SEOULTECH
Seoul National University of Science & Technology

Term Project for ITM 519: Full-Stack JavaScript RESTful Web Application

Project Overview

Description:

The scope of this project is "modern website design and implementation." Students will develop a full-stack RESTful web application using a modern JavaScript technology stack. This project encompasses a broad range of tools and practices including HTML, CSS, Node.js, Express, React, Git, CI, and security measures. The application must expose a RESTful API on the back end to manage resources and handle data interactions, while the front end (built with React) consumes these APIs.

Project Topic:

While the project topic is free for you to choose, here are two example ideas inspired by projects from previous years:

1. **Food Recommendation App:**
   Develop an application that provides personalized food recommendations based on diet preferences. Features may include integrated maps for locating nearby restaurants, user reviews, and dynamic filtering via RESTful endpoints.
2. **Fitness Tracking App:**
   Create a platform for tracking food intake and calorie counts. Users can log meals, monitor daily calorie consumption, and view analytics. All data interactions (e.g., logging, querying, updating entries) should be performed through a RESTful API.

Technology Integration

Your application must incorporate the following technologies and practices:

- **HTML/CSS:**
  Design a modern, responsive user interface with an emphasis on clean aesthetics and usability.

- **Node.js & Express:**
  Implement the server-side using Node.js and Express. Your backend must expose a RESTful API for resource management, including standard HTTP methods (GET, POST, PUT, DELETE) to perform CRUD operations.
    - **User Authentication:** Integrate authentication using Passport.js or another suitable Node.js technology.
    - **Security:** Implement basic security measures to mitigate common threats (e.g., input validation, protection against XSS and CSRF).
- **React:**
  Develop dynamic and interactive front-end components using React that consume your RESTful API.
- **Git:**
  Use Git for version control and collaborate using GitHub. (A clear commit history on GitHub is required, though it will not be part of the grading.)
- **CI:**
  Integrate Continuous Integration practices (unit tests) to ensure your application is reliably tested.

Grading & Deliverables

Points: The project is worth **30 points** of your final grade.

Final Presentation (30 points):

- Your final presentation should last approximately 10 minutes per team. With live demonstration of your application deployed.
- It should include a live demonstration of your deployed web application, a detailed explanation of your RESTful API endpoints, and a discussion of how user authentication and security measures are implemented.
- Supplementary materials must be submitted, including presentation slides, a deployment manual, and all source code via a GitHub repository.

Evaluation Criteria:

- **Implementation & Completeness:**
  The application must be fully functional and deployed online. All required features—including a fully RESTful API—must be implemented.
- **Technology Usage:**
  Correct and effective use of HTML, CSS, Node.js, Express (for RESTful API), React, Git, and CI practices.
- **User Authentication & Security:**
  Proper implementation of user login and basic security measures.
- **RESTful API Design:**
  The back end must follow RESTful conventions, with appropriate HTTP methods, status codes, and resource-oriented endpoints.
- **Final Presentation:**
  Clarity, organization, and depth of explanation during the presentation. Demonstrated problem-solving and troubleshooting skills.
- **Collaboration:**
  Each team member's contribution will be assessed via the Git commit history and supplementary materials.
- **Feedback:**
  Feedback will be provided during class time and office hours throughout the project lifecycle.

Team Formation

- **Team Size:**
  Groups of **3 students (one team of 4 students)**.
- **Collaboration:**
  Effective collaboration is essential. GitHub usage is mandatory and peer evaluations of team contributions will be considered separately.

SEOULTECH
Seoul National University of Science & Technology

Submission Requirements

- **Deployed Application (optional):**
  Provide a live link to your deployed web application.
  - If you only deployed your application locally, provide instructions on how to run the application locally.
- **Presentation Slides & Supplementary Materials:**
  Submit your presentation slides and any supplementary materials (e.g., local deployment manual, design documents) via the e-class system.
- **GitHub Repository:**
  A public GitHub repository containing all project files. The commit history should reflect consistent development progress and clearly show each team member's contributions.
- **Final Presentation:**
  The final presentation will be scheduled as per the course timeline (usually Week 14 or Week 15 of the semester).