

Mauve.in

COP290 Lab Assignment 3 Report

KUSHAGRA GUPTA, PARTH PATEL, HARSH VORA, RAJAT GOLECHHA

April 30, 2023

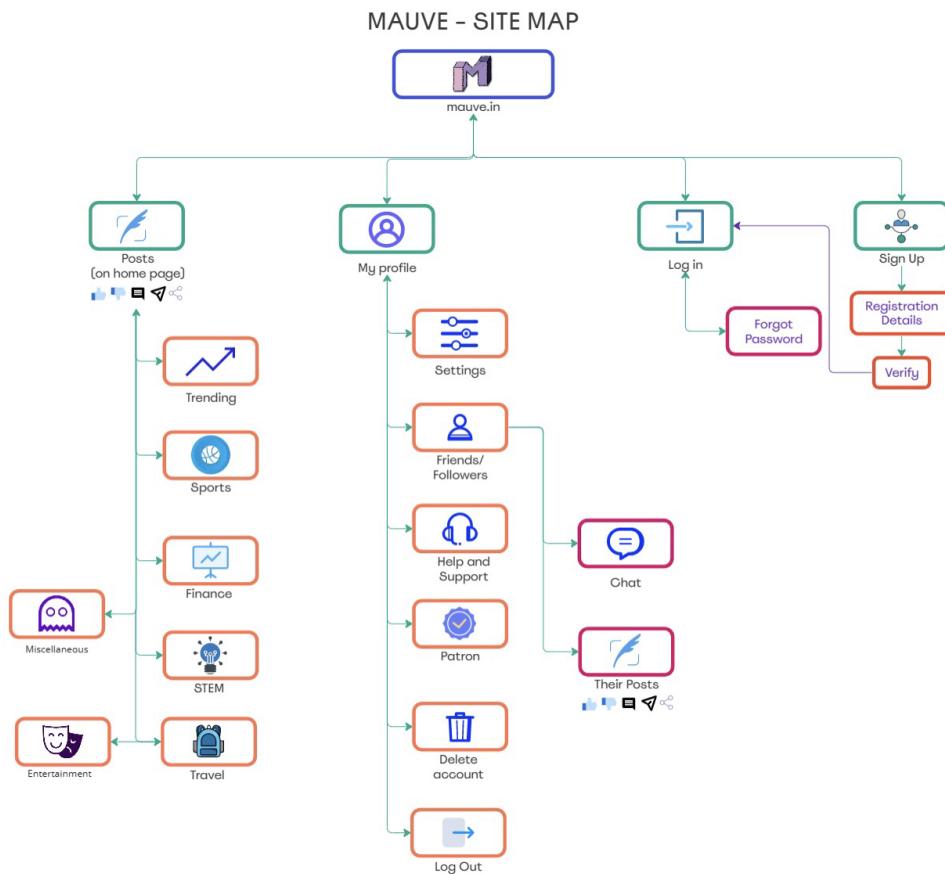
§1 Overview

Mauve, is a social blog posting forum, where members of the mauve global community can post about variety of topics like sports, entertainment, travel, tech & finance.

Users can discuss, chat with their friends, and comment on their posts.

Why the name? —— Mauve, refers to a pale shade of purple, around which our entire website is themed and built.

