Assignment :- 01

⊙ Aim :- Develop responsive web design using HTML5. containing a form, style the pages using CSS, use of tag selector, class selector and id selection. use Inline, Internal and Eternal CSS. Apply Bootstrap CSS.

⊙ Objectives.
> To understand HTML tags.
> To learn the styling of web pages using CSS.
> To learn Bootstrap front End framework.

⊙ Theory
Q. Define Responsive we Design (RWD) - what is its. Primary goal?
→ Responsive web Design. (RWD) is a web development approach that creates websites able to adapt and respond to different screen size and devices. (like phones, tablets and desktop).

Primary goal:
To ensure a seamless user experience by making the. layout, context and images, adjust automatically to fit the screen, without requiring users to zoom or scroll horizontally.

Q. Explain the role of the <meta name = "Viewpt".> tag. why is this tag essential for RWD?
→ The <meta name = "viewport"> teg tells the browser. how to control the page's dimensions and Scaling. for example. <meta name = "viewport" content = "width = device = width. initial = scale = 1.0">.

Without this tag mobile browsers display web pages with a desktop width viewport, making the content too small to read. This tag enables proper scaling and layout adjustment for mobile devices, which is a core requirement for responsive design.

Q. How does Bootstrap assist in creating a responsive layout?

→ Bootstrap is a popular front end framework that makes it easy to build responsive websites using predesigned components and Flexible grid sys. Grid System, explaination.

Bootstrap's grid system divides the pages into 12 columns you can combine these columns in diff. ways using classes like col-, col-sm-, col-md-, col-lg- and col-xl- to control how much space each element takes at various screen sizes.

- col-sm-6 means the elements takes 6 column. (half width) on small screens and above.

→ The grid automatically stacks or resizes columns to fit different screens, ensuring responsiveness

Q. Different betn tag, class and ID selectors.

⇒ Tag selectors targets all elements at a specific type: example p{color: blue;} changes clor of all <p> tags. close selector; targets elements targets a specific targets elements that share the same class attribute.

eg: box{padding: 10px;} -

applies styles to any element with class = "box".

Page No.: YOUVA
Date:

ID selectors: targets a specific elements with a unique ID

eg: #header { background: gray; } - applies styles to the elements. with. id = "header".

⊖ Conclusion.

In this, we learned how to build responsive web pages using Html, Css & Bootstrap. I understand the diff css selectors. with. adept to all screen size.

31/2/2025

Name:- Nikhil Pankaj Patil
Roll No:-60.
PRN:-1032232897

Page No. :
Date :

FSD [Assignment:-2].

⊙ 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 characters required for password, regular expressions for certain format of the file etc. Use the mysql database

⊙ Objectives:-
i) To understand what form validation
ii) To learn basic functioning of Dom objective.
iii) To learn how to apply various techniques to implement it.

⊙ Theory:-
i) Explain the role of regular expressions, why are they a suitable tool for validating data formate like a phone numbers or checking for the presence of specific characters in a password
→ Regular expressions (Regen) are patterns used to match and manipulate text. They are suitable for validating data formats like phone no. or checking password rules because they can
- Define precise patterns
- Quick verify if input matches required format.
- Detect presence labsence of specific characters

This makes regen a powerful and efficient tool for input validation.

Page No.:
Date :

ii) Explain the fundamental difference betⁿ a session and a cookie in the content of web application development. How do they work together to maintain a user's logged in state?

→ Difference.
- Cookie - Stared on the clients browser helds. small pieces of data.
- session:- Stored on the server holds actual user-related data.

Working together.
when a used logs in the servers.
i) use, logs in
ii) Server, sends a session ID to client.
iii) On subsequent requests, the browser sends. the cookie.
iv) The server verifies it and retrieves the session.

iii) 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 vulb.

→ Purpose.
- Client side validation - improves user exp. by catching errors early.
- Server side validation - ensures data integrity and security.

Page No. :
Date :

Vulnerability scenario.
If only client-side validation cheks form input (eg- email format) an attackers can bypass it by disabling js or sending requests directly to the servers, injecting maliciouces data (like sql injection).

iv) provide a simple example of how a js script can interact with the Dom to dynamically charge the content of a webpage often, such as form submission?

