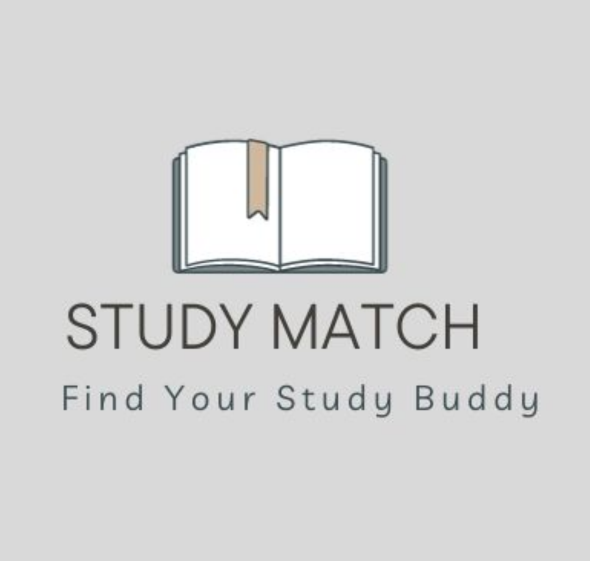
StudyMatch

Developers: Sidharth Eage, Elvis Sun,Vansh Reddy



What is StudyMatch?

- StudyMatch is an online platform to help students find study partners or sessions
- It aims to help students schedule sessions with other interested students
- It also allows students to chat with other student prior to scheduling a session
- StudyMatch offers a variety of topics or subjects that a student may be interested in
- Simple, but effective

Background

- LMS doesn’t offer options to chat with fellow classmates
- Classes that use Webex do, but students often don’t check them leading to reply times of over a few weeks
- Asking fellow classmates to study in class can be awkward
- Studying with a partner is much more effective than studying alone
- We wanted to create a platform that would be simple but effective for students to find study partners

Features

- Ease of scheduling time and location through platform
- Allow for quick in app messaging
- Our algorithm will match students with the most compatible partner through a scoring system
- Rating system to improve future match quality
- Better studying session organization and better study habits

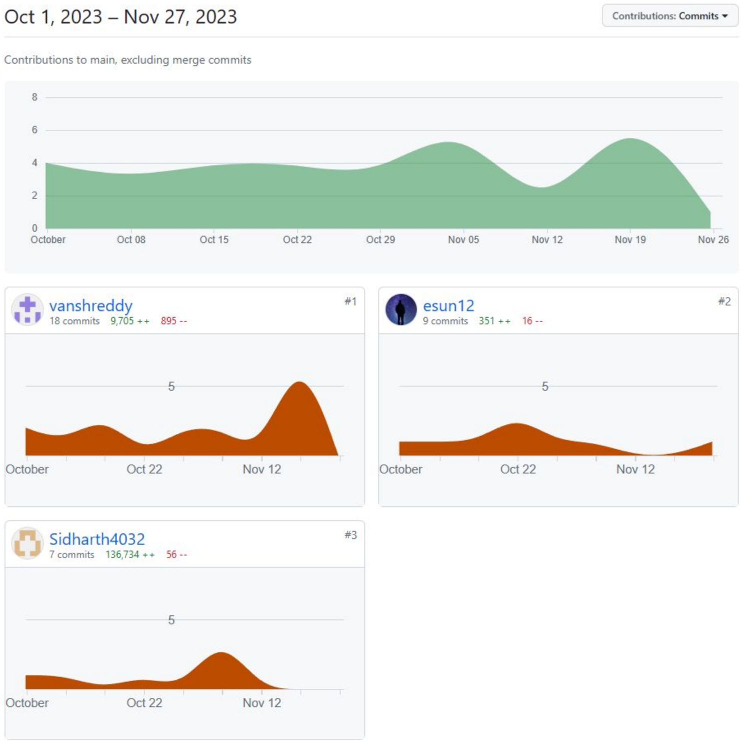
Project Structure

Hosted on Github

License: MIT License

We used Github to managed our codebase to push any new changes and additions.

