# KEC BMI CALCULATION SYSTEM

TEAM MEMBERS:

|  |  |
| --- | --- |
| ROLL NUMBER | NAME |
| SHAM SUNDAR K | 22ALR090 |
| PRANAV A S | 22ALR068 |
| VISHNU S | 22ALR112 |

# INTRODUCTION:

Body Mass Index (BMI) is a widely used metric that assesses an individual's body fat based on their height and weight. This project aims to explore the significance of BMI, its applications, and its limitations. By understanding BMI, we can better appreciate its role in health assessments, identify potential health risks, and promote healthier lifestyle choices.

BMI is calculated using a simple formula: weight in kilograms divided by the square of height in meters (kg/m²). It categorizes individuals into various groups such as underweight, normal weight, overweight, and obese. Despite its simplicity and widespread use, BMI has certain limitations. It does not account for muscle mass, bone density, and distribution of fat, which can lead to misleading interpretations, particularly in athletes and individuals with high muscle mass.

In this project, we will delve into the methodology of calculating BMI, analyze data to understand the distribution of BMI across different populations, and examine its correlation with other health indicators. Additionally, we will address the controversies surrounding BMI and discuss alternative measures that may provide a more comprehensive understanding of an individual's health.

By the end of this project, we aim to present a balanced view of BMI, highlighting both its utility and its shortcomings, and propose ways to enhance health assessments beyond this single measure.

**Key features:**

1. Patient Registration and appointment scheduling using react
2. You can find the location of the hospital through our web application. 3.Doctors complete information of every individual doctor.

4.Appointment is through calendar scheduling you can verify whether the doctor is available. 5.These data are stored in API FireBase backend.

1. You have a two-step verification process for login. 7.Your BMI can be calculated.

### Project Requirements

#### Front-End Programming

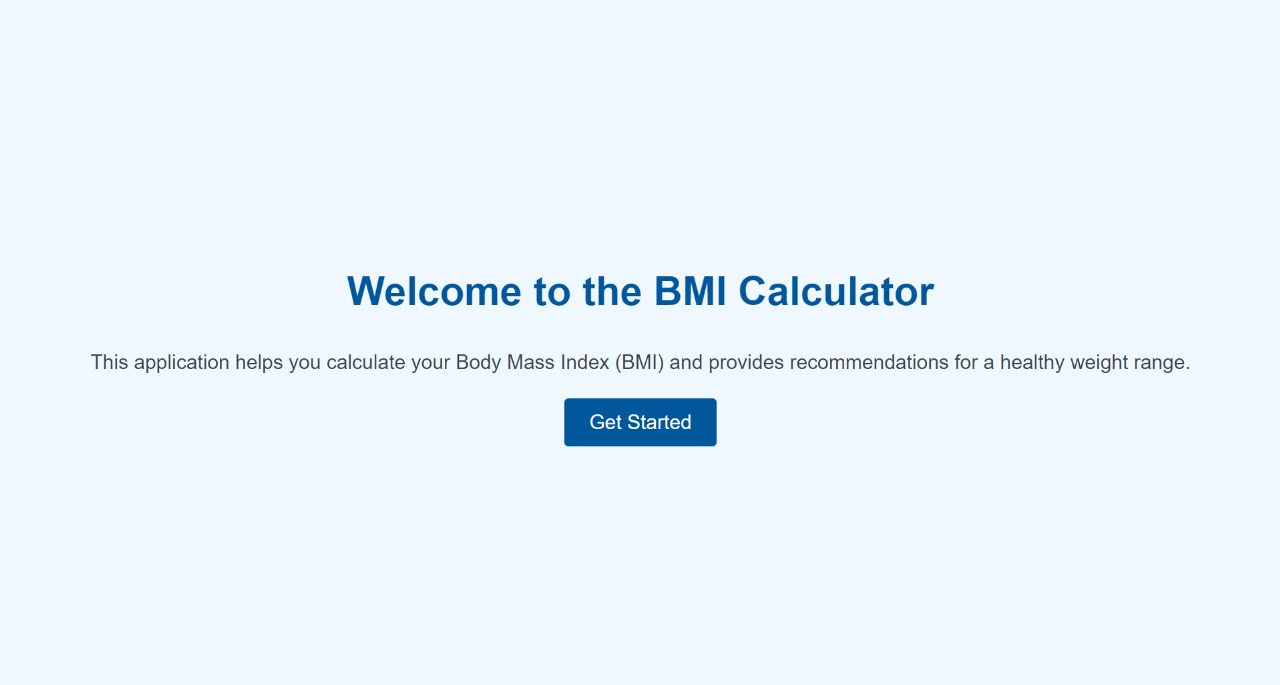
1. **CSS (Cascading Style Sheets)**: CSS is essential for defining the visual appearance and layout of web pages. It allows developers to apply styles, such as colors, fonts, spacing, and positioning, to HTML elements, creating a visually appealing and consistent user interface.
2. **JavaScript**: JavaScript is a powerful programming language that adds interactivity and dynamic behavior to web pages. It enables developers to manipulate and modify HTML and CSS, handle user interactions, perform calculations, validate forms, and make asynchronous requests to servers.
3. **React**: React is a popular JavaScript library for building user interfaces. It provides a component-based approach to web development, allowing developers to create reusable UI components that update efficiently based on changes in data. React uses a virtual DOM (Document Object Model) to optimize performance and facilitate the building of complex web applications.

