Name :- Ashish Ajay Sharma
Class :- TYCSF
PRN :- 10322319O7 Roll No :- 36
Batch :- A1

FSD Laboratory 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 selector. Use of Inline, Internal and External CSS apply Bootstrap CSS

Objectives :

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

Theory :

Q1) Define Responsive Web Design (RWD). What is its primary goal?

Responsive Web Design (RWD) is an approach to web design that makes web pages render well on variety of devices and screen sizes. A responsive site fluidly changes its layout resizing, hiding, shrinking or enlarging content - to adapt viewing environment. This ensures that whether user is on mobile phone, tablet, laptop or large display monitor, the website is easy to read and navigate without excessive resizing planning or scrolling.

The primary goal of RWD is to provide an optim user experience (UX) for everyone regardless device they use. It aims to make web content accessible and functional across the entire spe of digital devices, improving usability of us selection.

Q2) Explain the role of <meta name = "viewport ....> tag. Why is it essential for RWD?

→ It is an HTML element placed in the <head> se of a page that gives the browser instructions on to control the pages dimensions and scalings. Standar implementation is as -

<meta name = "viewport" content = " width = device - w
initial - scale = 1.0 ">

where width device width tells browser to set the of viewport to the physical width of the device's scr initial scale = 1.0, sets the initial zoom level page to 100% (No zoom when first loaded).

This tag is foundational step for building a res site, without it mobile browsers will assume no non-responsive fixed width desktop site to display it, they render the page in a wide vir viewport (eg 980px) and then scale it down to fit the small scale. This makes text and im

Name - Ashish Ajay Sharma
PRN - 1032231907
Roll no - 36
Batch - A I

Ilegible, viewport tag prevents this behaviour, ensuring the browser uses actual device width allowing responsive CSS techniques to correctly apply styles based on the screen size

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

Bootstrap helps create responsive layouts by providing a mobile first, 12-column grid system that automatically adjusts elements based on screen size.

> Grid System → The layout is divided into 12 equal colums and you can combine them to create different widths

> Responsive Breakpoints → Bootstrap uses predefined classes (.col-, .col-sm-, .col-md-, .col-lg) that adapt to devices ranging from mobile to desktop.

> Fluid containers → elements are wrapped in .container or .container-fluid to ensure proper spacing and allignment

> Automatic Stacking → On smaller screens, columns stack vertically while on vertical screens they stack side by side.

These ensure that same web page are readable and well arranged on all devices

4) Differentiate between Tag, Class & ID selectors

Selector	Symbol/ Syntax	Usage	Example
Tag Selector	tagname	Targets all HTML elements of a given tag type	p { color: ; } changes text co
Class Selector	.classname	Targets one or more elements with a given class attribute	.highlig background }
ID Selector	#idname	Targets exactly one unique element with a given Id attribute	#heade font-s 24p }

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

→ The three main ways to apply CSS to an HTML document are as follows :-

Name - Ashish Vijay Sharma
PRN - 10322319007
Roll no - 36
Batch - A1

Inline CSS → Using the style attribute directly inside an HTML tag.

 Hello World

Used for quick, one time changes but not recommended for large projects

Internal CSS → Writing CSS inside a <style> tag in the <head> section of the HTML file

<style>
p { color: green; }
</style>.

Keeps styles within same file but still seperates them from HTML content

3) External CSS → linking an external .css file using the <link> tag.

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

Best practice for maintainability, reusability & keeping the code organized

Problem Statement : 3 (Roll Nos 25 to 36)

Conclusion :

Thus learned about various HTML tags, their syntax and usage. Came to know about Bo various methods of CSS styling and hence responsive web design using concepts used.

BJ
12/8/25

Name - Ashish Ajay Sharma
Class - TYCSF
PRN - 1032231907
Batch - A1.

FSD 2ak Assignment - 02

Aim :

Develop a web application using javascript to implement sessions, cookies, DOM. Perform validations such as checking for emptiness, only numbers for phone number, special characters requirement for password, regular expressions for certain format of the fields etc. Use the mySQL database.

Objectives :

> To understand what form validation is
> To learn basic functioning of DOM elements
> To learn how to apply various techniques to implement it.

Theory :

Explain the role of Regular expressions. Why are they a suitable tool for validating data formats like a phone number or checking for presence of specific characters in a password.

A regular expression (regex) is a formal sequence of characters that specifies a search pattern. Its primary role is in pattern matching and string

manipulation. In the context of data valid
regex provides a powerful and concise m
for defining a set of rules that input d
must adhere to.

For example, a phone number format can be
using a pattern like ^\(?\d{3}\)?[-.\s]?
[-.\s]?\d{4}$ to allow for various
common such formats.

