IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 10, November 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1210.08

RideLink: A Flask-Based Modular TNC Framework for Real-Time Safety and Equitable Ride Management

Keerthana HR, Madhura Chandrashekhar Naik
UG Scholar, Department of CSE
MVJ College of Engineering, Bengaluru, India
keerthikeerthi81056@gmail.com, madhuranaik589@gmail.com
M H Prashanth, Madan M
UG Scholar, Department of CSE
MVJ College of Engineering, Bengaluru, India
mhprashanth24@gmail.com
madan06032004@gmail.com
Sushma D
Asst. Professor, Dept. of CSE
MVJ College of Engineering, Bengaluru, India
sushmadevaraj11@gmail.com

Publication History
Manuscript Reference No: IJIRAE/RS/Vol.12/Issue10/OCAE10090
Research Article| Open Access | Double-Blind Peer-Reviewed | Article ID: IJIRAE/RS/Vol.12/Issue10/OCAE10090
Received:22,October 2025, Revised: 28, October 2025, Accepted:31, October 2025, Published Online: 21, November 2025. https://www.ijirae.com/volumes/Vol12/iss-10/08.OCAE10090.pdf
Citation:Keerthana,Madhura,Prashanth,Madan,Sushma(2025), RideLink: A Flask-Based Modular TNC Framework for Real-Time Safety and Equitable Ride Management , IJIRAE: International Journal of Innovative Research in Advanced Engineering, Volume 12, Issue 10 of 2025 pages 393-400
doi:>https://doi.org/10.26562/ijirae.2025.v1210.08
BibTeX Key: Sushma@2025Ridelink
IJIRAE papers should be cited as IJIRAE (International Journal of Innovative Research in Advanced Engineering, AM Publications, India 2025, ISSN 2349-2163, https://doi.org/10.26562/ijirae.2025.v1210.08 The journal's official abbreviation is IJIRAE. Orcid: https://orcid.org/0009-0004-9398-7488

Abstract: RideLink is a user-friendly and secure ride-sharing app designed to meet the needs of drivers, passengers, and administrators alike. Built with the Flask framework, it offers drivers a simple way to create and manage rides, while passengers enjoy an easy booking experience including a special feature that lets women filter for women-only rides, promoting safety and comfort.To keep everyone connected in real-time, RideLink includes chat rooms powered by Flask-SocketIO, making coordination smooth and hassle-free. Security is a top priority: the app uses bcrypt for strong user authentication and features an Emergency SOS option, which sends instant alerts via Twilio SMS if a passenger or driver needs urgent help.Admins can maintain high platform standards by reviewing and approving driver documents, ensuring trustworthiness across the board. Plus, a built-in scheduler automatically updates ride statuses to keep data accurate and up-to-date without manual effort. Overall, RideLink combines smart ride management, real-time communication, and essential safety tools to create a reliable and modern shared transportation experience for everyone.
Keywords: Ride-sharing, user authentication, bcrypt, Twilio, Emergency SOS, women-only rides, secure booking, administrator control, shared mobility, intelligent transportation systems.

I. INTRODUCTION

The Transportation Network Companies (TNCs) like Uber and Ola have revolutionized urban transportation by making it easier and more convenient to get around. They've tackled tough challenges like optimizing routes and ensuring enough vehicles are available when needed. However, this centralized model has also brought some downsides, including unpredictable pricing, a lack of transparency with data, and ongoing concerns about the safety of both users and drivers. Recent research has been exploring ways to overcome these issues by looking into decentralized systems, advanced demand forecasting, fairer pricing models, and important safety measures such as monitoring driver behavior and verifying vehicle authenticity. Inspired by the need for a modern ride-hailing platform that combines essential features with enhanced safety and administrative oversight, we developed RideLink. RideLink is built on the Flask framework and designed to serve Drivers, Passengers, and Administrators with a focus on security and ease of use. One of its standout features is a women-only ride filter, promoting inclusivity and safety for female passengers.There's also an Emergency SOS alert system that uses Twilio SMS to quickly notify help in urgent situations. Real-time communication between users is supported through Flask-SocketIO chat rooms, while strong authentication is ensured with bcrypt. Additionally, a background scheduler automatically updates ride statuses to keep everything running smoothly.This paper presents the design and implementation of RideLink, showcasing how it offers a secure, automated, and user-friendly ride-sharing experience that addresses the key operational and safety challenges faced by today's ride-sharing platforms.

