

Assignment - 01

AIM: Develop responsive web design using HTML5, containing a form. Style the pages using CSS, Use of tag selector, class, id selector. 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 the Bootstrap front end framework.

Theory:

Q) Define responsive Web design (RWD). What is its primary goal?

Ans RWD is an approach to web design that ensures websites look and function well on a variety of devices and screen sizes, including desktops, tablets and smartphones.

Primary goal: The main goal of RWD is to provide an optimal user experience across all devices by automatically adjusting layout, images, and content based on the screen resolution and orientation.

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

Ans: It tells the browser how to control the page's dimensions and scaling on different devices.

eg. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

width=device-width: Sets the viewport width to the width of the user's device.
initial-scale=1.0: Sets the initial zoom level when the page is first loaded.

Why is it essential?

Without this tag, mobile browsers assume the desktop width (usually 980 px) and shrink the content to fit the screen. This makes text too small, layouts broken, and navigation difficult on small screens. It is crucial for responsive design to ensure content scales correctly on different devices.

Q) How does Bootstrap assist in creating a responsive layout? Discuss the concept of a grid system briefly. Ans. b. How is it adapted to different screen sizes

Ans. Bootstrap is a powerful, open-source CSS framework that simplifies the process of creating responsive and mobile-first websites.

How - Bootstrap helps:

- Includes a responsive grid system, pre-styled components, and utility classes.
- Provides ready-to-use styles for buttons, forms, navigation bars, and more.

Reduces the amount of custom CSS and speeds up development work (MTM)

Bootstrap grid system = 12 units and 12 columns.

Divides the page into 12 columns.

Used empty columns to align content.

Columns can span 1-12 units depending

How it adapts: . On smaller screens, columns stack vertically.

On larger screens, they are placed side by side.

You can define diff. column sizes for diff. devices to control layout behaviour.

Q) Differentiate b/w Tag, Class, ID selectors?

Ans Tag selector → tag name() - select all elements of a certain tag and style all <P> tags

Class selector → classname() - select all elements of a specific class - style multiple elements similarly.

ID selector → #idname() - Select one unique element with that id - style a single specific element.

Note: "Class" tag can be used across multiple elements.

- IDs must be unique within a page.
- Tags affect all elements of that type.

Page No.:
Date:

Page No.:
Date:

(b) Describes the 3 main ways to apply CSS to an HTML document.

Ans: Indian CSS.

Western dialect in the style attributed
of James J. THURBER.

<pi style="color: green;">>This is green

2. Internal CSS will define styles inside a <style> tag in the head section of the HTML file.

6.

front single: 30 pcp; middle cap

~~Handwritten copy of a letter from Dr. John C. Calhoun to Dr. James Madison, dated April 1, 1816.~~

~~(Handy - 2nd) simple~~

3. External CSS

such as `Depositor`, `Customer` and `Supplier`

Q. Cloody
A. Linkwood: "Style shoot" has "my focus"

~~1/10/2009~~ ~~1/10/2009~~

10-394

and the same

12

四

Assignment -02

Aim: Develop a web application using JavaScript to implement validations such as checking for emptiness, only no. for phone no., special character usage, for password, regular expressions for certain formats of the fields etc. Use the MySQL database.

- Objectives:**
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 they like a phone no. for checking for the presence of specific char. in a password? Ans. Regular expression (RegEx) and patterns used to match characters combinations in strings.

- Phone no. → must be digit only & 10 digits.
- Email → must contain @ and . in a proper format.
- Password → must contain uppercase, numbers, and special characters.

Regex ensures data consistency, accuracy and reduces errors before sending to the server.

- 2) Explain the fundamental diff. b/w a session and a cookie and the context of web app. dev.

How do they work together to maintain a user's logged-in state?

Ans. Cookie → Stored in the client's browser. Used for small data like preferences, remember me login, file upload.

Session → Stored in the server, linked to a unique session ID, saved in a cookie.

How they work together?

User logs in → server creates a session. Browser stores a cookie with session ID. Each request sends this ID → server identifies user identity.

- 3) What is the purpose of performing both client-side and server-side validation? Describe a scenario where relying solely on client-side validation could lead to a security vulnerability?

Ans. Client-side (JavaScript) → Immediate feedback. Server-side (PHP, Node.js, MySQL) : Final layer of security.

