"ShareMyRide : A Car Pooling Application Using Flutter"

Suraj Shirke1 , Aditya Shitole2 , Akshit Surkutlawar3 , Udaysinha Rasal4 , Prof. Shyamsundar P. Magar5 .
1,2,3,4 Student, Zeal College of Engineering and Research, Pune, India
5 Assistant Professor, Zeal College of Engineering and Research, Pune India .

Abstract

This research presents the development of a mobile- grounded carpooling operation aimed at promoting sustainable and cost-effective transportation. The operation facilitates lift- sharing by connecting motorists with available seats to passengers heading in the same direction. Targeted primarily at scholars and civic commuters, the system addresses rising energy costs, business traffic, and limited access to public conveyance in certain regions. crucial features include stoner enrollment , trip scheduling, real- time lift matching, in- app messaging, and post-trip standing to enhance trust and usability. erected using Flutter for the frontend and Firebase with Node.js for backend operations, the app integrates ultramodern technologies to insure flawless stoner experience. A name point, ' Journey Mates,' intelligently matches druggies grounded on route, timing, and preferences, perfecting collaboration and social commerce. Real- world testing demonstrates that the app significantly reduces trip time, lowers exchanging costs, and contributes to a drop in carbon emigrations, making it a practical result for enhancing civic mobility and environmental sustainability

Keywords: Carpooling, transport efficiency, in-app messaging, recommendation algorithm

Introduction

The growing demand for private transportation has led to a significant rise in vehicle ownership, contributing to traffic congestion, increased fuel consumption, and escalating levels of air pollution. These issues are especially pronounced in urban areas, where public transportation options may be limited or unreliable. To address these concerns, carpooling has emerged as a practical and environmentally responsible alternative to solo driving. By allowing multiple passengers to share a single vehicle for similar routes, carpooling not only optimizes vehicle usage but also helps reduce traffic density, fuel expenditure, and greenhouse gas emissions.

This research focuses on the design and implementation of a carpooling system that leverages modern technology to streamline and enhance the ride-sharing experience. The system incorporates a user-friendly mobile and web application interface, enabling users to register, offer or request rides, and communicate securely. The platform is powered by intelligent matching algorithms that consider route, timing, and user preferences to pair drivers and passengers effectively. In addition to convenience, the system places a strong emphasis on safety through features such as identity verification, background checks, and user rating mechanisms.

Beyond functional benefits, the platform fosters community interaction and collaboration, encouraging a shift toward more sustainable commuting behaviors. The shared cost of travel also
makes it an economical option for both students and professionals. Furthermore, by reducing the number of vehicles on the road, the system contributes to easing pressure on urban infrastructure and lowering the environmental impact of daily travel.

This paper explores the complete development cycle of the carpooling system, from conceptualization to implementation, and evaluates its effectiveness in addressing contemporary transportation challenges. Through technological innovation and a user-centered approach, this project aims to offer a scalable solution for sustainable, cost-effective, and socially engaging commuting.

Objective

The primary ideal of this exploration is to design and develop a app- grounded lift- participating operation that connects vehicles with available seats to passengers traveling in the same direction. The system aims to enhance transportation effectiveness by promoting participated trip, thereby reducing business traffic, lowering carbon emigrations, and minimizing exchanging costs. crucial pretensions include enforcing stoner enrollment , secure communication, and real- time trip operation. The operation also seeks to make stoner trust through standing and feedback systems while offering advanced features like filtered quests, smart recommendations, and implicit integration with external transport services.

Keywords: Ride-sharing, transportation efficiency, messaging feature, recommendation algorithms

Motivation

The provocation behind this exploration lies in addressing the diurnal challenge of chancing suitable lift- participating companions, especially for individualizes exchanging to the same destination at analogous times. Being lift- participating platforms frequently fall suddenly due to limitations like defined service areas, language walls, or complex interfaces, making them inapproachable or inconvenient for a large member of druggies. This design seeks to develop a further inclusive, stoner-friendly result that can be fluently espoused into druggies' daily routines. By offering a dependable, simple, and widely accessible operation, the end is to promote sustainable trip habits and reduce individual auto operation. In doing so, the system contributes to lower business traffic, reduced energy consumption, and dropped environmental impact. Also, the platform encourages community engagement by connecting people with participating routes and fostering social commerce. Eventually, this design aspires to support a more eco-conscious and connected commuting culture through the power of technology.

