



Project Proposal: Fitness Tracker API

Course: Applied Software Development (CS489)

University: Maharishi International University

Student: Osama Abdelhamid

Professor: Bright Varghese

GitHub Repository: [Fitness Tracker API](#)

1. Project Overview

The Fitness Tracker API provides backend services for a fitness tracking application, focusing on tracking workouts, diet, and fitness goals. Designed for flexibility and scalability, this project includes key components of backend software engineering, such as data persistence, API security, containerization, and continuous integration/continuous deployment (CI/CD).

2. Problem Statement

With health and fitness being a primary concern for many, tracking progress over time is essential for effective fitness management. This project provides an API that helps users manage workout records, dietary habits, and fitness goals securely and efficiently. By implementing a robust RESTful backend, the API will allow users to manage and retrieve fitness data with ease, creating a structured and reliable tool for fitness applications.

The project will integrate essential topics from the course, including software deployment, security, testing, and automation, making it a comprehensive exercise in backend software development.

3. Objectives

1. **Develop RESTful Services:** Expose endpoints for managing user workouts, diet entries, and fitness goals.
2. **Data Persistence:** Implement a reliable data storage solution using SQL and Spring Data JPA.
3. **Secure the API:** Secure endpoints using Spring Security for user authentication and authorization.
4. **Implement Unit Tests:** Ensure code quality and stability with thorough Java unit testing.
5. **Containerization:** Build a Docker container for easy deployment and scalability.
6. **Deployment on Microsoft Azure:** Deploy the containerized API to Azure for cloud accessibility.
7. **Automate CI/CD:** Use GitHub Actions to automate testing, building, and deployment processes.

4. System Architecture

The system architecture follows a layered approach for modularity and scalability:

- **Controller Layer:** Exposes RESTful endpoints and manages HTTP requests and responses.
- **Service Layer:** Contains business logic for handling fitness data, such as workout and diet tracking, goal management, and user-specific data operations.
- **Repository Layer:** Manages interactions with the SQL database via Spring Data JPA.
- **Security Layer:** Ensures authentication and authorization using Spring Security.
- **Containerization:** Docker for environment consistency and deployment.
- **CI/CD Pipeline:** Automated testing and deployment with GitHub Actions.

5. Technologies Used

- **Java 23:** Core language for the application.
- **Spring Boot:** Framework for developing RESTful APIs and managing the project's architecture.
- **Spring Data JPA:** For ORM and SQL database management.
- **Spring Security:** API security through authentication and authorization.
- **Docker:** Application containerization.
- **Microsoft Azure:** Deployment platform.
- **GitHub Actions:** CI/CD for automation in testing and deployment.
- **Postman:** API testing and verification tool.

6. Conclusion

This project will serve as a comprehensive exercise in backend development, enabling a practical understanding of RESTful services, data persistence, security, containerization, and automation through CI/CD. By deploying the application on Microsoft Azure, the Fitness Tracker API project will be accessible and demonstrate cloud-based deployment, showcasing a modern and efficient backend service design.