CIS557 Project

Photo & Video-sharing Social Network APP -

Implementation (Backend) & API Testing (HW4)

Overview

The project will consist of the following stages, described in further detail below:

- Team formation: create a team of 3 students
- Sprints: you will meet weekly with your project manager to discuss the implementation of the project and to receive graded feedback
- Presentation: you will present your project in front of two members of the teaching staff at the end of the semester

In completing this project, you will learn to

- Develop software as part of a team using agile methodology
- Design a software system consisting of web components
- Develop a web app using React
- Develop a server-side application using Node Express and MongoDB
- Write documentation for a software system

GitHub Project:

We will use GitHub projects and **Extreme Programming** (XP) methodology when implementing this project. As a reminder, XP provides 29 simple rules to be followed in terms of **Planning**, **Managing**, **Designing**, **Coding**, and **Testing.**

1. You will create and configure a **project in your GitHub repository**. (see useful links below).
2. You should create a **wiki page in your GitHub repository** listing and describing your user stories and story points.
3. Each user story should be listed in the **GitHub's tracker** as an **issue**. You must label your issues and assign them to specific member(s) of your team
4. You will use the GitHub repo created when working on HW1

Overview:

This is the first step in implementing our app backend. In this assignment, you will:

- Implement an Express.js server running on NodeJS for the backend
- Use MongoDB as your database to store data
- Test your backend code

Since we are using MongoDB as our database, we are building a MERN (Mongo + Express + React + Node) application.

Make sure you follow the setup instructions carefully to avoid headaches down the line.

Setup:

Setting up the backend

- Run `npm init` to initialize your Nodejs project
- Install express and eslint
- Develop your backend in a separate branch, and do not forget to add the `.gitignore` file

Connecting to a database

- Sign up for an account here.
- A guide to setting up a free database with Atlas can be found here
- We recommend installing the mongodb driver to connect and send queries to your database.

App Specifications:

For this homework, you will implement the backend version of the following features

#	Features	Useful Packages
1	User registration	
2	Login/Auth	
3	User profile page	
4	Create post / Photo upload	
5	Activity feed	
6	Follow/unfollow users	
7	Comments	

At the end of this homework the above features should be fully implemented.

Testing and validation:

- To increase your code coverage, extract the logic from your components' files and put them in separate modules
- You will use jest and supertest to implement your tests
- Your tests must achieve **60% code coverage for full testing credit**
- Your code must be clean, readable, and ***ESLint errors and warning-free*** (Airbnb style), **ask the course staff before disabling any ESlint**
- In addition, all your code must be clean, readable, properly indented, and well-structured

Design:

- You will realize that your app implements some of the logic in the view (like most modern apps)
- It is likely that your implementation will not exactly match your design

- Since software development is an iterative process, it is fine to update your design to match your implementation
- However, be aware that a complete redesign of your app will likely slow down your implementation
- Meet regularly to address any design/implementation conflicts as soon as possible

Submission:

- Submit your work to Gradescope (include all team members)
- Put the URL of your GitHub repository in your Readme file
- Do **NOT** push `node_modules` to Github/Gradescope or **we will have to deduct points off your assignment**
- Download the gitignore file **here**, rename it to `.gitignore` and add it at the root of your GitHub repo (do not forget to push), it will ensure that the `node_modules` folder and other configuration files are not pushed to your repo
- Do not forget to commit your work to GitHub regularly.
- Only the last push before the due date will be graded.
- The due date is **December 2nd at 11:59PM**

Project Management

- Sprint grades account for 10% of the homework grade
- You should schedule your sprint meetings during the following dates:
    - Sprint 10: **11/11 - 11/18**
    - Sprint 11: **11/19 - 11/25 (meet after the break)**
    - Sprint 12: **11/26 - 12/2**

- Before your sprint meeting with your PM, you should:
    - Prepare your requirements backlog: identify all the tasks that need to be performed during the sprint
    - Create your milestones and issues in GitHub project: label and assign all issues to members of your team
    - Compute your project velocity
    - Identify any questions that you might have for your PM
- Your PM should fill out the sprint feedback assignment (in Gradescope) during or shortly after the meeting

Grading:

- The TAs should be able to download your code (from GitHub) and run it locally. **We will deduct points if your app cannot run on their computer**
- Include in your wiki any relevant information about how to install, configure and run your app
- You must fill out the individual members' contribution form (posted later)
- The member's contribution is the average of all their teammates' entries
- Contribution of each member defines the penalty for HW grade (for a 3 member group, ideally each member should contribute 33.3% of the deliverables)

Contribution (%)	**Penalty: % deduction from the group grade**
29.7+ (90+)	0
24.75+ (75+)	-15
16.5+ (50+)	-25
8.25+ (25+)	-50
<8.25	-100

- In case of a contribution dispute, the PM will check the GitHub project for assigned issues, commit ownership, pull requests, code contributions, and slack communications. To avoid any dispute, we recommend that you create a slack channel (or any other group chat) and add your PM to it.