Date

NAME - MATBARJOSHT
Email - matbar9105669237@gmail.com
mob - 9536590837

Q1

FrontEnd :- It mainly include HTML, CSS and java Jscript.
HTML :- used to write a web page
Front End is also know as client Sid.
CSS :- Cascading style sheets :- used to create a good interface of website we can add in html also if we want to create another file we create .css file

javaScript :- used to build logic in developer tool we can see on consol.

Backend :- The backend is also known as server side which hand how the website work there are many part in Back
① Node.js ② Database
etc.

Date

Q2

Tags are used to mark up different content of web page. They are enclosed in angle brackets "<" and ">"

opening tag <>

closing tag </>

e.g. Heading <h1> </h1>
<h2> </h2>
<h3> </h3>
<h4> </h4>
<h5> </h5>
<h6> </h6>

Paragraph   used for para

image :- to add image

link :- to add link

Q3

Dom is a concept used in web development framework. like React to improve performance and efficiency the use.

(1) Initial Render :- When you first load a web application or a webpage, build with a framework that uses the Virtual Dom the framework.

(2) Diffing :- After the new version of the Dom is generated then react the

framework. compare it with the previous version of the virtual DOM.

Performance Benefit :-

The use of the virtual DOM provides performance benefits by minimizing the number of actual DOM manipulations during updates instead of directly modifying the real DOM multiple times for every change.

- By using virtual DOM framework can optimize the rendering process and enhance the performance of web application making them responsive and efficient in handling updates and user interactions.

Q4

Both NoSQL and mySQL are different types of database management system the difference below.

Scalability :- mySQL :- mySQL traditionally scales vertically which means it can handle increased loads by adding more resource (CPU, RAM) to a single

Date

server but may have limitations in handling massive scalability needs

NO SQL: Data bases are designed for horizontal scalability the distribution of data across multiple servers

Data model:- my SQL is a relational database management (RDBMS) that follows a structured schema with tables, rows and columns, it enforces a fixed schema, meaning the data has to confirm to a predefined structure

NOSQL:- data bas employ various data models, including key-value document columns oriented and graph They offer flexibility

Data Relational and joins:
my SQL excel in handling complex relationships b/w data entities. They support JOIN operations to combine data from multiple table based on common columns

NO SQL:- Database typically not support complex joins or relations b/w data entities in the same way as relational

Date _____

~~database~~.

Flexibility - MySQL → Enforces a rigid schema, requires table & columns with fixed data types.

NoSQL - Data base are schema flexible, allows dynamic and evolving data structure.

uses - MSQL traditional use relational database application, Content management system

NoSQL use
Real time analytic

Q5

DBMS technology is mongoDB mongodb is a NoSQL DBS that uses, data oriented model. in simple terms it store data in the forms of flexible and self-contained document which as user can use it.

In mongo DB data is organized into collections which are analogous to table in relation database.

Each Document within collection represent a record and it store

Date

in Binary JSON objects.
BSON allow for efficient
storage and retrieval of
complex data types
like array nested objects
and binary data
mongoDB provides a rich qu
language (MQL)
mongo DB commonly used in
modern web application
context mongoose system.

Create an HTML webpage which calculates the BMI of your body. Use JavaScript to validate the Calculation

Code:-

```
<!DOCTYPE html>

<html>

<head>

 <title>BMI Calculator</title>

 <script>

  function calculateBMI() {

   // Get input values

   var weight = parseFloat(document.getElementById('weight').value);

   var height = parseFloat(document.getElementById('height').value);


   // Validate input

   if (isNaN(weight) || isNaN(height)) {

    document.getElementById('result').innerHTML = "Please enter valid weight and height values.";

    return;

   }


   // Calculate BMI

   var bmi = weight / (height * height);


   // Display the result

   document.getElementById('result').innerHTML = "Your BMI is: " + bmi.toFixed(2);

  }
```

