Karthik Kaushik
1032231250
Roll No- 26
T.Y CSF

FSD Lab Assignment - 1

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:- (i) Understand HTML tags
(ii) Learn styling of web pages using CSS
(iii) Learn bootstrap front end framework

Problem Statement: Develop a responsive event registration page with a detailed form. The form should collect attendee information, including name, email, phone number, and a choice of workshop sessions using radio buttons or a dropdown menu. The page must be styled using Bootstrap CSS to create a professional and user friendly interface. The styling for the form elements and the page as a whole should be implemented with tag, class and ID ~~sectors~~ selectors, primarily through an external CSS file.

Theory:

① Define Responsive Web Design (RWD), what is its primary go

Ans Responsive web design 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 ~~updates~~ adapts layout, images and other elements based on device's screen size.

Kartik Kaushik

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

Ans The tag tells browser how to control page dimensions & scaling.
width = device-width → width matches device screen width
initial-scale = 1.0 → 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 different screen size.

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

④ Differentiate b/w tag, class, ID selectors.

Ans

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

Kartik Kaushik
1032231250
Roll No - 26
T.Y CSF

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

(i) Inline CSS

→ Applied directly to an element using style attribute

(ii) Internal CSS

→ Written inside a <style> tag in HTML <head>

(iii) External CSS

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

Conclusion:- Learnt the development of Responsive Web Page and CSS bootstrap.

7/2/8/25

Kartik Kaushik
10322350
T.Y CSF
Roll No - 26

FSD 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 numbers, special character requirement for password etc. Use 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) Role of Regular Expressions (Regex)

→ Regex are patterns for matching strings, useful for validating formats like phone numbers, passwords, or emails by enforcing rules on characters.

2) Explain the fundamental difference b/w 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?

→ Difference b/w session and cookie.

- Cookies are small pieces of data stored on the client side used to remember user preferences or identifiers.
- Sessions store user data on the server, identified by a Session ID usually stored in a cookie.

Together, they maintain user login states securely by keeping sensitive data server side (sessions) while using cookies for identification.

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 provide instant feedback and reduces server load.
• Server-side validation ensures security by validating data regardless of client manipulation. Relying only on client-side validation can be bypassed by attackers, leading to security risks such as SQL injection or data corruption.

4) Provide a simple example 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, form submission.

→
```
<p id="msg"> Hello </p>
<button onclick="changeText()"> Click me </button>

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

5) Give the steps for connectivity from FrontEnd using HTML CSS JS to my sql

→ Steps for Frontend to mysql connectivity.
- Create HTML form to collect data.
- Use Javascript / AJAX to send data to server.
- Server - side script (Node.js, PHP, etc.) receives data.
- Server – side script connects to MySQL database.
- Insert or query data in MySQL.
- Send response back to front end.

FAQs:

Q1) Write 3 reasons why form validations are important.

Ans Three reasons why form validations are important:
1) Prevent submission of incomplete or incorrect data.
2) Improve user experience with instant feedback.
3) Enhance security by avoiding malicious inputs.

Q2.) Give an example of how to modify an attribute value using DOM

Ans document.getElementById("myImage").setAttribute('src','newImage.jpg');

Q3.) * What are different features of java script?

Ans →
- Client-side scripting language
- Dynamic typing
- Event driven programming
- DOM manipulation capability
- Supports object-oriented and functional programming.

Conclusion:

This assignment demonstrates the importance of client-side form validation, efficient DOM manipulation using JS and integration of jquery for enhanced interactivity.

9/9/25

Kartik Kaushik
10322312150
Roll No-26
T.Y B.Tech CSF

FSD Assignment - 03

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.

Objective: To develop a responsive, interactive front-end application using React.js that effectively demonstrates component-based architecture, state management & event handling with scalable UI components, dynamic data via states & props & seamless user interactions across devices.

Theory:

1) Explain the role of State & props in React. How do they differ. & what is primary purpose of each in manging data flow within a component-based application.

Ans **State:** Represent mutable data and owned & managed by a component. It allows components to create dynamic and 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, enabling data sharing across components.

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

Q2.) What is a React component? Differentiate b/w a class component & a functional component & discuss the advantages of using a functional component with hooks like useState & useEffect over a class component.

Ans Component: A reusable independent piece of UI is React.

Class Components: Defined using ES6 classes, use this state & lifecycle methods.

Functional components: Defined as functions, use hooks like useState & useEffect for state & lifecycle management.

Advantages of Functional + Hooks: - Clearer syntax, less boiler plate, better performance, easier to test, & moder React development favors hooks over classes.

3.) 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?

Breaks UI into reusable, modular components

Superior to monolithic HTML → improves reusability, scalability & maintainability.

How do you handle user events in React? Provide a simple code snippet to demonstrate how an event handler is defined in a component & how it can be used to update the component's state.

Ans Handlers defined as functions; update state using:- useState

