Final Group Project Instructions

CS 3398: Software Engineering

1. Introduction

Congratulations on all of your accomplishments and learning so far!

I want you to take a moment to think about how much you knew about software engineering and how comfortable you felt programming at the beginning of this class.

Now, I want you to reflect on where you are now and how much you have learned. I hope you can see the growth you have had — I am so happy to have been able to enjoy this journey with you.

Now it's time for you to take everything you have learned so far and apply it to our final project! In this project, you will design and create a web app in groups of 2 (groups of 3 only by exception). **We do not have a final exam in this class because this project IS the final (no meeting on Dec 7th)** 🙂. It will give you the opportunity to cumulatively apply your skills and knowledge into an idea you come up with on your own!

2. Due Date

The final group project is due December 1st, 2022 by 5:00pm CT. NO LATE SUBMISSIONS WILL BE ACCEPTED. DO NOT GET A 0 BY MISSING THE DEADLINE. Partial credit will be awarded if your project is not fully done so turn something in 🙂

You will submit via Gradescope, including your Github repository link, your Group Proposal, and your fly-deployed URL in your README. Each group will only have **ONE SUBMISSION.**

3. Group Selection - jump to rubric

The majority of you should already have an idea who your partner will be as we did an activity in class to finalize partner selection. If you were not in class, please find a partner using the ***#final-project-partner-search*** channel on our class Discord.

EVEN IF YOU ALREADY SELECTED YOUR PARTNER YOU MUST DO THIS
ONLY 1 SUBMISSION PER GROUP
You must fill out this form in order to finalize your group selection: CS 3398 Final Project Group Selection Form

4. Group Proposal - jump to rubric

In the past, you were given well-defined, concrete specifications for your project. Now — it is your turn to write your own! **The specs don't have to be final - in practice, they may change over time.** Your proposal will take a big idea and translate it into a concrete plan of action so you can split up work among your time. Your proposal must also explain what your project does and what its purpose is (Solution section).

- **You will be required to submit a planning document (by including the link to your document in your README when you submit your entire project), you can use the example we went over in class as a reference: Group Project Planning**
- **Idea suggestions and requirements:** Your project must not be a clone of some popular existing technology (Uber, Snapchat, Twitter) - you can always do a twist on one though. Also, your project may **not** be an extension of Project 1 (Music Discovery). Get creative!

- ***Hint:*** *Sometimes, looking at technologies you can leverage can help inspire ideas.* Check out all these APIs *- such as Reddit, Google Maps, Census Data, NASA, Twilio, Pokemon, Twitter, Open Weather Map, Yahoo! Finance, Yelp, and so much more!*
- Here's another (more up-to-date) list of APIs.

5. README.md - jump to rubric

Your README is critical as it is **what we will use to grade your project**

For details on what you must include, **check the rubric.**

6. Source Control - jump to rubric

You will be creating A SINGLE, PUBLIC GitHub repository this time. If we do not have access when we first grade it, you will get a -50% penalty for this section of the project.

Each team member will be required to submit at least 2 pull requests which must be reviewed and commented on by another team member before being merged. This means you must add your teammate as a collaborator in your Git.

For more information on creating and merging pull requests you can check out:

- Lecture 17: More Git
- Github documentation for Creating Pull Requests
  - Note, there are different methods for creating PRs I suggest you use the **GitHub CLI method**
- Github documentation for Merging Pull Requests

7. Deployment - jump to rubric

We will be using your deployment link to grade the project so it must be present

Fly.io is recommended, but any publicly available host is okay.

- Only one of you needs to set up the deployed link (do it early so you can all test to make sure that latest commits to *main* are working).
- You can add your teammate as a collaborator to your app in fly.io so everyone can see the logs and environment variables.
- **You will need to create a new fly.io account or delete your old instance so you don't go over the limit of 3 VMs. DO NOT DELETE YOUR INSTANCES FOR PROJECTS YOU HAVE NOT RECEIVED GRADES FOR (i.e If you don't see a grade in Canvas for a project, then don't delete it. If you see a grade in Canvas, it is safe to delete).**

8. Technical Requirements - jump to rubric

In order to give you as much flexibility as possible while also assessing the skills and knowledge we have gone over in this class, we will **only require completion of 4 OUT OF 6 Technical Requirements** out of this set of Technical Requirements

Here is the list of Technical Requirements (**YOU ONLY HAVE TO COMPLETE 4 OUT OF 6 OF THESE**):

- **App runs on Flask server written in Python (as in previous projects)**
- **Postgres Database used to persist data**

- **REST API Integration:** This means you must integrate at least one external API into your project. Here is a list if you are looking for ideas.
- **User login:** Basic login as required in Individual Project Milestone 2 is the minimum and will earn you a full 20 points

  HOWEVER you can meet a technical requirement AND get extra credit if you implement one of the following login methods
  - **Option 1:** Use third-party authentication to safely handle login. With an OAuth Login API such as "Sign in with Google," you can authenticate the user and then store their email in your own database with any app-specific data you need
    - Add Google Login to your React Apps in 10 mins
    - How to build Google login into a React app and Backend API
    - How to read environment variables in JavaScript code
  - **Option 2:** Implement email + password registration and login. For this case, you cannot store passwords in your database in plain text, you must store a hashed form. You will also need to make sure you handle authentication via Flask-Login.
- **Beautification:** This means you went well above and beyond in the development of your UI. A UI like we have had in previous projects will not cut it. It must look professional, this is the bar we will use to judge.

