

KATHMANDU UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DHULIKHEL, KAVRE



A PROJECT REPORT

ON

“ROUTE HIVE”

[COURSE CODE : COMP -206]

(FOR PARTIAL FULFILMENT OF II /I IN COMPUTER ENGINEERING)

SUBMITTED BY

Aarogya Ghimire (16)

Basanta Khanal (26)

Ghanshyam Khatri (27)

Shaswat Poudel (36)

Shasank Singh Thakuri (55)

SUBMITTED TO

Mr. Suman Shrestha

Department of Computer Science and Engineering

SUBMITTED ON

5th August 2025

Acknowledgement

We would like to express our sincere gratitude to all individuals and institutions who contributed to the successful completion of this project.

First and foremost, we extend our deepest appreciation to our project supervisor, **Asst. Prof. Sushil Nepal**, for his invaluable guidance, continuous support, and constructive feedback throughout the course of this work. His expertise and encouragement were instrumental in shaping the objectives, methodology, and overall quality of the project.

We also acknowledge the dedication, cooperation, and teamwork of all project members - **Aarogya Ghimire, Basanta Khanal, Ghanshyam Khatri, Shaswat Poudel, and Shasank Singh Thakuri**. The collective effort, commitment, and perseverance of each member were essential to the successful completion of this work.

Furthermore, we are grateful to the **Department of Computer Science and Engineering, Kathmandu University**, for providing the resources, guidance, and platform necessary to carry out this project.

Finally, we extend our thanks to all those who, directly or indirectly, supported and encouraged us during this endeavor.

Bonafide Certificate

This project Work on

“Route Hive”

is the bona fide work of

Aarogya Ghimire

Basanta Khanal

Ghanshyam Khatri

Shasank Singh Thakuri

Shaswat Poudel

who carried out the project work under my supervision.

Project Supervisor

Mr. Sushil Nepal

Assistant Professor

Department of Computer Science and Engineering

5th August 2025

Abstract

Route Hive is a modern, web-based transportation management system designed to streamline and digitize university bus operations. Recognizing the inefficiencies of traditional paper-based systems still in use at institutions like Kathmandu University, this project introduces a platform that improves punctuality, communication, and operational transparency. Built using React for the frontend and Flask with SQLite on the backend, Route Hive supports three key user roles: Admin, Driver, and Student. The system offers secure login, role-specific dashboards, enrolment management, and simulation of bus location. Admins can review applications, assign routes, and manage notifications; students can monitor bus locations and view their profiles; drivers can access personal information. Route Hive emphasizes automation, user-friendliness, and scalability, significantly enhancing the reliability, convenience, and overall efficiency of university transportation systems.

Keywords: Route-Hive, transportation system, bus tracking, React, Flask, SQLite, role-based access, enrollment, automation, transparency, scalability

Table of Contents

Acknowledgement	i
Bonafide Certificate	ii
Abstract.....	iii
List of Figures.....	vi
Acronyms and Abbreviations.....	vii
CHAPTER I: Introduction	1
1.1. Background	1
1.2. Objectives.....	2
1.3. Motivation and Significance.....	2
CHAPTER II: Related and Existing Work	3
2.1. Sajha Plus	3
2.2. Track School Bus.....	3
CHAPTER III: Design and Implementation.....	4
3.1. Design Pattern	4
3.2. System Design.....	4
3.2.1. System Flow Chart.....	4
3.2.2. Database Design	7
3.3. System Requirements and Specifications	8
3.3.1. Software Requirements	8
3.3.1. Hardware Specification	8
CHAPTER IV: Discussion on Achievements.....	9
4.1. Features.....	9
4.2. Limitations.....	10

CHAPTER V: Conclusion and Recommendations	11
5.1. Future Enhancements.....	11
5.3. Conclusion	11
Reference	13
Appendix I	14
Appendix II.....	15

List of Figures

FIGURE 3. 1 : ENROLLMENT MODULE	5
FIGURE 3. 2 : LOGIN MODULE	6
FIGURE 3. 3 : DASHBOARD MODULE	7
FIGURE I. 1: GANTT CHART.....	14
FIGURE II. 1: ADMIN DASHBOARD ROUTE_HIVE.....	15
FIGURE II. 2: EMAIL DEMONSTRATION	15
FIGURE II. 3: MAP SIMULATION ROUTE_HIVE	16
FIGURE II. 4: PROFILE ROUTE_HIVE	16
FIGURE II. 5 :USER BUS CARD	17
FIGURE II. 6:HOME PAGE ROUTE_HIVE.....	17
FIGURE II. 7 : BUS DETAILS VIEW ROUTE_HIVE	18

Acronyms and Abbreviations

API – Application Programming Interface

ER – Entity Relationship

GPS – Global Positioning System

GUI – Graphical User Interface

HTTP – Hypertext Transfer Protocol

JSON – JavaScript Object Notation

KU – Kathmandu University

REST – Representational State Transfer

SQL – Structured Query Language

UI – User Interface

UX – User Experience

Flask – Lightweight Python Web Framework

React – JavaScript Library for Building User Interfaces

SQLite – Lightweight SQL Database

CHAPTER I: Introduction

Efficient transportation management is vital for modern educational institutions, directly influencing punctuality, safety, and convenience for students and staff. However, many universities, including Kathmandu University, still rely on outdated, paper-based systems that lack real-time updates, automation, and transparency. These shortcomings often result in miscommunication, delays, and operational inefficiencies.