II. LITERATURE REVIEW

This comprehensive review paper presents the current landscape of phishing website detection techniques. It categorizes existing methods into rule-based, heuristic-based, and machine learning-based approaches, detailing how each class works, their advantages, and their limitations. [1]. The study emphasizes the growing importance of automation and adaptive learning to counter increasingly sophisticated phishing tactics. It also highlights the trend toward hybrid systems that combine multiple techniques to improve accuracy and reduce false positives. This paper is valuable for researchers looking to understand the breadth and depth of phishing detection technologies.[2]This paper focuses on the unique vulnerabilities of IoT-based smart business applications in the context of Industry 4.0 and how phishing attacks can exploit these systems. It reviews a variety of countermeasures, including traditional security protocols, machine learning, and deep learning-based strategies. The paper also discusses the role of network architecture, data security policies, and AI-driven defense mechanisms in combating phishing in the IoT ecosystem. It offers a forward-looking view of how businesses can prepare for future phishing threats by integrating intelligent security frameworks into their operations. This study proposes a browser extension that leverages machine learning to detect phishing websites in real time. The system automatically extracts features from the webpage, such as URL patterns, HTML tags, and embedded scripts, and then classifies them using supervised learning algorithms. [3].The browser extension provides a user-friendly interface that warns users about suspicious websites before any data is compromised. The paper evaluates the model's performance on benchmark datasets and emphasizes its potential for deployment in consumer grade cyber security tools. This paper addresses the growing concern of phishing on mobile platforms by applying data mining techniques to detect and block malicious websites. It uses a curated dataset of phishing and legitimate websites, focusing on features like URL characteristics, page layout, and script content. The study employs classification algorithms such as Decision Trees and Naïve Bayes to achieve high detection accuracy. [4] The system is optimized for mobile environments, ensuring low computational overhead and real-time processing, making it suitable for mobile applications. This paper introduces a novel multi-layered framework that tackles phishing detection from a social engineering perspective. It integrates multiple detection layers such as blacklists, whitelists, machine learning classifiers, and heuristic filters.[5]The authors demonstrate the system's effectiveness across different platforms including mobile apps and browser extensions. The framework is designed to identify not only technical indicators but also social cues often present in phishing attempts, enhancing the overall detection rate. The paper suggests future improvements using AI and behavioral analytics. This paper provides an in-depth analysis of machine learning techniques applied to phishing detection. It evaluates several algorithms such as Random Forest, Support Vector Machine, and k-NN for classifying phishing sites based on various features like domain name length, use of HTTPS, and JavaScript behavior. [6] The study also explores the integration of these models into browser extensions, showcasing real-time protection against phishing attacks. Performance metrics including accuracy, precision, and recall are discussed to validate the models' effectiveness. This paper provides a comprehensive study on the anatomy of phishing attacks, focusing on their evolution and the psychology behind social engineering techniques. It introduces a new conceptual model that maps the lifecycle of a phishing attack, from initial lure to data exhilaration. [7]The study also evaluates current detection mechanisms and suggests areas for improvement, particularly in user behavior analysis and adaptive learning systems. This research applies Natural Language Processing (NLP) and machine learning to detect phishing URLs.[8]The study focuses on analyzing URL text patterns and linguistic cues to differentiate phishing sites from legitimate ones. The approach is lightweight and suitable for real-time systems, particularly in browser and email filters. [9] This review paper explores the use of deep learning to detect malicious browser extensions and phishing links. It discusses how hybrid architectures combining CNNs and RNNs are particularly effective for analyzing behavioral and contextual features.[10] The paper emphasizes the importance of detecting threats not just from web pages but also from browser extensions, which are often overlooked. It includes comparative analyses of different models and datasets, advocating for robust training data and continual model updates to adapt to new threats. [11] This paper provides an extensive review of machine learning techniques used for detecting malicious URLs. [12]It categorizes existing methods into rule-based, heuristic-based, and machine learning-based approaches, detailing their mechanisms, advantages, and limitations. The study emphasizes the importance of automation and adaptive learning to counter sophisticated phishing tactics and highlights [13] the trend toward hybrid systems that combine multiple techniques to improve accuracy and reduce false positives.

III. METHODOLOGY

The RideLink platform is built with a modular, three-tier architecture using the lightweight and flexible Python Flask framework. This approach ensures the app is both reliable and easy to maintain. Scalability, maintainability, and effective performance are guaranteed by the modular, three-tier architecture of the RideLink platform. The application is separated into three logical layers by this structure: the data layer (database), the application layer (backend logic), and the presentation layer (frontend/UI).

