Name - Khushi Tiwari
RollNo. - 32 PRN - 1032231420
Program - TY CSE CSF
Batch - A1

DATE - 10/08/25
Pg No

FSD Laboratory - 1

Aim - Develop Responsive web design using HTML5, Containing a form. Style the pages using CSS, use of tag selector, class selector and id selectors. 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 Bootstrap Front End framework.

Theory -

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

→ Responsive Web Design (RWD) is a web design approach that ensures a web site's layout and content automatically adjust to fit different screen sizes, resolutions, and devices (such as desktops, tablets, and smartphones) without losing usability or visual appeal.
The primary goal is to provide an optimal viewing and interaction experience across a wide range of devices ensuring easy reading, smooth navigation, and minimal resizing, planning or scrolling.

2) Explain the role of the <meta name = "view port"...> tag. Why is this tag essential for RWD?

Ans → The <meta name = "viewport" content = "width = device-width initial-scale = 1.0"> tag tells the browser how to control the page's dimensions

scaling. On responsive web design, it ensures that the web page's width matches the device's screen width, making the layout adjust correctly for different devices.

Why It is Essential for RWD:
Without this tag, mobile browsers typically display web pages scaled to a desktop-width viewport, causing the content to appear small and requiring zooming or horizontal scrolling. The tag is essential because it:

① Enables proper scaling of content on different screen sizes.

② Ensures responsiveness by allowing CSS media queries to work as intended.

③ Improves user experience by avoiding unnecessary zooming and scrolling.

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

Ans → Bootstrap is a popular front-end framework that simplifies the process of creating responsive websites. It provides predefined CSS classes and a grid system that automatically adjusts layouts for various devices, ensuring content looks good on all screen sizes without extensive custom CSS.

Grid system concept:

- Bootstrap uses a 12-column grid system where the page is divided into up to 12 equal parts horizontally.
- Developers place content inside rows and columns, specify how many columns an element should span.

Name- Khushi Tiwari
Roll No.- 32
PRN- 1032231420

DATE
Pg. No.

- The grid is flexible and fluid, allowing elements to resize, stack, or align based on screen width.

Adaption to Different Screen Sizes:
Bootstrap's grid system is responsive because it uses breakpoints (predefined screen width ranges):

Class Prefix	Targets Device Size
.col-	Extra small (< 576px)
.col-sm-	Small (≥ 576px)
.col-md-	Medium (≥ 768px)
.col-lg-	Large (≥ 992px)
.col-xl-	Extra large (≥ 1200px)

When the screen sizes changes:
- Columns stack vertically on smaller screens.
- Columns sit side-by-side on larger screens.

4> Differentiate between Tag, class, and ID selectors.

Ans → Selector	Syntax	Description	Example
Tag selector	tagname	selects all elements of the specified HTML tag.	p { color: blue; } → changes text color of all <p> elements.
Class selector	.classname	Selects all elements with the specified class attribute; can be applied to multiple elements.	.highlight { background: yellow; } → applies yellow background to all elements with class = "highlight".
ID selector	#idname	Selects a single element with the specified unique ID attribute; should be unique within the page.	#main { font-size: 20px; } → applies font size only to the element with id = "main".

DATE
Pg. No.

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

Ans →

Method	Description	Example	When to use
① Inline CSS	CSS styles are applied directly to a specific HTML element using the style attribute.	Hello	When styling only one element or making quick changes.
② Internal CSS	CSS rules are placed inside a <style> tag within the <head> section of the HTML document.	html <h><head> <style> p { color: blue; } </style> </head>	When styling a single HTML page without affecting others.
③ External CSS	CSS rules are written in a separate .css file and linked to the HTML document using <link>.	html <h><head> <link rel="stylesheet" href="style.css"> </head>	When styling multiple pages with the same design for consistency.

Problem statement → 3. Event Registration form (Roll No. 25 to 36).

20/1/25

Conclusion-
Thus, learned about various HTML tags their scope, syntax and usage, come to know about bootstrap, various methods of CSS styling and hence implemented a responsive web design using the concepts used.

