Project Id: SCR845NGFF

Develop a RESTful Blog Application API

Objective: Create a RESTful API for a blog application that allows users to create, read, update, and delete blog posts, as well as manage comments.

Duration: 25 days

Task Description

Day 1-2: Requirement Analysis
- Understand Requirements:
  - Identify the core features and functionalities needed for the blog application.
- Define API Endpoints:
  - List the endpoints required for blog post and comment management (e.g., /posts, /posts/{id}, /comments, /comments/{id}).

Day 3-5: Database Design
- Design Schema:
  - Create a database schema to store blog posts and comments.
  - Example Tables:
    - `users`: id, username, password, email
    - `posts`: id, title, content, author_id, created_at, updated_at
    - `comments`: id, post_id, content, author_id, created_at
- Setup Database:
  - Configure the database using PostgreSQL, MongoDB or MySQL.
  - Create necessary tables and relationships.

Day 6-10: API Development
- Set up Environment:
  - Configure the development environment with necessary tools and frameworks.
- Implement Blog Post Endpoints:
  - Create Post: `POST /posts`
  - Read Posts: `GET /posts`
  - Read Single Post: `GET /posts/{id}`
  - Update Post: `PUT /posts/{id}`
  - Delete Post: `DELETE /posts/{id}`
- Implement Comment Endpoints:
  - Create Comment: `POST /comments`
  - Read Comments: `GET /comments?post_id={post_id}`
  - Read Single Comment: `GET /comments/{id}`
  - Update Comment: `PUT /comments/{id}`
  - Delete Comment: `DELETE /comments/{id}`
- Validation and Error Handling:
  - Implement input validation and error handling for all endpoints.

Day 11-13: User Authentication and Authorization
- Implement Authentication:
  - Use JWT for user authentication.
  - Endpoints:
    - `POST /register`: Register new users.
    - `POST /login`: Authenticate users and provide tokens.
- Role-Based Access Control:
  - Ensure only authenticated users can create, update, or delete posts and comments.

Day 14-16: Testing
- Unit Testing:
  - Write unit tests for each endpoint to ensure they function correctly.
- Integration Testing:
  - Perform integration testing to verify the interaction between different components.

Day 17-18: Documentation (Optional)
- API Documentation:
  - Use Swagger or Postman to document the API endpoints, including request and response formats.
- User Guide:
  - Write a brief guide on how to use the API, including authentication and example requests.

Day 19-21: Review and Refactor
- Code Review:
  - Review the code to identify any potential improvements or optimizations.
- Refactoring:
  - Refactor the code to enhance performance, readability, and maintainability.

Day 22-25: Submission
- Final Commit and Push:
  - Commit the final version of the code to a GitHub repository.
- Submit GitHub Link:
  - Submit the GitHub repository link in the task submission form.

Deliverables:
- Fully functional RESTful API for blog posts and comments.
- Database schema and setup scripts.
- Comprehensive API documentation.
- Unit and integration test cases.
- GitHub repository with the final code.

Tools and Technologies:
- Programming Language: Python/Node.js/Java
- Framework: Flask/Express/Spring Boot (based on programming language)
- Database: PostgreSQL/MySQL/MongoDB
- Authentication: JWT