Methodology

The development of the carpooling application follows a structured and iterative approach based on standard software development lifecycle (SDLC) practices. The process begins with requirements gathering, where both user expectations and business objectives are identified. This includes collecting information through stakeholder interviews, market analysis, and user surveys to determine the essential features and functionalities such as ride booking, route matching, and profile management.Following this, the design and architecture phase involves creating the application's structural blueprint. This includes designing intuitive user interfaces using wireframes and mockups, defining user flows, and planning the backend architecture using Firebase services. Data models and APIs are also conceptualized to ensure seamless communication between components.In the development phase, the frontend is built using Flutter to support cross-platform compatibility, while the backend is powered by Firebase services, including Firestore for data storage, Firebase Authentication for user management, and Cloud Functions for business logic. Google Maps API is integrated to enable real-time geolocation, route planning, and navigation. Code implementation follows a modular approach, with continuous integration and version control.The testing phase incorporates unit testing, integration testing, and system testing to validate the app's functionality, performance, and security. Bugs are identified and resolved to enhance app stability before deployment.During deployment, the application is launched to a production environment. Firebase services are configured for hosting and real-time data synchronization, and the mobile apps are submitted to the Google Play Store and Apple App Store with necessary compliance.Lastly, the maintenance and support phase ensures the app's long-term success by monitoring performance, fixing issues, and implementing regular updates based on user feedback and technological advancements. This ongoing process ensures the app remains reliable, user-friendly, and competitive in the evolving market.

Code and implementation

The ride-sharing application is developed using Flutter for the frontend, Spring Boot for the backend, and Firebase as the database. This process involves creating both the frontend and backend components, integrating third-party APIs, and performing comprehensive testing to ensure functionality and performance.

Design the Frontend: Create a user interface with Flutter, focusing on a responsive and dynamic experience.

Build the Frontend: Develop the Flutter-based UI components, ensuring cross-platform compatibility.

Set Up the Persistence Layer: Use Firebase for real-time data management and storage.

Build the API: Implement the backend logic using Node.js, connecting with Firebase for data handling.

Frontend(Flutter)

The frontend of the ride-sharing application is developed using Flutter, a cross-platform framework known for its ability to deliver responsive and visually appealing user interfaces. Flutter's reactive programming model enables the application to provide real-time updates, ensuring a smooth and dynamic user experience. Instead of relying on traditional data-fetching methods like polling or AJAX, the app utilizes Firebase's real-time database to instantly reflect changes such as ride availability or booking status. This integration allows users to seamlessly interact with the system—viewing rides, making bookings, and receiving notifications in real time.

Backend(Node.js)

The backend is built using Node.js, which supports the development of efficient server-side applications with its event-driven and non-blocking architecture. The backend adopts a Model-View-Controller (MVC) structure, where models define the core data entities like ride requests and user profiles, controllers handle logic such as booking management and user authentication, and views are represented by the Flutter frontend. RESTful APIs are implemented to facilitate communication between the frontend and backend, transferring data in JSON format. Additionally, security is enhanced using Spring Security for authorization and Firebase Authentication for secure user logins, ensuring safe and controlled access to application resources.

Database(Firebase)

For the database, Firebase is employed due to its real-time capabilities and scalability. As a NoSQL cloud database, Firebase stores data in a JSON-like format, allowing for flexible and efficient handling of complex data structures such as ride details and booking histories. Its native compatibility with both Flutter and Node.js simplifies integration and data flow across the system. One of Firebase's key advantages is its real-time synchronization, which enables features like live ride tracking and instant updates without requiring constant server requests, significantly improving performance and user experience.

User Interface

Login/Register: Users can either log into their existing accounts or register to create a new one,providing essential information to participate in ride-sharing services.

Post Ride: Users post ride details, such as departure location, destination, and time, to connect with passengers. Drivers can review vehicle details, like model and rental conditions, and check feedback from other users to ensure a reliable experience.

Ride Status: This feature allows users to view the current ride status. Once the trip is completed, the driver or passenger can mark the ride as finished, officially ending the journey.

Chat Page: A built-in chat feature facilitates real-time communication between drivers and passengers, allowing them to coordinate trip details or address any immediate concerns.

Merits and Demerits

Advantages:

Cost Savings: By sharing rides, users can split transportation costs, lowering individual expenses on fuel, tolls, and other travel fees.

