FSD - ASSIGNMENT 1

Name: Sindhura Gulhane
PRN: 1032230397 Roll No: 06
TY-CSF - PANEL A

- 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 Frameworks.
- Theory:

1) Responsive Web Design (RWD):
Design approach where web pages adjust layout and content automatically based on the device's screen size & orientation.
Primary goal: Ensure usability & readability across all devices.

2) <meta name="viewport" ----> tag:
Role: Tell browsers how to scale & display the page on different device
Essential because: without it pages may not scale correctly on mobile devices which would breaks responsiveness
Ex: <meta name="viewport" content = "width = device-width, initial-scale = 1.0" >

3) Bootstrap & Grid system:
Bootstrap: A CSS framework with prebuilt responsive components.
Grid system: Uses 12-column layout with classes (.col, .col-sm-, .col-md-, .col-lg-) that adjusts based on screen size
Adaptation: Automatically stacks or resizes elements for small screens and arranges them in rows for larger screens.

Sundaram
FOR EDUCATIONAL USE
Scanned with OKEN Scanner

4) Tag, Class & ID selectors:

Tag selector: Selects all elements of a given HTML tag. Ex:

```
p { color : red ; }
```

This changes the colors of all <p> tags to red.

Class selector: selects one or more elements that share the same class name. can be reused across multiple elements. Ex:

```
.highlight { background-color : yellow; }
```

Any element with class = "highlight" will get a yellow background.

ID selector: selects a single unique element with a specific ID. IDs must be unique within a page. Ex:

```
#header { font-size : 24 px ; }
```

Only the element with id = "header" will have 24 px.

5) Three main ways to apply CSS:

1) Inline CSS - Applied directly inside the HTML tag using the "style" attribute. Ex

```
<p style = "color : blue ;" > This text is blue.
```

2) Internal CSS - written inside a <style> tag within the <head> section of an HTML file. Ex

```
<style>
  h1 { color : green }
</style>
```

3) External CSS - stored in a seperate .css file & linked with rel = "stylesheet" href = "style.css" > in the

Ex:

```
body { font-family : Arial ; }
```

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

Conclusion: This assignment thought us to build a styled & responsive HTML page using CSS Bootstrap.

Sundaram
FOR EDUCATIONAL USE
Scanned with OKEN Scanner

FSD - ASSIGNMENT 8

Name : Sindhura Gulhan
PRN : 1032230397 Roll:06
TY CSE - PANEL A

- Aim: Develop a web application using javascript to implement sesions, cookies, DOM. Reform validations such as checking for emptiness, only 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 of for validating data formats like a phone number of checking for the presence of ~~steps~~ specific characters in a password?

A Regular expressions (regex) define patterns for matching strings making them perfect for validating formats like phone numbers or passwords. For example: Regex can check if a phone number has exactly 10 digits or if a password contains uppercase letters numbers & special characters. They are highly effecient as they can specify the exact formats & rules [like the structure of a phone number / email].

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

A A cookie is a small text file stored on the user's 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. ~~Cookies are~~

A session is a server side storage mechanism that holds user specific data across multiple requests, for example user authentica

For Educational use

Scanned with OKEN Scanner

status, shopping cart contents or other temporary information. U
user logs in the server creates a session & generates a unique
ID which it sends to the client as a cookie. On subsequent
the browser sends the cookie back allowing the server to retrieve
associated session data & recognize the user thus maintaining th
logged in state securely.

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

A Client side validation happens in the browser before the data is sen
to the server. It provides immediate feedback to users - for example
them if a req. field is empty or if the email format is incorrect. It
user experience & reduces server load.
Server side validation occurs on the backend after the data arri
is crucial for security because client side validation can be bypassed
disabling JavaScript or tampering with requests. This server must
all inputs to protect against malicious data.
Ex: If a website only relies on the client side validation
attacker could disable & attack Javascript directly, bypassing all
This could allow injection of malicious SQL commands or uploading
files leading to data breaches.

