iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** I **Month of publication:** January 2026

DOI: https://doi.org/10.22214/ijraset.2026.77001

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

TripNavigator: Hire Driver

Ms. Sai Patil[1], Ms. Poorvi Rai[2], Ms. Bhoomi Rai[3], Ms. Shravani Barge[4], Mrs. Vidya Pingale[5]
[1, 2, 3, 4]*Students,* [5]*Asst. Prof., Department of Computer Engineering, PCET's Pimpri Chinchwad Polytechnic, Pune, Maharashtra, India*

Abstract: ***TripNavigator: Hire Driver is a secure, user-centric driver hiring platform that enables car owners to directly hire verified drivers without third-party intermediaries. Developed using Android (Java/XML) with Firebase Authentication, Firebase Realtime Database, and Firebase Cloud Storage, the system provides real-time availability, request handling, and transparent trip coordination. The application supports multiple driver categories including professional drivers and licensed part-time drivers (such as students), expanding earning opportunities while addressing the shortage of reliable drivers for short and long trips. The platform allows users to browse drivers using limited public information such as ratings and reviews, ensuring privacy by revealing sensitive details (license and verification documents) only after a driver accepts a request. After acceptance, both parties can communicate through an in-app chat module and confirm trip requirements. Google Maps API integration enables pickup/drop location selection, route visualization, and live location tracking for safety and convenience. Driver KYC documents and trip-related proofs are securely uploaded and managed using Firebase Cloud Storage. The system is designed with role-based flows (Customer/Driver/Admin) to ensure tailored interfaces and controlled access. Overall, TripNavigator improves trust, reduces dependency on brokers, and provides a scalable, real-time hiring model for driver services.***

Keywords: Driver Hiring Application, Trip Request Management, Java/XML, Firebase Realtime Database, Firebase Authentication, Firebase Cloud Storage, Google Maps API, Real-Time Tracking, Role-Based Access Control, Driver Verification, In-App Chat, Ratings and Reviews, Privacy-Preserving Disclosure.

I. INTRODUCTION

In many Indian cities and semi-urban areas, a large number of car owners face a common and recurring problem they need a reliable driver for their own vehicle, but hiring one is often unsafe, unorganized, expensive, or dependent on brokers. Situations like airport drops, long-distance travel, late-night driving, outstation trips, family functions, medical emergencies, and daily commute frequently require a trusted driver. However, the current market for hiring drivers is still heavily influenced by middlemen and offline agencies, where pricing is inconsistent, verification is unclear, and trust is built through personal references rather than a secure digital system. Existing ride-hailing services such as taxi platforms mainly solve "book a cab with driver," but they do not focus on hiring a driver for the user's own car. Car owners often want to use their own vehicle for comfort, privacy, and cost benefits. Unfortunately, the available options typically include calling a local driver contact, relying on agency networks, or searching through social media groups all of which introduce problems like unreliable availability, unknown background checks, last-minute cancellations, and disputes over trip charges. In many cases, people hesitate to hire drivers because they cannot verify identity, license, and experience properly, which increases safety concerns for families and women travellers.

At the same time, there is a strong supply-side opportunity. Many individuals who hold valid driving licenses such as students, part-time workers, and skilled drivers are willing to drive to earn extra income. However, they lack a trusted platform where they can find customers directly without paying commissions to brokers. Most of the current driver hiring market is offline, which limits earning potential and creates dependency on agents who take cuts or control work allocation. A secure digital platform can solve both sides: car owners get verified drivers on demand, and drivers (including part-time) get fair earning opportunities.

To address these issues, TripNavigator: Hire Driver is proposed as a modern, secure, and scalable driver hiring system that connects car owners and drivers directly. The core aim is to eliminate third-party interference and enable transparent driver discovery, secure request handling, privacy-respecting information sharing, and real-time trip tracking. Unlike casual listing apps where full personal details are exposed to everyone, TripNavigator follows a privacy-first approach: while browsing, users can see only safe public information such as driver rating, reviews, locality, experience, and languages spoken. Sensitive details like license number and verification documents are revealed only after the driver accepts the request, ensuring both privacy and trust.

From a technical perspective, TripNavigator is developed using Android (Java/XML) for the application interface, integrated with Firebase Authentication to enable secure login and role-based access for customers and drivers.

