Info. Sec. Mgmt Task

Deadline: Saturday, 15th March 2025, 11:59 PM
Submission: Upload your project to a GitHub repository and submit the repository URL via the attached Google Form.

Task Description

Your task is to design a database and develop a RESTful API. The API should handle authentication using JWT and support CRUD operations on users and products while ensuring authorization via tokens.

---

1. Database Schema Design

Design and implement a database with two tables:

Users Table

- `id` (Primary Key, Auto-increment)
- `name` (String, Required)
- `username` (String, Unique, Required)
- `password` (Hashed, Required)

Products Table

- `pid` (Primary Key, Auto-increment)
- `pname` (String, Required)
- `description` (Text)
- `price` (Decimal, Required)
- `stock` (Integer, Required)
- `created_at` (Timestamp)

---

2. API Development

Develop a RESTful API to manage users and products.

Authentication

- Implement **JWT-based authentication**.
- Create a `SignUp` endpoint to register users.
- Implement a `Login` endpoint that generates a **JWT token valid for 10 minutes**.
- Protect certain routes by requiring a valid token.

User Operations

1. **POST /signup** → Register a new user.
2. **POST /login** → Authenticate user and return JWT token.
3. **PUT /users/{id}** → Update user details (Only authorized users with valid tokens).

Product Operations (Require JWT Token)

4. **POST /products** → Add a new product.
5. **GET /products** → Retrieve all products.
6. **GET /products/{pid}** → Retrieve a single product by ID.
7. **PUT /products/{pid}** → Update product details.
8. **DELETE /products/{pid}** → Delete a product.

---

3. Security & Best Practices

- Hash passwords before storing them in the database.
- Implement authentication middleware to validate JWT tokens.
- Ensure that only authenticated users can access protected routes.
- Use environment variables for secret keys and database connections.

---

4. Submission Guidelines

- Upload your complete project to a **GitHub repository**.
- Submit the **GitHub repository URL** through the attached **Google Form**.

Evaluation Criteria

- Correct database schema design with ORM

- Proper implementation of authentication and JWT

- API functionality for CRUD operations

- Secure password handling and token validation

- Code readability and documentation