FSD Assignment-1

Rithin Shaju
1032231269.

Aim: Develop responsive web design using HTML 5, containing a form. style the pages using CSS, use of tag selector, class selector & id selector. Use inline, Internal & external CSS, Apply Bootstrap CSS.

Objectives: 1. To understand HTML tags.
2. To learn the styling of web pages using CSS.
3. To learn Bootstap Front end Framework.

Theory: 1. Responsive Web Design (RWD):
Design approach where web pages adjust layout & content auto-matically based on the devices screensize & Orientation.

2. <Metaname = "Viewport" --> tag:
Role: Tell browsers how to scale & display the page on different devices essential because, without it pages may not scale coredtly on mobile devices which would break responsiveness.

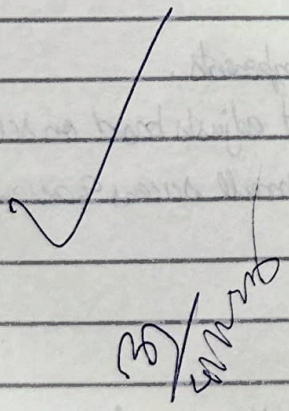
3. Bootstrap & Grid system:
Bootstrap: A CSS framework with prebuilt responsive components.
Grid system: Uses 12-column layout with classes that adjusts based on screensize.
Adaptation: Automatically stacks or resizes elements for small screens & arranges them in rows for larger screens.

4. Tag, Class & ID Selector:
- Tag Selector: Selects all elements of a given HTML tag.
Class selector: Selects one or more elements that share the same class names can be reused across multiple elements.
ID selector: Selects a single unique element with a specific ID. IDs must be unique.

5. Three main ways to apply for CSS:
- Inline CSS - Applied directly inside the HTML tag using the "style".
- Internal CSS - Written inside a <style> tag within the <head> section of an HTML file.
- External CSS - Stored in a seperate .css file & linked with it.

Internal & Inline can be used for small single paged projects, it is the best for large pages. Keeps style seperate from HTML, allows reuse across multiple pages.

- Conclusion: This assignment thought us to build a styled & responsive HTML page.



FSD - Assignment 2

Rithin Shaju
1032231269

Develop a web application using java script to implement sessions, cookies, DOM. Perform validations such as directing for emptiness, only number, special character requirement for password, regular expressions for certain format of the fields etc. Use the My SQL database.

Objectives : 1. To understand what form of validation is.
2. To learn basic functioning of DOM objects.
3. To learn how to apply various techniques to implement it.

Theory:

Q1. Explain the role of regular expressions. Why are they a suitable tool for validating data formats like a phone number of checking for the presence of specific characters?

Ans. Regular expressions (regex) define patterns for matching strings making them perfect for validating formats like phone number or passwords. It can check if a phone no has 10 digits or the password has any uppercase or special characters.

Q2. Explain the fundamental difference between a session & a cookie in the context of full stack dev. How do they work together to maintain a user logged in state?

Ans. A cookie is a small text file stored on the users browser by the web application. It typically contains minimal data such as unique session identifier or user preferences & is sent back to the server with every HTTP request.
A session is a server side storage mechanism that holds user specific data across multiple request for example user authentication.

Q3. What are the different features of Java script?

Ans. * Client side scripting : Runs in the browser to create dynamic
* Event driven : Responds to user interactions like [redacted] or submissions.
* Light weight & interpreted : No compilation needed, browsers can Javascript code directly.
* Cross platform compatibility : Works across different browsers & systems without modification.
* Object oriented programming: Supports objects prototypes inheritance encapsulation for better code organisation.

* Built in libraries & API's : Comes with many built in funtions & API's for manip
DOM handling even HTTP requests & more.
* Dynamic typing : Vaiables can hold any type of data & datatypes during execution
Java scipt for error prone.

• Conclusion: Understanding key concepts such as requests, expressions, sessions & cookies, fo
DOM manipulation backend connectivity is essential for building secure efficien
applications. Combining client side & sever side techniques ensues smooth
experiences & robust security.

