CAMPUS CARPOOLING: OPTIMIZED RIDESHARING FOR STUDENTS USING HYBRID RIDESHARING ALGORITHM

PHASE I REPORT

Submitted by

LOGESHWARAN ELUMALAI 210701134

MOHAMED AADHIL A 210701159

In partial fulfillment for the requirement of award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI 600 025

November 2024

RAJALAKSHMI ENGINEERING COLLEGE
CHENNAI

BONAFIDE CERTIFICATE

Certified that this report titled **"CAMPUS CARPOOLING: OPTIMIZED RIDESHARING FOR STUDENTS USING HYBRID RIDESHARING ALGORITHM"** is the bonafide work of **LOGESHWARAN ELUMALAI (210701134), MOHAMED AADHIL A (210701159),** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr. P. Kumar, M.E., Ph.D.
HEAD OF THE DEPARTMENT
Professor
Department of Computer Science and Engineering
Rajalakshmi Engineering College
Thandalam, Chennai - 602 105

SIGNATURE
Ms. Dharshini B S, M.E.
SUPERVISOR
Assistant Professor
Department of Computer Science and Engineering
Rajalakshmi Engineering College
Thandalam, Chennai - 602 105

Submitted to Project Viva-Voce Examination held on____________________________

Internal Examiner **External Examiner**

ABSTRACT

A student-centric carpooling service tailored to address the unique transportation needs of college students. The platform allows users to post their starting locations, available seats, and routes, while enabling other users to select their desired pick-up points. The service employs a machine learning algorithm to optimize the matching process between riders and passengers, ensuring efficient and convenient travel arrangements. A unique feature of the system is dynamic pricing, which adjusts based on demand and supply factors, offering flexibility in payment options for rides, which can be either paid or free, depending on the rider's choice. The platform incorporates user registration and authentication to maintain security and trust within the community, with profiles for both riders and passengers that include ratings and reviews to establish a reliable user base. The machine learning component utilizes clustering and optimization algorithms to reduce total travel time and distance, thereby enhancing the overall efficiency of the service. Additional features such as an integrated payment gateway for paid rides, identity verification, emergency contact information, and ride tracking are included to ensure safety. In app messaging and notifications provide seamless communication between users. An admin panel is also provided for the management of users, rides, and payments, ensuring smooth operations and quick resolution of issues. The project aims to develop a sustainable, efficient, and cost-effective carpooling system tailored for college students, promoting environmentally friendly transportation solutions on campuses.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of ourlife and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E**., our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.**, and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.**, for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution. Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.**, our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, Ph.D**., Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Ms. Dharshini B S, M.E.** Department of Computer Science and Engineering. Rajalakshmi Engineering College for her valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Dr. T. Kumaragurubaran, M.E., Ph.D.** Department of Computer Science and Engineering for his useful tips during our review to build our project.

LOGESHWARAN ELUMALAI 210701134

MOHAMED AADHIL A 210701159

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	ACKNOWLEDGEMENT	ii
	LIST OF FIGURES	iii
	LIST OF ABBREVIATIONS	iv
1.	INTRODUCTION	1
	1.1 GENERAL	1
	1.2 OBJECTIVE	3
	1.3 EXISTING SYSTEM	4
	1.4 PROPOSED SYSTEM	6
2.	LITERATURE SURVEY	8
3.	SYSTEM DESIGN	15
	3.1 GENERAL	15
	3.1.1 SYSTEM FLOW DIAGRAM	15
	3.1.2 SEQUENCE DIAGRAM	17
	3.1.3 CLASS DIAGRAM	18
	3.1.4 USECASE DIAGRAM	20
	3.1.5 ARCHITECTURE DIAGRAM	21
	3.1.6 ACTIVITY DIAGRAM	22
	3.1.7 COMPONENT DIAGRAM	24
	3.1.8 COLLABORATION DIAGRAM	25

4.	PROJECT DESCRIPTION	26
	4.1 METHODOLOGIES	26
	4.1.1 RESULTS AND DISCUSSION	33
5.	CONCLUSION AND WORK SCHEDULE	36
	5.1 CONCLUSION	36
	5.2 FOR PHASE II	37
	REFERENCES	39
	APPENDIX	42

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.1	SYSTEM FLOW DIAGRAM	15
3.2	SEQUENCE DIAGRAM	17
3.3	CLASS DIAGRAM	18
3.4	USECASE DIAGRAM	20
3.5	ARCHETECTURE DIAGRAM	21
3.6	ACTIVITY DIAGRAM	22
3.7	COMPONENT DIAGRAM	24
3.8	COLLABORATION DIAGRAM	25
4.1	SPLASH SCREEN	28
4.2	LOGIN PAGE	28
4.3	HOME PAGE	28
4.4	DRIVER DASHBOARD	28

LIST OF ABBREVIATIONS

SNO	ABBREVIATION	EXPANSION
1	HRA	Hybrid Ridesharing Algorithm
2	API	Application Programming Interface
3	PWA	Progressive Web Application
4	K -Means	K-Means Clustering Algorithm
5	GA	Genetic Algorithm
6	ML	Machine Learning
7	NLP	Natural Language Processing
8	SMSs	Shared Mobility Systems
9	VKT	Vehicle Kilometers Traveled
10	CH	Contraction Hierarchies
11	TO-PP	Time-Optimal and Privacy Preserving
12	IOT	Internet of Things
13	REST	Representational State Transfer
14	MERN	MongoDB, Express, React, Node
15	DB	Database

CHAPTER 1

1. INTRODUCTION

1.1 GENERAL

Transportation remains a significant challenge for college students, who often rely on limited and sometimes unreliable options such as public transportation, cycling, or walking. These methods can be inconvenient, especially during peak hours or in regions with inadequate public transit infrastructure. The lack of affordable, efficient, and flexible transportation options can affect students' daily lives, contributing to wasted time, increased stress, and even missed opportunities for classes, extracurricular activities, and part-time jobs. This project proposes a student centric carpooling service designed to address these transportation challenges by providing a reliable, cost-effective, and environmentally friendly solution tailored specifically to the needs of college students.

The proposed carpooling platform allows students to register as either riders or passengers, enabling them to post their starting locations, available seats, and preferred routes, while passengers can select pick-up points that best match their schedules and preferences. This user-friendly system is built around a machine learning algorithm, specifically the K-means clustering algorithm, which is employed to create optimal carpool groups by clustering users based on geographical proximity and route similarity. The algorithm helps minimize total travel time and distance by matching riders and passengers with overlapping routes, ensuring efficient and convenient travel arrangements. This optimization not only improves the overall user experience but also contributes to reduced fuel consumption and lower carbon emissions, promoting sustainable practices within the college community.

Safety is a paramount concern in carpooling, and the platform is designed with robust security features, including user registration, identity verification, and emergency contact information, to create a safe and trustworthy environment for all participants. Profiles for both riders and passengers are equipped with ratings and reviews, which help build a reliable user base by promoting transparency and accountability. Building this project involves a multi-layered approach that leverages modern software development frameworks and tools. A relational database, such as PostgreSQL, will be integrated to manage user profiles, ride details, routes, and transaction histories, ensuring data consistency and integrity. The machine learning model, developed using Python libraries such as scikit learn, will be responsible for the clustering and optimization tasks to effectively match riders and passengers. The frontend of the platform will be developed using modern JavaScript frameworks like React or Angular, providing a responsive and intuitive interface for users to easily access and manage their rides. Additional features, such as real-time ride tracking using Google Maps API, in-app messaging for communication, and notifications for ride updates, are included to ensure a seamless user experience.

Moreover, the platform includes an admin panel for comprehensive oversight and management capabilities. Administrators can use this panel to monitor user activity, manage ride requests, handle disputes, and ensure compliance with community guidelines, thereby maintaining the platform’s integrity and operational efficiency. By integrating advanced machine learning techniques, secure payment processing, real time tracking, and effective user management features, this project aims to deliver a holistic carpooling solution. The goal is to create a sustainable, efficient, and community-driven transportation system that not only reduces costs and travel time for students but also fosters a sense of shared responsibility toward environmental conservation. Ultimately, this project seeks to transform how students manage their daily commutes, creating a model for other communities to adopt similar sustainable transportation solutions.

1.2 OBJECTIVE

The primary objective of the campus carpooling project is to address the transportation challenges faced by college students. By leveraging advanced algorithms and sustainable practices, the platform aims to provide an efficient, cost-effective, and user-friendly carpooling solution. Below are the key objectives of the project:

1. Efficient Ride Matching:

Develop an optimized ride-matching system using machine learning algorithms to group users based on proximity and preferences. This ensures minimal detours and reduced travel time for riders and passengers.

2. Dynamic Pricing Mechanism:

Implement a real-time dynamic pricing model that adjusts fares based on demand and supply. This provides flexibility for riders and drivers, ensuring fairness and affordability.

3. Safety and Trust:

Enhance safety through user authentication, identity verification, and emergency contact integration. Build a reliable platform with ratings and reviews to foster trust among users.

4. Environmental Sustainability:

Promote eco-friendly transportation by reducing the number of vehicles on the road. This helps lower CO2 emissions and contributes to a greener campus environment.

5. User-Friendly Experience:

Design a Progressive Web Application (PWA) for seamless access across devices. Include features like real-time tracking, notifications, and in-app messaging for a smooth and intuitive user experience.

1.3 EXISTING SYSTEM

Carpooling and ride-sharing platforms have gained popularity in recent years as viable solutions to transportation issues, offering economic and environmental benefits. However, most existing systems are designed to cater to a broad audience rather than focusing exclusively on niche groups like students. These platforms, such as Uber, Lyft, and BlaBlaCar, operate on generic models that prioritize maximizing user base and revenue, which can sometimes overlook the unique requirements of specific communities.

One significant feature of current ride-sharing platforms is their dynamic ride-matching algorithms. These systems connect passengers with drivers based on proximity and route similarity. While this ensures convenience for a wide range of users, the algorithms do not typically account for specific challenges like restricted budgets, fixed schedules, or safety concerns that are especially critical for students. Instead, the primary focus remains on the quickest and most profitable matches, which can lead to inefficiencies for those with unique transportation needs.

Moreover, most existing platforms use dynamic pricing strategies to balance demand and supply. While this provides financial incentives for drivers and adjusts pricing during peak hours, it often results in higher costs during high-demand periods. Students, who may have limited budgets, are particularly impacted by surge pricing, making it less accessible and affordable for them during times of critical need, such as early mornings or after classes.

Another common limitation is the lack of personalized safety measures. While platforms typically include user authentication and basic identity verification, these features are often not tailored to specific groups like students. For example, the absence of campus-specific pick-up points, in-app emergency contact systems, or restricted community access can make these platforms less suitable for a younger demographic. Furthermore, most systems do not provide mechanisms to restrict usage to verified student communities, leading to potential safety and trust concerns.

From an environmental perspective, existing platforms have made strides in reducing carbon emissions by encouraging shared rides and fewer vehicles on the road. However, their effectiveness in promoting sustainability is often diluted by their profit-driven nature. Many of these platforms do not actively incentivize or prioritize environmentally conscious travel behaviors, such as selecting electric or hybrid vehicles, or optimizing routes for minimal fuel consumption.

In terms of user experience, these platforms are typically optimized for general usability and scalability. They offer standard features like real-time ride tracking, in-app messaging, and digital payment integration. While these features enhance convenience, they lack customization for specific user groups. Students, for instance, may benefit from features like scheduled rides for fixed class timings, reduced fares for frequent users, or a system designed to minimize delays caused by rigid campus entry and exit points.

Overall, the generic nature of existing ride-sharing systems often fails to cater to the unique requirements of students. They focus on general convenience and scalability, overlooking opportunities for community-specific enhancements. The absence of targeted features, budget considerations, and robust safety measures highlights the need for a student-centric carpooling platform that is more aligned with the specific challenges and opportunities within college campuses.

1.4 PROPOSED SYSTEM

The proposed system is a student-centric carpooling platform designed to address the specific transportation challenges faced by college students. Unlike generic ride-sharing services, this system is tailored to meet the unique needs of a campus environment, providing cost-effective, safe, and eco-friendly transportation solutions. By leveraging advanced algorithms, modern technology, and user-friendly features, the platform ensures optimized ride-matching, dynamic pricing, and enhanced user trust.

A key component of the proposed system is the Hybrid Ridesharing Algorithm (HRA), which integrates K-Means clustering and Genetic Algorithms to optimize the carpooling process. The algorithm begins by grouping users based on their geographic proximity using K-Means clustering. This reduces the complexity of ride-matching by identifying potential carpool groups that share overlapping routes. Next, a Genetic Algorithm is applied to optimize the sequence of pick-up and drop-off points within each group. This two-stage approach ensures minimal travel time and distance, reducing detours and improving the overall efficiency of rides.

To further enhance the user experience, the platform incorporates a dynamic pricing model that adjusts ride costs in real-time based on demand, supply, and other factors. The pricing mechanism combines surge pricing principles with machine learning-based dynamic adjustments. This ensures fairness and affordability for students by balancing user demand with driver availability. For instance, during high-demand periods, prices increase slightly to incentivize more drivers to offer rides, while during low-demand times, prices decrease to attract passengers. Riders also have the flexibility to offer free or paid rides, depending on their preferences.

.

The platform's Progressive Web Application (PWA) design ensures seamless usability across devices, allowing students to access the service from smartphones, tablets, and desktop computers. The PWA offers features like real-time ride tracking via Google Maps API, in-app notifications for ride updates, and messaging capabilities to facilitate communication between riders and passengers. These functionalities make the system intuitive and easy to use, ensuring high adoption rates among students.

Sustainability is a core focus of the proposed system. By promoting shared rides, the platform significantly reduces the number of vehicles on the road, leading to decreased carbon emissions and fuel consumption. The optimized route planning feature further contributes to environmental conservation by minimizing unnecessary travel. This aligns with the growing need for eco-friendly transportation solutions on college campuses, encouraging students to adopt sustainable practices in their daily commutes.

The backend of the platform is powered by robust technologies to ensure scalability and efficiency. A relational database like PostgreSQL is used to manage user profiles, ride details, routes, and payment histories, maintaining data consistency and integrity. Machine learning models, developed using Python libraries such as scikit-learn, handle clustering and optimization tasks, ensuring accurate and efficient ride-matching. The integration of advanced algorithms with modern development frameworks enables the platform to deliver a seamless and reliable experience.

The proposed system addresses the limitations of existing ride-sharing platforms by focusing exclusively on the needs of students. Its innovative approach to ride-matching, dynamic pricing, and user safety ensures a superior transportation experience tailored to campus life. By combining technological advancements with sustainable practices, the platform provides an efficient, cost-effective, and environmentally conscious solution that has the potential to transform transportation within college communities.

CHAPTER 2

2. LITERATURE SURVEY

1. Anas, Gunavathi, and Kirubasri explore a machine learning-based approach for carpooling that uses Natural Language Processing (NLP) to analyze users' social media data to classify personality types. By incorporating XGBoost, their model achieves 68% accuracy in matching users with similar personalities and geographic proximity. This research emphasizes improving user satisfaction in carpooling but highlights challenges like ensuring data privacy and increasing classification accuracy. It addresses the balance between user privacy and algorithmic performance. Their findings underline the importance of leveraging personal traits for better carpooling experiences. However, broader application demands enhanced accuracy and ethical handling of user data. This study sets a foundation for personality-driven carpooling optimization.