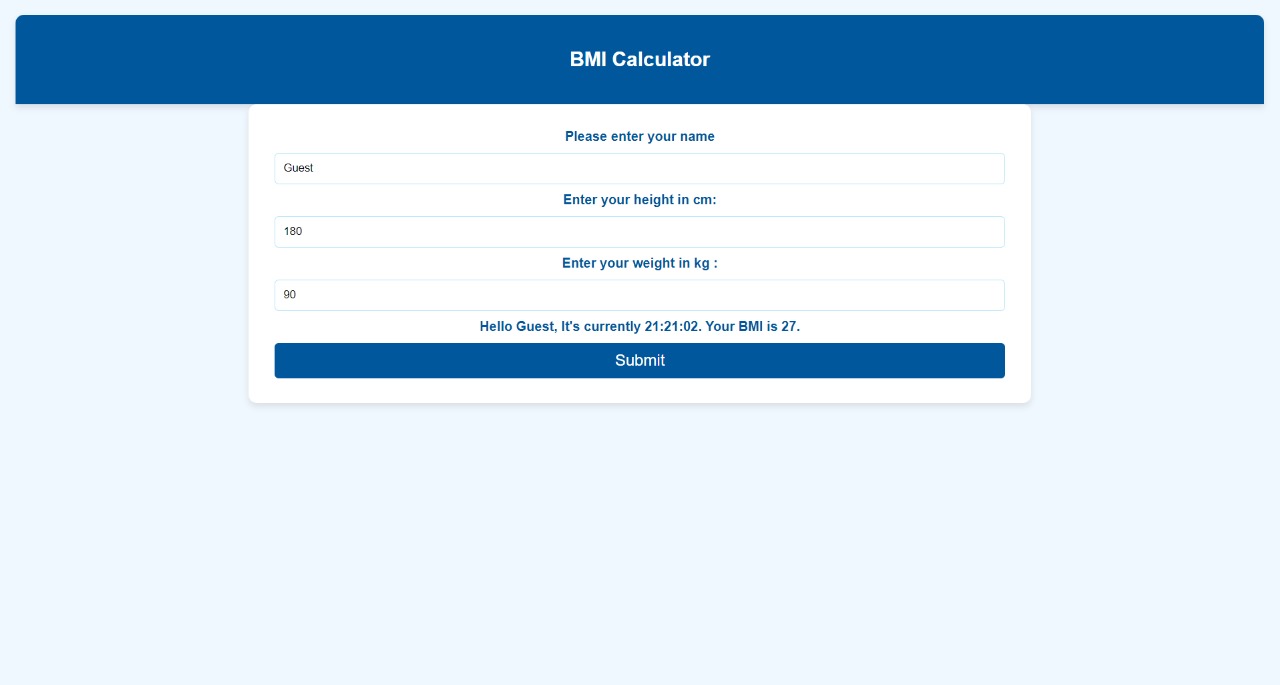
#### Back-End Programming

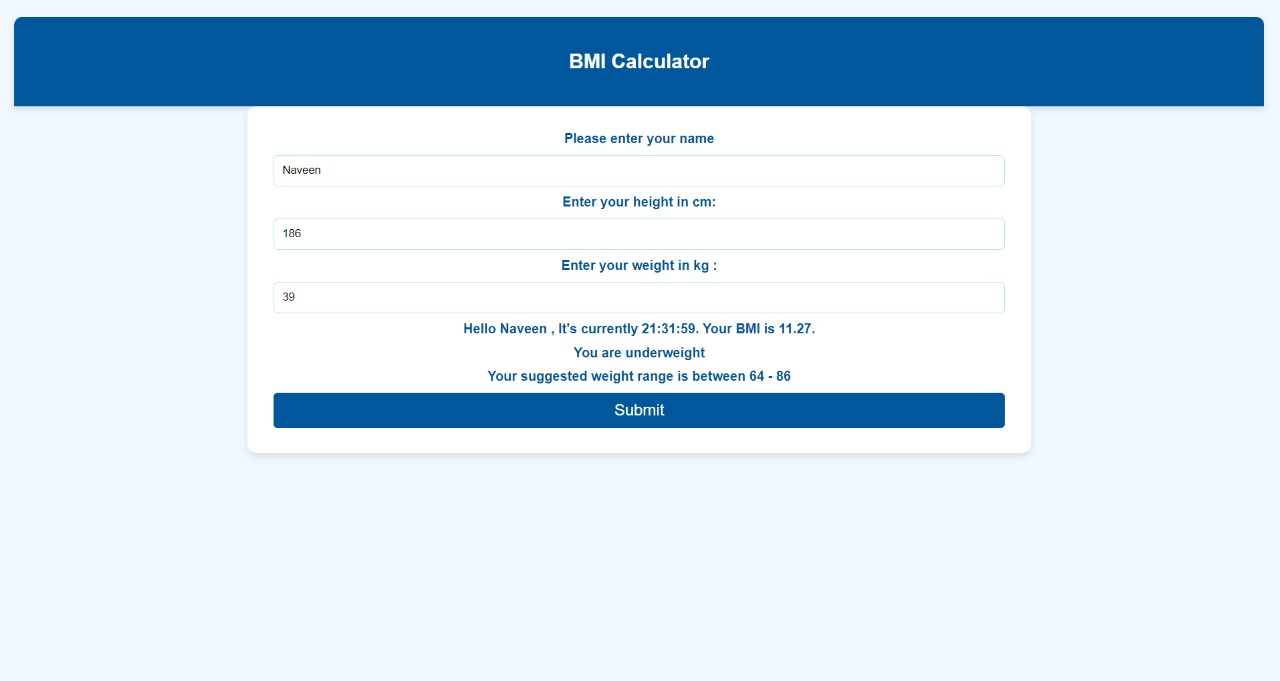
1. **Database Management**: Backend programming involves working with databases to store and retrieve data. Developers interact with databases using query languages like SQL (Structured Query Language) or NoSQL databases. They design and implement database schemas, optimize queries, and ensure data integrity and security.
2. **Server-Side Logic**: Backend developers are responsible for writing the server-side logic that powers web applications. This includes handling business logic, managing user authentication and authorization, processing requests, and returning appropriate responses to the front-end. This logic is typically written in languages such as Python, Java, Ruby, or Node.js.

