export, and handle book and cart data using JSON.

Tasks:

1. **Create a JSON data file or object** containing a list of books with the following properties:
 - o title
 - o author
 - o price
 - o availability (in stock or out of stock)
2. **Structure your code using JavaScript modules:**
 - o A module for displaying book listings.
 - o A module for managing the shopping cart (add, remove, calculate total).
 - o A module for updating the UI dynamically.
3. **Use ES6 module syntax** (import / export) to separate and reuse functionality across scripts.
4. **Implement interactivity using event listeners:**
 - o Add event listeners for "Add to Cart" buttons.
 - o Handle removal of items from the cart.
 - o Dynamically update the UI when the cart is modified.
5. **Display the cart contents and total price:**
 - o Show a summary of books in the cart.
 - o Allow users to remove items or proceed to a mock checkout view.

Expected Output:

- A dynamic web page displaying a list of books from JSON data.
- Modular JavaScript code using ES6 syntax to handle:
 - o Book listing
 - o Cart management
- Interactive functionality to add/remove books from the cart.
- Real-time cart content display with total price calculation.
- Screenshots showing:
 - o Initial book listing.
 - o Cart updates after adding/removing books.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 46 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 47 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 48 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|-----------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 49 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva:

1. What are ES6 modules and how are they used in a project?
2. How do you import and export functions or variables in JavaScript?
3. What is JSON and how is it used to represent book data?
4. How do event listeners manage cart operations in the UI?
5. What are the benefits of organizing code into modules?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 50 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|-----------------------------------|---|
| | Marks Secured: ___ out of 50 |
| Evaluator Remark (if Any): | Signature of the Evaluator with Date |

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 51 of 81 |

Exercise 8: To-Do List Application with React Hooks

Scenario:

You are building a productivity tool that helps users manage their daily tasks. Each task should be displayed with the option to mark it as completed or delete it. The UI should be clean and user-friendly, using Tailwind CSS for styling.

Tasks:

- Create a functional React application using `useState` to manage the task list.
- Allow users to add new tasks via a controlled form.
- Implement list rendering with unique keys to display tasks.
- Enable task completion toggling and deletion using event handlers.
- Use conditional rendering to show a message when no tasks are available.
- Implement all CRUD operations using task data stored in a local JSON file.

Expected Output:

- A React application with:
 - Task creation, display, completion toggle, and deletion features.
 - Conditional display of "No tasks found" message when the list is empty.
 - Tasks loaded from and updated into a local JSON file.
- Clean and responsive UI styled using Tailwind CSS.
- Screenshots demonstrating:
 - Adding a new task.
 - Task list with delete and complete options.
 - No-task view.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 52 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 53 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 54 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 55 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva :

1. What is useState and how is it used to manage tasks in a React application?
2. How do controlled components work in a React form, and why are they important?
3. What is conditional rendering in React and how would you implement it for an empty task list?
4. How can CRUD operations be performed using data from a local JSON file in a React app?
5. What are the advantages of using Tailwind CSS in building responsive user interfaces?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 56 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|-----------------------------------|---|
| | Marks Secured: ___out of 50 |
| Evaluator Remark (if Any): | Signature of the Evaluator with Date |

| | | |
|----------------|----------------------------------|-----------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 57 of 81 |

Exercise 9: Book Explorer with React Router and Props

Scenario:

You are developing a single-page application (SPA) for a book review website. The application should allow users to browse a list of books on the homepage and click on a book to navigate to a detail view. The detail page must display comprehensive information about the selected book using dynamic routing and React component props.

Tasks:

1. **Set up routing with React Router:**
 - Define two main routes:
 - / for displaying the book list.
 - /book/:id for viewing detailed information about a selected book.
2. **Create a parent component** to manage the list of books using useState.
3. **Use props** to pass book data from the parent to:
 - A reusable **BookCard** component (for the list view).
 - A **BookDetail** component (for the detail view).
4. **Implement dynamic routing:**
 - Use the useParams() hook from react-router-dom to retrieve the book ID from the URL and display corresponding details.
5. **Create reusable components:**
 - BookCard: Displays basic info (title, author) in the list.
 - BookDetail: Displays title, author, description, and rating.
6. **(Optional)** Simulate an API fetch using useEffect to load book data on initial render.

Expected Output:

- A functional React application with:
 - Homepage displaying a list of books.
 - A detail page for each book accessible via dynamic routing.
- Reusable components with data passed using props.
- Proper use of React Router (Routes, Route, useParams) and state management (useState).
- Screenshots showing:
 - The homepage with a list of books.
 - A selected book's detail page via route /book/:id.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 58 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 59 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 60 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva:

1. What is the purpose of React Router in a single-page application?
2. How do you use useParams() to retrieve route parameters?
3. How are props used to pass data between React components?
4. What is the difference between <Route> and <Routes> in React Router v6?
5. Why is component reusability important in React applications?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 61 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 62 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|----------------------------|---|
| Evaluator Remark (if Any): | Marks Secured: ____ out of 50 Signature of the Evaluator with Date |
|----------------------------|---|

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 63 of 81 |

Exercise 10: Feedback Collector with Redux Toolkit

Scenario:

You are developing a feedback collection tool for a training program. Users should be able to rate the session (on a scale from 1 to 5) and optionally leave comments. All submitted feedback entries should be displayed in real time below the form. The application must use **Redux Toolkit** for efficient and scalable state management.

