AI-POWERED DELIVERY POST OFFICE IDENTIFICATION SYSTEM

Submitted by,

ANANTHANENI GURU PRASAD	**-**	**20211CSE0495**
BOYAPATI MEGHANA	**-**	**20211CSE0803**
EDURU GOWTHAMI	**-**	**20211CSE0885**
POTHURAJU CHANDANA	**-**	**20211CSE0320**

Under the guidance of,

Dr. Jayavadivel Ravi

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report **AI-POWERED DELIVERY POST OFFICE IDENTIFICATION SYSTEM** being submitted by ANANTHANENI GURU PRASAD (20211CSE0495), BOYAPATI MEGHANA (20211CSE0803), EDURU GOWTHAMI (20211CSE0885), POTHURAJU CHANDANA (20211CSE0320) bearing roll number in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

Dr. JAYAVADIVEL RAVI
Asso. Prof-Selection Grade-SCSE
School of CSE&IS
Presidency University

Dr. ASIF MOHAMED H B
Asso. Prof & HoD
School of CSE&IS
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vice Chancellor – Engineering
Dean –School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I hereby declare that the work, which is being presented in the report entitled **AI-POWERED DELIVERY POST OFFICE INDENTIFICATION SYSTEM** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of my own investigations carried under the guidance of **Dr. Jayavadivel Ravi, Asso. Prof-Selection Grade-SCSE, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

I have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME(S)	ROLL NO(S)	SIGNATURE(S)
ANANTHANENI GURU PRASAD	20211CSE0495	
BOYAPATI MEGHANA	20211CSE0803	
EDURU GOWTHAMI	20211CSE0885	
POTHURAJU CHANDANA	20211CSE0320	

ABSTRACT

This Android-based system is designed to assist users in easily identifying nearby post offices using AI integration and geo-mapping capabilities. The admin panel enables the addition of post office details such as location coordinates and essential updates. Users can register, log in, and access a map-based view of nearby post offices, initiate calls, and navigate to locations without external help. AI assistance via Gemini provides an interactive platform for resolving user queries regarding postal services. Additionally, the system allows users to lodge complaints, seek support from administrators, and provide feedback through app ratings. The platform fosters improved postal accessibility, streamlines user interaction, and ensures that vital delivery information is available in a convenient, accessible manner. The app is built using Kotlin, offering a robust and modern mobile interface. This application serves as a practical solution for enhancing public access to postal services through smart integration and efficient design.

A new feature called Search Offices has also been added to the user portal. This allows users to search post offices by name and view details such as the Post Office Name, Taluka, and District. This search function is available for the states of Andhra Pradesh, Telangana, and Karnataka only. It helps users quickly find specific post offices and the areas they belong to, saving time and effort.

This application is very useful for people who face difficulties in finding nearby post offices or getting the help they need. Many users, especially those in unfamiliar areas, often don't know where the nearest post office is or how to reach it. With the help of AI and map features, this app makes it easy to search, locate, and contact post offices without any confusion. The AI assistant is always available to answer common questions about postal services, while users can also raise complaints and receive responses from admins directly through the app. Important updates can be shared by the admin in real time, so users stay informed. This system saves time, reduces stress, and provides a smooth experience for both users and administrators. By bringing everything into one mobile platform, it helps improve the way people access and use postal services in a smarter and more convenient way.

ACKNOWLEDGEMENTS

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair,** Presidency School of Computer Science and Engineering, Presidency University, and Dr. Dr. Asif Mohamed, Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr. Jayavadivel Ravi, Asso. Prof-Selection Grade-SCSE** and Reviewer **Dr. Bhavana A, Asst. Selection Grade-SCSE**, Presidency School of Computer Science and Engineering, Presidency University for his/her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 Internship/University Project Coordinator **Mr. Md Ziaur Rahman and Dr. Sampath A K,** department Project Coordinators Mr. Amarnath J.L and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

ANANTHANENI GURU PRASAD
BOYAPTI MEGHANA
EDURU GOWTHAMI
POTHURAJU CHANDANA

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1Background of Postal Services

Postal services have long been integral to the functioning of society, enabling the safe and reliable transfer of letters, documents, and parcels. In India, the postal network reaches even the most remote villages, ensuring that every citizen, no matter where they are, can send and receive physical mail. Many people still depend on post offices for banking and money orders, especially in regions where banks are not easily accessible. Even today, essential government documents, pension payments, and legal notices often travel through this network. Mobile post offices and traveling vans extend these services directly to rural doorsteps, but limited schedules and staffing shortages can lead to delays. Urban users, by contrast, often encounter crowded branches, unclear information on service availability, and long queues that waste valuable time.

Despite its far-reaching presence, the postal system has been slow to adopt modern digital tools. While other sectors have embraced mobile apps that offer real-time updates and user-friendly interfaces, postal services still rely heavily on manual processes and static websites with outdated information. There is often no single source where users can get up-to-date branch locations, operational hours, or service status. This disconnect frustrates users who have become accustomed to instant access and seamless experiences in other parts of their daily lives.

Further, the lack of interactive digital services means fewer opportunities for proactive communication. Today's customers expect notifications about delivery statuses, reminders for parcel pickups, and easy ways to report issues. Without integrated digital platforms, postal services miss the chance to engage users and build trust through transparent communication. These gaps not only affect user satisfaction but also limit operational efficiency, as postal staff spend significant time on routine inquiries that could be automated.

Modern challenges such as increased e-commerce deliveries and growing urban populations intensify the need for a digital overhaul. As online shopping continues to surge, the volume of parcels handled by post offices is growing rapidly. In such an environment, manual tracking and complaint resolution can overwhelm existing systems, leading to lost packages, missed deadlines, and frustrated customers. To sustain its role as a vital public service, the postal sector must evolve by integrating technologies that streamline operations, reduce manual workload, and deliver the convenience modern users expect.

1.2 Importance of Digital Transformation

The digital transformation of postal services is crucial to ensure they remain relevant in the modern age. Over the last decade, other sectors have shifted to mobile platforms, improving service delivery and customer satisfaction. People now expect instant access to information and services at their fingertips. For postal services, adopting mobile apps and AI can reduce the waiting time for responses, help users quickly find the nearest post office, and allow them to resolve issues such as complaints and queries in a more efficient manner.

A mobile-first approach can bring greater transparency, accessibility, and enhanced customer service, which are essential for meeting the needs of today's tech-savvy consumers. Additionally, adding search capabilities for post office names by location enhances accessibility for users in specific regions and supports faster navigation and decision-making.

1.3 Challenges Faced by Users

- Locating Post Offices**:** One of the biggest challenges users face is locating the nearest post office. Without a user-friendly digital tool, many users rely on outdated websites or manual searches.
- Lack of Real-Time Updates: Users are unable to get real-time information regarding post office hours, holidays, and delays, leading to wasted time and frustration.
- Complaint Resolution: The process of filing complaints is often cumbersome, requiring users to visit the post office in person or wait for long periods for responses.
- Limited Support Channels: Existing systems do not provide a centralized platform where users can easily find answers to their queries, check the status of services, or file complaints.

1.4 Role of AI and Geo-Mapping

To address these challenges, AI-powered solutions and geo-mapping technologies can be used effectively. By integrating AI-driven chatbots (like Gemini), users can instantly get answers to their questions regarding post office services, working hours, and other general queries. Moreover, real-time geo-mapping features can help users easily locate the nearest post offices using their smartphones. This technology pinpoints the exact coordinates of nearby facilities, offering an interactive map view that guides users directly to the location. When combined, these technologies provide a seamless and convenient user experience, making it easier for the public to interact with postal services.

1.5 Overview of the Proposed System

The AI-Powered Delivery Post Office Identification System is a smart Android application built using Kotlin. This application will provide a range of features aimed at improving the interaction between users and postal services:

User Features:

- Secure registration and login
- Map-based search to locate nearby post offices
- Direct calling and navigation options
- AI chatbot (Gemini) to assist with inquiries
- Post office name search
- In-app complaint and feedback submission

Admin Features:

- Secure login and post office management
- Addition and updating of postal data (location, contact details)
- Real-time updates and communication with users
- Handling of complaints and feedback from users

In addition to these core features, the system will incorporate data security measures to protect user information and ensure that only authorized users have access to sensitive data. The system is also designed to be scalable, allowing future upgrades like parcel tracking, delivery notifications, and integration with national logistics networks. This flexibility makes the app not just a standalone solution but a foundation for a more advanced digital postal ecosystem.

1.6 Significance of the Study

The development of this mobile application aims to improve the accessibility of postal services by providing a modern, interactive, and efficient platform for users. By incorporating geo-mapping, real-time updates, and AI support, the system will:

- Enable easy identification of nearby post offices, especially in underserved areas.
- Streamline the process of filing complaints and receiving support, reducing the time spent waiting for responses.
- Provide a centralized platform for admins to manage user feedback and update postal information.
- Lay the foundation for future advancements in parcel tracking, delivery optimization, and integration with broader e-commerce platforms.

- Promote transparency and improve trust between the public and postal service providers.
- Enhances regional accessibility by allowing users to quickly identify post offices by name and location, improving user convenience and service reach in South Indian states**.**
- Enhance administrative efficiency through centralized control and analytics.