Name- Khushi Tiwari
RollNo.-32 PRN-10322231420
Batch-A1
2/09/25 Program- TY CSE CSF

Web Technologies Laboratory-2

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 character requirement for password, regular expressions for certain format 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 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 (regex) are patterns used to match, search, and manipulate text.
They are suitable for validating data formats (like phone numbers, emails, dates, etc.) because:
① They define strict rules for how the text should look (eg, digits only, fixed length, special characters).
② They can quickly check if input follows a specific pattern.

For passwords, regex can check the presence of required

A React Portal allows you to render a child component into a DOM node outside the main parent hierarchy (root).

Use Cases:
- Modals / Popups
- Tooltips
- Floating elements that needs to visually "escape" container boundaries.

Example:
```
ReactDOM.createPortal(
  <ModalContent />,
  document.getElementById("modal-root")
);
```

Q5) Discuss the importance of Error boundaries in React.

Ans → • Definition:- Error boundaries are React components that catch JavaScript errors in their child components and display a fallback UI instead of crashing the whole app
• Importance:
Improves robustness by preventing app wide crashes.
• Provides better User experience with custom error messages.

Example :-
```
class ErrorBoundary extends React.Component {
  constructor(props) {
    super(props);
    this.state = { hasError: false };
  }
  static getDerivedStateFromError() {
    return { hasError: true };
  }
}
```

the browser before submission, helping users quickly fix mistakes (like missing fields or wrong formats) without sending a request to the server. However, client-side validation alone cannot be trusted because it can be bypassed by disabling JavaScript. Server-side validation ensures that only properly validated and safe data is processed or stored, protecting the application from malicious input.
For example, in a payment form, if only client-side validation is used, an attacker could disable JavaScript and inject malicious input like SQL code, leading to a serious security breach.

(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 a form submission.

Ans → Example →

```
<!DOCTYPE html>
<html>
<head>
  <title>DOM Interaction Example</title>
</head>
<body>
  <form id = "myForm">
    <input type = "text" id = "nameInput" placeholder =
                                "Enter your name" required>
    <button type = "submit"> Submit </button>
  </form>
  <p id = "message"></p>
```

Q3) What are the different features of JavaScript?

→ Key features include:

(a) Light weight and Interpreted → runs directly in the browser without compilation.

(b) Object-oriented → supports objects, prototype and Inheritance.

(c) Event-driven → reacts to User actions like clicks or keypresses.

(d) Asynchronous support → handles tasks with callbacks, promises and async/await.

(e) Cross-platform - works on all major browsers and devices.

(f) Dynamic typing - variables can hold different data types at runtime.

Problem Statement:

(1) Write a program to design Student registration form by using HTML, CSS having following fields: Username, Email, Phone number, Password, Confirm Password and write external javascript code to achieve following. (Implemented in VS Code)

(2) Write a client-side script with JavaScript to access and manipulate Document Object Model (DOM) objects in an HTML web page. Develop a dynamic web page using java Script and DOM. Make use of the following for accessing elements.
(Implemented in VS code)

04/12/25

Name - Khushi Tiwari
Roll No. - 32
PRN - 1032231420

ASSIGNMENT-3

FSD

- Aim - Design an interactive frontend 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 frontend 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 events, ensuring a seamless user experience across various devices and screen sizes.

- Theory -

Q(i) 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 a component-based application.

Ans → State ⇒ State represents mutable data that is managed inside a component.
- It allows components to create and update their own data over time in response to user actions, network responses or other events.
- State updates cause the component to re-render and reflect the changes in the UI.

breaks the UI into small reusable components (eg Navbar, footer, profile card).
- Components act as templates that can accept props and display dynamic data.

Why its superior to traditional HTML:
- Promotes reusability and reduces code duplication.
- Easier maintenance since UI is modular
- Encourages scalability for large applications.
- Facilitates dynamic updates with state management.

Q4) 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 the component's state.

Ans → ⇒ Events in React are similar to Javascript DOM events but use camelCase syntax.
- We can attach an event handler function to an element and update state.

Example: Button Click Counter

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

function Counter(){
  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>
  );
}

export default Counter;
```

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

Ans → Definition : Responsive Web Design (RWD) ensures a website adapts to different screen sizes and devices (desktop, tablet, mobile).

Importance : • Improves user experience across all platforms.
• Essential for accessibility and SEO.
• Reduces the need for multiple device-specific versions.

Implementing responsiveness in React.

① CSS Media Queries

```
.container {
  width: 100%;
  padding: 20px;
}

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

Example :- A counter value that increases when the user clicks a button.