→ Ex.

```
<!DOCTYPE HTML>
<HTML>
<body>.
  <form onsubmit="Update msg (); retur fab;"
<input type = "Text" id="name" placeholders="
  Enter your name">.

<button type = "submit > submit </button>
<p id = "message ></p>.
<script>.
  funⁿ update. msy ( ){.
let name = documents + (Element By Id.
value document get Element By Id("msg"
innertext= "Hello", + name + ")"!
}.
  </script >
</body>.
</html>.
```

Page No. :
Date :

when the form submitted - It updates the <p> content dynamically.

iv) Give the steps for connectivity from front end using HTML CSS JS to mysql

→ Steps.
i) frontend from / UI (HTML (CSS / JS).
ii) Send data vig JS (Fetch / AJAX).
iii) Backend (PHP / Node / python) handle req.
iv) Backend connects to my SQL.
v) JS updates UI.

⊙ FAQ's.

Q.1 → i) Ensures correct and complete data entry
ii) prevents invalid / malicious inputs.
iii) Improves user exp - with instant feedback.

Q.2 ⇒ Ex.
document.getElementById("myImg").setAttribute("src", "newimage.jpg");

Q.3 ⇒ ~~what~~ Light weight, interpreted and object based.
→ supports event driven and fun^ programming.

19/9/25

FSD.

Assignment :- 03.

⊙ 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 front-end app. using React.js that effectively demostrates the fund' 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 interaction with events

⊙ Theory :-
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.
→ State :- State represents mutable data is mangged inside a component.
→ It allows components to create and update their own data ever time in response to user actions network responses or other events.
eg. A counter Value that increases when the user clicks a button.
props :- props are immutable data passed from a parent component to child component.

→ They are used for data flow and make component reusable by supplying external values.
→ Eg:- 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 betn components

ii) What is React component ? Differentiate betn a class component and a functional Component and discuss the advantages of using a funn component with hooks like use state. and use state and use effect over a class components.
→ Areact component is a building block of a. react application that regpresents part of the UI.
Componenents are reusable, independent and modular.
class components.
→ written using ES6 classes.
→ uses this, state and this setstate() for state.
→ uses lifecycle. methods. like components Did mount()
management Class welcome extends React.component{
Constructor(props){
super(props);
this.state={count=0};
}
render(){.

Why its Superior to traditional HTML.
→ promotes reusability and reduces code duplication.
→ Easeir maintanance since UI is modular.
→ Encourages sealability for large applications.
→ facilities dynamic updates, with State manageyement.

4. How do you handle user events in React (eg a button click?
→ Events in React are similar to js Dom events. (e
but use camel case syntax.
→ we can attach an event handless. funn to an element and update state.

eg- Button click counter.

```
import React, {use state} from 'react';
funn counter(){.
const{count, set count}= use state(0);
const handle click=()={
    set count+1);
};
return(
  <div>
  <p> you clicked {count} times </p>.
<button on click ={handle click} >clickme</button
</div>.
);
}.
export default Counter;
```

```
    return <h1> Hello, {this.props.name}! </h1>;
  }
}
```

functional components.

→ written as simple JS. fun^n.
→ use Hooks (like use state, use effect) for state.
and lifecycle management fun^ 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 opt^n.
→ Hooks provide powerful features like state and side effects. (use effect).

3. Describe the concept of templating using componed in React why is this approach - consider superiror to traditional web development methods that rely on monolithic HTML files?

→ Concept: Instead of writing one long html files. React breaks the UI into small reusable comp. eg. Nabar, footer, Profile(card).
→ Components acts as templates that can accept. props and display dynamic data.

5. 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.

→ Define.

Responsive web design (RWD) ensures a website adapts to different screen sizes and devices (desktop, tablet, mobile)

Imp:

Improve user exp across all platform.

css media queries.

```
Container{
width: 100%;
padding: 20px;
}
```

⊙ Conclusion.

This exe demo the core principle of React. & UI modular and resuable.

10/10/25

FSD.

Assignment :- 04.

⊙ Aim
Enhance web page developed in earlier assignment by rendering lists and portals, errors handling, Routers and style with React css also make it a responsive design to scale across pc. tablet. and mobile phone.

⊖ Objective.
Enhance user interface and experience.
Improve Application Robstness and navigation.