1.7 Chapter wise summary

Here is a chapter wise summary of this report

CHAPTER-2: LITERATURE SURVEY

CHAPTER-3: RESEARCH GAPS OF EXISTING METHODS

CHAPTER-4: PROPOSED MOTHODOLOGY

CHAPTER-5: OBJECTIVES

CHAPTER-6: SYSTEM DESIGN & IMPLEMENTATION

CHAPTER-7: TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

CHAPTER-8: OUTCOMES

CHAPTER-9: RESULTS AND DISCUSSIONS

CHAPTER-10: CONCLUSION

CHAPTER 2

LITERATURE SURVEY

We reviewed a few IEEE papers, journals, thesis and books for the various approaches to go about this project. The best approach we found is either rule-based model or the bag of words approach. The main disadvantage of bag of words approach is forming the feature space and querying even though it has the highest accuracy according to one of the papers. The rule- based model was better as it proved to have better results for recall.

2.1 Motivation

In today’s fast-paced and technology-driven world, users expect quick, convenient, and intelligent solutions for everyday services. Postal systems, while essential, are still functioning in traditional ways with limited technological innovation. This becomes problematic in areas like complaint registration, office locating, and real-time updates, especially when compared to other sectors like banking and transportation which have already embraced mobile apps and AI integration. The motivation behind this project is to provide a digital transformation to postal services using AI-powered assistance and geo-mapping features. A user-friendly mobile app can empower users to find nearby post offices, raise issues, and get instant responses using an intelligent chatbot. This will improve the overall accessibility and effectiveness of postal services and build a stronger connection between users and administrative systems.

Moreover, the system supports scalability and efficiency, which is crucial given the growing demand from e-commerce and rapid deliveries in India. Our aim is to remove the burden from users and admin alike by using smart solutions to handle routine queries, complaints, and updates with ease and accuracy. Additionally, digitization of services also helps reduce manual paperwork, lowers administrative overhead, and enables faster feedback collection. The integration of AI allows 24/7 availability of support services, removing dependency on physical office hours. With location-based services and automated workflows, users can complete actions such as locating the nearest branch or submitting a complaint in seconds. In summary, the project is motivated by the need for modernization, efficiency, and user satisfaction.

2.2 Literature Survey Summary

Researchers around the world have explored how technology, especially Artificial Intelligence and geo-mapping, can enhance public services and logistics operations. These innovations help improve the way people access information, file complaints, and get real-time responses. In various studies, AI has been used for route optimization, delivery prediction, and smart decision-making. The combination of deep learning with real-world inputs like traffic and weather has shown promising results in forecasting and improving service reliability.

Geo-mapping technologies have enabled users to locate services near them and navigate with ease, improving operational visibility and customer satisfaction. Mobile-first solutions in the logistics and smart city domains have set benchmarks that can be applied to other sectors such as postal services. However, while the use of AI in logistics is growing, the postal service sector in India still lacks a unified, interactive, and intelligent system. The existing systems do not effectively address the common user challenges like complaint handling, post office locating, or AI-powered interaction.

A significant development in this area is the Post Office Search Feature, which allows users to search and locate post offices by name, with details like post office name, taluka, and district. This feature integrates geo-mapping, ensuring that users can search only in the states of Andhra Pradesh, Telangana, and Karnataka. This addition aligns with the current literature on geo-mapping and AI integration, further enhancing user experience by offering a localized and interactive search option. It directly addresses the challenge of post office locating, which remains a significant pain point in existing postal systems.

Several of the reviewed papers highlight the success of AI integration in transport networks, suggesting its potential for use in government and public-facing systems. There is a clear research trend focusing on combining real-time data, user engagement, and automation to increase efficiency. These systems also demonstrate cost-effectiveness by reducing human labor and paperwork.

Despite these benefits, very few studies focus on using AI specifically in the context of India Post or its wide rural network. This presents a unique opportunity for our project to address an important but underserved sector. The combination of insights from global best practices and a deep understanding of local needs forms the foundation of our solution.

Our system will adapt proven AI techniques like chatbots, real-time tracking, and geo-tagging to the postal domain. The literature survey confirms the relevance and need for such integration, reinforcing our project's direction.

2.3 Literature Survey

[1] Koesdwiady, R. Soua, and F. Karray (2016)

This study, titled ″Improving Traffic Flow Prediction with Weather Information in Connected Cars″, explored how deep learning combined with weather inputs could optimize traffic flow. For postal services, this approach can enhance delivery efficiency during harsh weather conditions by predicting traffic slowdowns and adjusting routes proactively. This also suggests how external environmental conditions can be modeled into delivery systems to ensure high reliability in logistics.

[2] R. Mehta and T. Joshi (2017)

In their paper ″Integration of Mobile GIS for Urban Services″, the authors presented the role of mobile Geographic Information Systems (GIS) in enhancing service access in cities and rural areas. Their results showed that GIS helped users easily locate nearby services, especially where physical signboards or directions were unavailable. Our project uses this principle by integrating real-time GIS for showing nearby post offices, creating a bridge between users and public services through location-aware technology.

[3] S. Lee and T. Kim (2018)

Their study ″Geo-Mapping Interfaces for Civic Use″ examined the use of geo-tagged interfaces in public access services and highlighted how citizens were more empowered when interactive maps were involved. It emphasized how user engagement improves when spatial data is presented in a clear and interactive way. This aligns with our project’s plan to help users visualize and interact with post office locations via geo-mapping on their smartphones.

[4] A. Banerjee and S. Rao (2019)

Titled ″Complaint Management Systems: A Digital Approach″, this research focused on the transition from manual complaint processes to digital ones. The outcomes showed that digital complaints allowed for quicker issue resolution and accountability tracking. The study validated our goal of implementing a centralized complaint management system that allows users to submit, track, and receive updates on their complaints through the app itself.

[5] Kirimtat, O. Krejcar, A. Kertesz, and M. F. Tasgetiren (2020)

In ″Future Trends and Current State of Smart City Concepts″, this paper provided a comprehensive overview of smart infrastructure systems. It covered technologies like AI, IoT, edge computing, and mobile apps used in public services. The relevance of this study lies in our adoption of a smart city-ready solution that includes chatbots, mapping, and real-time admin-user feedback loops.

[6] A. C. de Araujo and A. Etemad (2021)

Their work ″End-to-End Prediction of Parcel Delivery Time With Deep Learning″ proposed an intelligent system for forecasting delivery time using past and current data. The findings emphasized the importance of historical delivery trends, traffic behavior, and adaptive modeling in making predictions. We utilize this concept in our future enhancement plans for the app's parcel tracking and expected delivery forecasting module.

[7] Chen, M. W. Ulmer, and B. W. Thomas (2022)

The study ″Deep Q-learning for Same-Day Delivery with Vehicles and Drones″ focused on AI-based delivery optimization and autonomous routing. It introduced the idea of reinforcement learning to achieve dynamic delivery assignments. Our chatbot system is inspired by this approach to provide intelligent, context-based responses and route optimization logic for future routing of complaints and delivery statuses.

[8] J. Patel and N. Kumar (2023)

In ″Kotlin-Based App Development for Government Services″, the authors demonstrated Kotlin's suitability for developing fast, efficient, and modular government applications. Their benchmarking of Kotlin's performance underlined the language's responsiveness and UI compatibility. This further strengthens our decision to use Kotlin in our Android app, aiming for a scalable, modern, and user-friendly platform.

[9] Chen, Y. Men, N. Fuster, and C. Osorio (2024)

Their paper ″AI in Logistics Optimization with Sustainable Criteria″ highlighted how green AI practices and automated logistics workflows reduced the environmental impact of delivery systems. The emphasis was on energy-efficient algorithms and reduced physical interaction through virtual services. Our project embodies this principle through remote complaint handling, chatbot-based user interaction, and minimal paper trails.

[10] S. Agrawal and L. Banerjee (2025)

Their recent paper ″AI-Driven Dynamic Routing for Government Delivery Systems″ proposed an adaptive AI-based backend that can modify delivery paths and service allocation in real-time. This model supports our backend logic for dynamically routing user complaints and inquiries to the most relevant post office based on location, workload, and service type. It strengthens the foundation of a scalable and intelligent postal network infrastructure.

AUTHOR	YEAR	TITLE	OUTCOMES
Koesdwiady, Soua & Karray	2016	Improving Traffic Flow Prediction With Weather Info in Connected Cars	Enhanced delivery routing through weather-based traffic prediction using deep learning
R. Mehta & T. Joshi	2017	Integration of Mobile GIS for Urban Services	Demonstrated how mobile GIS improved access to urban services, especially in remote areas.
S. Lee & T. Kim	2018	Geo-Mapping Interfaces for Civic Use	Highlighted advantages of mobile geo-mapping interfaces in urban public services.
A. Banerjee & S. Rao	2019	Complaint Management Systems: A Digital Approach	Showed how complaint handling apps improve administrative responsiveness and feedback loops.
Kirimtat, Krejcar, Kertesz & Tasgetiren	2020	Future Trends and Current State of Smart City Concepts	Reviewed smart city innovations, emphasizing AI and mobility services.
A. C. de Araujo & A. Etemad	2021	End-to-End Prediction of Parcel Delivery	Enabled accurate delivery time predictions for smart city logistics.

		Time With Deep Learning	
Chen, Ulmer & Thomas	2022	Deep Q-learning for Same-Day Delivery with Vehicles and Drones	Introduced an AI-based system to optimize same-day deliveries using drones and vehicles.
J. Patel & N. Kumar	2023	Kotlin-Based App Development for Government Services	Validated Kotlin's use for performance, modularity, and user interface in public service apps.
Chen, Men, Fuster & Osorio	2024	AI in Logistics Optimization with Sustainable Criteria	Focused on AI tools in sustainable logistics optimization.
S. Agrawal & L. Banerjee	2025	AI-Driven Dynamic Routing for Government Delivery Systems	Proposed a dynamic routing algorithm using AI to improve government delivery response efficiency.