For password validation, a regex can enfor
complexity requirements - such as length, ch
types (uppercase, lowercase, numbers, symbol
in a single expression, which is significantly
efficient than writing verbose conditional lo

Q2) Explain the fundamental difference between a
and a cookie in the context of web applica
development. How do they work together to m
a users logged in state?

→ The fundamental difference between a Sess
and a cookie is the storage location. A
cookie is a small piece of data stored in
client side within the users browser.

In contrast session is a collection of d
stored on the server side. They work together

Maintain a users state, such as being logged in. Upon successful login, the server creates a session and sends unique id to the browser. The browser stores this ID in a cookie. For every subsequent request the browser sends this cookie back, allowing the server to identify the user, retrieve their session data & maintain continous stateful experience

What is the purpose of performing both client side & server side validations? Describe a scenario where relying solely on client side validation could lead to a security vulnerability.

The purpose of performing both client side and server side validations is to layer user experience with security. Client-Side validation executing in browser using Javascript provides immediate feedback to the user, improving usability catching errors before a form is submitted.

Server side validation is the authoritative, non-negotiable security check. It protects the application from malicious data as client side scripts can be easily bypassed by disabling Javascript or manipulating HTTP requests.

eg →

A critical security vulnerability arises when solely relying on client side validation in an e commerce

checkout. A user could intercept the form and change the item price from ₹5000 to ₹1. Without a server side validation to check the price against the database the system would process the fraudulent transaction.

Q4) Provide an 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.

→ The Document object model (DOM) is a programming interface for web documents. JavaScript can use this to manipulate the page.

following is a simple example on how JavaScript can change text on a page after a button click.

HTML:

```
<p id="message">Hello, world!</p>
<button onclick="changeText()">Click Me</button>
```

Javascript

```
function changeText() {
    // 1. Find the element by its id.
    const ele = document.getElementById("message");

    // Change its content.
    ele.textContent = "You Clicked Me!";
}
```

When the user clicks the button, the changeText() function is called. It finds the paragraph with the ID message and changes its text content.

3) Give the steps for connectivity from Front end using HTML CSS JS to mysql

Direct connectivity from HTML CSS JS to a MySQL database is not possible and represents a major security flaw. The connection must be mediated by a backend server. The process involves a three-tier architecture

> Client (Frontend):

The user interacts with an HTML form. Javascript's Fetch API or Axios sends the form data to specific API endpoint on the server via an HTTP request (eg POST).

> Server (Backend)

A server side application (build with Node.js, Python, PHP, etc) receives the request. This layer contains the security logic & database credentials

> Database

The server connects to the MySQL database, validates and

and sanitizes the received data, and executes appropriate SQL query (eg Insert). Then response back to frontend.

FAQ

⇒ Write 3 reasons why Form Validations are important.

→ Form validations are crucial for three main as follows.

⇒ Firstly they ensure data integrity by ensuring the data entered in the database conforms to required format and business rules, preventing corruption.

⇒ Second, they enhance application security. Server side validation is a critical defence a common web vulnerabilities like SQL injection Cross Site Scripting (XSS) by sanitizing & rej malicious user inputs.

⇒ Third Client side validation significantly imp user experience by providing immediate, co feedback to the users allowing them to corr errors instantly without the delay of a r trip to the server, which reduces user frustra and increases form completion rates

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

To modify the attribute value of an element using DOM, you can select the element and then change the property of its attribute.

for src → to change source of an Image

HTML:

<img id="myImage" src="old_image.jpg" alt="old img">

Javascript

// find the image element by its ID

const image = document.getElementById("myImage")

// Modify its src attribute

image.src = 'new_image.jpg';

// You can also using change to other attributes like

image.alt = 'A hand new image';

3) What are the different features of Java Script.

Javascript is a versatile, high level programming language with several key features. It is an interpreted language, meaning code is executed line by line without needing a compiling step.

-> It supports dynamic typing, where variables are determined at runtime.

-> Functions in JavaScript are first class citizens allowing them to be stored in variables, passed as arguments, and returned to other functions.

-> It utilizes prototype based inheritance where objects inherit properties directly from other objects.

-> Crucially it has non blocking asynchronous nature, handling long running tasks like API calls efficiently without freezing the main program execution, primarily through Promises and the async/await syntax.

~~Problem Statement~~

Conclusion

Thus developed a web application using JavaScript to implement sessions, cookies, DOM. Performed both Client side and Server side validations using regular expressions & integrating SQL.

11/12/25

Assignment - 3

Aim

Design an interactive front end application using React by implementing templating using 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 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 components, managing dynamic data with State and props and handling user interactions with events, ensuring a seamless user experience across various devices & screen sizes.

Theory

1) Explain the role of State and Props in React How do they differ, and what is the purpose

purpose of each managing data flow within a component based application

→ State → These are internal, Mutable data spe to a component.
Controls dynamic behaviour and r rendering.

