

FSD Assignment - 1

Aim → Develop responsive web design using HTML 5, containing a form. Style the pages using CSS, use of `id` tag selector, class selectors and `id` selectors. Using inline, internal and external CSS, Apply Bootstrap CSS.

Objective → i) To understand HTML tags
ii) To learn the styling of web pages using CSS.
iii) To learn Bootstrap framework.

Theory → RWD is a web design approach that makes web pages adjust and display properly on all devices.

- Q) Define responsive web design (RWD) what is its primary goal?

Ans → RWD is a web design approach that makes web pages adjust and display properly on all devices.

Primary goal → To provide an optional viewing and interactive experience across different screen sizes.

- 2) Explain the value of `<meta name = "viewport" tag`. why is this essential for RWD?

Ans → It controls how a web page is displayed on mobile devices by setting the layout's size and scale.

Essential for RWD : without it, RWD may appear zoomed out for or not fit properly on small devices.

3) Show class Bootstrap abstract in creating a responsive layout ? Discussing the concept of a grid system and how it adapts to different screen sizes.

Ans → Bootstrap uses a flexible grid system to design layouts that automatically adapts to different screen sizes without writing complex media queries.

- The bootstrap grid is based on 12 columns.
- A web page layout is created by placing content inside rows and dividing it into columns.
- It uses breakpoints (x_s , x_m , x_l) to define layout changes automatically stack vertically on smaller screens and align horizontally on larger screens.

4) Differentiate b/w tag, class and ID

Selector

Syntax

Purpose

i) Tag

`p { color: red; }`

styles all element
of particular tag

ii) class

`.btn { color: blue; }`

styles all element
with some class
name

iii) ID

`#header { color: green; }`

styles unique element
with a specific
ID.

5) Describe the three main ways to apply CSS
to a HTML document.

i) Inline CSS: Directly in HTML element's
style attribute.

`<p style = "color: red;"> </p>`

ii) Internal CSS: Inside ~~as style tag~~
in `<head>` section of HTML
document.

`< style> p { color: red; } </style>`

iii) External CSS : Linking an external .css file with <link> tag

<link rel="stylesheet" href="style.css">

Assignment - 2

Aim: Develop a web application using java script to implement sessions, cookies, DOM. Perform validations such as checking for emptiness, only numbers for phone number, special character requirement for password, regular expressions for certain format of the fields etc. use the my SQL database.

Objectives:

- i) To understand what form validation is
- ii) To learn basic functioning of DOM objects
- iii) 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 format like a phone number or checking for the presence of specific character in a password?

→ Regular expressions (regex) are patterns used to match and manipulate text. They are suitable for validating data formats like phone numbers or checking passwords rules because they can.

• Define precise patterns (eg. digits, letters, special, symbols).

- Quick verify if input matches required format
 - Detect presence / absence of specific character.
- this makes regex a powerful and efficient tool for input validation.

2 Explain the fundamental difference between a session and a cookie in the context of web application development. How do they work together to maintain a user's logged-in state?

Ans → Difference:

- cookie: stored on the client's browser, holds small pieces of data (like session ID).
- session: stored on the server, holds actual user-related data

Working together:

When a user logs in, the server creates a session and sends its ID to the client via a cookie. The browser then sends this cookie with each request, allowing the server to identify the user and maintain the logged-in state.

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.

- Client side validation: improves user experience by catching errors early.
- Server side validation: Ensures data integrity and security.

vulnerability scenario:

If only client-side validation checks form input (e.g., email format), an attacker can bypass it by disabling JavaScript or sending requests directly to the server, injecting malicious data (like SQL injection).

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

Ex -

<!DOCTYPE html>

<html>

```
<body>
  <form onsubmit = "updateMessage();>
    return false;">
    <input type = "text" id = "name" placeholder
      = "Enter your name">
    <button type = "submit"> Submit </button>
    <p id = "message"></p>
```

```
<script>
  function updateMessage () {
    let name = document.getElementById
      ("name").value;
    document.getElementById ("message").innerHTML =
      "Hello, " + name + "!" ;
  }
</script>
```

```
</body>
</html>
```

when the form submitted , JS updates the content dynamically .

5

give the steps for connectivity from Front end using HTML CSS JS to mysql.

Ans →

Steps :

- i) Front end form / UI (HTML / CSS / JS)
- ii) Send data via JS (fetch / AJAX)
- iii) Backend (PHP / Node / Python) handles request

- iv) Backend connects to my SA2.
- v) Backend sends response
- vi) JS updates UI.

FAQ

- 1) write 3 reasons why form validations are important
 - i) ensures correct and complete data entry
 - ii) prevents invalid / malicious inputs
 - iii) improves user experience with instant feedback.
- 2) Given an example of how to modify an attribute value using DOM.

Ans:- ex:-

```
document.getElementById("my img").setAttribute  
("src", "new Image.jpg");
```

- 3) what are the different types features of JavaScript?

- Ans:-
- Light weight, interpreted and object-based.
 - supports event-driven and functional programming.
 - enables dynamic HTML / CSS manipulation & validation.

By
26/08/2022
Lat

FSO Assignment - 3

Ques :-

Design an interactive front end application using React by implementing templating components, states and 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 state fundamental concept 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 with component managing dynamic data with states, ensuring a seamless user experience across various devices and 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 in managing data flow within the

component based application?

state → internal data of a component

that can change during the component's lifecycle when state changes

see - `useState`

- props - short for properties, these are inputs passed from parent to child components.