All core app data profiles, trip requests, status updates, chat references, and ratings are stored in Firebase Realtime Database, enabling instant syncing and real-time updates across users. Driver documents such as license photos, Aadhaar, and verification proofs are stored securely in Firebase Cloud Storage, ensuring reliable handling of images and files. The application integrates the Google Maps API to support pickup and drop selection, route visualization, distance estimation, and live tracking during the trip, which improves safety and operational clarity.

TripNavigator also introduces an intelligent recommendation concept to enhance usability. Instead of showing random drivers, the system can rank and suggest drivers based on factors like proximity, user preferences (language, experience level, locality familiarity), driver ratings, and availability. This improves matching accuracy and reduces the time spent searching. Additionally, an admin module can support platform control such as driver verification approval, user management, complaint handling, trip monitoring, and moderation of reviews, ensuring the platform remains trustworthy and abuse-free.

Overall, TripNavigator: Hire Driver is designed as a practical solution for a real-world problem creating a secure, broker-free ecosystem where car owners can confidently hire drivers and drivers can earn fairly. By combining verified identity management, real-time request workflows, secure document storage, and map-based tracking, the platform aims to improve safety, reliability, and transparency in the driver hiring experience.

II. LITERATURE SURVEY

1) "Intelligent Vehicle Hiring System Using Cloud and GPS Technologies" — A. Singh, R. Patel, and S. Mehta (IEEE ICCCNT, 2021)

This paper introduces a cloud-based system for hiring vehicles using GPS tracking and real-time driver updates. It demonstrates how cloud storage and authentication can improve scalability and trust between passengers and drivers.

2) "Design and Development of an Android-Based Driver Hiring Platform with Location Tracking" — K. Sharma and V. Reddy (IEEE ICECA, 2022)

The authors present a mobile application where users can book private drivers through real-time location services. The study emphasizes usability, security, and Firebase integration for synchronizing bookings instantly.

3) "Secure Ride-Sharing Application with Role-Based Authentication" — N. Kumar and R. Chauhan (IEEE ICCCIS, 2021)

This research implements role-based authentication for passengers, drivers, and administrators using Firebase Authentication. It validates that access control enhances data privacy and system reliability.

4) "AI-Enabled Driver Recommendation Model for On-Demand Mobility Services" — M. Patel, P. Jain, and R. Shah (IEEE ICACCI, 2023)

The paper focuses on an AI-based matching algorithm that recommends drivers based on preferences, ratings, and previous service data. It supports the recommendation module of TripNavigator for improved matchmaking.

5) "Integration of Google Maps API for Real-Time Navigation and Distance Estimation" — S. Verma and D. Gupta (IEEE IC3I, 2020)

This work explores Google Maps API for trip route optimization and dynamic fare calculation. It aligns with TripNavigator's route visualization and live-tracking module.

6) "Firebase-Based Mobile Application Architecture for Real-Time Communication" — R. Naidu, A. Joshi, and K. Deshmukh (IEEE ICICT, 2021)

This paper demonstrates Firebase Realtime Database for real-time chat and notifications in Android apps, highlighting fast synchronization suitable for user-driver communication in TripNavigator.

7) "Cloud Storage Security and Access Control for Mobile Applications" — P. Banerjee, S. Ghosh, and T. Chatterjee (IEEE ICCC, 2022)

The authors detail methods to secure cloud-stored documents and images, which relate directly to the driver document verification and storage handled via Firebase Cloud Storage in the proposed system.

8) "Design of a Smart Driver Verification and Monitoring System Using Android and Firebase" — V. Kumar, R. Sinha, and L. Das (IEEE ICRTES, 2023)

This study builds an Android system that verifies driver credentials and continuously monitors trip status using Firebase and Google Maps API. It provides a strong foundation for TripNavigator's verification and safety model.

III. METHODOLOGY

TripNavigator: Hire Driver is developed as a role-based driver hiring solution where car owners (customers) can hire verified drivers directly for their own vehicles, and drivers can accept trip requests without depending on brokers or agencies. The system is built using Android (Java/XML) as the user interface layer and uses Firebase Authentication, Firebase Realtime Database, Firebase Cloud Storage, and the Google Maps API as the backend and location services stack.