⊖ Theory:

1. 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() fun^n.
Keys.
Each list item must have a unique keys. prop. which helps React identity which have changed., been added or removed.
Purpose. - Keys improve, rendering performance and. avoids bugs during re-renders.
eg. Const items = ["Applie", "Banana", "Cherry"];

```
<ul>
{items.map.(fruit, index) => !
  < li key = {index} > {fruit} </li>.
  )}.
</ul>;
```

2. What is a React partial and when could you use one?

→ A React portal allows you to render a child component into a DOM node outside the main parent hierrchy (root). use cases.

- models / popups.
- Tootlips.
- floating elements that need to visually "escape".

Example:-

```
React Dom, create portal (
    <modalcontent/>.
  document.get element By Id ("model-root").
);
```

3. Discuss the importance of error Boundaries in. React.

→ Definition:-

error boundaries are react components that catch javascript errors in their child components and. display a fallback UI instead of crashing the app.

Importance.

Improves robustness by preventing app wide. washes provides better user experencience with. custom error msgs.

eg. class error Boundary extends React compare{

```
constructor(props) {.
    super(props);
(this.state={has error:false};
}.
```

```
static get Derived State from error () {.
    return { has error: true };
  }.
  render () {.
  if (this.state.has (error) return <h2> Sorry.
wrong! </h2>
    return this.props.children.
  }
}.
```

4. How does React Router enable Single page app. functionality?

→ React Router allows navigation betn pages w/o 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 exp:

Route based rendering.

eg: import { Router, Route, Routes } .
    from "react-router-dom";
<Browser Route>.
<Routes>
<Routes path = "/" element = {<Home /> } />.
<Route path = "/about" element = {<About /> } />.
</Routes>
</Browser router>.

5. Explain the different ways to style a react app.
→ css stylesheets.
traditional .css files imported into components.
Inline styles.
Directly style elements. using js objects.
Css modules
scoped CSS that avoids. naming conflicts.
css in js libraries.
Styled components, emotion (Styles written in Js).
·UI frameworks.
Bootstrap, material ·UI, tailwind CSS for pre. built responsive deslgns.

⊙ Conclusion.
By using lists, and keys, portals, errors Bound React Router. and different styling techniques. React applications become;
∴ more dynamic (lists).
> more visually appealing and responsive
etc.
These enhancements, collectively improve both user experience and application reobustnets.

10/10/25

FSD.

Assign - OS.

⊙ Aim

Develop a responsive web design using express framework to perform CRUD oper^n and deploy with Node.js use mongo DB.

⊙ Objectives.

Develop a full stack web Application.
Demostrate Backend deve-lopmend and deploy it..

⊙ Theory:

i) What is the role of express. Js. as a web. framework for Node.js?

→ express Js is a minimal and flexible. web framework built for top of Node. JS. It provides powerful features such as routing. middleware support request and response handling and template rendering which makes backened development faster and easier instead of writing complex server code with just node.j's developer. use express.js to build Restfull APIs.

iii) Explain the concept of CRUD op^n in the Context of a web applications.

→ CRUD stands for create, read, update and delete. which are the four basic op^n needed to manage data in any application.

→ Create → Add new data.
→ Read → Retrive existing data.

> update → modify existing data.
> Delete → Remove data.
In a web application CRUD is usually implement through API endpt that interact with the dB.

3. Why is MongoDB suitable choice for this projet?
→ MongoDB is a document- oriented NoSQL dB that stores data is JSON like format. This makes it very natural to use with JS and nodejs. It's flexible Schema allows developers to store different type of data without needing a fixed structure. which speeds up development. MongoDB also supports. Scalability high performance and easy integration with express js. through librabries like mongodB.

4. What steps are involved in deploying a nodejs and express application?
→ Deploying a node.js and express apption. typically
i) Develop locally. →
Build your app with experts routes & mongoDB.
ii) Version control →
push your code to git or another resposity
iii) Choose hosting →
use platforms like heroker, render Aws & DO.
iv) Install Dependencies →
Run npm install or the server.
v) Ranfigure environment variable →.
Set up mongodB connection string.
port etc.

> Run App →
Start using node server.js or a process manger like PM2.

> Test Deployment →.
Ensure routes and dB connections work.
Correctly online.

⊙ Conclusion:-
express.js plays a crucial role in simplifying server-side development while CRUD operations. form the foundation of dynamic applications. mongoDB is well-being suited due to its. flexibility. JSON Based storage and scalability Deploying on express app involves preparing the code, hosting, configure DB and run efficiently.

10/10/25