→ difference:

state → managed within the component, mutable.

props → passed from outside, immutable

purpose → together they manage data flow. State handles dynamic data, props handle communication between components.

- Advantages of functional with Hook:
 - simpler and shorter code
 - easier to read and maintain
 - hooks provide powerful features without complex lifestyle methods.

- describe the concept of "templating" using components in React. why is this approach considered superior to traditional methods that rely on monolithic HTML files?

- what is a React component? differentiate between a class component & a functional component and discuss the advantages of using a functional component with tools like useState, useEffect etc. is a class component?

- React broke UI into reusable component instead of writing one big HTML file. Each component handles its own structure and logic. superiority over traditional HTML:

component → A reusable unit of code that return JSX

component → define using ES6 classes used this state and life cycle methods.

functional component → Defined as functions, used as hooks for state and life cycle.

- Encourage readability
- Easier maintenance and scalability

→ Designing websites that adjust automatically to diff. screen sizes.

- Events like (onclick, onChange) are handled using functions inside components.

4)

- How do you handle user events in React?

Example:

```
function Counter () {
  const [ count, setCount ] = React.useState(0)
  const handleclick = () => setCount(count + 1)
  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={handleclick}>Increase</button>
      <br/>
      <button onClick={()=>setCount(0)}>Reset</button>
    </div>
  )
}
```

Implementation:

```
. container {
  width: 100% ;
}
@ media (max-width: 768px) {
  . container {
    flex-direction: column;
  }
}
```

Importance → Improves user experience, accessibility & readability

- what is responsive web design and why is this crucial for modern applications?
- how you would implement a responsive web design in react.

Q
Ans

Assignment - 4

- Aim → Enhance the previously developed web page by implementing lists and portals, error handling, routers and styling using React, CSS, ensuring the designs is responsive across PC, tablet and mobile devices.
- objectives →
 - To enhance the robustness, and visual consistency of the application
 - To make the long layout adaptive and scalable on different screen sizes.

Theory

- what How do lists and keys work in React ?
 - Lists render multiple elements dynamically using map () .
 - key uniquely identify list items, helping react update or re-render efficiently.

Example → `items.map((item, i) =>
 key = {i} > { item })`

- what is a React portal and when should you use one ?

Ans: A portal lets you render a child component outside its parent DOM hierarchy using React DOM, create 'hosted' (child, target node).

- used for modals, pop-ups, or tooltips
- to avoid layout or Z-index issues.

3) discuss the importance of Error boundaries in React

- Ans →
 - catch javascript errors in child components and show a fallback UI instead of breaking the whole app.
 - improves stability and user experience.

4) How does React Router's SPA functionality?

- Ans →
 - React Router changes components based on URL without reloading the page.
 - Maintains a single HTML file and updates only the requested part.
 - User < Router > < Routes > < Routes >, and < Route > for navigation.

- i) CSS Modules
 - what → CSS files scoped locally by default
 - How → Name CSS files as module.css and import them as objects
 - Pros → scoped locally, avoids naming conflicts, still simple CSS.
 - Cons → slightly more setup, less dynamic.
- ii) Inline Styles

- Aim → serve express framework to perform operations and deploy with Node

• what → styled defined directly on JSX

• elements as objects

• elements attribute with

• Now → use style attribute with

• now → use except objects.

• now → dynamic styles easily applied, no external CSS files

• now → limited features (no pseudo classes)

• can check in JSX

• 1) Styled components (less in JS)

what → use javascript to define styled components like styled components

now → use libraries like styled-components or emotion.

• 2) ~~Node.js~~

what is the role of express.js as a web framework for Node.js?

→ Express.js is a lightweight and flexible framework built on top of Node.js framework. It simplifies the process of building server-side applications by providing features like routing, middleware.

→ It helps in creating RESTful APIs, managing routes and handling client-server communication efficiently.

• 3) Explain the concept of CRUD operations in the context of web application.

Install dependencies → Run npm install

Ans → CRUD stands for create, read, update,

and delete. It's common for front-end frameworks to do this.

Set up a server environment → Use platforms like Render, Heroku, AWS.

• Create : Add new data

• READ : Retrieve existing data

• UPDATE : Modify existing data

• DELETE : Remove data

CRUD operations from the foundation of data - base driven applications.

3) why is Mongo DB a suitable choice

Open this project with node.js

Ans → Mongo DB is a NoSQL, document-oriented database that stores data in flexible JSON format.

It allows easy scalability, faster access and schema-less structures. Making it ideal for applications like dynamic or unstructured data.

4) what style code involved in deploying?

o Node.js And express application!

Ans → Expose the application to host and finalize code, configure environment variables.

finalise code, configure environment variables.

(2)

i) Create

• what → Add new data to the system

• Example → Submitting a form to add a new

blog post or registering a new user.

• HTTP method → usually corresponds to POST.

ii) Read

• what → Retrieve or view existing data

• Example → viewing a list of blog posts or fetching user details.

• HTTP method → usually corresponds to GET.

iii) update

• what → modify existing data

• Example → Editing a blog post or updating a user profile.

• HTTP method → usually corresponds to PUT or PATCH.

PATCH.

- iv) Delete → remove and clean up Material
- what → Remove data from the System
 - Example → Deleting a blog post or removing a user account.
 - HTTP method → usually corresponds to **DELETE**.

✓
CM
10/10/25