CAMPUS HIVE

A PROJECT REPORT

BY **TEAM NO. 3**

TEAM MEMBER 1 (E23CSEU0361)
TEAM MEMBER 2 (E23CSEU2426)

SUBMITTED TO

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND TECHNOLOGY, BENNETT UNIVERSITY

GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

April 2025

DECLARATION

I/We hereby declare that the work which is being presented in the report entitled “Campus Hive”, is an authentic record of my/our own work carried out during the period from JAN, 2025 to April, 2025 at School of Computer Science and Engineering and Technology, Bennett University Greater Noida.

The matters and the results presented in this report has not been submitted by me/us for the award of any other degree elsewhere.

Signature of Candidate
(GEETESH DALAL)

GEETESH DALAL
(Enroll. No. E23CSEU0361)

VINAYAK SINGH
(Enroll. No. E23CSEU2426)

ACKNOWLEDGEMENT

I/We would like to take this opportunity to express my/our deepest gratitude to my/our mentor, **Dr. Sanchali Das** for guiding, supporting, and helping me/us in every possible way. I/we was/were extremely fortunate to have him as my/our mentor as he provided insightful solutions to problems faced by me/us thus contributing immensely towards the completion of this capstone project. I/We

would also like to express my/our deepest gratitude to VC, DEAN, HOD, faculty members and friends who helped me/us in successful completion of this capstone project.

Signature of Candidate
(GEETESH DALAL)

GEETESH DALAL
(Enroll. No. E23CSEU0361)

VINAYAK SINGH
(Enroll. No. E23CSEU2426)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Communication between college peers is a major problem which can vary for different domains whether it be just regular socializing, finding partners for projects, finding mates for carpooling to reduce the travelling expenses, renting out vehicles for an outing or just finding or reporting lost/found items. Communication system is just not so well developed within universities which led to the developing of Campus Hive – a web-based application which houses real time chat features to connect people to solve all the above problems all at once at a single place.

The project came into view when we surveyed different college students about their travelling expenses, their lost items which they never found again, hardships in finding collaborators for projects. That's what gave the idea of developing Campus Hive which would solve all of the problems by making the communication way easier than before.

The solution was created using React.js, Node.js, Express.js, MongoDB and Socket.IO. Users can have public discussions, private chats, add carpool requests, post lost/found items, post car rentals, and post project collaboration requests creating a seamless connectivity between university students.

1. INTRODUCTION

Campus Hive is a student-focused platform designed to enhance campus life by connecting university students with features like real-time chat, carpool requests, lost and found posts, and project collaborations, it provides a seamless way for students to interact, share resources, and stay connected. Campus Hive simplifies communication, making university life more collaborative, efficient and easier.

1.1. Problem Statement

University students often face challenges in campus communication, including finding carpools, reporting lost items, connecting with people, and collaborating on projects. There is a lack of a unified platform that streamlines these interactions in a secure and user-friendly manner. Campus Hive aims to solve this by providing an all-in-one web-based solution meant to cater to student needs.

2. BACKGROUND RESEARCH

University students often struggle with disconnected communication and resource-sharing on campus. There's a need for a platform that simplifies interactions like carpooling, lost and found items info, and academic collaboration. Campus Hive solves this by offering a centralized place for all the students to overcome these problems faced by them in daily life.

2.1 Proposed System

We propose a web application that connects university students through features like real-time chat, carpooling requests, and project collaborations. Designed for simplicity and usability, it would offer better connectivity and resource sharing within the University campus.

2.2 Goals and Objectives

Table 1: Goal and Objectives

#	Goal or Objective
1	Make the system extensible – future updates that can be done easily
2	Make the system easy to support – provide good documentation, configuration/build files, administrator's manual
3	Make the system very easy to use – users would agree that minimal to no training is needed
4	Build a prototype that demonstrates the user interface to get early feedback from the customer/users
5	Have fun working on the project

3. PROJECT PLANNING

This section covers the details of the project planning. Selecting the lifecycle of the development, project stakeholders, resources required, assumptions made (if any) are detailed in the sections below.

3.1 Project Lifecycle

We used an Agile lifecycle of multiple design and test iterations. Each sprint saw planning, prototyping, feedback, and iterative refinements.