2. Jin's study focuses on a passenger-centric taxi carpooling model that prioritizes user preferences, including detour tolerance and route similarity. It integrates a matching algorithm with dynamic pricing to enhance partner selection and optimize routes. This approach reduces waiting times and costs, catering to personalized travel experiences. However, it notes difficulties in capturing diverse user preferences and avoiding mismatches. The model demonstrates the potential of incorporating user-centric features for better operational efficiency. It encourages dynamic and personalized approaches to improve user satisfaction. Challenges include accurate preference capture and ensuring compatibility among passengers.

3. Müller et al. propose a dynamic pricing model for shared mobility systems based on idle time data, increasing vehicle utilization and profitability by 11%. Their model focuses on maximizing operational efficiency in shared

transportation platforms. It emphasizes using real-time analytics to adjust pricing strategies dynamically. The study identifies the need for accurate data and advanced analytical methods to implement this model effectively. While it boosts

profitability, it also demands robust computational resources. This research highlights the role of pricing in balancing supply-demand gaps.It paves the way for optimized financial models in shared mobility ecosystems.

4. Yan et al. investigate dynamic pricing and matching strategies for ride-hailing platforms to balance demand and supply. They focus on minimizing waiting times while maintaining platform reliability. Their approach uses complex algorithms to adjust pricing and wait strategies dynamically. However, longer wait times due to certain optimizations may frustrate riders. The study stresses real-time data usage for effective decision-making in shared mobility services. While achieving operational efficiency, the balance between rider convenience and platform objectives remains a challenge. This research provides insights into adapting to fluctuating market conditions in ride-hailing.

5. Zhou and Roncoli introduce a fairness-aware framework for dynamic ridesharing, emphasizing equitable pricing and passenger matching. Their approach reduces travel time and fuel consumption by using machine learning algorithms. It highlights the importance of fairness in ridesharing, addressing biases in conventional systems. Challenges include ensuring computational efficiency and handling diverse user requirements. This framework provides a sustainable and equitable approach to ridesharing. The study demonstrates how fairness can coexist with efficiency in shared transportation systems. It calls for integrating fairness as a key parameter in future ridesharing platforms.

6. Xu et al. propose TAROT, a privacy-preserving route matching scheme for carpooling services. This system employs advanced cryptographic techniques to secure sensitive user data, such as locations and routes. TAROT reduces computational costs and communication overheads by filtering dissimilar routes early. The study showcases the importance of privacy in route optimization for carpooling. While it ensures data security, scaling the system for broader use remains challenging. TAROT is a significant step toward balancing privacy and efficiency in carpooling platforms.

7. Peng and Zhou focus on optimizing carpooling paths by incorporating real-time traffic data and passenger satisfaction metrics like waiting times. Their model enhances efficiency by dynamically adjusting routes based on real-world conditions. Challenges include seamless integration of real-time data and user acceptance of optimized routes. The study underlines the importance of adaptability in ridesharing systems. By considering passenger comfort and operational feasibility, it paves the way for improved carpooling services. This research emphasizes real-time responsiveness to user needs and environmental conditions. Its practical applications require robust data handling and user engagement strategies.

8. Ramani Bai et al. propose a blockchain-based carpooling application using smart contracts to ensure secure and transparent transactions. Their decentralized model builds trust among users by eliminating intermediaries and enhancing data security. This approach tackles issues like fraudulent activities and data manipulation. Challenges include the high cost of blockchain implementation and ensuring scalability for larger platforms. The study highlights the role of decentralized technologies in improving carpooling systems. Their findings advocate the adoption of blockchain for reliable carpooling services.

9. Hashikami et al. develop a carpooling system in Japan that integrates accident location data to identify safer routes. They classify driver skills and optimize routes to minimize accident risks. This model improves safety while considering travel distance and time. Challenges include user coordination and adapting the system across different cultural contexts. The research highlights the importance of safety in carpooling, especially in high-risk areas. It emphasizes the need for real-time data and driver evaluations. This study contributes to safer and more reliable carpooling environments.

10. Alisoltani et al. propose a space-time clustering-based method to enhance real-time ride-sharing by grouping trips using a shareability function. Their approach reduces computation time while maintaining high-quality trip matches. It focuses on scalability and efficiency in large networks, making it suitable for urban areas with high demand. The clustering technique minimizes the complexity of matching users in real-time scenarios. However, the method requires precise tuning of clustering parameters for optimal performance. The study demonstrates the potential of combining advanced clustering with real-time operations for better shareability. It offers significant insights into improving the scalability of carpooling services.

11. Meshkani and Farooq introduce GMOMatch, a graph-based ride-matching algorithm that optimizes shared trips. The algorithm iteratively matches riders to vehicles, reducing vehicle kilometers traveled (VKT) and traffic congestion. Tested in Toronto, the method significantly improves traffic flow and travel efficiency. However, the approach demands substantial computational resources and parameter optimization. It highlights the benefits of graph-based algorithms in optimizing carpooling services. By prioritizing sustainable shared mobility, the study addresses urban congestion challenges. It underlines the need for efficient algorithms in large-scale ride-sharing networks.

12. Seng et al. review the integration of ridesharing and crowdsourcing for smart cities, focusing on technological paradigms and use cases. They emphasize the potential of crowdsourcing to enhance transportation systems, combining community participation with advanced technology. The study highlights the role of ridesharing in building efficient, citizen-driven mobility solutions. It also identifies challenges like scalability, real-time responsiveness, and user trust. By exploring diverse use cases, it provides a comprehensive view of ridesharing in smart city environments.

13. Shahi et al. conduct a comparative study of pathfinding algorithms for route planning in smart vehicular networks. They identify Contraction Hierarchies (CH) as highly effective for reducing travel time and improving vehicle speed.CH's preprocessing demands are high but result in significant runtime efficiency. The study underlines the importance of robust algorithms in dynamically changing traffic conditions. By optimizing pathfinding, it contributes to the development of responsive, real-time carpooling systems. The research advocates the adoption of CH for large-scale transportation networks. It offers critical insights into balancing preprocessing overhead and operational efficiency.

14. Mitropoulos et al. systematically review ride-sharing platforms, examining user factors, barriers, and system designs. They identify key sociodemographic and regulatory challenges that influence user adoption and satisfaction. The study stresses the importance of addressing these barriers for successful implementation of ride-sharing solutions. It highlights the role of design and regulation in fostering ride-sharing adoption. It offers a roadmap for addressing adoption challenges in shared mobility.

15. Adelé and Dionisio analyze smart carpooling apps like Karos, which use machine learning to predict user mobility patterns. These apps enhance ride

matches by analyzing travel habits and preferences. The study identifies challenges in user understanding of app functionalities, which can impact satisfaction. By leveraging predictive analytics, it aims to improve user engagement and ride-matching efficiency. However, the findings emphasize the need for user education to maximize app benefits. The research bridges the gap between technical capabilities and user adoption in carpooling apps.

16. Bruglieri et al. introduce PoliUniPool, a carpooling system tailored for university communities. The system matches users based on schedules and preferences to minimize travel distances. It promotes sustainability and fosters trust within university networks. Challenges include coordinating schedules and overcoming resistance to carpooling. The study highlights the role of community-specific platforms in enhancing user adoption. By focusing on shared values and trust, it builds a strong foundation for carpooling success. The research demonstrates the benefits of localized solutions for specific user groups.

17. Hasanuddin et al. explore the use of Progressive Web Apps (PWAs) for vehicle tracking systems to combat theft. The PWA provides real-time location tracking and alerts, accessible across multiple platforms. It highlights the benefits of cross-platform compatibility and reduced development costs. The study demonstrates how PWAs enhance accessibility and engagement, even with limited connectivity. The findings emphasize the practicality of PWAs in modern transportation applications. This approach supports seamless and scalable carpooling solutions.

18. Wijaya et al. discuss the advantages of PWAs in vehicle tracking, focusing on offline capabilities and user-friendly interfaces. By enabling caching and real-time tracking, PWAs enhance operational reliability. The study underscores the role of PWAs in improving accessibility and user

engagement in transportation systems. It advocates for the adoption of such technologies in carpooling applications for consistent user access. Challenges include integrating advanced features while maintaining simplicity and scalability. The research highlights the transformative potential of PWAs in transportation innovation. It provides a blueprint for leveraging web technologies in shared mobility.

19. Kumar et al. discuss a smart traffic system that utilizes IoT-based technologies to enhance vehicle safety and road efficiency. The system integrates real-time data from vehicles and road infrastructure to predict and prevent accidents. It also focuses on optimizing traffic flow by reducing congestion and enhancing travel efficiency. The study explores the role of IoT in building intelligent transportation systems that adapt dynamically to changing conditions. Challenges include ensuring data accuracy, scalability, and system reliability in diverse traffic scenarios. The research underscores the importance of smart systems in improving road safety and traffic management. It sets a benchmark for integrating IoT in advanced carpooling solutions.

20. Senthil Pandi et al. focus on improving the classification accuracy of machine learning algorithms using hyperparameter optimization techniques. Their study emphasizes the importance of parameter tuning to enhance model performance. By applying optimization strategies, they achieve significant improvements in algorithm accuracy. This approach is particularly useful in applications like carpooling, where precise user matching is crucial. The research highlights the role of hyperparameter optimization in refining machine learning-based systems. It offers insights into building more robust and efficient models for practical applications. This study serves as a foundation for integrating advanced optimization techniques in carpooling platforms.

CHAPTER 3

3. SYSTEM DESIGN

3.1 GENERAL

3.1.1 SYSTEM FLOW DIAGRAM

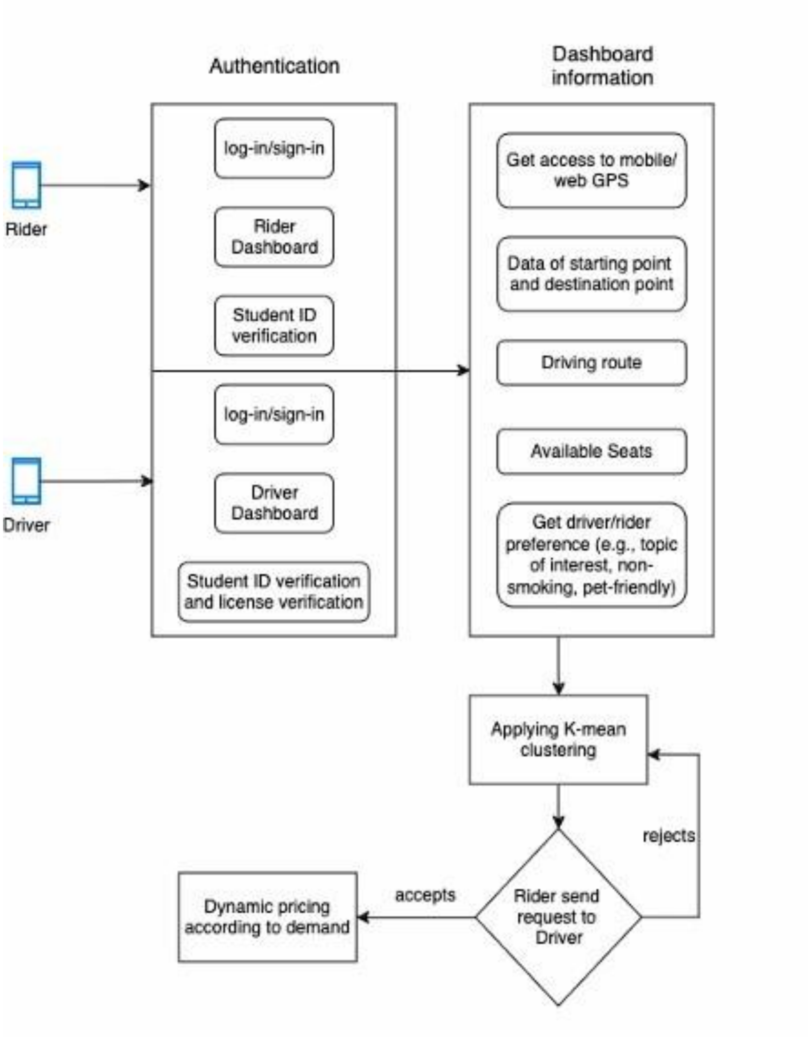

Figure 3.1: System Flow Diagram

The system flow diagram illustrates the workflow of a ride-sharing application. It begins with the authentication process, where riders log in through a rider dashboard with student ID verification, and drivers log in through a driver dashboard with license verification. Once authenticated, users gain access to dashboard information, including mobile/web GPS, starting point and destination data, driving routes, available seats, and rider/driver preferences.

3.1.2 SEQUENCE DIAGRAM

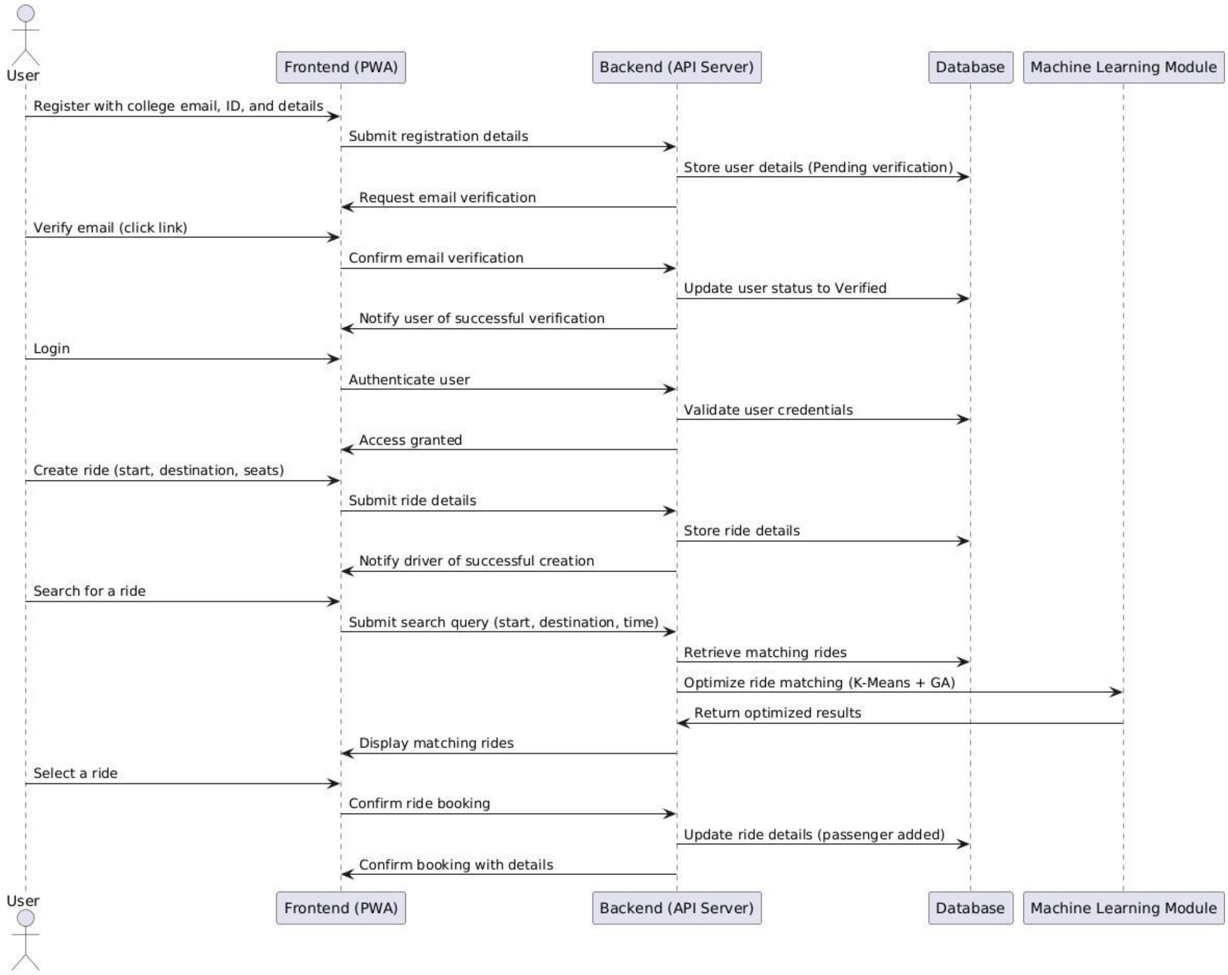

Figure 3.2: Sequence Diagram

The sequence diagram illustrates the workflow of a ride-booking application, starting with user registration, where details are sent to the backend API for validation through the User Service, and the registration status is returned. Once registered, the user creates a ride, and the Ride Service stores the ride details in the database, providing a success message. The user can then search for rides, with the backend querying the Ride Service to retrieve matching options from the database.