System Architecture and Technology Stack

RideLink's system is divided into three main layers: the Data Tier, the Application Tier, and the Presentation Tier. The Data Tier uses PostgreSQL together with SQLAlchemy ORM to securely store and manage all essential information such as users, rides, and documents. This setup provides a solid foundation to handle the continuous flow of data typical in ride-sharing services. The Application Tier, powered by Flask, handles the core business logic, RESTful APIs, and all security and communication functions. Lastly, the Presentation Tier delivers a smooth and responsive user experience through web interfaces built with HTML, CSS, and Jinja2 templates.

IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 10, November 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1210.08

These interfaces are designed to be clear and user-friendly for admins, passengers, and drivers, which is especially important for a service focused on safety and efficient transport.

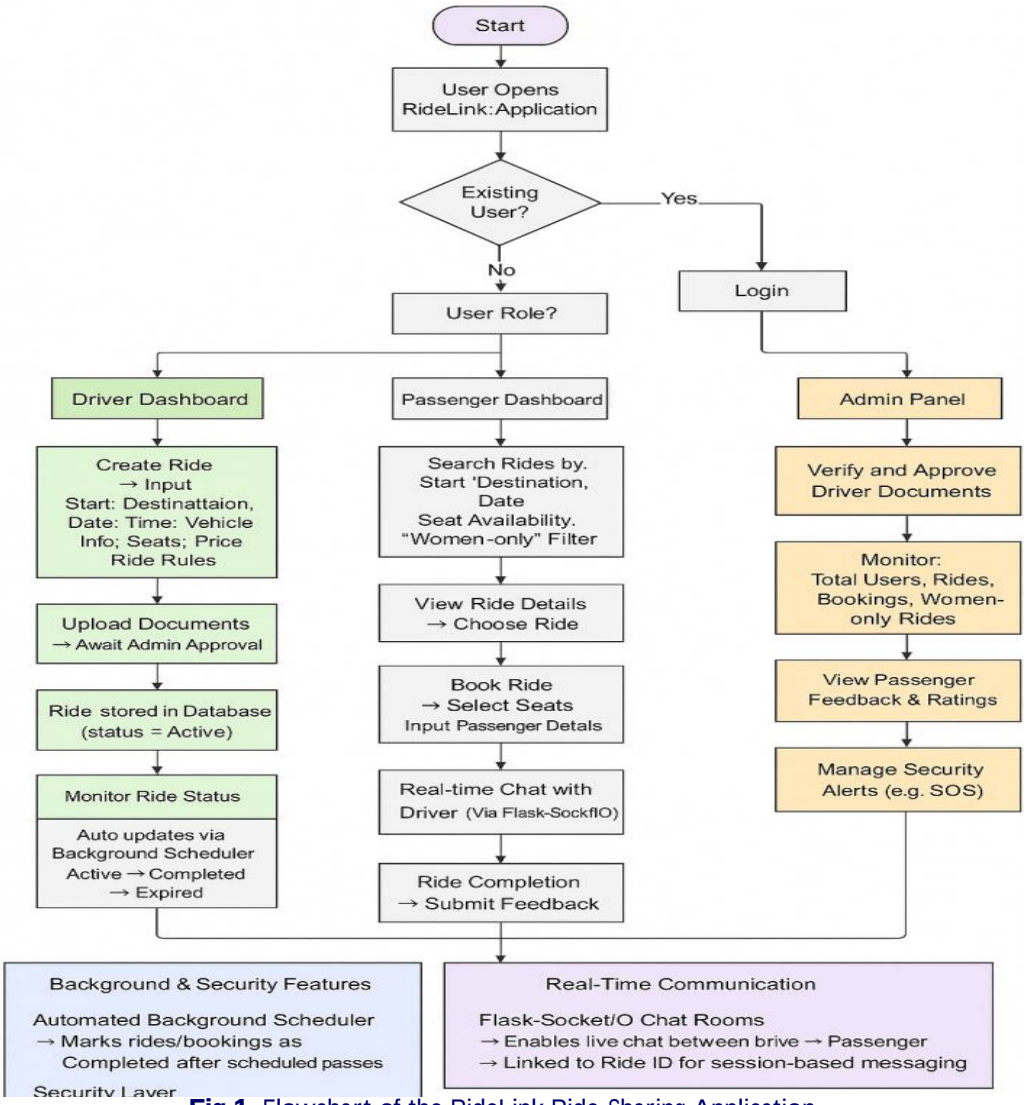

Fig.1. Flowchart of the RideLink Ride-Sharing Application

Access Control and User Authentication

Users can either register as new users or log in using their existing credentials during the application's initial user authentication process. The bcrypt hashing algorithm is used to safely store all passwords, guaranteeing defense against unwanted access. Users are taken to dashboards according to their roles Driver, Passenger, or Administrator after successfully completing the authentication process. To stop brute-force attacks, the system also limits multiple unsuccessful login attempts and uses session-based role management.