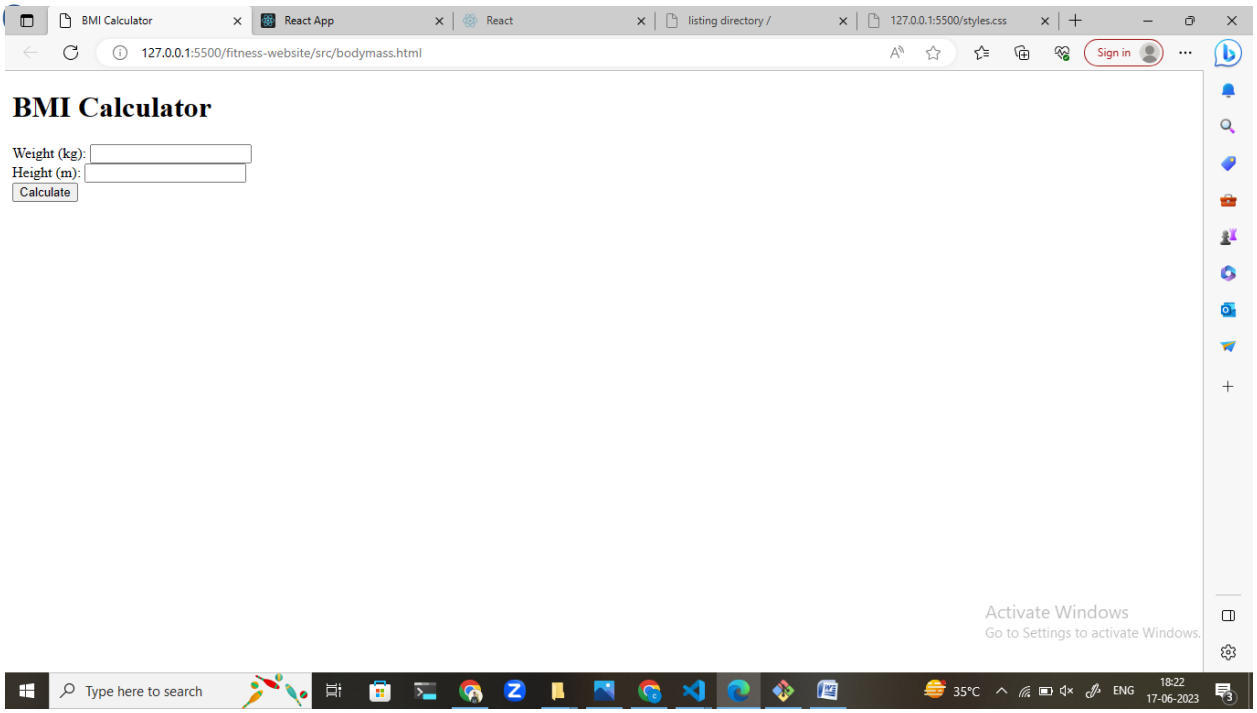
```
  </script>
</head>
<body>
  <h1>BMI Calculator</h1>

  <label for="weight">Weight (kg):</label>
  <input type="number" id="weight" step="0.01"><br>

  <label for="height">Height (m):</label>
  <input type="number" id="height" step="0.01"><br>

  <button onclick="calculateBMI()">Calculate</button>

  <div id="result"></div>
</body>
</html>
```



Create the React app which acts as the Record Form and Collects necessary data of Student for Admission Purpose with mentioned below Requirements- (a. Title of the Web APP must be “Admission Form”. b. It should have a valid CSS file and the form box must be at centre. c. Use CSS box Model to add relevant padding and other styling content.

Code

App.css

```css
.App {
  text-align: center;
}

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}
```

App.js

```jsx
import React, { useState } from 'react';
import './styles.css';

function App() {
  const [formData, setFormData] = useState({
```

```
  firstName: '',
  lastName: '',
  email: '',
  phone: '',
  address: '',
});

const handleChange = (e) => {
  const { name, value } = e.target;
  setFormData((prevData) => ({
    ...prevData,
    [name]: value,
  }));
};

const handleSubmit = (e) => {
  e.preventDefault();
  console.log(formData);
  // Add logic to submit the form data
};

return (
  <div className="container">
    <form className="form" onSubmit={handleSubmit}>
      <h1>Admission Form</h1>
      <label htmlFor="firstName">First Name:</label>
      <input
        type="text"
        id="firstName"
        name="firstName"
        value={formData.firstName}
        onChange={handleChange}
        required
      />

      <label htmlFor="lastName">Last Name:</label>
      <input
        type="text"
        id="lastName"
        name="lastName"
        value={formData.lastName}
        onChange={handleChange}
        required
      />
```

```
        <label htmlFor="email">Email:</label>
        <input
          type="email"
          id="email"
          name="email"
          value={formData.email}
          onChange={handleChange}
          required
        />

        <label htmlFor="phone">Phone:</label>
        <input
          type="tel"
          id="phone"
          name="phone"
          value={formData.phone}
          onChange={handleChange}
          required
        />

        <label htmlFor="address">Address:</label>
        <textarea
          id="address"
          name="address"
          value={formData.address}
          onChange={handleChange}
          required
        ></textarea>

        <button type="submit">Submit</button>
      </form>
    </div>
  );
}

export default App;
```

app.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
```

```
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

Index.css

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}
```

Index.css

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

log.svg