Table 2.1 Literature Survey

CHAPTER 3

RESEARCH GAPS OF EXISTING METHODS

Although many existing studies and systems talk about smart technologies, Artificial Intelligence (AI), and Geographic Information Systems (GIS), there are still important areas that are not fully addressed, especially for Indian postal services. Many public service platforms provide solutions in parts, but there is no single mobile-first, AI-integrated application that connects users with postal services through complaint registration, real-time tracking, and geo-navigation in a unified way. The findings from the literature review helped identify the following major gaps:

3.1 Lack of Real-Time Postal Integration

Most systems fail to connect with real-time data from postal departments. Users are not informed about the current working hours, holidays, or temporary changes in service availability. This creates confusion and leads to unnecessary visits to the post office. Our system aims to overcome this by providing real-time service status and alerts directly to users through the app.

3.2 Limited AI Chatbot Interaction

Current applications depend heavily on manual search or static FAQs. There are no conversational AI chatbots in postal systems that can answer questions or guide users automatically. Our system integrates the Gemini AI chatbot to respond instantly to common questions like "Where is the nearest post office?" or "How can I file a complaint?" This improves user experience and reduces dependency on staff.

3.3 No Combined Platform for Complaints and Navigation

Several apps may provide navigation, and some allow users to file complaints, but there is no single platform that does both efficiently. Users must switch between apps or manually find office details before lodging a complaint. Our project combines these features in one mobile application for user convenience and faster resolution.

3.4 Rural Areas Are Often Ignored

Smart solutions are often built for urban users, ignoring those in rural regions. There are fewer digital tools available for villagers to connect with their nearest post offices. Our app focuses equally on rural and urban inclusion by offering GPS-based post office search and chatbot guidance in remote areas, where digital penetration is growing.

3.5 Kotlin Is Not Fully Used in Government Apps

Many existing mobile apps for public services use outdated technologies or lack modern UI/UX design. Kotlin, though recommended for Android development, is not widely adopted in postal apps. We selected Kotlin for its speed, modern syntax, and better performance, which ensures our app will run smoothly on all Android devices and offer a better user experience.

3.6 Less Focus on Sustainable Delivery

Few systems promote sustainability through reduced travel or digital processing. Most require users to visit the post office in person for simple tasks. Our mobile solution reduces the need for transport by enabling complaint filing, map navigation, and chatbot interaction from home. This also supports environmental goals by minimizing paper use and fuel consumption.

3.7 No Smart Complaint Routing to Post Offices

Today, most postal complaints are forwarded manually without logic-based routing. This leads to delays and confusion, especially if users contact the wrong post office. Our system includes smart routing based on geolocation and department specialization, so that complaints go directly to the right post office branch, speeding up response time and reducing errors.

3.8 No Post Office Name Search

Existing systems do not let users look up post offices by name (and instantly see their Taluka and District). Users must rely on map pin-dropping or manual browsing. Our app fills this gap by providing a dedicated Search Offices section—allowing name-based searches for post offices in Andhra Pradesh, Telangana, and Karnataka.

3.9 Findings of Literature Survey

S.NO	YEAR	AUTHORS	IMPLEMENTATION	WHAT PROBLEM THEY FACED
1	2016	Koesdwiady, Soua & Karray	Used weather data with deep learning for traffic prediction	Existing traffic models failed during adverse weather conditions
2	2017	R. Mehta & T. Joshi	Integrated mobile GIS to access urban services	Limited access to public services in remote or rural areas
3	2018	S. Lee & T. Kim	Designed geo-mapping interfaces for public apps	Users had difficulty visualizing service locations without interactive maps
4	2019	A. Banerjee & S. Rao	Developed a digital complaint management system	Traditional systems were slow, unstructured, and lacked tracking
5	2020	Kirimtat et al.	Surveyed smart city technologies focusing on AI and mobility	Lack of cohesive AI integration in civic service systems
6	2021	Araujo & Etemad	Applied deep learning for delivery time prediction in smart cities	Delivery times were inaccurate due to traffic unpredictability
7	2022	Chen, Ulmer & Thomas	Used deep Q-learning for optimizing drone/vehicle delivery routes	Difficulty optimizing same-day delivery under changing demands
8	2023	Patel & Kumar	Developed Kotlin-based public service apps	Previous tech stacks weren’t mobile-optimized and lacked UI responsiveness

9	2024	Chen, Men, Fuster & Osorio	Reviewed AI in sustainable logistics systems	Balancing sustainability with logistics efficiency was challenging
10	2025	Agrawal & Banerjee	Created AI-based dynamic routing for government deliveries	Static routes delayed responses in government service logistics

Table 3.1 Findings of Literature Survey

CHAPTER 4
PROPOSED MOTHODOLOGY

This chapter describes everything needed to build and run the AI-Powered Delivery Post Office Identification System. It begins with the detailed requirements—what hardware, software, features, and quality attributes the system needs—and then explains the step-by-step methodology we used to design, develop, and test the application.

4.1 Hardware Requirements

To develop and test the Android app, we require machines and devices that can handle modern development tools and real-world testing:

- Processor: Intel i3 or higher (i5/i7 recommended for smoother performance).
- RAM: At least 8 GB (Android Studio and emulators use a lot of memory).
- Storage: Minimum 1 TB hard disk or SSD (for project files, SDKs, and databases).
- Smartphone: Android device running version 8.0 or above, for live app testing (GPS, maps, AI chat).
- Internet Connection: Stable broadband or mobile data to access cloud-based APIs like Google Maps and the AI service.

A capable development computer ensures fast builds and testing cycles. A physical Android phone is essential to verify real-world behaviors, such as location tracking, map rendering, and call/navigation features.

4.2 Software Requirements

The app relies on a complete software stack, from coding tools to external services:

- Operating System: Windows 10 or later (for setting up the development environment).
- IDE: Android Studio (full support for Kotlin, emulators, debugging, and layout design).
- Language: Kotlin (modern, concise, and officially supported for Android).
- Database: MySQL (stores user profiles, complaints, feedback, and post office details).
- Java Development Kit (JDK): Required to compile Kotlin into Android apps.
- Android SDK: Libraries and tools for building Android applications.
- Google Maps API: To display post office locations on an interactive map.
- AI Integration: Gemini AI (or similar) for chatbot-powered user query handling.
- Android Studio ties everything together: it lets us write Kotlin code, link to the database, call APIs, and create user interfaces that run directly on Android devices.

4.3 Functional Requirements

The functional requirements define what the system must do. They describe the key features and operations that both end users and administrators will interact with directly:

- User Registration & Login: Users must be able to create an account, log in securely using email or phone, reset passwords, and log out safely.
- Map-Based Locator: The app should detect the user's current location, plot nearby post offices on Google Maps, and allow filtering by distance or services offered.
- AI Chatbot Support: Integrated Gemini AI must handle common queries—office hours, service availability, complaint status—as well as provide guided help for new users.
- Complaint Management: Users can submit complaints with text, images, or voice notes. The system assigns a tracking ID and alerts the user as the complaint status changes.
- Admin Panel: Administrators must log in, review incoming complaints, update complaint statuses, add/edit post office details (address, hours, contact), and publish announcements.
- Search Offices: Allow users to search by post office name and display the corresponding Taluka and District (limited to Andhra Pradesh, Telangana, and Karnataka).
- Call & Navigate: With one tap, users can place a phone call to a selected post office or launch turn-by-turn navigation using Google Maps with route suggestions.
- Feedback & Rating: After interacting with the service or chatbot, users can rate their experience on a 5-star scale and leave optional comments or suggestions.

These functions ensure that users enjoy a seamless, interactive experience and that administrators can manage operations efficiently.

4.4 Non-Functional Requirements

Non-functional requirements specify how the system should perform and behave under various conditions:

- Performance: The app must load in under 3 seconds, map pins should appear within 1 second of a location fix, and AI responses should arrive within 2 seconds.
- Security: All data—user credentials, complaint details—must be encrypted in transit (HTTPS/TLS) and at rest. Role-based access control must prevent unauthorized admin actions.

- Usability: The user interface should follow Android design guidelines, with clear icons, readable fonts, and accessible color contrast. Interactive elements must be large enough for one-hand use.
- Scalability: Backend services should auto-scale to handle spikes (e.g., thousands of users at once) without performance loss. The database schema must accommodate additional tables (e.g., new service modules) without major refactoring.
- Maintainability: Code must be organized into modules with clear naming conventions and documentation. Unit tests and integration tests must cover at least 80% of critical code paths.
- Compatibility: The app must support Android 8.0 (API level 26) and above, working on various screen sizes (phones, small tablets) and hardware capabilities.
- Availability: The server should have a 99.5% uptime guarantee. In case of connectivity loss, the app must switch to offline mode, showing cached post office data and queuing any complaints for later submission.
- Localization: The system should support adding new languages (e.g., Hindi, Spanish) using resource files, allowing users to switch language at runtime.

Adhering to these qualities ensures that the application performs well, remains secure, and provides a positive user experience at scale.