4 Provide an example of how JS can interact with the DOM to
-cally change the content of a webpage after user action
a form submission.

A JS can dynamically update a webpage by interacting with
Ex: When user submits a form JS can intercept the subm
prevent default page reload & update content dynamicall
JS:

```
document.getElementById('myForm').addEventListener('subm
(event) { event.preventDefault(); document.getElementBy
```

SAFFRON
We make things happen...
For Educational use
Scanned with OKEN Scanner

• text content = 'Submitted';

5 Given the steps connectivity from front end using HTML CSS to mysql.
A - The frontend (HTML/CSS/JS) sends a request to the backend server using AJAX or fetch API.
- The backend (e.g Node.js) recieves the request & processes it.
- Backend connects to the Mysql database using a DB driver or ORM.
- Backend runs SQL queries to fetch or update data.
- Result are sent back to frontend as JSON or other formats
- The fronted recieves this data & updates web page dynamically.

• FAQ's:

1 write 3 reasons why form validations are important.
A - Prevent incorrect data: Validations ensure users enter data in the correct format, reducing errors & improving data quality
- Enhance user experience: Instant feedback on errors helps users fix mistakes before submitting, saving time.
- Improve security: validations help prevent malicious data from being sent to the servers

2 Example of modifying an attribute value using DOM.
A In Javascript, you can dynamically change an element's attribute using the setAttribute() method. For instance if you want to change the source of an image you first select the element by its ID and then update its src attribute. This changes the displayed image without having to refresh the page.

```
document.getElementById('my image').setAttribute('src', 'new-
                                                    image.jpg
```

For Educational use

Scanned with OKEN Scanner

3 What are the different features of Java script?

A - Client side scripting : Runs in the browser to create dynamic w
- Event driven : Responds to user interactions like clicks or submission
- Lightweight & interpreted : No compilation needed, browsers exec
Javascript code directly.
- Cross platform compatibility : Works across different browsers & op
systems without modification
- Object oriented programming : Supports objects prototypes inheritanc
encapsulation for better code organi
- Asynchronous programming : Handles tasks like data fetch
timers without blocking the main thre
tools promises & async/await.
- Built in Libraries & API's : Comes with many built in functions
APIs for manipulating DOM, handling user
HTTP requests & more.
- Dynamic typing : variables can hold any type of data & c
types during execution making Java script fle
error prone

• conclusion: Understanding key concepts such as reg
expressions, sessions & cookies, form validation, DOM manipulation
backend connectivity is essential for building secure efficient & user
applications. Combining client side & server side techniques ensures
smooth user experiences & robust security.

31
2/9/25

SAFFRON
We make things happen...
For Educational use
Scanned with OKEN Scanner

FSD - ASSIGNMENT 3

Name : Sindhura Gulhane

PRN: 1032230397

Roll No: 06; TYCSF-A

- Aim: Design an interactive front end application using React by implementing templating using components states & 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 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 and handling user interactions with events ensuring a seamless user experience across various devices & screen sizes.
- Theory :

1 Explain the role of state & Props in React. How do they differ & what is the primary purpose of managing data flow within a component based application?

A - States:
- Managed inside a component
- Mutable : can be updated with setstate or [the exclusive states] Usestate.
- Used to store dynamic data that changes over time (like form input counters, toggles)

- Props [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 : External, controlled by parent, read only
- States: Internal, controlled by component, write available to edit.

FOR EDUCATIONAL USE

Scanned with OKEN Scanner

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

2 What is a React component? Differentiate between a class compone functional component and discuss the advantages of using a functio ment with hooks like useState & useEffect over a class component.

A A react component is a reusable, independent piece of UI, like card or form.

