CS/CE 353/374-L2 SOFTWARE ENGINEERING

MILESTONE MARVEL

Project Proposal, Interfaces, Features, Server & Tech Stack

Professor:

Yousuf Bin Azhar

Team Members:

Ifrah Chisti, Amnah Qureshi, Roozain Zehra, Ayesha Eiman

Course Swap Platform

This project aims to develop a course swap website for students at Habib University to exchange courses after the enrollment process. The platform streamlines the process of finding and connecting with students who are willing to swap or offer courses.

A. TECH STACK

- Front END : CSS Bootstrap
- Back END :
  1. Framework: Node.js + Express.js
  2. Database: MongoDB Atlas
  3. Authentication: Firebase Auth

B. SERVER + CLOUD

- Database & Storage (MongoDB Atlas)
  1. Store course swap requests in a cloud database.
  2. Enable real-time updates when a match is found
- Authentication & User Management (Firebase Auth)
  1. Secure student logins using university emails.
  2. Enable profile management and request history.
- Push Notifications & Messaging (Firebase Cloud Messaging)
  1. Notify users when a match is found.
  2. Enable direct messaging for swap negotiations.
- Web Hosting & API (Firebase Hosting)
  1. Host a web-based front end for accessibility.

C. FEATURES:

- Login/Sign-up
  1. Existing students can log in using their Habib ID and password (We would be using dummy passwords).
- Catalog
  1. Users can view the catalog page that contains all the open offers for courses

2. A students can click on the “Interested” button, notifying the original poster that someone is interested in their offer.

- Profile
  1. Users can view each other’s HU email addresses via the profile page to contact others. Users can optionally add their phone numbers.
  2. Profile will be automatically created for each user upon signing in.
- Put up course offering
  1. Students can put up an offer for a course swap OR request a course.
  2. They can also put up an open request i.e. any course that fulfils XYZ requirement OR any course that fits in a certain time slot
  3. Students can also request a course in case they do not have anything to offer.
- Algorithmic matching
  1. When a student puts up an swap offer or request, an algorithm would suggest existing offers that align with the students request

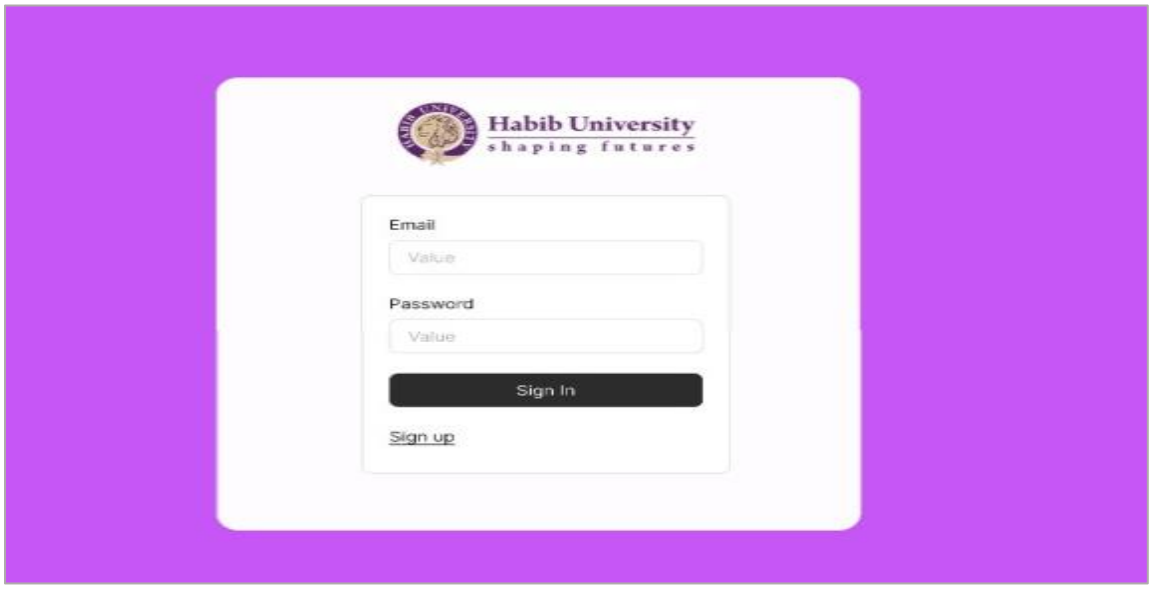
RESOURCES:

Firebase:
https://firebase.google.com/docs/cloud-messaging
https://firebase.google.com/docs/auth
https://firebase.google.com/docs/hosting

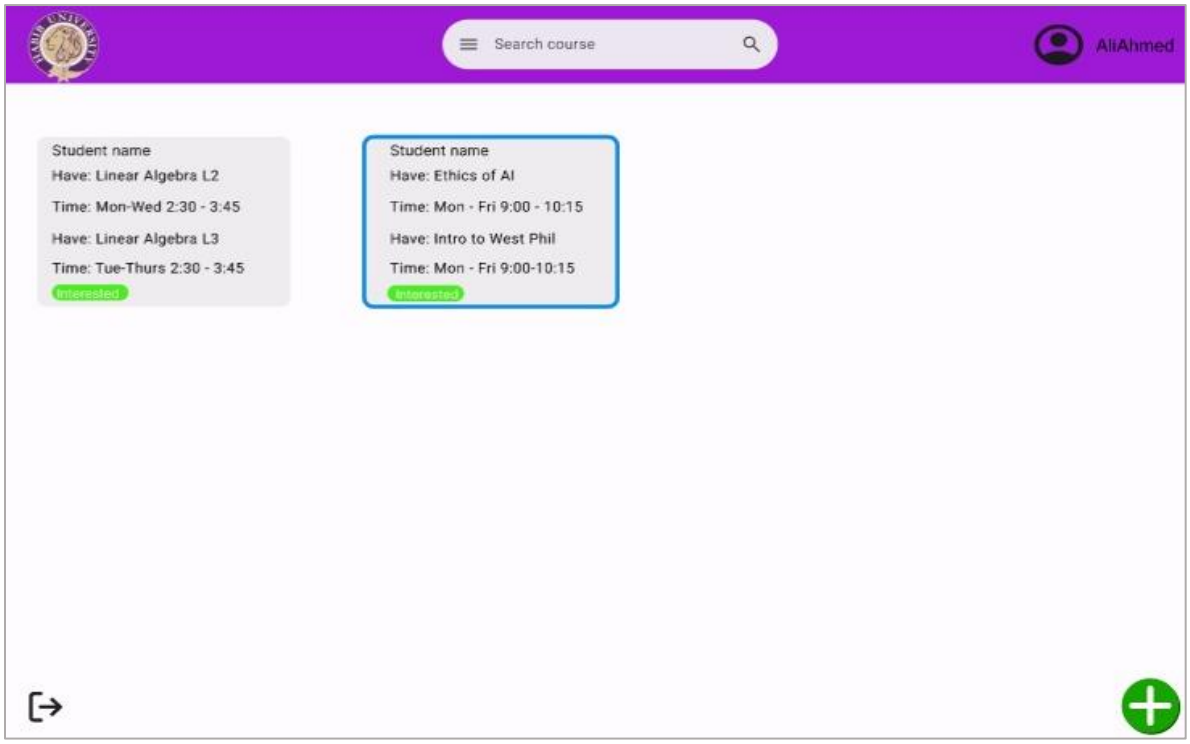
MongoDB:
https://www.mongodb.com/resources/basics/databases/cloud-databases
https://www.mongodb.com/docs/atlas/getting-started/
https://www.youtube.com/watch?v=bBA9rUdqmgY

BASIC INTERFACE:

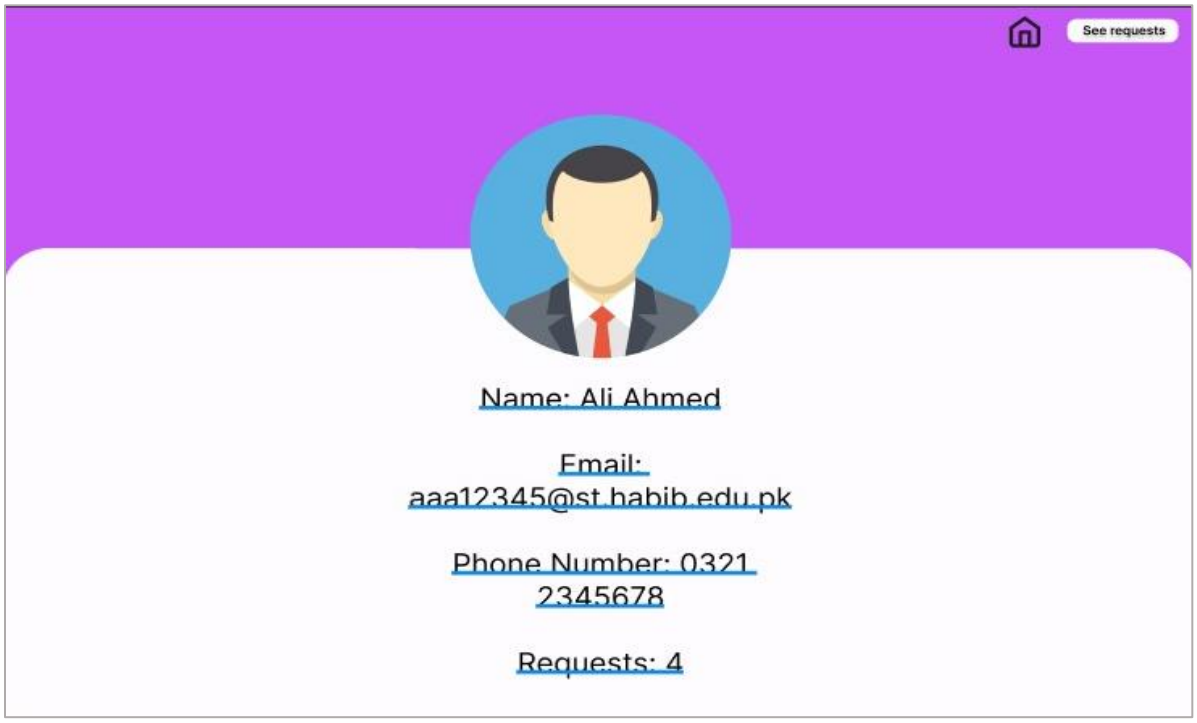
1. **Login/Signup with your HU credential:**



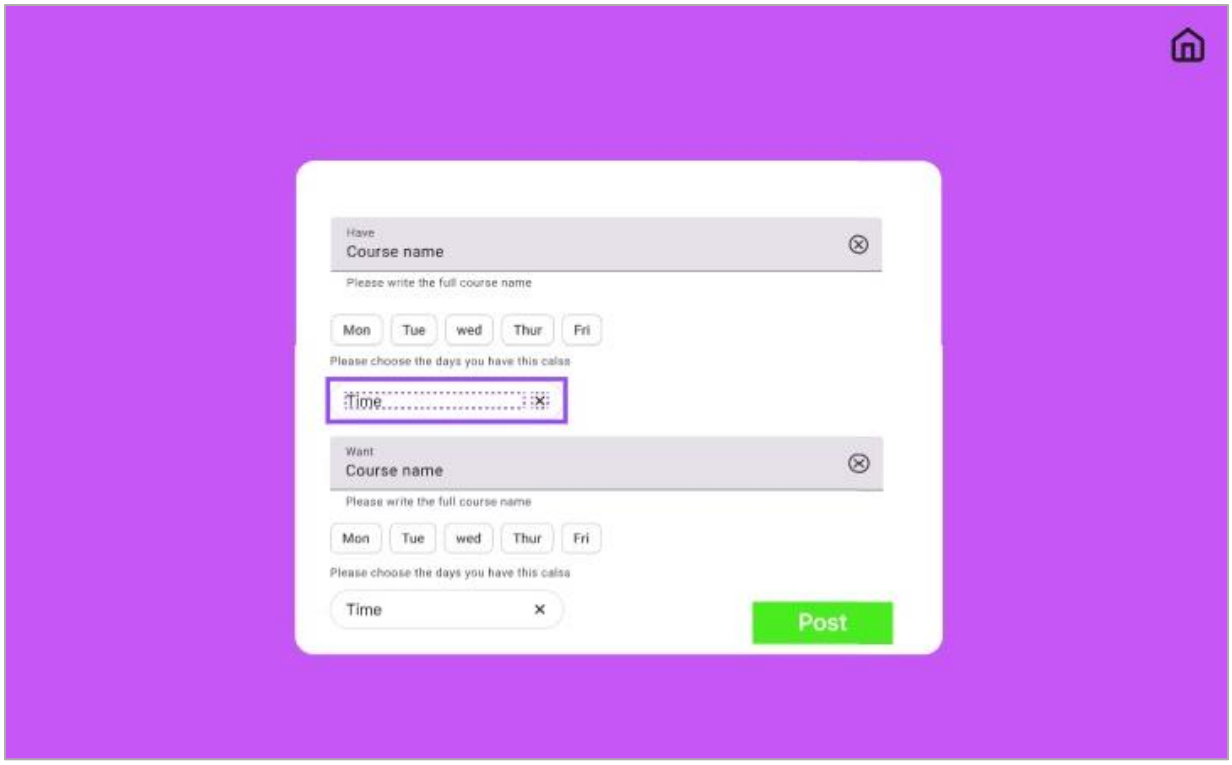
2. Main Catalogue



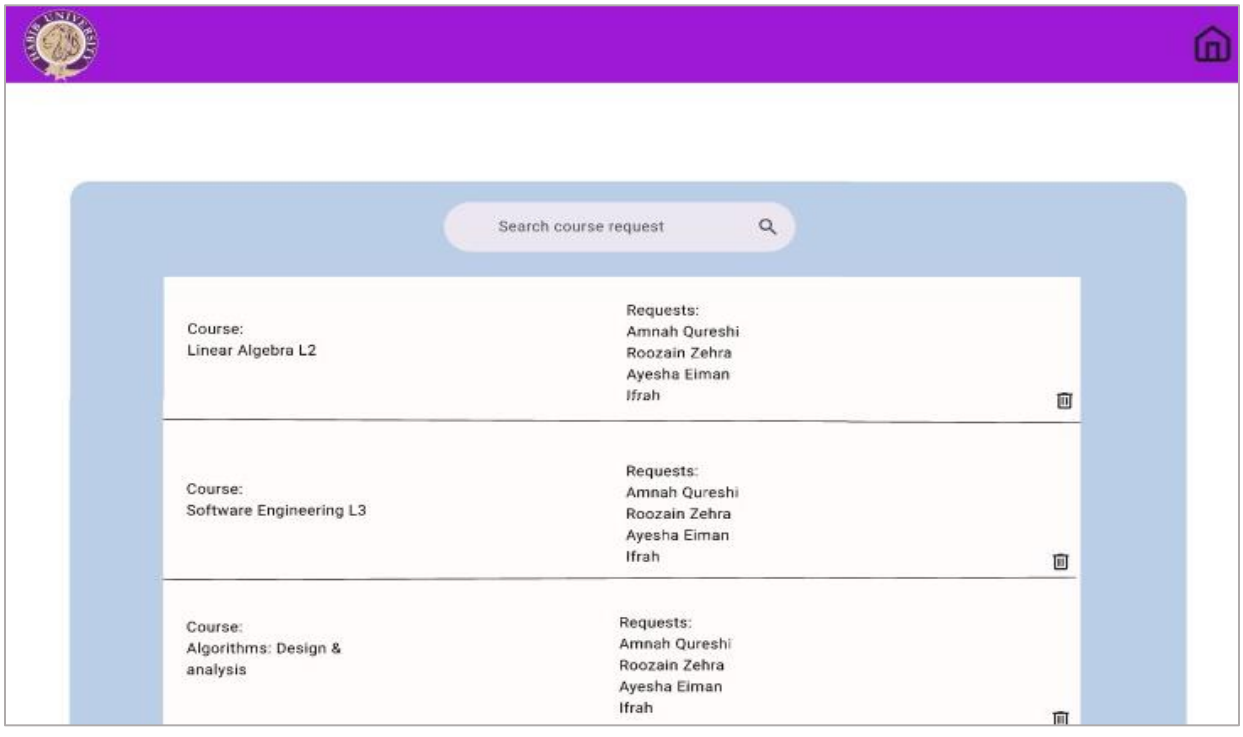
3. User Profile



4. Add Request



5. Your Request Catalogue



NON-FUNCTIONAL REQUIREMENTS:

1. Performance Requirements
    - The system should handle at least 100 concurrent users without significant lag.
    - Response time for search queries should be under 2 seconds.
    - The platform should be able to process 1000 swap requests per day efficiently.
2. Usability Requirements
    - The UI should be simple and intuitive for students with minimal technical knowledge.
    - The platform should be mobile-responsive, ensuring accessibility on both phones and desktops.
    - Users should be able to complete a swap request in under 5 clicks.
3. Security Requirements
    - Only authenticated students should access the platform using university credentials.
    - All chat messages and personal data should be encrypted.
    - The platform should implement role-based access, ensuring students can only manage their own requests.
4. Availability Requirements
    - The system should have 99.5% uptime, allowing access at all times except for scheduled maintenance.
    - Scheduled maintenance should be notified 24 hours in advance.
5. Maintainability & Scalability Requirements
    - The system should be modular to allow easy updates and feature additions.
    - It should be scalable to support future integrations, such as automated course prerequisite checks.
6. Legal & Compliance Requirements
    - The platform should comply with university data policies, ensuring no student information is shared without consent.
    - All logs should be retained for at least 6 months for auditing purposes.

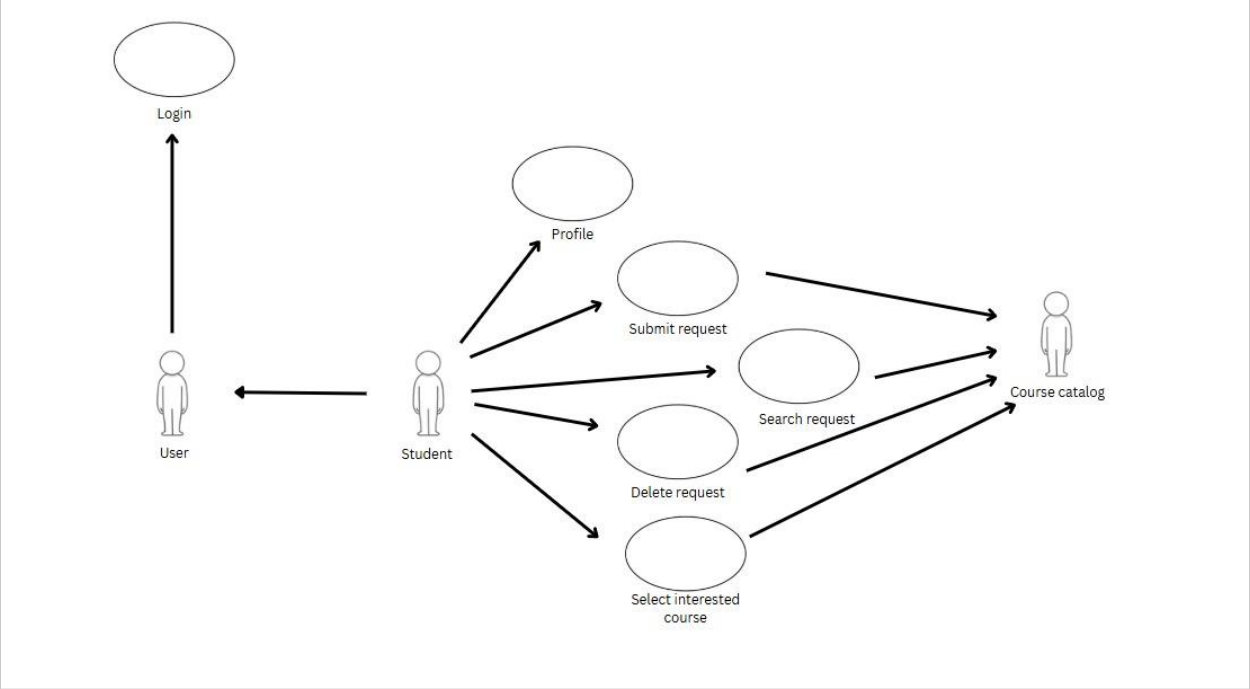
TECHNICAL CONSTRAINTS:

1. Real Time Processing for matching Algortihm
    - Would get very expensive if many users are submitting requests at the same timed.
    - The alternative is batch processing, which would result in a delay in receiving matches.
2. Complex Matching Queries
    - Matching swaps using multiple conditions (course code, time, elective type) may require multiple queries per request, which would be expensive.
3. Node.js

- Node.js runs on a single thread, making CPU-heavy operations (e.g., matching swaps for many users) slow.

4. Cloud Messaging
    - Firebase Cloud Messaging only sends push notifications, it does not support real-time messaging.
    - Solution could be to have students contact each other using third party apps (Teams, WhatsApp, etc.)

USER-FLOW DIAGRAM:



GIT-FLOW DIAGRAM:

Course-Swap-Portal/README.md at main · RoozainZehra/Course-Swap-Portal