# WEB TECHNOLOGY
# BMC201
# ASSIGNMENT PLAN

Unit 1: Web Page Designing (HTML & CSS)
Assignment 1: Create a simple personal webpage using HTML, including your name, a short biography, and a photo. Apply appropriate HTML tags for headings,
Assignment 2: Develop a multi-section webpage that utilizes `<div>` and `<span>` elements to organize content hierarchically. Include different color backgrounds
Assignment 3: Create a webpage that features a navigation bar with hyperlinks to different sections of the page. Use both internal links (using `id` attributes) and
Assignment 4: Implement an organized list of your favorite books or movies using HTML lists (`<ul>` and `<ol>`) with anchors for further details.
Assignment 5: Design a contact form using HTML, ensuring to incorporate different form elements like input fields, radio buttons, checkboxes, and a textarea.
Assignment 6: Create a basic responsive webpage using Bootstrap's grid system. Include at least three columns that collapse into a single column on smaller
Assignment 7: Write a CSS style sheet to enhance the above webpages by applying different styles using selectors, the box model, and layout properties (such as

Unit 2: JavaScript Scripting
Assignment 1: Write a JavaScript program that declares variables for your age and favorite color, then log them to the browser console.
Assignment 2: Create a function in JavaScript that takes two numbers as parameters and returns their sum. Display the output in an alert dialog.
Assignment 3: Develop a simple webpage that responds to a button click event by changing the background color of the page. Utilize event listeners for this.
Assignment 4: Create a JavaScript program that demonstrates conditional logic by checking user input (like a number) and displaying whether it is odd or even.
Assignment 5: Write a JavaScript code snippet that utilizes a loop to display the numbers 1 to 10 and the corresponding square of each number in a list on a
Assignment 6: Create an interactive object that represents a book with properties (title, author, year) and a method that returns a string describing the book.
Assignment 7: Build a simple to-do list application where users can add tasks, remove tasks, and mark them as completed, utilizing DOM manipulation.

Unit 3: Web Application Development using JSP & Servlets
Assignment 1: Create a simple Java Servlet
Assignment 2: Develop a Java Servlet that handles both GET and POST requests. Display a form in the HTML and process it to display submitted data back to the
Assignment 3: Build a servlet that demonstrates session tracking. Create a session that stores user information like username and displays it on a subsequent page.
Assignment 4: Create a JSP page that invokes a Java Servlet and displays data passed from the servlet as a response.
Assignment 5: Develop an application using JSP that features implicit objects. Use `request`, `response`, and `session` objects effectively to display user data.
Assignment 6: Create a web application that incorporates cookies. Set a cookie storing user preferences and retrieve it on another page.
Assignment 7: Build a simple JSP application that uses custom tags to display a list of books stored in a Java collection.

Unit 4: Spring Framework
Assignment 1: Set up a Spring project and create a simple bean using XML configuration. Display the bean properties on a webpage.
Assignment 2: Implement a Spring application that uses annotations for dependency injection. Display a message from the injected service on a webpage.
Assignment 3: Use the Factory Design Pattern in a Spring application. Create a bean that generates other beans using the factory approach.
Assignment 4: Develop an application that demonstrates the use of Spring AOP by creating aspects that log method execution times.
Assignment 5: Create a web application that demonstrates various bean scopes in Spring (singleton, prototype, session, etc.) and display the results.
Assignment 6: Implement a Spring application that uses auto-wiring for dependency management. Display both constructor and setter injection forms.
Assignment 7: Build a simple Spring application that utilizes lifecycle callbacks within beans, demonstrating `@PostConstruct` and `@PreDestroy`.

Unit 5: Spring Boot and RESTful Web Services
Assignment 1: Create a basic Spring Boot application with a simple RESTful API that returns a "Hello World" message in JSON format.
Assignment 2: Develop a Spring Boot application that accepts and processes user input via POST requests. Create an endpoint that takes a user's name in JSON
Assignment 3: Implement a RESTful service in Spring Boot that manages a list of books. Include endpoints for CRUD operations (create, read, update, delete).
Assignment 4: Create a Spring Boot application with an actuator and explore its metrics and monitoring features using Spring Boot Admin.
Assignment 5: Use Spring Boot to build a Restful API that integrates with a database using Spring Data JPA, showcasing how to save and retrieve entities.
Assignment 6: Develop a Spring Boot application featuring authentication (basic or JWT) for the Restful services. Secure your API endpoints.
Assignment 7: Create error handling and validation for your Spring Boot REST services, ensuring proper responses for invalid input and other exceptions.