

KUMAR UJJAWAL

PRN-1032232774

R.NO-56

TY CSE

*** AIM:-**

→ To develop responsive web design using HTML containing a form.
Style pages using CSS, use tag selector, class selector & id selector,
use inline, internal & external CSS. Apply bootstrap CSS.

*** OBJECTIVES:-** ① Understand HTML tags.

② Learn styling of web page using CSS.

③ Learn bootstrap front end framework.

*** THEORY:-**

① Define Responsive Web Design (RWD). What is its primary goal.

Ans → It is a web design approach that ensures website looks and functions well on all devices. Its primary goal is to create a flexible, user-friendly interface that automatically adapts layout, images and other elements based on device's screen size.

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

Ans → This tag tells browser how to control page dimensions & scaling
 $\text{width} = \text{device-width}$ → width matches ^{device} screen width.

$\text{initial-scale} = 1.0 \rightarrow \text{sets initial zoom} = 100\%$.

Without it, mobile may shrink or zoom out pages.

③ How does bootstrap assist in creating responsive layout?

Discuss concept of grid system & how it adapts to diff. screen size.

Ans →

P.T.O

Ques *

Q3. Ans Bootstrap helps create responsive layouts by providing 12 column grid system. We place content inside columns and bootstrap automatically arranges them.

(4) Diff. b/w tag, class, ID selectors.

Ans

	Syntax	Applies to	Uniqueness
Tag selector	<code><p> {color:blue}</code>	All <code><p></code> elements	Not unique
Class selector	<code>.highlight {color:red}</code>	All elements with <code>class = "highlight"</code>	Can be reused
ID selector	<code>#main {background: yellow;}</code>	Single element with <code>id = "main"</code>	Must be unique.

(5) Describe three main ways to apply CSS to HTML.

Ans i) Inline CSS

→ Applied directly to an element using `style` attribute.

2) Internal CSS

→ Written inside a `<Style>` tag in HTML `<head>`

3) External CSS

→ Stored in separate .css file & linked using :-
`<link rel = "stylesheet" href = "style.css" >`

* CONCLUSION :-

In this assignment, we learned about how to build RWD pages using HTML, CSS & Bootstrap - I understood the use of different CSS selectors, the ways to apply CSS.

KUMAR UJJAWAL

PRN-1032232774

R.NO-56

7Y CSF

*** AIM:-**

- To develop a web application using javascript to implement sessions, cookies, DOM. Perform validations such as checking for emptiness, only numbers for phone numbers, special character req. for password, regular expressions for certain format of the fields etc. Use mySQL database.

*** OBJECTIVES:-**

1. To understand what format 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 expressions. Why are they a suitable tool for validating data formats like a phone number or checking for the presence of specific characters in a password?

Ans: Regular expressions are patterns used to match, search, and manipulate strings based on specific rules. They are suitable for validating data formats because they allow you to define exact patterns.

for ex. we can use regex to ensure a phone no has the correct no. of digits etc. This enables quick and efficient validation.



(03)

(02)

② Explain the fundamental diff. b/w a session & a cookie in context of web application development. How do they work together to maintain a user's logged-in state.

Ans A cookie stores small data on the user's browser, while a session stores data on the server.

Cookies hold a session ID, which the server uses to retrieve session data, maintaining the user's logged-in state across requests.

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 Purpose :-

- Client-based - ensures immediate feedback to user, improving user experience and reducing server load.

- Server-side - ensures data integrity and security, as it cannot be bypassed by the user.

Scenario of security vulnerability?

If we rely solely on client-side validation, a user can bypass it by disabling JavaScript.

Ex → A user could submit a weak password directly to the server, causing a security risk. Without server-side validation, this could lead to compromised accounts.

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

Ans:

index.htm

```

<html lang="en">
  <head>
    <meta charset = "UTF -8">
    <title> Example </title>
  </head> <body>
    <h1> DOM interaction </h1>
    <button id="change Button"> Content Change </button>
    <script src = "script.js"> </script>
  </body> </html>

```

→ script.js:

```

const button = document.getElementById('change Button');
const msgpara = document.getElementById('message');
button.addEventListener('click', function() {
  msgpara.textContent = 'Content has been updated';
});

```

⑤ Give the steps for connectivity from front end using HTML CSS JS to mysql.

1. frontend (HTML/CSS/JS) → send request (AJAX/Fetch)
2. Backend (Node.js/PHP/Python) → connect to MySQL
3. Backend runs SQL → returns JSON
4. frontend shows response.

Flow: frontend → Backend API → MySQL → Backend → frontend.

~~Q~~ ~~False :-~~

1. 3 reasons why form validations are important.

Ans → ① Accuracy - Ensures correct and properly formatted data.
 ② Security - Blocks malicious or harmful inputs.
 ③ User Experience - Gives instant feedback and reduces errors.

2. Give an example of how to modify an attribute value using DOM.

Ans →

```
<!DOCTYPE html>
<html>
  <body>
    
    <button onclick="changeImage()">Change Image </button>

    <script>
      function changeImage() {
        document.getElementById("myImg").setAttribute("src", "new.jpg");
      }
    </script>
  </body>
</html>
```

Clicking the button changes the src attribute of image.

3. What are the different types features of JavaScript?

Ans → ① Lightweight & interpreted - Runs directly in browser without compilation.
 ② Object-Oriented - supports objects, prototypes.
 ③ Dynamic Typing - no need to declare variable.
 ④ Cross-Platform
 ⑤ DOM - manipulation.

ASSIGNMENT-03

- * AIM:- Design an interactive front end application using React by implementing template using components, states & props, class, events.
- * OBJECTIVE:- To develop a responsive, interactive front end application using react.js that effectively, demonstrating component-based architecture, state management & event handling with scalable UI components, dynamic data via states & props.

* THEORY:-

- ① Explain 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.

Ans State: Represents mutable data owned & managed by a component. It allows components to create dynamic & interactive UI's by updating themselves when the state changes.

Props: short for 'properties', props are read-only inputs passed from a parent component to a child.

Difference: state is internal & changeable, while props are external & immutable together, they enable unidirectional data flow in react.

Q2) What is a react component? Diff. b/w a class component & a functional components & discuss the adv. of using a fn. components with hooks like useState & useEffect.

Ans:- Component :- A reusable, independent piece of UI in React.

Class components:- Defined using class classes, use this state

Functional components :- Defined as functions, use hooks like useState & useEffect

Adv. -> Cleaner syntax, less boilerplate better performance easier to test & modern React development favors hook based classes.

Q3) 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 files?

Ans:-

- Breaks UI into reusable, modular components.
- Superior to monolithic HTML → improves reusability, scalability & maintainability.

Q4) How do you handle user events in React?

Provide a simple code snippet to demonstrate how an event handler is defined.



→ Handlers defined as functions; update state using :-
use state.

```
import React {use state} from "react";
function counter () {
  const [count, set count] = use state(0);
  return (
    <div>
      <p> {count} </p>
      <button on click = {() => set count
        (count + 1)}> Inc </button>
    </div>
  );
}
```

Q) What is responsive design, why is it crucial for modern app? Describe how would you implement a responsive design in a React application using CSS media queries or a CSS in JS library.

Ans) Ensures UI adopts to all screen sizes/devices
Implement using CSS media queries as CSS in JS
Eg:- media (max-width: 600px) {
 div { font-size: 10px; }}

CB
14/10/28

* AIM:- Enhance web page developed in earlier assignment by rendering lists and portals. Error handling, routes and style with react CSS, also make it a responsive design to scale well across PC, tablet and mobile phone.

* OBJECTIVES:-

- Enhance UI and experience.
- Improve application robustness and navigation.

* THEORY :-

Q1. How do lists and Keys work in react?

Ans → Lists in react allows you to render multiple elements dynamically using JS map() method to loop over data (e.g. an array of objects).

→ Keys are unique identifiers assigned to each element in a list to help react efficiently update and render only the changed items during re-renders.

Q2. What is a react Portal and when would you use one?

Ans → It is a way to render children outside their parent components hierarchy.

* It is useful when you need to render components that are visually or functionally separate like modals, tool tips or pop-ups.

Q3. Discuss the importance of error boundaries in React.

Ans → Error boundaries are components that catch JS errors anywhere in their child component tree, log those errors and then finally display a fall back UI.

They prevent the entire application from crashing when an error occurs, providing a more user-friendly experience.

Q4. How does React Router enable single page application functionality.

Ans By allowing you to define routes and navigate b/w diff. components without causing full page reloads, It dynamically changes the view in response to URL changes, providing a seamless user experience within client-side routing.

Q5. Explain the diff ways to style a react application.

Ans → Inline styling :- Using the style attribute with objects.

CSS style sheets :- Importing regular .css files & applying class names.

CSS modules:- Imported styles using module .css files to avoid naming conflicts.

* CONCLUSION:-

This assignment has my react knowledge by adding lists, portals, error handling, routing and responsive styling enhancing usability, navigation and reliability across devices.

26
14/10/20

ASSIGNMENT - 05

* AIM:- Develop a responsive web design using express framework to perform CRUD operations & deploy with Node.js using MongoDB.

OBJECTIVE:- ① Develop a full-stack web app.
② Demonstrate Backend Development & Deployment proficiency.

* THEORY :-

- Q1. Role of Express.js as a web framework for node.js.
Ans → Provides a lightweight fast & flexible web framework built on top of node.js.
 - Simplifies handling HTTP req & responses.
 - Supports routing.
 - Middle ware support for req. processing.

Q2. Explain concept of CRUD operations in context of web opp.

Ans → CRUD stands for Create, read, update delete.

CREATE - Add new data

READ - Retrieve or view existing data

UPDATE - Modify existing data

DELETE - Remove data from DB.

Q3. Why is MongoDB a suitable choice for this project?

Ans → MongoDB is ideal because it is a NoSQL, document-oriented database that stores data in JSON-like format, making it easy to integrate with Node.js and Express. It offers high scalability, flexibility and fast performance.

Q4. What steps are involved in deploying a node.js and express app.

Ans → Main steps are:-

- ① Build app: Develop a express app and connect to MongoDB.
- ② Set Environment variables: Configure database URLs and ports.
- ③ Test locally: Run and verify the app on localhost.
- ④ Deploy to a server: Use services like Render, Vercel.
- ⑤ Install dependencies on server: npm install.
- ⑥ Run app: Start app with node app.js
- ⑦ Verify deployment: Ensure the app runs correctly thru live URL

* CONCLUSION:-

Express JS simplifies backend development for node.js while MongoDB offers flexible scalable database soln.

This helps to demonstrate both backend & deployment skills.

✓ (B)
14/10/23