To address these challenges, Route Hive introduces a smart, digital transportation management platform designed to enhance coordination and reliability. Built using modern technologies such as React for the frontend and Flask with SQLite for the backend, the system provides a user-friendly, role-based interface for Admins, Students, and Drivers. It ensures secure login, supports user and driver enrollment, and allows efficient route allocation. Admins can manage routes and respond to applications, while students can view their assigned routes and monitor relevant updates.

By focusing on simplicity, transparency, and automation, Route Hive aims to streamline operations, improve punctuality, and increase overall satisfaction with university transportation services.

1.1. Background

Transportation management plays a crucial role in ensuring that university operations run smoothly. In many institutions, especially in developing regions, transport systems are still managed using manual, paper-based processes. These traditional methods often lead to slow data handling, limited communication, and difficulty in tracking buses or updating schedules.

Previous approaches to solving these problems have included partial digitalization through spreadsheets, simple mobile communication channels, or third-party tracking applications. However, these solutions often fail to provide a centralized, role-specific, and automated platform tailored to the unique needs of universities.

At Kathmandu University, the transportation system continues to operate without a dedicated digital platform, resulting in inefficiencies such as delayed bus schedules, unclear communication

between drivers and students, and difficulty managing enrollment records. Recognizing this gap, our team designed Route Hive as a complete, web-based solution that integrates route allocation, real-time location simulation, and role-based dashboards into a single system.

1.2. Objectives

- To digitize and optimize university transportation services.
- To enable secure, role-based access for Admin, Drivers, and Users.
- To manage and assign bus routes efficiently.
- To ensure effective coordination between Users, Administration and Drivers

1.3. Motivation and Significance

Current transportation systems at Kathmandu University face several issues. Paper-based ID cards are prone to wear and damage, slowing down verification. There is no reliable method for providing real-time information or notifying users about bus delays or cancellations. These issues cause confusion and disrupt daily routines.

Route Hive solves these problems by offering a digital platform that delivers real-time alerts, eliminates the need for physical IDs, and facilitates effective communication between users and administrators. With a simplified enrolment process and intelligent route management, the system greatly enhances reliability and user experience in university transport services.

CHAPTER II: Related and Existing Work

There are similar applications related to our work. Some of them are described as follows:

2.1. Sajha Plus

Sajha plus is an easy to use, bilingual android-based application. It allows users to select the route, arrival time of the bus, next stop, location etc. The desired stations can be selected from the Route List or users can also search for specific routes filtered by station on the search bar. The Route Cards display the number of active operational buses, number of stops on the route, and total distance covered by the route. Selecting a particular Route Card opens a map page that shows the buses closest to the user's current GPS location. Tapping off the red circles on the map shows the station name and tapping on the bus icons displays the bus number.

2.2. Track School Bus

The other widely used commercial school transportation management platform is Track School Bus. It can be accessed through websites, android and iOS apps. It allows administrators, drivers, parents, and students to interact with the platform based on their roles, enabling seamless management and monitoring of daily transport operation. The system provides real-time alerts and notifications for bus arrivals, delays, and route changes via SMS or mobile push notifications, thereby improving communication and safety. Administrators can efficiently assign students to buses based on stops, manage bus schedules.

CHAPTER III: Design and Implementation

Before approaching the design pattern, it is important to note that the development of Route Hive followed a systematic approach focusing on modularity, scalability, and user experience. The team adopted a collaborative workflow, assigned role-specific tasks, and utilized version control to ensure smooth coordination throughout the development process.

3.1. Design Pattern

The design and implementation of Route Hive aimed to create a fast, user-friendly, and scalable web application using React + Vite, Tailwind CSS, Flask, and Python. The system was modularly structured into three primary roles: Admin, Driver, and User (Student/Teacher). Each role has dedicated functionalities and views. The project was collaboratively developed by five team members using Git for version control.

3.2. System Design

To maintain scalability and clarity, the system was divided into modules based on user roles:

- **Admin:** Manages route assignments, handles user and driver enrollment requests, and sends system notifications.
- **Driver:** Logs in to access and view their profile information.
- **User (Student/Teacher):** Views personal profile and tracks live bus location on the map.

These role-specific modules ensure a clear separation of responsibilities and functionalities, making the system easier to maintain and extend. The interactions between these roles and the system's core processes are visually represented in the following subsections.

3.2.1. System Flow Chart

The system flow chart represents the overall flow of the Route Hive system, outlining how data moves between users and the system. It visually explains how users interact with different modules: Enrollment, Login, and Dashboard and how the system responds to their actions.

3.2.1.1. Enrollment Module

New users (students or drivers) access the registration form, fill in their details, and submit it. The system validates the inputs and saves the data to the database. Upon successful enrollment, users are redirected to the login page.