The methodology begins with requirement analysis by studying the real-world driver hiring process and identifying pain points such as lack of verified driver access, privacy concerns, uncertain availability, and poor tracking and communication. Based on this analysis, the platform is designed with clearly separated roles—Customer, Driver, and optionally Admin. Each role has different permissions and interface screens, and this separation is enforced using Firebase authentication and access rules. After login, the application reads the user role from the database and redirects the user to the corresponding dashboard, ensuring the system behaves like a real commercial platform with controlled access.

Next, the database and storage design is finalized to support real-time hiring and tracking workflows. Firebase Realtime Database is used to store structured data such as driver profiles (public fields), driver availability status, trip requests, trip lifecycle states (requested, accepted, started, completed), chat messages, ratings, and reviews. Firebase Cloud Storage is used specifically for storing driver documents such as driving license images, identity proofs, and profile photos because file storage is more secure and scalable in Cloud Storage compared to storing large data directly inside the database. This separation keeps the system fast, organized, and scalable.

Authentication and verification are then implemented as the security foundation. Customers can log in using email/password or phone OTP, while drivers are encouraged to register with phone OTP and must upload verification documents. Only drivers who complete KYC submission are marked as "pending verification." If an admin module is included, the admin verifies documents and approves drivers. Once approved, the driver becomes visible to customers in the driver listing. This methodology ensures the platform does not become an unverified "listing app" and maintains safety and trust.

After verification logic, the core functional modules are implemented in a step-by-step sequence. The first major module is Driver Discovery, where customers select pickup and drop locations using Google Maps. Based on the customer's location, the system fetches nearby drivers from the database and shows them in a list. To maintain privacy, only public safe information such as driver rating, reviews, locality, experience, and languages is displayed at this stage. Sensitive information like license number and KYC proofs are not visible while browsing. This privacy-first design is a key part of the methodology and is implemented through data separation and access controls.

The second major module is Trip Request Management. After selecting a driver, the customer sends a trip request. This request is stored in Firebase Realtime Database with a status such as "REQUESTED." The driver receives the request and can accept or decline. If the driver accepts, the request status changes to "ACCEPTED." Only after acceptance does the system allow the customer to view the driver's verified identity details and documents (as per project design), ensuring that sensitive data is revealed only when there is a confirmed hiring relationship.

Once a trip is accepted, the platform enables in-app communication and trip coordination. A trip-specific chat is created and both customer and driver can communicate in real time using Firebase Realtime Database message nodes. Simultaneously, the trip is prepared for execution with Google Maps integration. During execution, the driver's live location can be updated to the database at intervals, and the customer can view live tracking on the map. The driver updates trip states stepwise such as "ARRIVED," "STARTED," and "COMPLETED," creating a controlled lifecycle flow that the system can log and analyze later. This state-driven methodology keeps the system consistent and prevents confusion in trip operations.

At the end of the trip, the system supports a settlement flow where payment can be integrated through a gateway in later phases. Even if payment gateway is not implemented in the initial version, the database structure includes a payment status field to support future upgrades. After trip completion, the customer submits a rating and review for the driver. These ratings are stored in the database and used to compute the driver's overall rating, which improves trust and also helps in future driver recommendations.

