GROUP 5:
Quan Le
Jainil Patel
Gavin McClure
Rahul Gadhavi

CourseScope: Smarter Course Planning with Grade Insights

1. Title / Name : CourseScope

CourseScope is a web program that allows UIC students to arrange their semester schedules by combining degree eligibility tracking with historical grade distribution data. The software enables students to make informed choices by displaying which courses they can take next and how difficult each course may be based on previous grade trends.

2. Interactive Experience Introduction

Big Idea

Our project is a web application that helps UIC students plan their semester schedules by showing them which courses they are eligible to take and providing grade distribution data for each course. Students will input their major, check off which courses they have completed, and instantly receive a personalized list of available courses (Image 1). For each course, they will see visualized grade distributions by year so they can gauge course difficulty and plan their workload accordingly (Image 2). This centralizes course planning and performance insights into one interface, reducing the guesswork involved in semester scheduling.

Interactive Features & Engagement

Students will start by selecting their major and marking completed courses. The system will then display all remaining courses they are eligible to take, grouped by semester or requirement category. Students can filter these results by semester, course level, or grade distribution trends (For example: courses with the highest percentage of A's). Clicking on a course reveals a detailed grade distribution chart, allowing them to compare difficulty across years. This personalized dashboard encourages students to

revisit the site each semester, update their progress, and make informed decisions for the next term.

User Experience, Design Objectives, & Elevator Pitch

Our primary users are UIC students who need a clearer, faster way to plan their courses and ensure they remain on track to graduate. The interface will be interactive and visual: a checklist for completed courses, a dynamically updated list of eligible courses, and intuitive grade distribution charts for each course. The design emphasizes usability (minimal steps to update progress), clarity (visual representation of eligibility and grade data), and discoverability (filters and search to quickly find courses). **Our elevator pitch: "Our project turns course planning into a personalized experience. Students select their major, mark off completed courses, and instantly see the courses they can take next, along with grade distribution charts to help them plan their workload. This saves time, reduces stress, and ensures they stay on track for graduation."** This approach ensures we can gather valid data during our required user studies and iterate toward a more useful, polished tool.

3. Sketches & Diagrams

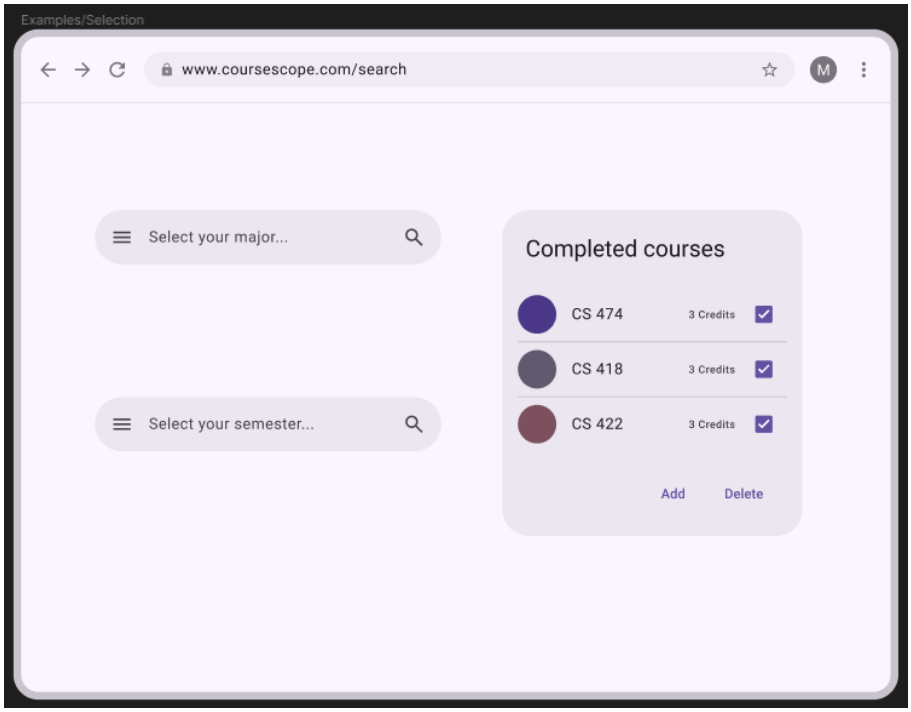

Image 1. Users can add their major, semester, and completed courses for each semester. Uses both client and server side code. A profile will be generated for users in which the courses they took and the semester they were taken are stored by utilizing a backend database.