Props → These are External, Immutable data passed from parent to child.
Enables component reuse and flow.

Differences → State changes within the compo Props are read only from the Child's perspective.

2) What is a React Component? Differentiate between a Class Component and a functiona Component and discuss the advantages of us a functional component with hooks like State and use Effect over a class compo

→ Components are a self contained reusa UI building block in React that can ac inputs (props) and manage internal data

Class Comonet are defined using class syntax and extend React.Component. Can maintain internal state, access lifecycle methods (componentDidMount, componentDidUpdate etc.) and handle complex logic.

Functional Component defined as a plain JavaScript function returning JSX. Initially stateless, but with Hooks (useState, useEffect) it can manage state, perform side effects, and replicate lifecycle behaviour.

Advantages of Functional component~~state~~ and Hooks are

- Cleaner, more readable code with less boilerplate
- Easier to manage state and side effects.
- Encourages modular, testable code.
- Better performance in some cases due to simpler structure and avoidance of unnessary lifecycle overhead.

3) How do you handle user Events in React (eg. a button click)? Provide a simple code snippet to demonstrate how an event Handler is defined in a component and how it can be used to update components state.

→ React uses & exposes Synthetic Events (Cross-browser wrapper around native events). You attach event handlers to elements via JSX attributes (eg. onClick) and update setState (class) or useState (functional) to trigger re-renders.

Notes →

- Use arrow functions or .bind in class components to preserve this.
- Prefers components and hooks used together for simpler code.
- Don't directly mutate state - always use state setters (setState / setX).

example of functional component.

```
import React, { useState } from "react"

function Counter() {
    const [count, setCount] = useState(0)
    const handle = () => {
        setCount(prev => prev+1)
    };

    return (
```

```
<div>
    <p> Count : {count} </p>
    <button onClick={ handleClick }> Increment </button>
</div>
);
}

export default counter;
```

Q4) 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?

→ In React, "Templating using components" helps us to build user interfaces (UI) by composing many small components (Button, Card, Header, Production List). Each component encapsulates markup (template) Style and behaviour. The app formed is then a tree of components.

Its superiority to monolithic HTML file are as follows.

→ Maintainability → Small focused files are easier to read change and test than larger HTML files plus scattered scripts

- Reuse and DRY → Small components avoid repetition, update once, affect all
- Separation of Concerns → The logic related piece of UI stays with the piece (no loo global scripts manipulating DOM)
- Scalability → easier to scale teams and codebase components map to features/

5) What is responsive web design, and why is crucial for modern application? Describe you would implement a responsive desig a React application using CSS media ques or a CSS in JS library.

→ A responsive web design (RWD) ensures a website adapts to different screen sizes and de like desktop, tablet, mobile etc.

- Essential for accessability and search engi optimization
- Reduces the need for multiple device specif versions implementing responsiveness in React.

CSS media queries:

```
.container {
    width = 100%;
}
@ media (max - width : 768 px) {
    .container {
        padding : 10 px;
        font - size : 14 px;
    }
}
```

CSS in JS

```
import Styled from "styled - components";
const Box = Styled.div
width : 100 %.
padding : 20 px;
@ media (max width : 718 px) {
    padding : 10 px;
    font - size : 14 px;
};
```

UI libraries -> Bootstrap, Material UI, Tailwind C etc, these libraries provide us with built in responsive classes.

Problem Statement

{Roll no 21 to 40} Design and develop a responsive react application for a user inter

dashboard. The application should display user information, including a profile picture, name and a list of recent activities or posts.

Conclusion

Thus learnt and implemented some of the principles of React development, components & how to make UI modular and reusable. State & props ~~excerise~~ and ensure efficient data transfer & dynamic updates.

26/9/25

Assignment - 4

Aim

Enhance Web page developed in earlier assignment by rendering Lists and Portals, Error handling, Routers and Style with React CSS also make it a responsive design to scale well across PC, tablet and Mobile Phones.

Objectives

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

Theory

1) How do Lists and Keys work in React?

→ When rendering dynamic array of elements, React lets us use .map() function to produce a list of component / elements

Code snippet:

```
Const fruits = ["Apple", "Banana", "Cherry"]
return (
  <ul> {fruits.map(fruit => <li key={fruit}>{fruit}
  </li>)} </ul>
```

While keys are special prop used to he identify which items have changed and removed. Keys should be unique stable predictable (to avoid using array ind data can reorder) without keys, React unnecessarily re-render / reuse the wr DOM nodes.

Q2) What is React Portal and when would you one?