Scenario (risk if only client-side used):

- 1) Only client-side validation is used, an attacker can disable JavaScript and submit malicious input (SQL injection, invalid data).

server-side onwards data integrity and prevents hacking.

- 4) Provide a simple ex. of how a JavaScript script can interact with the DOM to dynamically change the content of a web page after a user action, such as a form submission.

<button onclick="changeText()"> Click Me button

<p id="demo">Original Text</p>

```
<script>
function changeText() {
    document.getElementById("demo").innerHTML = "Text changed!";
}
</script>
```

↳ If we click on the button, it will change the content of the paragraph element with id="demo".

↳ In this example, we have used innerHTML to change the content of the element.

FAQ:

1. What is ~~incorrect~~ why form validations are important?
- Ans: Prevents invalid or malicious data entry. Enhances user experience by reducing load.

2. Give an ex. of how to modify an attribute value of an element using DOM?

document.getElementById("myImage").src = "image.jpg";

3. What are the diff. features of JavaScript?

↳ lightweight - interpreted language
↳ supports DOM manipulation
↳ event - driven & asynchronous program.
↳ platform-independent, cross-platform
↳ object-oriented features with prototypes.

~~Ques 1. Frontend (HTML | CSS | JS) - collect user data.~~

1. A JAX / REST API - send data to backend.
2. Backend (PHP / Node.js / Java / JSP) - process request.
3. Database (MySQL) - store / retrieve data using SQL queries.
4. Response back end sends result back from database to user.

Assignment -03

Page No. :
Date :

AIM: Design an interactive front end app using React by implementing simplifying using component based approach. It must be responsive for web users diff. platforms.

Objectives: To do a responsive interactive front end app by using React's built-in concept of component-based architecture, state management, and event handling. The app will have a practical exercise in building a scalable user interface by imp. React along with building component managing dynamic data with states, props, and handling user interactions with events, context, a zoomable user comp. across various devices & screen sizes.

Theory: 1. Explain the role of state and props in React. How do they differ and what is the primary purpose of each managing data flow within a component-based app.

Ans: State is a mutable data structure local to a comp. It holds dynamic data that can change over time within the comp., allow the UI to respond to user interactions or other events. props: Immutable data passed from a parent comp. to a child comp. to configure or provide data to that child.

Primary purpose: State manages internal comp. data that can change while it helps deliver external data from parent to child to maintain unidirectional data flow.

Q2 What is React component? Diff b/w a class component and functional and discuss the adv. of using a functional comp. with hooks like useState and useEffect over a class comp.

Ans: A React component is a reusable, independent piece of UI defined as either a JS Class or function.

Class Comp. and ES6 class extending React Component, featuring lifecycle methods and state via this.state and this.setstate.

Functional Comp. are plain JS functions returning JSX. With React hooks like useState and useEffect, with React component can handle local state and lifecycle events.

Adv. of function comp. with hooks:

- Simpler syntax and easier to read.
- No need for this. Returns see binding hooks.
- Better code reusability via custom hooks.
- More lightweight with potentially better performance.
- Unified handling of life cycle effects via useState, reducing complexity of multiple生命周期 methods in classes.

Q3 Discuss the concept of "template" using "comp." in React. Why is this approach considered superior to traditional web dev. methods particularly on monolithic HTML files?

Ans: It means breaking down UI into small, reusable, self-contained components.

5. Responsive Web design and Imp. in React.

Responsive web design ensures that web app render well across a variety of devices. It is crucial for accessibility and credibility of an application accessed via desktop, tablet and smartphones.

In React, responsive design can be implemented using:

- CSS media queries in external or modular CSS files to apply different styles depending on media queries inside component styles.

Ex: Responsive web design ensures that web app render well across a variety of devices. It is crucial for accessibility and credibility of an application accessed via desktop, tablet and smartphones.

4. Handling User Events in React:

- **Key:** React handle user event like click, passing event handler function as props to elements.
- Event handlers are defined in comp. and complete state, covering UI as demanded.

Code:

```
import React, { useState } from 'react';
function ClickCounter() {
  const [count, setCount] = useState(0);
  const handleclick = () => {
    setCount(count + 1);
  };
  return (
    <div>
      <p> You clicked {count} times</p>
      <button onClick={handleclick}> Click me</button>
    </div>
  );
}
```

Component definition Click Counter;

Assignment -04

Page No.:
Date:

Q3 Discuss the importance of outer boundaries in React.

Ans: Dynamic web page developed in e-commerce platform by combining list and portfolio. Hence React and style with each CSS adapt make it a responsive design to work well across PC, tablet and mobile phone.

Objectives:- Enhance Usability, Interactivity & experience Interface & experience Improve Application and Navigation.

Q4 How does React Router enable single page Application (SPA) functionality?

Theory:
1. How do list and keys work in React

Ans. In React lists are rendered using array methods like map() each item in the array is transformed into a react element.

Keys provide unique attributes that help React identify which items in the list have changed, added or removed.

Without keys → React re-renders inefficiently.
With unique keys → React updated only the modified elements, implementing performance.

Q2 What is a React portal and when would we use?

Ans: Portal: A feature that allows rendering children into a DOM node outside of the parent component's hierarchy.

Use case:- Models, tool tips, popups, dropdowns. They help in avoiding CSS conflicts and ensure proper layering.

Page No.:
Date:

eg
<h1 style = { color: "blue", font-size: "20px" }>Hello</h1>

Assignment 04Q4

Ans How does it work =

- It listens for changes in the browser's URL path.
- Based on the current path, router dynamically renders the appropriate component without reloading the page.
- Routes are defined to associate URL paths with exact components.

Benefit for SPA :-

- Fast Navigation - Because page reloads are avoided, navigation is quick and smooth.
- Reserves application state - Since the app isn't fully reloaded, global state or component state can persist across navigation.
- Better user exp - The app feels like a native app with instant page changes.
- SEO and deep linking - URLs reflect the current page, so users can bookmark or share links.

eg -

<Browser Router>

<Routes>

<Route path="/" element={<Homepage/>} />

<Route path="/about" element={<AboutPage/>} />

</Routes>

</Browser Router>

By
10/10/24

10/10/24

Fast

Rea

Flexi

an

Fa

Scal

d

Rid

Assignment-05

Page No.:
Date:

Page No.:
Date:

AIM: Develop a responsive web design using Express framework to perform CRUD operations and deploy with Node.js. Use MongoDB.

Objective: Develop a Full-Stack Web App. Demonstrate Backend Development and Deployment proficiency.

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

Ans: Express.js is a minimal and flexible web framework for Node.js. Provides tools for building web servers and APIs. Handles routing of middleware segment, response handling and integrates easily with databases. Makes backend development faster and structured compared to plain Node.js.

2. Explain the concept of CRUD operations in the context of a web application?

Ans:
CREATE (POST) → Add New data (e.g. product, user)
Read (GET) → Retrieve data (all items or specific item).
Update (PUT/PATCH) → Modify existing data. (edit post)
Delete (DELETE) → Remove data permanently.

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

Ans: NoSQL database → stores data as flexible

JSON - like document:
• Easy to integrate with Node.js via Mongoose
• Use native driver.
• Good for projects with unstructured or evolving data models.
• Building scalability and high performance.
• Supports cloud deployment easily.
Q1: What steps are involved in deploying a Node.js based express application?

Ans 1. Prepare Application:

• Use environment variables for DB connection (local)

• Build Express Server, connect to MongoDB.

1.1.2: Choose hosting platform
eg: Render, Railway, Heroku, Vercel, etc. MigrateDB.

3. Setup MongoDB

i. Use MongoDB atlas (cloud database) or install locally on server.

4. Deploy application

• Push project code (gitHub)
• Connect hosting platform with gitHub Repo.
• Install dependencies (npm install)

5. Configure environment variables
in hosting service.

6. Test live app

• Verify endpoints with Postman/Browser.

Assignment - 05

Q3

Reasons why MongoDB is a good fit:

- Flexible schema: MongoDB stores data as JSON-like BSON documents, which allow dynamic and flexible schemas; this is helpful during development as the data model can evolve without complex migration.
- Easy integration with Node.js: MongoDB pairs well with Node.js because both use JS / JSON formats, making data handling seamless b/w the backend and database.
- Scalability: MongoDB is designed to scale horizontally, handling large volumes of data and high traffic, which benefits growing applications.
- Rich querying: Supports complex queries, indexing, and aggregation pipelines, enabling efficient data retrieval and manipulation.
- Fast prototyping and strong community and ecosystem.