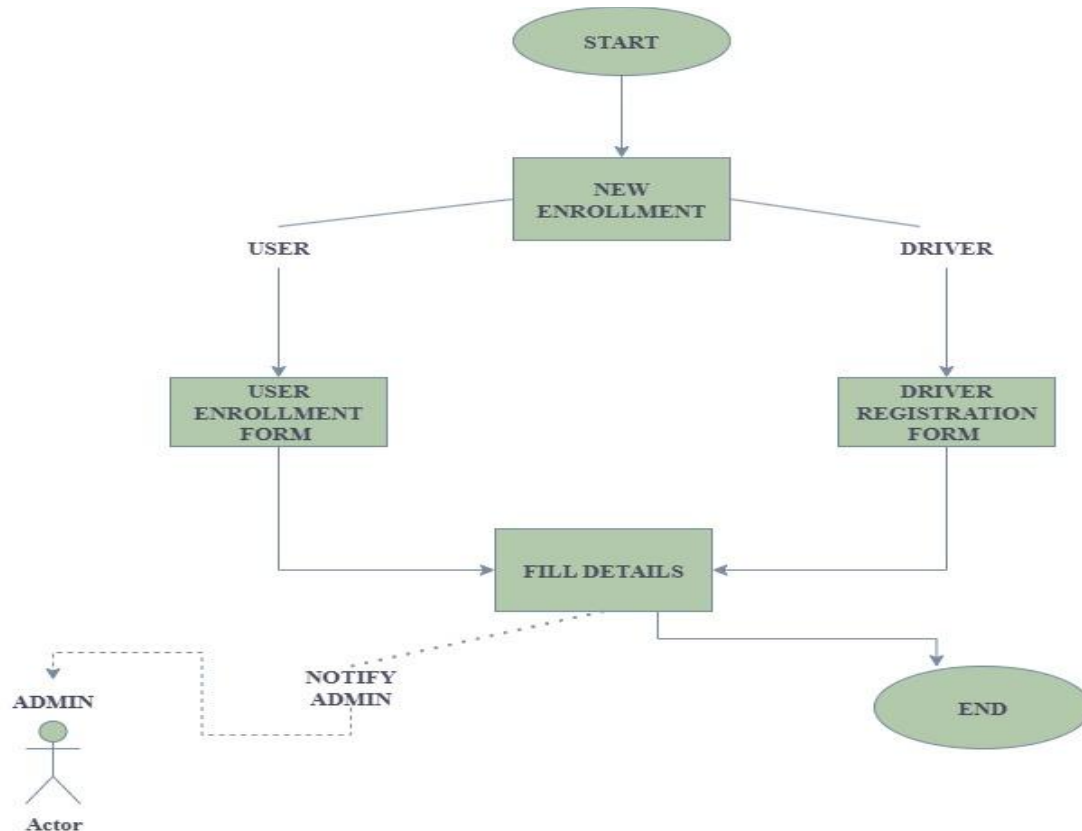


FIGURE 3. 1 : ENROLLMENT MODULE

3.2.1.2. Login Module

Users input their credentials to access the system. The system checks the validity of the entered email and password, identifies the role (admin, driver, or student), and redirects them to the respective dashboard. Errors are displayed for invalid logins.

