

TEAM 106 - DATA BUDS
STAGE 4
Mentor AI Project Report

Changes in Direction From Original Proposal

The direction of our final project slightly diverged from what we intended in our original proposal, as we tried to simplify the goal of our project a bit more and specifically focus on enhancing the user's experience with the chatbot rather than incorporating miscellaneous features. In our original proposal in stage 1, we planned on using the student's academic history, such as previous exam scores and grades, to evaluate their current understanding of the material and to accordingly adjust the kinds of questions the tutor will ask them. We did not implement this feature in our final project, as we knew we could not reasonably expect to get access to users' academic histories.

We also originally intended on adjusting the difficulty level of questions the tutor would ask the students depending on how well the user answered previous questions. We eventually decided against implementing this feature since we weren't able to find an effective method for assessing the user's proficiency level in relation to how accurate their answers were to difficult questions. Additionally, we found it challenging to determine the appropriate number of problems the user needed to answer in order to increase the difficulty level of questions. Instead, we tried to focus more on ranking classes by difficulty, so users can decide if they want to challenge themselves with harder topics. We also do not include tutorials, study guides, progress trackers, or group study sessions, making our application less personalized than what we initially envisioned for our project.

Usefulness of Application

We believe that our application will be a valuable resource for all students who take Computer Science courses at UIUC, especially since our application offers a wide range of topics

to choose from, with AI tutors available for all CS courses offered at the university. These AI tutors can ask users questions tailored to the topics for their selected course and provide users with feedback and detailed explanations, helping students gain a better conceptual understanding of the topics. This makes our chatbot particularly useful for students who are interested in getting extra help outside the classroom, as they now have an additional resource for practice problems for exams and explanations for certain concepts.

Unfortunately, we were limited in the variety of questions we could ask for certain classes due to the size of our dataset, thus our application proved to be more useful for AI related courses than others. We also couldn't make the AI tutors more personalized, since we did not have access to the users' academic histories and did not evaluate the proficiency level of the user depending on how well they answered questions, thus we could not adjust the difficulty level of problems. Though our application is not as personalized as we originally intended it to be, our AI tutors application is still a useful tool for students who are looking for additional practice in certain topics.

Schema or Data Source Changes

For the most part, our schema and source of the data for our application remained the same. We continued to use the same datasets of questions and answers for Computer Science concepts that we had at the beginning of our project.

Changes in Design:

We made several updates to our table implementations as the project progressed, including introducing a new table, *UserProgress*, and adding new data fields. The *UserProgress* table was created to address the need for tracking user progress across multiple courses. This table allows us to record details about each user's performance in individual classes and recommend resources tailored to their needs. Additionally, we expanded the initial design by incorporating new table entries to support enhanced functionality. For example, we added a

counter to track how often specific resources are utilized, providing valuable insights for further improvements.

Changes in Functionality

The main change in functionality between the app and what we have currently is that there is no group study function, but we increased the functionality of the app on the side of making sure that the individualized tutor is more isolated and personalized as we have added transactions to ensure only one user is accessing a tutor at any given time making sure we have no issues with data being sent to wrong users during the api calls.

Advanced Database Programs

We implemented several advanced database programs to complement our application. Using stored procedures, we determine what the most popular courses are depending on the number of users enrolled for those courses. We then output a certain number of the most popular courses, depending on the total number of enrollments, on the registration page of our application to show users what courses other users found to be the most interesting. We also use stored procedures to rank classes by difficulty, and we determine those rankings by calculating the average difficulty of problems for each course. We also output the hardest courses on our registration page so users can see which courses other users found to be the most difficult.

We also use triggers to adjust the difficulty level of problems. As users progressively start answering certain problems correctly, the difficulty level of that particular problem decreases, otherwise, the difficulty level increases. One of the triggers we implemented also randomly chooses three resources we have available and increments the recommendation count for the resources, so we can promote different sources for students to use for additional help.

To ensure the tutor is being used by only one user at a time (based on the course), we use transactions to lock the tutor. This prevents race conflicts by two users occupying the one tutor at

the same moment. We lock the tutor, update the tutor's current user information and once released another user can use it.

Technical Challenges

One technical challenge our team encountered involved integrating data across multiple sources into our database, specifically for the Problems table. Our biggest challenge with integrating data was finding a way to convert all the datasets of different formats into the same format, and merge all these datasets into one unified structure. Since each dataset followed a different format, we had to write several different Python scripts to parse through the datasets and clean them. For some of the datasets, certain questions would be missing the answer field, so we had to completely scrap those questions from the dataset. We also noticed that some of the datasets included html tags in the answers, so we had to clean the dataset by removing the tags before we could integrate it into our database. We also had one dataset that consisted of two different .csv files: one for the questions and one for the answers. For this dataset, we had to find a way to match up the questions with the corresponding answers and merge them into one .csv, formatted like the other datasets, with one column for questions and one for answers. After cleaning up the datasets, we could then merge all the data from the different sources into our Problems table.

Another technical challenge our team encountered was incorporating AI into the design. Due to the non-deterministic output of AI models it was many times hard to extract elements to query our database. Incorporating AI also added additional complexity through the calling of external apis which required more careful designing of asynchronous functions. In the end we had to do a fair amount of prompt engineering in order to prompt engineering + redesign a lot of the functions to work asynchronously in order to create a chat bot that would be usable.

Lastly, another challenge we faced was making sure our advanced queries involving multiple interconnected tables worked efficiently. As the queries often required the joining of User, Courses, Problems, and Tutor tables to fetch the data that is relevant to the functionality we

wanted to implement, such as assigning a tutor or preventing it to be used by more than one user at the same time.

Additional Changes

Some other changes that we made from the original proposal is the addition of an “Info” table. We provide additional links to websites and videos depending on the Topics (via the TopicID). This is to provide more resources to the students for better understanding.

Future Improvements

While our project has achieved its primary goals, we are planning on some future improvements to cater to the users needs. We aim to incorporate a larger set of courses beyond AI, and Python to increase the application utility and make it more diverse for students to use (an AI tutor for each course). We also plan to include group study sessions between users where they could discuss their questions based on their common courses that they are enrolled in. For improvements, we could probably build on what we had originally thought for making it more personalized by adjusting the difficulty level. As mentioned above we could not go along with it for now, however we can try finding an effective way to bring back this idea

Division of Labor

The work was divided into 3 parts:

AI Tutor : Philip Montgomery

Frontend Development: Shreya Kalyanasundaram

Backend Development and Database Integration: Shreya Kalyanasundaram and Shreya Sharma