```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 841.9 595.3"><g
fill="#61DAFB"><path d="M666.3 296.5c0-32.5-40.7-63.3-103.1-82.4 14.4-63.6 8-
114.2-20.2-130.4-6.5-3.8-14.1-5.6-22.4-5.6v22.3c4.6 0 8.3.9 11.4 2.6 13.6 7.8
19.5 37.5 14.9 75.7-1.1 9.4-2.9 19.3-5.1 29.4-19.6-4.8-41-8.5-63.5-10.9-13.5-
18.5-27.5-35.3-41.6-50 32.6-30.3 63.2-46.9 84-46.9V78c-27.5 0-63.5 19.6-99.9
53.6-36.4-33.8-72.4-53.2-99.9-53.2v22.3c20.7 0 51.4 16.5 84 46.6-14 14.7-28 31.4-
41.3 49.9-22.6 2.4-44 6.1-63.6 11-2.3-10-4-19.7-5.2-29-4.7-38.2 1.1-67.9 14.6-
75.8 3-1.8 6.9-2.6 11.5-2.6V78.5c-8.4 0-16 1.8-22.6 5.6-28.1 16.2-34.4 66.7-19.9
130.1-62.2 19.2-102.7 49.9-102.7 82.3 0 32.5 40.7 63.3 103.1 82.4-14.4 63.6-8
114.2 20.2 130.4 6.5 3.8 14.1 5.6 22.5 5.6 27.5 0 63.5-19.6 99.9-53.6 36.4 33.8
72.4 53.2 99.9 53.2 8.4 0 16-1.8 22.6-5.6 28.1-16.2 34.4-66.7 19.9-130.1 62-19.1
102.5-49.9 102.5-82.3zm-130.2-66.7c-3.7 12.9-8.3 26.2-13.5 39.5-4.1-8-8.4-16-
13.1-24-4.6-8-9.5-15.8-14.4-23.4 14.2 2.1 27.9 4.7 41 7.9zm-45.8 106.5c-7.8 13.5-
15.8 26.3-24.1 38.2-14.9 1.3-30 2-45.2 2-15.1 0-30.2-.7-45-1.9-8.3-11.9-16.4-
24.6-24.2-38-7.6-13.1-14.5-26.4-20.8-39.8 6.2-13.4 13.2-26.8 20.7-39.9 7.8-13.5
15.8-26.3 24.1-38.2 14.9-1.3 30-2 45.2-2 15.1 0 30.2.7 45 1.9 8.3 11.9 16.4 24.6
24.2 38 7.6 13.1 14.5 26.4 20.8 39.8-6.3 13.4-13.2 26.8-20.7 39.9zm32.3-13c5.4
13.4 10 26.8 13.8 39.8-13.1 3.2-26.9 5.9-41.2 8 4.9-7.7 9.8-15.6 14.4-23.7 4.6-8
8.9-16.1 13-24.1zM421.2 430c-9.3-9.6-18.6-20.3-27.8-32 9 .4 18.2.7 27.5.7 9.4 0
18.7-.2 27.8-.7-9 11.7-18.3 22.4-27.5 32zm-74.4-58.9c-14.2-2.1-27.9-4.7-41-7.9
3.7-12.9 8.3-26.2 13.5-39.5 4.1 8 8.4 16 13.1 24 4.7 8 9.5 15.8 14.4 23.4zM420.7
163c9.3 9.6 18.6 20.3 27.8 32-9-.4-18.2-.7-27.5-.7-9.4 0-18.7.2-27.8.7 9-11.7
18.3-22.4 27.5-32zm-74 58.9c-4.9 7.7-9.8 15.6-14.4 23.7-4.6 8-8.9 16-13 24-5.4-
13.4-10-26.8-13.8-39.8 13.1-3.1 26.9-5.8 41.2-7.9zm-90.5 125.2c-35.4-15.1-58.3-
34.9-58.3-50.6 0-15.7 22.9-35.6 58.3-50.6 8.6-3.7 18-7 27.7-10.1 5.7 19.6 13.2 40
22.5 60.9-9.2 20.8-16.6 41.1-22.2 60.6-9.9-3.1-19.3-6.5-28-10.2zM310 490c-13.6-
7.8-19.5-37.5-14.9-75.7 1.1-9.4 2.9-19.3 5.1-29.4 19.6 4.8 41 8.5 63.5 10.9 13.5
18.5 27.5 35.3 41.6 50-32.6 30.3-63.2 46.9-84 46.9-4.5-.1-8.3-1-11.3-2.7zm237.2-
76.2c4.7 38.2-1.1 67.9-14.6 75.8-3 1.8-6.9 2.6-11.5 2.6-20.7 0-51.4-16.5-84-46.6
14-14.7 28-31.4 41.3-49.9 22.6-2.4 44-6.1 63.6-11 2.3 10.1 4.1 19.8 5.2
29.1zm38.5-66.7c-8.6 3.7-18 7-27.7 10.1-5.7-19.6-13.2-40-22.5-60.9 9.2-20.8 16.6-
41.1 22.2-60.6 9.9 3.1 19.3 6.5 28.1 10.2 35.4 15.1 58.3 34.9 58.3 50.6-.1 15.7-
23 35.6-58.4 50.6zM320.8 78.4z"/><circle cx="420.9" cy="296.5" r="45.7"/><path
d="M520.5 78.1z"/></g></svg>
```

Vital.js