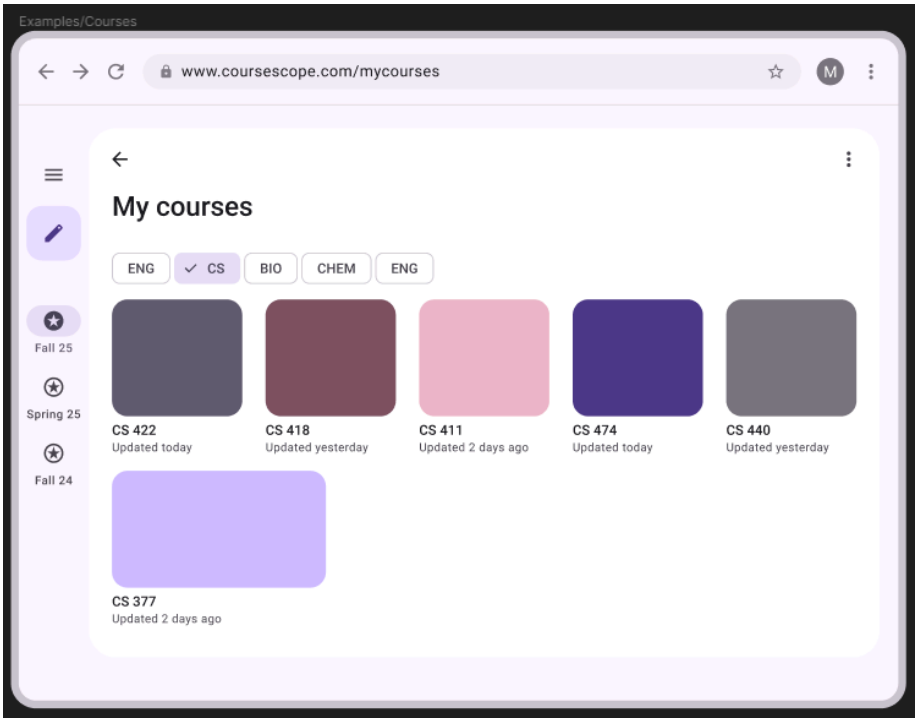

Image 2. Users are able to see all previously taken courses, as well as modify which courses they are currently enrolled in. This utilizes both client and server side code by pulling the user's stored course information from the database, and if they edit their selected courses the client will send an update to the server.