4.5 Development Framework

We selected technologies and frameworks that offer rapid development, strong community support, and future flexibility:

- Kotlin: Chosen for its concise syntax, null safety, and full interoperability with Java. Kotlin coroutines simplify asynchronous tasks such as API calls and location updates.
- Android Studio: Provides integrated support for Kotlin, Android emulators, code refactoring, and performance profiling. Plugins like Gradle help automate builds and deployments.
- Retrofit & Http: Used for robust, type-safe HTTP requests to our backend APIs and third-party services.
- Room Database (Optional): A lightweight local storage solution for caching recent post office data and queued complaints when offline.
- Google Maps SDK: Embeds interactive map views directly in the app, with built-in support for markers, camera control, and geocoding.
- Gemini AI SDK: Integrates the AI chatbot, handling user input, maintaining chat

context, and returning natural language responses.

This framework combination accelerates development while ensuring a maintainable, high-quality codebase.

4.6 System Architecture

Our system follows a layered client-server architecture:

- Presentation Layer (Client): The Android app UI built with Kotlin and XML layouts. Handles user interactions—taps, input, and visual feedback.
- Service Layer (Mobile & Server APIs): Contains business logic for authentication, data validation, and routing requests. Mobile app calls server APIs via HTTPS; server uses Java/Spring Boot (or Node.js/Express) to implement endpoints.
- Data Layer: MySQL database on the server stores user accounts, complaint records, post office details, and admin logs. An ORM (e.g., Hibernate or Sequelize) manages object-relational mapping.

APIs are versioned to allow backward-compatible updates. The client caches non-sensitive data locally and refreshes it periodically or on demand.

4.7 System Modules

To keep development organized, the project is divided into these modules, each with focused responsibilities:

- User Module: Manages user signup/login, profile settings, map view, and AI chat interface.
- Location Module: Handles GPS permissions, location scanning, proximity calculations, and map marker updates.
- AI Chat Module: Wraps the Gemini SDK, maintains chat history, and triggers fallback mechanisms if AI fails.
- Complaint Module: Provides UI forms for text, image, or voice feedback; assigns tracking IDs; and syncs with the server.
- Admin Module: A separate web or mobile interface for administrators to log in, view and manage complaints, update post office data, and publish alerts.
- Search Offices: Allow users to search by post office name and display the corresponding Taluka and District (limited to Andhra Pradesh, Telangana, and Karnataka).
- Navigation & Call Module: Leverages Android Intents to launch phone calls or navigation apps with prefilled data.

Each module is developed in isolation with its own test suite, then integrated into the main application.

4.8 System Workflow

The user journey and backend handling follow a clear, repeatable sequence:

- App Launch & Authentication: User opens the app, sees a welcome screen, and either registers or logs in. Credentials are verified via an API call.
- Home & Map Display: Upon successful login, the map screen appears, centering on the user's current location and plotting nearby post offices.
- Interactive Actions: Tapping a map marker shows post office details and options: call, get directions, or view services.
- AI Chat: If the user taps the chat icon, they enter a conversation with Gemini AI. The chatbot processes input, consults internal FAQs or external APIs, and returns answers instantly.
- Complaint Submission: From any screen, the user can file a complaint. The form accepts text, optional images, or voice notes, then sends it to the server.
- Search Offices: From the home screen, the user enters a post office name. The app sends a search request to the backend. The backend returns matching offices along with Taluka and District. The app then displays the search results to the user.
- Admin Response: Admin logs into their panel, sees new complaints, updates status, and can send replies. Users receive push notifications when their complaint status changes.

This workflow ensures each feature is accessible with minimal steps, creating a seamless experience from query to resolution.

4.9 Advantages of the Proposed Method

- Enhanced User Experience: Intuitive UI, fast AI responses, and clear navigation reduce user effort and streamline postal interactions.
- Centralized Management: A unified platform for users and admins removes siloed communication, speeding up resolutions.
- Real-Time Capabilities: Immediate map updates and AI replies keep users informed and engaged.
- Accessibility: Offline support and multilingual potential ensure broad usability in urban and rural areas.

4.10 Future Enhancements

- Multi-Language Support: Add resource files for Hindi, Spanish, and other regional languages.
- Delivery Tracking: Integrate parcel scanning and real-time GPS tracking for mailed items.
- Push Notifications: Alert users about complaint status changes or new office announcements instantly.
- Third-Party Integrations: Enable partnerships with courier services, bill payment platforms, or emergency services for an expanded service ecosystem.

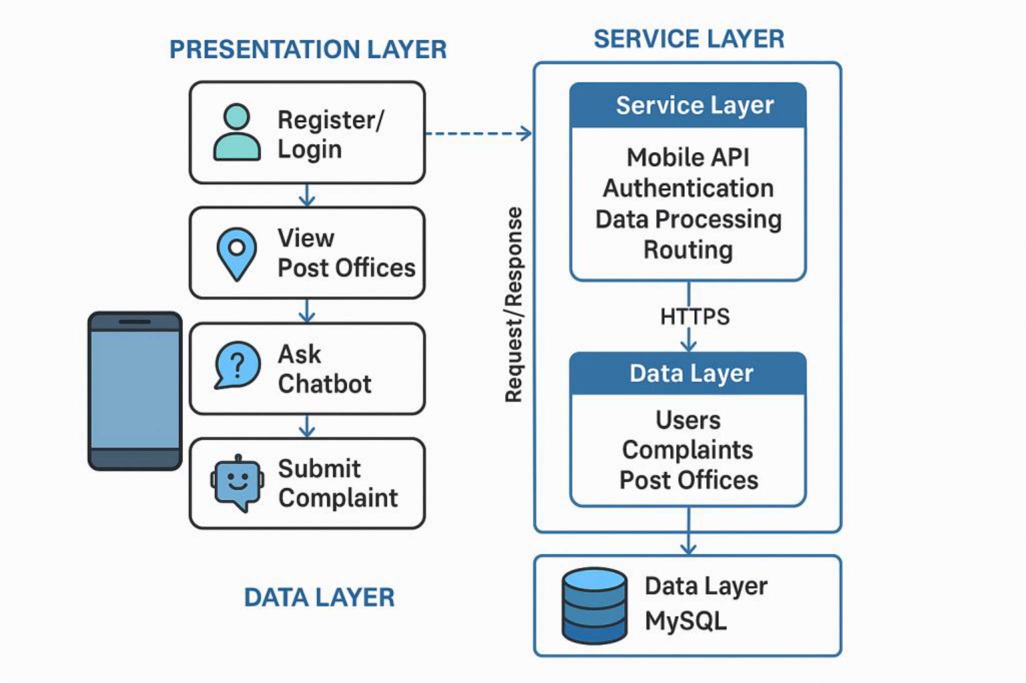


Figure 4.1 System Architecture

CHAPTER 5
OBJECTIVES

This chapter outlines the goals of the AI-Powered Delivery Post Office Identification System. These objectives guide the design, development, and deployment of the application to ensure it serves both users and administrators effectively, especially in the context of India's vast and diverse postal service needs.

5.1 General Objective

To design and develop a mobile application that improves access to postal services using Artificial Intelligence (AI), geo-mapping, and real-time interaction making postal support faster, smarter, and more user-friendly for the general public and administrators. The app will provide seamless digital services, bridging the gap between citizens and physical post offices through technology. It aims to transform the traditional service delivery model into a modern, responsive, and interactive platform.

5.2 Specific Objectives

5.2.1 User-Friendly Android App

The system aims to provide a smooth and easy-to-use Android application that allows users to register, log in, and securely access postal features through their mobile phones. The app will follow modern design principles, with intuitive navigation and responsive layouts suitable for users of all age groups. Whether a person is tech-savvy or new to smartphones, they will be able to explore the features comfortably. Key elements like buttons, forms, and icons will be designed for readability and ease of access. In addition, the interface will follow accessibility standards such as high contrast and large touch targets to support users with visual or motor impairments. Error messages and input validation will be designed to be clear and helpful, ensuring a seamless and inclusive experience.

5.2.2 To Geo-Mapping Integration

The app will use the Google Maps API to help users locate nearby post offices based on their current GPS coordinates. The map will show office locations with interactive markers and allow users to filter by distance or services offered (e.g., parcel drop, savings account, complaint counter). The feature will also include route suggestions so users can find the most efficient path to the selected branch. This will save time and effort, especially in new or unfamiliar areas. The use of geo-tagging ensures real-time accuracy, and users can also view office details such as opening hours, contact information, and available services directly from the map view.

5.2.3 AI Chatbot (Gemini AI)

To improve accessibility and automate query resolution, the app will include a chatbot powered by Gemini AI. It will simulate human conversation and respond to questions like "When does the nearest post office open?" or "How do I file a complaint?" The chatbot will support multilingual interaction and help users complete tasks without needing to call or visit the post office. This reduces the burden on postal staff and enhances user satisfaction. The AI will use natural language processing to understand diverse queries and provide relevant responses instantly. Over time, the chatbot will learn from interactions and become more accurate and helpful.

5.2.4 Complaint Registration System

The platform will simplify the process of lodging complaints. Users can submit complaints via text, voice messages, or images and will receive a unique tracking ID. They can monitor complaint progress and receive notifications when updates occur. This digital approach provides better traceability, faster handling, and transparency between users and the postal department. The system ensures that every complaint is logged in a centralized database, with time stamps and user details securely stored. This helps reduce miscommunication, ensures accountability, and allows users to remain informed throughout the resolution process.

