24SDCS01R

FRONT END DEVELOPMENT FRAMEWORKS

RECORD

Front End Development Frameworks Team
KLEF (Deemed to be University), Bachupally, Hyderabad-500043

24SDCS01R

FRONT END DEVELOPMENT FRAMEWORKS

RECORD

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.

Department Vision and Mission

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.

Mission:

1. To provide high-quality, tool-based hands-on education that integrates theory with industry-driven learning.
2. To foster a research ecosystem that promotes innovation and practical applications in IT.
3. To equip students with leadership skills, entrepreneurship mindset, and interdisciplinary adaptability.
4. To instill ethical and responsible computing practices in emerging domains.
5. To prepare graduates to excel in the IT industry with technical expertise and continuous learning.

PROGRAM OUTCOMES		
PO	**Graduate Attributes**	**Program Outcome Description**
1	Engineering Knowledge	To impart mathematics, science, & engineering knowledgeto develop skills to solve complex engineering problems.
2	Problem Analysis	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	Design/ development of solutions	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.
4	Conduct investigations of complexproblems	An ability to 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.
5	Modern tool usage	Ability to create, select and apply appropriate techniques, resources and modern engineering activities, while understanding its limitations.
6	The engineer and society	Ability to apply reasoning and the contextual knowledge toassess social & health, safety, legal and cultural issues and the consequent responsibilities relevant to the professionalengineering practices.
7	Environment and sustainability	Ability to demonstrate the engineering knowledge to find solutions to contemporary issues by understanding their impact on societal and environmental contexts, towards sustainable development
8	Ethics	An ability to apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.
9	Individual and teamwork	To inculcate abilities to be able to act as a leader as well as team player effectively in multi-disciplinary settings
10	Communication	To develop oral and written communication skills to articulate the complex engineering activities with the engineering community and and society effectively through reports and design documentation, make effective presentations, and give and receive clear instructions.
11	Project management and finance	To develop working knowledge and understanding of the engineering and management principles to manage projects in multi-disciplinary environments.
12	Lifelong learning	To inculcate the habit of constant knowledge upgrading habit to meet the ever-changing technology and industry needs.
PROGRAM SPECIFIC OUTCOMES		
PSO1	An ability to design and develop software projects as well as to analyze and test user requirements.	
PSO2	Working knowledge on emerging technologies as per the industry requirements	

SYLLABUS:

CO-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, Working with .gitignore, 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.

CO-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, debugging and developer tools.

CO-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.

CO-4: Advanced React, Testing, and Backend with Node.js/Express

Context API for global state management, custom hooks for reusable logic, lazy loading and code splitting for performance, error boundaries for robust applications, unit testing with Jest and 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 (logging, parsing JSON, helmet for security), input validation and sanitization.

LAB EXPERIMENTS

EXP NO	DATE	EXPERIMENT NAME	PG.NO	Sign
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				

2025-26 ODD SEMESTER FEDF LAB CONTINUOUS EVALUATION

Lab No.	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						

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: 24SDCS01R
FRONT END DEVELOPMENT FRAMEWORKS

WEEK 1

Date of the Session: _____/_____/_____ **Time of The Session:**

Aim: To Project Initialization and First Commit

Description:

Tasks:

1. Install Git and configure your name and email.
2. Create a local Git repository using `git init`.
3. Create a basic `index.html` file that displays your name, title, and a short paragraph.
4. Track the file using `git add` and commit it using `git commit -m "Initial commit with HTML structure"`.
5. View the commit history using `git log`.

Programs:

OUTPUTS:

RESULTS:

VIVA QUESTIONS:

1. **How do you configure your username and email address in Git after installation?**
2. **What command is used to create a new local Git repository and what does it do?**
3. **Write a basic HTML structure that displays your name, title, and a short paragraph.**
4. **Explain the purpose of the git add and git commit -m "Initial commit with HTML structure" commands.**
5. **Which command is used to view the commit history in Git and what details does it show?**

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation**
	Marks Secured:_____out of _____ Signature of the Evaluator &Date of Evaluation:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: 24SDCS01R
FRONT END DEVELOPMENT FRAMEWORKS

WEEK 2

Date of the Session: _____/_____/_____ **Time of The Session:**

Aim: To Implement Branching and Feature Integration

Description:

Tasks:

1. Create and switch to a new branch named `about-feature`.
2. In `about-feature`, add an About section with headings and text.
3. Commit your changes with an appropriate message.
4. Create another branch `contact-feature` from main.
5. In `contact-feature`, add a contact form with name, email, and message fields.
6. Commit changes and merge both branches into the `main` branch one by one.
7. Handle merge conflicts (if any) and verify final content.