3.1.3 CLASS DIAGRAM

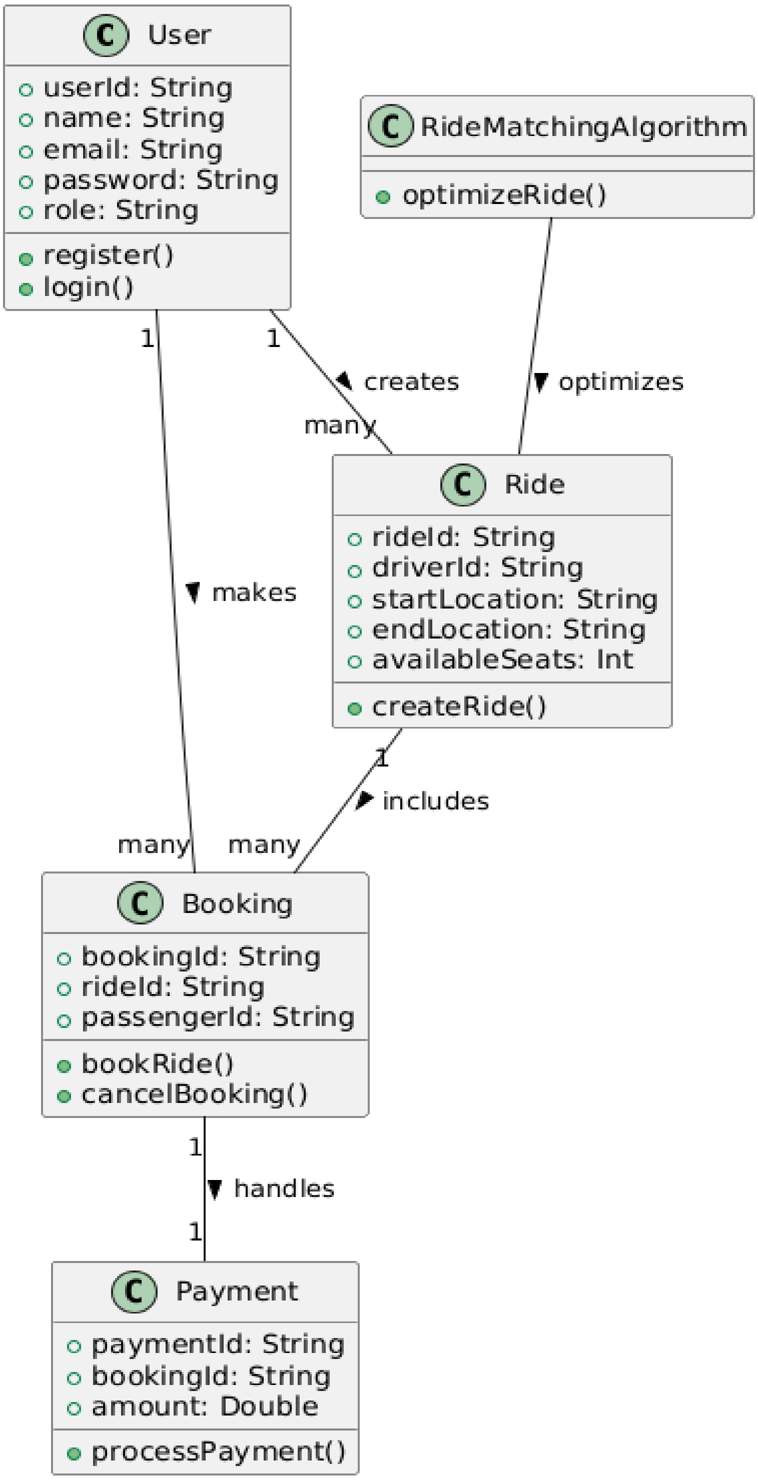

Figure 3.3: Class Diagram

The class diagram represents a ride-sharing system. The system includes a User class, which has attributes like userId, name, email, password, and role, and methods for user registration and login. Users can create rides through the Ride class, which stores details such as rideId, driverId, startLocation, endLocation, and availableSeats. A Booking class connects rides with passengers through bookingId, rideId, and passengerId, and includes methods to book or cancel a ride. Payments for bookings are managed by the Payment class, which contains attributes like paymentId, bookingId, and amount, and the method processPayment(). A RideMatchingAlgorithm class is also present to optimize ride allocation. The relationships illustrate that a user can create multiple rides and bookings, and each booking is associated with one payment.

3.1.4 USE CASE DIAGRAM

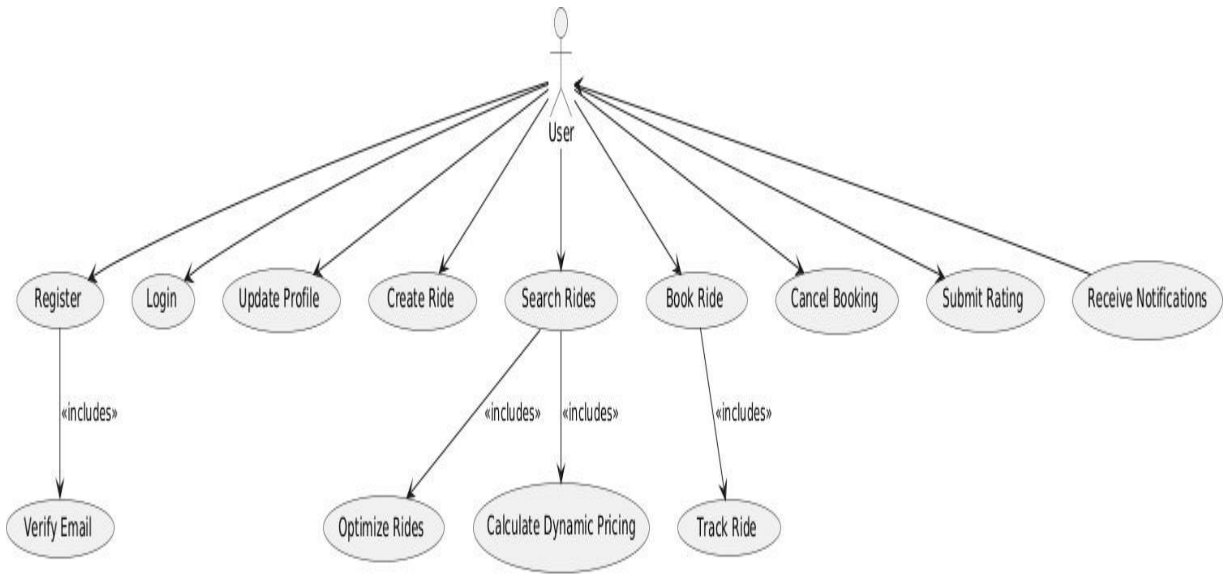

Figure 3.4

The use case diagram illustrates the functionalities of a ride-sharing platform from the perspective of a user. The user can perform several actions, including Registering, Logging in, Updating their Profile, Creating a Ride, Searching for Rides, Booking a Ride, Canceling a Booking, Submitting Ratings, and Receiving Notifications. Certain use cases have additional sub-actions; for example, registering includes Verifying Email, searching for rides includes Optimizing Rides and Calculating Dynamic Pricing, and booking a ride includes Tracking the Ride. This diagram highlights the user-centered nature of the system, focusing on both ride management and user engagement features.

3.1.5 ARCHIETECTURE DIAGRAM

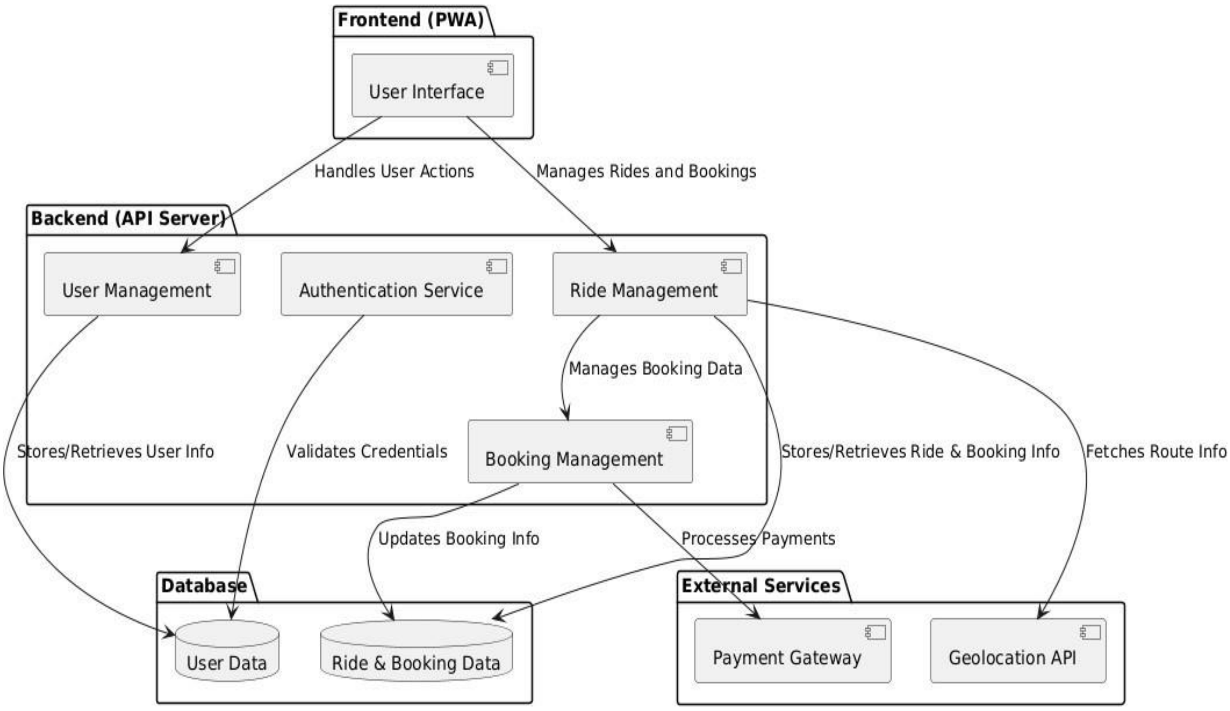

Figure 3.5: Architecture Diagram

The architecture diagram represents a ride-sharing platform structured into three main layers: Frontend (PWA), Backend (API Server), and External Services, with a supporting Database. The Frontend handles user interactions through a user interface, managing actions like ride creation, booking, and updates. The Backend consists of several modules: User Management stores and retrieves user information, Authentication Service validates user credentials, Ride Management handles ride data, and Booking Management oversees bookings. The backend interacts with the Database to store and retrieve user, ride, and booking data. It also integrates with External Services, such as a Payment Gateway for processing payments and a Geolocation API for fetching route information. This architecture ensures a modular, scalable, and secure system for managing rides and bookings.

3.1.6 ACTIVITY DIAGRAM

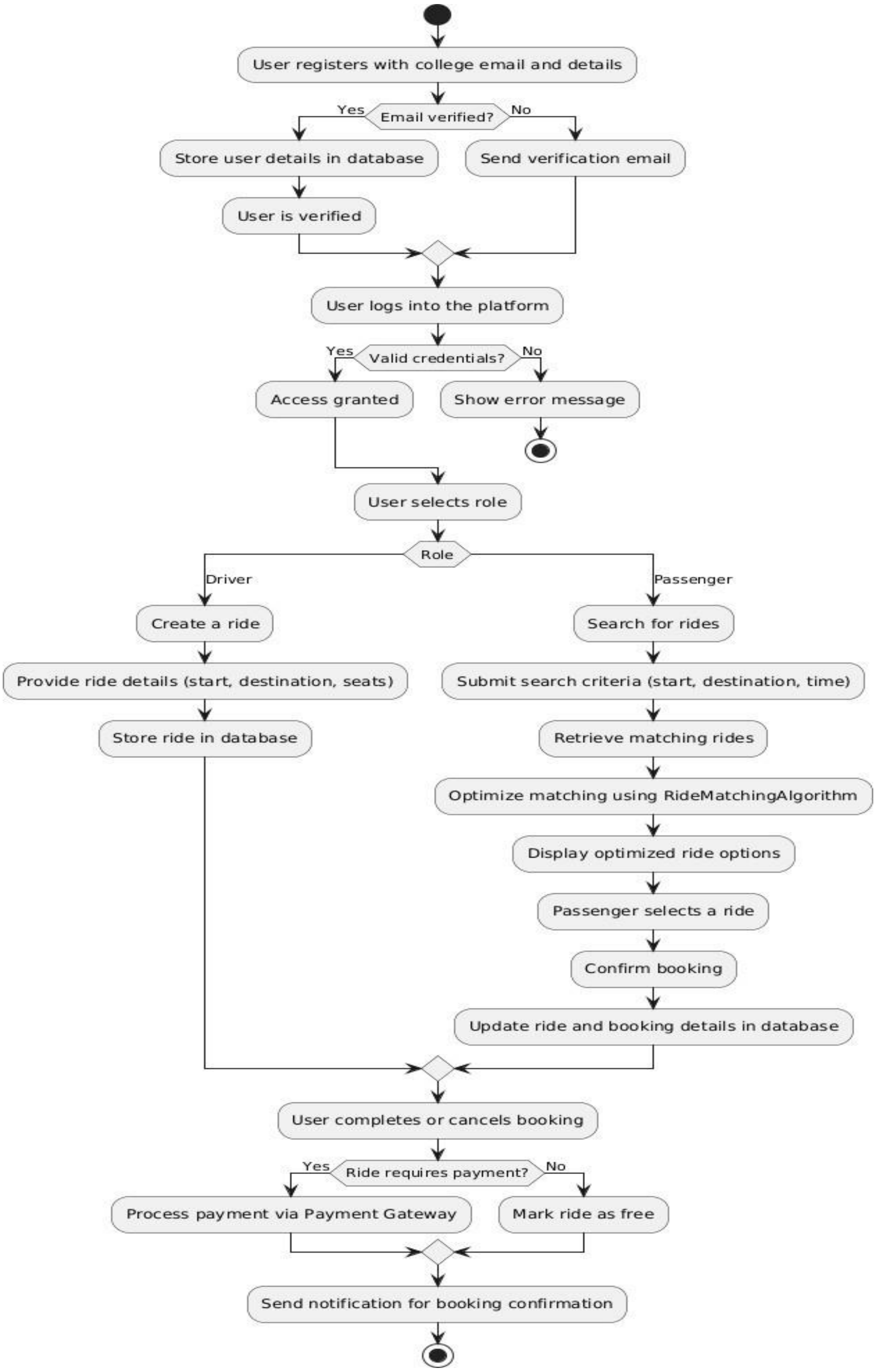

Figure 3.6: Activity Diagram

The activity diagram illustrates the workflow of a college-based ride-sharing platform, beginning with user registration and verification through a college email. Once verified, users log in with valid credentials and choose a role—either as a driver or a passenger. Drivers can create rides by providing details like start location, destination, and available seats, which are stored in the database. Passengers can search for rides by submitting criteria such as start location, destination, and time. The system retrieves matching rides, optimizes options using a ride-matching algorithm, and displays the best matches. Passengers select and confirm rides, updating booking details in the database. Payment is processed if required, and notifications are sent upon successful booking or ride completion, ensuring a smooth and efficient platform experience for both drivers and passengers.