Tasks:

1. Set up Redux Toolkit:

- Create a Redux slice (e.g., `feedbackSlice`) to manage feedback entries.
- Include actions and reducers for adding feedback.

2. Design the feedback form using controlled components:

- A dropdown or radio button group for selecting a rating (1 to 5).
- A textarea for optional comments.

3. Handle form submission:

- Dispatch an action to add the new feedback to the Redux state.
- Implement basic form validation (e.g., ensure a rating is selected before submission).

4. Display submitted feedback:

- Use a Redux selector (`useSelector`) to retrieve and display all feedback entries below the form.
- Each entry should show the rating and the comment (if provided).

5. Style the application:

- Use **Tailwind CSS** or **Material UI** for a clean and responsive layout.

Expected Output:

- A fully functional React + Redux Toolkit application that:
 - Allows users to submit a feedback rating and optional comment.
 - Validates required fields.
 - Stores feedback entries in Redux state.
 - Displays all feedback entries in a list below the form.
- Clean UI styled using Tailwind CSS or Material UI.
- Screenshots showing:
 - The form before and after submission.
 - The feedback list displaying submitted entries.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 64 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 65 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 66 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 67 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 68 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva:

1. What is Redux Toolkit and how is it different from traditional Redux?
2. What is a slice in Redux and what does it contain?
3. How does useSelector help in accessing state in components?
4. What are controlled components and how are they used in form inputs?
5. How do you dispatch an action in Redux Toolkit?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 69 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 70 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|-----------------------------------|---|
| | Marks Secured: ___ out of 50 |
| Evaluator Remark (if Any): | Signature of the Evaluator with Date |

| | | |
|----------------|----------------------------------|-----------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 71 of 81 |

Exercise 11: Full-Stack Feedback Form with MongoDB

Scenario:

You are building a feedback form where users can submit their name and comments. The form should store the feedback in a MongoDB database, and all submitted feedback should be displayed below the form in real-time.

Tasks:

1. Set up a MongoDB connection using Mongoose in a Next.js API route.
2. Create a form on the frontend (/feedback) to collect name and comment.
3. Use fetch() or axios to POST data to an API route (/api/feedback).
4. Fetch and display stored feedback using another API call.

Expected Output:

- Working form with validation and live display of feedback.
- API integration using MongoDB and Mongoose.
- Screenshots of form and feedback list.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 72 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 73 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva:

1. How does Mongoose connect to a MongoDB database?
2. What are API routes in Next.js and how are they structured?
3. How is data submitted from a form to a backend using fetch or axios?
4. How do you fetch and display data dynamically in a frontend?
5. What is the importance of form validation in full-stack applications?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 74 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|-----------------------------------|---|
| | Marks Secured: ___out of 50 |
| Evaluator Remark (if Any): | Signature of the Evaluator with Date |

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 75 of 81 |

Exercise 12: Authentication with NextAuth.js

Scenario:

You are implementing authentication for a members-only dashboard. The app must use NextAuth.js with GitHub or Google login. Authenticated users can see a welcome message and their profile information.

Tasks:

1. Install and configure **NextAuth.js** with either GitHub or Google OAuth.
2. Secure a route (/dashboard) so that it can only be accessed by authenticated users.
3. Display the user's name, email, and profile picture on the dashboard.
4. Implement a sign-in and sign-out button on the homepage.

Expected Output:

- Authentication working using NextAuth.js.
- Protected dashboard displaying user details.
- Screenshots of sign-in page and authenticated dashboard.

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 76 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 77 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 78 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

Viva:

1. What is NextAuth.js and how is it used for authentication?
2. How do OAuth providers like Google or GitHub integrate with NextAuth?
3. What is the purpose of protecting a route in an application?
4. How do you display authenticated user information using session data?
5. What is the difference between client-side and server-side authentication?

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 79 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | | |
|----------------|----------------------------------|---------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 80 of 81 |

FRONT END DEVELOPMENT FRAMEWORKS - WORKBOOK

| | |
|-----------------------------------|---|
| | Marks Secured: _____ out of 50 |
| Evaluator Remark (if Any): | Signature of the Evaluator with Date |

| | | |
|----------------|----------------------------------|-----------------------------|
| Course Title | Front End Development Frameworks | Academic Year : 2025 - 26 |
| Course Code(s) | 24SDCS01A/24SDCS01E | Page 81 of 81 |