Reduced Traffic Congestion: Ride-sharing helps decrease the number of vehicles on the road, which can reduce travel times, ease congestion, and diminish pollution levels.

Environmental Benefits: Fewer cars on the road contribute to a reduction in greenhouse gas emissions, supporting a cleaner environment.

Convenience: The carpooling platform allows users to quickly find or offer rides, making it easier to arrange shared transportation.

Increased Social Interaction: Carpooling promotes socialization, as passengers can interact and connect with others on their journey.

Real-Time Tracking: Users can track ride status in real-time, enabling easy ride coordination and offering flexibility for potential changes.

Disadvantages:

Varied Timing Requirements: Passengers may have different schedules or need to make stops along the way, which can disrupt a smooth, uninterrupted trip.

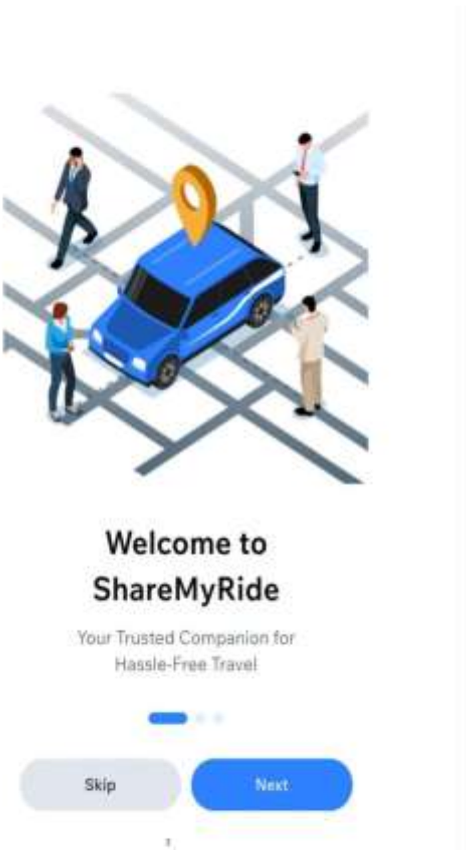
Reduced Privacy: Sharing a vehicle requires passengers to give up some level of privacy, which may be uncomfortable for certain users.

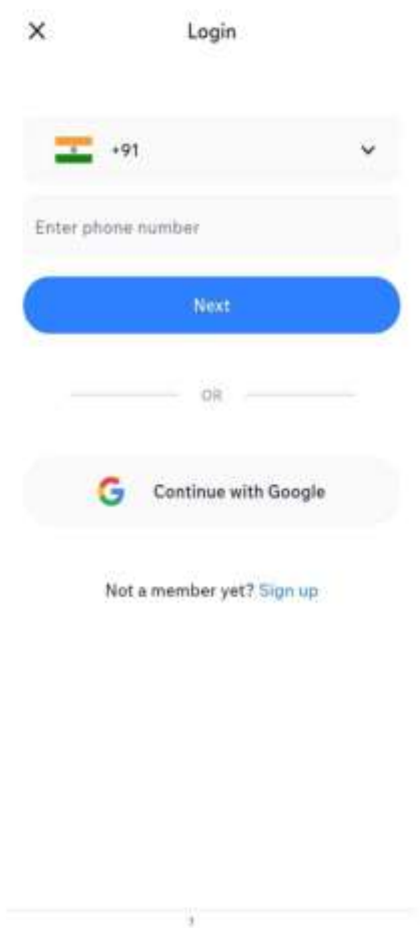
Ongoing Maintenance: Continuous updates and support are necessary to maintain the application's usability and performance, which can require significant resources.