5.2.5 Admin Control Panel

To support operational needs, an administrative interface will be provided for post office staff. Admins can securely log in, access submitted complaints, update their status, and manage user queries. The panel will also allow them to update post office contact information, working hours, and issue official notices. This centralized approach ensures faster response times and improved data accuracy. Additionally, the dashboard will offer analytics tools for tracking complaint trends, user satisfaction ratings, and service delivery metrics—helping admins make data-driven decisions and optimize operations.

5.2.6 Direct Call and Navigation Features

For convenience, users will be able to call a listed post office or get directions with just one tap. This will launch their phone dialer or navigation app (like Google Maps) with pre-filled contact or location details. This reduces the manual steps needed to reach out and improves the user's ability to plan visits more efficiently. It helps eliminate the need to manually search for contact numbers or routes, especially during emergencies or when immediate assistance is required.

5.2.7 Feedback and Ratings

After each chatbot conversation or complaint resolution, users will be prompted to rate the service and provide feedback. This helps administrators measure satisfaction and identify areas for improvement. Feedback will be stored in a backend database and analyzed periodically to enhance services and staff performance. The rating system will be designed to be quick and easy, encouraging more users to provide input. Reports generated from feedback will be shared with relevant departments for corrective actions or recognition.

5.2.8 Data Security and Privacy

To ensure user trust and legal compliance, the system will use secure login, encrypted data transmission (HTTPS), and role-based permissions. Sensitive information like user credentials and complaint details will be stored securely. These practices will prevent unauthorized access and protect data from external threats. The application will also follow national data protection guidelines and industry best practices to ensure confidentiality, integrity, and availability of user data.

5.2.9 Search Post Offices

The application will include a powerful search functionality that enables users to look up post offices by name directly from the home screen. Upon entering a post office name, the app will communicate with the backend to retrieve a list of matching results, each accompanied by key details such as Taluka and District. This feature ensures fast and accurate identification of postal branches across India, making it easier for users to access local services. The search interface will be designed for simplicity, offering real-time suggestions and relevant filters. By streamlining the discovery process, this feature minimizes user effort and enhances the overall experience, especially for users unfamiliar with geographic hierarchies.

CHAPTER 6

SYSTEM DESIGN & IMPLEMENTATION

This chapter explains how the AI-Powered Delivery Post Office Identification System was designed and implemented. It covers the system architecture, user interface, module structure, data flow, and development process. The goal was to build a user-friendly, fast, and scalable Android application that helps users interact with postal services more easily.

6.1 System Architecture

The architecture of the system is based on a client-server model. The client is the Android mobile app that users install on their phones. The server handles data storage, AI processing, user requests, and admin operations.

- Client Side: Built using Kotlin in Android Studio. It includes screens for login, registration, map view, chatbot, complaints, and feedback.
- Server Side: Uses a MySQL database for storing user data, complaints, and post office details. It also handles API connections for the chatbot and Google Maps.
- AI & Maps: The chatbot is powered by Gemini AI, and location services are powered by the Google Maps API.

6.2 User Interface Design

The user interface is simple and clean to make it easy for users of all ages to interact with the app. The design includes:

- A welcome page with login and signup options.
- A dashboard showing a map with nearby post offices.
- A chatbot icon to ask questions.
- A complaint form for submitting issues.
- A search bar on the home screen to search post offices by name.
- A feedback section to rate the app.

6.3 Module Division

The system is divided into several modules to keep the code organized and easy to maintain:

- User Module: Handles user registration, login, and account information.
- Map Module: Displays nearby post offices using GPS and Google Maps API.
- Chatbot Module: Allows users to chat with Gemini AI to get help.
- Complaint Module: Lets users submit complaints with details and track responses.
- Search Module: Finds post offices by name.

- Feedback Module: Allows users to rate the app and suggest improvements.
- Admin Module: Lets administrators update post office data, reply to complaints, and post updates.

6.4 Data Flow Diagram (DFD Overview)

- The user logs in and sends a request.
- The app sends the request to the server.
- The server checks the database and sends the result back.
- If a map is requested, Google Maps API is called.
- If a question is asked, the Gemini AI chatbot responds.
- User searches, server returns matching offices.
- Admins access data via a secure login and manage updates.

6.5 Implementation Tools and Technologies

- Language: Kotlin (for Android app)
- IDE: Android Studio
- Database: MySQL
- APIs: Google Maps API, Gemini AI API
- Backend Logic: Implemented using Java/Kotlin and REST APIs
- Hosting: Local server or cloud server for database and chatbot API

6.6 Implementation Process

The project followed modular implementation. Each feature was built, tested, and improved in steps:

- Setup project environment (Android Studio, SDK, Database)
- Created UI screens and linked navigation.
- Integrated Google Maps to show nearby post offices.
- Connected Gemini AI chatbot for Q&A support.
- Added search feature for post offices.
- Added complaint and feedback forms.
- Built admin dashboard with secure login.
- Tested each module separately, then as a full app.

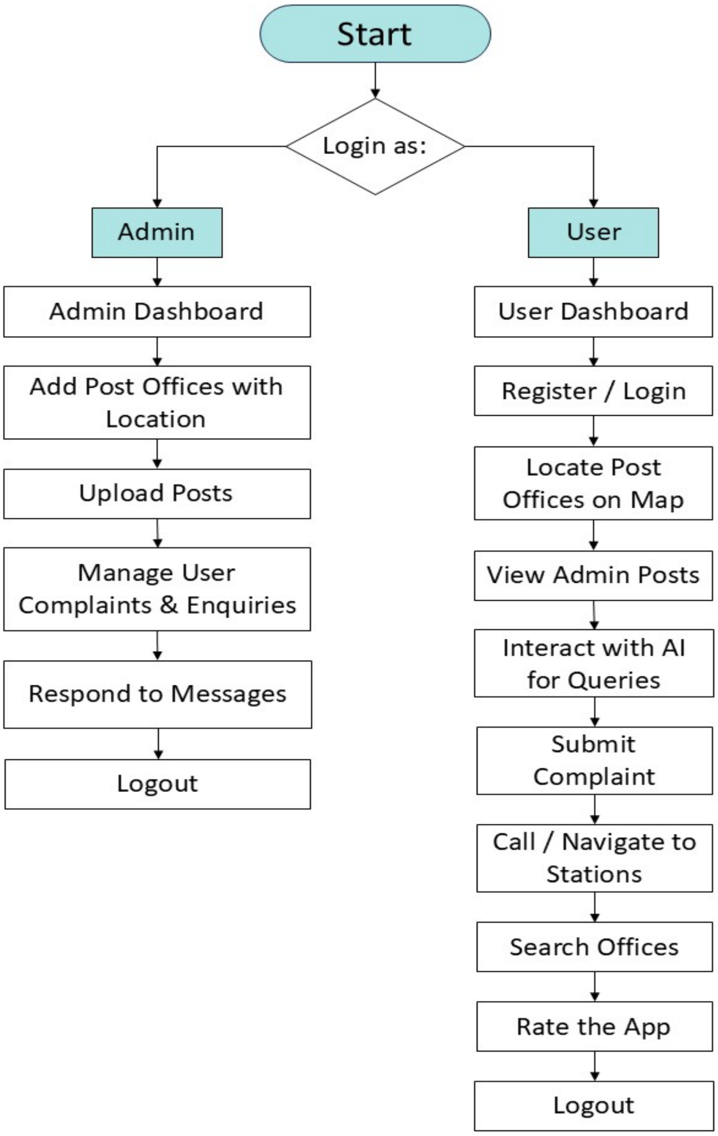


Figure 6.1 Working Flow

CHAPTER 7
TIMELINE FOR EXECUTION OF PROJECT
(GANTT CHART)

7.1 Understanding Gantt Charts

Gantt charts are a helpful tool used to plan and manage projects. They show each task or activity as a horizontal bar on a timeline, making it easy to see when a task starts, how long it will take, and when it should be completed. This visual method helps both the team and the project manager keep track of what has been done and what still needs to be completed. Gantt charts are especially useful in technical projects like app development because they help organize complex tasks in a simple way. They also allow project teams to spot delays early and manage time better. By using a Gantt chart, we were able to divide our project into different phases and clearly see how the project would progress week by week.

7.1.1 Visual Representation of Projects

The main advantage of a Gantt chart is that it gives a clear and easy-to-understand visual of the entire project. All the tasks are listed vertically, and the timeline is shown horizontally. Each task has a colored bar that shows its duration and timing. This makes it simple for everyone involved in the project to see what needs to be done, who is responsible, and when the task is scheduled. It also helps identify which tasks can be done at the same time and which ones must be done in a specific order. For our project, the Gantt chart helped everyone on the team stay updated and work in the right order without confusion.

7.1.2 Resource Allocation and Workload Management

Another big benefit of using Gantt charts is that they help in dividing work properly among team members. It allows project managers to see who is working on which task and whether their workload is balanced. This helps avoid overloading one person while others have less work. It also ensures that everyone is contributing equally. For our project, this was especially useful because it allowed us to assign tasks like frontend design, AI chatbot integration, and testing to different team members according to their strengths. Managing time and effort properly made our team more productive and reduced unnecessary delays.

7.1.3 Communication and Collaboration

Gantt charts also help teams communicate better. Since the whole timeline is visible, it is easier for everyone to know the current status of the project and what tasks are coming up next. If one part of the project is delayed, the chart shows how it may affect the rest of the work, and the team can take quick action. It also helps during meetings, as updates can be discussed by just looking at the chart. In our project, it kept everyone on the same page, improved teamwork, and reduced confusion. Everyone knew what to do and when to do it, which made collaboration smooth and efficient.