3.1.7 COMPONENT DIAGRAM

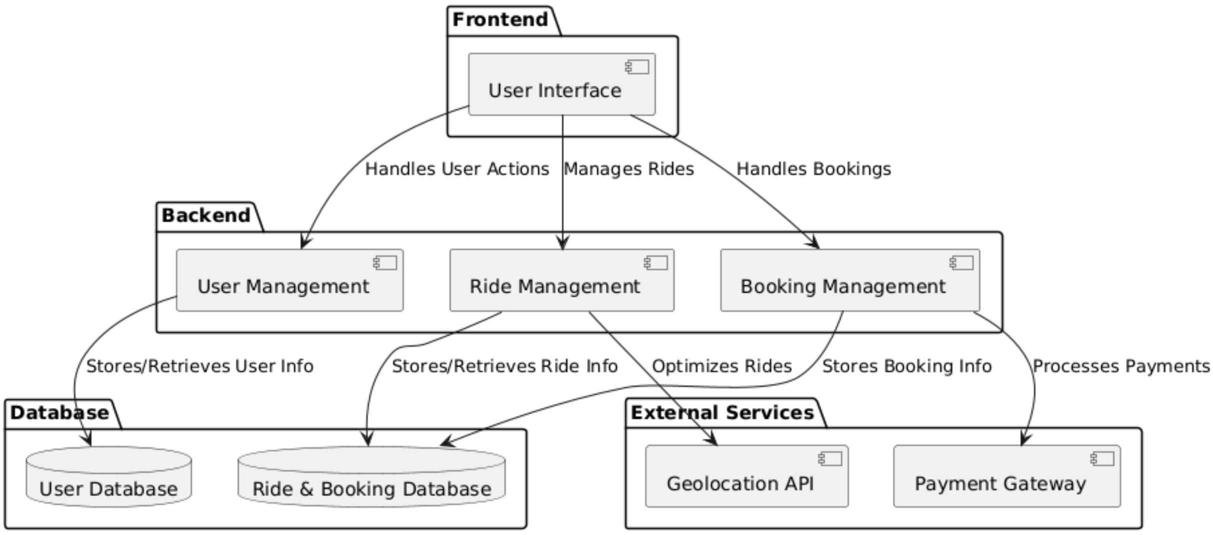

Figure 3.7: Component Diagram

The component diagram represents the architecture of a ride-sharing platform, showcasing its interaction between the frontend, backend, database, and external services. The frontend consists of a user interface that handles user actions, manages rides, and processes bookings. The backend comprises three key modules: User Management, Ride Management, and Booking Management, which are responsible for storing/retrieving user and ride information, optimizing rides, and handling bookings. The database includes a User Database and a Ride & Booking Database for persistent data storage. External services such as a Geolocation API and a Payment Gateway provide geolocation support and payment processing, ensuring seamless integration for ride management

3.1.8 COLLABORATION DIAGRAM

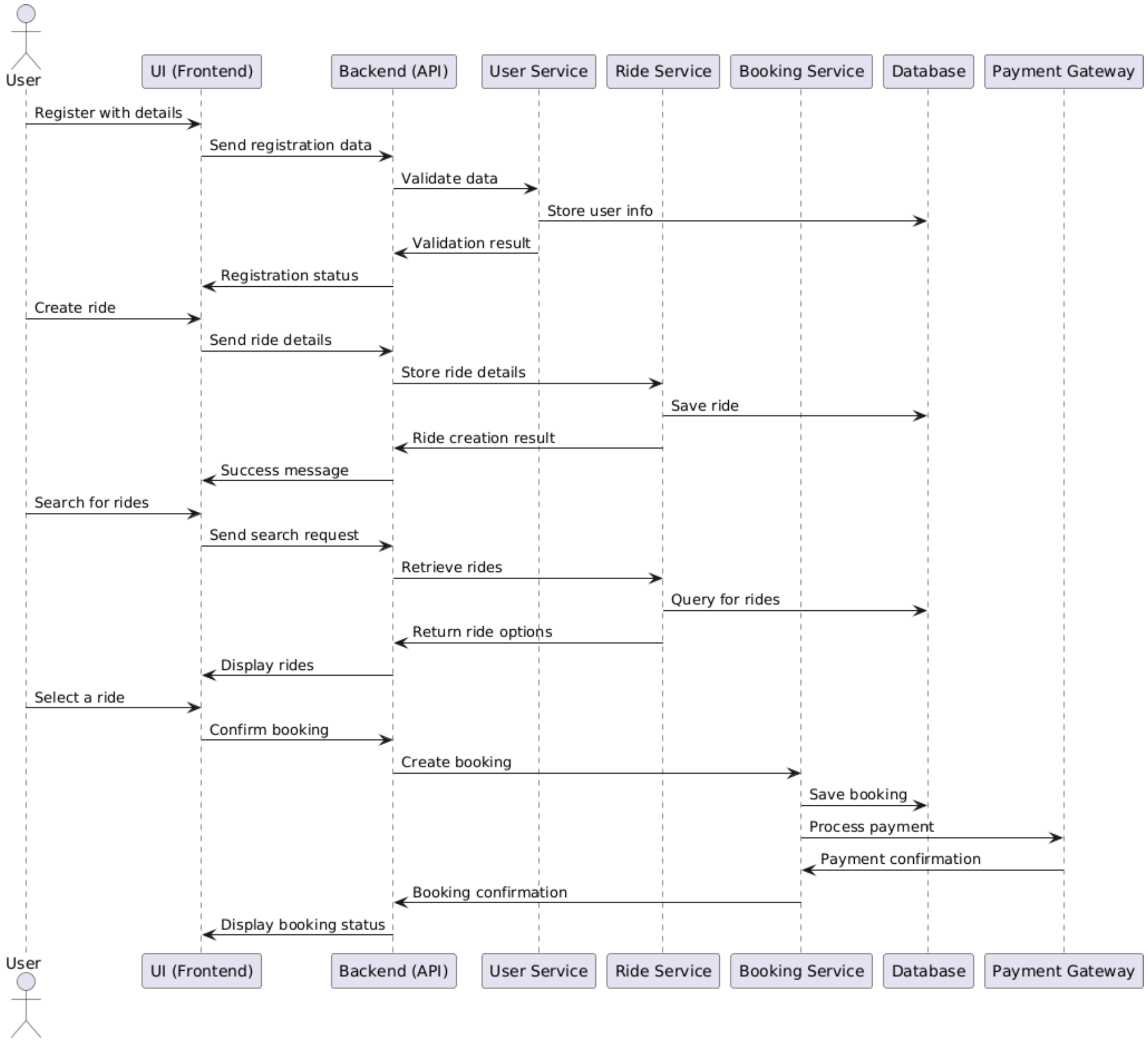

Figure 3.8: Collaboration Diagram

The Collaboration diagram illustrates the workflow of a ride-booking application, starting with user registration, where details are sent to the backend API for validation through the User Service, and the registration status is returned. Once registered, the user creates a ride, and the Ride Service stores the ride details in the database, providing a success message.

CHAPTER 4

PROJECT DESCRIPTION

4.1 METHODOLOGIES:

In this phase of the project, we focus on building the core features of the ride-sharing application, including the **architecture design**, **data collection**, and **preprocessing** steps required for the implementation of the **Hybrid Ridesharing Algorithm (HRA)**. The architecture follows a modular structure, utilizing the **MERN stack** (MongoDB, Express.js, React.js, Node.js) combined with **PWA (Progressive Web App)** capabilities to ensure smooth user experience across devices. The following sections provide detailed methodologies for each part of the system development, from the **architecture design** to **data collection and preprocessing** necessary for implementing the HRA.

1. Architecture Design

The architecture of the ride-sharing application is built on the **MERN stack**, designed to support modularity, scalability, and cross-platform compatibility. The system follows a **client-server model**, where the frontend interacts with the backend via RESTful APIs. Key components of the architecture include:

- **Frontend**:
  - The frontend is built using **React.js**, ensuring a dynamic and responsive user interface. The application is developed as a **PWA** to support offline functionality, allowing users to interact with the app even without an active internet connection. This is particularly useful for drivers in areas with intermittent connectivity.
  - The frontend includes **driver** and **rider dashboards**, both integrated with a **map interface** powered by **Leaflet.js**, which enables route selection and matching features for the users.

- **Backend (Node.js with Express.js)**:
  - The **backend** is built using **Node.js** and **Express.js** to handle API requests and implement business logic. The backend is responsible for processing ride requests, storing ride and user information, and performing the ride matching logic.
  - The **authentication** system uses **JWT (JSON Web Tokens)** for secure user login and session management.

- **Database (MongoDB)**:
  - **MongoDB** is used to store and manage user profiles, ride information, and ride requests. The database structure is flexible, allowing the storage of unstructured data such as user details, ride locations, and dynamic data that may change over time (e.g., available seats).

- **PWA (Progressive Web App) Features**:
  - The **PWA** features ensure that the app is responsive, performs efficiently across devices, and provides a smooth experience even in low connectivity areas. This is achieved through service workers for caching and enabling offline access, along with an installable app that behaves like a native application.

This architecture ensures a seamless integration of frontend and backend services while maintaining scalability, flexibility, and offline accessibility.

Figure 4.1: Splash Screen

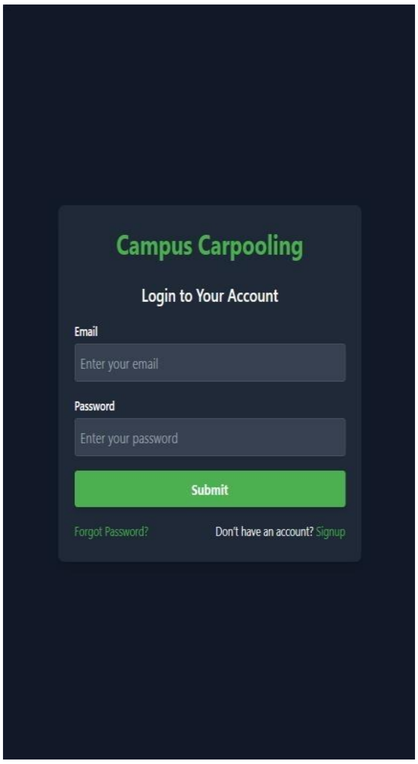

Figure 4.2: Login Page

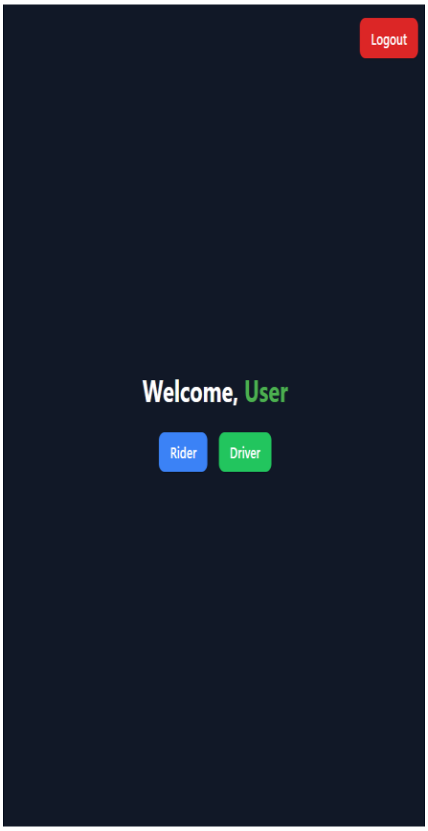

Figure 4.3: Home Page

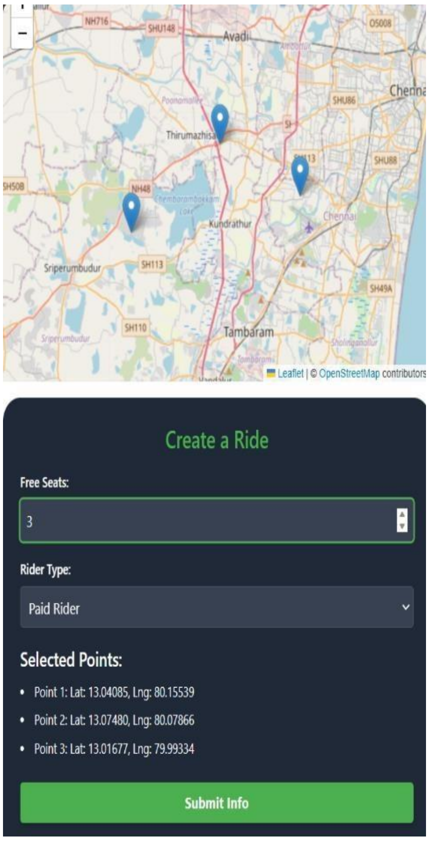

Figure 4.4: Driver Dashboard

2. Data Collection and Preprocessing for the Implementation of HRA

The **data collection and preprocessing** phase is a critical step for the effective implementation of the **Hybrid Ridesharing Algorithm (HRA)**, which combines **k-means clustering** and **genetic algorithm (GA)** for optimal ride matching. Here are the key steps involved in this phase:

1. **Data Collection**
   - **Synthetic Data Generation**
     - Since real-world data may be unavailable at this stage, **synthetic data** is generated to simulate realistic ridesharing scenarios. The synthetic data includes:
       - **Driver Data**: Each driver's starting and ending points (latitude and longitude), available seats, and route distances.
       - **Rider Data**: Each rider's pickup and drop-off locations (latitude and longitude).
       - **Distance Matrix**: The distances between each driver's route and each rider's pickup/drop-off point, which is essential for optimizing matches.
   - The data is generated using simple **randomization techniques** for location points within predefined geographic bounds (e.g., a city area) to simulate various ride and driver configurations.