Module for the Driver

With the help of a customized dashboard, drivers can create and manage rides by entering information about the vehicle, starting point, destination, date, time, seat availability, cost, and unique ride rules. Official documents that they upload are pending until the administrator verifies them. After being created, rides are initially marked as "Active" in the database. In addition to accepting or rejecting passengers, the driver can keep an eye on booking requests and view ratings and comments. Depending on the time, a background scheduler automatically changes the ride status from "Active" to "Completed" or "Expired"

The Passenger Module

In order to improve safety and inclusivity, passengers can search and filter rides by origin, destination, date, seat availability, and "women-only" preference using an easy-to-use dashboard. Once a ride has been chosen, passengers can reserve seats, enter their personal information, and simulate a payment confirmation to complete the reservation. Flask-Socket IO-powered integrated real-time chat functionality allows drivers and passengers to communicate directly for coordination. After the ride is over, riders can rate their experience and leave comments.

Module for Administrators

The general integrity of the platform must be preserved by administrators. Driver verification, ride and user management, and system performance metrics, such as the number of users, active rides, and rides reserved for women, are all accessible through their dashboard.

IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 10, November 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1210.08

Administrators have the ability to handle security alerts, including user-triggered emergency SOS notifications, review feedback, and keep an eye on booking statistics. This oversight guarantees platform-wide data consistency, security, and service quality.

Emergency Alerts and Real-Time Communication

In order to facilitate smooth interaction between drivers and passengers during ongoing rides, the application integrates real-time communication channels through Flask-Socket IO chat rooms. Additionally, in emergency situations, users can send emergency alerts to predefined contacts via an SOS feature integrated via the Twilio SMS API. This feature supports RideLink's user protection objectives by providing an essential layer of security.

Context and Security Activities

To keep the database consistent and current without requiring human intervention, a background scheduler runs on a regular basis to automate repetitive tasks like updating completed reservations and expired rides. Sensitive user data is protected during all transactions thanks to additional security measures like role-based session control, encrypted credential management, and input validation for uploaded files.

IV. RESULTS

Using the Python Flask framework, the RideLink platform was effectively created as a safe and effective ride-sharing web application. Through an easy-to-use interface, the system lets drivers create and oversee rides while facilitating passenger bookings. Flask-Socket IO was used to implement real-time communication so that user coordination was seamless. For female passengers, a women-only ride filter improves inclusivity and safety. In order to provide immediate alerts in emergency situations, the platform also incorporates an Emergency SOS feature that uses Twilio SMS. To protect data privacy, bcrypt encryption is used to secure user authentication. Administrators are able to maintain service quality and check driver documentation. Furthermore, ride statuses are updated automatically by a scheduler that doesn't require human input.

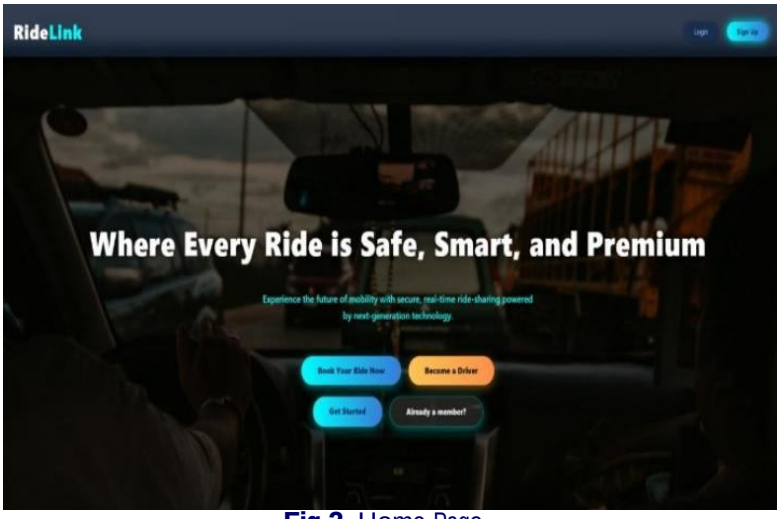

Fig.2. Home Page

The interface displays the primary booking form and navigation, emphasizing security features like the Emergency SOS within a modular Flask framework.