- Multiple repos
- Protected main branch
- Separate branches for frontend and backend
- Test before merging
- Discord is our main communication method



Oct 1, 2023 – Nov 27, 2023

Contributions: Commits

Contributions to main, excluding merge commits

vanshreddy #1
18 commits 9,705 ++ 895 --

esun12 #2
9 commits 351 ++ 16 --

Sidharth4032 #3
7 commits 136,734 ++ 56 --

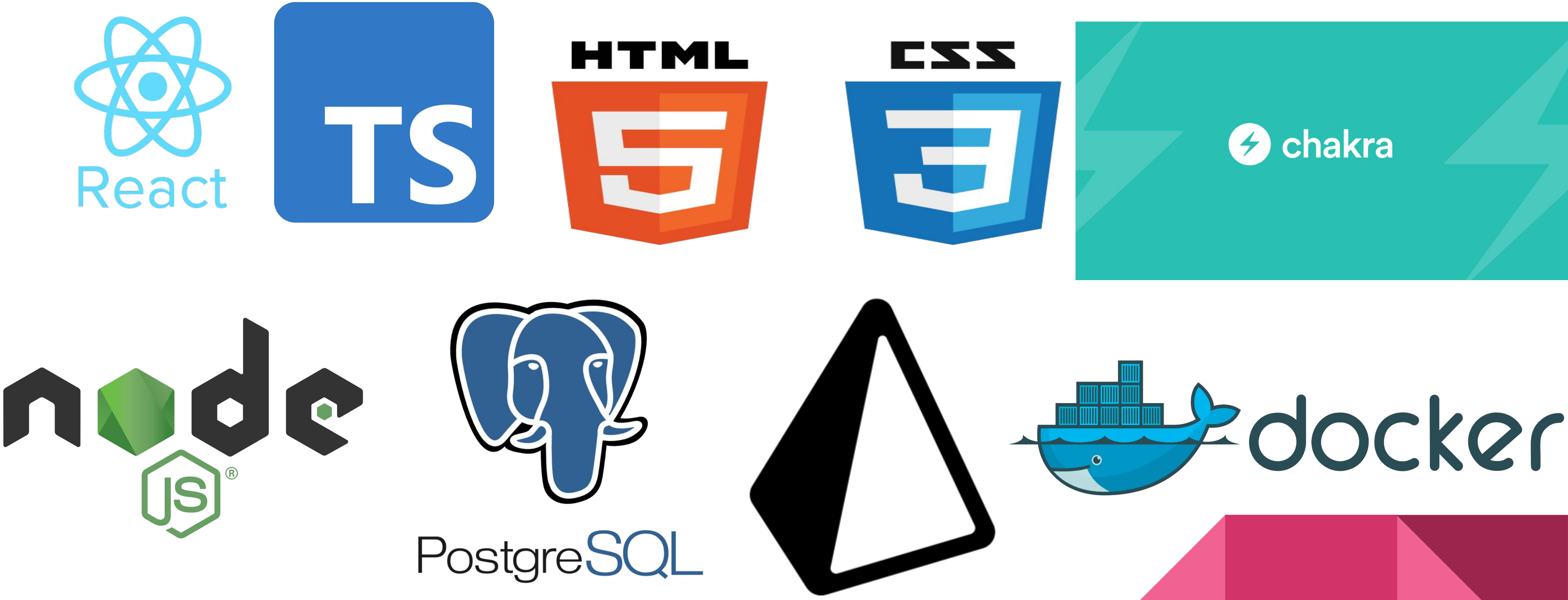
Progress Management

Since this project was started from scratch, we set up a more structured approach

1. Brainstorming & Ideas
2. Research & Planning
3. Initial Bootstrap & Programming
4. Launch & Test

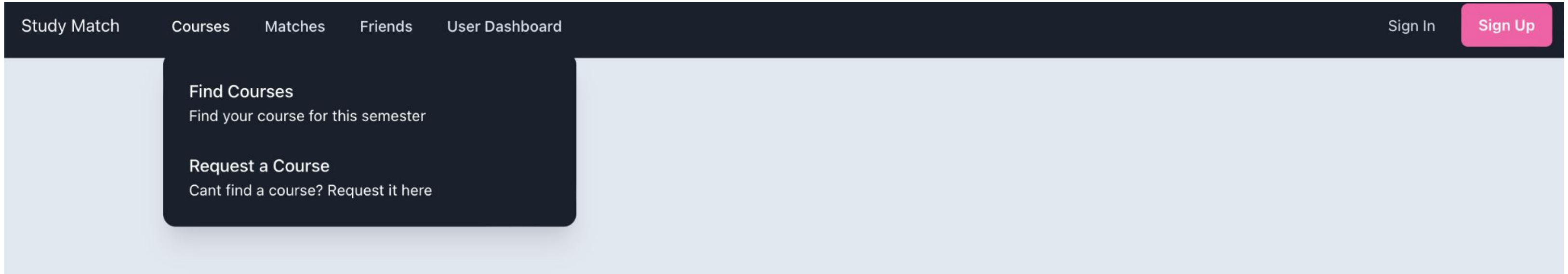
Currently, we are still in phase 3 of finishing up the platform. We still have some chunks of the frontend & backend to finish up before we can launch.

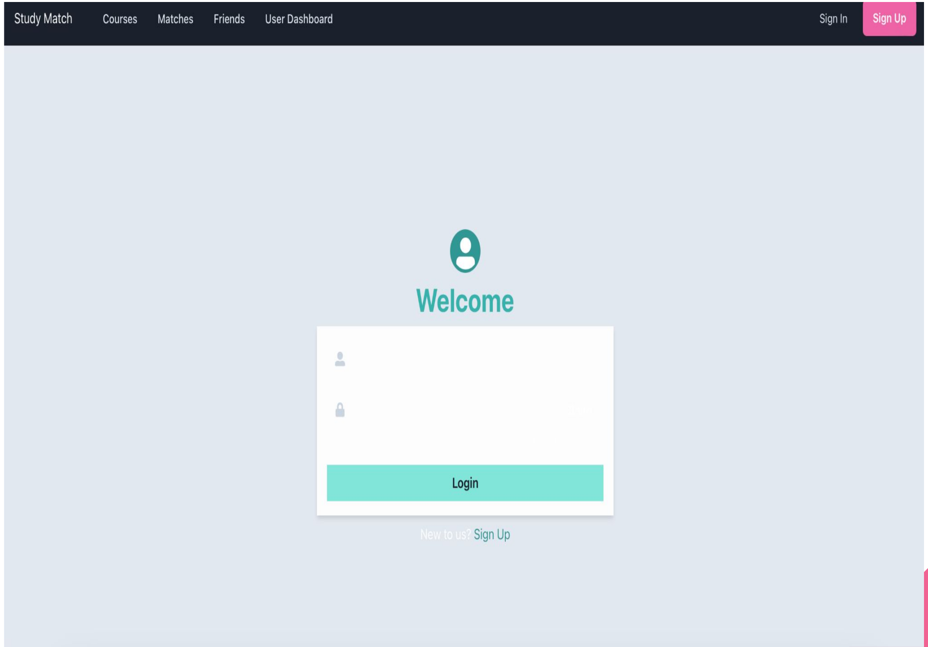
Technologies Used

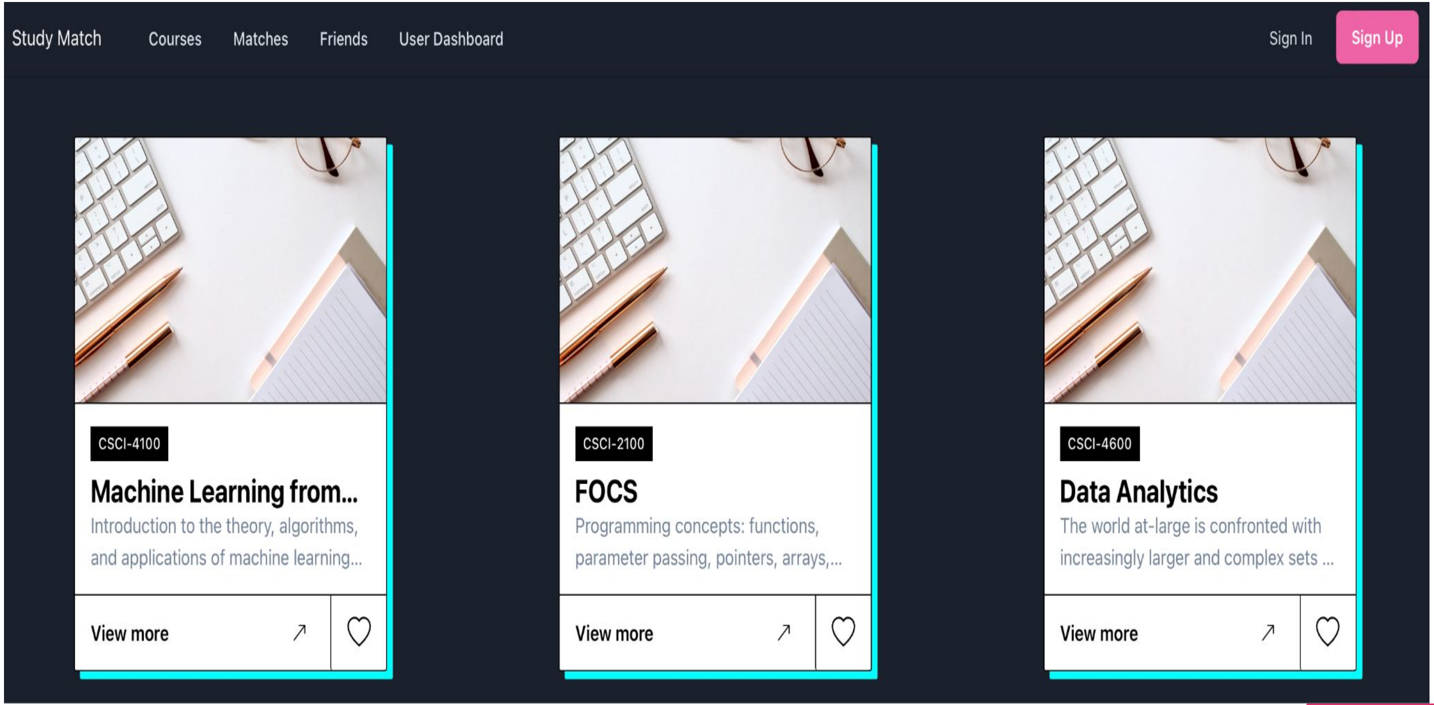


Frontend

- Designed the Frontend theme including light and dark mode using Chakra UI
- Created the login page and registration page tied with the backend and tested
- Designed and implemented a landing page
- Created the courses view page for the students to select courses.


Study Match
Courses
Matches
Friends
User Dashboard
Sign In
Sign Up
Find Courses
Find your course for this semester
Request a Course
Cant find a course? Request it here


Study Match
Courses
Matches
Friends
User Dashboard
Sign In
Sign Up
Welcome
Login
New to us? Sign Up



CSCI-4100

Machine Learning from...

Introduction to the theory, algorithms, and applications of machine learning...

CSCI-2100

FOCS

Programming concepts: functions, parameter passing, pointers, arrays,...

CSCI-4600

Data Analytics

The world at-large is confronted with increasingly larger and complex sets ...

User Registration/Login endpoints

- **Endpoint**: POST /register
- **Functionality**:
  - Users can create an account by providing a username and password.
- **Input Validation**:
  - Username: Must be at least 3 characters.
  - Password: Must be at least 5 characters.
  - Sanitization: Trims and escapes username to prevent SQL injection.
- **Error Handling**:
  - Checks if the username already exists in the database.
  - Returns appropriate error messages for invalid inputs or database errors.
- **Security**:
  - Passwords are securely hashed using bcrypt before storing in the database.

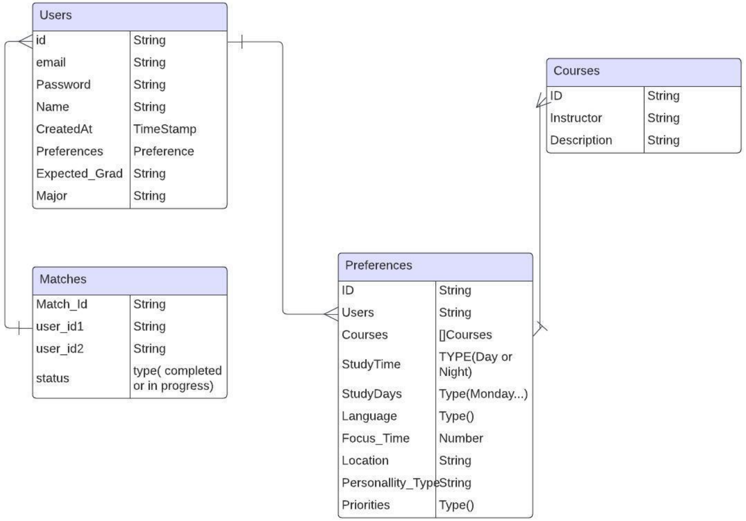
- **Endpoint**: POST /login
- **Functionality**:
  - Users log in using their username and password.
- **Input Validation**:
  - Username: Trimmed and escaped.
  - Password: Must be provided.
- **Authentication Process**:
  - Verifies username and password against the database.
  - If credentials are valid, generates a JWT token for session management.
- **Security**:
  - bcrypt is used to compare the stored hashed password with the entered password.
  - JWT token is used for maintaining user sessions securely.

Security and User Management

- Basic input sanitization is also implemented to prevent common web vulnerabilities such as SQL injection and cross-site scripting (XSS).
- Implementation of a modular code structure especially as the separation of concerns is maintained, with distinct functions for different tasks.
- We also make sure that once a password is created it is hashed using using bcrypts before being stored in the database making sure that even in the event of a data breach, the actual passwords remain protected
- We are making effective use of environment variables making sure that sensitive information like JWT secret keys are stored in environment variables, not in the codebase.

Database

- Created the DB schema
- Migrated the schema to Prisma ORM
- Preferences table is used for matchmaking
- Courses table is manually added by students
- Completed working on CRUD API for the tables



Users	
id	String
email	String
Password	String
Name	String
CreatedAt	TimeStamp
Preferences	Preference
Expected_Grad	String
Major	String

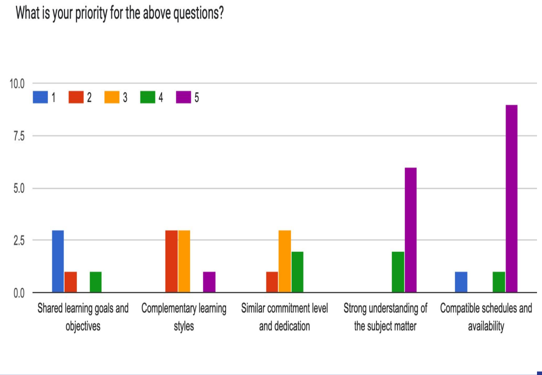
Matches	
Match_Id	String
user_id1	String
user_id2	String
status	type(completed or in progress)

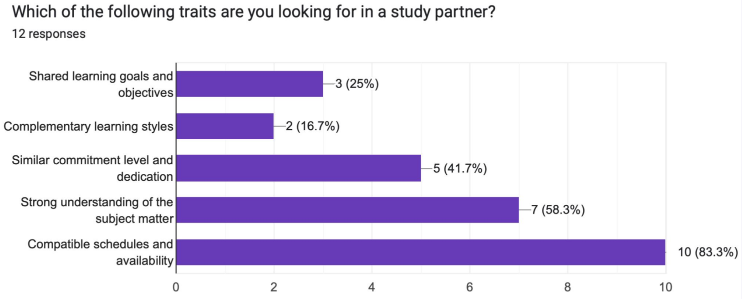
Courses	
ID	String
Instructor	String
Description	String

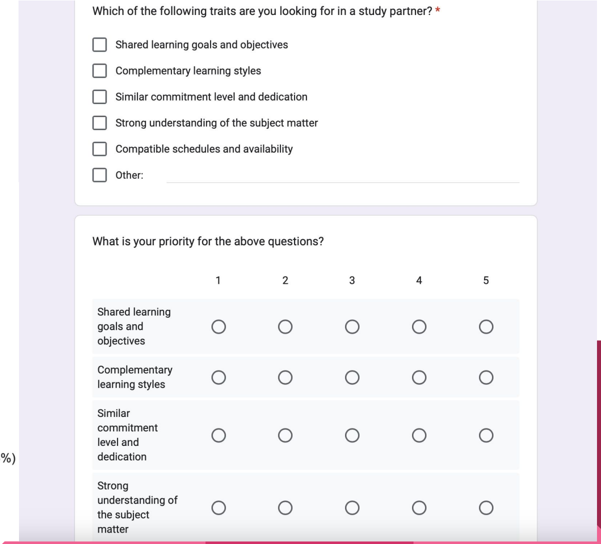
Preferences	
ID	String
Users	String
Courses	[]Courses
StudyTime	TYPE(Day or Night)
StudyDays	Type(Monday...)
Language	Type()
Focus_Time	Number
Location	String
Personallity_Type	String
Priorities	Type()

Questionnaire

- We prepared the following Questionnaire and asked our friends to complete it.
- We will try implementing the matching algorithm based on the results.
- Most of the people prefer a compatible schedule and also someone who understands the topics.
- The platform will also ask for preferences and priority during the registration process.







Which of the following traits are you looking for in a study partner? *

□ Shared learning goals and objectives
□ Complementary learning styles
□ Similar commitment level and dedication
□ Strong understanding of the subject matter
□ Compatible schedules and availability
□ Other:

What is your priority for the above questions?

	1	2	3	4	5
Shared learning goals and objectives	○	○	○	○	○
Complementary learning styles	○	○	○	○	○
Similar commitment level and dedication	○	○	○	○	○
Strong understanding of the subject matter	○	○	○	○	○

Algorithm

Phases:

- **Data Collection**:
  - Collect data about students such as their grade, subjects, availability, and proficiency levels
- **Matching Algorithm**:
  - We will weigh each student’s preferences and assign a score for each user
  - People with the closest scores will get matched
  - Ensure schedule & availability align
- **Rating System**:
  - The rating system will exist more so to verify and boost or lower a student’s proficiency level. However, this will be kept private as to not leak any sensitive information about student proficiency levels. After a session, students will be able to rate each other.

Plan for Future

- Some ideas for the future:
- Improve our front-end to be more visually appealing
- Add an option for learning styles such as visual, auditory, read/write, and kinaesthetic
- Start issue tracking and wiki documentation on how to use or setup when launched
- Integrate CI/CD patterns
- Add a scraper that will scrape the catalog and fill the courses in the DB
- Research hosting options

Thanks!