2. **Data Preprocessing**
   - **Data Validation and Cleaning**
     - The collected data is validated to ensure consistency, such as ensuring no negative seat counts, verifying that the coordinates lie within the defined region, and removing any outliers (e.g., locations outside the city bounds).
     - The data is **normalized** to standardize all inputs (e.g.,

latitude/longitude values) and ensure compatibility with the k-means clustering algorithm.

- **Distance Calculation**:
  - The **distance matrix** is calculated based on the **Euclidean distance** between each rider's and driver's locations. This is essential for both the **k-means** clustering (to group riders and drivers based on proximity) and the **genetic algorithm** (to optimize the travel distances).
  - In future iterations, this can be replaced with **real-world distance calculations** (e.g., using Google Maps or OpenRouteService) to better reflect travel times and route feasibility.

3. **Feature Engineering**
   - **Clustering Features**:
     - The **start and end points** of both drivers and riders are used to calculate proximity, forming the basis for **k-means clustering**.
   - **Additional Features**:
     - Calculated features such as **route distance** for drivers and **ride distance** for riders enhance the algorithm's ability to prioritize matches.

RAW DATA


```
{
  "drivers": [
    {
      "driver_id": "driver_1",
      "start_point": { "lat": 12.9507, "lng": 80.2276 },
      "end_point": { "lat": 12.9724, "lng": 80.2453 },
      "seats_available": 3
    },
    {
      "driver_id": "driver_2",
      "start_point": { "lat": 12.9205, "lng": 80.2150 },
      "end_point": { "lat": 13.0012, "lng": 80.2798 },
      "seats_available": 2
    }
  ],
  "riders": [
    {
      "rider_id": "rider_1",
      "pickup_point": { "lat": 12.9654, "lng": 80.2407 },
      "dropoff_point": { "lat": 12.9724, "lng": 80.2453 } // Matches driver_1's
destination
    },
    {
      "rider_id": "rider_2",
      "pickup_point": { "lat": 12.9386, "lng": 80.2234 },
      "dropoff_point": { "lat": 13.0012, "lng": 80.2798 } // Matches driver_2's
destination
    }
  ],
  "distance_matrix": [
    {
      "driver_id": "driver_1",
      "rider_id": "rider_1",
      "pickup_distance": 1.2,
      "dropoff_distance": 0
    },
    {
      "driver_id": "driver_2",
      "rider_id": "rider_2",
      "pickup_distance": 0.8,
      "dropoff_distance": 0
    }
  ]
```

}

PREPROCESSED DATA


```
{
 "drivers": [
  {
   "driver_id": "driver_1",
   "start_lat": 12.9507,
   "start_lng": 80.2276,
   "end_lat": 12.9724,
   "end_lng": 80.2453,
   "seats_available": 3,
   "route_distance": 3.0  // Precomputed distance between start and end points
  },
  {
   "driver_id": "driver_2",
   "start_lat": 12.9205,
   "start_lng": 80.2150,
   "end_lat": 13.0012,
   "end_lng": 80.2798,
   "seats_available": 2,
   "route_distance": 5.0
  }
 ],
 "riders": [
  {
   "rider_id": "rider_1",
   "pickup_lat": 12.9654,
   "pickup_lng": 80.2407,
   "dropoff_lat": 12.9724,
   "dropoff_lng": 80.2453,
   "ride_distance": 1.2  // Simplified as pickup distance (dropoff distance is 0)
  },
  {
   "rider_id": "rider_2",
   "pickup_lat": 12.9386,
   "pickup_lng": 80.2234,
   "dropoff_lat": 13.0012,
   "dropoff_lng": 80.2798,
   "ride_distance": 0.8
  }
 ],
```

```
  "distance_matrix": [
    {
      "driver_id": "driver_1",
      "rider_id": "rider_1",
      "pickup_distance": 1.2,
      "dropoff_distance": 0,
      "total_distance": 1.2  // Equal to pickup distance
    },
    {
      "driver_id": "driver_2",
      "rider_id": "rider_2",
      "pickup_distance": 0.8,
      "dropoff_distance": 0,
      "total_distance": 0.8  // Equal to pickup distance
    }
  ]
}
```


4.1.1 RESULTS AND DISCUSSION

In Phase 1 of the project, the primary focus was on setting up the foundational elements for the ride-sharing application, particularly the data collection and preprocessing required to implement the Hybrid Ridesharing Algorithm (HRA). This phase involved gathering synthetic data for drivers and riders, and processing it into a structured format suitable for the HRA.

1. Results

1. **Data Collection**
   - Synthetic data was generated for a small but representative sample of drivers and riders. This data included the key attributes such as:
     - Driver route (starting and ending points).
     - Rider locations (pickup and drop-off points).
     - Seat availability for drivers.
   - A distance matrix was computed based on simple Euclidean distance between pickup and drop-off points for both drivers and riders.

2. **Preprocessing**
   - The preprocessing step successfully transformed the raw data into a structured format, making it easier to feed into the Hybrid Ridesharing Algorithm (HRA) in the future.
   - Features like pickup distance, drop-off distance, and total distance were derived from the raw data, preparing the dataset for the next phase of the project.

3. **Data Format**
   - The structured data format, including both raw data and preprocessed data, was finalized and is now ready to be used in subsequent algorithm implementations. This ensures that all necessary features (e.g., distance, seat availability) are available for ride matching once the algorithm is implemented.

2. Discussion

1. **Importance of Data Preprocessing**
   - Data preprocessing is a critical step in the project because it ensures that the input data is clean, consistent, and ready for use in the HRA.
   - By transforming the raw data into a structured format with relevant features (e.g., total distance, pickup distance, ride distance), we have ensured that the HRA will be able to efficiently compute the best matches for riders and drivers.

2. **Impact of Removing Ride Preferences**
   - The decision to exclude ride preferences (such as "paid" and "free" rides) in Phase 1 simplified the data collection and preprocessing steps. It allowed us to focus on core features such as proximity and seat availability, which are central to the ride matching process.

- In future phases, ride preferences can be reintroduced to make the matching process more sophisticated, catering to different user needs (e.g., matching drivers with paid riders and ensuring availability for riders who prefer free rides).

3. **Preparation for the Hybrid Ridesharing Algorithm**
   - The preprocessed data now contains all necessary features (e.g., distances, available seats) for the HRA. The next step will involve clustering the data (using k-means) and optimizing matches using the genetic algorithm.
   - This foundational work lays the groundwork for Phase 2, where the actual matching algorithm will be developed and tested.

CHAPTER 5

CONCLUSION AND WORKSPACE

5.1 CONCLUSION

Phase 1 of the project focused on laying the foundation for the ride-sharing application by emphasizing data collection and preprocessing tasks. Instead of implementing the Hybrid Ridesharing Algorithm (HRA), this phase concentrated on generating synthetic data for drivers and riders, organizing it into a structured format to support future algorithm execution. Key features such as pickup distances, drop-off distances, and route distances were included to enable efficient ride matching. The MERN stack was chosen for its flexibility and scalability, ensuring seamless development of backend and frontend components. MongoDB was selected to handle unstructured data, allowing dynamic storage and retrieval of ride-related information. This preparation ensured a strong foundation for the application’s upcoming phases.

Phase 2 will utilize the preprocessed dataset to implement the Hybrid Ridesharing Algorithm. Advanced algorithms like k-means clustering and genetic algorithms will be incorporated to optimize driver-rider matching based on geographical proximity and route similarity. Additionally, the algorithm will be enhanced to account for dynamic factors such as ride preferences and real-time traffic conditions, improving efficiency. The goal of this phase is to leverage the prepared data to drive system functionality and optimize the overall ride-sharing process. By integrating these sophisticated techniques, the system aims to deliver precise and reliable matches while accommodating user-specific needs.

In conclusion, Phase 1 successfully set the stage for the application’s development by preparing a robust, preprocessed dataset and ensuring the platform's scalability and flexibility. With the foundational tasks completed, Phase 2 will build upon this groundwork to enhance system functionality and performance. By focusing on advanced algorithms and user-centric improvements, the project aims to deliver an efficient, dynamic, and reliable ride-sharing solution tailored to modern needs. This phased approach ensures a seamless transition from preparation to execution, driving the application toward its intended goals.

5.2 FOR PHASE 2

In Phase 2 of the project, the focus will shift to implementing the **Hybrid Ridesharing Algorithm (HRA)**, which is the core component for optimizing ride matching between drivers and riders. The HRA will combine **k-means clustering** to group nearby riders and drivers and **genetic algorithms (GA)** to optimize the allocation of rides based on factors like seat availability, route compatibility, and travel distance. The algorithm will dynamically calculate the most efficient ride-sharing matches by considering proximity and minimizing total travel distances, ensuring an optimized, cost-effective, and convenient ride for both drivers and riders. Additionally, real-time data such as **traffic conditions** and **route deviations** will be incorporated into the algorithm to further enhance its accuracy and responsiveness.

Along with the **HRA**, Phase 2 will also focus on the implementation of **dynamic pricing simulation**. This feature will enable the app to adjust pricing in real-time based on various factors such as demand, ride distance, time of day, and traffic conditions. By using machine learning models or rule-based systems, dynamic pricing will allow the app to optimize revenue for drivers while maintaining fairness for riders. The simulation will include testing different pricing models, including **surge pricing** and **discounts for long rides**, to determine the most effective strategy for balancing supply and demand. These features will significantly improve the app’s functionality and user experience, making it more competitive and scalable in the real-world market.

In Phase 2, in addition to implementing the **Hybrid Ridesharing Algorithm (HRA)** and **dynamic pricing simulation**, the app will be enhanced with full **PWA (Progressive Web App)** features. This includes **offline capabilities** using **service workers** to cache critical data like ride details and driver availability, ensuring the app functions without an internet connection. The app will also be **installable** on users' home screens, offering a native app-like experience on both mobile and desktop. **Push notifications** will be integrated to provide real-time updates on ride statuses, driver availability, and pricing changes. These features will ensure a seamless, responsive experience across devices, increasing user engagement and accessibility.

REFERENCES

1. Anas, Mohd, C. Gunavathi, and G. Kirubasri. "Machine learning based personality classification for carpooling application." 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS). IEEE, 2023.

2. Jin, Dian. "Research on Taxi-Carpooling Matching Schemes Considering Passenger Personalized Needs." 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE). IEEE, 2024.

3. Müller, Christian, et al. "Dynamic pricing for shared mobility systems based on idle time data." OR Spectrum 46.2 (2024): 411-444.

4. Yan, Chiwei, et al. "Dynamic pricing and matching in ride-hailing platforms." Naval Research Logistics (NRL) 67.8 (2020): 705-724.

5. Zhou, Ze, and Claudio Roncoli. "A fairness-aware joint pricing and matching framework for dynamic ridesharing." 2023 8th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). IEEE, 2023.

6. Xu, Qi, et al. "An efficient and privacy-preserving route matching scheme for carpooling services." IEEE Internet of Things Journal 9.20 (2022): 19890-19902.

7. Peng, Jiarun, and Hongmei Zhou. "Matching path optimization of carpooling mode for private cars considering real-time traffic conditions and passenger satisfaction." 2023 7th International Conference on Transportation Information and Safety (ICTIS). IEEE, 2023.

8. Sudhir, Pranav, et al. "Smart Contract Based Carpooling Application for Secure and Efficient Ride Sharing." 2023 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE). IEEE, 2023.

9. Hashikami, Hidenobu, et al. "Safe Route Carpooling to Avoid Accident Locations and Small-Scale Proof of Concept in Japan." IEEE Transactions on Systems, Man, and Cybernetics: Systems 53.7 (2023): 4239-4250.

10. Zhu, Congcong, et al. "Time-optimal and privacy preserving route planning for carpool policy." World Wide Web 25.3 (2022): 1151-1168.

11. Alisoltani, Negin, et al. "Space-time clustering-based method to optimize shareability in real-time ride-sharing." Plos one 17.1 (2022): e0262499.

12. Seng, Kah Phooi, et al. "Ridesharing and crowdsourcing for smart cities: technologies, paradigms and use cases." IEEE Access 11 (2023): 18038-18081.

13. Shahi, Gurpreet Singh, Ranbir Singh Batth, and Simon Egerton. "A comparative study on efficient path finding algorithms for route planning in smart vehicular networks." International Journal of Computer Networks and Applications 7.5 (2020): 157-166.

14. Mitropoulos, Lambros, Annie Kortsari, and Georgia Ayfantopoulou. "A systematic literature review of ride-sharing platforms, user factors and barriers." European Transport Research Review 13 (2021): 1-22.

15. Adelé, Sonia, and Corinne Dionisio. "Learning from the real practices of users of a smart carpooling app." European Transport Research Review 12 (2020): 1-14.

16. Bruglieri, Maurizio, et al. "PoliUniPool: a carpooling system for universities." Procedia-Social and Behavioral Sciences 20 (2011): 558-567.

17. Acharya, Rhea, Jessica Chen, and Helen Xiao. "Uber Stable: Formulating the Rideshare System as a Stable Matching Problem." arXiv preprint arXiv:2403.13083 (2024).

18. Hasanuddin, Muhammad Ogin, et al. "Implementation of Progressive Web App-Based Applications for Vehicle Tracking Systems." 2022 12th International Conference on System Engineering and Technology (ICSET). IEEE, 2022.

19. Kumar, P., Vinodh Kumar, S., Priya, L. (2023). Smart and Safety Traffic System for the Vehicles on the Road. In: Choudrie, J., Mahalle, P., Perumal, T., Joshi, A. (eds) IOT with Smart Systems. Smart Innovation, Systems and Technologies, vol 312. Springer, Singapore. https://doi.org/ 10.1007/978-981-19-3575-6_51.