Programs:

Output:

RESULTS:

VIVA QUESTIONS:

1. How do you create and switch to a new branch in Git? Write the command?
2. Why do we use separate branches like about-feature and contact-feature in a project?
3. What Git command is used to merge a feature branch into the main branch, and what happens during a merge?
4. How would you resolve a merge conflict in Git? Explain briefly.
5. What steps do you follow to verify that both about-feature and contact-feature changes are successfully merged into the main branch?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation**
	Marks Secured:_____out of _____ Signature of the Evaluator &Date of Evaluation:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: 24SDCS01R
FRONT END DEVELOPMENT FRAMEWORKS

WEEK 3

Date of the Session: _____/_____/_____ **Time of The Session:**

Aim To Implement Responsive Web Page and Git History Exploration

Description:

Tasks:

1. Modify the existing HTML page to include:
   - A responsive image.
   - A layout that adapts to screen size using CSS media queries.
2. Add a CSS file (`style.css`) and link it to the HTML file.
3. Track changes using `git add`, `commit`, and use `git status` throughout.
4. Use `git log`, `git diff`, `git checkout`, and `git reset` to demonstrate version control and history exploration.

Programs:

OUTPUTS:

RESULTS:

VIVA QUESTIONS:

1. How can you verify that the responsive image you added to your HTML page displays correctly on different devices?
2. Explain how CSS media queries contribute to making a webpage responsive.
3. Apply a CSS rule from style.css that changes the background color of a specific when the screen width is less than 600 pixels.
4. Analyze the differences between git diff and git log commands in Git.
5. Demonstrate the use of git checkout and git reset commands to revert changes made to a specific file after staging but before committing.

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** Marks Secured:_____out of _____ Signature of the Evaluator &Date of Evaluation:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: 24SDCS01R
FRONT END DEVELOPMENT FRAMEWORKS

WEEK 4

Date of the Session: _____/_____/_____ **Time of The Session:**

Aim: To Implement HTML5 and CSS Positioning with Responsive Design

Description:

Tasks:

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

Programs:

OUTPUTS:

RESULTS:

VIVA QUESTIONS:

1. *What is the purpose of using semantic HTML5 elements like <header>, <nav>, <section>, and <footer> in a webpage?*
2. *How does the position: fixed property affect the navigation bar, and when should it be used?*
3. *What is the difference between position: relative and position: absolute in CSS layout design?*
4. *Which CSS property is used to create auto-playing slideshows using keyframes, and how does it work?*
5. *How do CSS media queries help in making a webpage responsive across various device screen sizes?*

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** Marks Secured:_____out of _____ Signature of the Evaluator &Date of Evaluation:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: 24SDCS01R
FRONT END DEVELOPMENT FRAMEWORKS

WEEK 5

Date of the Session: _____/_____/_____ **Time of The Session:**

Aim: To Implement JavaScript DOM Manipulation and Event Handling for Form Validation

Description:

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.

Programs:

OUTPUTS:

RESULTS:

VIVA QUESTIONS:

1. Which HTML5 attributes can be used to enforce validation rules directly within form input elements?
2. How would you use a regular expression in JavaScript to validate an email address format?
3. What JavaScript method is used to attach an event listener to an input field for real-time validation?
4. How can DOM manipulation be used to provide dynamic visual feedback for invalid inputs in a form?
5. What technique can be used in JavaScript to prevent form submission when some fields are invalid?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** Marks Secured:_____out of _____ Signature of the Evaluator &Date of Evaluation:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: 24SDCS01R
FRONT END DEVELOPMENT FRAMEWORKS

WEEK 6

Date of the Session: _____/_____/_____ **Time of The Session:**

Aim: ES6+ Features and Asynchronous JavaScript (Weather App)

Description:

Tasks:

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.