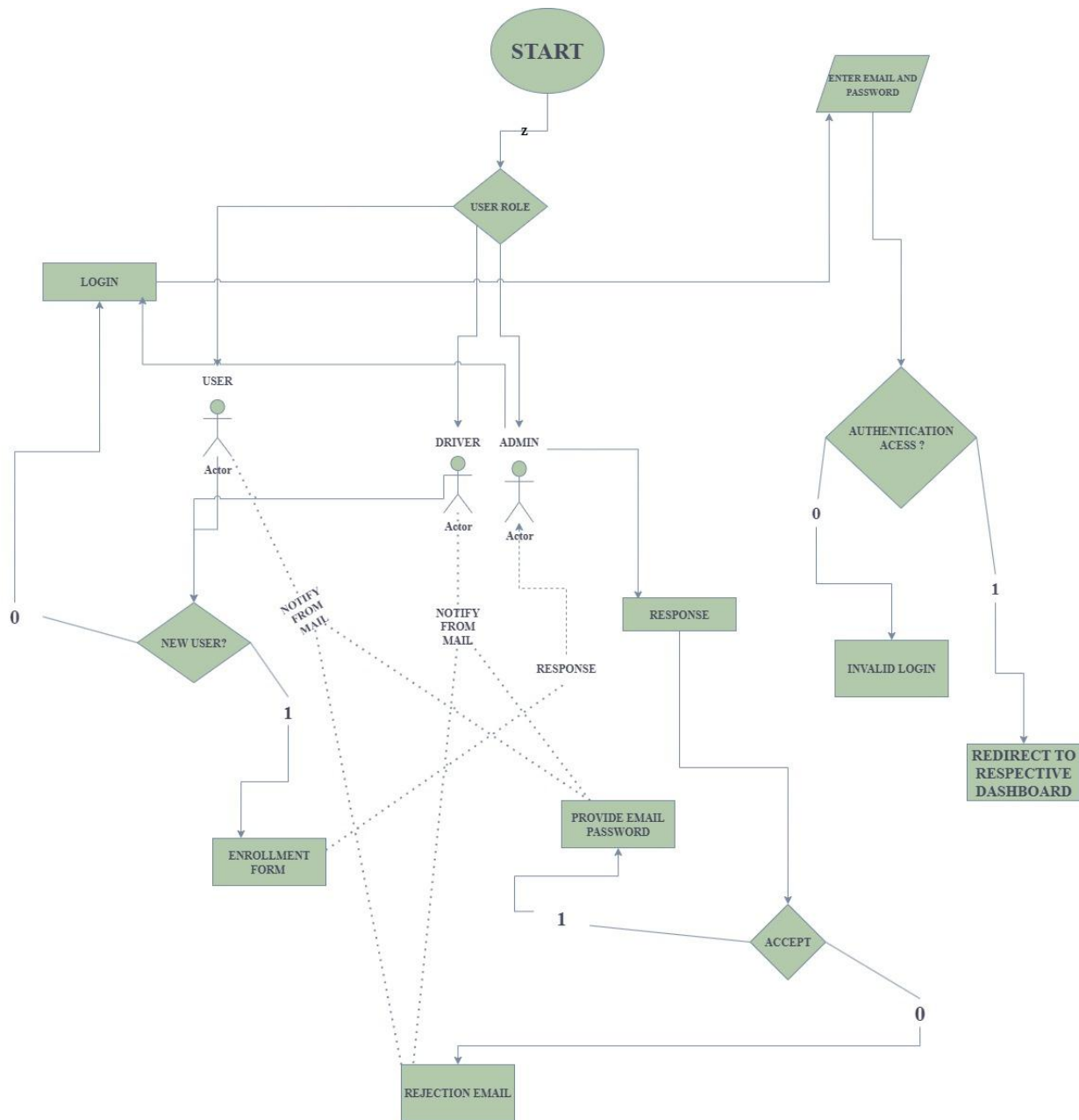


FIGURE 3. 2 : LOGIN MODULE

3.2.1.3. Dashboard Module

After logging in, each user is taken to a role-based dashboard. Admins can manage users and routes, drivers can view and manage their assigned routes, and students can view transport details. All interactions are processed and reflected in real time.

3.3. System Requirements and Specifications

The Route Hive system requires specific software and minimal hardware resources to operate effectively. The chosen technologies were selected for their compatibility, ease of development, and ability to meet the project’s functional requirements.

3.3.1. Software Requirements

The Route Hive system is built using a combination of programming languages, frameworks, and database technologies that work together to deliver a functional, secure, and user-friendly transport management application.

3.2.1.1. Front-End and Back-End Technologies

- **Python** – A high-level, general-purpose programming language valued for its readability, dynamic typing, and versatility. Supports multiple paradigms and is used here for backend development.
- **React** – A JavaScript-based UI library for building dynamic, component-based user interfaces, enabling responsive and scalable frontend development.
- **Flask** – A lightweight Python web framework ideal for developing web applications and APIs, offering simplicity, flexibility, and extensibility.

3.2.1.2. Database

- **SQLite** – A serverless, self-contained SQL database engine known for its efficiency, reliability, and ease of integration, making it suitable for small to medium-scale applications.

3.3.1. Hardware Specification

The application can operate on any modern desktop or laptop capable of running the required software stack, making it easy to deploy without specialized hardware.

CHAPTER IV: Discussion on Achievements

Route Hive implements key features such as secure role-based login, an efficient enrollment system, and simulated bus tracking to enhance user experience. The admin dashboard provides centralized control over enrollments and bus management. Together, these features improve the coordination and reliability of university transportation services.

4.1. Features

The Route Hive system successfully integrates several key features to improve university transport management, with a focus on usability and administrative control. The main functionalities include:

- **Secure Login** - Users input their registered email and masked password to access the system, ensuring secure and controlled access. Passwords are never stored in plain text. Instead, we use bcrypt(Python module), a secure one-way hashing algorithm, to hash and store passwords safely. During login, the entered password is compared with the hashed version using bcrypt verification. If a user forgets their password, an OTP (One-Time Password) is sent to their registered email. The OTP is stored in the database along with a 2-minute expiry timestamp. The system verifies the OTP before allowing the user to reset their password, ensuring both security and time-bound access.
- **Enrollment System** – Both Drivers and Users can apply for enrollment through an online form. The admin reviews these accordingly. Users and drivers submit their personal information (name, email, phone number, etc.) via a form. The system checks for existing records in the enrolled user/driver tables to prevent duplicate registrations. After enrollment, the admin can either accept or reject. In the both cases the notification is sent in their email notifying the message sent according to them.
- **Profile Access** – After logging in, all users can view their personal profile, which contains relevant information such as contact details, assigned bus, and pickup location .When the Login route checks if the user is admin or user or driver accordingly. It sets the session data accordingly and approves the users.

- **Map Integration** – The system currently simulates bus movement on an interactive map, providing users with a visual representation. Simulating the bus's starting point by generating a random location 500–1000 meters away from the user's pickup point. Uses a function to calculate a random point around the user using spherical geometry (Earth's radius). This is used to visualize the bus approaching on the frontend map. Notifications are sent when the simulated bus is within 100 meters of the user's stop. Implementation of real-time GPS tracking is planned for future versions.
- **Admin Dashboard** – A centralized panel for administrative tasks, including:
 - **Response**– Viewing and managing incoming enrollment applications from Drivers and Users.
 - **Management Module** – Adding bus details, assigning routes, and accessing detailed information about buses and routes.

4.2. Limitations

While Route Hive incorporates several useful features to improve university transport management, some limitations remain that should be addressed in future work:

- **No GPS Live Tracking:** The current implementation lacks support for real-time GPS updates, limiting the accuracy of bus location tracking.
- **Lack of Real-Time Communication:** There is no functionality for direct messaging between drivers and users, which could enhance coordination and responsiveness.
- **No Load Balancing Strategy:** The system does not include mechanisms to handle high traffic or scale efficiently when managing multiple buses and users simultaneously.
- **No Feedback System:** Users currently have no option to submit feedback or report issues, which limits the ability to gather user input for continuous improvement.

CHAPTER V: Conclusion and Recommendations

5.1. Future Enhancements

To further improve the functionality and user experience of Route Hive, the following are proposed:

- **Implementation of Real-Time GPS Tracking:** Integrate GPS hardware and APIs to provide accurate, live bus location updates, improving the reliability of the tracking system.
- **Development of a Communication Module:** Enable real-time messaging between drivers and users to facilitate better coordination and instant communication.
- **Scalability and Load Balancing:** Introduce strategies such as load balancing and optimized database queries to ensure smooth performance under heavy usage and accommodate a larger user base.
- **User Feedback and Issue Reporting:** Implement a feedback system that allows users to report problems, suggest improvements, and rate services, enabling continuous system enhancement based on user input.
- **Mobile Application Support:** Develop mobile apps for both Android and iOS platforms to increase accessibility and ease of use for all stakeholders.
- **Advanced Notification System:** Enhance notification features to include customizable alerts based on user preferences and bus schedules.

5.3. Conclusion

We acquired important knowledge and abilities during the Route Hive project's development, which greatly enhanced our capacities. To make sure our application can manage expansion and user demand; we learned how to create a scalable system. We expanded our technical knowledge by investigating and utilizing industry-standard technologies like SQ Lite, python, flask, react. Also, we researched the map integration section which was very beneficial for us. We improved our knowledge of database joins and relationships, which helped us handle and retrieve data more effectively. Our ability to collaborate was enhanced by teamwork, emphasizing the value of

coordination and communication. To improve user engagement and learning, we also investigated the gamification of computer systems. These experiences have given us a broad range of skills that will help us tackle future software development and teamwork challenges.

(Thakuri, LoginModule for RouteHive, 2025)

Reference

Admin. (2023, 1 26). *Sajha Yatayat launched new mobile application “Sajha Plus”*. Retrieved from What The Nepal: <https://whatthenepal.com/2023/01/26/sajha-yatayat-launched-new-mobileapplication-sajha-plus/>

Thakuri, S. S. (2025, August 2). *enrollment module diagram for RouteHive*. Retrieved from diagrams.net(formerlydraw.io):<https://app.diagrams.net/#G1tORIkJXMDIimVSDYji04I2hEiWVIX4T#%7B%22pageId%22%3A%22jWakUcpwKSntBfZPDVc5%22%7D>

Thakuri, S. S. (2025, August 2). *LoginModule for RouteHive*. Retrieved from diagrams.net (formerlydraw.io):<https://app.diagrams.net/#G1E3W67YpCRbInLX5FNpMBX4mz4mYXYkm-#%7B%22pageId%22%3A%22f6YY3VQTGaUbHhZigtlB%22%7D>

Appendix I

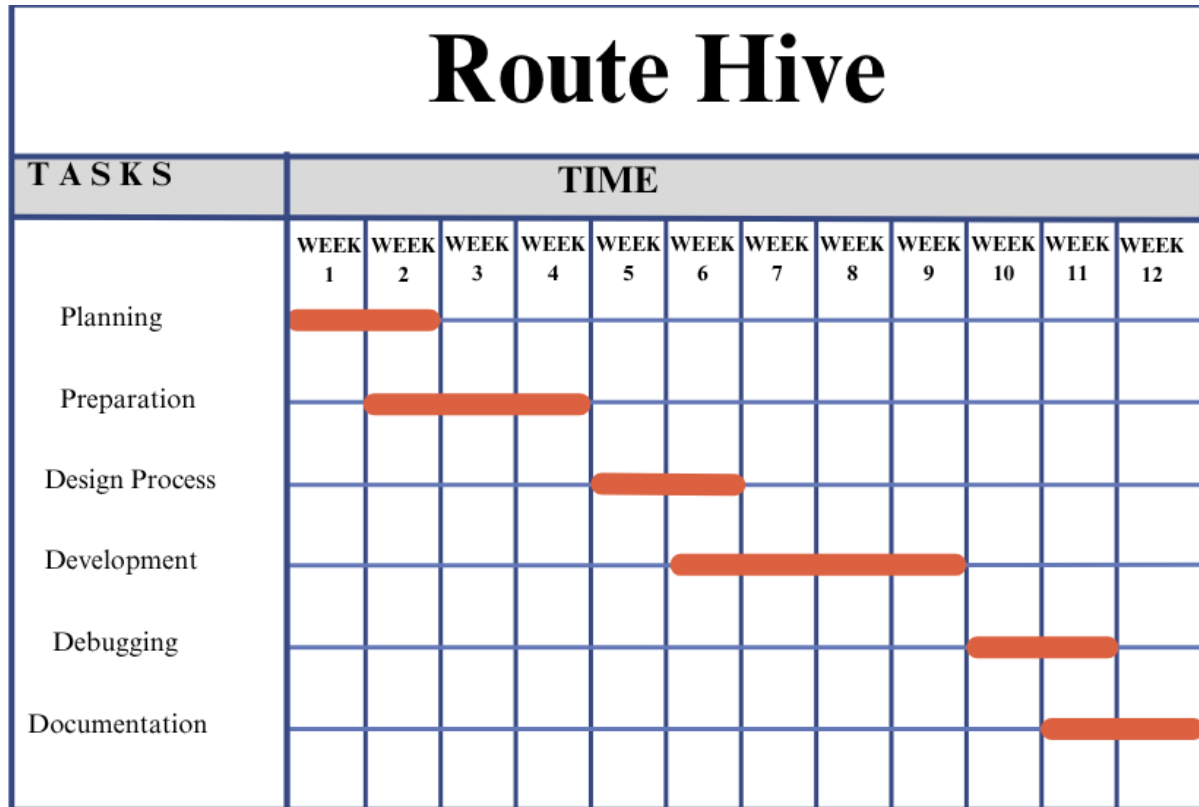


FIGURE I. 1: GANTT CHART

Appendix II

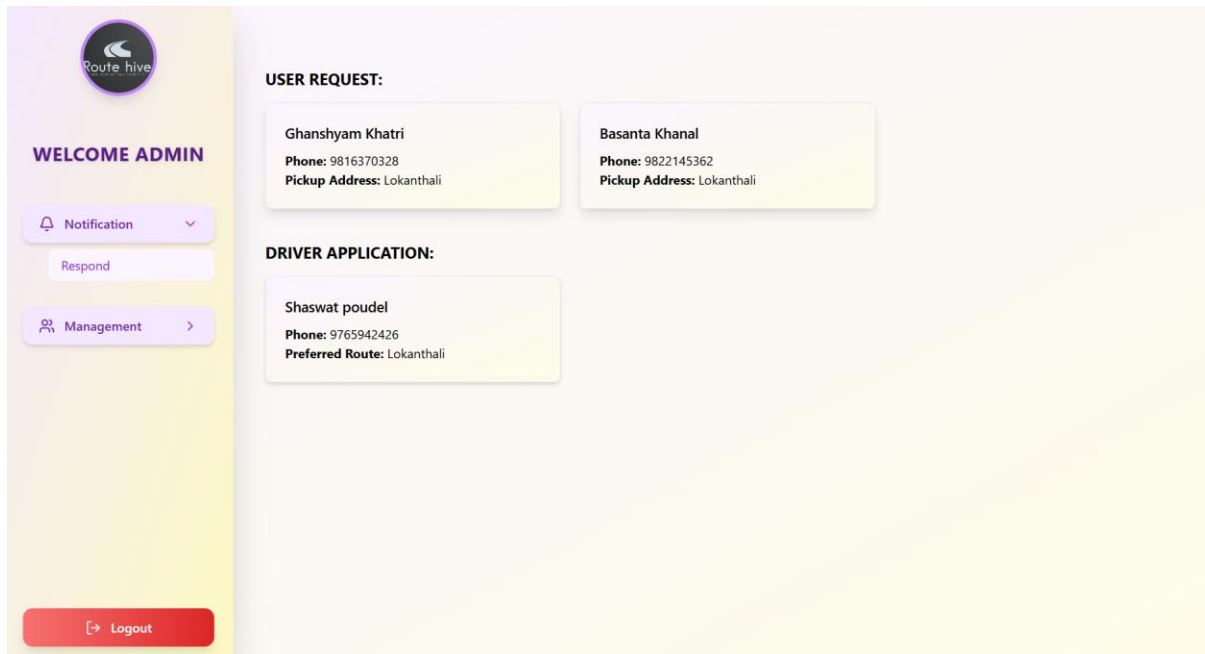


FIGURE II. 1: ADMIN DASHBOARD ROUTE_HIVE

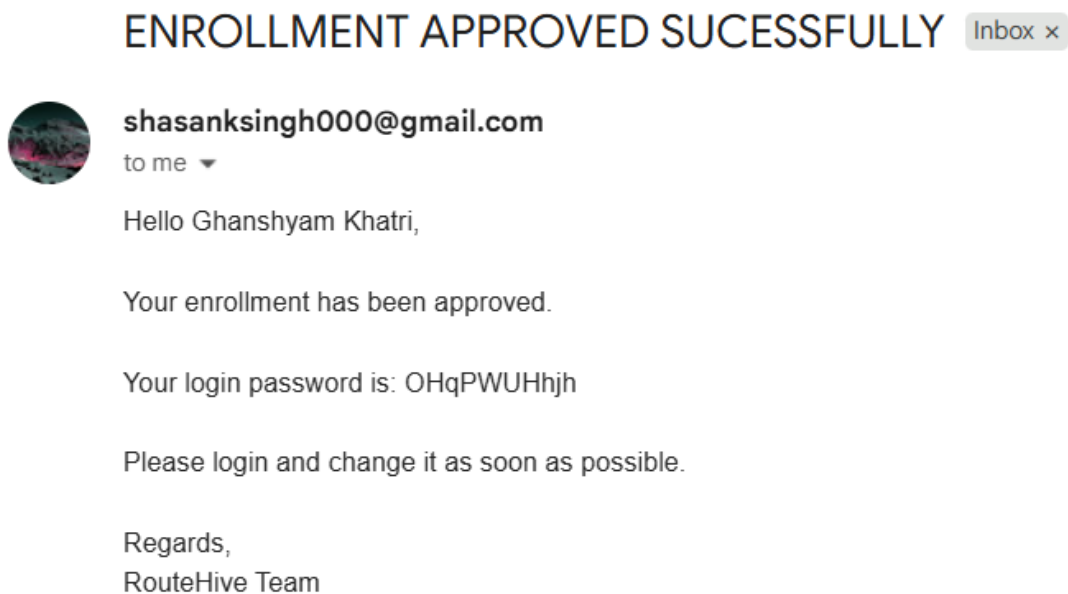


FIGURE II. 2: EMAIL DEMONSTRATION

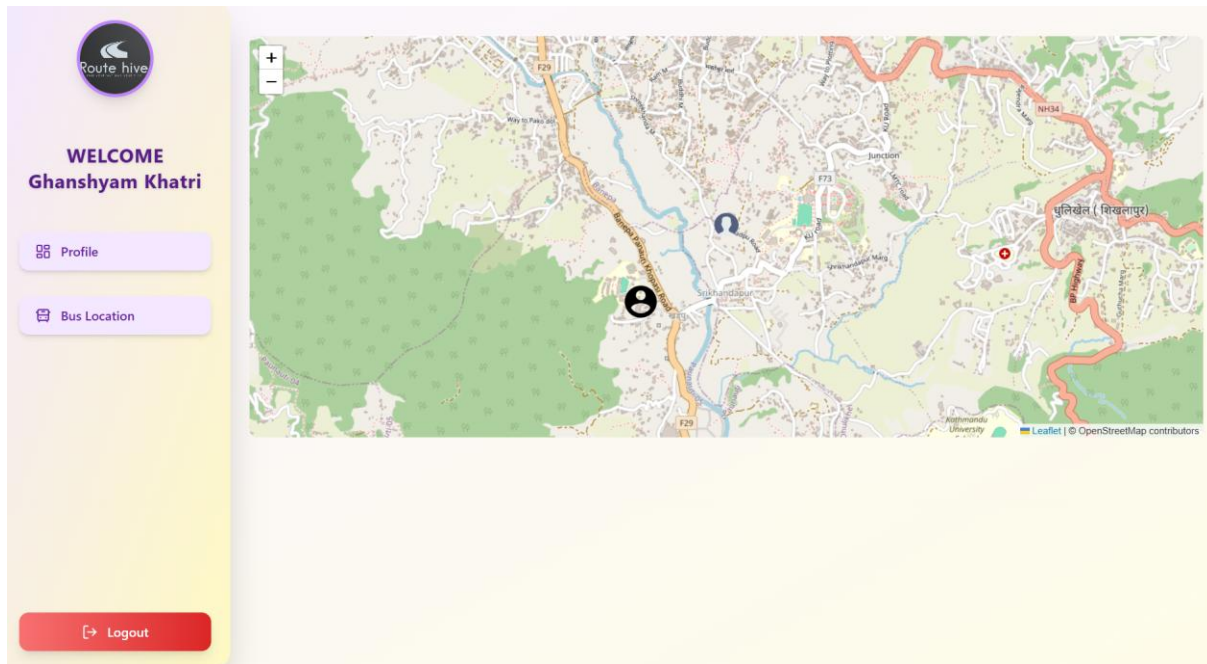


FIGURE II. 3: MAP SIMULATION ROUTE_HIVE

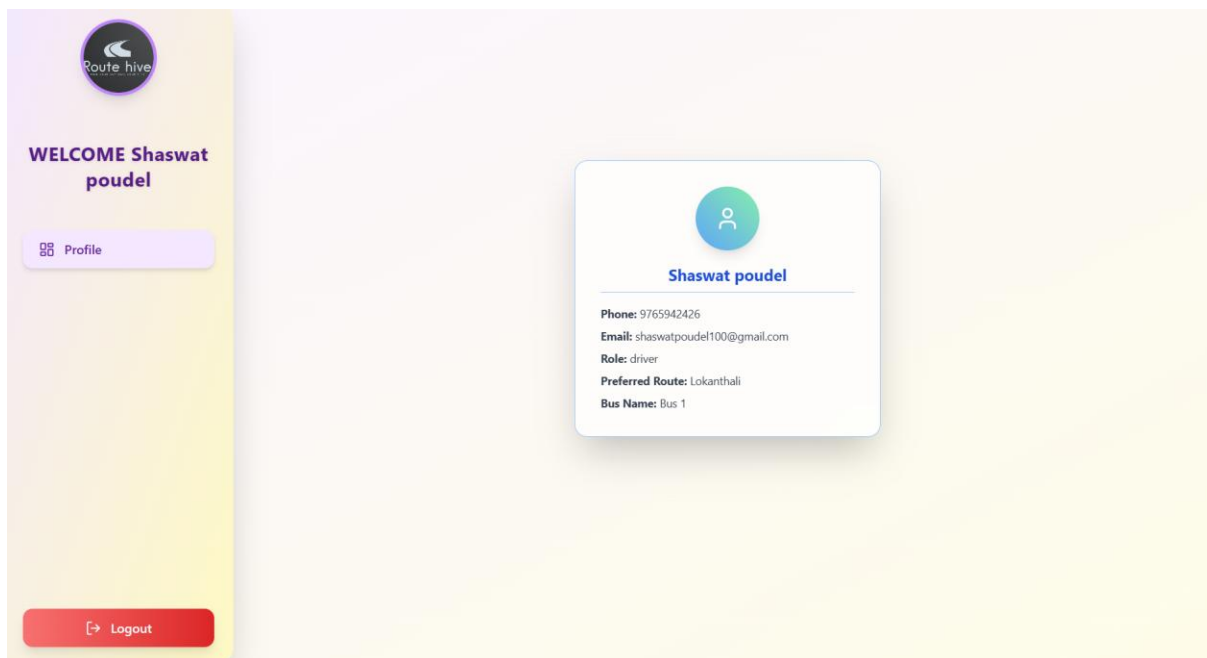


FIGURE II. 4: PROFILE ROUTE_HIVE

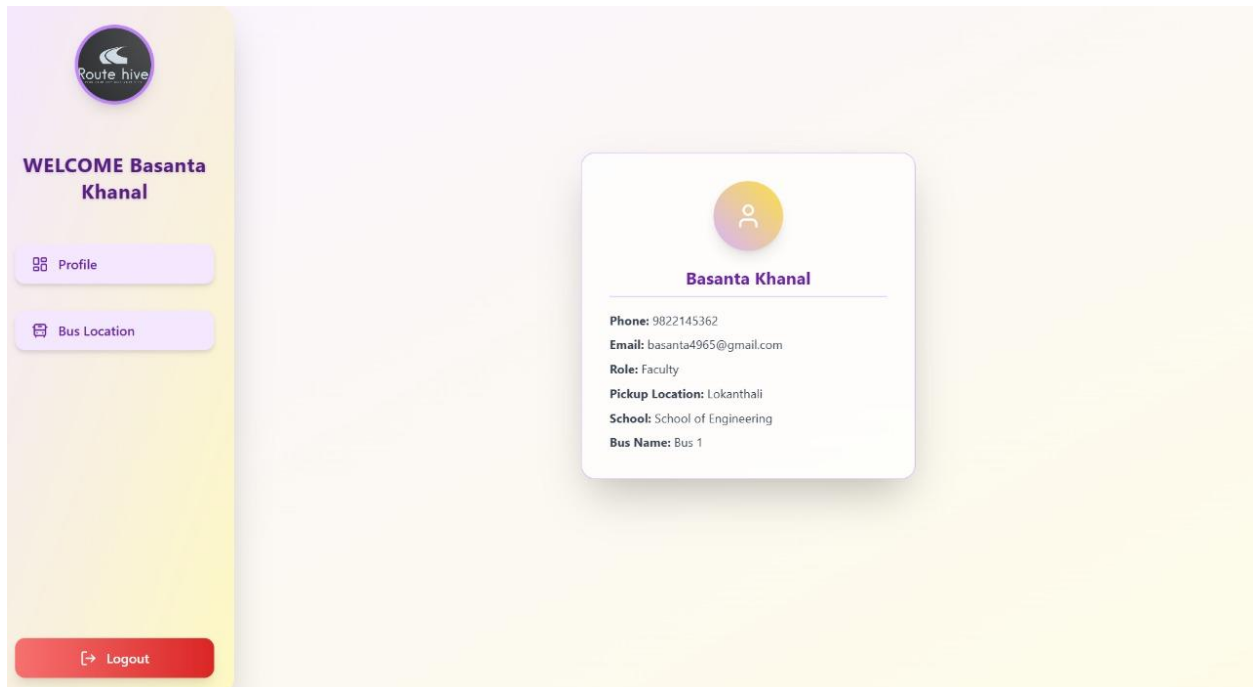


FIGURE II. 5 :USER BUS CARD

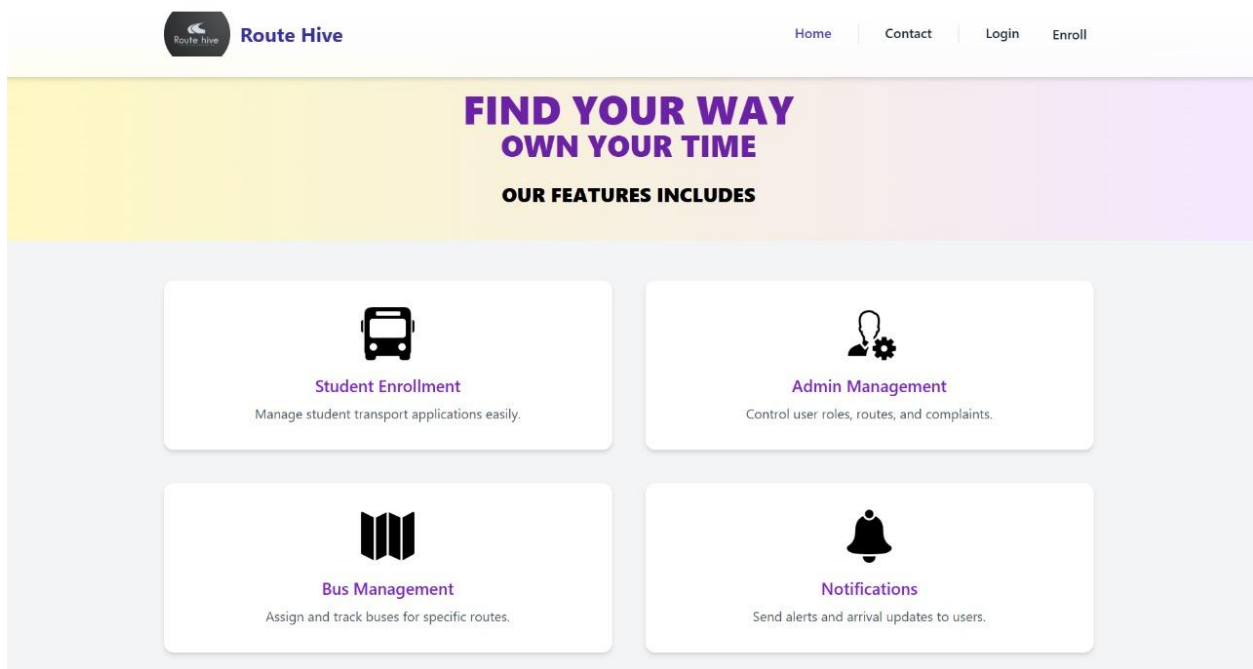




FIGURE II. 6:HOME PAGE ROUTE_HIVE




WELCOME ADMIN

Notification >

Management 

Add Bus

Bus Detail

 Logout

Users Assigned to Bus 1

Name	Email	Phone	Pickup Location	Role	School
Shasank Singh Thakuri	shasanksingh000@gmail.com	9761615866	Lokanthali	Research Assistant	School of Engineering
Ghyanshyam Khatri	shasank5024@student.ku.edu.np	9861370328	lokanthali	Student	School of Engineering
Ghanashyam Khatri	ghanashyamkhatri958@gmail.com	9816370328	Lokanthali	Student	School of Engineering
Basanta Khanal	basanta4965@gmail.com	9822145362	Lokanthali	Faculty	School of Engineering

Back to Bus List

FIGURE II. 7 : BUS DETAILS VIEW ROUTE_HIVE