  Your app must be beautiful and presentable. This is vague, but use your judgment (and feel free to ask) in adding color themes, layout, typography. Imagine this as a startup - if you presented this to someone and asked for an investment, would they consider it polished and professional? That's our bar.
  - ***Hint**: Are you bad at styling with plain* CSS? *Check out Bootstrap (for create-react-app)*
  - ***Hint**: Check out Adobe's 4 Golden Rules of UI Design for ideas on improving your interface.*
- **React frontend:** Use React for your app's frontend

9. Collaboration Expectations - jump to rubric

It is required that each and every person contributes to the project. If someone is not pulling their weight, it is YOUR responsibility to:

1. **First, try to have a conversation with the person and resolve the issue**
   a. **Set a time for check-ins**
   b. **Make sure they are pushing code to the repo**
2. **Second, if you are unable to resolve the issue — notify the Professor ASAP**

If a person is not doing work (e.g. not making Pull Requests, not completing their tasks, etc.) and you do not let us know in advance — then the entire group will be penalized. A "bad partner" is not an excuse for not turning in your project by the deadline.

10. Extra Credit - jump to rubric

You will NOT be eligible for extra credit if you have not completed all of the required parts of the project above (Group Selection, Group Proposal, README.md, Source Control, Deployment, and Technical Requirements)

I am going to be giving an ***extremely generous*** extra credit opportunity for the final project for a few reasons:

- I want to give students the opportunity to push themselves and come up with ideas for stretch features and have fun
- I want to give those who do not have a passing grade in the class and those who would like to improve their grades in the class an opportunity to bring their grades up. **Anyone who wants to pass this**

class or get a great grade can do so as long as they do the necessary work.

In this project you will be able to earn 20 points for implementing a "Stretch Feature" outside of the requirements of the project. You can do a maximum of 4 additional features (for a total of 80 additional points).

These features could include additional Technical Requirements you did not complete as part of your 4 minimum Technical Requirements. **<— YOU DO NOT NEED A VERIFICATION CODE FOR THIS**

YOU WILL NOT GET CREDIT IF YOU DO NOT GET VERIFICATION CODE FOR YOUR EXTRA CREDIT FEATURE

This could also include Stretch Features you come up with and have verified with the Professor during office hours. **YOU MUST OBTAIN A VERIFICATION TO GET CREDIT FOR YOUR FEATURE, EVEN IF IT IS LISTED BELOW.** To obtain a verification, at least one member from your group must attend office hours, explain the feature to the Professor, and once the two parties agree — the Professor will provide a verification code which you must paste next to your feature in your README.

Some examples of "Stretch Features" include but are not limited to:

- Having a full-fledged profile/account management system
- Including a video player in your site
- Commenting system that allows you to like others' comments and delete your own comments
- 3D UI elements using external libraries
- …the possibilities are endless and up to you — if you have an idea, reach out to the professor and get it verified in office hours!

11. Rubric (170 points total)

This project is 25% of your final grade.

Group Selection (10 points)	10 pts	Your group has filled out **ONE SUBMISSION** of the group selection form
Group Proposal (30 points)	5 pts	Problems Section (at least 5, unbiased, exploratory questions)
	5 pts	Information-gathering (at least 2-3 bullets of answers to each question)
	5 pts	Analysis (at least 5 insightful takeaways overall from your questions/answers
	5 pts	Solution (**at least 5 solution statements explaining what your project does/what its purpose is**)
	10 pts	Specs (at least 10 detailed specifications with an owner and set of TODOs/questions)
	-50%	If we do not have Google Doc access when we first try to grade it
README.md (10 points)	2 pts	- Link to PUBLIC Github repo is at the top of README - Each member lists links to their 2 pull requests in the README as well
	2 pts	Google Doc link with access to Group Proposal is in README
	2 pts	fly.io deployment link is included in README
	2 pts	- 2+ examples of things you enjoyed about or learned from this project - 2+ examples of things you didn't enjoy or wanted to learn from this project
	2 pts	- List your 4 Technical Requirements in your README under a section called "Technical Requirements" - List any stretch features you undertook in your README under a section called "Stretch Features". You must also include your verification for your

		feature. (Leave empty if you did not complete any stretch features.
Source Control (10 points)	1 pts	Github repo is **PUBLIC**
	1 pts	Link to Github repo is at the very top of the README.md
	8 pts	**EACH** member of the group has submitted at least 2 pull requests which has been commented on by another team member and merged into the GitHub repo
Deployment (30 points)	30 pts	Successfully deployed on fly.io — link is in README
Technical Requirements (80 points)	80 pts	Implement **4 out the 6 Technical Requirements** (20 pts each): - Flask server - Postgres database - REST API Integration (minimum 1 API) - User login - Beautification - React frontend
Penalties (Deducted points) Note: There are **no late days** on this project.	-X%	If a particular team member is **not** pulling their weight as described by peer feedback or by code contributions to the repository, the instructor/TA will have final decision on penalties. It is **your** responsibility to let us know in advance so we can take action.
Extra Credit (+20 points each, maximum of 4)	20 pts	**You will not be eligible for extra credit if you have not completed all of the required parts of the project above (Group Selection, Group Proposal, README.md, Source Control, Deployment, and Technical Requirements)** You can earn 20 points for implementing a "Stretch Feature" outside of the requirements of the project. You can do a maximum of 4 additional features. These features could include additional Technical Requirements you did not complete as part of your 4 minimum Technical Requirements. This could also include Stretch Features you come up with and have verified with the Professor during office hours (You must include verification in your README). Some examples include: ● Having a full-fledged profile/account management system ● Including a video player in your site ● Commenting system that allows you to like others' comments and delete your own comments ● 3D UI elements using external libraries ● …the possibilities are endless and up to you — if you have an idea, reach out to the professor and get it verified in office hours!