Q. Purpose of client-side & Sever-side Validation.
Ans. Client-side (Java Scipt / DOM validation)
- immediate feedback to user
- reduces sever load
- ex- showing 'phone must be 10 digits' instantly.

• Sever-Side (PHP, Python, Node, Java etc)
- Final line of defense.
- Protects against malicious input or bypassing of client-side scipt
A haeker disables Java Scipt in the browser & subtracts invalid
malicious SQL Code in the form → leads to SQL injection attac

Q. Provide a simple example of how Java Scipt can interact with H
to dynamically change the content of a web page after a use
such as a form submission.

Ans. <P ID = "message"> Hello ! </P>)
<button on click = change text()"> click me

</button>
<Script>
function changeText() {
document.getElementById("message").inner
HTML = "You" }
</Script>

FSD - ASSIGNMENT 3

Ruthin Shaju
1032231269.

Design an interactive front end application using React by implementing templating using components states & props, class, events. It must be responsive to scale across diffuent platforms.

tive: To develop a responsive interactive front end application using React. is that effectively demonstrate the fundamental concepts of component based architecture state management & 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 & props & handling user interaction with events ensuring a seamless user experience across various devices & screensize.

y:

Q1. Explain the role of state & props in react. How do they differ & what is the primary purpose of managing data flows within a component based application?

Ans. States:- Managed inside a component
- Mutable: can be updated with set state or use state.
- Used to store dynamic data that changes overtime.

• Properties:- Passed from parent to child components
- Immutable: cannot be modified by the child.
- Used for data sharing & configuration
ex: a button label passed from a parent.

• Key difference:- Props: Internal, controlled by parent, read only.
- States: Internal, controlled by component, write available to edit.

• Primary Purpose:- Props: enable component reusability & data flow.
- State: manage interactive dynamic behaviour with in a component.

Q3. Describe the concept of "templating using components" in React. Why is this approach considered superior to traditional web development methods that relyon monolythic HTML files?

Ans :. Instead of one giant HTML file, React breaks the UI into small, reusable components.
• Each component manages its own logic & style.
• ex: A blog page can be composed of <header />, <post> , <footer /> function Post ({title, content})

```
return (
  <div>
   <h2>{title}</h2>
   <p>{content}</p>
   </div>
   );}
```

Q4. How do you handle user events in React? Provide a simple to demonstrate how an ev
is defined in a component it can be used to update the components state.

Ans. Events are handled with camel case & by passing functions not state.

```
ex: import {useState} from "react";
function clickExample(){
  const [message, setMessage] = useState("Hello");}
  function handleClick(){
  setMessage("You clicked the button");}
  return (
  <div>
  <p>{message}</p>
  <button onClick={handleClick}>Click Me</button>
  </div>);}
```

Q5. Definition: Responsive design makes a website adapt to different (desktop, tablet, mobile
It ensures usability, accessibility & better user experience.
CSS Media Queries: adjust layout based on size.
CSS in JS: Write responsiveness styles directly in components.

- Conclusion: Learnt to develop a responsive, attractive front end application using React.js, by
the code.

2. What is react component ? Diff b/w a class component functional component.

A react component is a reusable, independant piece of UI, like card or forum.

- Class components.
  - written as ES6 classes
  - uses render() method to return JSX
  - can use the state & life cycles methods
- Functional components:
  - written as plain functions, returning JSX.
  - use hooks like use state, use effect for state & side effects.

Advantages of functional components with hooks:
- Simpler, short syntax.
- No need to worry about this keyword.
- hooks like use state, use effect, use context give access to state & life cycles in a clean way.
- Easier to test & optimize, has smaller reusable logic with custom hooks.

28/10/25

FSD - Assignment 4

Ruthin Shaju
1032231269

M: Enhance web page developed in earlier assignment by rendering lists & Portals, Error handling, routes & style with react CSS also make it a responsive design to scale well across PC, tablet & mobile phone.

Objectives: 1. Enhance user interface & experience.
2. Improve application robustness & navigation.

Theory:
1. How do lists & keys work in React?
* Keys: You render list items should have a unique key to help React identify which items have changed. This improves rendering performance.
* Lists: You render lists by mapping over an array & returning JSX for each item.

