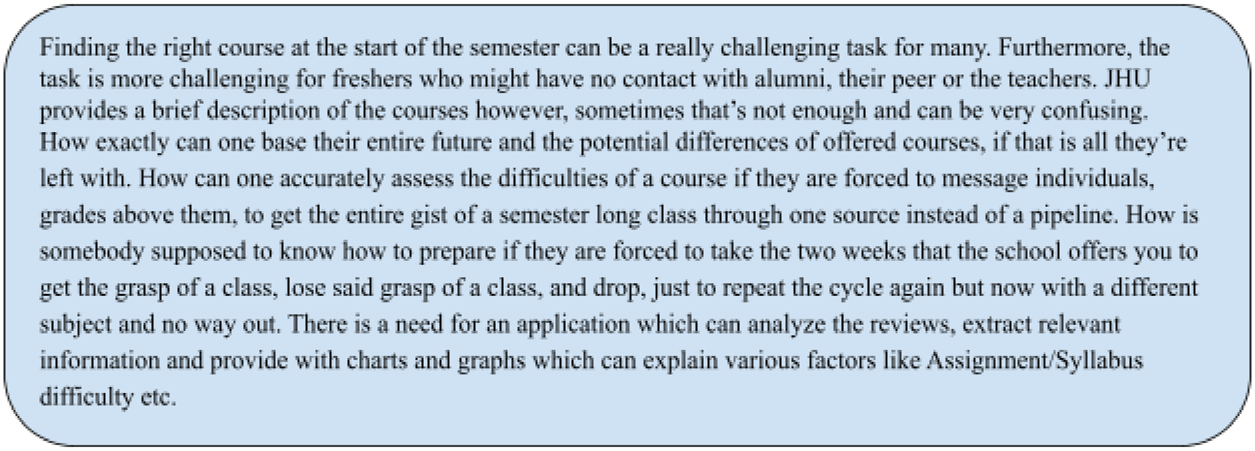
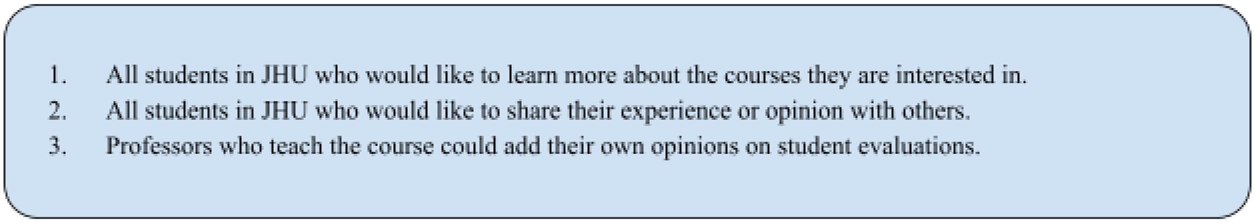
Software Requirement Specification

Problem Statement:



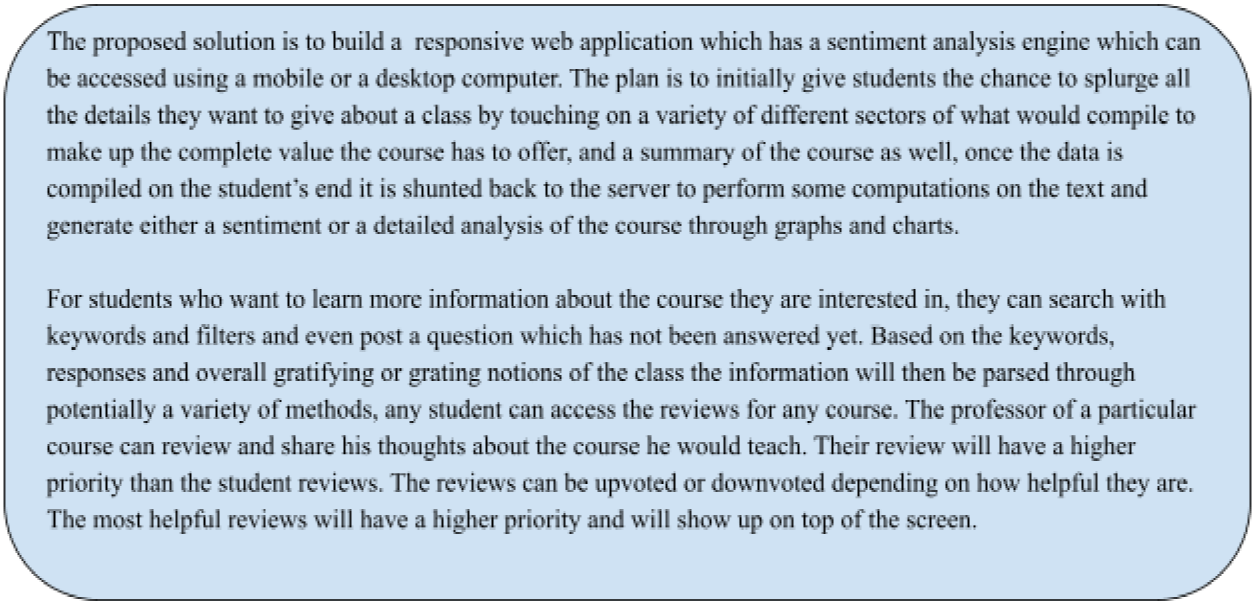
Finding the right course at the start of the semester can be a really challenging task for many. Furthermore, the task is more challenging for freshers who might have no contact with alumni, their peer or the teachers. JHU provides a brief description of the courses however, sometimes that's not enough and can be very confusing. How exactly can one base their entire future and the potential differences of offered courses, if that is all they're left with. How can one accurately assess the difficulties of a course if they are forced to message individuals, grades above them, to get the entire gist of a semester long class through one source instead of a pipeline. How is somebody supposed to know how to prepare if they are forced to take the two weeks that the school offers you to get the grasp of a class, lose said grasp of a class, and drop, just to repeat the cycle again but now with a different subject and no way out. There is a need for an application which can analyze the reviews, extract relevant information and provide with charts and graphs which can explain various factors like Assignment/Syllabus difficulty etc.

Potential Clients:



1. All students in JHU who would like to learn more about the courses they are interested in.
2. All students in JHU who would like to share their experience or opinion with others.
3. Professors who teach the course could add their own opinions on student evaluations.

Proposed Solution:



The proposed solution is to build a responsive web application which has a sentiment analysis engine which can be accessed using a mobile or a desktop computer. The plan is to initially give students the chance to splurge all the details they want to give about a class by touching on a variety of different sectors of what would compile to make up the complete value the course has to offer, and a summary of the course as well, once the data is compiled on the student's end it is shunted back to the server to perform some computations on the text and generate either a sentiment or a detailed analysis of the course through graphs and charts.

For students who want to learn more information about the course they are interested in, they can search with keywords and filters and even post a question which has not been answered yet. Based on the keywords, responses and overall gratifying or grating notions of the class the information will then be parsed through potentially a variety of methods, any student can access the reviews for any course. The professor of a particular course can review and share his thoughts about the course he would teach. Their review will have a higher priority than the student reviews. The reviews can be upvoted or downvoted depending on how helpful they are. The most helpful reviews will have a higher priority and will show up on top of the screen.

Functional Requirements:

Must have:

1. As a user I would want only reviews from JHU students or professors therefore the user login must be restricted.
2. The search bar should display all the courses for the students to see or atleast the courses the students are interested in. This will make accessibility for the app easy.
3. Reviews should be visible for everyone - even guests who do not login).
4. The reviews should be text-based as well as numerical. A sentiment
5. Only previous or current students should be allowed to give a review. This would prevent malafide reviews.
6. Charts and Graph analysis based on information given through the sentiment analysis form (Might not need a certain sample size threshold for this specific analysis, because even one review could lead to a world of information), these graphs have to relate to different parts of the sentiment analysis form.

Nice to have:

1. An in-depth analysis of the reviews.
2. A review voting system to put the most helpful reviews on top, this will give priority to the important reviews and make it easier for the student to see useful reviews.
3. Professor's own reviews may be displayed. Important to have the view of both the professor and the current/past students.
4. Recommended courses using a recommendation algorithm which displays the popular courses as well as the courses that suit the student. Make it easier to search for similar courses (i.e the courses the students are interested in)
5. Keyword detection. Before a comment is successfully posted, the app checks for blocked keywords to avoid things like abuse.

Non-functional Requirements:

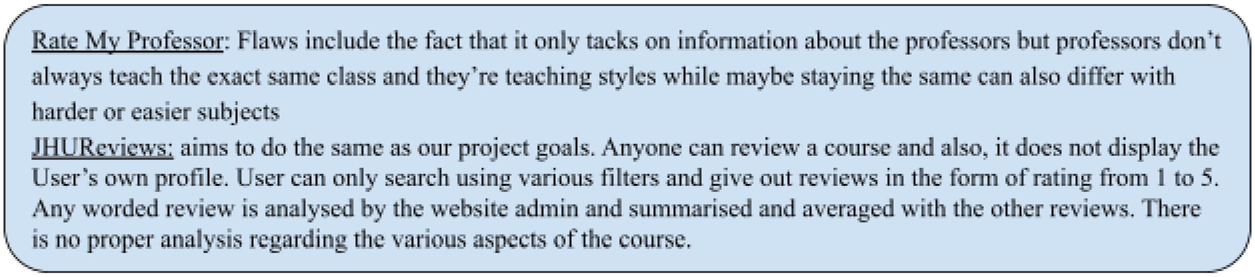
1. No huge load time: Any of the graphs or charts which are generated, as a result of the algorithms we use to account for any of the information being given, anything over 1-2 seconds is most likely overkill
2. Preferences don't bog each other down; making sure that they don't clash and slow down the program is a must.
3. Cross-Compatibility with Different Browser systems. A good user experience must be provided for a user who is accessing the website in any browser.
4. Security of the users are paramount, reviews can be anonymous if need be. Important info like password should be kept secure.
5. System meets the WCAG 2.1

Software Architecture & Technology Stack:

Website Application development; the most preferable is using the MERN stack
Language: Python for Data Analysis and Machine Learning.

Architecture: Client - Server Architecture.

Similar Apps:



Rate My Professor: Flaws include the fact that it only tacks on information about the professors but professors don't always teach the exact same class and they're teaching styles while maybe staying the same can also differ with harder or easier subjects

JHUReviews: aims to do the same as our project goals. Anyone can review a course and also, it does not display the User's own profile. User can only search using various filters and give out reviews in the form of rating from 1 to 5. Any worded review is analysed by the website admin and summarised and averaged with the other reviews. There is no proper analysis regarding the various aspects of the course.