Final Project

The purpose of the final project is to showcase your knowledge and skills gained in the course throughout the semester, which is cumulative work from the first day of class to the last day of class. This project is a collaborative group project, simulating real-world corporate life where each group member does specific tasks delegated and works together collaboratively to reach the common end goal: a functional and robust final product.

You will work on this project in parallel with NMAD 261, developing a REST API server for this course while developing the front-end in NMAD 261 for the UI/UX. Your API server will handle all requests from the front-end, including but not limited to processing data, storing data, fetching data, inquiring additional databases from an external API resource to supplement, storing data with a third-party cloud provider. Your product must incorporate the API best practices such as exception handlings, data validations, and among other things.

This project will have three phases with specific deadlines, just like milestones in the real-world software development lifecycle.

Phase I

Due on April 11
at 11:59 pm

- Information gathering
- Project requirements
- Architecture requirements
- Identify external API resources (OpenLibrary API)
- Identify data requirements
- Design a database ERD

Phase II

Due on April 14
at 11:59 pm

- OpenAPI Specification Design
    - Endpoints
    - Properties
    - Inputs
    - Output
    - Response / Status codes
    - Set up mock API server

Phase III

Due on May 9th
Presentation on May 9th
at 10:45 am

- Implement from mock API server in your code (C#, Python, etc.)
- User Documentation
- Presentation

Project Description

Tiger university has hired your team to design a RESTFUL API that allows the students to sell books to each other. According to their requirements, the students must be able to do the following operations:

- The system must store the profile of all users, including and not limited to: Username, first, last name, email address, and profile picture.
- The API must be able to store book information such as author, name, edition, description, ISBN, associating courses, book condition(new, used, etc.)
- The API must keep track of the seller(the student who posted the book for sale) and the buyer.
- The system must keep track of all interested patrons of a book and the winning patron.
- Payment transaction. You are required to use a third-party payment provider to handle all transactions. For this project, you will be using the sandbox mode.
- A filtering system should allow students to search books based on author, title, ISBN, and courses.
- The book can be removed from the database prior to being sold by the seller.
- The book can only be removed by the seller who owns the book.
- The interested patron can withdraw their interest in a particular book.
- The system must display a history of all books the seller had posted.
- The system must display a history of all books the buyer bought
- Sellers may make modifications to the book after it has been posted.
- The seller must upload images showing the condition of the book, and those images must be stored on a cloud provider data storage such as AWS s3 bucket, Azure blob, etc.

Extra Credit Challenge – adds up to 5% to your final project grade

- Extra credit challenge is an Individual work, no collaboration with the group members.
- Create an API endpoint to show dashboard data. For example, a dashboard would show how many books were sold today, yesterday, last week, last month, etc.
- The dashboard endpoint would allow you to specify date range to summarize the results.
- Display the following:
    - Number of books posted in the past week
    - Top 5 sellers
    - Total sales earned
    - Total fees earned
    - Total views, top 5 views, etc.