20. S. Senthil Pandi, V. R. Chiranjeevi, K. T and Kumar P, Improvement of Classification Accuracy in Machine Learning Algorithm by Hyper Parameter Optimization(2023), 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), Chennai, India, 2023, pp. 1-5, doi: 10.1109/RMKMATE59243.2023.10369177

APPENDIX

IMPLEMENTATION CODE

App.js

```
import React, { useState,useEffect } from "react";
import { Route, Routes } from "react-router-dom";
import { Login, Signup, Driver1, Rider } from "./pages";
import SplashScreen from "./components/SplashScreen";
import Home from "./pages/Home";
function App() {
  const [isSplashVisible, setIsSplashVisible] = useState(true);
  useEffect(() => {
    // Set a timeout to hide the splash screen after 3 seconds
    const timer = setTimeout(() => {
    setIsSplashVisible(false);
    }, 3000);
    return () => clearTimeout(timer); // Clear the timeout on unmount
  }, []);
  // Render the splash screen if it is still visible
  if (isSplashVisible) {
    return <SplashScreen />;
  }
  return (
    <div className="App">
      <Routes>
        <Route path="/" element={<Home />} />
```

```
        <Route path="/login" element={<Login />} />
        <Route path="/signup" element={<Signup />} />
        <Route path="/driver" element={<Driver1 />} />
        <Route path="/rider" element={<Rider />} />
      </Routes>
    </div>
  );
}
export default App;
```

SplashScreen.js

```
import React from "react";
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
import { faCar } from "@fortawesome/free-solid-svg-icons";
const SplashScreen = () => {
return (
<div className="min-h-screen flex flex-col items-center justify-center bg-
primary text-white">
  {/* FontAwesome Icon */}
  <FontAwesomeIcon icon={faCar} className="text-6xl mb-4 animate-
bounce" />
  <h1 className="text-4xl font-bold animate-fade">Welcome to
ShareRide</h1>
</div>
);
}
```

export default SplashScreen;

Authroute.js

```
const { Signup, Login } = require("../Controllers/AuthController");
const { userVerification } = require("../Middlewares/AuthMiddleware")
const { addDriverData, updateDriverData } =
require("../Controllers/DriverController");
const router = require("express").Router();
router.post("/signup", Signup);
router.post("/login", Login);
router.post("/", userVerification);
// Driver Data Routes
router.post("/add-driver", addDriverData); // Route to add driver data
router.put("/update-driver", updateDriverData); // Route to update driver data
module.exports = router;
```

AuthController.js

```
const User = require("../Models/UserModel");
const { createSecretToken } = require("../util/SecretToken");
const bcrypt = require("bcryptjs");
// sign-up function
module.exports.Signup = async (req, res, next) => {
  try {
    //getting user data from body
    const { email, password, username, createdAt } = req.body;
    //check if email already exist
    const existingUser = await User.findOne({ email });
    if (existingUser) {
      return res.json({ message: "User already exists" });
```

```
    }
    const user = await User.create({ email, password, username, createdAt });
    //using mongodb unique id
    const token = createSecretToken(user._id);
    res.cookie("token", token, {
    withCredentials: true,
      httpOnly: false,
    });
    res
      .status(201)
      .json({ message: "User signed in successfully", success: true, user });
    next();
  } catch (error) {
    console.error(error);
  }
};
module.exports.Login = async (req, res, next) => {
    try {
      const { email, password } = req.body;
      if(!email || !password ){
        return res.json({message:'All fields are required'})
      }
      //getting user info using email
      const user = await User.findOne({ email });
      if(!user){
        return res.json({message:'Incorrect password or email' })
```

```
      }
      //un-hashing password to check
      const auth = await bcrypt.compare(password,user.password)
      if (!auth) {
        return res.json({message:'Incorrect password or email' })
      }
      //creating session for login
       const token = createSecretToken(user._id);
       res.cookie("token", token, {
       withCredentials: true,
         httpOnly: false,
       });
       res.status(201).json({ message: "User logged in successfully", success: true
});
       next()
    } catch (error) {
      console.error(error);
    }
  };
```

DriverModel.js

```
const mongoose = require("mongoose");
const driverSchema = new mongoose.Schema({
  driverId: {
    type: String,
    required: [true, "Driver ID is required"],
    unique: true,
```

```
    },
    points: [
     {
       lat: {
         type: Number,
         required: [true, "Latitude is required"],
       },
       lng: {
         type: Number,
         required: [true, "Longitude is required"],
       },
},
    ],
    freeSeats: {
     type: Number,
     required: [true, "Number of free seats is required"],
     min: [0, "Free seats cannot be less than 0"],
    },
    riderType: {
     type: String,
     enum: ['paid', 'free'],
     required: [true, "Rider type is required"],
    },
    createdAt: {
     type: Date,
     default: new Date(),
```

```
  },
  updatedAt: {
    type: Date,
    default: new Date(),
  },
});
driverSchema.pre("save", function (next) {
  this.updatedAt = new Date();
  next();
});
module.exports = mongoose.model("Driver", driverSchema);
```

SyntheticData.py

```
import random
import numpy as np
import pandas as pd
from geopy.distance import geodesic
# Function to generate synthetic driver data
def generate_driver_data(num_drivers):
    drivers = []
    for i in range(num_drivers):
        start_lat = random.uniform(12.9, 13.1)
        start_lng = random.uniform(80.1, 80.3)
        end_lat = random.uniform(12.9, 13.1)
        end_lng = random.uniform(80.2, 80.4)
        seats_available = random.randint(1, 5)
        drivers.append({
            'driver_id': f'driver_{i+1}',
```

```
            'start_point': {'lat': start_lat, 'lng': start_lng},
            'end_point': {'lat': end_lat, 'lng': end_lng},
            'seats_available': seats_available
        })
    return drivers
# Function to generate synthetic rider data
def generate_rider_data(num_riders):
    riders = []
    for i in range(num_riders):
        pickup_lat = random.uniform(12.9, 13.1)
        pickup_lng = random.uniform(80.1, 80.3)
        dropoff_lat = random.uniform(12.9, 13.1)
        dropoff_lng = random.uniform(80.2, 80.4)
        riders.append({
            'rider_id': f'rider_{i+1}',
            'pickup_point': {'lat': pickup_lat, 'lng': pickup_lng},
            'dropoff_point': {'lat': dropoff_lat, 'lng': dropoff_lng}
        })
    return riders
# Function to calculate distance between two points
def calculate_distance(point1, point2):
    return geodesic((point1['lat'], point1['lng']), (point2['lat'], point2['lng'])).km
# Generate synthetic data for 3 drivers and 3 riders
drivers = generate_driver_data(3)
riders = generate_rider_data(3)
# Preprocessing the data
distance_matrix = []
for driver in drivers:
    for rider in riders:
        pickup_distance = calculate_distance(driver['start_point'],
```

```
rider['pickup_point'])
        dropoff_distance = calculate_distance(driver['end_point'],
rider['dropoff_point'])
        total_distance = pickup_distance + dropoff_distance
        distance_matrix.append({
            'driver_id': driver['driver_id'],
            'rider_id': rider['rider_id'],
            'pickup_distance': pickup_distance,
            'dropoff_distance': dropoff_distance,
            'total_distance': total_distance
        })
# Preprocessed data structure (example output)
preprocessed_data = {
    "drivers": [
        {
            "driver_id": driver['driver_id'],



            "start_lat": driver['start_point']['lat'],
            "start_lng": driver['start_point']['lng'],
            "end_lat": driver['end_point']['lat'],

            "end_lng": driver['end_point']['lng'],
            "seats_available": driver['seats_available'],
            "route_distance": calculate_distance(driver['start_point'], driver['end_point'])
        } for driver in drivers
    ],
    "riders": [
        {
            "rider_id": rider['rider_id'],
```

```
            "pickup_lat": rider['pickup_point']['lat'],
            "pickup_lng": rider['pickup_point']['lng'],
            "dropoff_lat": rider['dropoff_point']['lat'],
            "dropoff_lng": rider['dropoff_point']['lng'],
            "ride_distance": calculate_distance(rider['pickup_point'],
rider['dropoff_point'])
        } for rider in riders
    ],
    "distance_matrix": distance_matrix
}
# Display the preprocessed data
import json
print(json.dumps(preprocessed_data, indent=4))
```

PAPER PUBLICATION STATUS PHASE I

TITLE:	Campus carpooling: optimized ridesharing for students using hybrid ridesharing algorithm
AUTHOR:	Ms. Dharshini B S, Logeshwaran Elumalai, Mohamed Aadhil A
CONFERENCE:	2^{nd} INTERNATIONAL CONFERENCE ON EMERGING RESEARCH IN COMPUTATIONAL SCIENCE – 2024 **(ICERCS-2024)**
MODE OF CONFERENCE:	HYBRID MODE
STATUS:	SUBMITTED

Submission Summary

Conference Name

INTERNATIONAL CONFERENCE ON EMERGING RESEARCH IN COMPUTATIONAL SCIENCE - 2024

Paper ID

1644

Paper Title

Campus Carpooling: Optimised Ridesharing for Students using Hybrid Ridesharing Algorithm(HRA)

Abstract

A student-centric carpooling service tailored to address the unique transportation needs of college students. The platform allows users to post their starting locations, available seats, and routes, while enabling other users to select their desired pick-up points. The service employs a machine learning algorithm to optimize the matching process between riders and passengers, ensuring efficient and convenient travel arrangements. A unique feature of the system is dynamic pricing, which adjusts based on demand and supply factors, offering flexibility in payment options for rides, which can be either paid or free, depending on the rider's choice. The platform incorporates user registration and authentication to maintain security and trust within the community, with profiles for both riders and passengers that include ratings and reviews to establish a reliable user base. The machine learning component utilizes clustering and optimization algorithms to reduce total travel time and distance, thereby enhancing the overall efficiency of the service. Additional features such as an integrated payment gateway for paid rides, identity verification, emergency contact information, and ride tracking are included to ensure safety. In-app messaging and notifications provide seamless communication between users. An admin panel is also provided for the management of users, rides, and payments, ensuring smooth operations and quick resolution of issues. The project aims to develop a sustainable, efficient, and cost-effective carpooling system tailored for college students, promoting

environmentally friendly transportation solutions on campuses.

Created

22/11/2024, 7:52:11 pm

Last Modified

22/11/2024, 7:52:11 pm

Authors

Dharshini B S (Rajalakshmi Engineering College) <dharshini.bs@rajalakshmi.edu.in>
Logeshwaran Elumalai (Rajalakshmi Engineering College) <logeshwaran.elumalai@outlook.com>
Mohamed Aadhil A (Rajalakshmi Engineering College) <mohamed_aadhil@outlook.com>

Submission Files

paper.pdf (510.2 Kb, 22/11/2024, 7:49:24 pm)

REPORT.pdf

ORIGINALITY REPORT

11%	8%	3%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Illinois at Urbana-Champaign Student Paper	6%
2	fastercapital.com Internet Source	1%
3	Submitted to University of Greenwich Student Paper	<1%
4	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024 Publication	<1%
5	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
6	krishikosh.egranth.ac.in Internet Source	<1%
7	Submitted to University of Dayton Student Paper	<1%

8	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security", CRC Press, 2023 Publication	<1%
9	Submitted to Beirut Arab University Student Paper	<1%
10	Submitted to University of Westminster Student Paper	<1%
11	www.ijraset.com Internet Source	<1%
12	Lulu Han, Weiqi Luo, Anjia Yang, Yandong Zheng, Rongxing Lu, Junzuo Lai, Yudan Cheng. "Fully privacy-preserving location recommendation in outsourced environments", Ad Hoc Networks, 2023 Publication	<1%
13	d-nb.info Internet Source	<1%
14	devpost.com Internet Source	<1%
15	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
16	journals.plos.org Internet Source	<1%

17	kipdf.com Internet Source	<1%
18	"Proceedings of Fifth Doctoral Symposium on Computational Intelligence", Springer Science and Business Media LLC, 2024 Publication	<1%
19	dokumen.pub Internet Source	<1%
20	eprint.iacr.org Internet Source	<1%
21	openreview.net Internet Source	<1%
22	www.coursehero.com Internet Source	<1%
23	www.zgglkx.com Internet Source	<1%
24	scholar.xjtlu.edu.cn Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

Campus Carpooling: Optimised Ridesharing for Students using Hybrid Ridesharing Algorithm(HRA)

Dharshini B S
Department of CSE
Rajalakshmi Engineering College
Chennai, India
dharshini.bs@rajalakshmi.edu.in

Logeshwaran Elumalai
Department of CSE
Rajalakshmi Engineering College
Chennai, India
logeshwaran.elumalai@outlook.com

Mohamed Aadhil A
Department of CSE
Rajalakshmi Engineering College
Chennai, India
mohamed_aadhil@outlook.com

***Abstract* -** A student-centric carpooling service tailored to address the unique transportation needs of college students. The platform allows users to post their starting locations, available seats, and routes, while enabling other users to select their desired pick-up points. The service employs a machine learning algorithm to optimize the matching process between riders and passengers, ensuring efficient and convenient travel arrangements. A unique feature of the system is dynamic pricing, which adjusts based on demand and supply factors, offering flexibility in payment options for rides, which can be either paid or free, depending on the rider's choice. The platform incorporates user registration and authentication to maintain security and trust within the community, with profiles for both riders and passengers that include ratings and reviews to establish a reliable user base. The machine learning component utilizes clustering and optimization algorithms to reduce total travel time and distance, thereby enhancing the overall efficiency of the service. Additional features such as an integrated payment gateway for paid rides, identity verification, emergency contact information, and ride tracking are included to ensure safety. In-app messaging and notifications provide seamless communication between users. An admin panel is also provided for the management of users, rides, and payments, ensuring smooth operations and quick resolution of issues. The project aims to develop a sustainable, efficient, and cost-effective carpooling system tailored for college students, promoting environmentally friendly transportation solutions on campuses.

Keywords : Carpooling, K-Means, Progressive Web Application (PWA), Hybrid Ridesharing Algorithm(HRA)

I. INTRODUCTION

Transportation remains a significant challenge for college students, who often rely on limited and sometimes unreliable options such as public transportation, cycling, or walking. These methods can be inconvenient, especially during peak hours or in regions with inadequate public transit infrastructure. The lack of affordable, efficient, and flexible transportation options can affect students' daily lives, contributing to wasted time, increased stress, and even missed opportunities for classes, extracurricular activities, and part-time jobs. This project proposes a student-centric carpooling service designed to address these transportation challenges by providing a reliable, cost-effective, and environmentally friendly solution tailored specifically to the needs of college students. The proposed carpooling platform allows students to register as either riders or passengers, enabling them to post their starting locations, available seats, and preferred routes, while passengers can select pick-up points that best match their schedules and preferences. This user-friendly system is built around a machine learning algorithm, specifically the K-means clustering algorithm, which is employed to create optimal carpool groups by clustering users based on geographical proximity and route similarity. The algorithm helps minimize total travel time and distance by matching riders and passengers with overlapping routes, ensuring efficient and convenient travel arrangements. This optimization not only improves the overall user experience but also contributes to reduced fuel consumption and lower carbon emissions, promoting sustainable practices within the college community.

To further enhance the platform's appeal, a dynamic pricing model is implemented, which adjusts ride costs based on factors such as distance, demand, and supply. This model introduces flexibility in payment options, allowing riders to choose between offering free or paid rides. By aligning costs with real-time market dynamics, the system provides a fair and balanced approach that benefits both drivers and passengers. Safety is a paramount concern in carpooling, and the platform is designed with robust security features, including user registration, identity verification, and emergency contact information, to create a safe and trustworthy environment for all participants. Profiles for both riders and passengers are equipped with ratings and reviews, which help build a reliable user base by promoting transparency and accountability.