3.2 Project Setup

Table 2: Project Setup

#	Decision Description
1	MERN (React.js, Node.js, Express.js, MongoDB) stack
2	GitHub for source control
3	Flowcharts and diagrams to visualize

3.3 Stakeholders

Table 3: Stakeholders

Stakeholder	Role
Students	Primary consumers
University mentors	Project critics
Geetesh Dalal	Team member
Vinayak Singh	Team member

3.4 Project Resources

Table 4: Project Resources

Resource	Resource Description	Quantity
Database Server	A database server provided by the sponsoring company.	1
Capstone Team	Our team of students who will be the primary developers of the project.	4
Jim Somebody	The mentor who will be able to provide us with technical assistance.	1
Mac Workstation	An OS X workstation with X Code for developing the OS X version of the software.	1
Android Phone	An Android phone to be used as test hardware for the mobile version of the software.	2

3.5 Assumptions

Table 5: Assumptions

#	Assumption
A1	The capstone team and mentors will be able to meet face to face once a week.
A2	Users possess basic internet ability
A3	College data will be static in prototype
A4	Team will have sufficient time to complete a working model to present by mid-semester
A5	Recommendations are suggestive rather than prescriptive
A6	The development test data provided will be sufficient to create an accurate prediction of user actions
A7	The models developed will be easily extended to other forms within the time frame

4. PROJECT TRACKING

4.1 Tracking

Table 6: Project Tracking

Information	Description
Code Storage	Source control through GitHub
Bug Tracking	Bug tracking will be done with Trac.
Project Documents and Assignments	Weekly reports, specification and design documents, etc.
Continuous Integration	Continuous integration will be done.

4.2 Communication Plan

Table 7: Regularly Scheduled Meetings

Meeting Type	Frequency/Schedule	Who Attends
Conference Call/Skype	Weekly	Project team and mentor
Team Meeting	Weekly	Project team
Short Meeting	Weekly in class	Project team
Sprint Planning Meeting	Start of each sprint	Project team and mentor
Sprint Retrospective Meeting	End of each sprint	Project team
Sprint Review Meeting	End of each sprint	Project team, ***mentor, and sponsor***

Table 8: Information To Be Shared Within Our Group

Who?	What Information?	When?	How?
Project team	Task assignments & General scrum information	Weekly	Team meetings, listing in Project Specification.

Table 9: Information To Be Provided To Other Groups

Who?	What Information?	When?	How?
Sponsor and mentor	Final deliverables	At completion of project	Project specification doc., code, Power Point presentation
Sponsor and mentor	Weekly report	Weekly	Email and Trac site access
Sponsor and mentor	Project baselines ***(optional)***	At the end of each sprint	Onsite customer demo, access to repository

Table 10: Information Needed From Other Groups

Who?	What Information?	When?	How?
Sponsor and mentor	Requirement changes	Start of each sprint	Conference call or meeting with sponsor and mentor.

4.3 Deliverables

Table 11: Deliverables

#	**Deliverable**
1	Study results ***(if any)***
2	Code
3	Test and test results
4	Build process documents ***(if any)***
5	Install process documents ***(if any)***
6	Administrator or user manual ***(if any)***
7	Postmortem document
8	Final report (final PowerPoint presentation, 3 minute video, and final sprint)

5. SYSTEM ANALYSIS AND DESIGN

This section describes in detail about design part of Campus Hive.

5.1 Overall Description

This project aims to develop a web-based platform Campus Hive to improve communication and collaboration within university campuses. The system focuses on enhancing student interactions through features like public and private chats, carpool requests, lost & found posts, and project collaborations. Developed with a student-first mindset, the platform was shaped through user feedback, ensuring it to be a students' problems solver.

The technical foundation is built using the MERN stack (React.js, Node.js, Express.js, MongoDB), enabling real-time functionality through Socket.IO for chats and structured REST API for data exchange. The interface is designed to be clean and responsive, supporting file uploads, user authentication, and easy navigations for the user. Design flows and user journeys were mapped using flowcharts and wireframes created with Figma to guide frontend structure and backend logic.

