

Project Title:

Course Reco - Grainger Course Recommendation App

Project Summary:

Course Reco acts as a Grainger course recommendation app that helps engineering students find a set of courses that they can take next to fulfill their major requirements. Based on the user's interests and completed prerequisites, our application utilizes databases from platforms such as "Rate My Professor," "UIUC GPA Disparity," and "Course Catalog" to create a list of tailored course suggestions that are relevant for the user. Additionally, the application uses a weighting system that ranks these course suggestions based on the user's preferences for what classes they are looking for. These preferences can include what grade they want in the class, how many credit hours the class is, how hard the professor is, and whether it is online or in person. Our goal for this project is to create an efficient way for users to make informed decisions when selecting their classes, optimizing their college experience and academic success.

Description:

When time cards start rolling out, students are too busy with their current classes to spend the time to form their next semester's schedule with classes that will probably be too full by the time they register. Having to choose a class that fulfills degree requirements, is taught by a quality professor, and is genuinely interesting is a difficult and time-consuming process when having to navigate between multiple platforms to get a complete overview of the class. Even after finding the perfect class, most of the time all sections are filled before your assigned

registration time. Scrambling, left with no other options, students have to settle with the leftover classes that no one else wants.

To solve the problem and stress of registering for classes, our application will simplify the process of choosing classes. Course Reco will allow Grainger students to list their year, major/minor, any classes or prerequisites taken, how many credit hours they want to take, and their interests. The user will then be prompted with questions that gauge the type of classes they are looking for; this will help the app rank the recommended courses based on the user's customization. These questions can include course difficulty, professor ratings, average GPA distribution, and even how many times the course meets. All of this inputted information will then be pushed to the backend, where we will utilize the "Course Catalog" "Rate My Professor" and "UIUC GPA Disparity" platforms and the "Rate My Professor" API to analyze the best course matches. The created schedule will then be pushed back to the front end, where it will be available for the user to download. From the various options we give our users, they are bound to create their dream schedules with Course Reco even if their number one class choice is filled.

Creative Component:

Course Reco will portray all of these recommendations in an interactive and visually appealing manner. While the recommendations will be initially placed in order based on our algorithm, we will allow users to select each potential option to further view the attributes that make it a good candidate for a potential class, allowing them to manually sort through their options. These options will be displayed in a sequential list and will highlight and slightly enlarge when you hover over each class. Additionally, we can add a function where the user can select the courses they are interested in, and Course Reco will fit it onto a schedule based on the

course section times using a calendar-making API. This way, the user can visually see how each course will fit into their future schedule or if classes they want to take have conflicting times.

Further, Course Reco could also have an option to allow users to see further recommendations (if available) that become available or are recommended for taking a class that it is currently recommending. The ability to branch off of an initial search and look into specific future classes without ever leaving the original search provides a very simple and efficient way for students to gain foresight into the path options that each class that is being recommended opens up.

Usefulness:

Course Reco is useful because it simplifies selecting courses for engineering students by providing personalized recommendations based on their academic goals, past coursework, and personal preferences. Most students wait until the last minute to decide what courses to take because checking various websites to gain insight into course difficulty, workload, professor ratings, etc. is time-consuming. Some students believe researching what courses they should take is a “waste of time” because they probably will not receive the courses they want due to their late time ticket. By streamlining and making the selection process more efficient, students will be more encouraged to look into their different course options with Course Reco, building themselves a manageable and interesting schedule they will genuinely enjoy.

On the Course Reco website, one functionality includes the user’s option to enter courses they have recently taken, which will be used to suggest relevant follow-up courses that fulfill degree requirements based on the course catalog. For instance, if a student inputs CS 225, the app will automatically suggest courses that require CS 225 as a prerequisite, such as CS 411. The

user can also fill out a short questionnaire to rank their preference, including average GPA, Professor Ranking, Class Size, and Interest alignment. The app will use these provided characteristics to create a weighting system to rank courses accordingly.

Although resources are available to help students determine future courses, they are widespread and hard to keep track of. Our app integrates external sources into a single, cohesive platform. By pulling data from resources like “Rate My Professor,” “UIUC GPA Disparity”, and the university's “Course Catalog,” the app centralizes course information. This allows students to easily access professor ratings, GPA trends, workload, and course availability in one place, streamlining the process of course selection and schedule planning.

Realness:

The datasets we are using include UIUC’s “Course Catalog,” “Rate My Professor,” and “GPA Disparity.” The “Course Catalog” and the “GPA Disparity” datasets are publicly available for use by Wade Fagen-Ulmschneider, a CS professor here at UIUC. We can directly import the CSV files he created to utilize this data. Since only the raw form of the file can be viewed on Wade’s GitHub, we are unable to provide the data size of these two datasets; however, we can assume both are large since “Course Catalog” contains all of the available courses for the current semester and “GPA Disparity” has all of the average GPA’s for courses by section from now since 2016. We also plan to use data from “Rate My Professor,” but there is no publicly available dataset containing ratings from 2024. To work around this, we are going to use API calls instead.

Functionality:

Upon visiting our website, the user will be greeted with input fields for the user to select their grade level, major, the courses they have taken in the engineering department, and the total credit hours they want to take in the upcoming semester. Doing this will tailor the recommended courses to the specific major, making sure that courses previously taken will not be recommended and allowing certain courses with prerequisites to be recommended. When the form is submitted using the “submit” button, the user will then be taken to a different page for the user to add more preferences for the types of courses they are searching for. The page contains input fields for the user to write about their interests: ideal credit hour number, GPA level, professor score of their ideal course, and whether the course they want is in-person or online. **There will also be a clear button for each field, which will delete all information in the corresponding field. Once this second form is submitted, the user will be given a list of courses with the best-fitting courses put first. If this is a first-time user, a user and a preference entity will be created, and if this is a returning user, the preference entry fields of this specific user will be updated. For example if a student decides to switch majors, they will re-fill out this form and their major will be updated.**

We structured our recommendation system based on two pages. Based on what the user inputted on the first page, courses will be automatically discarded from or added to the list. For example, suppose the user selects Computer Engineering from the major field; all courses from other non-overlapping majors (such as Civil or Environmental) will be automatically disqualified from being displayed in the final recommendation list. For another example, CS 225 will not be added to the list of courses recommended unless the user were to input CS 173 and CS 128 (or ECE equivalents) as courses that they have taken; this is since the prerequisites of CS 225 are CS 173 and CS 128. Essentially, the first form gives a base of courses that have a possibility of being

recommended and ranks them based on popularity (i.e., most common for students to take CS 340 or CS 233 after CS 225), without any reference to what the user is searching for. The second page provides an opportunity for the user to input information to change the rankings of the most popular courses. We will weigh out each category and calculate a final score for each course based on what fields the user inputs and how closely the course section's attributes (professor ratings, average GPA, course description, credit hours, in-person/online) match the student's preferences. The final page will display the score given to each course section and rank them from the highest scores to the lowest.

UI Mockup:

<https://course reco.com>

Welcome!

Please enter the information below to help Course Reco assist you

GRADE

V SELECT

FRESHMAN
SOPHOMORE
JUNIOR
SENIOR

MAJOR/MINOR

V SELECT

COMPE
MECE
CS
BIOE
CIVIL
PHYSICS
...

CLASSES TAKEN

TYPE HERE

TOTAL CREDIT HRS

TYPE HERE

SUBMIT

<https://course reco.com/interests>

We Still Want More!

Course Reco wants to make your courses interesting too

INTERESTS

TYPE HERE

CREDIT HRS PER CLASS

V SELECT

1
2
3
4

Avg GPA

V SELECT

3.7 to 4.0
3.3 to 3.7
3.0 to 3.3
2.7 to 3.0
2.3 to 2.7
...

PROFESSOR SCORE

V SELECT

1
2
3
4
5

TYPE

V SELECT

IN PERSON
ONLINE

SUBMIT

<https://course reco.com/recommendations>

Based on all your information,

Course Reco Suggests...

Course Number	Course Name	Description	Credit Hrs	Prof Score	Matching Attr.
CS 411	Database Systems	...	3 or 4	5	SQL, data
CS 415	Game Development	...	3 or 4	4	AI, dev, game
...
...
...
...
...
...

will be more detailed on real website!

Project Work Distribution:

We plan to distribute the work based on the different website pages we have as well as assigning API implementation and user experience features as tasks. Below is an outline of the different requirements that will be distributed across members.

SQL Logic Basic Info Page (Jeremy Lee, Sofia Mark):

Referring to the UI mockup design, the front/home page takes in basic information about the user, including their year, major, classes or prerequisites they have taken, etc. All of this information from the frontend will be sent to the backend, where it will be used in the querying done by Jeremy and Sofia. Specifically, Jeremy will be responsible for using SQL to find a list of courses from the Course Catalog dataset that fulfill requirements for the user's specific major. From there, Sofia will use the user's provided prerequisites they have taken to then filter the data to list courses that the user is allowed to take. This information will then be carried onto the next step of the process below.

SQL Logic Preferences Page (Jeremy Lee, Sofia Mark)

After we have our preliminary list of courses created from the basic information the user has provided, we move into querying based on the second page of the website that asks for the user's interests and ideal course structure. Sofia will be in charge of cross-checking the courses we have listed with the GPA disparity dataset; this will ensure that all courses we return will be in the average GPA range the user requests. Jeremy will continue to query the data based on the user's preference for online or in-person classes, number of credit hours, and interests. We both will work on ensuring courses with high "Rate My Professor" scores by utilizing the

implemented API from Tony. We also will work together on developing a formula or system to weigh the different preferences of the user, which will determine the order of courses on the final list.

Implementation of External Platforms and API (Tony Leng):

Tony will be responsible for the implementation of external platforms and APIs. The platforms that we are planning on using include the “Rate My Professor” API (<https://pypi.org/project/RateMyProfessorAPI/>), and the “Course Catalog” API (<https://courses.illinois.edu/cisdocs/api>). We are planning on cleaning and filtering unnecessary data down to the relevant engineering course sections—including courses like math or computer science—with their corresponding professors. We are also planning on manually inputting the degree requirements for each major ourselves or finding a way to web scrape this information from the “Course Explorer”, in which Tony will implement the queries into the project interface.

Data Visualization/User Experience Features (Kyle Smith):

Kyle will be responsible for the user experience features of our website, including creating and inserting data visualization for GPAs of specific sections and the “Rate My Professor” scores. Additionally, Kyle will use an API to implement our creative component ideas, including a schedule visualization option. This feature will allow the user to check off potential classes to see how they might fit together into a schedule. At the end of the final page, the user will have the option to save all of the recommended courses as a PDF, creating an easily accessible list for the user to reference when it is time to register for classes. We will put all of these features on separate tabs, allowing the user to switch between the various attributes.