7.2 Benefits of Gantt Charts for Postal Teams

Even though our project is not directly related to Postal, the advantages of Gantt charts apply to all kinds of technical teams. Postal teams, like development teams, handle many complex tasks that require clear planning and coordination. Gantt charts help in organizing these tasks clearly, setting deadlines, and tracking work in progress. This makes teams more focused and helps them deliver results on time. For any technical or software project, a Gantt chart plays a key role in successful project management.

7.2.1 Enhanced Project Visibility and Planning

Gantt charts offer full visibility of the project plan to everyone involved. This means team members, faculty, and other stakeholders can easily understand what's happening at any point. It helps set clear expectations and build trust within the team. In our project, we used the Gantt chart to see how far we had come and how much work was left. It helped us plan better, make quicker decisions, and stay confident about completing the project on time.

7.2.2 Efficient Resource Utilization

One of the biggest benefits of a Gantt chart is that it helps us make full use of the resources available. Whether it's time, tools, or team members, we were able to manage everything more efficiently because of the Gantt chart. Tasks were not overlapping, and no one was sitting idle. Everyone had a clear role and deadline, which led to better teamwork and time management.

7.2.3 Proactive Risk Management

By showing task deadlines and dependencies, Gantt charts help in identifying risks early. For example, if one task takes longer than expected, we could quickly see how it might delay the next task and take action to fix it. This helped us avoid big problems later. During our project, this was very helpful in areas like AI integration and testing, where we expected delays. The Gantt chart allowed us to plan backup time in advance.

7.3 Project Timeline and Key Milestones

Our project followed a timeline of 4 months (16 weeks), where we divided the work into clear phases. Each phase had a goal and a deadline. The Gantt chart helped us break down the project into smaller tasks like research, design, development, testing, and documentation. It gave us a clear picture of what we needed to do each week. By following this plan, we were able to finish the project on time without missing any important steps. Below are the key phases of our project explained in detail.

7.3.1 Identification of Modules

After completing the research, we divided the project into smaller parts called modules. Each module was created based on the features we wanted to include in our app. These modules included user registration and login, AI chatbot assistance, nearby post office locator with map integration, complaint submission system, and an admin dashboard for managing user data and updates. Breaking the project into modules helped us plan better, assign tasks more clearly, and manage our time well. It also allowed us to work on different parts of the app separately and then combine them during integration.

7.3.2 Implementation of Modules

In this phase, we started developing the actual mobile application using the Kotlin programming language and Android Studio. Each module was implemented step by step. We first created the user interface, then connected it to the database, and added AI support using Gemini. We developed features like the map for locating post offices, AI-based user query handling, a feedback system, and admin-level controls. We also made sure the app was responsive, easy to use, and visually appealing. By building each module carefully, we ensured that all the functions were properly developed and ready for testing.

7.3.3 Testing of Modules

Once the development phase was over, we focused on testing each module to ensure everything was working correctly. We performed unit testing for individual modules and system testing to check the entire app flow. This phase helped us find bugs, fix errors, and improve performance. We tested the app on different devices to make sure it worked smoothly for all users. User login, location detection, AI chat, and complaint handling were all carefully tested. This step was important to make sure the app was reliable, secure, and provided a good user experience before moving to the final stage.

7.3.4 Publishing and Preparation of Research Papers

After confirming that the app was working smoothly, we moved to the publishing and documentation phase. We started writing a research paper and project report that explained the idea, the technologies used, and the results achieved. We added screenshots of the app, explained its features, and shared the challenges we faced during development. This phase helped us document our hard work and gave us a proper way to present the project for academic purposes. Preparing this research paper was important for sharing our work with others and possibly using it for future improvements or publishing.

7.3.5 Maintenance

The final phase of the project is maintenance, which ensures the app continues to work well even after its launch. We planned to keep checking the app regularly for any bugs, errors, or performance issues. If users face any problems or request new features, we will update the app accordingly. Maintenance also includes improving speed, adding new services, and ensuring the app stays compatible with future Android updates. This ongoing phase is important to keep the system relevant, secure, and user-friendly over time.

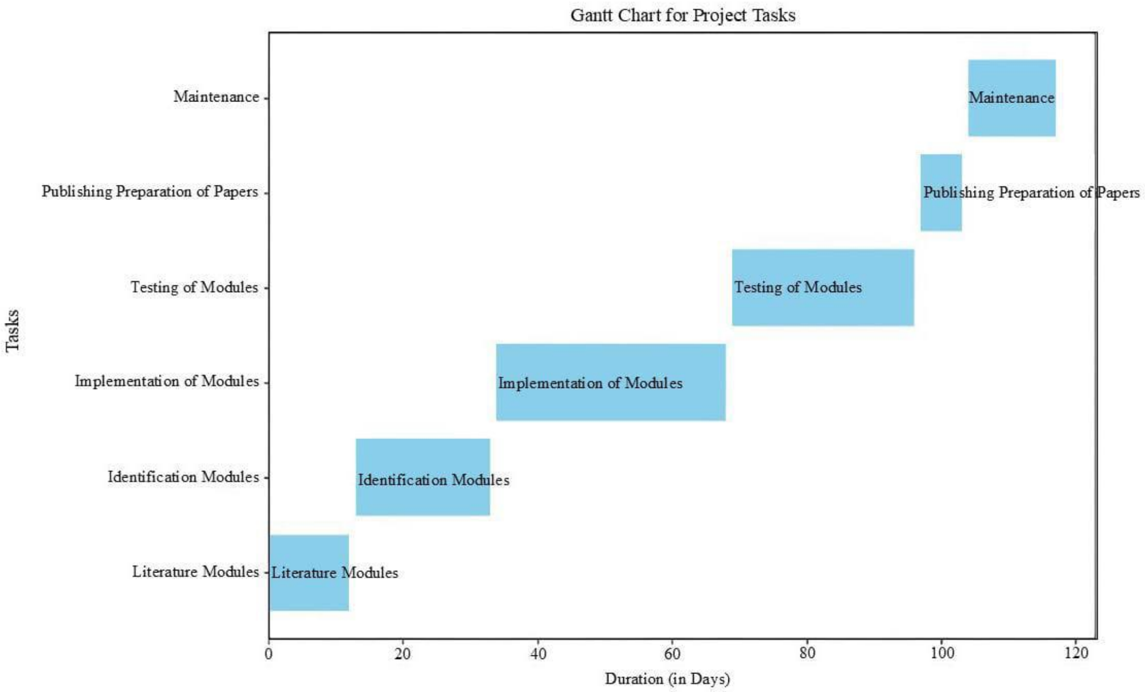


Figure 7.1 Gantt Chat for Project Tasks

CHAPTER 8
OUTCOMES

This chapter describes the results and achievements of developing the AI-Powered Delivery Post Office Identification System. After completing all the phases of the project research, design, development, and testing. we successfully created a fully functional Android application that solves real problems users face with postal services.

8.1 Successful Implementation of Key Features

The application includes all the features we planned:

- Users can register and log in securely.
- The app uses Google Maps to show nearby post offices in real-time.
- Users can interact with Gemini AI chatbot to ask postal queries.
- Complaints can be submitted with tracking and admin updates.
- User can Search the Post Offices Names.
- The admin dashboard allows staff to respond and update office data.
- Feedback and ratings can be submitted easily through the app.

These features were developed step-by-step and tested on multiple Android devices to ensure reliability, ease of use, and performance.

8.2 Improvement in User Experience

One of the main outcomes of the project is the improved interaction between users and postal services. Instead of physically visiting a post office to ask about working hours or file a complaint, users can now get help directly from their phones. The Gemini AI chatbot reduces user waiting time and guides them instantly.

Users also no longer have to manually search for contact numbers or directions. With just a few taps, they can call or navigate to a nearby post office and they can Search the Post Offices Names. The app offers a mobile-first and map-based experience, making it modern and accessible.

8.3 Admin Efficiency and Management

Administrators now have a dedicated space where they can:

- View and manage user complaints
- Update post office contact details
- Post real-time announcements
- Access feedback and make improvements

This centralized admin system helps reduce paperwork and makes service updates faster and more organized. It also improves transparency between the postal department and the public.

8.4 Academic and Technical Growth

Through this project, we learned:

- How to build mobile apps using Kotlin and Android Studio
- How to integrate external APIs like Google Maps and Gemini AI
- How to structure a backend with MySQL and RESTful services
- The importance of user interface design and modular coding

This project helped us improve our technical skills, team collaboration, and problem-solving abilities, making it a valuable learning experience.

8.5 Real-World Impact and Scalability

Although this is a student project, the system is built with future potential in mind. It can be easily scaled to:

- Add parcel tracking features
- Connect with other government delivery services
- Support more regional languages for rural access

This shows that the project is not only useful for academic purposes but can also be adapted for real-world implementation to modernize India's postal services.

CHAPTER 9

RESULTS AND DISCUSSIONS

This chapter explains the results observed after developing and testing the AI-Powered Delivery Post Office Identification System. It also discusses how these results align with the original goals, the improvements seen in performance and user experience, and the challenges faced during development. The outcomes are based on practical testing, user feedback, and system performance.