§2 Sitemap



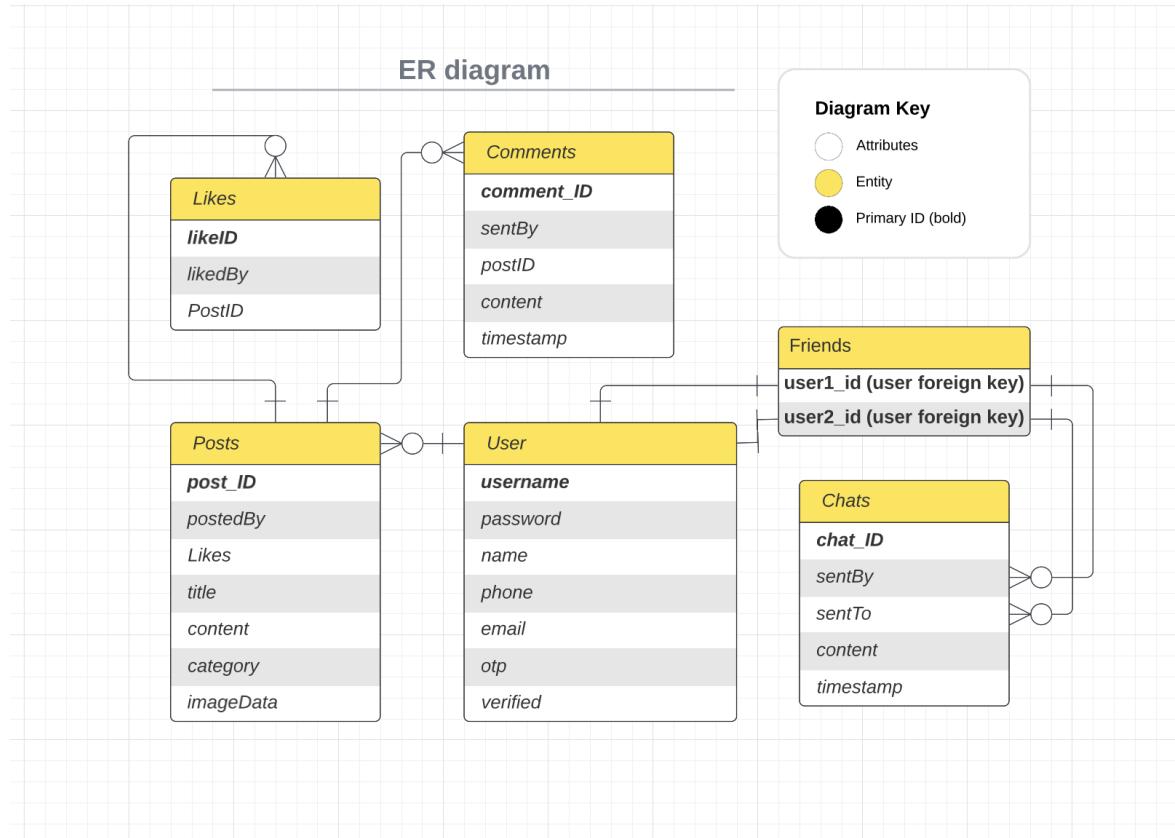
The purple arrow → represents automatic redirection to the destination from the source.
The double ended arrows ← ↔ represent that we can access both the destination and source from one another.
Every page will contain the access to the default home page - mauve.in, from where the navigation starts.

miro

§3 Baadal VM Website Link

Our website is up and hosted on Baadal
Link <http://10.17.6.13:5000/home>

§4 ER Diagram



§5 Github Link

[Github Website](#)

§6 API Descriptions

§6.1 Schema - YAML Links

[YAML file](#)

[YAML file converted to HTML](#)

[PDF generated by the YAML HTML](#) We have used GET and POST APIs in our app.py file, the API descriptions of which are in the YAML file.

§7 Design Choices

Filter/Sort : Our implementation sorts only on the basis of time uploaded.

Patrons: We have 4 different kinds of patrons, inspired by the name of houses in Harry Potter.