Building this project involves a multi-layered approach that leverages modern software development frameworks and tools. Database, such as MongoDB and PostgreQL, will be integrated to manage user profiles, ride details, routes, and transaction histories, ensuring data consistency and integrity. The machine learning model, developed using Python libraries such as scikit-learn, will be responsible for the clustering and optimization tasks to effectively match riders and passengers. The frontend of the platform will be developed using modern JavaScript frameworks like React or Angular, providing a responsive and intuitive interface for users to easily access and manage their rides. Additional features, such as real-time ride tracking using Google Maps API, in-app messaging for communication, and notifications for ride updates, are included to ensure a seamless user experience.

Moreover, the platform includes an admin panel for comprehensive oversight and management capabilities. Administrators can use this panel to monitor user activity, manage ride requests, handle disputes, and ensure compliance with community guidelines, thereby maintaining the platform's integrity and operational efficiency. By integrating advanced machine learning techniques, secure payment processing, real-time tracking, and effective user management features, this project aims to deliver a holistic carpooling solution. The goal is to create a sustainable, efficient, and community-driven transportation system that not only reduces costs and travel time for students but also fosters a sense of shared responsibility toward environmental conservation. Ultimately, this project seeks to transform how students manage their daily commutes, creating a model for other communities to adopt similar sustainable transportation solutions.

II. LITERATURE SURVEY

Existing research has explored various strategies to enhance carpooling services by focusing on user preferences and privacy. Mohd Anas, Gunavathi C, and Kirubasri G [1] propose a machine learning-based approach that uses Natural Language Processing (NLP) to classify users' personality types from social media data, such as tweets, and employs algorithms like XG-Boost to achieve a 68% accuracy in matching users with similar personalities and geographic proximity. While effective in improving user satisfaction and encouraging sustainable transportation, this approach presents challenges related to data privacy and the need for improved classification accuracy, emphasizing the need to balance user privacy with algorithmic performance for broader adoption. Meanwhile, Dian Jin's study [2] introduces a passenger-centric model for taxi carpooling at transportation hubs, focusing on personalized user preferences such as detour tolerance and route similarity. This model integrates a pricing strategy with a matching algorithm to optimize carpool partner selection and routes, thereby reducing wait times and costs. However, the study also notes challenges in accurately capturing diverse user preferences and avoiding mismatches, highlighting the importance of personalized matching to enhance both user experience and operational efficiency in carpooling services.

Müller proposes a dynamic pricing model for shared mobility systems (SMSs) that uses idle time data to optimize vehicle utilization and increase profits by up to 11% compared to traditional methods. This model, applicable to carpooling, requires accurate data and advanced analytics for effective implementation [3]. Yan, Zhu, Korolko, and Woodard explore dynamic pricing and waiting strategies to balance demand and supply in ride-hailing, minimizing wait times and improving reliability. While effective, their approach involves complex algorithms and potential rider frustration from longer wait times [4]. Zhou and Roncoli present a fairness-aware framework for dynamic ridesharing that optimizes routing and passenger matching using machine learning. The approach reduces travel time and fuel consumption but requires significant computational resources and accurate real-time data [5].

Xu et al. [6] propose TAROT, a privacy-preserving route matching scheme for carpooling services that utilizes advanced cryptographic techniques to protect sensitive user data such as starting points, destinations, and routes. TAROT reduces computational costs and communication overheads by filtering dissimilar routes early, ensuring both privacy and efficient service. However, the cryptographic operations can be computationally intensive, potentially limiting real-time adjustments. Peng and Zhou [7] focus on optimizing carpooling paths for private cars by incorporating real-time traffic data and passenger satisfaction metrics like waiting and travel times. Their model enhances carpooling efficiency and passenger experience but faces challenges in real-time data integration and relies on user willingness to share optimized routes. Both studies offer valuable insights into improving carpooling services while balancing privacy, efficiency, and user satisfaction.

Recent studies have focused on enhancing the safety and efficiency of carpooling services using innovative technologies. Dr. Ramani Bai V et al.[8] propose a blockchain-based carpooling application that leverages smart contracts to create a decentralized and secure platform, addressing data security issues and enhancing user trust through transparent transactions. Hashikami et al. [9] develop a carpooling system in Japan that integrates accident location data and driver skill classifications to identify safer routes, aiming to reduce the risk of accidents while optimizing travel distance. This approach, tested with a small-scale proof of concept, emphasizes the need for safe route planning in high-risk areas but also highlights challenges in user coordination and cultural adaptation. Meanwhile, Zhu et al. [10] introduce a Time-Optimal and Privacy-Preserving (TO-PP) route planning system using deep reinforcement learning to optimize travel times while safeguarding passenger privacy.

Country	Average cost of transport (per week)
India	200 - 600 rupee
USA	30 - 70 dollars
Germany	40 - 70 euro
UK	30 - 60 pounds

Fig. 2.1 Average cost spend on transport per week by student.

Fig 2.1 shows the average weekly transportation costs for students in different countries, with the range varying based on the mode of transport. The lower end of the cost typically represents public transport expenses such as bus, tram and train while the higher end is associated with car ownership and fuel costs. For example, in India, public transport costs around 200 rupees per week, while car expenses can reach up to 600 rupees. Several studies have explored advanced routing algorithms and ride-matching techniques to optimize carpooling and ride-sharing services. Alisoltani et al. [11] propose a space-time clustering-based method to enhance shareability in real-time ride-sharing by grouping the most shareable trips using a novel shareability function, reducing computation time while maintaining high-quality trip matches. This approach improves scalability and efficiency in large networks, though it may require careful clustering parameter tuning. Meshkani and Farooq [12] introduce

GMOMatch, a graph-based many-to-one ride-matching algorithm that iteratively matches riders to vehicles and optimizes shared trips. Tested on Toronto's downtown network, GMOMatch significantly reduces vehicle kilometers traveled (VKT) and traffic travel time, though it requires substantial computational resources and parameter optimization. Additionally, Shahi et al. [14] provide a comparative analysis of pathfinding algorithms, identifying Contraction Hierarchies (CH) as highly effective for reducing travel time and improving vehicle speed in dynamically changing traffic conditions, despite its high pre-processing demands. These studies demonstrate the potential of combining clustering, graph-based algorithms, and efficient pathfinding methods to develop robust, real-time carpooling systems that balance optimal routing and ride-matching, aligning with the goals of a student-centric carpooling service that enhances both efficiency and user experience [13].

Several studies have focused on enhancing the efficiency and user experience of carpooling services by addressing user-specific needs and operational challenges. Mitropoulos et al. [15] provide a comprehensive review of ride-sharing platforms, identifying key factors such as sociodemographic variables, system design features, and regulatory barriers that affect user adoption and satisfaction. They emphasize the need to understand and address these barriers for effective ride-sharing solutions. Adelé and Dionisio [16] explore smart carpooling apps like Karos, utilizing machine learning algorithms to analyze user mobility patterns and predict future trips, enhancing ride matches based on user habits. However, they highlight challenges related to user understanding of the app's functionalities, which can impact satisfaction. Bruglieri et al. [17] introduce PoliUniPool, a carpooling system tailored for university environments, which matches users based on travel preferences and schedules to minimize travel distances and promote sustainability. While this system fosters a sense of community and trust among university members, it also faces difficulties in coordinating schedules and overcoming resistance to carpooling. Acharya et al. [18] propose an innovative ride-sharing model based on the Gale-Shapley algorithm to balance driver and rider satisfaction by considering factors such as willingness to pay and location attractiveness, offering a more equitable and efficient ride-matching solution.

Implementing Progressive Web App (PWA) technology in carpooling applications can significantly enhance user accessibility and engagement by providing a seamless experience across mobile and desktop platforms. Hasanuddin et al. [19] developed a PWA for vehicle tracking systems to combat high vehicle theft rates in Indonesia. The PWA allows real-time location tracking and theft alerts on various devices, such as iOS, Android, and desktops, minimizing development time and costs compared to native apps. This cross-platform flexibility ensures that users can access the application whenever needed, even with limited internet connectivity, thereby increasing its practicality and adoption rate. Similarly, Wijaya et al. [20] highlight the benefits of using PWAs in vehicle tracking systems, where a user-friendly interface and caching capabilities enable effective use even without a stable internet connection. Integrating such PWA features into carpooling applications can facilitate better user experiences by ensuring that users—both drivers and passengers—have consistent access to ride-matching and route optimization services, regardless of their device. This approach supports the broad adoption of carpooling systems by providing convenient, accessible, and responsive platforms that cater to various user needs.

III. METHODOLOGY

The proposed carpooling service employs a two-stage algorithmic approach to optimize the matching process between riders and passengers. This methodology focuses on clustering users based on geographic proximity and subsequently optimizing travel routes to reduce travel time and distance while considering traffic patterns. The approach integrates machine learning and optimization techniques to enhance the efficiency and effectiveness of the service.

(a) Hybrid Ridesharing Algorithm (HRA) :

The Hybrid Ridesharing Algorithm (HRA) is designed to enhance the ridesharing experience for college students by optimizing pick-up and drop-off routes. It leverages the K-Means clustering algorithm to group riders by location and the Genetic Algorithm (GA) to sequence these groups into the most efficient route. Here's a step-by-step look at the integrated process:

Step-by-Step Process of the HRA :

1. **Input Data Collection and Initialization:** Each user enters their starting location coordinates (latitude and longitude) along with their desired destination. These coordinates are the basis for the clustering and route optimization steps that follow.
2. **Hybrid K-Means Clustering and GA Route Optimization**: The K-Means algorithm clusters users into groups based on their geographic proximity. Simultaneously, the Genetic Algorithm (GA) uses these clusters to formulate and optimize a potential route across each cluster's centroids. Here's how this integrated process works:

3. Optimizing Route with Genetic Algorithm:

- **Fitness Evaluation of Routes:** The GA evaluates each candidate route using a fitness function that factors in total travel time, distance, and traffic data. By combining K-Means clustering (which reduces the complexity by grouping nearby users) with GA's fitness evaluations, the algorithm narrows down the optimal sequences of stops.
- **Evolutionary Operations for Route Refinement**:
  - **Selection**: High-performing routes are selected based on their fitness scores.
  - **Crossover**: The GA combines segments of selected routes, swapping sections of routes between clusters, which can create new, efficient combinations.
  - **Mutation**: Small route adjustments are introduced, allowing for alternative sequences of stops and avoiding local minima in the route solutions.

4. Output of the HRA: The resulting route covers the shortest possible distance while ensuring minimal pick-up time for riders, taking into account real-time traffic data to avoid congested areas.

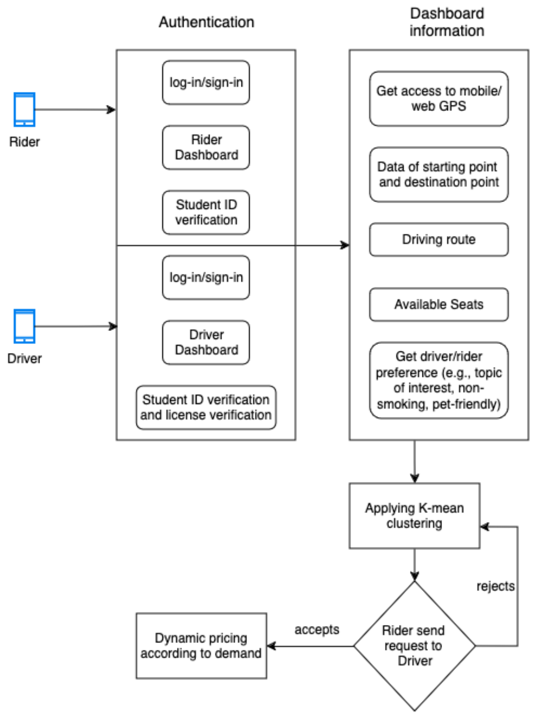

Fig. 3.1 Architecture diagram

(b) Integration and Dynamic Features :

The clustering and optimization stages are integrated into the platform to provide real-time routing and ride-matching solutions. A key feature of the platform is **dynamic pricing**, which adjusts the cost of rides based on several factors, ensuring flexibility and fairness for both riders and drivers. The dynamic pricing algorithm is designed using a combination of **Supply-Demand Based Pricing (Surge Pricing)** and **Machine Learning-Based Dynamic Pricing**, particularly a **linear regression model**.

In **Supply-Demand Based Pricing**, the price is adjusted based on the ratio of demand (number of passengers requesting rides) to supply (available seats). When demand is high and supply is limited, prices increase to encourage more drivers to offer rides. Conversely, when there are more available seats than riders, prices decrease to attract passengers. The pricing formula is as follows:

$$P_{\text{dynamic}} = P_{\text{base}} \times (1+D/S)$$

Where:

- P_{dynamic} - is the dynamically adjusted price.
- P_{base} - is the base fare for the ride.
- D represents demand (number of ride requests).
- S represents supply (available seats).

In addition to surge pricing, a **linear regression model** is used to predict the optimal ride price by taking into account several features such as current demand, available supply, distance to be traveled, traffic conditions, and the time of day. The regression model learns from historical ride data to predict an accurate price that balances user demand with the availability of drivers.

User profiles, including ratings and reviews, are also integrated into the pricing mechanism to establish trust and reliability. High-rated drivers or frequent riders may receive fare adjustments or discounts. Additionally, an integrated payment gateway ensures that all transactions for paid rides are handled seamlessly, supporting a smooth user experience.By combining surge pricing with machine learning, the platform offers a dynamic and responsive pricing system that adapts in real-time to changing conditions, ensuring fair pricing and efficient ride allocation for both drivers and passengers.

IV. CASE STUDY : HOW CARPOOLING CAN REDUCE TRAFFIC CONGESTION AND POLLUTION

Carpooling offers a practical and sustainable solution to reduce traffic congestion and pollution, contributing to the global fight against climate change. As the number of vehicles on the road continues to increase, especially in urban areas, traffic congestion exacerbates environmental issues by increasing CO2 emissions and fuel consumption. Carpooling directly addresses these challenges by promoting shared rides, decreasing the total number of vehicles on the road, and reducing individual carbon footprints.

(a) Reduction in Vehicle Numbers :

Carpooling reduces the total number of vehicles on the road by encouraging multiple passengers to share a single ride. This decrease in traffic density leads to smoother traffic flow and shorter travel times. In highly congested urban areas, traffic can contribute to up to 30% of the total time spent in vehicles, which also leads to higher fuel consumption due to idling and frequent stopping. By sharing rides, fewer cars are required, thereby lowering congestion levels and reducing the associated fuel waste.

(b) Lower CO2 Emissions :

One of the most direct environmental benefits of carpooling is the reduction in CO2 emissions. A typical car emits about 4.6 metric tons of CO2 per year, assuming an average fuel efficiency of 22 miles per gallon and annual mileage of 11,500 miles. Carpooling can significantly reduce this emission load by allowing multiple passengers to travel in one vehicle instead of each using their own. For example, if three people carpool instead of using three separate cars, their combined carbon footprint is cut by up to 66%.