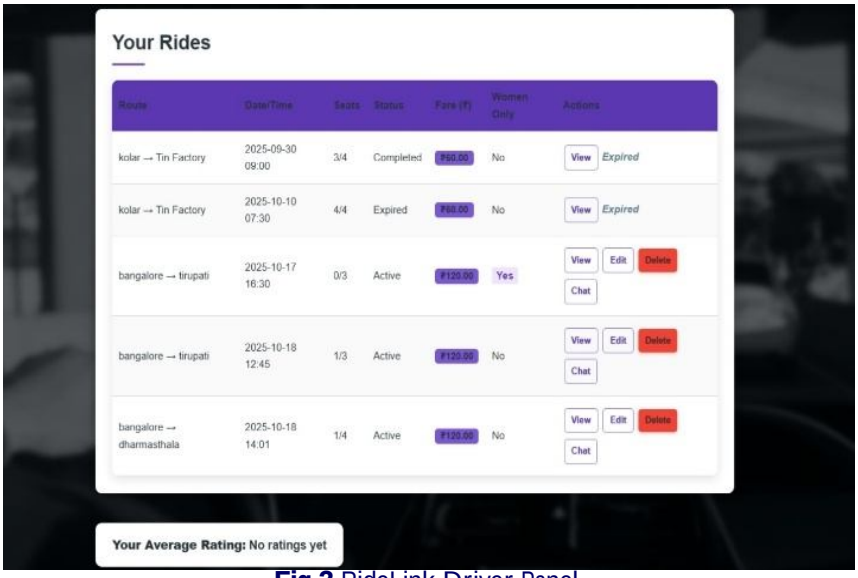

Fig.3 RideLink Driver Panel

IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 10, November 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1210.08

The table lists driver-created rides, displaying status, seats booked, and the 'Women Only' flag, illustrating the system's RBAC and specialized security features.

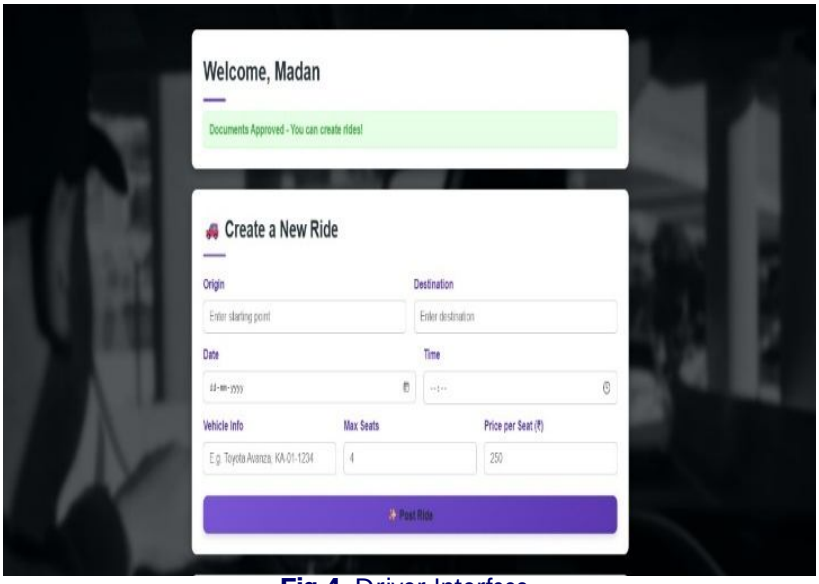

Fig.4. Driver Interface

This interface allows verified drivers to post a ride by inputting the route, schedule, vehicle information, and price per seat, illustrating the system's driver verification and ride-sharing functionality.

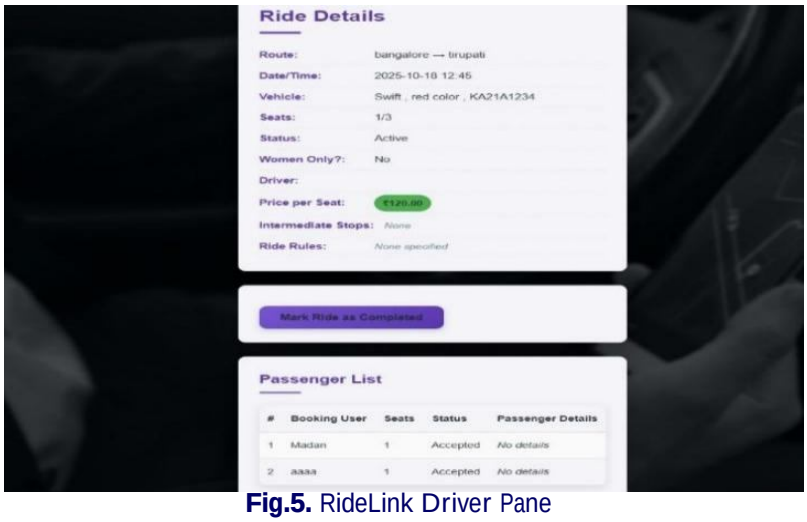

Fig.5. RideLink Driver Pane

This view displays comprehensive ride specifics, the 'Women Only' flag, and the list of accepted passengers, illustrating the granular control and accountability of the system's ride life-cycle.