- class components:
  - written as ES6 classes
  - Uses render() method to return JSX
  - Can use the state & lifecycle methods (componentDidMount, componentDidUpdate, etc)

Ex:

```
Class counter extends React.component {
state = { count : 0 } ;
increment = () => this.setState ({ count : this.state.count + 1 });
render() {
return (
<div>
<p> count : { this.state.count } </p>
<button onClick = { this.increment } + </button>
</div>
); } }
```

- Functional component
  - written as plain functions, returning JSX.
  - Use hooks like useState, useEffect for state & side effects

Sundaram
FOR EDUCATIONAL USE
Scanned with OKEN Scanner

Ex: import { useState } from "react";

```
function Counter() {
  const [count, setCount] = setState(0);
  return (
    <div>
      <p>count: {count}</p>
      <button onClick={() => setCount(count + 1)}>+</button>
    </div>
  );
}
```

Advantages of Functional Components with Hooks

- simpler, shorter syntax
- No need to worry about this keyword.
- Hooks like useState, useEffect, useContext give access to state & lifecycle in a clean way.
- Easier to test & optimize, has smaller reusable logic with custom hooks

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?

A - 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>
  );
}
```

Sundaram
FOR EDUCATIONAL USE
Scanned with OKEN Scanner

Why better than traditional HTML:

- Reusability: write once, use everywhere.
- Maintability: easy to update just one component
- Dynamic Data: React Renders data driven UI unlike static HTML.

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

A Events are handled with camelcase & by passing functions not st

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> ); }
```

Event is attached onClick, function updates state → react re-render
updates automatically.

5th Ans 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 responsive styles directly in components.

Ex:

```
.container {width: 80%}
@media (max-width: 768px) {.container {width: 100%}}
```

Conclusion: I learnt to develop a responsive, interactive Front end appli
using React, by implementing states, props, class, events.

Sundaram
FOR EDUCATIONAL USE
Scanned with OKEN Scanner

FSD - ASSIGNMENT 4

Name : Sindhura Gulhane
PRN: 1032230397 Roll No.: 06
TY - CSF - PANEL A

- 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: 1) Enhance User Interface & experience
  2) Improve application robustness & navigation
- Theory:

1 How do Lists and keys work in React?

A - Lists : You render lists by mapping over an array & returning JSX for each item.
- Keys: Each list item should have a unique key to help React identify which items have changed. This improves rendering performances.

Ex:

```
const items = ['Apple', 'Banana', 'cherry'];
const listItems = items.map((item) => <li key={item}>{item}</li>);
```

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

A React Portal allows you to render childeren 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 modals, tooltips or dropdowns, where the component 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/Popovers: To ensure they're not constrained by parents containers or other overflow/scrolling issues.
- Dialogs: similar to modals, they often needed a dedicated space in the DOM

Ex:

```
import ReactDOM from 'react-dom';
function Modal(props) {
  return ReactDOM.createPortal(
```

Scanned with OKEN Scanner

FSD - ASSIGNMENT 4

Name : Sindhura Gulhane
PRN: 1032230397 Roll No.: 06
TY - CSF - PANEL A

- 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: 1) Enhance User Interface & experience
  2) Improve application robustness & navigation
- Theory:

1 How do Lists and keys works in React?

A - Lists : You render lists by mapping over an array & returning JSX for each item.
- Keys: Each list item should have a unique key to help React identify which items have changed. This improves rendering performances.

Ex:

```
const items = ['Apple', 'Banana', 'cherry'];
const listItems = items.map((item) => <li key={item}>{item}</li>);
```

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

A React Portal allows you to render childeren 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 modals, tooltips or dropdowns where the component 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/Popovers: To ensure they're not constrained by parents container or other overflow/scrolling issues.
- Dialogs: similar to modals, they often needed a dedicated space in the

Ex:

```
import ReactDOM from 'react-dom';
function Modal(props) {
  return ReactDOM.createPortal(
```

Scanned with OKEN Scanner

```
<div className = "modal">
  <p> { props.content } </p>
</div>,
  document.getElementById('modal-root')
); }
```

3 Discuss importance of errors in react

A Error boundaries are React components that catch JavaScript errors child component tree, log those errors & display a fallback UI of crashing the whole app. Without error boundaries, if an error o part of the app, it could unmount the entire component tree.