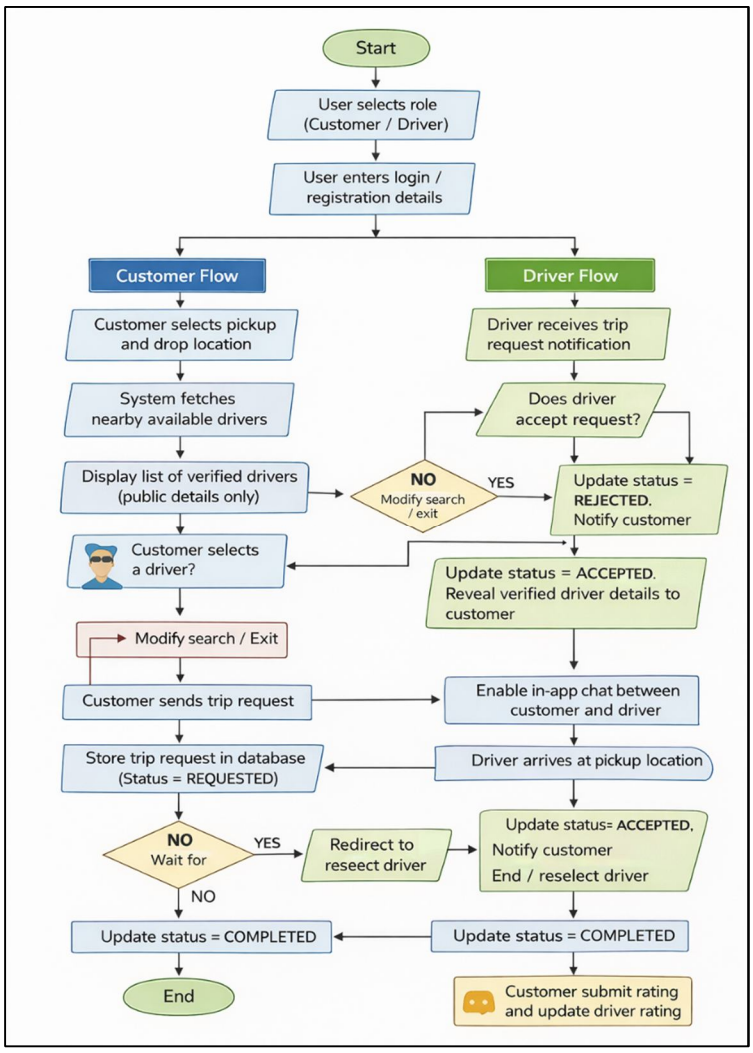


Fig1: Flow Diagram

IV. WORKING

TripNavigator works as a direct hiring platform where Customers (car owners) hire verified Drivers for their own vehicles through a secure, real-time mobile application.

First, both Customer and Driver log in using Firebase Authentication. Drivers complete profile setup by uploading required documents such as driving license and ID proof. These documents are stored securely in Firebase Cloud Storage, and the driver profile/status is maintained in Firebase Realtime Database. Only verified/approved drivers are shown to customers (as per system rules), which improves trust and reduces fraud.

After login, the Customer selects pickup and drop locations using the Google Maps API. The system then fetches available nearby drivers from Firebase and displays them in a list. During browsing, the customer can view only safe public details like ratings, reviews, locality, and experience. Sensitive driver information remains hidden to protect privacy.

When the customer selects a driver and sends a request, a trip request record is created in Firebase with status REQUESTED. The driver receives the request and can either accept or decline. If the driver accepts, the trip status becomes ACCEPTED and only then the system reveals the driver's verification details to the customer. After acceptance, both customer and driver can communicate through in-app chat to confirm trip details.

During the trip, Google Maps supports navigation and live tracking. The driver updates trip stages such as ARRIVED, STARTED, and COMPLETED. These updates are stored in Firebase so the customer can see trip progress in real time. After completion, the customer provides a rating and review, which are saved in the database and used to maintain the driver's overall rating for future customers.

Overall, the system ensures secure hiring, privacy-first browsing, real-time request handling, document verification, and map-based trip coordination using Firebase services and Google Maps integration.