Programs:

OUTPUTS:

RESULTS:

VIVA QUESTIONS:

1. Which JavaScript method is used to make API calls to retrieve weather data, and how does it work?
2. What is the role of async/await in handling asynchronous API requests, and how is it different from using .then()?
3. How can you update the DOM to display weather details like temperature and description dynamically?
4. What is localStorage in JavaScript, and how can it be used to store the last searched city name?
5. How would you handle errors such as an invalid city name or a failed API request in a user-friendly way?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** Marks Secured:_____out of _____ Signature of the Evaluator &Date of Evaluation:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: 24SDCS01R
FRONT END DEVELOPMENT FRAMEWORKS

WEEK 7

Date of the Session: _____/_____/_____ **Time of The Session:**

Aim: Modular JavaScript and Working with JSON (Online Bookstore)

Description:

Tasks:

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

Programs:

OUTPUTS:

RESULTS:

VIVA QUESTIONS:

1. How is JSON data structured to represent a list of books with properties like title, author, price, and availability?
2. What are JavaScript ES6 modules, and how do import and export help in organizing code?
3. How would you use event listeners to handle "Add to Cart" functionality in a book listing application?
4. What is the role of a shopping cart module in a modular JavaScript app, and what key functions should it contain?
5. How can the DOM be dynamically updated to reflect changes in the cart, such as item removal or total price calculation?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** Marks Secured:_____out of _____ Signature of the Evaluator &Date of Evaluation:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: 24SDCS01R
FRONT END DEVELOPMENT FRAMEWORKS

WEEK 8

Date of the Session: _____/_____/_____ **Time of The Session:**

Aim: To-Do List Application with React Hooks

Description:

Tasks:

1. Create a functional React app with `useState` to manage the task list.
2. Allow users to add new tasks via a controlled form.
3. Implement list rendering to display tasks with keys.
4. Enable task completion toggling and deletion using event handlers.
5. Use conditional rendering to show a message when no tasks are present.
6. Implement all the crud operation using the data stored in a json file

Programs:

OUTPUTS:

RESULTS:

VIVA QUESTIONS:

1. How does the useState hook help in managing the state of a task list in a functional React component?
2. What is a controlled component in React, and how is it used in forms to add new tasks?
3. Why is it important to use a unique key prop when rendering a list of tasks in React?
4. How can event handlers be used to toggle task completion status or delete a task in a React component?
5. What is conditional rendering in React, and how can it be used to display a message when the task list is empty?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** Marks Secured:_____out of _____ Signature of the Evaluator &Date of Evaluation:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: 24SDCS01R
FRONT END DEVELOPMENT FRAMEWORKS

WEEK 9

Date of the Session: _____/_____/_____ **Time of The Session:**

Aim: Book Explorer with React Router and Props

Description:

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.

Programs:

OUTPUTS:

RESULTS:

VIVA QUESTIONS:

1. What is the purpose of using useParams() in React Router, and how does it help in dynamic routing?
2. How do you define and configure routes using react-router-dom for both list and detail views of books?
3. How can props be used to pass book data from a parent component to a child like BookCard or BookDetail?
4. What is the role of the useState and useEffect hooks in managing and loading book data in a React app?
5. Why should we use reusable components like BookCard and BookDetail in a React application, and what are the benefits?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** Marks Secured:_____out of _____ Signature of the Evaluator &Date of Evaluation:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COURSE CODE: 24SDCS01R
FRONT END DEVELOPMENT FRAMEWORKS

WEEK 10

Date of the Session: _____/_____/_____ **Time of The Session:**

Aim: To implement Feedback Collector with Redux Toolkit

Description:

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.

Programs:

OUTPUTS:

RESULTS:

VIVA QUESTIONS:

1. What is a Redux slice in Redux Toolkit, and how is it used to manage state like feedback entries?
2. How do controlled components help in managing form input fields such as rating and comments in React?
3. What is the purpose of the useDispatch hook in Redux, and how is it used during form submission?
4. How does the useSelector hook work, and how can it be used to display submitted feedback from the Redux store?
5. What are the benefits of using Tailwind CSS or Material UI when styling a React-Redux application?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** Marks Secured:_____out of _____ Signature of the Evaluator &Date of Evaluation: