Software Development: BuffAI

James Simmons, Arianna Baer, Spencer Hoehl, Grigory Shatalin, Marcus DeLuca

Project Description:

Our idea is a more cohesive course planner for CU Boulder students that utilizes different tools and brings them all into one place. The system will have comprehensive knowledge of all courses offered at CU and will understand the requirements for each major and minor. This solution will provide students with the ability to find information about classes, the locations and distances between classes, and details about professors — all within a single webpage.

Functionally, our webpage will have you register with some basic information, such as student ID, email, major, minor, year, and type of degree. Using the Ollama (the AI model we chose) API, you can add the classes you've taken, and the model will recommend additional classes required for your major or minor. The website will also include functionalities that allow you to find possible elective courses based on your interests and skills, view Rate My Professor ratings for instructors of courses you are considering, and use a Google Maps plug-in to show you routes between your classes. Each location on the map will be labeled for convenience. You can choose between a pin and a search feature.

Our model has a working register, login, logout, home page, maps page, AI Advisor page, and a RateMyProfessor page. BuffAI helps with academic planning in many different ways. By centralizing academic planning into one single webpage, BuffAI aims to reduce stress and difficulty when planning your time at CU Boulder by offering a selection of different helpful tools.

Individual Contributions:

Arianna Baer:

I volunteered to work on the front end of BuffAI. First I worked on the login and registration page. This was needed before anything else could be done. I made a get and post request for each page in the index.js file. James helped me work on the registration post request because we ran into some bugs. Next I formatted the maps and RateMyProfessor page. I worked on linking my pages to the back end and making them cohesive with the other pages on the website. Lastly I finished by polishing the nav bar and adding images throughout the webpages. To summarize, I did the style and front end for the nav bar, login page, registration page, Maps page, and RateMyProfessor page and made get and post requests for each page.

Spencer Hoehl:

My primary duty was developing the front end of BuffAI and integrating it with the backend. I designed the front end of the main home page where you add classes and hobbies. With the assistance of Marcus, I made the get and post routes which ensured classes and hobbies were being inserted into the database reliably and appearing on the front end. I also worked on an html script for a calendar page which we did not have time to develop the backend or api routes for in the end. This script appears in my github branch nonetheless.

James Simmons:

I was in charge of implementing the Google Maps API and Rate my professor, however I had contributions in several other areas as well. I implemented the google maps api, wrote my own scraper to get live results off of rate my professor to display on the website, created a docker container for the Ollama installation, made all the dependencies and the model download on their own, fixed the /stream route to work in the new docker, re-formatted the database because it was improperly done and did not work, and made the /register route.

Grigory Shatalin:

I was responsible for researching, implementing, and integrating Ollama into BuffAI. I created the /stream route on the backend, which connects to the Ollama API, streams live AI responses, and saves conversations locally. I also set up the Docker containers to automatically

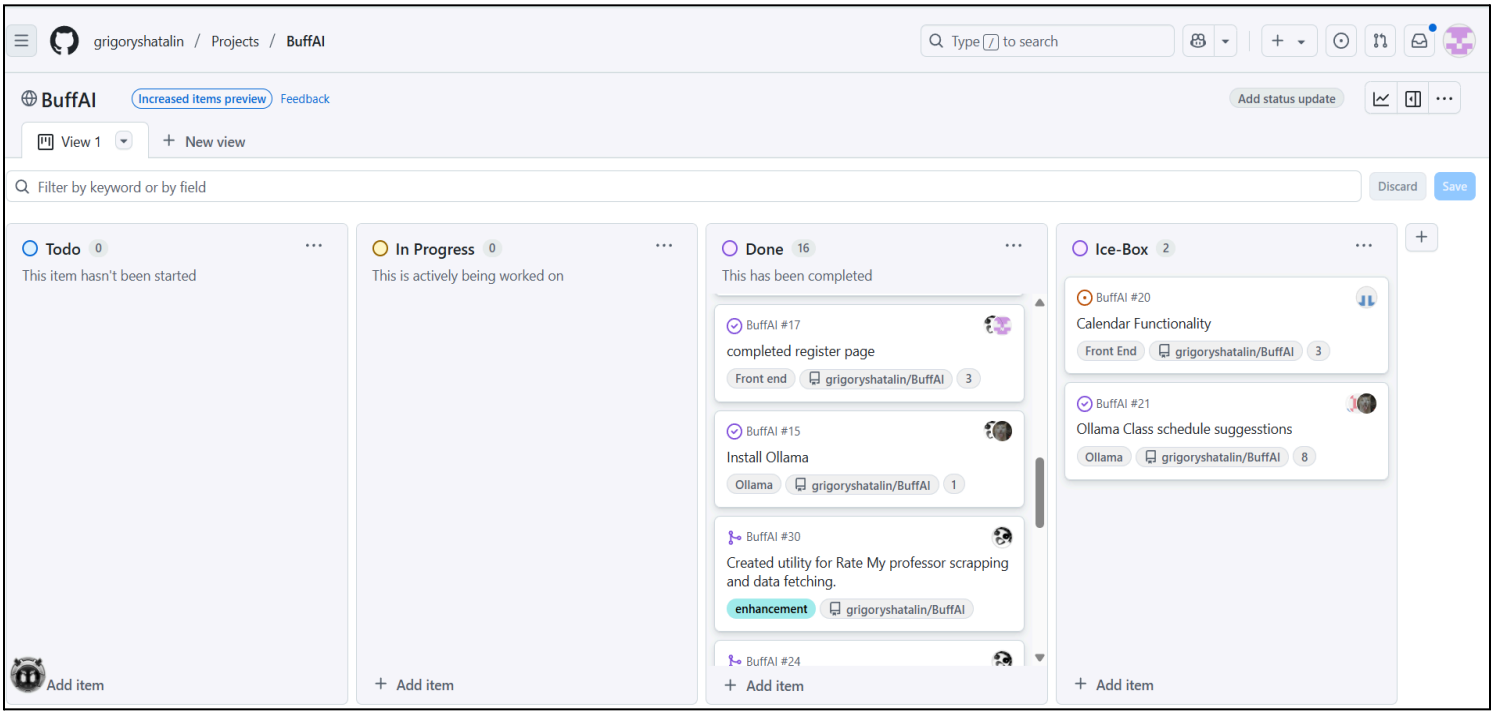
pull and serve the Ollama model. In addition to backend work, I built the front-end page that displays streamed AI responses in real-time, allowing users to interact smoothly with the AI advisor system.

Marcus DeLuca:

I volunteered to work on the back end – specifically, I created and filled all of the sql tables that our project would need. Each table needed its own parser that could go through the thousands of lines of data, and organize them in a convenient way for later use. After the database was created I also helped write post routes on the .js file such as queries used in the /login and /register routes. Another task was writing / editing the home post route, in conjunction with the add_class post route, so that the student_courses table could be correctly read and formatted on the home page.

Project Tracker - Github Project Board:

https://github.com/users/grigoryshatalin/projects/1/views/1



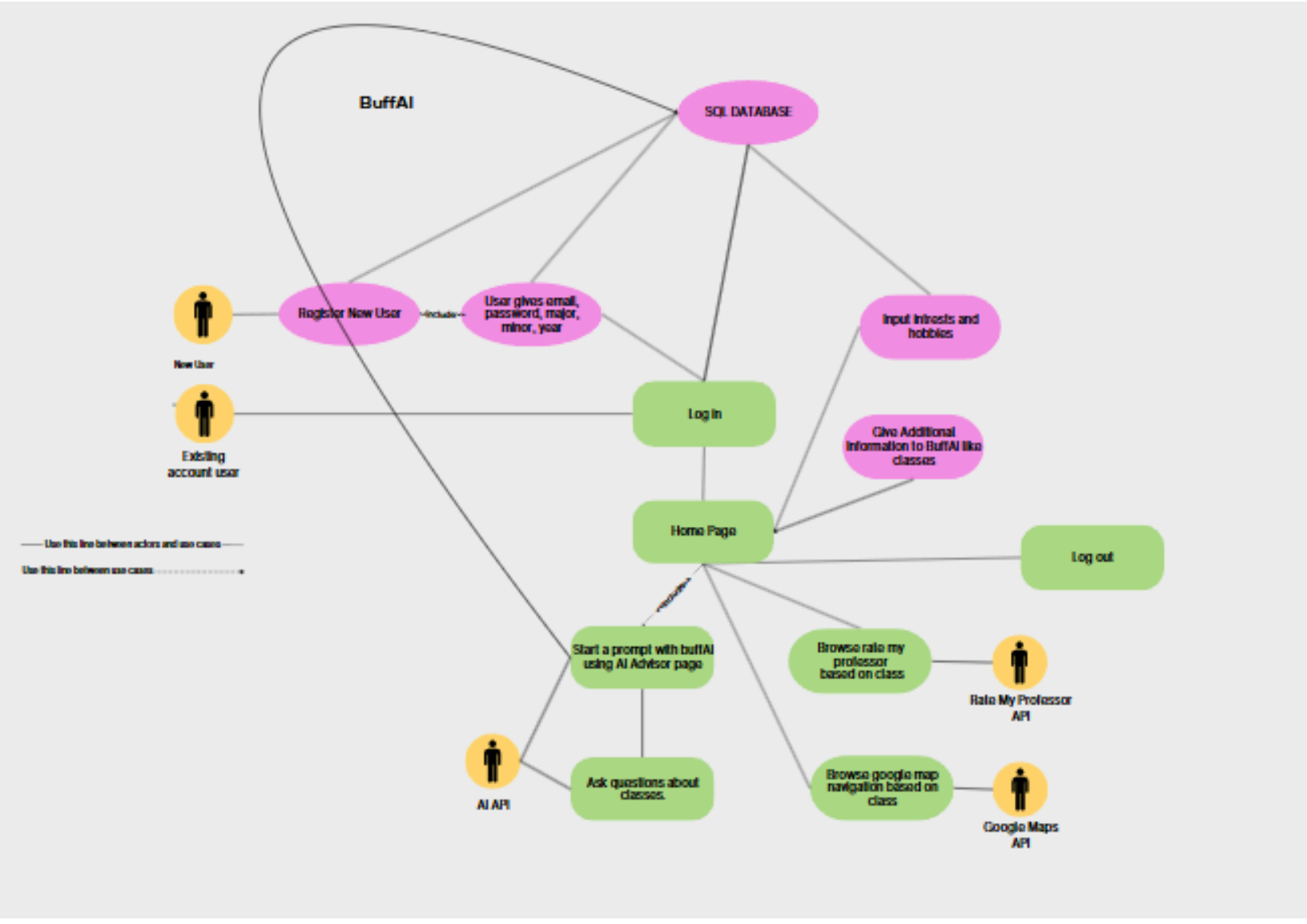
VCS:

https://github.com/grigoryshatalin/BuffAI

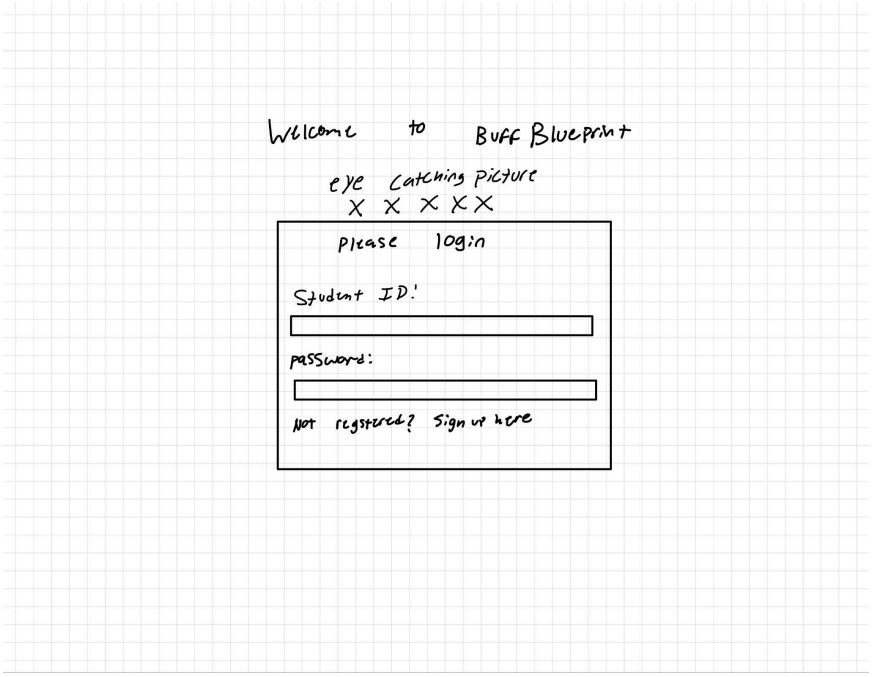
Video Demonstration:

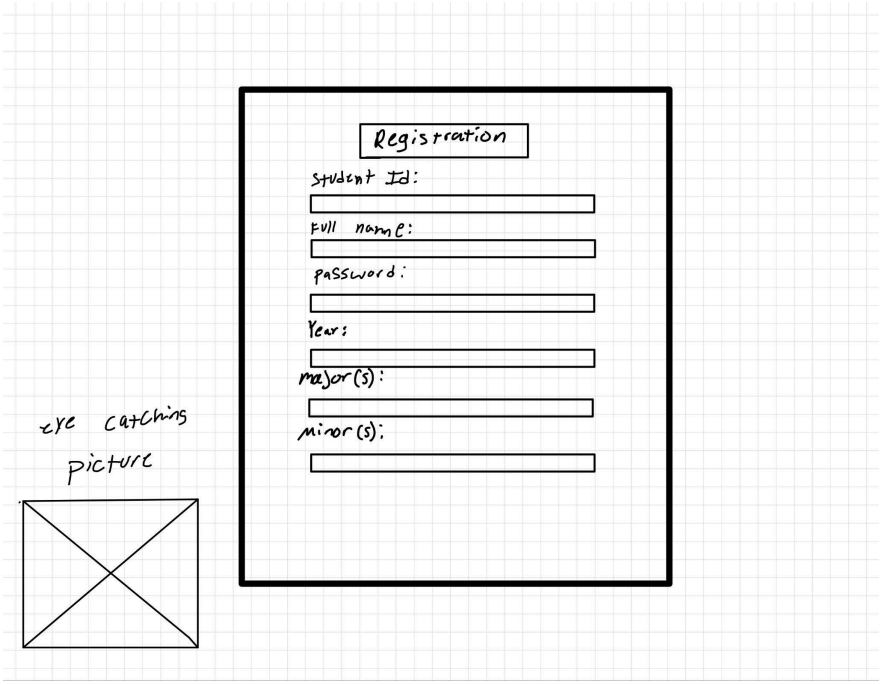
https://drive.google.com/file/d/1MdvLVVpw51Sp4Mldh-2GVUZOWMUwCrbz/view?usp=sharing

Case Diagram:

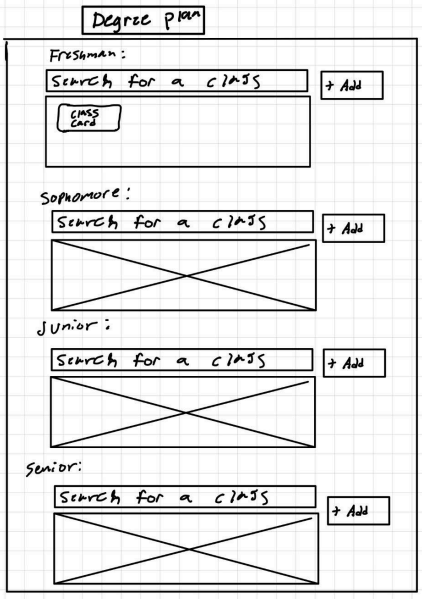


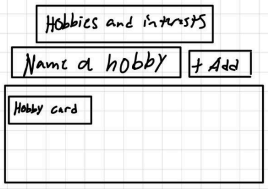
WireFrames:


Welcome to Buff Blueprint
eye catching picture
Please login
Student ID:
password:
Not registered? Sign up here


Registration
student Id:
Full name:
password:
Year:
major(s):
minor(s):
eye catching picture

Home page



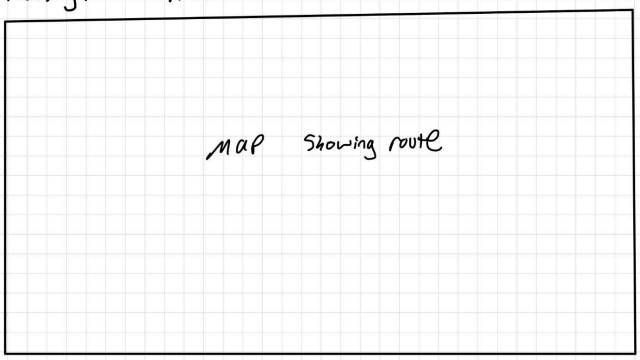


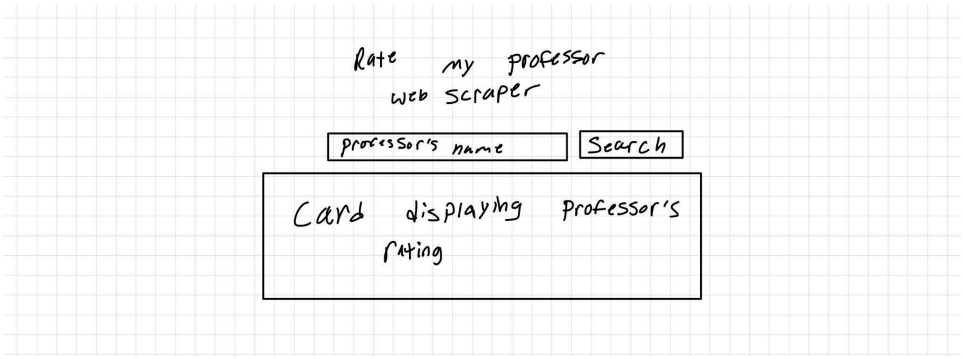
Classroom finder:

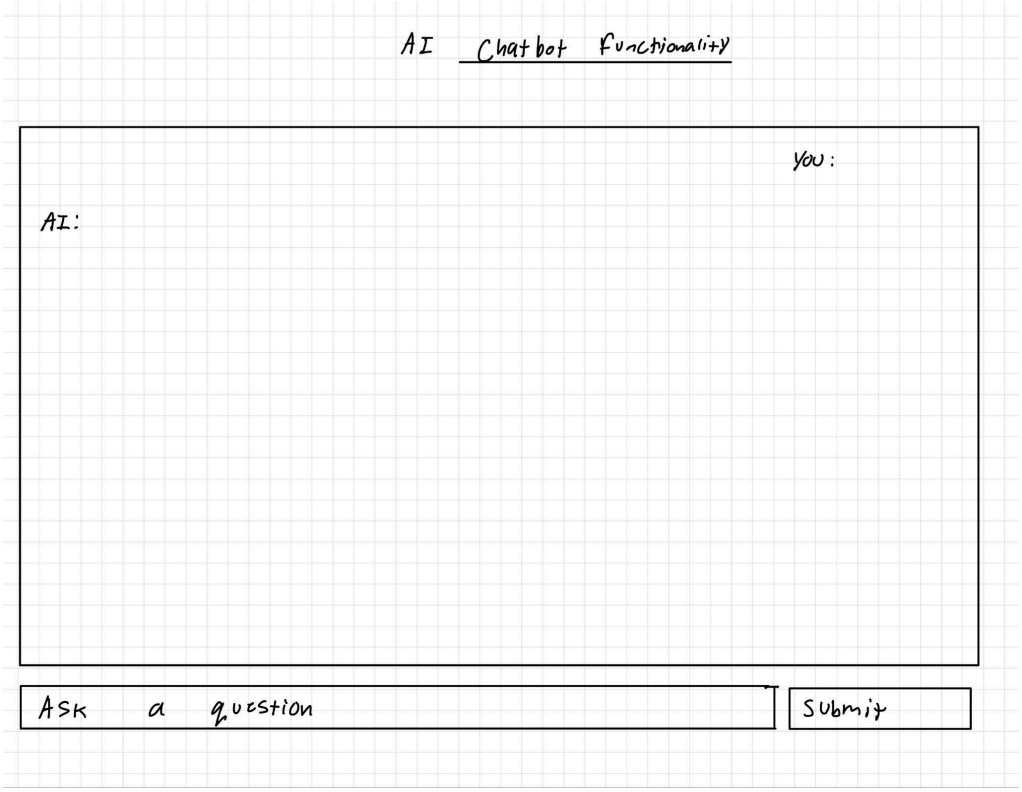
Choose a starting and ending location or place points on map.

Start: Ehd:

Routing: Time and distance between locations:







Test Results:

Feature 1: Ollama

Test ID: TC001

Result: Inputting “Hello” does yield a nonempty response from the AI chatbot. It answer may vary depending on when you say it in the course of the conversation, but it does response with a sensical response to “hello”.

Test ID: TC002

Result: The AI does not return a CSV file or any type of file for it to load up an example degree plan. We were not able to develop that functionality, but the AI does successfully recognize majors and minors input during registration and also knows what classes you have inserted yourself in your degree plan from the homepage. If you ask it to recommend more classes you have to take in your degree, it will do so, but will not give correct answers. It would take a better LLM to be capable of doing this.

Feature 2: Google Maps API

Test ID: TC003

Result: The map shows directions between 2 locations if they are valid input origins and destinations on campus. The map window will show a suggested route between the two locations and will give distance and transit time information.

Test ID: TC004

Result: Does not display any sort of warning. Instead, when you're typing it shows you the list of most similar valid locations to prevent that in the first place. You must actually click on a valid location so that it recognizes the location as being valid before submitting it.

Test ID: TC005

Result: Takes user input fields if they're valid Google Maps locations. You are able to either input them in search bars for an origin and destination or click the map and place two pins. Either one will route an appropriate path.

Feature 3: Rate My Professor Integration

Test ID: TC006

Result: API mimic returns professor ratings. The return is the top result from Rate My Professor. The professor's name, rating, and department are returned. The number of reviews is returned, but since there are usually multiple reviews we decided to not return any specific review.

Test ID: TC007

Result: Does not return an error, instead returns the most similar professor just like the Rate My Professor website. It is up to the user to determine whether or not this was the professor they were actually looking for.

Deployment:
https://buffai-1.onrender.com/