

Assignment 1: BMI Calculator with Routing in Node.js

Objective:

The objective of this assignment is to introduce students to Node.js routing and handling HTTP requests. You will build a simple BMI (Body Mass Index) calculator web service using Node.js that calculates the BMI based on the user's input (weight and height) and returns the result via HTTP.

Requirements:

1. **Create a new Node.js project:**
 - Ensure Node.js is installed on your machine. If not, download and install it from [Node.js Official Website](https://nodejs.org/).
 - Initialize a new project using `npm init` and set up basic dependencies
2. **Create an Express Server:** Set up a basic Express server that listens on port 3000.
3. **Create the BMI Calculator Endpoint:** The BMI calculation endpoint should handle GET or POST requests where a user can send their height and weight.
 - **GET /:** This route should render a simple HTML form where the user can input their weight and height.
 - **POST /calculate-bmi:** This route should accept the user's input (weight and height) and calculate the BMI.
4. **Return the BMI Result:** After calculating the BMI, return the result with a simple message indicating if the BMI is underweight, normal, overweight, or obese.
 - Use the formula: $BMI = \text{weight (kg)} / \text{height}^2 \text{ (m)}$
 - Based on the BMI value, determine the category:
 - i. **Underweight:** $BMI < 18.5$
 - ii. **Normal weight:** $18.5 \leq BMI < 24.9$
 - iii. **Overweight:** $25 \leq BMI < 29.9$
 - iv. **Obese:** $BMI \geq 30$
 - After calculation, return a response to the user indicating their BMI value and the category.
 - Display the result on the web page

5. **Enhance User Interface:** Create a clean and responsive HTML form.
- Check if weight and height are positive numbers. Show an alert or message if inputs are invalid.
 - Add CSS for styling:
 - i. Input fields, buttons, and layout.
 - ii. Color-coded BMI result display for different categories.
 - iii. Example: Green for "Normal," yellow for "Overweight," and red for "Obese."

Submission:

- Submit the following:
 - A .zip file or a GitHub repository link containing your project.

Grading Criteria:

Criteria	Description	Weight
1. Functionality and Implementation	The application correctly implements the BMI calculation, returns appropriate responses, and displays results on the web page.	30%
2. Code Structure and Organization	The code is well-organized, modular, and follows best practices for Node.js and Express. Proper use of routes and methods.	10%
3. Styling and User Interface	The user interface is well-designed, easy to navigate, and visually appealing.	10%
4. Defense	The student clearly explains the functionality, structure, and decisions behind the code during the defense. They should also answer theoretical questions related to the project and the concepts used.	50%