Props :- Props (short for properties) are immutable data passed from a parent component to a child component.

- They are used for data flow and make components reusable by supplying external values.
- Example :- Passing a username from a parent component to a child component for display

- Difference

Aspect	State	Props
Ownership	Managed inside a component	Passed from parent to child
Mutability	Mutable (can be updated)	Immutable (read-only)
Purpose	Handles dynamic, local data	Passes data/config between components

Q2] What is a React component ? Differentiate between a class component and a functional component and discuss the advantages of using a functional component with hooks like use state and use effect over a class component.

Ans → A react component is a building block of a React application that represents part of the UI. Components are reusable, independent and modular class components.

- Written using ES6 classes.
- Uses this.state and this.setState() for state management.

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

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

⇒ React Router allows navigation between pages without refreshing the browser.

- It turns a react app into a single page application (SPA) by dynamically updating the view when the URL changes.

Benefits:

- faster navigation
- Smooth user experience
- Route-based rendering

Example - import {BrowserRouter, Route, Routes} from "react-router-dom";

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

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

⇒ ① CSS stylesheets - Traditional .css files imported into components.

④ CSS In-JS (styled components / Emotion)

```
Import Styled from "styled-components";
const Box = Styled.div
  width : 100%;
  padding : 20 px;
@ media (max-wid : 768 px){
    padding : 10 px;
    font-size : 14px;
  };
```

③ UI libraries (Bootstrap, Material UI, Tailwind CSS)

- These libraries provide built in responsive classes.

Problem Statement : (Roll No. 21 - 40) (Implemented in VS code)

Conclusion :

This exercise demonstrates the core principles of React development.

- components make the UI modular and reusable.
- state and props enable efficient data flow and dynamic updates.
- Event handling makes applications interactive.
- Responsive design ensures accessibility across devices.

26/9/25

Name- Khushi Tawani
Roll No. - 32
PRN - 1032231420

ASSIGNMENT-4

(Batch-A1)

- Aim: Enhance web page developed in earlier assignment by rendering lists and rentals, Event 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?

- Lists: In React, lists are used to render multiple items dynamically from an array using the map() function.
- Keys: Each list item must have a unique key prop, which helps React identify which items have changed, been added, or removed.

- Purpose: Keys improve rendering performance and avoid bugs during re-renders.

Example:-
```
const items = ["Apple", "Banana", "Cherry"];
<ul>
  {items.map((fruit, index) => (
    <li key={index}>{fruit}</li>
  ))}
</ul>
```

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

```
<script>
const form = document.getElementById("myform");
const message = document.getElementById("message");
form.addEventListener("submit", function(event) {
    event.preventDefault();
    const name = document.getElementById("nameInput").value;
    message.textContent = "Hello, " + name + "! Welcome!";
});
</script>
</body>
</html>
```

- How it works:

→ The user enters their name in the form. Then on submission, JavaScript prevents the page reload (event.preventDefault()). It takes the input value and updates the <p> element with a greeting.

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

flow → ① Frontend (HTML, CSS, JS) → Collect user input using forms and JavaScript.

② Send Request → Use AJAX / Fetch API in JS to send the data to the backend.

③ Backend (eg, PHP, Node.js, Python/Django, etc.) → Receive the request, process it, and connect to MySQL.

④ Database Query → Backend runs SQL queries (INSERT, SELECT, UPDATE, DELETE) on the MySQL database.

⑤ Send Response → Backend sends the result (eg success message or data) back to the frontend.

- Uses lifecycle methods like componentDidMount().

```
class Welcome extends React.Component {
  constructor(props) {
    super(props);
    this.state = {count: 0};
  }
  render() {
    return <h1>Hello, {this.props.name}!</h1>;
  }
}
```

Functional component →
- written as simple Javascript functions.
- Use Hooks (like useState, useEffect) for state and lifecycle management.

```
function Welcome({name}) {
  return <h1>Hello, {name}!</h1>;
}
```

- Advantages of functional components with Hooks
  → Cleaner and more concise syntax.
  → No need for this keyword.
  → Better performance optimization.
  → Hooks provide powerful features like state (useState) and side effects (useEffect).