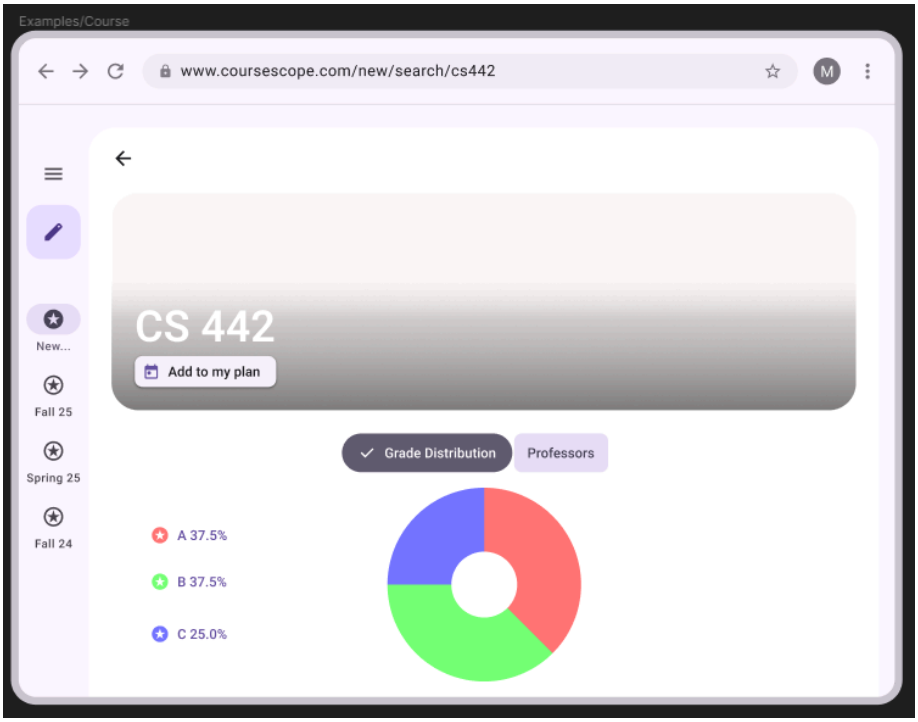

Image 3. Users can search for courses, and upon selecting a course are able to see the grade distribution and professors that taught the course. This section utilizes both client and server side code, and will get its information from the database where all course grade distributions are stored which will then update the graph on the page.

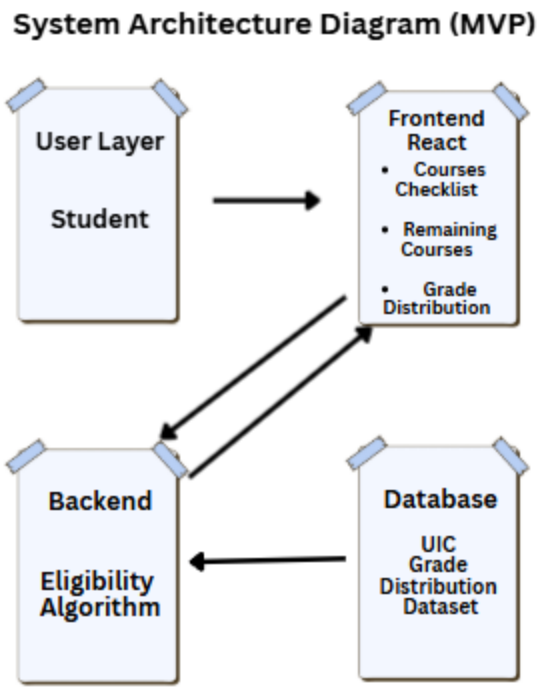

Image 4. System diagram showing the connection between user layer, frontend, backend, and database. Student's input will be processed by the frontend and sent to the backend via an API, which will query the database for necessary information and send results back to the frontend.

4. Users and Needs

Primary Users & Activities

- **User:** UIC undergraduate and graduate students planning semester schedules
- **Tasks include:**
    - Searching for courses
    - Comparing courses
    - Balancing workload difficulty
    - Maintaining or improving GPA

- **Current activities:** reviewing course descriptions, asking peers, and browsing UIC Data Portal grade tables

Problem / Need

- Need a single tool to track completed courses and show remaining options
- Visibility into grade distribution data to estimate workload
- Reduced reliance on manual checking of Degree Audit and PDF grade reports

Why It Matters

- Course selection is one of the most impactful student decisions each semester
- Current process is fragmented and time-consuming
- Research shows interactive decision-support systems improve clarity and efficiency [1]
- This project reduces stress, supports well informed choices, and promotes fair access to information
- Helps students stay on track for graduation and avoid course overload or GPA surprises by using historical grade data to plan schedules realistically.

Recruitment Plan

- Recruit through class group chats, Discord servers, and student organization mailing lists
- Find **at least 10 participants** for each user study (requirements, low-fidelity, functional)
- If needed, secure **written commitments** from ≥15 students to ensure valid data collection

5. Precedents

- List similar prior works (apps, systems, research papers, products).
- Do a comparative analysis:
- What do they do well?
- How is your project different or better?
- Include proper citations (APA style).
- Uicgrades.com [2]
    - Website developed by a UIC student
    - Does a great job of showing you grade distribution for courses
    - Does not store any user information
        - Unable to plan for future courses on website, which our project is
- Educational data mining with machine learning [3]
    - Utilizes deep learning and machine learning to predict best courses for students
    - Achieved a 78.04% multi-classification accuracy through a SVC (Support Vector Classification) model
        - Underscores the potential of machine learning and deep learning for student success predictions
    - Requires extensive knowledge of student's past course material and specific content performance
    - Interface is in a terminal, not designed for students to use
        - CourseScope will have a frontend that is easy to use for students
- Method and system for academic curriculum planning and academic curriculum mapping [4]