Why are they important :

- Prevent crashes : Without them a single child component cou the entire app.
- Gracefull degradation : Fallback UI allows users to still navigate & functions.
- Error Logging : Error boundaries allow you to log errors to exte like sentry or LogRocket to improve the app's stability o

Ex :

```
Class ErrorBoundary extends React.component {
  constructor(props) {
    super(props);
    this.state = { hasError : false };
  }
  Static getDerivedStateFromError (error) {
    return { hasError : true };
```

4 How does react router enable single page application (SPA) functionality

A React router allows navigation b/w pages without refreshing the brow turns a react app into a single page application (SPA) by dynamicall the view when the URL changes.

Benefits:
- faster navigation
- smooth user experience
- Route Based rendering

Ex:

```
input { BrowserRouter, Route, Routes }
from "react-router-dom";
<BrowserRouter>
<Routes>
<Route path = "/" element = {<Home />} />
<Route path = "/about" element = {<About />} />
</Routes>
</BrowserRouter>
```

5 Explain the different ways to style a react application.

A a) CSS stylesheets
Traditional css files imported into components

b) Inline styles
Directly style elements using Javascript objects

c) CSS modules
scoped css that avoids naming conflicts

d) CSS in JS libraries
Styled component. * Emotion - styles written in JS

e) UI frameworks
Bootstrap material - UI, Tailwind CSS for pre built responsive designs.

Scanned with OKEN Scanner

conclusion:

By using lists & keys, portals, errors Boundaries, styling techniques. React applications become:

- more dynamic [lists]
- more flexible [portals]
- more user friendly [SPA navigation]
- more reliable [error boundaries]
- more visually appealing & responsive

These enhancements collectively, improve both user experience and application robustness.

26/9/25

FSD - ASSIGNMENT 5

Name : Sindhura Gulhane

PRN : 1032230397

- Aim : Develop a responsive web design using express framework to perform CRUD operations and deploy with Node. Js use. MongoDB

- Objectives : 1) Develop a full stack application
  2) Demonstrate Backend Development & Deployment profiency

- Theory :

1 What is the role of express. Js as a web frameworks for node. Js ?

A 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 & response handling and template rendering, which makes backend development faster & easier instead of writing complex server code with just node. js, developers use express Js to build RESTful API's, full stack applications & scalable web servers with less efforts & more structure.

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

A 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.

Scanned with OKEN Scanner

Update:
Used to modify / update existing data / values

Delete:
Used to delete / remove data.

In a web application CRUD is usually implemented through API that interact with the database. The operations form the backbone most dynamic applications.

3 Why is MongoDB suitable choice for this project?
A MongoDB is a document oriented NO SQL database that stores different types of data. Without needing a fixed structure which development. MongoDB also supports scalability, high performance integration with Express.js, through libraries like MongoDB. It it a perfect choice for modern web applications. where requirements may change frequently.

4 what steps are involved in deploying a Node.js and express appli
A Deploying a Node.js & express application typically involves the steps:

- 1) Develop locally:
  Build your app with express routes & MongoDB integration

- 2) Version control:
  Push your code into GitHub or another repository.

- 3) Choose hosting:
  Use platforms like Heroku, Render AWS or Digital

- 4) Install Dependencies →
  Run npm install on the servers.

-5) Configure environment variables :
set up MongoDB connection port etc.

-6) Run APP :
start using node server.js or a process manager like PM2.

-7) Test Deployment ⇒ :
ensure routes & 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 an express app involves preparing the code hosting it on server configuring the database and running it. Efficiently together these components enable the creation of a robust, responsive & fully deployable full stack application.

26/9/24

FOR EDUCATIONAL USE

Scanned with OKEN Scanner