CIS5570 Project

Photo & Video-sharing Social Network APP -

Implementation & Testing (HW5)

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

App Specifications:

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

#	Features	Description
1	Photo likes & unliking	
2	Editing/Deleting Posts (photo) & Comments	
3	Follower suggestions	
4	Tagging photos / @mentions in comments	
5	Live updates	You can use server polling or websockets
6	Privacy / Visibility control on photos	You can implement one of the following scenarios ● The author of a post can restrict who can see it or ● A follower can decide to hide a specific post on their timeline (the post should not be visible even after reloading the app or reauthentication)
7	infinite scroll	
8	Session management	We recommend using JWT but feel free to use any other technology
9	Security: account lockout policy	You should disable an account after a series if unsuccessful login attempts (for example 3). The account youd be lock out for a set period of time (for example 2 hours).
10	Security: invalidate web tokens after logout.	

At the end of this homework, you should have a fully implemented app

Extra credit:

You can earn 5 extra points (added to your project/homework total) if you deploy your app.

To earn extra credit the demo should be done on the deployed app.

The deployed app should work fully in order to earn the extra points.

There will be no partial extra credit.

Testing and validation:

- In addition tu unit and integration tests, **you will perform end-to-end testing using cypress**
- For end-to-end testing, **test 2 of the following scenarios**
  - **Registration**
  - **Posting a photo (including login)**
  - **Deleting a post (including login)**
  - **Following a user (including login)**
  - **Commenting on a post (including login)**
- To increase your code coverage, extract the logic from your components' files and put them in separate modules
- You will use jest, supertest, and cypress 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.
- **Submit your code to gradescope before your presentation**

Project Management

- There are no graded sprints for this homework assignment
- We strongly encourage you to keep meeting with your PM

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.