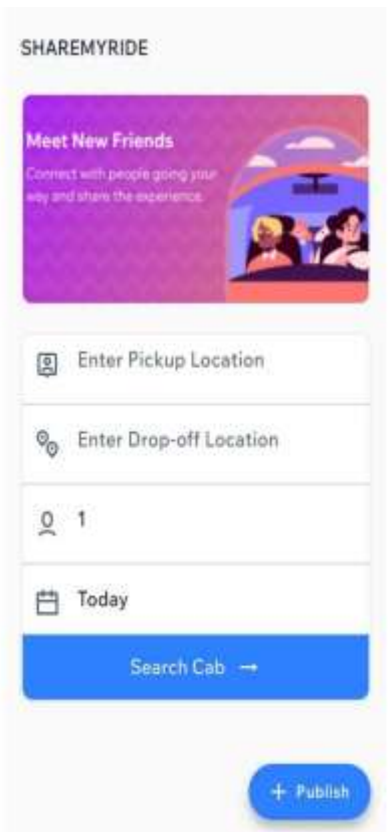
Future Scope

Future developments in carpooling applications are expected to focus heavily on personalization, integration with emerging technologies, and enhanced safety. Rather than relying solely on basic criteria such as location and departure time, future matching algorithms could incorporate more nuanced user preferences, including desired conversation levels, music choices, and even preferred driving styles. By leveraging machine learning, these systems can learn from past user behavior and feedback to improve match accuracy and rider satisfaction over time. This personalization aims to create more comfortable and tailored ride-sharing experiences. Additionally, integrating carpooling apps with smart city infrastructure and public transportation networks could enable real-time data sharing for improved route optimization and reduced traffic congestion. Such integration, potentially facilitated through partnerships with municipal governments and transit authorities, may also offer incentives for eco-conscious commuting. As autonomous vehicle technology continues to evolve, self-driving cars could become a valuable component of carpooling services, providing on-demand, accessible transportation for those who cannot drive, such as elderly or disabled users. Moreover, future carpooling platforms might incorporate sustainability tracking features, allowing users to view their environmental impact in terms of reduced CO_2 emissions or total distance saved through shared rides. Multi-modal transportation integration could further enhance flexibility by combining carpooling with other travel options like public transit, cycling, or ride-hailing, enabling seamless trip planning. In parallel, strengthening safety protocols will be essential. Advanced features such as biometric verification, comprehensive background checks, real-time tracking, and emergency response options can improve user trust and create a more secure environment. Altogether, these innovations have the potential to make carpooling more efficient, inclusive, and sustainable, positioning it as a critical component of future urban mobility solutions.

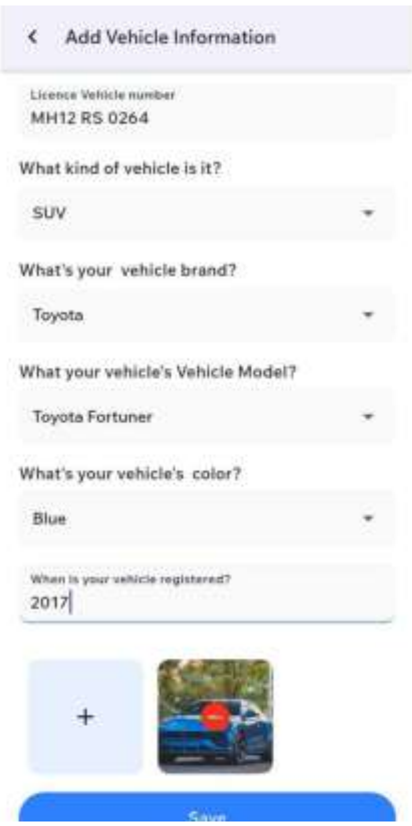
Implementation Results

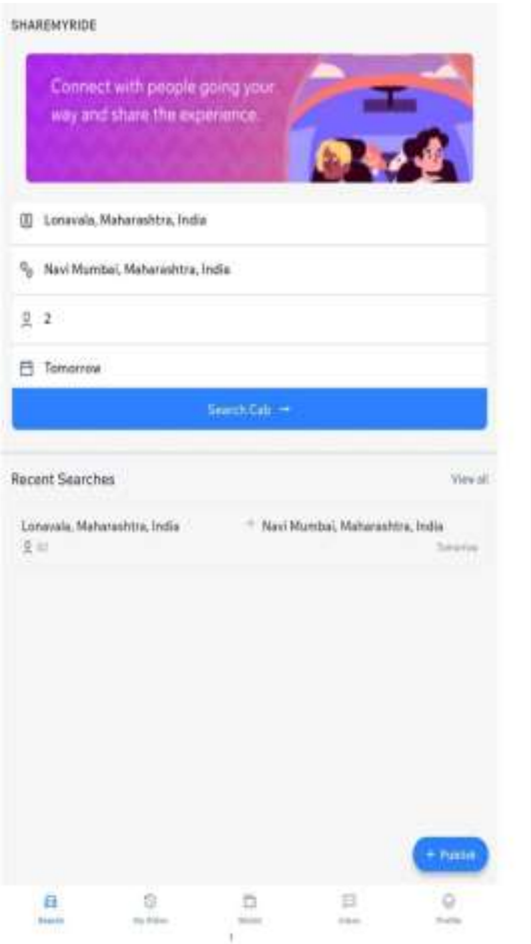



Login
+91
Enter phone number
Next
OR
Continue with Google
Not a member yet? Sign up


SHAREMYRIDE
Meet New Friends
Connect with people going your way and share the experience.
Enter Pickup Location
Enter Drop-off Location
1
Today
Search Cab
Publish


Account Verification
Upload Required Documents
Complete your registration by uploading the following documents.
ID Proof
Front And Back Photo


Add Vehicle Information
Licence Vehicle number
MH12 RS 0264
What kind of vehicle is it?
SUV
What's your vehicle brand?
Toyota
What your vehicle's Vehicle Model?
Toyota Fortuner
What's your vehicle's color?
Blue
When is your vehicle registered?
2017
+



Literature Review

Ridesharing has emerged as a transformative solution to address urban transportation challenges such as congestion, emissions, and underutilized vehicle capacity. Xingyuan Li et al. (2024) examined whether ridesharing could improve the reserve capacity of transportation networks. Their findings highlight that integrating ridesharing systems can enhance the efficiency and robustness of urban traffic infrastructure, especially under high-demand conditions.
Similarly, Luydia et al. (2024) developed a carpooling system using Flutter, focusing on the technical feasibility of cross-platform mobile applications for ridesharing. Their implementation underscores the growing importance of mobile app development in facilitating real-time ride-matching and improving user accessibility.

David Zar et al. (2023) explored the psychological aspect of ridesharing by analyzing how information disclosure influences user satisfaction. They found that transparency regarding route, co-passengers, and estimated travel time significantly boosts user trust and satisfaction, which are critical for ridesharing adoption.
From a data-driven perspective, Wenbo Zhang and Satish V. Ukkusuri (2021) proposed Share-aCab, a scalable clustering algorithm to group ride requests efficiently using geolocation data. This approach demonstrates how machine learning can optimize group rides and reduce computational overhead in large-scale systems.
Ajinkya Ghorpade et al. (2021) introduced Pool, a peer-to-peer ride-sharing application. Their work emphasizes decentralization, enabling users to connect directly without relying on centralized ride-hailing services, thereby promoting community-based transport solutions.
Finally, Hajra Qadir et al. (2018) presented an optimal recommendation framework for carpooling, which combines user preferences and travel patterns to improve matching accuracy and user experience.
Collectively, these studies cover technical development, user psychology, algorithmic optimization, and infrastructure impact—offering a comprehensive understanding of the multifaceted domain of ridesharing systems.

Conclusion

This carpooling web application presents an efficient and environmentally friendly way for people to share rides, cutting down on emissions and promoting smarter use of resources. By linking drivers and passengers traveling similar routes, it encourages both cost-effective commuting and stronger community connections. Built with Flutter for the frontend, JavaScript for the backend, and Firebase for data handling, the platform delivers a smooth and dependable user experience. Addressing everyday transportation issues, the application makes ride-sharing more accessible and convenient, ultimately supporting a greener lifestyle and a more connected society through sustainable commuting solutions.

References

1]Xingyuan Li ,Yan Chang,Huijie Peng,Xiaomin Dai: “Can Ridesharing Improve the Reserve Capacity ofTransportation Network?” IEEE Access(2024).

2]Agielia luydia , Manush Aditya , Mithun K : “Car Pooling System Using flutter”. IJNRD.ORG(2024).

3]David Zar , Noam Hazon , Amos Azaria: “Information Disclosure for Increasing User Satisfaction From a Sharing Ride “. IEEE Access(2023).

4]Wenbo Xhang ,Satish V. Ukkusuri: “Share-aCab: Scalable Clustering Taxi Group Ride Stand From Huge Geolocation Data”. IEEE Access(2021).

5]Ajinkya Ghorpade , Apurva Joshi , sumit Mali , Harsh Agarwal : “Pool : A Peer-to-Peer Ride Sharing App” . IJEAST(2021).

6]Hajra Qadir , Osman Khalid , Raheem Nawaz , Atta Ur Rehman Khan: “An Optimal Ride Sharing Recommendation Framework for Carpooling services” . IEEE Access(2018).