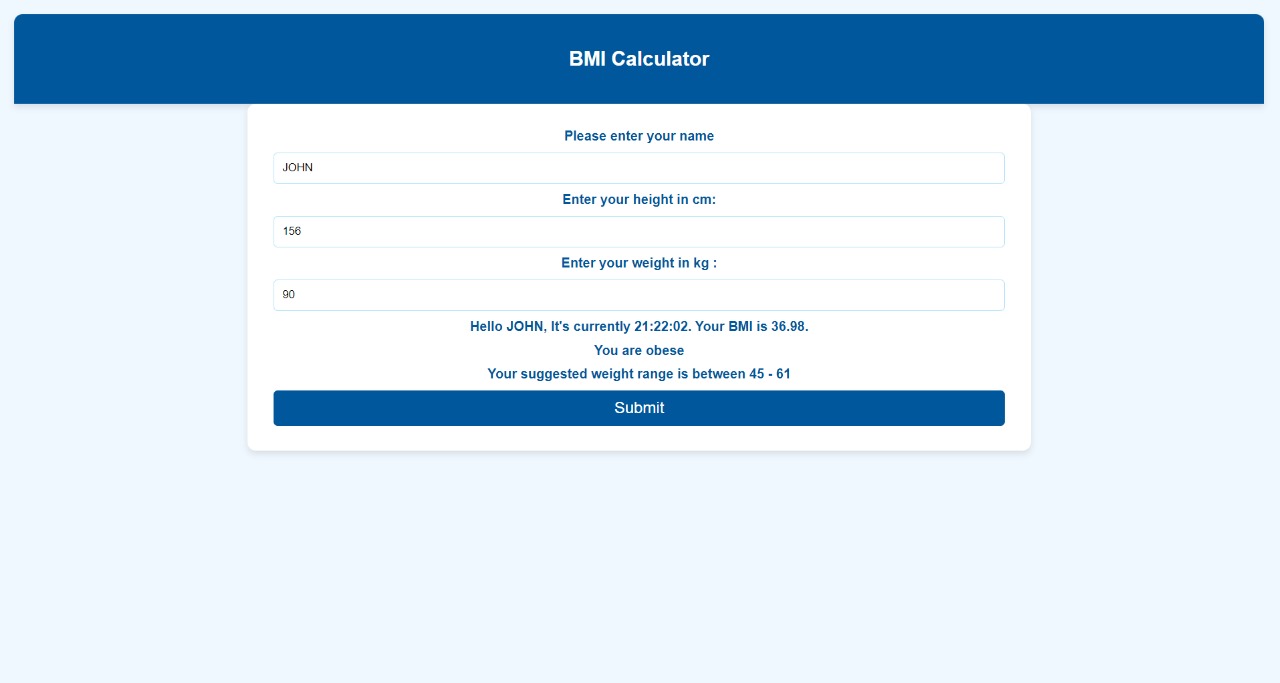
These requirements outline the essential technologies and skills needed for front-end and back-end development in this project. By leveraging these tools, we aim to create a robust, efficient, and user-friendly web application.

# UI DESIGN:

****

****

****

****

**SAMPLE CODING:**

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="shortcut icon" href="%PUBLIC\_URL%/favicon.ico">

<!--

Notice the use of %PUBLIC\_URL% in the tag above.

It will be replaced with the URL of the public folder during the build.

Only files inside the public folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running npm run build.

-->

<title>React App</title>

</head>

<body>

<div id="root"></div>

<!--

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the <body> tag.

To begin the development, run npm start.

To create a production bundle, use npm run build.

-->

</body>

</html>

index.html  
  
/\* Welcome Page CSS \*/

.WelcomePage {

text-align: center;

width: 100%;

height: 100vh;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

animation: fadeIn 1s ease-in-out; /\* Fade-in animation \*/

background-color: #f0f8ff; /\* Light cyan background \*/

}

.WelcomePage h1 {

font-size: 3em;

color: #01579b; /\* Dark blue for the welcome message \*/

margin-bottom: 20px;

animation: fadeInDown 1s ease-in-out; /\* Fade-in and slide-down animation \*/

}

.WelcomePage p {

font-size: 1.5em;

color: #37474F; /\* Consistent dark slate gray for text \*/

margin-bottom: 30px;

animation: fadeInUp 1s ease-in-out; /\* Fade-in and slide-up animation \*/

}

.WelcomePage .health-info {

font-size: 1.2em;

color: #37474F; /\* Consistent dark slate gray for text \*/

margin-bottom: 30px;

animation: fadeInUp 1s ease-in-out; /\* Fade-in and slide-up animation \*/

}

.WelcomePage button {

padding: 15px 30px;

font-size: 1.5em;

color: white;

background: #01579b; /\* Dark blue button \*/

border: none;

border-radius: 5px;

cursor: pointer;

animation: fadeIn 1s ease-in-out; /\* Fade-in animation \*/

transition: background-color 0.3s ease; /\* Smooth background color transition \*/