§8 CI/CD Test Coverage Report

Unit Test Report Integration Test Report

§9 Design Choices

§10 Figma Link

COP290 Figma Link

§11 Documentation Links

§11.1 Week 1 - Documentation

Week 1

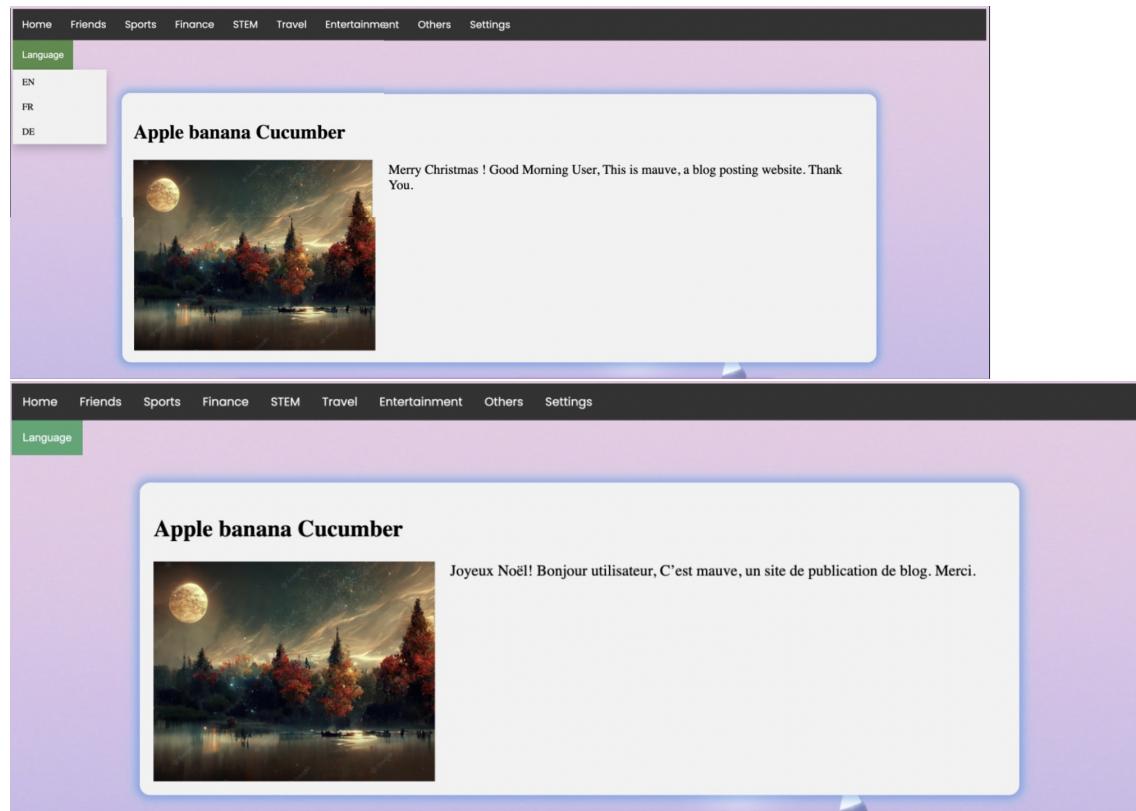
§11.2 Week 3 & 6

Week 3 & 6

§11.3 Presentation

Presentation

§12 ML API



We implemented the text translate, ML API by Microsoft Azure.

As you might have noticed that the posts when opened after read more have a dropdown

menu by the name of language.

Clicking on the desired language, translates the post into it.

The Post written in English, is translated by the API call into French.

§13 Unique features

§13.1 Responsiveness

Our website is compatible with all three types of devices - Personal Computer, Mobile (Handheld) and Tablets.

We have used media queries in CSS or HTML script to do so. Here's a sample code for the same

```

1      background image: url("1")
2  @media only screen and (max-width: 865px) {
3      background image: url("2")
4  }
5  @media (min-width: 865px) {
6      background image: url("2")
7  }
```

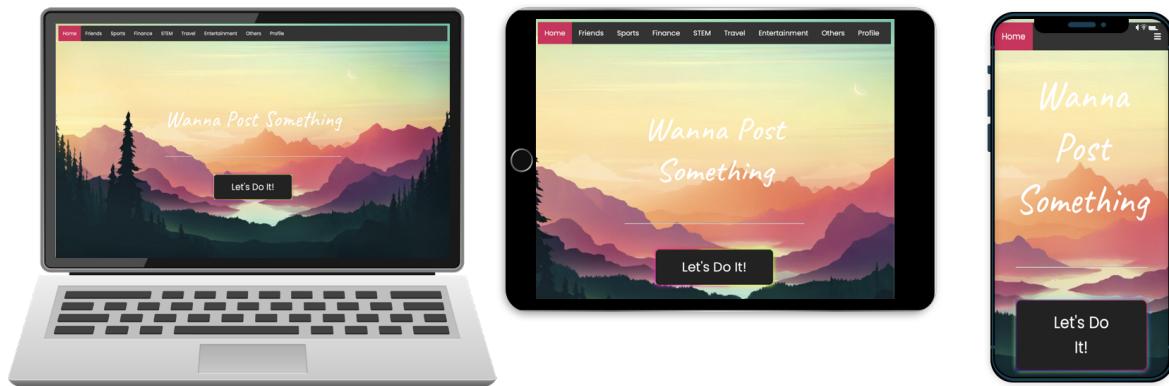


Figure 1. Home Page

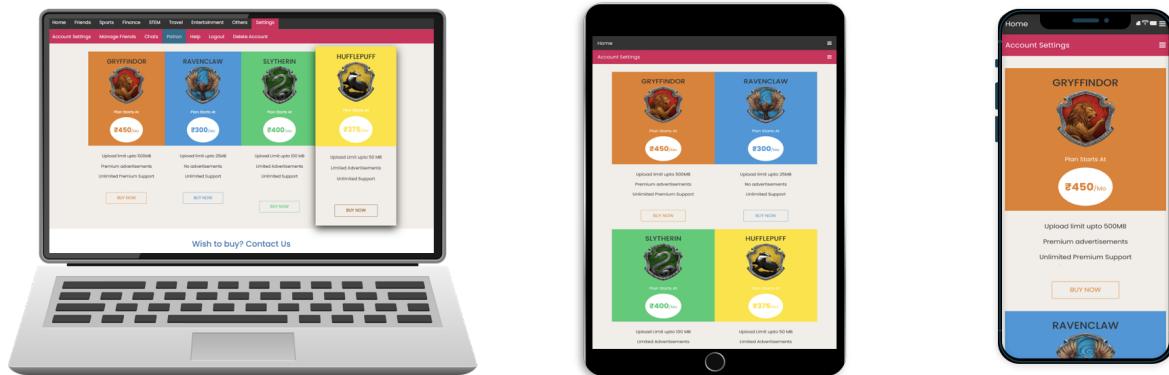


Figure 2. Patron Page

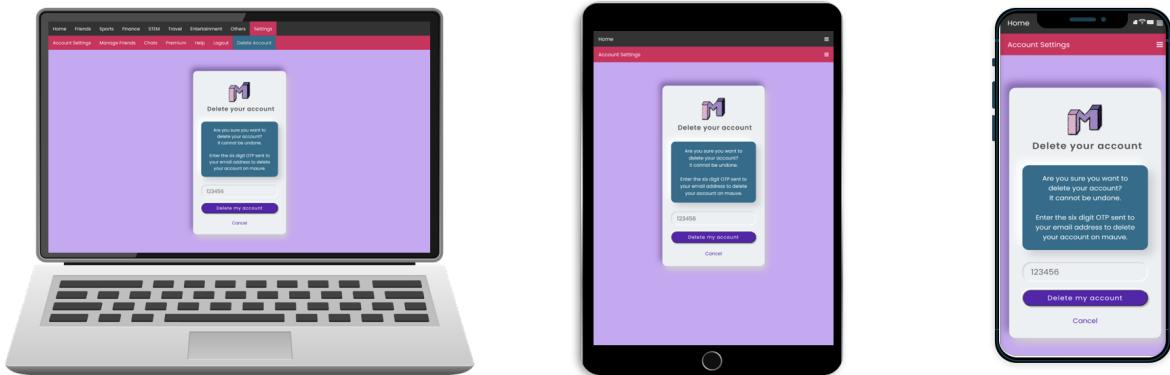


Figure 3. Delete Account

§13.2 Likes and Comments

We implemented the feature to like/unlike posts(as in youtube) and also add comments to the posts.

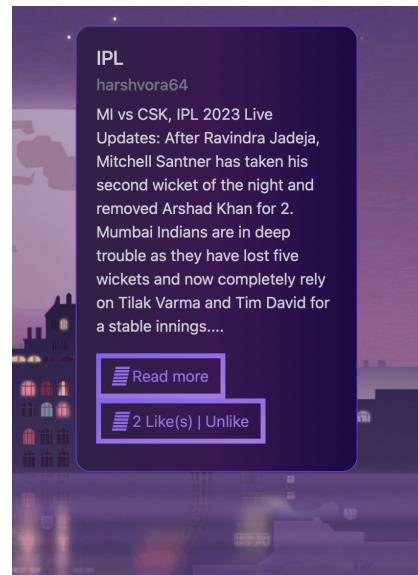


Figure 4. Likes

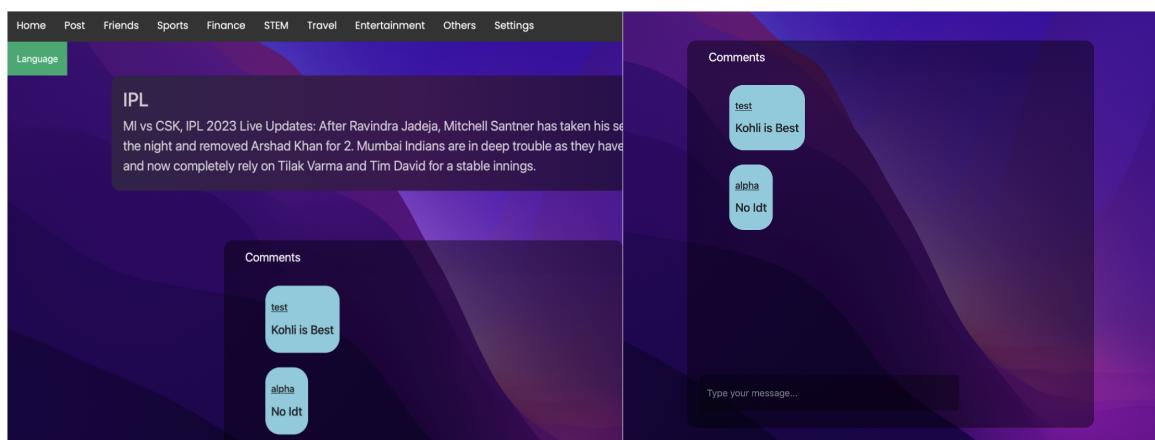


Figure 5. comments

§13.3 Chats

We also implemented the chat feature for users to chat with one another using our platform, you can easily add friends with our manage friends features and then chat with them with just one click. You can even make new friends with the people who post certain interesting articles via their usernames.

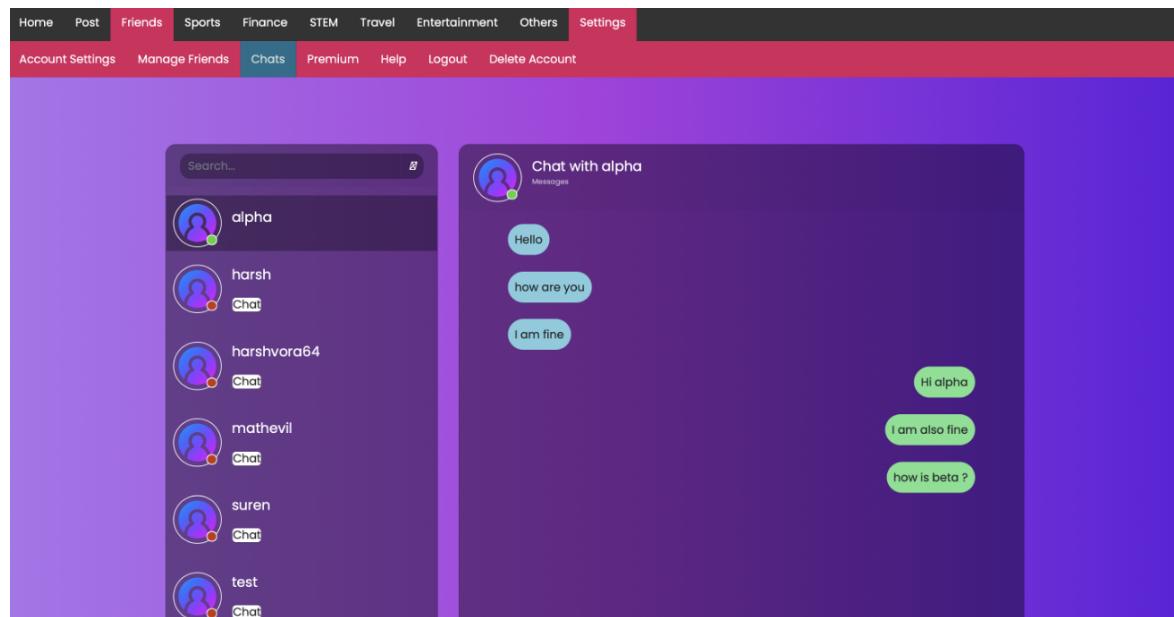


Figure 6. chats

§13.4 Patrons

We allow people to become patrons and give money to us so that we launch their posts and ads from our side and also give likes to those posts so that their credibility and reach increases (just like twitter's blue tick).

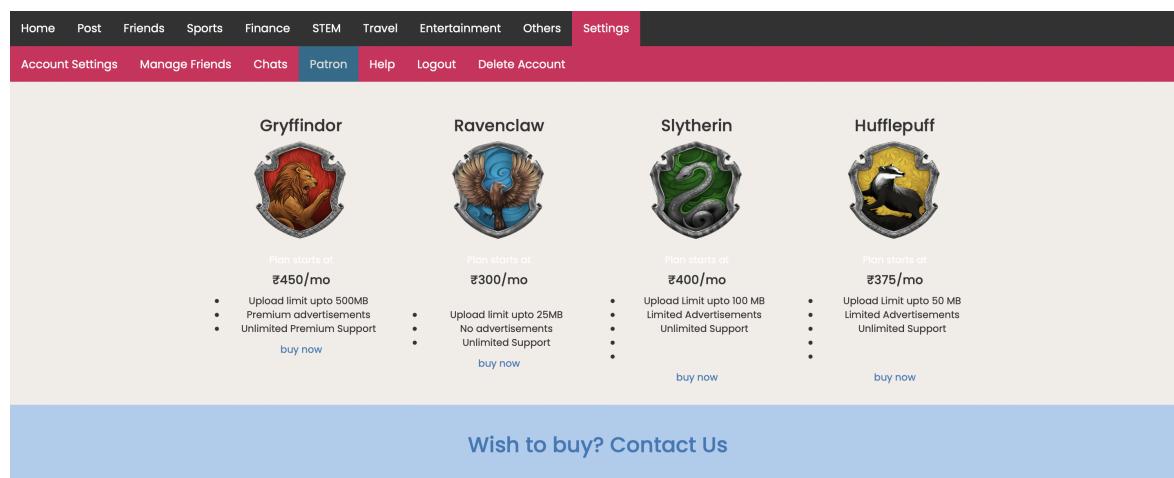


Figure 7. Patrons

§13.5 OTP

We have also implemented the feature for otp based verification when a person signs up for the first time. The OTP is randomly generated number using python and the person receives it on his/her registered email id.

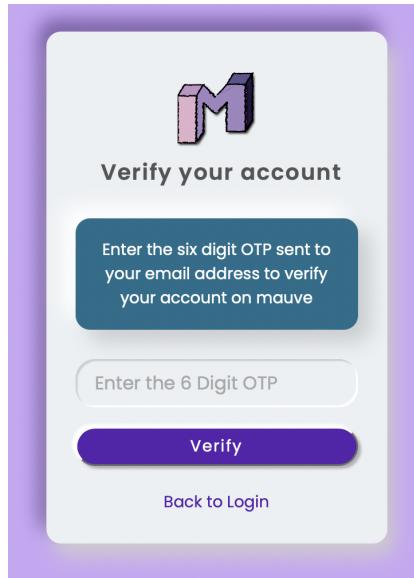


Figure 8. OTP



Figure 9. Gmail

§14 Challenges

- Finding a suitable external API : For securing an API almost everywhere on internet, credit card information is required, very few good open source api are available that can be used and integrated.
- Working with Baadal VM : The experience with Baadal was very bad, we had so many issues connecting with baadal, understanding how to use it. How to run it, how to upload and what not. The cloud server's working itself was a challenge in itself.

§15 Limitations / Future Work

- Chats : Currently our chat is not live , and the same can be improved upon using sockets, to build an interface that works live for users chatting on both end in real time.

- Filter/Sort : Our current implementation sorts only on the basis of time uploaded, but the same can be improved using functions and python libraries to make different sorting filters.
- Advertisements : The current patrons system works like an amalgamation of instagram sponsored and twitter blue tick, which only provides validation, but integrating the same with sorting and filters can allow us to also boost the sponsored ads as in Google Ads.
- Communities : We have some sections like Sports , Entertainment , STEM , etc. but these are all fixed, and no new community/section can be introduced this can be changed by updating the entire mysql database to make room for new sections and communities and subcommunities within them.

§16 Database Architecture

This is our database architecture tables.

Tables_in_mauve					
<hr/>					
chats					
comments					
friends					
likes					
posts					
users					
<hr/>					

Figure 10. Main tables

Field	Type	Null	Key	Default	Extra
chat_ID	int unsigned	NO	PRI	NULL	auto_increment
sentBy	varchar(100)	YES		NULL	
sentTo	varchar(100)	YES		NULL	
content	varchar(1000)	YES		NULL	
timestamp	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

Figure 11. Chats

Field	Type	Null	Key	Default	Extra
comment_ID	int unsigned	NO	PRI	NULL	auto_increment
sentBy	varchar(100)	YES		NULL	
postID	varchar(100)	YES		NULL	
content	varchar(1000)	YES		NULL	
timestamp	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

Figure 12. Comments

Field	Type	Null	Key	Default	Extra
likeID	int unsigned	NO	PRI	NULL	auto_increment
likedBy	varchar(100)	YES		NULL	
postID	varchar(100)	YES		NULL	

Figure 13. Likes

Field	Type	Null	Key	Default	Extra
user1_id	varchar(100)	NO	PRI	NULL	
user2_id	varchar(100)	NO	PRI	NULL	

Figure 14. Friends

Field	Type	Null	Key	Default	Extra
post_ID	int unsigned	NO	PRI	NULL	auto_increment
postedBy	varchar(100)	YES		NULL	
likes	int	YES		NULL	
title	varchar(100)	YES		NULL	
content	varchar(1000)	YES		NULL	
category	varchar(100)	YES		NULL	
imageData	varchar(200)	YES		NULL	

Figure 15. Posts

Field	Type	Null	Key	Default	Extra
username	varchar(100)	NO	PRI	NULL	
password	varchar(100)	YES		NULL	
name	varchar(100)	YES		NULL	
phone	varchar(100)	YES		NULL	
email	varchar(100)	YES		NULL	
otp	varchar(100)	YES		NULL	
verified	varchar(1)	YES		NULL	

Figure 16. Users

§17 Experience with LLMs

We extensively used chatGPT for motivating all the backend part. Be it strange errors or syntax for making an API or to check if some feature is supported in MySQL or not. Our architecture carries the potential to use a ChatGPT API along with the ML API, which could be used to **paraphrase, summarize the text, or correct grammatical errors** as they work along the same schemas.

For mysql part, sometimes we forgot the syntax such as describe tables, select * from table, etc. ChatGPT really helped us in here.

We managed to host our website up on baadal during the last checkpoint submission itself when other groups were still struggling to do. All this from ssh into the server to install libraries as well as killing ports through npx and destroying processes by PID was done with the help of chatGPT.

A surprising incident : We wished to render the posts in website and we wanted a for loop so that it can iterate through all posts in a database. Now, we had some intuition for this and when we asked chatGPT to describe how to render posts, it gave us the flask python and html codes. This reduced our efforts greatly. Having a tea-time discussion with another team, we got to know they spent 3 hours figuring out on google on how to insert such a for loop.

§18 Unit & Integration Test Coverage

app.py	258	98	62%	14, 74–93, 134–140, 156–162, 178–184, 200–206, 222–228, 244–250, 270–277, 282, 422–424, 443, 515, 520–524, 531–535, 566–576, 580–585, 589, 593, 597, 626
test.py	118	11	91%	74, 94, 132–134, 137–139, 144–153
TOTAL	376	189	71%	

§19 References

Here are the Kaggle links used for our datasets for database:

Data set used for Technology : <https://www.kaggle.com/datasets/thibalbo/techcrunch-posts-compilation?resource=download>

Data set used for entertainment : <https://www.kaggle.com/datasets/thedevastator/unlocking-movie-engagements-and-insights-through>

Data set used for travel :<https://www.kaggle.com/datasets/thedevastator/uncovering-travel-experiences-desires-and-opinion>

Data set used for sports :<https://www.kaggle.com/datasets/gpreda/euro-2020-on-reddit>

§20 Token Distribution

	Team Members			
Entries	Parth Patel	Harsh Vora	Kushagra Gupta	Rajat Golechha
Tokens	10	10	10	10
Contribution	Back-end	Back-end	Front-end	Front-end

Table 1 Token Distribution