2. What is react Portal & when would you use one?
3. React Portal allows you to Render children into a DOM node outside the current component hierarchy. This is useful when you need to break free form the confines of the components DOM structure, such as models, tooltips or dropdowns where the component DOM structure, stacking should visually be rendered else where.
When to use a React Portal:
- Modals: Often need to be rendered outside of the main layout for z-index or styling reasons.
- Tooltips: To ensure they're not constraint by parents containers or other overflow issues.
- Dialogs: Similar to modals, they often needed a dedicated space in the Dist.

Q3. Discuss importance of of errors in react.
Ans. Error boundaries are react components that catch Java script errors child components tree, log those errors & display a fallback UI, inst of crashing the whole app. Without error boundaries, If an error occurs in there part of the APP, it could unmount the entire component tree.
Why are they important:
- Prevent crashes: Without them a single component could crash the entire app.
- Graceful degradation: Fall back UI allows users to still navigate & functions.

- Error logging: Error boundaries allow you to log errors to external like sentry or log improve the apps stability over.

ex: Class ErrorBoundary extends React.component {
constructor(props) {
super(props);
this.state = { hasError: false };
}
static getDerivedStatefromError(error) {
return { hasError: true };

Q4. How does react router enable single page application (SPA) functionality?
Ans. React router allows navigation b/w pages without refreshing turns a react app into page application (SPA) by dynamically the view when the URL changes.
Benefits:- • Faster navigation
• Smooth user experience.
• Route based rendering

Q5. Explain the different ways to style a react application.
Ans. * CSS stylesheets: Traditional CSS files imported into components.
* Inline styles: Directly style elements using Javascript objects.
* CSS modules: Scoped CSS that avoids naming conflicts.
* CSS in JS libraries: Styled component.

• Conclusion: By using lists & keys, portals, errors boundries, styling techniques they help
- More dynamic
- More flexible
- More user friendly
- More visually appealing & responsive.

These all together improve the user experience.

20/10/2025

FSD - Assignment 5

Rithin Shaju
1032231269

: Develop a responsive web design using express framework to perform CRUD operations & deploy with NODE.JS use Mango DB.

jectives: 1. Develop a full stack application.
2. Demonstrate Backend Development & Deployement proficiency.

Theory:

Q1. What is the role of express.JS as a web framework for node.JS?

Ans. Express JS is a minimal & flexible web framework built on top of node.JS. It provides powerful features such as routing middleware support, request & response handling & template rendering, which makes backend development faster & easier instead of writing complex server code with just node.js, developer use expressJS to web severs with less effort & more structure.

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

Ans. CRUD:- C - Create
R - Read
U - Update
D - Delete

These are the 4 basic operations needed to manage data in any application.

- Create: This operation is used to add/Create new data/database
- Read: This operation is used to retrieve/read existing data.
- Update: Used to modify existing data
- Delete: Used to delete data.

Q3. Why is MongoDB suitable choice for this project?

Ans. MongoDB is a document oriented NOSQL database that stored different types of data without needing a fixed structure which development MongoDB also supports scalability, high perform & integration with Express.js. Through libraries like MongoDB. This it a perfect choice for modern web applications where requirements may change frequently.

Q4. What steps are involved in deploying a Node.js & express application.

Ans. Deploying a Node.js & express application typically involves these steps:-

* Develop locally: Build your app with express routes & Mongo DB integration.
* Version control: Push your code into GitHub or another repository.
* Choose Hosting: Use Platforms like Heroku, render, ASUS or digital code.
* Install Dependencies: Run NPM install on the servers.
* Configure environment variables: Set up Mongo DB connection port etc.
* Run App: Start using node server.js or a process manager like PM2.
* Test Deployment: Ensure routes & DB connections work correctly online.

- Conclusions: Express.js plays a crucial role in simplifying server side development while operations form the foundation of dynamic applications Mongo DB is well suite its flexibility. JSON Based Storage & scalability. Deploying an express a preparing the code hosting it on server configuring the database & running it. together these components enable the creation of a robust, responsive & fu deployable full stack application.

CO 4/11/25