Project Proposal

Project Title: CourseOverflow

Project Summary:
CourseOverflow is a centralized platform of all CS questions for UIUC courses.

The goal of our application is for university students to be able to easily access questions for courses they have previously taken, or courses they have not taken but want to learn more about. Our project will be built on an existing StackOverflow dataset to create test functionality, and a new database will be linked to the project website– this database will be populated with questions specific to selected UIUC Computer Science courses. Our new database will allow the creation of a singular platform for students to access relevant questions that have previously been asked for the selected courses. This is a classroom-specific feature for our application, but it can be utilized in other industries as a secondary source of Computer Science related FAQ.

Our project aims at eventually centralizing all the current platforms that students use for asking questions about their courses, such as Piazza, Ed, and Campuswire. This will help students easily access solutions to their questions, and decrease workload on professors and course assistants.

Description: Based on user queries, our website will process data and return posts based on similarity and relevance. Users will be able to access a myriad of filtration options as well, such as coding languages, or common topics within the course, time frame of the posts (recency) and much more. Our website also allows users to post new questions, and edit/delete their previous responses. Our project will be largely based on specific keyword targeting and aim to provide the most accurate results.

Features and functions:
Registration and Authentication: Users can create accounts and log in securely. This ensures that user data is protected, and it allows for personalization of the user experience as users will be able to keep track of their asked questions and be updated on all the questions being asked in their courses.
Search and Query: Users can search for FAQs and questions related to specific courses or topics using keywords. This forms the core functionality of the application, allowing users to find relevant information quickly.
Browse and Filter: Users can browse through a list of FAQs and filter them based on various criteria, such as course, coding language, time frame, or common topics within the course. This helps users narrow down their search results.

View FAQs: Users can view detailed FAQs, including the original question, answers, comments, and relevant metadata like the posting date. This provides context for understanding the information.
Ask New Questions: Users can post new questions related to their courses. This encourages active participation and the addition of new content to the database.
Edit/Delete Responses: Users have the option to edit or delete their own questions and responses. This ensures that users can maintain the quality of their contributions.

Creative Functionality: A creative component that could significantly enhance the functionality of our application is the implementation of an intelligent content categorization system. This system would filter posts based on various categories, such as coding languages, course topics, and difficulty levels.

We plan to achieve this by creating an algorithm that would identify key concepts, coding languages, and course-related terms within posts. Each post would then be tagged with relevant categories, providing a granular level of categorization.

With these detailed tags, we can offer highly personalized recommendations to students. When a user searches for a specific topic or coding language, the system would not only display FAQs related to the exact keywords but also suggest related topics and languages. For instance, if a student searches for "Python," the system might recommend FAQs on Python libraries, best practices, and even related languages like JavaScript or SQL.

To take it a step further, our system could compile a series of FAQs on a particular coding language or course topic and present them as a mini crash course. For example, if a student is interested in learning Python, the application could provide a curated list of FAQs covering Python basics, data manipulation, and common troubleshooting issues. This feature effectively turns the FAQ database into a learning resource.

Users could also have the option to create personalized learning paths. They could select a coding language or topic they want to master, and the application would generate a step-by-step plan based on relevant FAQs. As users complete each FAQ, they could mark it as done, allowing them to track their progress.

Usefulness: Our chosen application is useful because it provides a formal central platform for asking questions for courses at UIUC. The idea is based on the popular website "Stack Overflow", which is frequently used for asking technical questions related to any industry, but most commonly computer science/engineering. Our project aims at centralizing all the current platforms that students use for asking