```
import React { useState } from "react";
function Counter() {
  const [count, setCount] = useState(0);
  return (
    <div>
      <p> {count} </p>
      <button onClick={() => setCount(count+1)}>Inc
      </button>
    </div>
  );
}
```

Q5.) What is responsive design, why is it crucial for modern applications? Describe how you would implement a responsive design in a React application using CSS media queries or a CSS-in-JS library.

Ans

Ensures UI ~~adpts~~ adapts to all screen sizes / devices.
Implement using CSS media queries or CSS in JS.

Eg- `@media(max-width: 600px) { div { font-size: 14px; } }`

23/9/25

Conclusion: This assignment effectively demonstrates ~~how~~ how React's component-based architecture, state and props work together to build a scalable and responsive frontend.

Kartik Kaushik
1032231250
Roll No-26
T.Y CSF

FSD Assignment - 04

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 Phone.

Objectives:
- Enhance User Interface and Experience
- Improve Application Robustness and Navigation.

Theory:

Q1.) How do Lists and Keys work in React?

Ans → Lists in React allow you to render multiple elements dynamically using JavaScript's map() method to loop over data (eg. 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 ~~how~~ when would you use one?

Ans • React Portal is a way to render children outside their parent component's DOM hierarchy.

• It is useful when you need to render components that are visually or functionally seperate, like models, tooltips, or pop-ups, but want to

Keeps them logically within the same component structure.

Q3.) Discuss the importance of error boundaries in React.

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

They prevent the entire application from crashing when an error occurs, providing a more user-friendly experience by showing a graceful fallback istead of a broken UI.

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

React Router enables SPA functionality by allowing you to define routes and navigate b/w different components without causing full-page reloads. It dynamically changes the view in response to URL changes, providing a seamless user experience with client-side routing.

(05) Explain different ways to style a React application.

Ans
- CSS Stylesheets: Traditional external or internal CSS files.
- CSS-in-JS: Using libraries like styled-components or emotion to write scoped CSS in javascript files.
- Inline Styles: Using the style attribute with Javascript objects.
- CSS Modules: Scoping CSS locally to components to avoid global style clashes.
- Preprocessors: Using tools like Sass or LESS with React for enhanced CSS features.

Problem Statement - Expand the e-commerce product gallery application to include a responsive modal for displaying product details.

Conclusion: This assignment has ~~extra etc~~ my React knowledge by adding lists, portals, error handling, routing and responsive styling, enhancing usability, navigation and reliability across devices.

29/9/25

Kartik Kaushik
1032231250
Roll No-26
T.Y CSF

FSD Assignment -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:

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

Ans Express.js is a minimal and flexible web application framework for Node.js. It simplifies the development of web servers and APIs by providing a robust set of features, including routing, middleware, and template engines. It allows you to handle HTTP requests and responses more easily, and provides a solid structure for organizing your web application.

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

Ans CRUD stands for Create, Read, Update, Delete, the four basic operations for interacting with data in a datab
- Create: Insert new data (eg. add a new user)
- Read: Retrieve data (eg. get a list of users or a single user's d
- Update: Modify existing data (eg. update a user's information
- Delete: Remove data (eg. delete a user from database)

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

Ans MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. It is well suited for application that need high scalability and can handle unstructured or semi-structured data. MongoDB integrates seamlessly with Node.js and Express, allowing you to store, retrieve, and modify data with ease.

Q4.) What steps are involved in deploying a Node.js and Express application?

Ans 1) Set up your environment: Ensure Node.js, Express & MongoDB is installed

2) Write your application code: Develop the frontend and backend

3) Choose a cloud platform like AWS or Digital Ocean to host the application.

4) Push to version control: Store your code in a version control system like Git, and push it to a repository (Github).

5) Configure environment variables: Set up your application's environment variables for things like database URLs, ports etc.

6) Deploy: Deploy the app on your cloud platform using a service like Heroku which can automatically set up your app.

7) Connect the database: Ensure your app connects to MongoDB, either locally or using a cloud database service like MongoDB Atlas.

8) Test and Monitor: Once deployed, test and monitor it for issues such as errors or downtime.

Conclusion: In summary, Express.js simplifies backend development for Node.js, while MongoDB offers a flexible, scalable database solution. By integrating these with Node.js, we can efficiently handle CRUD operations in a full-stack web application. This project helps demonstrate both backend development and deployment skills.

21/9/25