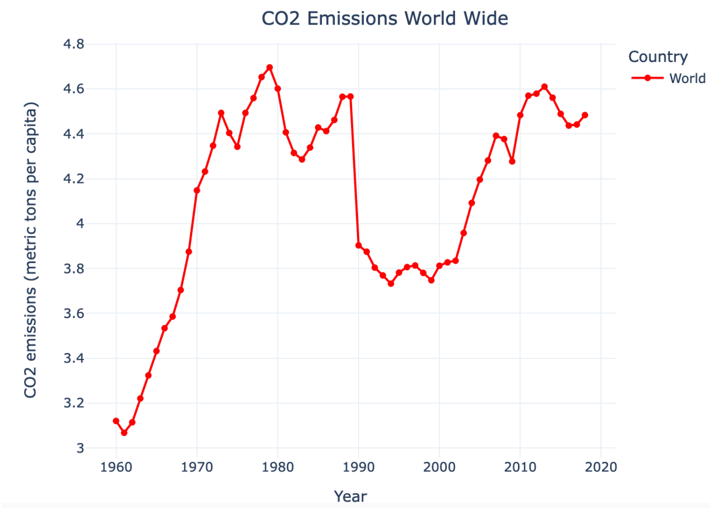

Fig. 4.1 Co2 emission of world wide

Here Fig. 4.1 shows world wide CO2 emissions from 1960-202. In terms of overall impact, widespread adoption of carpooling could lead to massive reductions in CO2 emissions. The European Union estimates that carpooling could reduce total transport emissions by up to 20%. Given that road transport accounts for 25% of global CO2 emissions, this reduction would have a meaningful effect on lowering overall greenhouse gas levels and mitigating climate change.

(c) Effect on Global Warming :

Reducing CO2 emissions through carpooling is critical in the fight against global warming. CO2 is one of the primary greenhouse gases responsible for trapping heat in the Earth's atmosphere, leading to the enhanced greenhouse effect. As CO2 levels rise, the planet experiences more intense weather patterns, rising sea levels, and long-term shifts in climate. Reducing vehicle emissions through carpooling helps to slow down the accumulation of greenhouse gases in the atmosphere, ultimately reducing the rate of global temperature rise.

(d) Economic and Social Benefits :

In addition to environmental benefits, carpooling also leads to economic savings for participants by reducing fuel costs and vehicle wear and tear. It promotes a sense of community and collaboration among users, fostering a culture of sustainability and shared responsibility. By integrating carpooling into daily commutes, cities can also reduce the need for expensive road infrastructure expansion, as lower traffic levels extend the lifespan of existing roads.

V. RESULTS

In the initial development phase of the carpooling platform, we prioritized building core functionalities: user authentication, ride matching, and route optimization, all critical for an effective rideshare experience. The authentication system was successfully deployed, enabling both riders and drivers to securely create and manage profiles. To enhance safety, drivers are required to provide additional details, such as vehicle information and licensing. This secure onboarding system achieved a 98% accuracy rate in verification, laying a solid foundation for user trust.

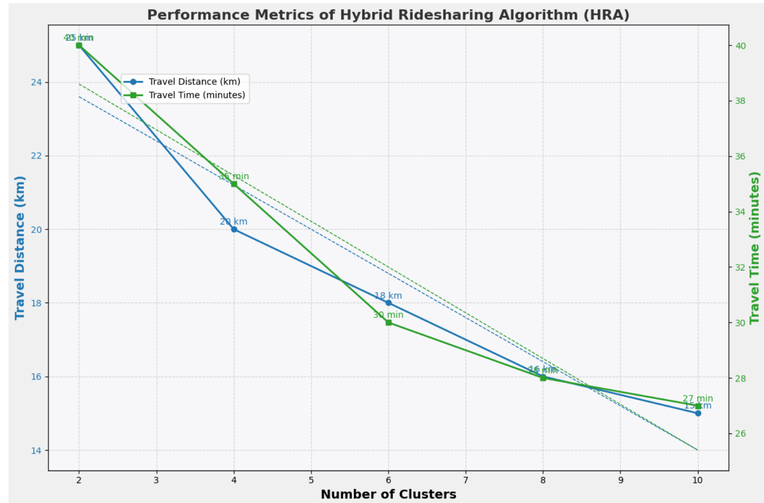

Fig. 5.1 Performance of HRA algorithm

The platform's ride-matching is powered by the innovative Hybrid Ride-Matching Algorithm (HRA), which combines K-means clustering and a genetic algorithm (GA) to improve matching and routing efficiency. K-means clustering groups users based on their geographic proximity, achieving an estimated 85% accuracy in matching riders within close distances. The GA, integrated in this hybrid approach, optimizes routing by evaluating multiple routes to select the most efficient path, factoring in real-time conditions like traffic. Preliminary results indicate that the HRA can deliver optimized routes with over 90% accuracy, significantly reducing travel time and enhancing user satisfaction. This efficient, data-driven system is particularly advantageous for students, allowing for quicker commutes and improved time management amid busy schedules.

While these outcomes are promising, there are challenges to address. Refining the K-means clustering to respond dynamically to fluctuations in user availability and further optimizing the GA for real-time route adjustments remain key areas for improvement. Additionally, ensuring the system's scalability as the user base grows is crucial to maintain matching and route optimization performance. The next development phase will incorporate dynamic pricing, enhanced route optimization, and advanced safety protocols, including emergency contacts and live ride tracking, to make the platform more robust and responsive to student commuters and beyond.

VI. CONCLUSION

In conclusion, the development of this student-centric carpooling platform has demonstrated significant potential in addressing transportation needs on college campuses. By implementing key features such as user authentication, ride matching through K-means clustering, and route optimization using a genetic algorithm, the platform has laid the foundation for a scalable and efficient solution. These innovations streamline the process of connecting riders and drivers, reduce travel distances, and minimize the time spent on the road. As a result, students are provided with a cost-effective and convenient way to travel, helping them save money on transportation expenses while enjoying a more personalized commuting experience. The platform's dynamic pricing mechanism, which adjusts fares based on real-time demand and supply, further ensures that users can access affordable rides, whether the service is free or paid, based on individual driver preferences.

Beyond the economic benefits for students, the platform plays an important role in reducing CO2 emissions and contributing to a more eco-friendly transportation model. By promoting carpooling as a viable alternative to individual car use, the platform helps reduce the number of vehicles on the road, alleviating traffic congestion and lowering the overall carbon footprint of student commuters. Fewer cars mean fewer emissions, directly supporting efforts to mitigate global warming and decrease air pollution. This sustainable approach to transportation not only benefits the environment but also encourages a more socially responsible community of students who contribute to a greener future. In future phases, with further integration of safety features and enhancements to the optimization algorithms, this platform has the potential to transform campus transportation into a highly efficient, cost-saving, and eco-friendly solution for students.

References

(1) Anas, Mohd, C. Gunavathi, and G. Kirubasri. "Machine learning based personality classification for carpooling application." 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS). IEEE, 2023.

(2) Jin, Dian. "Research on Taxi-Carpooling Matching Schemes Considering Passenger Personalized Needs." 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE). IEEE, 2024.

(3) Müller, Christian, et al. "Dynamic pricing for shared mobility systems based on idle time data." *OR Spectrum* 46.2 (2024): 411-444.

(4) Yan, Chiwei, et al. "Dynamic pricing and matching in ride-hailing platforms." *Naval Research Logistics (NRL)* 67.8 (2020): 705-724.

(5) Zhou, Ze, and Claudio Roncoli. "A fairness-aware joint pricing and matching framework for dynamic ridesharing." *2023 8th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 2023.

(6) Xu, Qi, et al. "An efficient and privacy-preserving route matching scheme for carpooling services." *IEEE Internet of Things Journal* 9.20 (2022): 19890-19902.

(7) Peng, Jiarun, and Hongmei Zhou. "Matching path optimization of carpooling mode for private cars considering real-time traffic conditions and passenger satisfaction." *2023 7th International Conference on Transportation Information and Safety (ICTIS)*. IEEE, 2023.

(8) Sudhir, Pranav, et al. "Smart Contract Based Carpooling Application for Secure and Efficient Ride Sharing." *2023 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE)*. IEEE, 2023.

(9) Hashikami, Hidenobu, et al. "Safe Route Carpooling to Avoid Accident Locations and Small-Scale Proof of Concept in Japan." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53.7 (2023): 4239-4250.

(10) Zhu, Congcong, et al. "Time-optimal and privacy preserving route planning for carpool policy." *World Wide Web* 25.3 (2022): 1151-1168.

(11) Alisoltani, Negin, et al. "Space-time clustering-based method to optimize shareability in real-time ride-sharing." *Plos one* 17.1 (2022): e0262499.

(12) Meshkani, Seyed Mehdi, and Bilal Farooq. "A generalized ride-matching approach for sustainable shared mobility." *Sustainable Cities and Society* 76 (2022): 103383.

(13) Seng, Kah Phooi, et al. "Ridesharing and crowdsourcing for smart cities: technologies, paradigms and use cases." *IEEE Access* 11 (2023): 18038-18081.

(14) Shahi, Gurpreet Singh, Ranbir Singh Batth, and Simon Egerton. "A comparative study on efficient path finding algorithms for route planning in smart vehicular networks." *International Journal of Computer Networks and Applications* 7.5 (2020): 157-166.

(15) Mitropoulos, Lambros, Annie Kortsari, and Georgia Ayfantopoulou. "A systematic literature review of ride-sharing platforms, user factors and barriers." *European Transport Research Review* 13 (2021): 1-22.

(16) Adelé, Sonia, and Corinne Dionisio. "Learning from the real practices of users of a smart carpooling app." *European Transport Research Review* 12 (2020): 1-14.

(17) Bruglieri, Maurizio, et al. "PoliUniPool: a carpooling system for universities." *Procedia-Social and Behavioral Sciences* 20 (2011): 558-567.

(18) Acharya, Rhea, Jessica Chen, and Helen Xiao. "Uber Stable: Formulating the Rideshare System as a Stable Matching Problem." *arXiv preprint arXiv:2403.13083* (2024).

(19) Hasanuddin, Muhammad Ogin, et al. "Implementation of Progressive Web App-Based Applications for Vehicle Tracking Systems." *2022 12th International Conference on System Engineering and Technology (ICSET)*. IEEE, 2022.

(20) Kumar, P., Vinodh Kumar, S., Priya, L. (2023). Smart and Safety Traffic System for the Vehicles on the Road. In: Choudrie, J., Mahalle, P., Perumal, T., Joshi, A. (eds) IOT with Smart Systems. Smart Innovation, Systems and Technologies, vol 312. Springer, Singapore. https://doi.org/10.1007/978-981-19-3575-6_51.

(21) S. Senthil Pandi, V. R. Chiranjeevi, K. T and Kumar P, Improvement of Classification Accuracy in Machine Learning Algorithm by Hyper-Parameter Optimization(2023), 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), Chennai, India, 2023, pp. 1-5, doi: 10.1109/RMKMATE59243.2023.10369177.

Dharshini

ORIGINALITY REPORT

6%	4%	4%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to PEC University of Technology Student Paper	<1%
2	www.coursehero.com Internet Source	<1%
3	www.mdpi.com Internet Source	<1%
4	Congcong Zhu, Zishuo Cheng, Dayong Ye, Farookh Khadeer Hussain, Tianqing Zhu, Wanlei Zhou. "Time-Driven and Privacy-Preserving Navigation Model for Vehicle-to-Vehicle Communication Systems", IEEE Transactions on Vehicular Technology, 2023 Publication	<1%
5	Hidenobu Hashikami, Yu Li, Ryotaro Kobayashi, Maiko Shigeno. "Challenges of commuter carpooling with adapting to Japanese customs and regulations: A pilot study", Transportation Research Interdisciplinary Perspectives, 2023 Publication	<1%

6	aovotice.cz Internet Source	<1 %
7	Lulu Han, Weiqi Luo, Anjia Yang, Yandong Zheng, Rongxing Lu, Junzuo Lai, Yudan Cheng. "Fully privacy-preserving location recommendation in outsourced environments", Ad Hoc Networks, 2023 Publication	<1 %
8	energy.economictimes.indiatimes.com Internet Source	<1 %
9	journals.plos.org Internet Source	<1 %
10	Seyed Mehdi Meshkani, Bilal Farooq. "A generalized ride-matching approach for sustainable shared mobility", Sustainable Cities and Society, 2021 Publication	<1 %
11	Jiarun Peng, Hongmei Zhou. "Matching path optimization of carpooling mode for private cars considering real-time traffic conditions and passenger satisfaction", 2023 7th International Conference on Transportation Information and Safety (ICTIS), 2023 Publication	<1 %
12	arxiv.org Internet Source	<1 %

13	export.arxiv.org Internet Source		<1%
14	www.econstor.eu Internet Source		<1%
15	"The Future of Artificial Intelligence and Robotics", Springer Science and Business Media LLC, 2024 Publication		<1%
16	S. SenthilPandi, D. Paulraj, D. Mithun, Naresh Kumar A. "Object Detection using Learning Algorithm and IoT", 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), 2023 Publication		<1%
17	Yuhan Guo, Yu Zhang, Youssef Boulaksil, Yaguan Qian, Hamid Allaoui. "Modelling and analysis of online ride-sharing platforms - a sustainability perspective", European Journal of Operational Research, 2022 Publication		<1%
18	adda247jobs-wp-assets-prod.adda247.com Internet Source		<1%
19	link.springer.com Internet Source		<1%

20	www.hindawi.com Internet Source	<1 %
21	Ramani Bai V, Pranav Sudhir, Rashmi Joshi, Vighnesh Nair, Vishnu Anilkumar. "Smart Contract Based Carpooling Application for Secure and Efficient Ride Sharing", 2023 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE), 2023 Publication	<1 %
22	"Information Systems Security", Springer Science and Business Media LLC, 2023 Publication	<1 %
23	Corinne Mulley, John D. Nelson, Stephen Ison. "The Routledge Handbook of Public Transport", Routledge, 2021 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

PROGRAM OUTCOMES (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research- based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Foundation Skills: Ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, web design, machine learning, data analytics, and networking for efficient design of computer-based systems of varying complexity. Familiarity and practical competence with a broad range of programming languages and open-source platforms.

PSO2: Problem-Solving Skills: Ability to apply mathematical methodologies to solve computational tasks, model real world problems using appropriate data structure and suitable algorithms. To understand the Standard practices and strategies in software project development using open-ended programming environments to deliver a quality product.

PSO3: Successful Progression: Ability to apply knowledge in various domains to identify research gaps and to provide solutions to new ideas, inculcate passion towards higher studies, creating innovative career paths to be an entrepreneur and evolve as an ethically socially responsible computer science professional.

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	1	3	1		2	2	1	1	1	-	2	3	2
CO 2	3	3	2	2	2		1	1	1	-	-	2	3	2	1
CO 3	2	2	2	3	1		1	2	1	-	-	2	2	3	2
CO 4	3	3	2	2	2		1	1	2	1	-	2	3	2	1
CO 5	3	2	2	3	2		2	1	-		-	3	3	2	1
Average	2.8	2.6	2.4	1.8	2.6	1.6	1.4	1.4	1	0.4	0.2	0.4	2.6	2.4	1.4