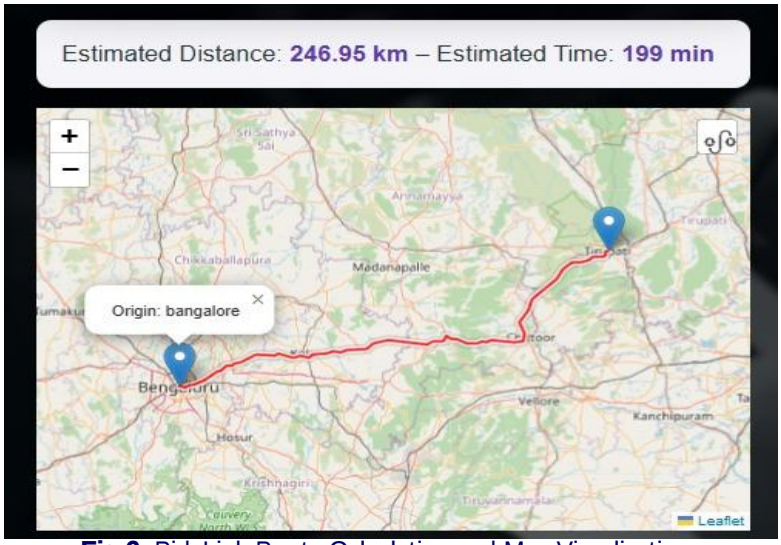

Fig.6. RideLink Route Calculation and Map Visualization.

IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 10, November 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1210.08

The map displays the optimal route between Bangalore and Tirupati, accompanied by the calculated distance (246.95 km) and estimated time (199 min), illustrating the system's core geospatial functionality.

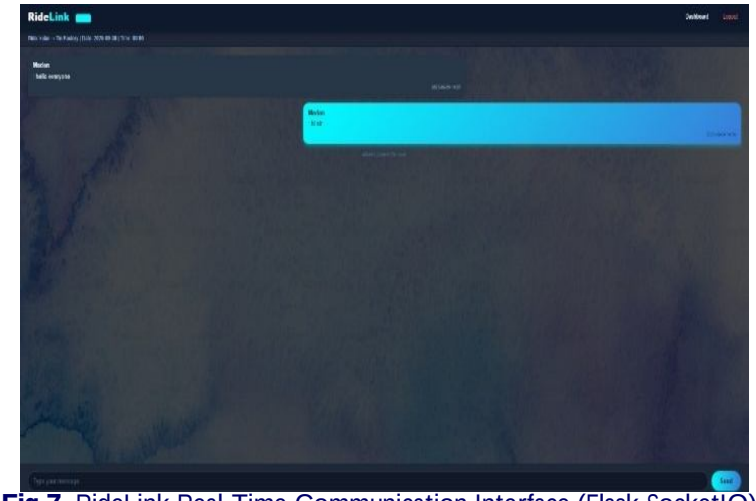

Fig.7. RideLink Real-Time Communication Interface (Flask-SocketIO).

This figure illustrates the in-app chat module, facilitating instantaneous and secure communication between the driver and passengers for crucial real-time coordination before and during the ride.

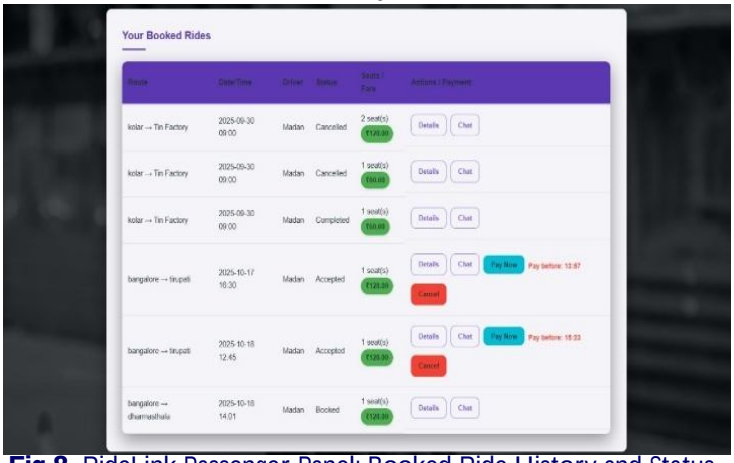

Fig.8. RideLink Passenger Panel: Booked Ride History and Status.