Future developments of the system may involve integrating dynamic college databases via APIs, using machine learning models for more intelligent and adaptive recommendations, and adding support to mobile platforms. The simplicity of the existing prototype renders it extremely extensible and provides a good basis for further development according to contemporary ed-tech trends.

5.2 Users and Roles

Table 12: User and Roles

User	Description
Student User	The primary end-user who uses the platform to connect with other students, post or view carpooling and lost & found updates, post or view rental cars and collaborate on different projects posted by other users.
System (Agent)	The backend logic and frontend interface that will handle all the user actions such as messaging, posting, and fetching the real time data.
Developer	Team members who are responsible for designing, coding, and maintaining the platform, including implementing all the features and ensuring responsiveness of the platform over different devices.
Mentor/Evaluator	Guide or instructor offering critique, checking on progress, and determining the system's functionality and usability.

5.3 Design diagrams/ UML diagrams/ Flow Charts

5.3.1 Use Case Diagram

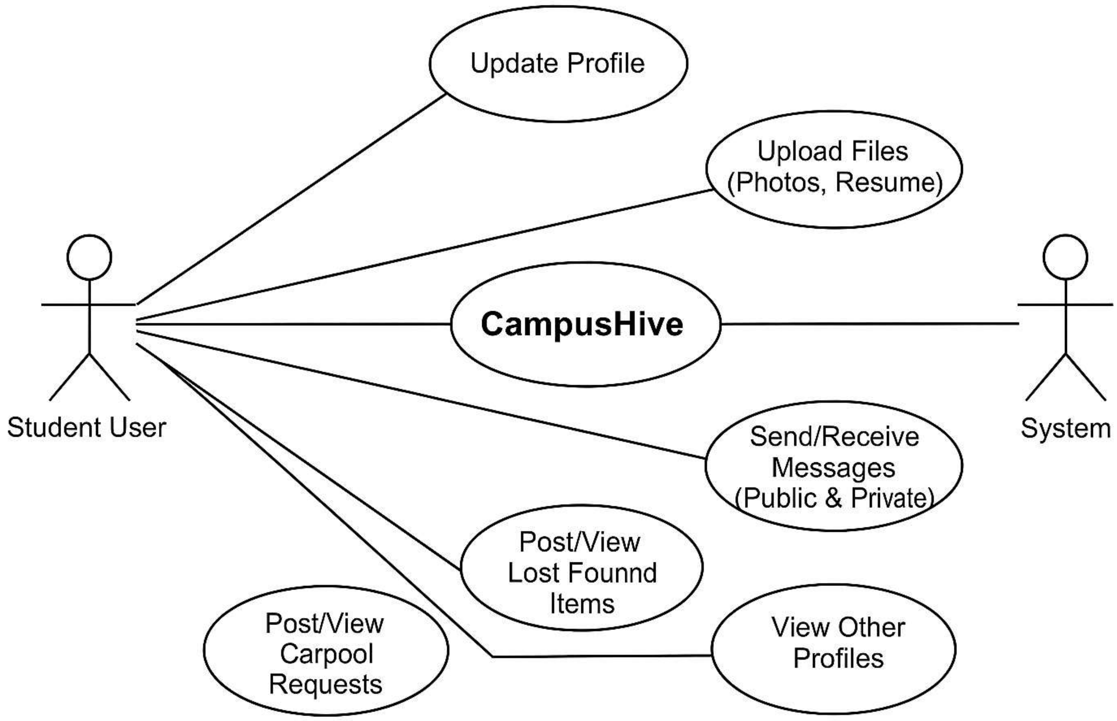

Update Profile
Upload Files (Photos, Resume)
Student User
CampusHive
System
Send/Receive Messages (Public & Private)
Post/View Lost Founnd Items
Post/View Carpool Requests
View Other Profiles

Figure 1: Use-case diagram

5.3.4 Class Diagram

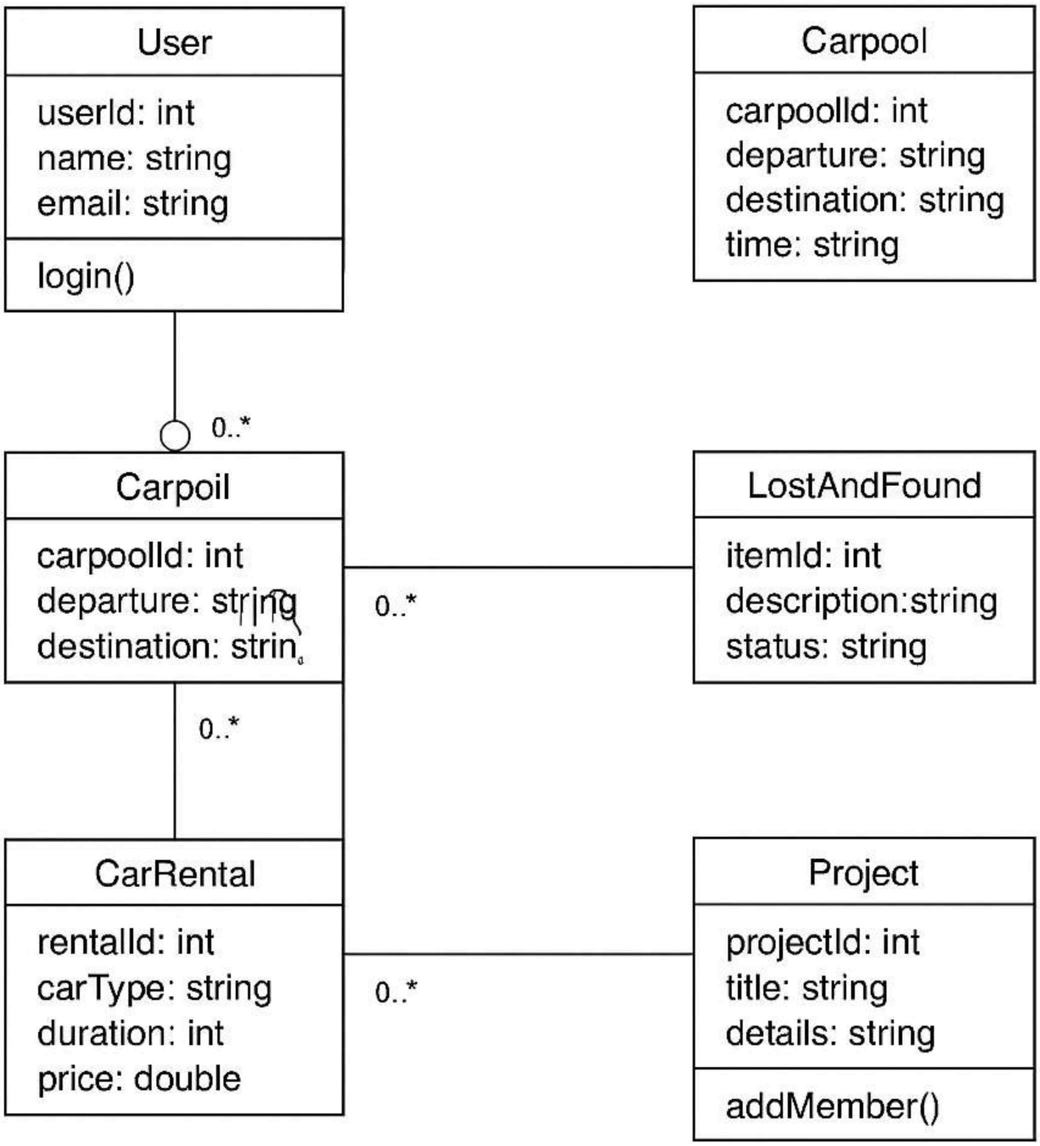


Figure 2: Class diagram

5.3.5 Activity Diagrams

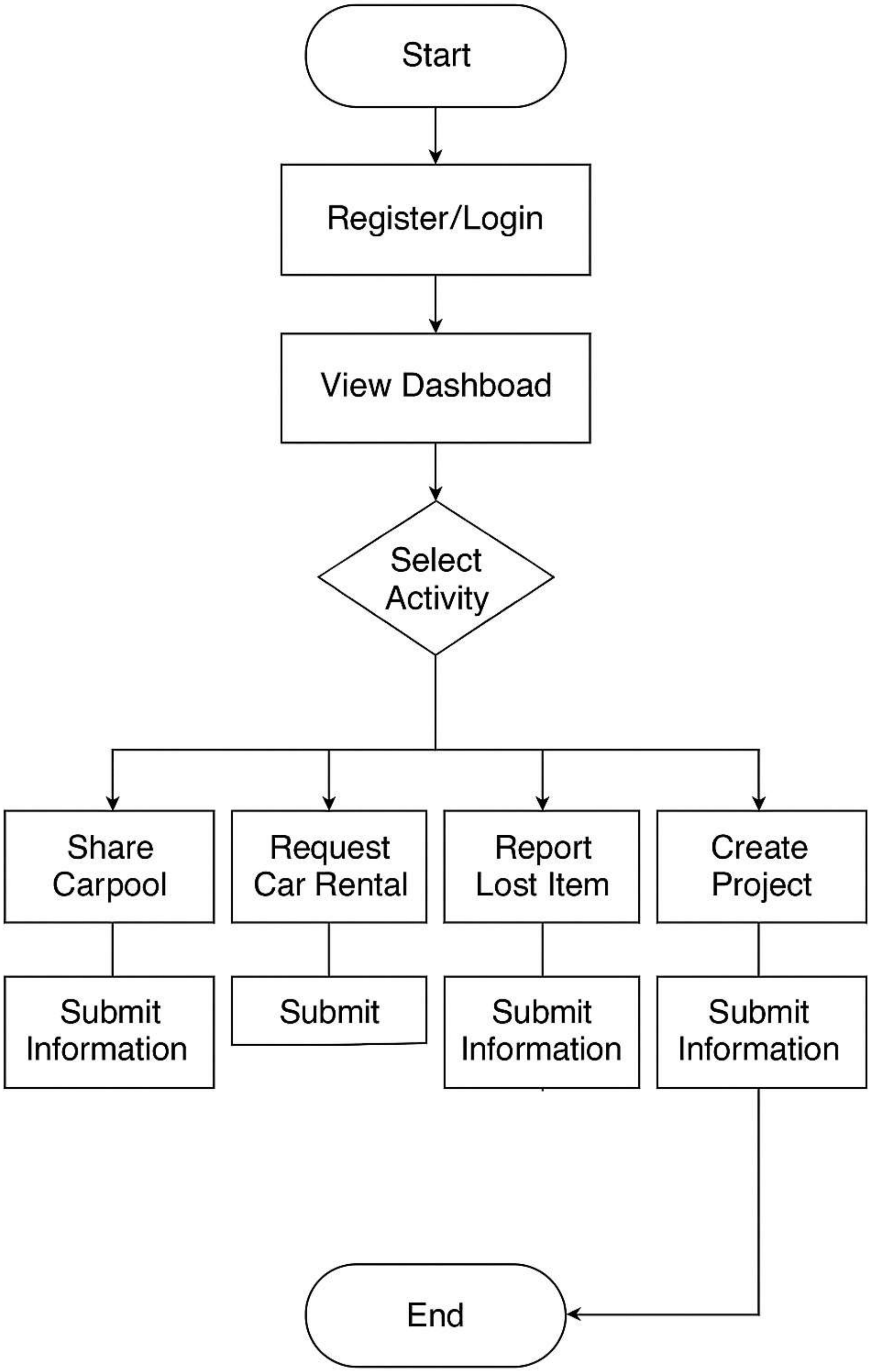


Figure 3: Activity diagram

5.3.6 Sequence Diagram

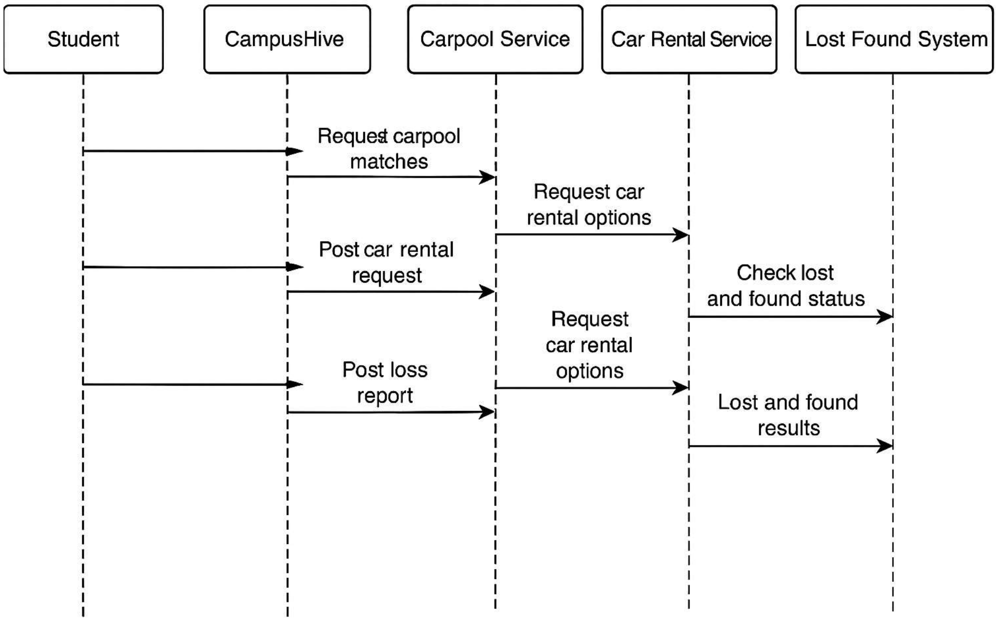


Figure 4: Sequence diagram

5.3.7 Data Architecture

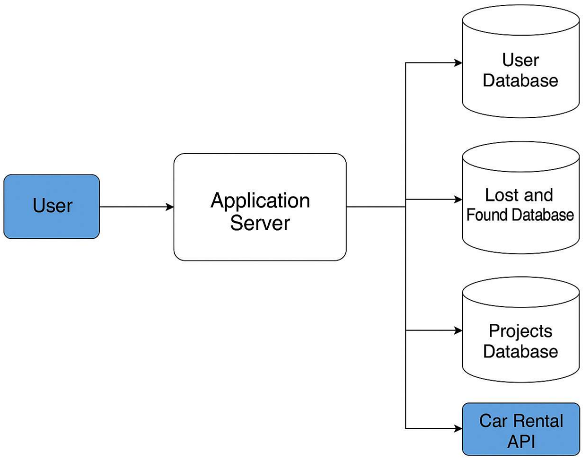


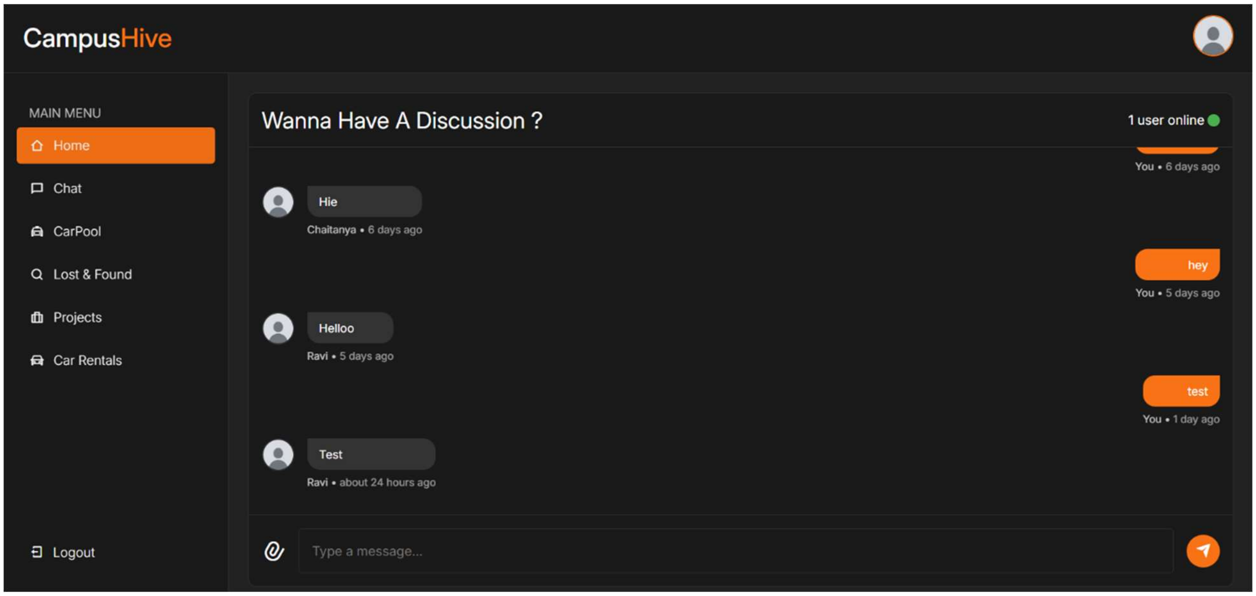
Figure 5: Data Architecture

6. USER INTERFACE

6.1 UI Description

Campus Hive is a web application developed using the MERN stack, with the frontend built using React.js. The website UI features a clean, responsive and consistent layout. The design provides usability of platform different screen sizes such as desktops, tablets and mobiles. The platform includes multiple pages for public chat, direct messages, lost &found, carpooling, and project collaboration, each designed for clarity and ease of use. Real-time interactions are enabled through WebSocket-based communication. Future improvements may include light mode, advanced filtering, and animations for better engagement.

6.2 UI Mockup



Wanna Have A Discussion ?

1 user online

You • 6 days ago

Hie

Chaitanya • 6 days ago

hey

You • 5 days ago

Helloo

Ravi • 5 days ago

test

You • 1 day ago

Test

Ravi • about 24 hours ago

Figure 6: UI Mockup

7.PSEUDO CODE OF CORE FUNCTIONALITY

```
const socketHandler = (io) => {

  const onlineUsers = new Map();

  io.on("connection", (socket) => {
    console.log("User connected:", socket.id);

    socket.on("join", (userId) => {
      onlineUsers.set(userId, socket.id);
      io.emit("onlineUsers", Array.from(onlineUsers.keys()))
    });

    socket.on("sendMessage", (data) => {
      io.emit("receiveMessage", data);
    });

    socket.on("privateMessage", ({ receiverId, message }) => {
      const receiverSocketId = onlineUsers.get(receiverId);
      if(receiverSocketId) {
        io.to(receiverSocketId).emit("receivePrivateMessage", message);
      }
    });

    socket.on("disconnect", (reason) => {
      for (let [userId, id] of onlineUsers.entries()) {
        if (id === socket.id) {
          onlineUsers.delete(userId);
          break;
        }
      }
      io.emit("onlineUsers", Array.from(onlineUsers.keys()));
      console.log("User disconnected:", socket.id, reason);
    });
  });
}

module.exports = socketHandler;
```

8. PROJECT CLOSURE

This section elucidates the overall lookup at the project and some of the future works that may enhance the solution.

8.1 Goals / Vision

Our initial vision with this project was to enhance communication within the University's campus and help students with their daily life problems and as the project progressed, we moved on to developing features like Carpools, Lost & Found, Projects, and Car Rentals which would work on the core functionality of messaging system.

The goal was to create such an app that students would use every day where features like Carpool, Lost & Found came in place. Campus Hive turned exactly how it was visioned and the problems it was meant to solve, that is making communication better not only publicly but wolving problems of students through better communication with each other.

8.2 Delivered Solution

Our intended deliverable was a web-based platform, Campus Hive, to connect university students and improve campus communication. The app was designed to include features like real-time chat, carpooling requests, lost and found posts, and project collaboration, all in an easy-to-use interface.

The delivered solution is a fully functional web prototype built with the MERN stack, covering all core features. It allows users to connect with other users, share resources, and collaborate on projects, with support for real-time messaging, and file sharing.

The current version is modular and scalable, with plans for future improvements like event management, job boards, and real-time campus data integration.

8.3 Remaining Work

To further develop the existing prototype, an admin panel can be added which would control the users, be able to send announcements publicly, Clubs feature where users can add their resume and the role-based club leaders can hire members for the clubs easily and efficiently.
Campus Data can be integrated within the application to make it an essential campus application for students for all their needs.
Conducting user acceptance testing to ensure all features are functional and operating well at a large-scale user base.

REFERENCES

1. https://developer.mozilla.org/en-US/docs/Web
2. https://www.w3schools.com
3. https://digiicampus.com/