Milestone 1 Proposal – Due Week 5 (Feb 14)

The project proposal is the first document required from the team. All the information listed below must be included and tools should be in place by the due date.

Team Name	Moniker to be used in all presentations and documentation.
Members	List of team members, first & last name.
Description	A short (2-3 paragraphs) description of the project. Provide enough information to explain what value your product will provide to users of your product.
Vision Statement	A simple, one-sentence statement describing the clear and inspirational desired state resulting from your team's efforts to create your product.

Motivation	Describe the background and reasons for developing this product.
Risks	What are the known risks that may prevent your team from completing this project on time? Risks could include: the working environment, lack of experience of the team in the area of focus, lack of access to a specific resource, etc.
Risk Mitigation Plan	A detailed plan showing how the team will mitigate each stated risk. Describe how you will succeed given the stated risks.
Version Control	Describe the version control method and repository you will be using for the deliverables created for the project. Github is strongly recommended. Once a repository is determined, you must share access to the repository with your instructor, our TA, grader, and all your project team members.
Development Method	Which software development methodology will your team follow? Describe the methodology and the features/steps you will follow. Common methodologies include waterfall, agile/scrum, iterative, spiral, etc.
Collaboration Tool	Select a collaboration tool for team members to utilize for coordination of their work and communication among team members. Popular tools are Slack and HipChat.

Submission format: This project milestone 1 submission can be included either as the Readme file to the github repository or you can also create a PDF named ProjectMilestone1_<TeamName> included in your github repository. (One person on the team needs to submit a link to your project's github repository on respective submission link in Moodle by due date)

Team Name: Out of Ideas

Link To Repo: https://github.com/Joshua-Ramos/groupProject

Team Members: Ruben Vargas, Joshua Ramos, Among Jahagirdar, Elliott Shugerman and Alexander Fisher

Description:

Our concept for this project involves a website in which students from a given university can communicate, share documents, and share resources, for a specific course. This website will be kind of like Chegg, or like Piazza. However, we want to make the user interface as simple as possible. We also want to allow users to freely explore posts related to different courses, not restricting them to a single course like Piazza does. This website will allow people to post notes, post digital copies of books, advertise used books for sale, and communicate with other students that may be in the same lecture for a given course.

For example, say that a student doesn't know anyone in lecture and can't make office hours for some reason. That same student could log into our website, enter their university, and find their particular course. In that manner, the website will direct the user to a page where others have posted resources for that designated course. They can post questions, share homework solutions, and maybe set up appointments to work on an assignment with other students. Another good aspect about our website is that there will be existing pages for the same courses taken years prior. Thus, resources for a given course will accumulate and all of this content will be directly accessible to the user.

Vision Statement:

Communicate, share, learn, and never sweat a class again by using our website (name TBD).

Risks

- Most prevalent risk is simply all of us barely have any web development background, at most simply playing around with web technologies.
- A risk related to the feature of the application would be potential academic dishonesty.
- A technical risk could be scaling file storage; many classes each have many files, and there needs to be an easy way for us to create a pipeline to upload these files, while not sacrificing for high latency.

Risk Mitigation Plan

- If a website were to be built, that may be a potential source of risk, as most of us have only touched the surface of web development. Since the website is simply a combination of HTML, CSS, and Javascript, and many of us do have solid development backgrounds, it would be simple to pick up the new technology stack. We could do a quick, simple online course (MOOC), so that the fundamentals are solid. As far as more detailed stuff, it's all a Google search away.
- Another risk, is related to the core features of our application. There's strong potential for academic dishonesty if the documents for a certain class are left unchecked by an instructor. As a result, there should be a way for university staff to be able to flag and remove documents that violate class policies.
- A technical risk can be how to handle numerous file uploads efficiently, such that latency is not significantly affected. Questions such as whether simply Dropbox should be used as part of our pipeline, or on the other hand AWS (Elastic Filesystem), are valid questions. One way to mitigate this risk is simply analyze which is the optimal combination of easy (for us) and high performance, and quickly make a decision.

Version Control

- Since the website is simply a mixture of static and dynamic(Javascript) content, with a backend (potentially in Python), it would be simplest to use one Github repository. Setting up a Flask or a Django application and storing it in one repo, where each of us in the group has our own separate branch could be best. If each of use were to focus on one part (i.e one on the backend), there could be a separate branch for each fundamental component.

Development Methodology:

We will use the iterative/incremental/evolutionary development methodology. There are a several reasons we anticipate this is the best choice:

- We are relatively new to web and app development, so the opportunity to regularly refactor and test will allow us to incorporate what we've learned from previous iterations.

- Our final product is a long way from fully specified, and it is not our intention to do so immediately. Rather, we plan to get a prototype up and running as soon as possible, put it to use ourselves, identify missing and potentially useful features, implement, and repeat.

- An incremental approach will mean that, when needed for a project milestone, we have a working product which reflects our latest efforts, with minimal crunch time.

Regular meetings will be held to discuss new features, bugfixes, or components, which will be implemented, integrated, and tested by teams of 1-3 people (i.e in "sprints"). Sprints will happen concurrently and with an emphasis on modest goals and fast turnaround. A team will create a git branch for the sprint which (ideally, in most cases) will be ready to merge with the master branch by the next meeting.

Collaboration Tool:

Slack