This table displays the passenger's history of booked rides, indicating status (Accepted, Cancelled, Completed), payment options, and links for detailed viewing and real-time chat.

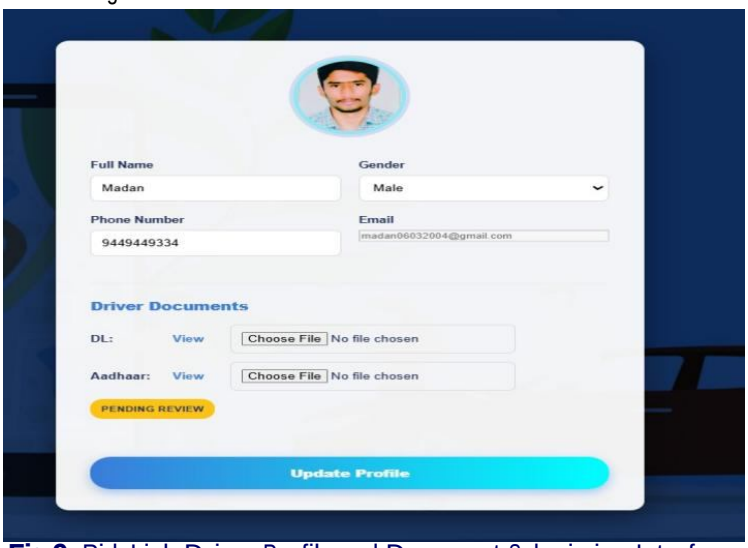

Fig.9. RideLink Driver Profile and Document Submission Interface.

This module is essential for driver onboarding, allowing submission of necessary documents (DL, Aadhaar) and displaying the "PENDING REVIEW" status before Admin Oversight grants ride-creation privileges.

IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 10, November 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1210.08

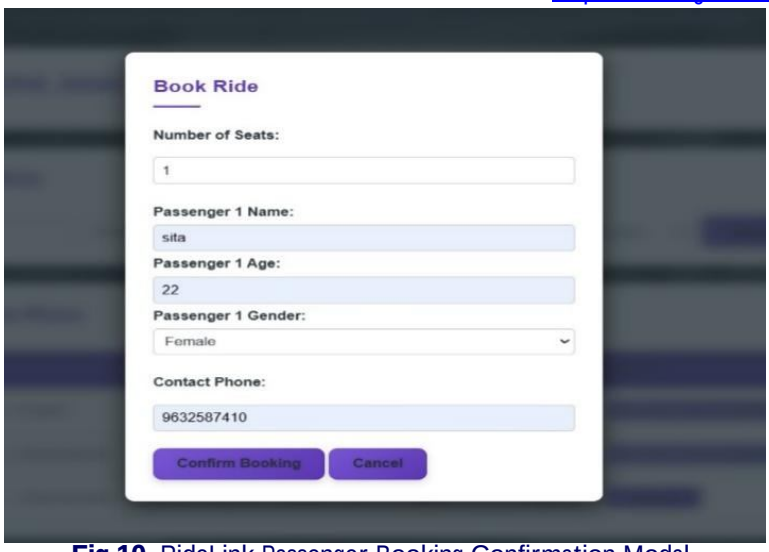

Fig.10. RideLink Passenger Booking Confirmation Modal.

This interface captures passenger details, including gender, number of seats, and contact information, essential for confirming the ride and enabling specialized security features like the women-only ride filter.

V. CONCLUSION

The RideLink project successfully addressed the crucial safety and transparency shortcomings in centralized TNC models by using Flask to create a feature-rich, modular, and secure ride-sharing platform. The platform's main contribution is the incorporation of strong RBAC and specialized security features, such as a women-only ride filter and an Emergency SOS alert (powered by Twilio), to guarantee system accountability and guard against integrity breaches. Additionally, Flask-SocketIO improves user coordination by facilitating real-time communication. As an example of how focused functional design in a lightweight architecture can address major industry challenges, RideLink serves as a proof-of-concept for safe and equitable mobility. The RideLink architecture is ready to grow into more sophisticated intelligent systems. The complete integration of predictive safety systems, like the Stress-Aware Recommendation System (SARS), is the main path forward. In order for the system to automatically suggest trip rejection in order to prevent accidents, sensor data must be incorporated to track driver stress and fatigue. Second, in order to achieve complete data transparency and do away with exorbitant centralized commission fees, we intend to investigate the use of Block chain in payment and rating systems. Lastly, by precisely projecting future driver ride requests, AI-powered demand forecasting will be created to improve operational efficiency.