}

.WelcomePage button:hover {

background: #014a79; /\* Slightly darker blue on hover \*/

}

.WelcomePage button:active {

background: #013a5c; /\* Even darker blue on click \*/

}

/\* Fade In Animation \*/

@keyframes fadeIn {

0% { opacity: 0; }

100% { opacity: 1; }

}

/\* Fade In and Slide Down Animation \*/

@keyframes fadeInDown {

0% { opacity: 0; transform: translateY(-50px); }

100% { opacity: 1; transform: translateY(0); }

}

/\* Fade In and Slide Up Animation \*/

@keyframes fadeInUp {

0% { opacity: 0; transform: translateY(50px); }

100% { opacity: 1; transform: translateY(0); }

}

/\* App CSS \*/

body {

margin: 0;

font-family: Arial, sans-serif;

background-color: #f0f8ff; /\* Light cyan background \*/

transition: background-color 0.3s ease; /\* Smooth transition for background color \*/

}

.App {

text-align: center;

padding: 20px;

animation: fadeIn 1s ease-in-out; /\* Fade-in animation \*/

}

.App-header {

background-color: #01579b; /\* Dark blue for header \*/

padding: 20px;

color: white;

border-top-left-radius: 10px;

border-top-right-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

transition: background-color 0.3s cubic-bezier(0.25, 0.46, 0.45, 0.94); /\* Smooth transition for header background color \*/

}

.App-intro {

font-size: 1.2em;

color: #37474F; /\* Consistent dark slate gray for text \*/

margin-bottom: 30px;

animation: fadeInUp 1s ease-in-out; /\* Fade-in and slide-up animation \*/

}

form {

margin: 0 auto;

width: 60%;

padding: 20px;

background: #ffffff;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

animation: fadeIn 1s ease-in-out; /\* Fade-in animation \*/

transition: box-shadow 0.3s ease, transform 0.3s ease; /\* Smooth transition for form box-shadow and transform \*/

}

label {

display: block;

margin: 10px 0;

font-weight: bold;

color: #01579b; /\* Darker blue for labels \*/

}

input {

display: block;

width: calc(100% - 22px); /\* Adjust input width \*/

padding: 10px;

margin: 10px auto;

border: 1px solid #b3e5fc; /\* Light blue border \*/

border-radius: 5px;

box-sizing: border-box;

transition: border-color 0.3s ease; /\* Smooth transition for input border color \*/

}

input[type='submit'] {

padding: 10px 20px;

border-radius: 5px;

background: #01579b; /\* Dark blue for button \*/

color: #fff;

border: none;

cursor: pointer;

font-size: 1.2em;

transition: background-color 0.3s cubic-bezier(0.25, 0.46, 0.45, 0.94); /\* Smooth transition for button background color \*/

}

input[type='submit']:hover {

background: #014a79; /\* Slightly darker blue on hover \*/

}

@keyframes fadeIn {

0% { opacity: 0; }

100% { opacity: 1; }

}

@keyframes fadeInUp {

0% { opacity: 0; transform: translateY(20px); }

100% { opacity: 1; transform: translateY(0); }

}  
  
  
import React, { Component } from 'react';

import './App.css';

class App extends Component {

constructor(props) {

super(props);

this.state = {

name: 'Guest',

weight: 90,

height: 180,

bmi: 27,

message: '',

optimalweight: '',

time: new Date().toLocaleTimeString(),

showWelcomePage: true // State to control the display of the welcome page

};

this.submitMe = this.submitMe.bind(this);

this.heightchange = this.heightchange.bind(this);

this.weightchange = this.weightchange.bind(this);

this.change = this.change.bind(this);

this.ticker = this.ticker.bind(this);

this.blur = this.blur.bind(this);

this.calculateBMI = this.calculateBMI.bind(this);

this.startApp = this.startApp.bind(this); // Bind the new method

}

heightchange(e) {

this.setState({ height: e.target.value });

e.preventDefault();

}

blur(e) {

this.calculateBMI();

}

weightchange(e) {

this.setState({ weight: e.target.value });

e.preventDefault();

}

calculateBMI() {

var heightSquared = (this.state.height / 100) \* (this.state.height / 100);

var bmi = this.state.weight / heightSquared;

var low = Math.round(18.5 \* heightSquared);

var high = Math.round(24.99 \* heightSquared);

var message = "";

if (bmi >= 18.5 && bmi <= 24.99) {

message = "You are in a healthy weight range";

} else if (bmi >= 25 && bmi <= 29.9) {

message = "You are overweight";

} else if (bmi >= 30) {

message = "You are obese";

} else if (bmi < 18.5) {

message = "You are underweight";

}

this.setState({ message: message });

this.setState({ optimalweight: "Your suggested weight range is between " + low + " - " + high });

this.setState({ bmi: Math.round(bmi \* 100) / 100 });

}

submitMe(e) {

e.preventDefault();

this.calculateBMI();

}

ticker() {

this.setState({ time: new Date().toLocaleTimeString() });

}

componentDidMount() {

setInterval(this.ticker, 60000);

}

change(e) {

e.preventDefault();

console.log(e.target);

this.setState({ name: e.target.value });

}

startApp() {

this.setState({ showWelcomePage: false });

}

render() {

if (this.state.showWelcomePage) {

return (

<div className="WelcomePage">

<h1>Welcome to the BMI Calculator</h1>

<p>This application helps you calculate your Body Mass Index (BMI) and provides recommendations for a healthy weight range.</p>

<div className="Carousel">

<div></div>

<div></div>

<div></div>

</div>

<button onClick={this.startApp}>Get Started</button>

<div className="Waves"></div>

</div>

);

}

return (

<div className="App">

<div className="cloud cloud1"></div>

<div className="cloud cloud2"></div>

<div className="App-header">

<h2>BMI Calculator</h2>

</div>

<form onSubmit={this.submitMe} className="animatedForm">

<label>

Please enter your name

</label>

<input type="text" name="name" value={this.state.name} onBlur={this.blur} onChange={this.change} />

<label>

Enter your height in cm:

</label>

<input type="text" name="height" value={this.state.height} onBlur={this.blur} onChange={this.heightchange} />

<label>

Enter your weight in kg :

</label>

<input type="text" name="weight" value={this.state.weight} onChange={this.weightchange} />

<label>

Hello {this.state.name}, It's currently {this.state.time}. Your BMI is {this.state.bmi}.

</label>

<label>{this.state.message}</label>

<label>{this.state.optimalweight}</label>

<input type="submit" value="Submit" />

</form>

</div>

);

}

}

export default App;  
  
  
body {

margin: 0;

padding: 0;

font-family: sans-serif;

}

index.css  
  
  
import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

// import Bpp from './BPP';

import './index.css';

ReactDOM.render(

<App></App>,

document.getElementById('root')

);

index.js

# CONCLUSION:

The Body Mass Index (BMI) is a valuable tool for assessing body fat based on height and weight. It offers a simple, quick method to categorize individuals into various weight categories, such as underweight, normal weight, overweight, and obese. This project has highlighted BMI's utility in health assessments and its widespread use in both clinical and personal health contexts.

However, BMI is not without its limitations. It does not account for factors such as muscle mass, bone density, and fat distribution, which can lead to inaccurate classifications, particularly in athletes and individuals with high muscle mass. Despite these shortcomings, BMI remains a useful starting point for identifying potential health risks and encouraging healthier lifestyle choices.

Moving forward, it's important to complement BMI with other health metrics and assessments for a more comprehensive understanding of an individual's health. Measurements like waist circumference, body fat percentage, and other clinical indicators can provide a fuller picture. By integrating these additional measures, healthcare providers and individuals can make more informed decisions about health and wellness.

In conclusion, while BMI is a valuable and accessible tool, its limitations necessitate a broader approach to health assessment. Combining BMI with other health indicators will enhance our ability to promote and maintain overall well-being.