- Uses official university curriculum data
- Outlines software architecture and processes for implementing curriculum mapping
- Does not include grade distribution in its calculations, unlike CourseScope

6. Ingredients

- Enabling technologies:
  - Hardware: Laptop, PC to build the website, and smartphone for users to interact with the application for on the go convenience along with other operable devices
  - Software:-

    Frontend- Our team has decided to use React.js,(also Chart.js and D3.js) as a frontend software since it is one of the most popular frameworks to build interactive web applications and we have previously worked with it and experienced that it is easy and comfortable to real time update

    Backend- We are using Firebase (Image 4) as it provides us with hosting, database support, and authentication straight out of the box, allowing us to move more quickly and without worrying about infrastructure management. It also integrates with React with ease. Additionally, we are preserving MongoDB as an optional solution in case we decide to enhance its functionality or add more sophisticated querying in the future.

    API: For grade distribution and course information, we will use the **UIC API** or UIC's official Data Portal. This ensures the data is accurate and up-to-date rather than relying on static files. Using this API, we can pull real grade history and tie it directly into the visualizations

- Key Points:

  In short, our main ingredients are:

  Frontend: React.js Chart.js and D3.js (for interactive dashboards and components).

  Backend: Firebase (authentication, database, hosting).

  Database: Firebase Firestore by default, MongoDB optional if scaling is needed.

  APIs: UIC API / Data Portal (for grade distributions and course data).

  Together these tools line up with our system architecture: React handles the user interface, Firebase manages user data and backend logic, and the UIC API provides course and grade information that gets visualized in the app.

7. Innovation

- We believe our project **stands out because** it ties core student needs into one interactive site because it takes a process that is currently scattered across multiple tools and puts it into one clear, interactive platform. Right now,us UIC students have to check Degree Audit for requirements, browse the Course Catalog for descriptions, and then go through the Data Portal for grade distributions. None of these tools talk to each other, and the experience is slow and frustrating. Our system combines all three: it tracks course eligibility (Image 1), shows remaining

requirements (Image 2), and gives grade distribution charts (Image 3), all in the same place.

- In comparison to the already existing tools, we have decided to scale up a bit, not just providing simple data or checklist, the important part of our project that stands out more than others is combining the personalised academic progress along with the updated grade distribution. With this we help students judge the difficulty of classes and balance their workload based on real past data (Image 4). That's something existing UIC systems do not provide in a user-friendly interactive way

- **Minimum Viable Project (MVP)**: The following sets of features that we will have in our project will be very useful for any student saving their time and reducing stress by helping them manage their academic progress and register for future classes and suggest to them a manageable and balanced schedule throughout the semester. And the data will be stored locally.
    - Students select their major and mark completed courses.
    - The system displays eligible courses.
    - Each course includes a grade distribution chart pulled from UIC's data.

8. Project Plan / Scope

Scope:

Our course planning tool, CourseScope, allows UIC students to plan their semesters at a glance. It is built around two core features: degree progress tracking and grade distribution display. It's straightforward: students sign in, select their major at the College of Engineering (Image 1), indicate the courses completed, and immediately understand which courses remain. For each remaining course, the system displays old grade information so students can decide regarding the workload and difficulty of the classes.

This dual emphasis of following eligibility and accessing grade information addresses a typical issue students experience if they need to toggle between varying instruments, such as Degree Audit PDFs and the UIC Grade Distribution gateway. Instead of guessing or asking others, our project CourseScope compiles this data into a single dashboard tool.

1. **Data Integration: Backend and Cleaning**
    - Source data will be taken from the official UIC Grade Distribution data set (Excel files).
    - Gavin will meticulously clean and organize the dataset, transforming it into a usable format, such as JSON or SQLite.
    - For scoping purposes, the data set shall be limited only to courses offered in the College of Engineering, such as Computer Science, Data Science specializations, and the various programs in Engineering.
2. **Eligibility algorithm**
    - An underlying algorithm will compare the student's record major against a schedule of prerequisite courses.

- Students indicate courses done, and then the algorithm generates a timetable of courses remaining.
- For the MVP, we will hard-code prerequisites of College of Engineering majors (e.g., BS Computer Science with specializations, BS Data Science with specializations such as Bioinformatics, Business Analytics, Health Data Science, etc.).