→ A react Portal allows rendering children in different part of the DOM Tree (outside the components hierarchy). While keeping the logically part of the React component t

```
→ import reactDOM from "react-dom";

function Modal ({children}) {
    return ReactDOM.createPortal (
        <div className="modal"> {child
        </div>,
        document.getElementById("modal-r
    ); }.
```

Models, tooltips dropdown, banners → elemen that need to visually read out of parent container (eg: outside overflow hidden or stacking context) but still controlled by React.

Q3) Discuss the importance of Error Boundaries in React.

→ Error Boundaries in React components that catch JavaScript errors in their child component tree and prevent the whole app from crashing. It can display a fallback UI ("like something went wrong") catch rendering errors, lifecycle errors and constructor errors but not event handler or async errors

eg → class ErrorBoundary extends React.Component {
state = { hasError: false };

static getDerivedStateFromError() {
return { hasError: true };
}

componentDidCatch(error, info) {
console.error("Error caught");
}

render() {
if (this.state.hasError) {
return <h2>Something went wrong</h2>
}
return this.props.children;
}
}

In production, it prevents a single broken component from bringing down the entire application

Q4) How does React Router enable Single Page Application (SPA) functionality?

→ React router enables Single Page Application (SPA) behavior by handling routing on the client side without full page reloads. When the user navigates to a different path (eg /about), React Router intercepts it and renders the corresponding component while the browser's history API keeps the URL in sync. This makes navigations seamless and fast unlike traditional multi page application.

For example.

```
<Router>
  <Route path="/" element={<Home />} />
  <Route path="/about" element={<About />} />
</Router>
```

Q5) Explain the different ways to style a React Application.

→ React supports multiple styling approaches based on the project needs.

• The simplest is plain CSS files with ClassName eg `<div className="card">`

- Inline styles use a JS object, for example <div style={{ color: "red" }}>.

- For modular & scoped CSS styles, CSS modules or CSS-in-JS libraries like styled-components are common. eg - const Button = styles.button`color: blue;`.

- Frameworks like Tailwind CSS provide utility first responsive styling

Choosing the method depends on scalability, performance and team preference.

Problem Statement

Expand the e-commerce product gallery to include a responsive modal for displaying product details (Roll 21 to 40)

Conclusion

Thus implemented and understood the core concepts of lists, keys, Portals, error boundaries React Router & styling techniques in react these enhancements collectively improve both user experience and application robustness

M 26/9/25

Assignment - 5

Aim

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

Objectives

-> Develop a Full Stack Web Application
-> Demonstrate Backend development and Deployment Proficiency

Theory

Q1) What is the role of Express.js as a web framework for Node.js?

-> Express.js plays the role of a lightweight yet powerful framework that extends Node.js with structured tools for building web applications and API's.

While Node.js on its own requires developers to handle low level ~~no~~ details like creating HTTP servers and parsing requests, Express simplifies this with built in routing, middleware support and request response handling.

It allows developers to organize code into routes controllers, making large projects easier to maintain

for eg → instead of writing a full HTTP request handler in express you can simply write app.get("/users", (req, res) => {
res.json(users);

and the framework takes care of the rest. This makes the backbone of many server side applications that need speed and scalability.

Q2) Explain the concept of CRUD operations in context of web application

→ CRUD stands for Create, Read, Update & Delete the four basic functions for interacting with data in web applications

Operations & Description with example.

Create → Add new data (eg register a new user)

Read → Retrieve existing data (eg view profile).

Update → Modify existing data (eg update user profile)

Delete → Remove data (eg delete a record)

```
// Create
app.post("/posts", (req, res) =>
    Post.create(req.body));

// Read
app.get("/posts", (req, res) =>
    Post.find())

// Update
app.put("/posts/:id", (req, res) =>
    Post.findByIdAndUpdate
    (req.params.id, req.body)

// Delete
app.delete("/posts/:id", (req, res) =>
    Post.findByIdAndDelete
    (req.params.id));
```

Q3) Why is MongoDB suitable choice for this project?

→ MongoDB is a Document oriented NoSQL database that stores data in JSON like format. This makes it very natural to use with JavaScript and Node.js. Its flexible schema allows developers to store different types of data without needing a fixed structure which speeds up the overall development. MongoDB also supports scalability, performance and easy integration with Express.js with libraries like Mongoose.

Q1) What steps are involved in deploying a Node.js and express app.

→ ① Develop locally

Build your apps with express routes and Mongo DB database.

② Version Control

Push your code to github or other repository.

③ Choose Hosting Platform

Use platforms like Heroku, Render AWS or Digital Ocean.

④ Install Dependencies

Run npm install on server.

⑤ Configure environment variables

Set up mongo DB connection string.

⑥ Run apps

Start node.js server or a process manager like PM2.

Conclusion

Thus learnt and implemented responsive web design using Express framework and used hosting services to build and deploy the MongoDB & Nodejs app.