

FULL STACK DEVELOPMENT LAB: 01

AIM: Develop responsive web design using HTML5 containing a team • style the pages using CSS . use of tag selected . class selector & id selector & use inline , internal & external \Rightarrow CSS . Apply Bootstrap CSS.

OBJECTIVES:

- 1) To understand HTML tags
- 2) To learn styling of web pages using CSS
- 3) To learn Bootstrap front end framework

THEORY:

- 1) **Responsive Web Design (RWD)**
 - It is an approach to web design that makes web pages render well on a variety of devices & windows & screen sizes . This layout adapts automatically to the screen size & orientation of platform , ensuring usability of website across desktop , tablets & smartphones.
 - Primary goal : provide an optimal viewing & interaction experience . easy reading & navigation with minimal resizing & scrolling . Across wide range of devices .
- 2) **Role of `<meta name="viewport">` tag** tells the browser how to render the page dimensions & scaling the page dimensions with width of web pages to match the visible width of initial zoom level .
 - It sets the width of initial zoom level .
 - It uses screen width width of mobile browser assume devices width . without this tag , mobile browser assume the essential for RWD .
 - Essential for RWD . without this tag , mobile browser assume a default viewport width & not scale to small screen size to appear zoomed out & not fit on screen .

- 3) **Bootstrap & Grid System**
 - bootstrap helps create responsive website that works on all devices .

- Uses a 12-column grid system where content is placed inside rows & col-element.
 - The grid adapt using breakpoint with prefixes:
 - col- → extra small (≤ 576 px)
 - col-sm → small (≥ 576 px)
 - col-md → medium (≥ 768 px)
 - col-lg → large (≥ 992 px)
 - col-xl → extra large (≥ 1200 px)
 - IV) Difference between Tag, Class & ID Selectors

Selection	Symbol	Purpose
→ Tag	none	Style all <P> tags
→ class	.	Style all elements with .given class
→ ID	#	Style one specific # element

 - v) Three ways to apply CSS: Element with and 1) inline CSS: written directly in HTML element style `<P style="color:blue">Hello</P>`
 - ii) internal CSS: written inside a `<Style>` tag in head `<Style>
color: green;
</Style>`
 - iii) External CSS: stored in a CSS file & linked in `<Link rel='stylesheet' href='Style.css'>`
- PROBLEMS STATEMENT: Simple Survey page (Roll no. 49 onwards)
- CONCLUSION: In this assignment, I learned how to build responsive web pages using HTML, CSS & Bootstrap. I understood the use of different CSS selectors with its 4 ways to apply CSS like the target selector, class selector, id selector and universal selector.

FULL STACK DEVELOPMENT LAB: 02

Page No. :

Date:

Aim: Develop a web application using Java script to implement sessions, cookies, DOM, perform validations such as checking for emptiness, only no. for phone no., special characters required - and for passwords, regular expression for certain format of other fields etc. used in mySQL database.

- Objective:
1. To understand what form validation is
 2. To learn basic functioning of DOM objects
 3. To learn how to apply various techniques to implement it.

Theory:

- 1) Explain the role of regular expression. why and they a suitable tool for validating data formats like a phone no. or checking for the presence of specific characters in a password?
- Regular Expressions are powerful tool for pattern matching so that they are a suitable tool for validating data formats like phone no. because they provide a concise & flexible way to define & check for specific sequences of characters. for eg., a regex can be written to ensure a phone no. has a certain no. of digits & an optional country code, following a specific structure. Similarly, the password strength, user can check for the presence of uppercase letters, no. of special characters.
- 2) Explain the fundamental difference between a session & a cookie in the context of web application development how do they work together to maintain user's logged-in state?
- Cookie: Stored on the client browser; holds small pieces of data i.e. session ID
 - Session: stored on the server; contains user-specific information together: cookie sends session ID to server with each request allowing the server to retrieve.

4 Provide a simple example of how a Java Script can interact with the DOM to dynamically change the content of a web page after user has submitted such as a form submission.

↳ Let's message "Hello! Up!"

<button onclick="document.getElementById('message').innerHTML = 'Hello! Up!'>

</button>

5. Give the steps for converting form input and using XMLHttpRequest to get JSON/JSONP from the user input.

• Create Input using JavaScript

• Create Input using fetch/AXHR

• Send data to backend using fetch/AXHR

• Backend convert to JSON via API

• Create SSL protocol

• Return results to frontend for display.

→ XSS

1. Three reasons why form validations are important.
 - Data security: Ensures only correct & properly formatted information is used.
 - Security: Prevent malicious inputs like SQL injection or script attacks.
 - User Experience: Gives instant feedback, helping user correct errors upon submission.

2. Example of how to modify an attribute value using DOM.

String id = "myImg";
String src = "old.jpg";

Script

document.getElementById("myImg").setAttribute("src", "new.jpg");
</script>

3. What are the different features of Java Script?

- Cross-platform: Works on all major browsers & devices
- Object-oriented & Reusable: runs directly on the browser & can be shared
- Event-driven & Dynamic
- Asynchronous Support: Handles APIs of server using callbacks & promises/async/await.

Page No. :
Date :

→ Problem Statement: Write a Client-side script with Taro Script to access & manipulates received object word object in an HTML webpage.

. Develop a dynamic web page using JavaScript & DOM manipulation

of the following

Conclusion: The web application successfully demonstrated TaroScript's sessions, cookies, DOM manipulation & JSON integration. Enabling secure & accurate data entry through effective form validation & regular expressions for enhanced functionality & user experience.

My Taro

FULL STACK DEVELOPMENT LAB: 03

Page No.:

Date :

Aim: Design an interactive front end application using react by implementing templating using components, states & Props, class & es6. It must be responsive to scale across different platforms.

Objectives: To develop a responsive, interactive front-end application using react.js that effectively demonstrates the fundamental concepts of component-based architecture, state management & event handling. The application will serve as a practical exercise in building a scalable user interface by implementing templating with components, managing dynamic data with states & props & handling user interactions with events, ensuring a seamless user experience across various devices & screen sizes.

Theory:

1. Explain the role of state & props in React. How do they differ? What is the primary purpose of each in managing data flow within a component-based application? props & state manage data flow in react. props are immutable & passed from parent to child, enabling component reusability. State is mutable, maintained within a component, handling dynamic, interactive data. Props configure component, while state triggers rendering. Together they ensure interactivity of data flow, making applications predictable, modular & user interactive.

2. what is a React component? Differentiate between a class Component & a functional Component & discuss the advantages of using a functional component with hooks like useState & useEffect over a class component.

of React Component is a reusable UI block. class component use this.state.setState(), & lifecycle methods, while functional component rely on hooks like useState & useEffect. functional component has preference due to less bloaty code, easier readability, better performance & Enhanced Reusability, making them more efficient for modern react development compared to class components.

3. Describe the concept of templating using Components in react. why is this approach considered superior to traditional web development, methods that rely on monolithic HTML file.
Templating using Components breaks UI into reusable parts like <Header>, <Footer>, <Button>, unlike monolithic HTML. This approach improves maintainability, reusability & dynamic rendering, making applications modular, easier to manage & more scorable than traditional static web development practices.
4. How do you handle user events in react? Review a sample code snippet to demonstrate how an event handler is defined in a component & how it can be used to update the component's state
Events in react use event handlers: functions update state on actions like clicks & onClick=handleClick calls setCount (count +1). State changes re-render UI, ensuring interactive application. event handling in react maintains clean, declarative & responsive component structure.
5. what is responsive web design & why is it crucial for modern applications? describe how you would implement a responsive design in a react application using CSS media queries or a CSS in JS library.
Responsive web design adapts the user's device, ensuring accessibility & user experience. In react, it's implemented using CSS media queries or CSS in JS libraries like styled-components, adjusting layouts, adjusting colors automatically for mobile, tablets & desktops.

Problem statement: Create a responsive, interactive photo gallery application using React.

Conclusion: This assignment contrasts with a responsive and effective user interface component, states, props, & event handling, showcasing scrollable interactive front-end design principles with effective user experience across platforms.

Conclusion: This assignment concluded with a responsive and adaptive alternative component, states, props, & event handling, showcasing scrollable interactive front-end design principles with effective user experience across platforms.

Problem statement: Create a responsive, interactive photo viewer application using React.

Fall

FULL STACK DEVELOPMENT LAB : 04

Aim: Enhance web page developed in earlier assignment by rendering lists & portals, Error Handling, routers & style with React CSS also make it a responsive design to scale well across PC, tablet & mobile devices.

Objectives :

- Enhance User Interface & Experience
- Improve Application Robustness & Navigation

Theory:

1. How do lists & keys work in React?
 - List renders multiple items dynamically using .map()
 - Keys are unique identifiers for list items, helping react efficiently update & re-renders only changed elements
 - Without keys, key performance drops & unexpected bugs may occur.
2. What is a React Portal & when would you use one?
 - A react portal allows rendering a child component into a root node outside its parent hierarchy
 - Used for modals, tooltips, bottom & cover layers where you need separation from parent styles & positioning.
3. Discuss the importance of Error boundaries in React.
 - Error boundaries catch errors in components during rendering, lifecycle methods, & constructors.
 - Prevents the entire app from crashing by displaying fall back UI.
 - Crucial for improving user experience & debugging in production.
4. How does react router enable SPA functionality?
 - React Router maps URLs to components without full-page reloads.
 - Uses client-side routing with history API for smooth navigation.
 - Provides routes, Link, & Switch > for examples of transition

5. Explain the different ways to style a React application.
- CSS Stylesheets → Import Standard CSS files
 - Babel styles → use style = JS with JavaScript objects.
 - CSS Modules → Scoped CSS with automatic class name mapping
 - Styled-Component (CSS in JS) → write CSS inside TS using `styled` template.
 - Figma works → Utilize figma CSS framework for rapid UI design.

Problem Statement: Create a dynamic user dashboard with multiple sections that can be navigated using tabs.

Conclusion: This assignment concluded with an Enhanced React application using Babel, CSS modules, error handling, routes & responsive CSS, delivering a robust, user-friendly & scable interface across PC, tablet & mobile devices.

J

✓ 07/11/23

FULL STACK DEVELOPMENT LAB : 05

Aim: Develop a responsive web design using Express framework to perform CRUD operations of deploy with node JS & MongoDB DB.

- Objectives:
 - Develop a full-stack web application
 - Demonstrate Backend Development & Deployment Efficiency

Theory:

1. What is the role of Express.js as a web framework for Node.js?
 - Express.js is a lightweight, flexible web framework built on Node.js.
 - It simplifies handling HTTP requests, responses & routing.
 - Provides middleware support for authentication, sessions & error handling.
 - Sets up API & web app development with minimal boilerplate code.

2. Explain the concept of CRUD operations in context of a web application.
 - CRUD stands for Create, Read, Update & Delete.
 - Create → Add new records
 - Read → Fetch/display data
 - Update → modify existing data
 - Delete → remove data

• CRUD ensures complete life cycle management of application data.
3. Why is MongoDB a suitable choice for this project?
 - MongoDB is a NoSQL, document-oriented database.
 - Stores data in flexible JSON-like documents, making it ideal for dynamic applications.
 - Scales easily for large datasets & distributed systems
 - Integrates seamlessly with Node.js & Express using Mongoose.
 - Schema flexibility, supports rapid development & iteration.

4. what steps are involved in deploying a microservices application?

1. Prepare application
2. Choose testing platform
3. Setup servers
4. Upload code
5. Install dependencies
6. Configure Process Manager
7. Reverse Proxy
8. Monitor & Scale

Problem statement: A recipe sharing platform

Conclusion: This assignment concludes with a full-stack web application using Spring Boot's MVC, successfully performing all operations of deployment, ensuring responsive design, backend proficiency & seamless functionality access services.

My
Work