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 → Concept: Instead of writing one long HTML file, React

② Inline styles :- Directly style elements using JavaScript objects.

③ CSS Modules :- Scoped CSS that avoids naming conflicts ~~conflit~~

④ CSS-in-JS Libraries :- Styled components, emotion (styles written in JS).

⑤ UI frameworks - Bootstrap, material-UI, Tailwind CSS for prebuilt responsive designs.

Conclusion :- By using hooks and keys, Portals, Error boundaries, React Router and different styling techniques, React applications become :
- More dynamic (Hooks)
- More flexible (portals)
- More reliable (Error boundaries).

26/9/25

Name- Khushi Tawani
Roll No. - 32
PRN - 1032231420

[Assignment - 5]

FSD

- 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 from Node.js?

Ans → Express.js is a minimal and flexible web framework built on top of Node.js. It provides powerful features such as routing, middleware support, request and response handling and template rendering engines which makes backend development faster and easier which instead of writing complex server code with just Node.js, developers use Express.js to build RESTful APIs, full-stack applications and scalable web servers with less effort and more structure.

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

Ans → CRUD stands for Create, Read, Update and Delete which are the four basic operations needed to manage data in any application.
- Create → Add new data (eg register a new user)
- Read → Retrieve existing data (eg view product details)
- Update → Modify existing data (eg change user profile info)
- Delete → Remove data (eg delete a record)

In a web application CRUD is usually implemented through API endpoints (like /create, /update) that interact with the

database. These operation form the backbone of most dynamic applications.

Q3) Why is MongoDB suitable choice for this project?

Ans→ 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 development. MongoDB also supports scalability, high performance, and easy integration with Express.js through libraries like Mongoose.

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

Ans→ ① Develop locally → Build your app with express routes & MongoDB integration.
② Version control → Push your code to GitHub or another repository.
③ Choose Hosting → Use platforms like Heroku, Render, AWS or Digital ocean.
④ Install Dependencies → Run npm install on server.
⑤ Configure environment variables → set up MongoDB connection string, port, etc.
⑥ Run app → Start using node server.js or a process manager like PM2.
⑦ Test Deployment → Ensure routes and database connections work correctly online.

26/2 Conclusion :- This assignment helped in understanding the importance of responsive web design and its implementation in full-stack applications. By using Express.js for backend, MongoDB for data storage, and responsive front-end design techniques, we created a seamless CRUD-based application that works across multiple devices.

⑥ Update UI → JavaScript updates the webpage dynamically based on the response.

Conclusion: Frontend (HTML, CSS, JS) handles user interaction, but it cannot directly connect to MySQL. Instead, it sends data to the backend (like PHP, Node.js or Python), which performs the database operations and returns the results.

FAQ:

Q1) Write 3 reasons why Form validations are important.

Ans → ① Ensures Data Accuracy and Consistency - Form validation makes sure users enter information in the correct format (like valid email, contact), which keeps the data clean, reliable and useful for further processing.

② Improves security - By checking and filtering inputs, validation prevents attackers from injecting harmful code (like SQL injection or XSS), protecting both the application and database.

③ Enhances User experience - Validations provide instant feedback when a user makes a mistake, helping them correct errors quickly and ensuring smooth interaction with the application.

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

Ans →
```
<img id = "myImage" src = "old.jpg" alt = "picture">
<script>
    document.getElementById("myImage").setAttribute
                                ("src", "new.jpg");
</script>
```
This changes the src attribute of the image dynamically using the DOM.

elements (uppercase, digits, special characters) in a single, efficient rule.

(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 → In web application development, a cookie is stored on the client's browser and holds small pieces of information, such as a session ID, while a session is stored on the server and contains the actual user data like login details or preferences. When a user logs in, the server creates a session and assigns it a unique session ID. This ID is then stored in a cookie on the user's browser. For every subsequent request, the browser automatically sends this cookie back to the server, allowing the server to identify the session and retrieve the user's data. In this way, cookies and sessions work together to maintain a user's logged-in state across multiple 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 → The purpose of performing both client-side and server-side validation is to ensure better user experience as well as security. Client-side validation checks data in