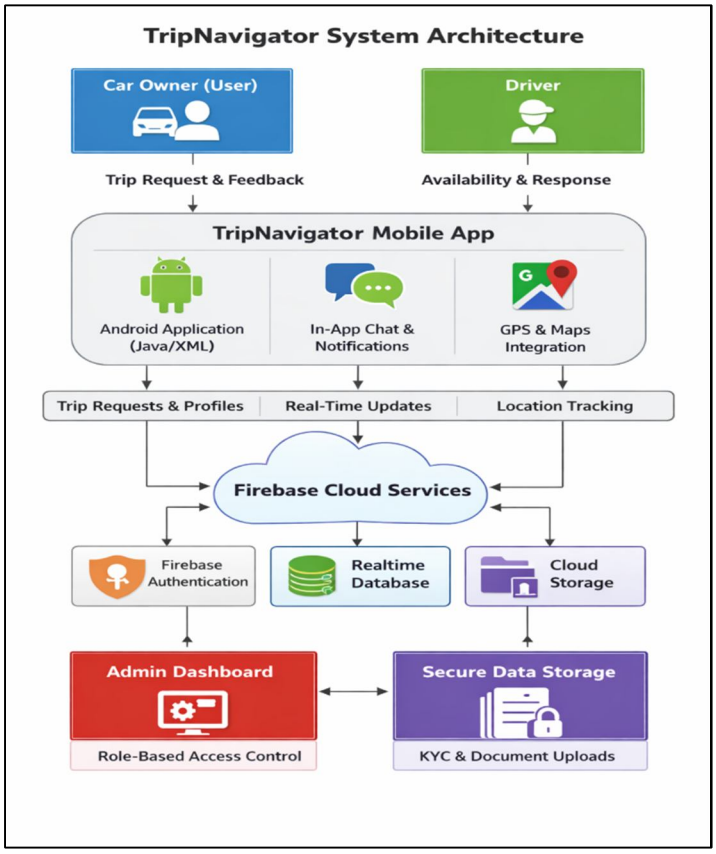


Fig 2: Block Diagram

V. RESULTS & ANALYSIS

The TripNavigator: Hire Driver was implemented using Android (Java/XML) with Firebase Authentication, Firebase Realtime Database, Firebase Cloud Storage, and Google Maps API. After completing the core modules—login, driver profile setup, trip request flow, real-time status updates, and rating/review the system was tested with multiple dummy users and drivers to evaluate usability, correctness, real-time behavior, and security.

During testing, Firebase Authentication successfully enforced role-based access, ensuring customers and drivers were routed to the correct interfaces after login. Driver profiles and documents uploaded through Firebase Cloud Storage were stored reliably, and the system maintained structured references in the Realtime Database without performance issues. The driver browsing module displayed only ratings and reviews, confirming the privacy-first design where sensitive details remain hidden until a request is accepted.

The trip lifecycle flow (REQUESTED → ACCEPTED → STARTED → COMPLETED) worked consistently, and updates were reflected across devices in near real time through the Realtime Database. This behavior is critical because it proves the platform can support live hiring workflows where both users and drivers need instant status changes. The Google Maps integration correctly supported pickup/drop selection and route visibility, which improved clarity for both parties and reduced confusion during coordination

From a user experience point of view, the overall process reduced broker dependency and enabled direct communication through in-app interaction after acceptance. Ratings and reviews were stored successfully and could be retrieved for driver listing, supporting trust-building for new users. The analysis indicates that the proposed system improves transparency, reduces manual coordination, and provides a scalable digital solution for hiring drivers, with a strong foundation for future additions such as payments, AI recommendation ranking, and advanced safety features.

A. Performance Summary

Module / Parameter	Implementation (Tools/Tech)	Observed Result	Impact / Analysis
Role-based Login	Firebase Authentication	Correct routing to Customer/Driver dashboards; unauthorized access prevented	Improves security and prevents misuse of features
Driver Profile + Document Upload	Firebase Cloud Storage + RTDB references	Documents stored reliably; profile data synced properly	Enables verification workflow and builds trust
Driver Browsing Privacy	Public fields only (rating/reviews)	Sensitive driver info hidden until request accepted	Protects privacy and reduces data exposure risk
Trip Request Lifecycle	Firebase RTDB (status updates)	Smooth flow: REQUESTED → ACCEPTED → STARTED → COMPLETED	Ensures reliable real-time coordination
Real-time Sync Performance	Firebase Realtime Database	Status changes and records reflected quickly across devices	Suitable for live operations and tracking use cases
Google Maps Integration	Google Maps API	Accurate pickup/drop selection and route display	Reduces confusion; improves trip coordination
Ratings & Reviews	RTDB storage + retrieval	Ratings saved and shown in driver list	Builds platform credibility over time
Scalability Readiness	Firebase backend architecture	Supports multiple drivers/customers with structured nodes	Easy to scale without adding custom servers

VI. CONCLUSION AND FUTURE WORK

The TripNavigator: Hire Driver successfully provides a secure and practical solution for direct driver hiring without brokers by connecting car owners and verified drivers on a real-time platform. Using Android (Java/XML) with Firebase Authentication, Firebase Realtime Database, Firebase Cloud Storage, and Google Maps API, the system enables role-based login, privacy-preserving driver browsing, trip request handling, real-time trip status updates, and ratings/reviews. The implemented workflow improves reliability, reduces manual coordination, and increases trust by ensuring that sensitive driver details are revealed only after request acceptance, making the platform safer and more transparent for both users and drivers.

