**7SENG014W Web Application Development**

**Project 01 (Backend RESTful Service)**

1. **Project Description:**

* Students are required to build a .NET Core (C#) RESTful backend service (of their own choice) with the following features:
  + SQL/SQL Server (local or external) database with at least 5 tables and foreign key relationships.
  + Implementation of JWT token for user authentication.
  + Integration of Identity Framework for user management, roles, and access control.
  + Email service triggered upon user sign-up.
  + Use of a GitHub repository for version control.
  + Deployment of the service on Microsoft Azure.

2. **Project Requirements:**

* The backend service should provide RESTful APIs for basic CRUD operations on relevant entities.
* Implement user authentication using Identity Framework.
* Configure user roles (e.g., Admin, User) with different access permissions.
* JWT tokens should be used for user authentication and authorization.
* An email service should be integrated to send a confirmation email upon user sign-up.
* Proper error handling and validation should be implemented.
* Use Dependency Injection for better code organization.
* Implement logging for tracking application events.

**Updated Code Organization:**

3. **Models and Controllers:**

* All database models should be organized in the "Models" namespace or folder.
* Controllers should be organized in the "Controllers" namespace or folder.
* Authentication-related functionality (Identity, JWT) should be encapsulated within the appropriate models and controllers.

**Updated Authentication and Authorization:**

4. **Identity Framework Implementation:**

* Implement user authentication using Identity Framework, encapsulating this functionality within dedicated Identity-related models and controllers.
* Define roles (e.g., Admin, User) and assign appropriate permissions, ensuring the role-related logic is centralized.

5. **JWT Implementation:**

* JWT token-related logic should be encapsulated within specific models and controllers to maintain code modularity.

6. **Database Schema:**

* Database models representing tables should be organized within the "Models" namespace or folder.
* Establish relationships between tables to reflect a well-structured relational database.

7. **Updated Project Presentation:**

* During the project presentation, students should highlight the organization of their project into distinct models and controllers.
* Emphasize the benefits of a clean separation of concerns and how this contributes to the maintainability of the codebase.

**Additional Tips:**

* Encourage the use of folders or namespaces to organize code within the project.
* Discuss the importance of code readability and how organizing code into models and controllers aids in understanding and collaboration.
* Remind students to adhere to the Single Responsibility Principle, ensuring that each class or module has a single responsibility.
* Response should be in JSON format; Postman should easily consume the API.
* Attendance at presentations is required, and failure to participate in the presentation will be regarded as a non-submission.

**Deadline: 5th March 2024 (13:00)**