

CIS557 Project

Photo & Video-sharing Social Network

APP -

Implementation (View) & Unit Testing

(HW3)

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

Project Manager Assignment

Your team will be assigned a Project Manager (PM) shortly after September 6th, at which point you will receive an email regarding the organization of your Kickoff Meeting. Please reply to your PM's email promptly so that they can get the meeting arranged!

During this meeting, your PM will discuss the project timeline, the scope of work, and the various project deliverables and answer any questions you have about the organization of the project. Note that each project milestone (HW) will have a list of features to implement and deliverables.

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 frontend version of the following features

#	Features	Useful Packages
1	Posts likes & unliking	
2	Posts comments	
3	Editing/Deleting Posts (caption/replacing images/videos) & Comments	
4	Follower suggestions	
5	Tagging photos or @mentions in comments	react-mentions

Backend Mocking:

- Since we do not have a backend ready during this phase of the software development, we have to mock it!
- You will use <https://mockapi.io/> or [Json-server](#) to mock your backend/endpoints

Implementation:

- You will implement the wireframes associated with the features listed above
- You will implement all the endpoints in your API documentation associated with the features listed above
- The media files must be displayed on the view
- For the follower suggestions, use the following rule, if two users are following (at least) the same 3 users, they should appear in each other follower suggestion view.

Testing and validation:

- You will write unit tests (including UI) for all the code you wrote
- To increase your code coverage, extract the logic from your components' files and put them in separate modules
- You will use [jest](#) and the [testing-library](#) 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**
- Your CSS file must pass validation at <http://jigsaw.w3.org/css-validator/validator>.
- In addition, all your code must be clean, readable, properly indented, and well-structured
- **You may not use jQuery in this assignment.** Ask the course staff before installing any JavaScript library
- **You must mock your HTTP client packages**

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 **November 11th at 11:59PM**

Project Management

- Sprint grades account for 10% of the homework grade
- You should schedule a **kickoff meeting** with your PM between **09/08 and 09/11**
- You should schedule your sprint meetings during the following dates:
 - Sprint 7: **10/28 - 10/30**
 - Sprint 8: **11/04 - 11/06**
 - Sprint 9: **11/10 - 11/11**
- Before 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.