3. Front-End User Interface (UI)
    - Course Checklist: A simple form or checklist wherein students mark courses finished (Image 1).
    - Remaining Courses Dashboard: An up-to-date roster of remaining courses in the degree plan (Image 2).
    - After selecting a course, you learn its distribution of past grades by semester and instructor.
    - Jai will create and implement the user interface in React, paying particular attention to usability, minimalism, and quick feedback.
4. Data Visualization
    - Distributions will be presented interactively in charts, e.g., pie charts of percentages of grades A, B, C, D, and F (Image 3).
    - Students may sift courses by grade tendencies (e.g., "highest percentage of A's" or "lowest past fail rates").
    - This enables wiser semester scheduling and workload distribution.
5. System Architecture
    - Frontend: React and Chart.js or D3.js for plotting.
    - Backend: Firebase for databases and backend programming.

- Storage: LocalStorage will be used for the MVP so students can store their progress between sessions. We can upgrade this with SQLite or Firebase for multi-user persistence if there's time.
- Source of Data: For the MVP, we will use a preprocessed export (Excel → JSON/SQLite) of the UIC Grade Distribution. As a stretch goal, we will integrate the UIC API/Data Portal directly.

Team Responsibilities: Team responsibility are divided into 4 parts based on the project progress;

1. **Rahul (Project Management, User Studies &Innovations)**

   Rahul will plan team meetings, oversee the project timeline, and ensure that deliverables are finished on time. By helping to shape the system's development and finding methods to enhance the user experience, he will also contribute to fresh concepts and innovations. Rahul will also create and conduct the user research, which will include participant recruitment, protocol preparation, and feedback interpretation to inform future developments. Regarding functionality, he will keep an eye on the connections between various components of the system and make sure the primary functions operate as planned. Rahul will contribute to the project's advancement and quality by combining management, creative, and functional duties.

2. **Jai (front-end & UX design)**

Jai will design the course checklist and dashboard, providing an unobtrusive environment for students to interact with the system. His priority will be to make the processes of course marking and viewing remaining options as smooth and simple as possible. He will also include grade distribution charts within the course detail view.

3. **Gavin (Backend & Data Handling)**

   Gavin is tasked with preprocessing and cleansing the UIC grade distribution Excel dataset. Additionally, he will implement the eligibility-checking algorithm that fuels the “remaining courses” list. His role guarantees that the integration of the backend or static dataset functions effortlessly with the frontend.

4. **Quan (System Architecture & Testing)**

   Quan will create the system architecture diagram, methodically organizing data flow between components. He will be in charge of uniting both frontend and backend systems, overseeing testing to ensure the eligibility logic and grade chart distribution function correctly.

Possible or not?

It's an ambitious but realistic project. By limiting ourselves only to the College of Engineering and the official UIC Grade Distribution dataset, we limit the scope to something we can handle. Our MVP will contain:

- A course checklist in which students mark off completed courses.
- An up-to-date, dynamic list of available courses.
- Historical grade distribution graphs for each remaining course.

- LocalStorage persistence for storing user progress.

This guarantees we will turn in a valuable and functional prototype within the semester schedule. Growth of other colleges or new features (recommendations, multiple filter types) can be entertained if time allows, but are not necessary for the MVP.

9. Skills

https://drive.google.com/file/d/1l0qz-mjFtvToY-Ig4ZR8nlgNZEPrUxM9/view?usp=sharing

10. References

1. Samaranayake, S., Gunawardena, A. D. A., & Meyer, R. R. (2023). An Interactive Decision Support System for College Degree Planning. Athens Journal of Education, 10(1), 101–116. https://doi.org/10.30958/aje.10-1-6
2. University of Illinois at Chicago. (n.d.). *UIC grade distribution: Explore and search courses*. https://uicgrades.com/
3. Kord, A., Aboelfetouh, A. & Shohieb, S.M. Academic course planning recommendation and students' performance prediction multi-modal based on educational data mining techniques. J Comput High Educ (2025). https://doi.org/10.1007/s12528-024-09426-0
4. Yaskin, D., & Ritter, G. (2012). Method and system for academic curriculum planning and academic curriculum mapping (U.S. Patent No. 8,265,968). U.S. Patent and Trademark Office.