REFERENCES

1. Y.Indulkar and A.Patil, “Sentiment Analysis of Uber and Ola Using Deep Learning,” in Proceedings of the International Conference on Smart Electronics and Communication (ICOSEC), 2020.
2. R.K.Yadav and G.Kishor, “Comparative Analysis of Route Planning Algorithms on Road Networks,” in Proceedings of the International Conference on Communication and Electronics Systems (ICCES), 2020.
3. U.Shroff and V.Kolwankar, “Fliver: An Enhancement to the Existing Autorickshaw System in India,” in Proceedings of the International Conference on Communication and Electronics Systems (ICCES), 2020.
4. Y.Indulkar, “Comparative Study of Machine Learning Algorithms for Twitter Sentiment Analysis,” in Proceedings of the International Conference on Emerging Smart Computing and Informatics (ESCI), 2021.
5. M.N.Indriani and A.C.Arif, “Mobile Application Design for Solo Traveler: Travemate,” in Proceedings of the International Conference on Smart Technology Applications and Information Engineering (APICS), 2022.
6. E.R.G.,S.Ganesh M., et al., “An Automated Cost Prediction in Uber/Call Taxi Using Machine Learning Algorithm,” in Proceedings of the International Conference on Advanced Computing and Innovative Technologies in Engineering (ICACITE), 2022.
7. E.E.Vigneswaran, V.P.Gowtham Raj, and M.Selvaganesh, “Authentication System for Automobiles Using Face Recognition Methodology,” in Proceedings of the International Conference on Smart Systems and Inventive Technology (ICSSIT), 2022.

A. Aggarwal, S. Nagpal, and S. Gaba, “Blockchain and Autonomous Vehicles: Architecture, Security and Challenges,” in Proceedings of the International Conference on Computational Intelligence and Communication Technology (CCICT), 2022.

8. O.Naik, N.Patel, S.A.Baba, and H.Dalvi, “Decentralized Ride Hailing System Using Blockchain and IPFS,” in Proceedings of the IEEE Bombay Section Signature Conference (IBSSC), 2022.
9. J.S.Subhan, Navaneethakrishnan, T.Avinash, and S.Thirumal, “Driver’s Safety Management System for Commercial Purposes Using IoT,” in Proceedings of the International Conference on Communication and Electronics Systems (ICCES), 2022.
10. Bhumika, D.Das, and S.K.Das, “RsSafe: Personalized Driver Behavior Prediction for Safe Driving,” in Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2022.

11. Bhumika, J.Vyas, and D.Das, "Stress-Aware Recommendation for Safe Driving Using MTL-ConvLSTM," in Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC), 2022.
12. J.Vyas, Bhumika, D.Das, and S.Chaudhury, "Dr. MTL: Driver Recommendation Using Federated Multi-Task Learning," in Proceedings of the IEEE Vehicular Technology Conference (VTC-Fall), 2023.
13. S.A.Jebamani, D.Devi, V.Saraswathi, N.Aishwarya, R. N., and K. U. A. K., "IoT-Based Vehicle Tracking and Monitoring System Using Smart Application," in Proceedings of the International Conference on Intelligent Computing, Control Engineering and Business Systems (ICCEBS), 2023.

A. Hegde, B.S.,S.H.A.,Anish, and C.T.,"METADRIVE – A Decentralized Ride Sharing App," in Proceedings of the International Conference on Computer Systems, Information Technology, and Sustainable Solutions (CSITSS), 2023.

14. M.Aamir and L.J.,"A Comparative Study of ML and DL Approaches for Twitter Sentiment Classification," in Proceedings of the International Conference on Electrical, Electronics and Computer Technologies (ICEECT), 2024.
15. G.S.,N.R.S.,G.A.Senthil,and A.K.B.,"Future Navigation Demand Trends: Accurate Ride Request Forecasting Optimization Using Machine Learning," in Proceedings of the International Conference on Smart Technologies, Systems and Next Generation Computing (ICSTSN), 2024.
16. M.N.Kumar, S.Madhulitha, G.S.Vardhan, and S.N.Reddy, "TravelX," in Proceedings of the IEEE India Council International Subsections Conference (INDISCON), 2024.
17. P.Jain, U.Raut, and T.Korulkar, "Travel Tinder: A Comprehensive Literature Review on Innovations in Solo Travel Companion Matching," in Proceedings of the IEEE International Conference on ICT for Business, Industry and Government (ICTBIG), 2024.