Tutoring Connection – Western Oregon University Computer Science

Project Description

This web application takes a modern approach to connecting Western Oregon University Computer Science students, tutors, and professors. Additional resources make the application useful for students outside of computer science.

General uses upon completion

Computer Science Students can see real-time tutor availability, including notices of delays and extensions. Students can schedule sessions and receive prompt feedback on scheduling requests. Students can participate in online tutoring sessions through an interactive interface with chat capabilities. Students can upload documents to aid sessions.

Professors can submit grading requests, grading criteria, preferred due date, and notes to aid in the grading process. Professors can view general information about tutoring sessions to identify student needs. Potentially, data output is displayed on useful graphs.

Tutors can update availability, including delays and extensions. Tutors receive an alert when scheduling requests are made and can respond within the application or directly to the user's registered email. Tutors can provide online sessions, including visual aids. Online sessions are tracked to ensure tutors are compensated for their time. Tutors can track total hours worked so that timesheets are up-to-date and accurate. This is done either through manual input or totaled from scheduled times plus online sessions.

All Students can access bonus features such as a variety of calculators: GPA, weighted grade, "what do I need on my final?" and "how to raise my GPA". Registered users can store information for easy grades/GPA tracking throughout terms. Students benefit from up-to-date external resources through carefully curated links.

What is new/original about this idea?

The combined features and bringing such an application to our university is a first. This idea would modernize CSD tutoring, and potentially serve as the gold standard for effective tutoring and communication between tutors, students, and professors at WOU.

What are related websites/apps?

At WOU: The Math Center, The Science Center, and The Writing Center have a moderate online presence. Webpages display general information, tutoring availability, and appointments can be requested through a *separate* interface accessed through Portal. The Writing Center provides online video tutoring sessions upon request. Technically, students can request tutoring appointments for any subject, but our testing shows the system does not include locations and times for the computer science department (CSD).

Outside of WOU: Chegg.com provides online tutoring sessions with visual aids and chatting capabilities for a fee. Rapidtables.com provides calculators but no storage for future retrieval. TSheets is an app that helps employees track hours worked.

Why is this idea worth doing?

Impact. The CSD should have a web presence that demonstrates the potential of its students. It will showcase the abilities of seniors, and the teaching talents of the faculty.

Need. In a major that frequently loses students, either due to change of major or failing crucial classes, readily available support is essential. Additionally, our web app will provide students with an easy way to make peer connections—something we recognize as not just helpful but necessary for success in the program.

Insight. Data collected by tutors will provide valuable information to help guide professors in course design. Professors can identify areas of difficulty and potentially tailor course

schedules and/or pace of assigned work. On a larger scale, observations can be made that allow faculty to assess flow of the overall CS program.

Why is it useful and not boring?

Our web app solves a real-world problem and provides useful resources to CS majors, and additional features that expand its use beyond the CSD. It would prove useful from day 1, as the user need already exists. An app *for students created by students* is an exciting addition to the CSD. Our team would conduct research to evaluate user needs and implement features to address those needs. The finished product is done by few, but the ideas belong to us all and we find that concept appealing (i.e. not boring).

What are a few major features?

Online interactive tutoring sessions is one major feature. We would take the best features of existing platforms and combine them to create a design that addresses the specific needs of computer science students. Another major feature is data availability for professors. There are many complex and interesting results we can derive from data.

What resources will be required for you to complete this project that are not already included in the class? Additional API's, frameworks or platforms you'll need to use.

This project would need several API's. mailboxlayer API will enable us to thoroughly check and verify email addresses right at the point of entry into the system. PurgoMalum API will enable us to filter and remove content of profanity, obscenity, and other unwanted text. Possible APIs for data analysis include QuickCharts.io and Keen.io. We will likely use one or more to-be-determined APIs for online sessions.

What algorithmic content is there in this project? i.e. what algorithm(s) will you have to develop or implement in order to do something central to your project idea?

We have not identified a specific algorithm, but an algorithm could possibly be used in data analysis or in a "bonus feature" that recommends classes and/or areas of computer science to a student, according to their interests/some point of reference from student input.

Rate the topic with a difficulty rating of 1-10.

We give our project a difficulty level of 9. On the surface, our ideas are simple, but would require extensive backend work. Because we are using student information, handling registration, and more—security would be a major concern/priority. The online tutoring capability would be a challenge and vital to the success of the project. Data analyzation could wind up being rather complex if we use data to make *predications*—which would be the likely route since we will not have substantial data to draw from.