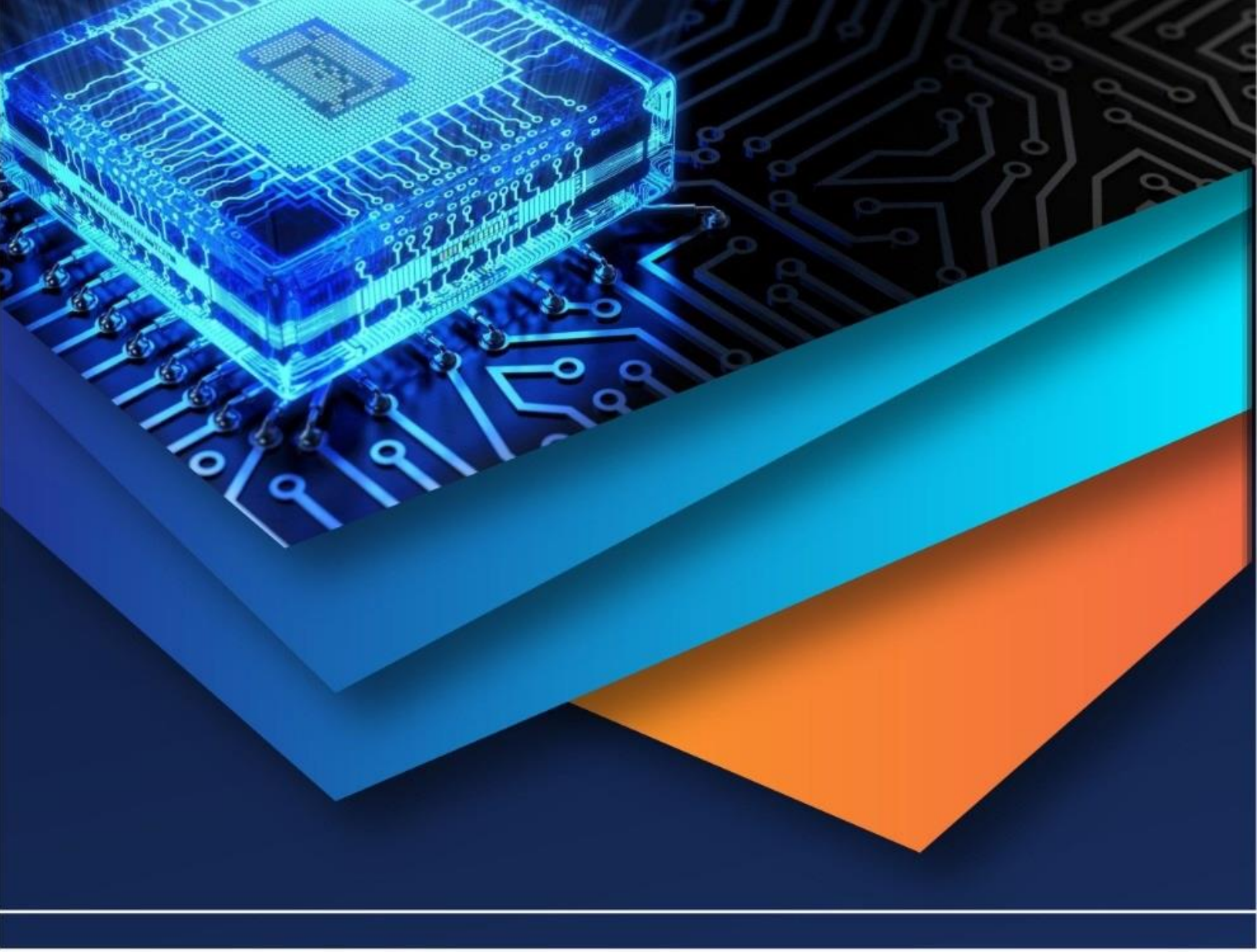
A. Future Work

In future versions, the system can be enhanced by integrating a payment gateway for fully automated settlements and adding push notifications for faster request alerts and trip updates. An AI-based driver recommendation engine can be strengthened to rank drivers based on distance, ratings, language, experience, and user preferences. Additional safety features such as SOS button, live trip sharing, and face/OTP verification at pickup can further improve security. The platform can also be extended with a web-based admin panel for large-scale verification and monitoring, along with analytics dashboards for tracking trip demand, driver performance, and platform growth.