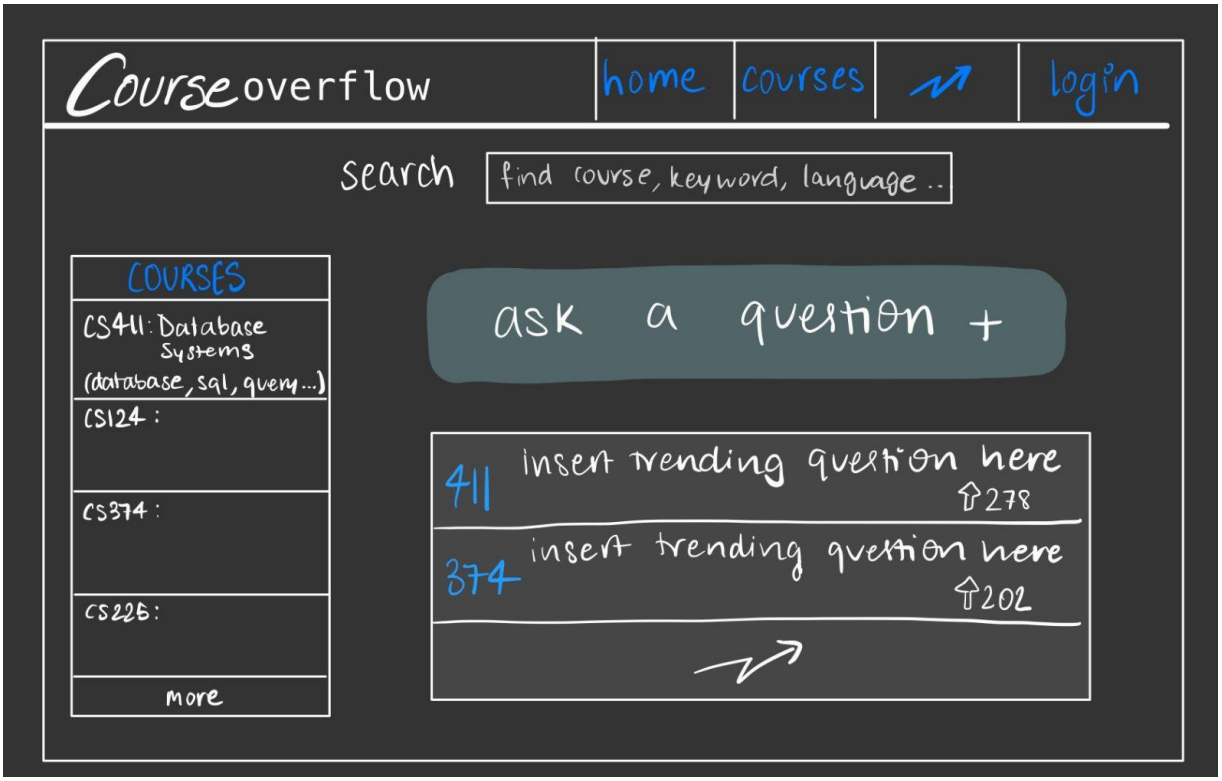
questions about their courses, such as Piazza, Ed, and Campuswire. These platforms do not give direct access to the questions asked in previous semesters or courses you have not taken, and requires students to be invited to the specific forums. By giving relevant search results to students by analysis of their query, the keywords involved in questions, and a few other factors such as query quality, our application can be used as a sort of a course FAQ bank managed by students/instructors/course assistants for the courses at UIUC.

Realness/Database:
We plant to use a Kaggle dataset that contains over 60,000 StackOverflow questions (https://docs.google.com/document/d/1W2vl_dd5voXU8vBiAx7tjqx1ud8MuX14MELyhj9OWzQ/edit). The questions are classified in High Quality, Low Quality with community edits and Low Quality without edits. For the purpose of our project, we will primarily be using the High Quality posts. The dataset contains information about the language for the post, the body of the post, and the title— all of which we would use for specific keyword targeting and grouping.

In the future, we aim to further update our dataset using the data from discussion forums such as campuswire and piazza for the courses at UIUC.

CRUD Functionality:
Create: The user is able to create their own posts in order to ask questions for specific classes.
Read: The user can retrieve specific questions by searching certain keywords or by looking up a certain class.
Update: The user can update a certain post by answering questions that have been asked by another user.
Delete: The user can delete posts that they have previously created.

UI Mockup:



Project Work Distribution:
Database cleaning, database normalization and relation: Vinit
Creating UIUC course database and grouping by courses: Dhruv
Creating user queries: Divyansh

Backend API and Q/A Testing: Meghna
Creative/UI design: Divyansh and Meghna