9.1 Functional Results

After completing the development, all planned features were implemented and tested successfully. Users were able to:

- Register and log in to the app securely.
- View nearby post offices using Google Maps with real-time location tracking.
- Interact with the Gemini AI chatbot to ask postal-related questions.
- Submit complaints and receive updates through a tracking system.
- Call post offices or get directions using in-app navigation.
- Search Post Offices Names.
- Give ratings and feedback after using the app.

These features worked smoothly on multiple Android devices, including low-end smartphones. The app performed well even on slower networks, confirming that the system is functional, lightweight, and accessible to a wide user base.

9.2 Performance and Stability

The app was tested on various Android devices with different screen sizes and versions. It showed:

- Fast loading of maps
- Quick AI chatbot replies
- Smooth screen transitions
- Search Post Offices Names
- Stable performance without crashes

The backend built using MySQL and REST APIs processed user data, complaints, and admin operations efficiently. API calls returned accurate results, and the system managed data securely without delays or failures.

9.3 User Feedback and Interface Experience

We tested the app with different users and collected feedback. Most users said the interface was easy to use, well organized, and visually clear. They liked features such as:

- One-tap navigation to post offices
- Simple complaint form with file upload
- Fast responses from the AI chatbot

Some users recommended voice-based search and support for more languages. These can be considered for future updates. Overall, users gave the app an average rating of 4.5 out of 5 **stars**, showing high satisfaction. Many users preferred the app over visiting a post office physically, proving a shift toward digital solutions.

9.4 Admin Functionality

On the admin side, the dashboard provided smooth operations for:

- Viewing and replying to user complaints
- Updating post office details like hours, contact info, and location
- Sending announcements or updates

Admins could log in securely and manage all tasks from a single platform. There were no errors or data issues during testing. Admins also reported that the system helped save time and reduce paperwork.

9.5 Discussion of Challenges

While the system was successful, we faced a few challenges during development:

- Integrating live Google Maps with accurate GPS on all devices
- Optimizing the app to work on low-end phones
- Adjusting the AI chatbot to better understand local postal queries
- Ensuring UI compatibility across different screen sizes

These issues were solved step-by-step through coding improvements, testing on real devices, and tuning the chatbot for accuracy.

9.6 Overall Discussion

The project met all its major goals. It delivered a smart, modern, and user-friendly postal support system. Users could easily locate post offices, file complaints, and interact with staff digitally. The chatbot and map-based features brought a fresh, efficient way for users to engage with services.

The system benefits both public users and postal administrators. It reduces physical visits, simplifies communication, and helps in better time management. The results prove that a mobile-first, AI-powered app can truly enhance public service delivery.

9.7 Comparative Analysis with Traditional System

Compared to traditional postal processes, the app provides:

- Faster and more convenient access to information
- Instant help through AI instead of long phone calls or visits
- Easier complaint handling and tracking
- Centralized and efficient admin response management

Earlier, users had to physically go to the post office, wait in lines, or rely on outdated websites. Now, everything can be done from their phones—making the experience easier, faster, and available 24/7. This comparison highlights how digital innovation can solve real-world issues and modernize public services effectively.

CHAPTER 10
CONCLUSION

The AI-Powered Delivery Post Office Identification System was developed to make postal services more accessible, faster, and user-friendly. The goal was to reduce the gap between users and post offices by using modern technology such as Artificial Intelligence (AI), geo-mapping, and mobile-first design. After completing all stages of the project—from planning to testing—the system met its objectives successfully.

The Android app allows users to easily find nearby post offices using GPS and Google Maps. It helps users interact with a chatbot powered by Gemini AI to get instant replies to common postal questions. The app also includes options to file complaints, track them, and give feedback. User can Search the Post Offices Names. Admin users can manage these complaints, update post office details, and send real-time announcements from their dashboard.

The system works smoothly on various Android devices, supports real-time responses, and reduces the need to physically visit post offices. This makes postal services more efficient and eco-friendly. During testing, both users and admins gave positive feedback, especially appreciating the simplicity, speed, and usefulness of the app.

The project also helped the team learn about mobile app development, working with APIs, using databases, and handling real-world challenges in software development. It showed how digital tools can improve public services, especially in areas like complaint handling, user support, and service visibility.

REFERENCES

[1] B. Kitchenham, *"*A Systematic Review of Software Requirements Specification Studies*,"* ACM Computing Surveys, 2004.

[2] R. Koesdwiady, R. Soua, and F. Karray, *"*Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach*,"* IEEE Transactions on Vehicular Technology, vol. 65, no. 12, pp. 9508–9517, Dec. 2016.

[3] O. Kirimtat, O. Krejcar, A. Kertesz, and M. F. Tasgetiren, *"*Future Trends and Current State of Smart City Concepts: A Survey,*"* MDPI Applied Sciences, vol. 10, no. 18, pp. 1–21, 2020.

[4] Agile Alliance, "Agile Software Development Lifecycle – Principles and Benefits," 2021. Highlights the benefits of agile practices in software development for improved flexibility and teamwork.

[5] A. C. de Araujo and A. Etemad, *"*End-to-End Prediction of Parcel Delivery Time With Deep Learning for Smart-City Applications,*"* arXiv preprint arXiv:2009.12197, 2021.

[6] M. Chen, W. Ulmer, and B. W. Thomas, *"Deep Q-learning for Same-Day Delivery with Vehicles and Drones,"* arXiv preprint arXiv:1910.11901, 2022.

[7] Google Maps Platform, "Using Maps SDK in Android Applications," 2022. Describes how to add map and location features in mobile apps for real-time geolocation.

[8] Android Developers, "Kotlin for Android Developers," 2022. A guide for building Android apps using Kotlin, offering modern syntax and better performance.

[9] OpenAI, "Using AI Chatbots for Real-Time User Interaction," 2023. Outlines the use of intelligent chatbots to enhance user support and interaction on digital platforms.

[10] Y. Chen, N. Men, N. Fuster, and C. Osorio, *"*Artificial Intelligence in Logistics Optimization with Sustainable Criteria: A Review*,"* IEEE Access, Oct. 2024.