```
const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
```

```
  }
};

export default reportWebVitals;
```

set up test.js

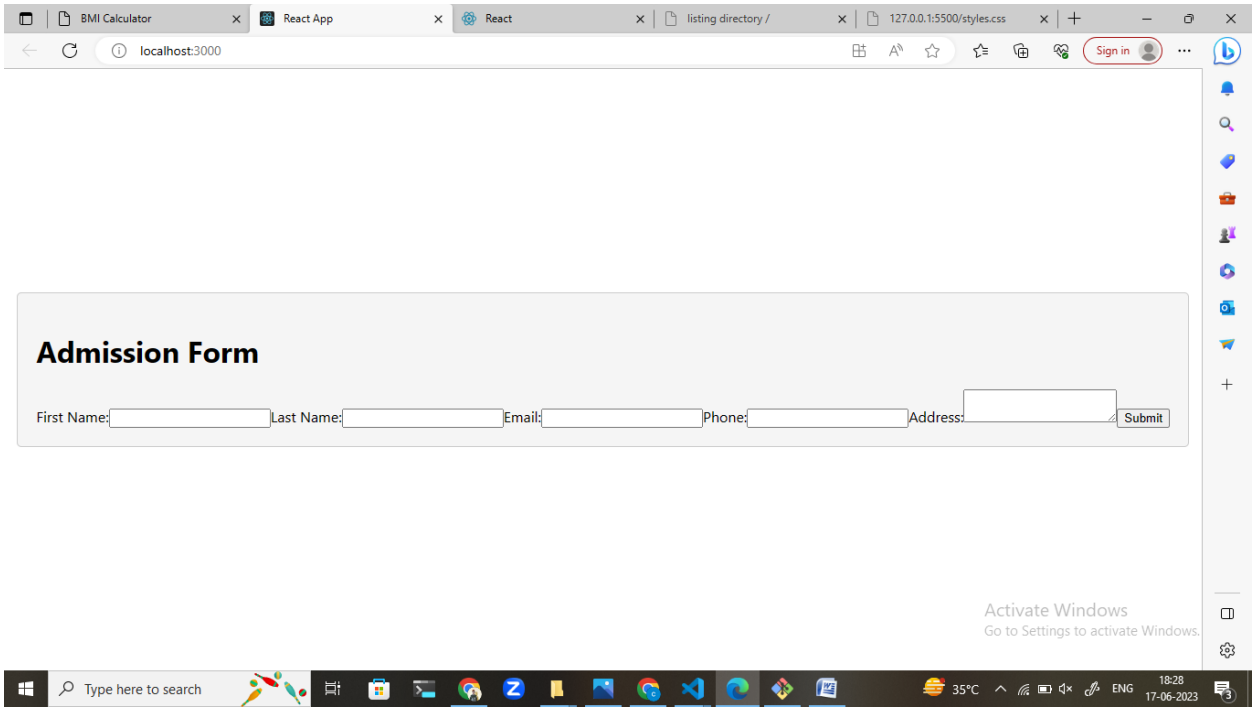
```
// jest-dom adds custom jest matchers for asserting on DOM nodes.
// allows you to do things like:
// expect(element).toHaveTextContent(/react/i)
// learn more: https://github.com/testing-library/jest-dom
import '@testing-library/jest-dom';
```

style.css

```
/* styles.css */
.container {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }

  .form {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #f5f5f5;
  }
```

Output;_


BMI Calculator
React App
React
listing directory /
127.0.0.1:5500/styles.css
localhost:3000
Sign in
Admission Form
First Name:
Last Name:
Email:
Phone:
Address:
Submit
Activate Windows
Go to Settings to activate Windows.
Type here to search
35°C
ENG
18:28
17-06-2023