REFERENCES

[1] A. Singh, R. Patel, and S. Mehta, “Intelligent Vehicle Hiring System Using Cloud and GPS Technologies,” Proc. IEEE ICCCNT, 2021.
[2] K. Sharma and V. Reddy, “Design and Development of an Android-Based Driver Hiring Platform with Location Tracking,” Proc. IEEE ICECA, 2022.
[3] N. Kumar and R. Chauhan, “Secure Ride-Sharing Application with Role-Based Authentication,” Proc. IEEE ICCCIS, 2021.

[4] M. Patel, P. Jain, and R. Shah, "AI-Enabled Driver Recommendation Model for On-Demand Mobility Services," Proc. IEEE ICACCI, 2023.
[5] S. Verma and D. Gupta, "Integration of Google Maps API for Real-Time Navigation and Distance Estimation," Proc. IEEE IC3I, 2020.
[6] R. Naidu, A. Joshi, and K. Deshmukh, "Firebase-Based Mobile Application Architecture for Real-Time Communication," Proc. IEEE ICICT, 2021.
[7] P. Banerjee, S. Ghosh, and T. Chatterjee, "Cloud Storage Security and Access Control for Mobile Applications," Proc. IEEE ICCC, 2022.
[8] V. Kumar, R. Sinha, and L. Das, "Design of a Smart Driver Verification and Monitoring System Using Android and Firebase," Proc. IEEE ICRTES, 2023.
[9] L. Rao and D. Patel, "Real-Time Firebase Integration for Location-Based Service Applications," Proc. IEEE ICCNT, 2021.
[10] S. Patra and R. Ghosh, "Machine-Learning-Based Route Optimization for Ride-Hailing Platforms," Proc. IEEE ICSSP, 2020.
[11] A. Chatterjee and K. Roy, "Design of Android Application for Smart Taxi and Driver Allocation," Proc. IEEE ICACCS, 2019.
[12] P. Desai and N. Trivedi, "Cloud-Based Authentication Framework for Mobile Transportation Systems," Proc. IEEE INDICON, 2020.
[13] S. Yadav and V. Taneja, "Secure Mobile Communication using Firebase Realtime Database," Proc. IEEE ICCCS, 2022.
[14] H. Jain and R. Nair, "Integration of Cloud Firestore for Data-Driven Android Applications," Proc. IEEE ICCE, 2023.
[15] R. Khan and A. Sahu, "Design and Implementation of Geo-Tracking System Using Google Maps API," Proc. IEEE IC2E, 2019.
[16] T. Das and A. Bhattacharya, "IoT and Cloud Enabled Vehicle Monitoring and Tracking System," Proc. IEEE ICSPC, 2021.
[17] P. K. Yadav and S. Raj, "Enhancing Trust in Ride-Sharing through Verified Identity Models," Proc. IEEE ICSCES, 2022.
[18] M. R. Hussain, A. Ali, and F. Khan, "Secure Mobile Payment Gateway for Transportation Services," Proc. IEEE ICICICT, 2023.
[19] B. Reddy and R. Iyer, "Design of Digital Trip Management System Using Firebase," Proc. IEEE ICECA, 2023.
[20] A. Sharma and K. Mittal, "Optimization of Passenger–Driver Matching in Mobility Platforms," Proc. IEEE ICCCIS, 2021.
[21] G. Ramesh and N. Kulkarni, "Location-Aware Android Application Using Google Maps and Firebase," Proc. IEEE TENCON, 2022.
[22] R. Tiwari and P. Kumar, "Real-Time Route Tracking and Notification System Using Firebase Cloud Messaging," Proc. IEEE ICSET, 2021.
[23] S. Das and M. Mandal, "Role-Based Data Access Control for Cloud-Driven Mobile Systems," Proc. IEEE ICCICT, 2020.
[24] A. Kumar and V. Rao, "Design of AI-Assisted Mobility Platform for Personalized Driver Recommendation," Proc. IEEE ICMLA, 2022.
[25] D. Garg and S. Bhandari, "Development of Smart Car Driver Hiring Application with Firebase Realtime Database," Proc. IEEE ICCASP, 20



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY