

# GroupMate: AI-Powered Student Team Recommendation System

**Track:** Development

**Team members:**

- Jocelyn Xu, [yux7@illinois.edu](mailto:yux7@illinois.edu)
- Krishnaveni Unnikrishnan, [ku18@illinois.edu](mailto:ku18@illinois.edu)
- Padma Pooja Chandran, [ppc2@illinois.edu](mailto:ppc2@illinois.edu)
- Aarthy Kesavan Padmanaban, [aarthyk2@illinois.edu](mailto:aarthyk2@illinois.edu)

**Project coordinator:** Jocelyn Xu

## Project Questions

*[Functions and Users] Define clearly what software tool you're planning on implementing. Clarify whether you plan to develop a standalone software tool, a new Web-based application, a new mobile app, an extension of an existing toolkit, or an extension of an existing application system. What are the major functions of the envisioned tool? Who are the users of your software tool/system?*

We are developing a **web-based application** designed to facilitate the formation of student groups in a course based on shared research interests, skill sets, and long-term vision for the subject. The application will also recommend project topics and ideas based on students' skills, vision essays, and other submitted abstracts.

The system allows instructors to onboard students into the course by providing a unique course code. Once enrolled, students can upload relevant documents, including their **resume, work samples, a code repository they are particularly proud of**, and most importantly, a **vision essay** outlining their aspirations for the course or their perspective on the future of technology. While all documents except the vision essay are optional, students are encouraged to provide more information to improve their chances of being matched with compatible teammates for course projects.

Using this data, our system employs **LLMs and a RAG model** to analyze student profiles, recommend optimal group formations, and even suggest potential project ideas aligned with their interests and expertise.

The primary users of this tool include:

- **Students**, who benefit from structured team formation and project recommendations.
- **Instructors**, who can oversee group formation and ensure well-balanced teams.

This tool aims to **streamline collaboration, enhance project synergy, and create meaningful connections** between students based on their academic and professional aspirations.

*[Significance] Why do we need the tool/system that you propose to develop? Does your tool/system address any existing "pain point"? How would our world be different because of your new tool/system? Does it address a societal need? Why is it important to address this need?*

Professors often spend significant time manually assigning student groups, leading to suboptimal team dynamics and inefficiencies; or students will try to form their own groups, but usually it takes a lot of time and energy searching for matching team members. Our system leverages RAG and LLMs to create smarter, compatibility-based group recommendations, reducing random mismatches and improving collaboration. By considering skills, preferences, and work styles, our tool ensures more productive and cohesive teams, ultimately enhancing the learning experience while saving time for both students and instructors.

*[Approach] How do you plan to build it? If this is a contribution to an existing piece of software, you should try to know the procedure for contributing to that software. If this is a standalone tool, identify what technologies you plan on leveraging to implement your software. This may be programming languages, supporting libraries, etc. What existing resources can you leverage? What risk or potential barrier do you anticipate and how you plan to mitigate the risk?*

We plan to build this as a web-based tool using Python, FastAPI, and React. To enable automatic group matching, we will integrate Retrieval-Augmented Generation (RAG) with LLaMA 3.2 to process student input such as courses taken, GitHub profiles, and resumes. Student data will be stored securely using PostgreSQL, with FAISS used for efficient vector-based retrieval. A potential challenge is making sure the group assignments are easy to understand and protect student privacy. To handle this, we'll clearly explain how groups are formed, remove any personal information from the data, and test the tool with small groups first to gather feedback and make improvements.

*[Evaluation] How will you demonstrate the usefulness of your tool/system and correctness of your implementation?*

We will evaluate our tool by analyzing the similarity scores generated by the RAG pipeline to assess how well student profiles align with their assigned groups. Additionally, we will conduct feedback surveys from students and instructors to gauge satisfaction with the recommended groups and suggested project topics. The surveys will ask users to rate the relevance and usefulness of the recommendations. This qualitative feedback will provide insights into the perceived effectiveness of our tool. Based on this feedback, we will iteratively refine the recommendation model to enhance compatibility and user satisfaction.

*[Timeline] Provide a rough timeline to show when you expect to finish what. List a couple of milestones if possible (they can be tentative).*

**Week 1:** Finalize requirements, set up development environment, and design input forms.

**Week 2:** Implement data ingestion, create database schema (PostgreSQL), set up FAISS for vector storage.

**Week 3:** Build RAG pipeline using LLaMA 3.2, integrate similarity scoring, conduct initial tests.

**Week 4:** Develop front-end with React, connect to FastAPI backend, display recommendations.

**Week 5:** Conduct small-scale testing, gather user feedback, and identify improvements.

**Week 6:** Optimize model, enhance privacy safeguards, and improve explanation generation.

**Week 7:** Perform final testing, evaluate performance, and deploy.

**Week 8:** Prepare documentation, create presentation materials, and deliver a live demo.

[Task division] Use one sentence to describe what each team member is expected to work on (can be tentative).

**Jocelyn Xu:** Develop the front-end using React, including student input forms and results display.

**Krishnaveni Unnikrishnan:** Build the back-end with FastAPI, handle data ingestion, and manage database integration with PostgreSQL.

**Padma Pooja Chandran:** Implement the RAG pipeline using LLaMA 3.2, integrate FAISS for similarity scoring, and optimize recommendations.

**Aarthy Kesavan Padmanaban:** Conduct testing, gather feedback, ensure privacy safeguards, and assist with implementation of the RAG pipeline and model evaluation.

**All members:** Collaborate on design decisions, provide feedback, and contribute to documentation and final presentation.