Day 1: Web Design Concepts

Topics:

- Concepts of effective web design
- Web design issues: browser, bandwidth, and cache
- Display resolution
- Look and feel of the website
- Page layout and linking
- User-centric design
- Sitemap creation
- Planning and publishing websites
- Designing effective navigation

Lab Problems:

1. **Website Review:** Analyze the design of two websites (one fast-loading and one slow-loading). Identify key design issues affecting performance.
2. **Sitemap Design:** Create a sitemap and design the page layout for a five-page personal blog.
3. **Navigation Flow:** Build a basic HTML mock-up of an e-commerce website's homepage, including a navbar with links to "Home," "Products," "Contact," and "About Us."

Day 2: HTML Basics and XHTML

Topics:

- Basics of HTML: formatting, fonts, comments
- Color, hyperlinks, lists, tables, and images
- Forms
- XHTML concepts
- Meta tags, character entities
- Frames and frame sets
- Browser architecture and website structure
- HTML5 features

Lab Problems:

1. **Portfolio Webpage:** Create a simple HTML portfolio page with sections for About Me, Skills, and Contact. Use lists to display skills and hyperlinks to projects.
2. **Feedback Form:** Develop a feedback form with fields for name, email, and feedback message. Use meta tags to improve page SEO.
3. **HTML5 Features:** Create a webpage with an embedded video, audio player, and canvas drawing.

Day 3: CSS Basics

Topics:

- Need for CSS
- CSS syntax and structure
- Using CSS: background images, colors, and text properties
- Fonts, borders, margins, and padding
- Positioning with CSS
- Overview of CSS3 features
- Introduction to Bootstrap

Lab Problems:

1. **Styled Webpage:** Use CSS to style the portfolio page from Day 2. Apply background colors, custom fonts, and border styles.
2. **Responsive Image Gallery:** Create a photo gallery with CSS grid/flexbox. Ensure the gallery is responsive across different screen sizes.
3. **Bootstrap Navbar:** Build a basic webpage with a Bootstrap navbar and footer. Include links to Home, About, and Contact pages.

Day 4: JavaScript Basics and DOM Manipulation

Topics:

- Variables, conditions, loops, and functions
- Pop-up boxes
- JavaScript objects and DOM manipulation
- Event handling
- Writing client-side validation code

- Use jQuery for UI Interaction and Effects
- Use of Ajax and JSON

Lab Problems:

1. **Form Validation:** Add client-side JavaScript validation to the feedback form from Day 2. Validate email format and ensure all fields are filled.
2. **Interactive Webpage:** Create a webpage that changes the background color each time the user clicks a button.
3. **Simple Calculator:** Develop a calculator with buttons for addition, subtraction, multiplication, and division. Use JavaScript to handle operations and display the result dynamically.

Day 5: Introduction to REST API

Topics:

- Introduction to REST API
- HTTP requests (GET, POST, DELETE, PUT)
- API structure and usage

Lab Problems:

1. **Weather Application:** Build a webpage that retrieves weather information for a user-entered city using a public weather API (GET request).
2. **User List App:** Fetch and display a list of users from a public API (e.g., JSONPlaceholder). Include a button to refresh the user list.
3. **Simple Product Manager:** Develop a basic webpage where users can add products to a list (locally). Implement a POST request to simulate saving data through an API.

Day 6: Introduction to React JS & Environmental Setup

Session Topics:

- What is React JS?
- React with TypeScript
- React With JavaScript
- Introduction to Single Page Applications (SPA)
- DOM vs Virtual DOM
- Advantages and Disadvantages of React

- React Project Setup using create-react-app

Lab Tasks:

1. Set up a React environment using create-react-app.
2. Explore the project directory structure.
3. Create a basic **functional component** to display a welcome message.
4. Compare the working of Virtual DOM vs Real DOM by inspecting browser tools.

Day 7: React Basics – TSX, Props, and State

Session Topics:

- Introduction to TSX and its syntax
- Difference between functional and class components
- Passing data between components using **props**
- Introduction to **state** in React and useState hook

Lab Tasks:

1. Create a product card component that uses **props** to display product information (name, price, description).
2. Implement **state** management using the useState hook to create a counter.
3. Build a component that changes its content based on a state variable.
4. Differentiate between props and state usage with practical examples.

Day 8: Event Handling and Conditional Rendering

Session Topics:

- Understanding the React Event System
- Handling events in functional components
- Passing arguments to event handlers
- Conditional rendering using logical operators and ternary expressions

Lab Tasks:

1. Implement a **click event** to update a counter value dynamically.
2. Create a **toggle component** to show or hide content with a button.

3. Use **conditional rendering** to display messages based on input state.
4. Handle form events (like onChange and onSubmit) using event handlers.

Day 9: Lists, Keys, and Forms in React

Session Topics:

- Rendering lists using the map() method
- Understanding **Keys** in lists and their importance
- Controlled vs Uncontrolled Components in forms
- Handling form inputs and submissions

Lab Tasks:

1. Create a list of items (e.g., names) and render them using the map() function.
2. Assign unique keys to items to prevent rendering errors.
3. Build a **controlled form** that takes user input and displays it on submission.
4. Implement form validation to ensure required fields are filled.