Date: __/__/____

* Assignment 1

* Aim: Develop responsive web design using HTML5 containing a form. Style the pages using CSS, use of tag selector, class selector and id selectors. Use Inline, Internal and External CSS, Apply Bootstrap CSS.

* Objectives:
1. To understand HTML tags.
2. To learn the styling of web pages using CSS.
3. To learn Bootstrap Front End Frame work.

* Theory:
1. Define Responsive Web Design (RWD). What is its primary goal?

→ RWD is a design technique where websites automatically adjust that layout, images and elements to fit the screen size like desktops to mobile phones. Its used to provide a smooth, readable and user-friendly experience across the devices without needing seperate website for each.

Date: __/__/____

2. Explain the role of the <meta name="viewport" ...> tag. Why is this tag essential for RWD?

→ • The viewport meta tag tells the browser how to control the page's dimensions & scaling.

• It's essential for RWD because it:

→ sets the page width equal to the device's screen width.

→ Helps have proper zooming and scaling.

→ Ensures media queries and layouts work as intended on mobile devices.

3. How does Bootstrap assist in creating a responsive layout? Discuss the concept of a grid system and how it adapts to different scree sizes.

→ • Bootstrap is a front end framework that involves a responsive 12-column grid system.

• Uses predefined classes like - col-md-6, col-sm-4, etc.

• col-lg-3 to define layout per screen size.

• Automatically rearranges columns or stacks things vertically on smaller devices.

Date: __/__/____

4. Differentiate between Tag, Class, and ID selectors.

→ • Tag selector: Targets all elements of a certain HTML tag, ex: p { color : blue; }.

• Class selector: Targets elements with a specific class attribute, ex: highlight { background: yellow; }.

• Id selector: Targets one unique ~~element~~ id attribute and can be applied to multiple elements & has to be unique. ex: #header { font-size : 20px ; }

5. Describe the three main ways to apply CSS to an HTML document:

→ 1. Inline CSS: Applies style directly to individual ~~HTM~~ HTML elements using the style attribute. The CSS rules are written in the opening tag of the HTML element.

ex: Example

2. Internal CSS: Defines CSS rules within <style> placed in the <head> section, it applies to the entire HTML document in which they are embedded.

```
ex: <html>
    <head>
    <style>
        body { background-color: gray; }
        h1 { color: green; }
    </style>
    </head>
    :
    </html>
```

#learnthesmarterway

Date: __/__/____

3. External CSS: Involves creating a seperate .CSS file containing all CSS rules & then linking to the HTML doc using <link> in <head>.

ex:
```
<html>
<head>
<link rel='stylesheet' href='styles.css>
</head>
```

style.css: `P {color:red; font-family: sans-serif;}`

* Conclusion:

→ In this assignment we built responsive web pages using HTML. We used different CSS selectors, the ways to apply CSS and how the viewport tag with bootstraps grid system ensures layout adapted to all screen sizes.

22/8/25

#learnthesmarterway

Date: __/__/____

X FSD-3

• Aim: Design an interactive front end application using React by implementing templating using components, States & Props, Class, Events. 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, and 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 and props, and handling user interactions with events, ensuring a seamless user experience across various devices and screen sizes.

• Theory:

1. Explain State & Props in React. How do they differ, and what is the primary purpose of each in managing data flow within a component-based application.

→ • Props (Properties): Props are read-only attributes passed from a parent component to its child components. Their primary purpose is to enable data flow

Date: __/__/____

from parent to child, allowing customization & reusability of components. Props are immutable within the child component.

- State: State is an internal, mutable object within a component that manages data specific to that component. It allows components to manage and update their internal data dynamically, triggering re-enders of the component when the state changes.
- Difference: Props facilitate undirectional data flow from parent to child and are immutable, while state manages internal component data & is mutable.

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

→ • React component: Its an independent, reusable piece of UI that encapsulates its own logic & rendering. It can be thought of as a building block for complex user interfaces.

- A Class component is a JavaScript class that extends React. component and involves a

#learnthesmarterway

Date: __/__/____

render() method. It can hold state & lifecycle methods. While functional component is a javascript function that accepts props as argument & returns JSX.

- Advantage of Functional Components with Hooks:
These are used, for ex: UseState for state management and useEffect for side effects are preffered due to:

→ Simplicity & Readability: They are more concise & easier to understand.

→ No 'this' binding issues which avoids complexities in its class components.

→ Hooks allow for logical grouping of related stateful logic.

8. Describe 'templating using components'.

→ 'Templating using components' in React involves breaking down the UI into smaller, self-contained components, each responsible for rendering a specific part of the interface. These components are then together to build the complete application.

→ Superiority to Traditional Methods:

- Modularity & Reusability: Components are independent & can be reused across the code, avoids code duplicacy.
- Maintainability: Change in 1 component are less likely to affect others.

Date: __/__/____

- Collaboration: Teams can work on parts simultaneously.
- Enhanced performance: React's Virtual DOM efficiently updates only the necessary parts of the UI, leading to better performance compared to re-rendering monolithic HTML files.

4. How do you handle user ~~reacts~~ events in React?

→ In React, user events like button clicks are handled by attaching event handler functions directly to JSX elements like camelCase attributes.

→ Code:

```
import React, {useState} from 'react';
function Counter() {
    const [count, setCount] = useState(0);
    const handleClick = () => {
        setCount(count+1); };
    return(
      <div>
          <p> Count: {count} </p>
          <button onClick={handleClick}>
          Increment </button>
      </div>
    ); }
export default Counter;
```

#learnthesmarterway

Date: __/__/____

5. What is responsive web design & why is it crucial for modern applications? Describe how you would implement a responsive design.

→ Responsive web design is an approach to web development that creates websites & applications that adapt and display optimally across a wide range of devices. It is crucial for modern applications because it enhances user experience and betters SEO rankings.

→ To implement responsive design in React:

1. CSS Media Queries: We can define different styles.

⇒ .container { padding: 20px; }

@media (max-width: 768 px

2. CSS-in-JS libraries: Libraries like Styled Components or Emotion allow you to write CSS directly within JavaScript components, offering dynamic styling and easier management of responsive styles.

⇒ import React from 'react';

import styled from 'styled-components';

const ResponsiveContainer = 'styled.div'

padding: 20px;

Date: __/__/____

```
@media (max-width: 768 px) {
    padding : 10 px;
    font-size: 14px;
}
`;

function App() {
  return (
      <Responsive Container>
        <h1> Welcome! </h1>
        <p> adjustable content </p>
      </Responsive Container>
  );
}

export default App;
```

(8)
26/4/26

Date: __/__/____

* FSD - 05

* Aim: Develop a responsive web design using Express Framework to perform CRUD operations and deploy with Node JS use MongoDB.

* Objectives:
  - Develop a Full-Stack Web Application.
  - Demonstrate Backend Development and Deployment Proficiency.

* Theory:

1. What is the role of Express.js as a web framework for Node.js?

→ Express.js is a minimal and flexible web application framework for Node.js. It simplifies the development of server-side applications by providing a robust set of features for web & mobile applications. Key roles:

- Routing: Handling HTTP requests & mapping them to specific functions based on URL & HTTP method.
- Middleware: Providing a mechanism for processing requests before they reach route handler.
- HTTP Utility Methods: Offering helper functions for handling HTTP requests & responses.

#learnthesmarterway

Date: __/__/____

- Simplifying API Development: Making it easier to create RESTful APIs and backend services with Node.js.

2. Explain the CRUD operations in the concepts of a web application.

→ CRUD is an acronym representation the 4 fundamentals operations in persistent storage (eg databases) and data management within a web application:

- CREATE (POST): Adding new data records to the system. In a web application, this typically involves sending a POST request to an API endpoint with the new data in the request body.
- READ (GET): Retrieving existing data records from the system. This usually involves sending a GET request to an API endpoint to fetch specific data or a collection of data.
- Update (PUT/PATCH): Modifying existing data records. This is done by sending a PUT/PATCH request to an API endpoint with the data.
- Delete (DELETE): Removing data records from

Date: __/__/____

the system. This operation is done by sending a DELETE request to an API endpoint targeting the specific data to be removed.

3. Why is MongoDB a suitable choice for this project?

→ MongoDB is a NoSQL document-oriented database that offers several advantages:

- Flexibility & Scalability: Its document-based structure allows for flexible schema design & easy horizontal scaling to handle large amounts of data & traffic.
- JSON-like Data Format: Documents are stored in BSON format, which aligns well with JavaScript & Node.js.
- High Performance: Designed for high performance and availability, making it efficient for handling read & write operations.
- Rich Query Language: Provides a powerful & expressive query language for retrieving and manipulating data.

4. What steps are involved in deploying a Node.js and Express application?

→ Deploying a Node.js & Express application has these steps:

Date: __/__/____

→ • Prepare the Application: Ensure all dependencies are installed, environment variables & app is configured.
• Choose a Hosting Provider: Select a suitable hosting platform thats supported by Node.js.
• Set up the environment: Configure the server environment for keeping the application running.
• Database setup & connection.
• Deploy the code: Transfer the application code to the server, often using Git for version control and deployment.
• Start the application and ensure it listens on the correct port and is accessible to the public.
• Configure Domain & SSL certificates for secure communications (HTTPS).
• Monitoring & Maintenance: Implement ~~monito~~ monitoring tools to track application performance and ensure ongoing stability and security.

26/9/24

Date: __/__/____

* FSD - 5

* Aim: Enhance web page developed in earlier assignment by rendering Lists and Portals, Error Handling, Routers & style with React CSS also make it a responsive design to scale well across PC, tablet and Mobile Phone.

* Objectives: · Enhance User Interface & Experience
· Improve Application Robustness & Navigation.

* Theory:

1- How do Lists & keys work in React?

→ React lists are a way to render multiple components or elements based on an array of data. When rendering lists, a unique key prop must be provided to each list item. Keys help React identify which items have changed, are added or are removed, enabling efficient updates to DOM. Without keys, React might struggle to correctly update the list, leading of performance issues & potential bugs, especially when order of items changes/adds/removes.

#learnthesmarterway

Date: __/__/____

The key should be a stable, unique identifier for each item within the list, such as an ID from a database.

2. What is a React Portal & when would you use one?

→ React Portal has a way to render children into a DOM node that exists outside the DOM hierarchy of the parent component. This is achieved using ReactDOM.createPortal(child, container). While the portal's content is rendered elsewhere in the DOM. It still maintains its position within the React component tree, allowing it to receive props & context from its parent.

→ Use Cases:

- Modals, Tooltips, Dropdowns
- Notifications or Toasts.

3. Discuss the importance of Error Boundaries in React.

→ Error boundaries are React components that catch JavaScript errors anywhere in their child component tree, log those errors, and display a fallback UI instead of crashing the entire applications. They prevent errors in one part of

#learnthesmarterway

Date: __/__/____

the UI from breaking the entire applications.
Error boundaries are crucial for:
- Improved user experience
- Debugging & Error Reporting.
- Preventing Cascading Failures.

4. How does React Router Enable Single Page Application (SPA) functionality?

→ • URL Management: It allows you to define routes that map to different components & it manages the browser's URL without requiring full page reloads.
• Component Based Routing: You define routes using components like BrowserRouter, Routes & Route making routing declerative & integrated with your structure.
• Navigation without Reloads: Components like 'Links' are used instead of traditional <a> tags. When a Link is clicked, React Router helps update the URL without a full page refresh, bettering user experience.
• History API Integration: It uses browser's History API to manage navigation history.

5. Explain the different ways to style a React application.

#learnthesmarterway

Date: __/__/____

→ • Inline Styles: style prop

⇒ {div style = {{ color: 'blue', fontSize: '16px' }}}>Hello

• CSS style sheets: external CSS files

⇒ • my-text { color: green; }

import './MyComponent.css';

function MyComponent () { - - - ; }

• CSS Modules: CSS stylesheets

⇒ • myText { color: purple; }

import styles from './MyComponent.module.css';

function MyComponent

{ return

Styled Text ; }

• Utility-First CSS Frameworks: Pre defined utility classes directly in RSX:

⇒

Styled Text

10/10/24

#learnthesmarterway