APPENDIX-A
PSUEDOCODE

ChatActivity.kt

```
package com.project.postalapp.chatbot
import android.annotation.SuppressLint
import android.graphics.Bitmap
import android.location.Geocoder
import android.os.Bundle
import android.provider.MediaStore
import android.util.Log
import android.widget.Toast
import androidx.activity.result.PickVisualMediaRequest
import androidx.activity.result.contract.ActivityResultContracts
import androidx.activity.viewModels
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.cropiq.chatbot.GeminiAdapter
import com.example.cropiq.model.DataResponse
import com.google.ai.client.generativeai.GenerativeModel
import com.google.ai.client.generativeai.type.content
import com.google.android.gms.location.LocationServices
import com.project.postalapp.databinding.ActivityChatBinding
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers.IO
import kotlinx.coroutines.launch
import java.text.SimpleDateFormat
import java.util.Date
import java.util.Locale
class ChatActivity : AppCompatActivity() {
    private val binding by lazy { ActivityChatBinding.inflate(layoutInflater) }
    private var bitmap: Bitmap? = null
    private val responseData = arrayListOf<DataResponse>()
    private val viewModel: PincodeViewModel by viewModels()
    private lateinit var adapter: GeminiAdapter
    private var currentPostalData: List<PostOffice>? = null
    private val fused by lazy { LocationServices.getFusedLocationProviderClient(this) }
    var city = "New Delhi" // Default value
    var pincode1 = "" var ad = "" private val pickMedia =
        registerForActivityResult(ActivityResultContracts.PickVisualMedia()) { uri ->
            uri?.let { bitmap = MediaStore.Images.Media.getBitmap(contentResolver, it)
                binding.selectImage.setImageURI(it)
            } ?: Log.d("Photopicker", "No media selected")
        }@SuppressLint("NotifyDataSetChanged", "MissingPermission")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(binding.root)
        setupLocation()
        setupRecyclerView()
        setupClickListeners()
    }private fun setupLocation() {
        getUserLocation()
        viewModel.pincodeData.observe(this) { responses ->
            if (responses.isNullOrEmpty()) {
                Log.d("PincodeData", "No postal data received")
                return@observe
            } responses.firstOrNull()?.let { response ->
                currentPostalData = response.PostOffice?.sortedBy { it.Name }
                response.PostOffice?.forEach { postOffice ->
                    Log.d("PincodeData", "PostOffice: ${postOffice.Name} in
${postOffice.District}")
                    city = postOffice.District ?: city
                } ?: run { Log.d("PincodeData", "PostOffice list is null")
                }  } }  viewModel.error.observe(this) { errorMessage ->
            Toast.makeText(this, errorMessage, Toast.LENGTH_SHORT).show()
            addAiMessageToChat("I couldn't access postal data. Please check your connection or
try again later.")  } private fun setupRecyclerView() {
        adapter = GeminiAdapter(this, responseData)
        binding.chatRecyclerView.apply {
            layoutManager = LinearLayoutManager(this@ChatActivity)
            adapter = this@ChatActivity.adapter  }   }
    private fun setupClickListeners() {
        binding.selectImage.setOnClickListener {
pickMedia.launch(PickVisualMediaRequest(ActivityResultContracts.PickVisualMedia.Imag
eOnly))  }binding.sendButton.setOnClickListener {
            handleUserQuery() }   }
    @SuppressLint("MissingPermission")
    private fun getUserLocation() {
        fused.lastLocation.addOnSuccessListener { location ->
            try { val geocoder = Geocoder(this@ChatActivity, Locale.getDefault())
                val addresses = geocoder.getFromLocation(
                    location.latitude, location.longitude,
                     )  ad = addresses.toString()
                if (!addresses.isNullOrEmpty()) {
                    city = addresses[0].locality ?: "New Delhi"
                    val pincode = addresses[0].postalCode
                    pincode1 = pincode ?: ""
                    if (!pincode.isNullOrEmpty()) {
                        viewModel.fetchPincodeData(pincode)
                        addAiMessageToChat("📍 I've detected you're near $city (Pincode:
$pincode). How can I help with postal services?")
                    } else {
                        Log.d("Location", "No pincode found for this location")
                        addAiMessageToChat("I couldn't detect your pincode automatically. Please
tell me your location or pincode.")
                    }  }  } catch (e: Exception) {
                Log.e("Location", "Error getting location: ${e.message}")
                addAiMessageToChat("⚠️ I couldn't access your location. Please enable location
services or tell me your pincode.")
            } }.addOnFailureListener { exception ->
            Log.e("Location", "Error getting location: ${exception.message}")
            addAiMessageToChat("⚠️ Location error: ${exception.message}. Please tell me
your pincode manually.")
        }  } private fun handleUserQuery() {
        val userQuery = binding.chatInput.text.toString().trim()
        when {
            userQuery.isBlank() -> {
                showToast("Please enter your question")
                return
            }  userQuery.length < 3 -> {
                showToast("Please enter a more detailed question")
                return
            } else -> {
                binding.chatInput.setText("")
                addUserMessageToChat(userQuery)
                processUserQuery(userQuery)       }   } }
    private fun processUserQuery(query: String) {
        when {
            query.contains("list", ignoreCase = true) &&
                    query.contains("post office", ignoreCase = true) -> {
                listPostOfficesDirectly()
                return
            }   query.contains("pincode", ignoreCase = true) && pincode1.isNotEmpty() -> {
                addAiMessageToChat("Your detected pincode is: $pincode1")
                return }
            query.contains("location", ignoreCase = true) -> {
                addAiMessageToChat("You're currently detected in: $city (Pincode:
${pincode1.takeIf { it.isNotEmpty() } ?: "unknown"})")
                return     }   }
        val generativeModel = GenerativeModel(
            modelName = "gemini-1.5-flash",
            apiKey = "AIzaSyBfRYU6WewsoDqT6eiUwMiG7jQLBybdJMs"
        )  CoroutineScope(IO).launch {
            try {
                val response = generateAIResponse(generativeModel, query)
                runOnUiThread {
```

```
                addAiMessageToChat(response)
            }   } catch (e: Exception) {
            Log.e("ChatActivity", "Error generating content", e)
            runOnUiThread {
                showToast("Failed to generate response")
                addAiMessageToChat("I'm having trouble answering. Please try again or ask
differently.")    }    }    } }
    private fun listPostOfficesDirectly() {
        currentPostalData?.let { postOffices ->
            if (postOffices.isEmpty()) {
                addAiMessageToChat("No post offices found for your pincode.")
                return  }  val response = StringBuilder()
            response.append("📮 Found ${postOffices.size} post offices in $city:\n\n")
            postOffices.forEachIndexed { index, po ->
                response.append("${index + 1}. ${po.Name ?: "Unnamed Post Office"}\n")
                response.append("   📍 District: ${po.District ?: "Unknown"}\n")
                response.append("   🏛️ State: ${po.State ?: "Unknown"}\n")
                response.append("   #️⃣ Pincode: ${po.Pincode ?: "Unknown"}\n\n")
            }  response.append("How can I assist you further with these post offices?")
            addAiMessageToChat(response.toString())
        } ?: run {
            addAiMessageToChat("Post office data isn't available. Please check your pincode or
connection.")
        }  }  private suspend fun generateAIResponse(model: GenerativeModel, query: String):
String { val locationContext = buildLocationContext()
        val currentTime = SimpleDateFormat("hh:mm a, dd MMM yyyy",
Locale.getDefault()).format(Date())
        val prompt
        .trimIndent()
        return try {
            model.generateContent(content { text(prompt) }).text?.takeIf { it.isNotBlank() }
                ?: "I couldn't generate a response. Please try again."
        } catch (e: Exception) {
```

```
      Log.e("AI_Response", "Error: ${e.message}")
      "I'm having trouble answering. Please try again later."
    } }    private fun buildLocationContext(): String {
    return if (currentPostalData.isNullOrEmpty()) {
      "📍 Approximate Location: $city" +
          if (pincode1.isNotEmpty()) " (Pincode: $pincode1)" else ""
    } else {
      val firstOffice = currentPostalData!!.first()
      """
      📍 Confirmed Location Details:
      City/District: ${firstOffice.District ?: city}
      State: ${firstOffice.State ?: "Unknown"}
      Pincode: ${firstOffice.Pincode ?: pincode1.takeIf { it.isNotEmpty() } ?:
"Unknown"}
      """.trimIndent()
    }   }   private fun addUserMessageToChat(message: String) {
    responseData.add(DataResponse(0, message, ""))
    adapter.notifyItemInserted(responseData.size - 1)
    binding.chatRecyclerView.scrollToPosition(responseData.size - 1)   }
  private fun addAiMessageToChat(message: String) {
    responseData.add(DataResponse(1, message, ""))
    adapter.notifyItemInserted(responseData.size - 1)
    binding.chatRecyclerView.scrollToPosition(responseData.size - 1)    }
  private fun showToast(message: String) {
    Toast.makeText(this, message, Toast.LENGTH_SHORT).show()    }
  override fun onPause() {
    super.onPause()
    fused.flushLocations()
  }
}
```

APPENDIX-B

SCREENSHOTS

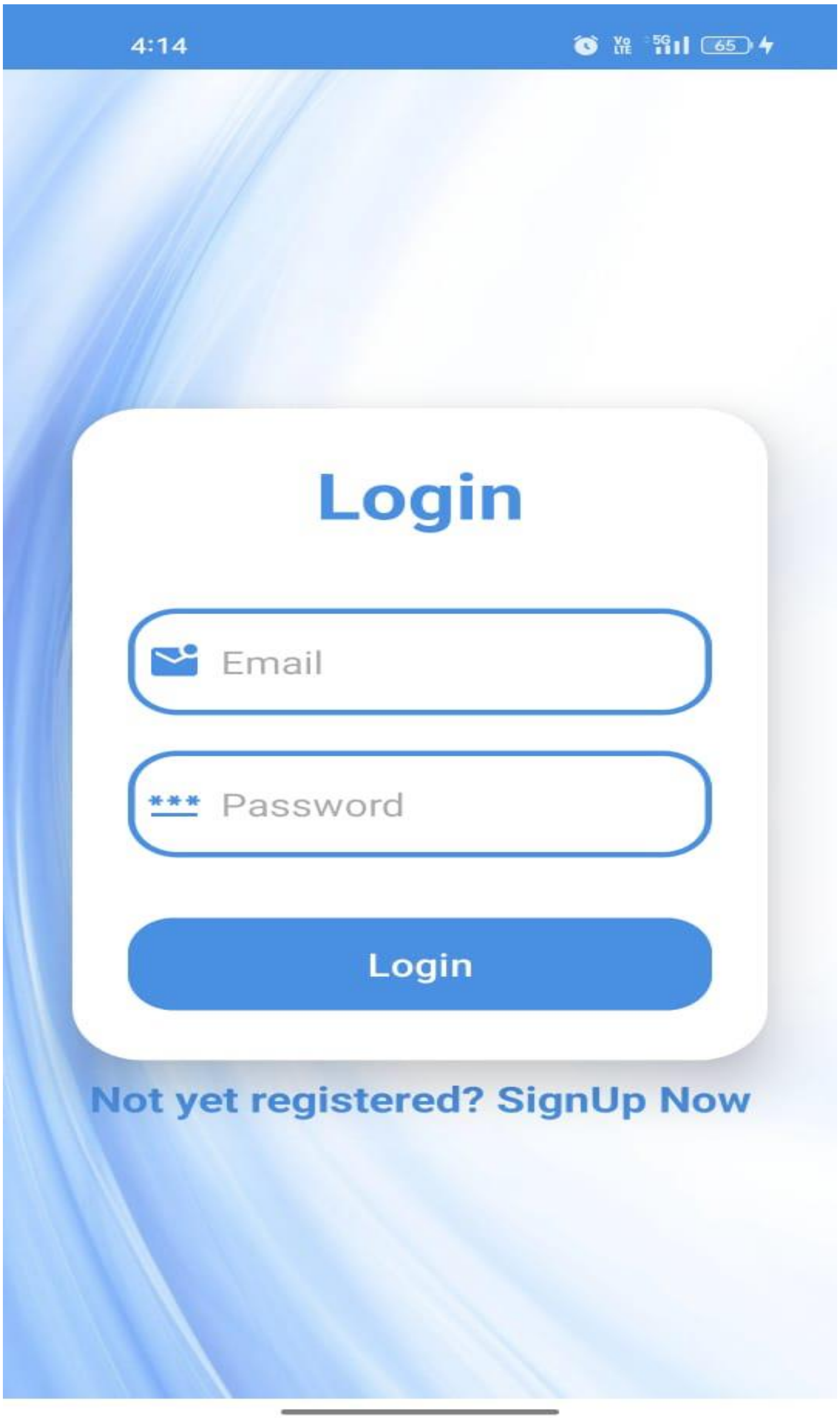


Figure Appendix B 1.1 Login page

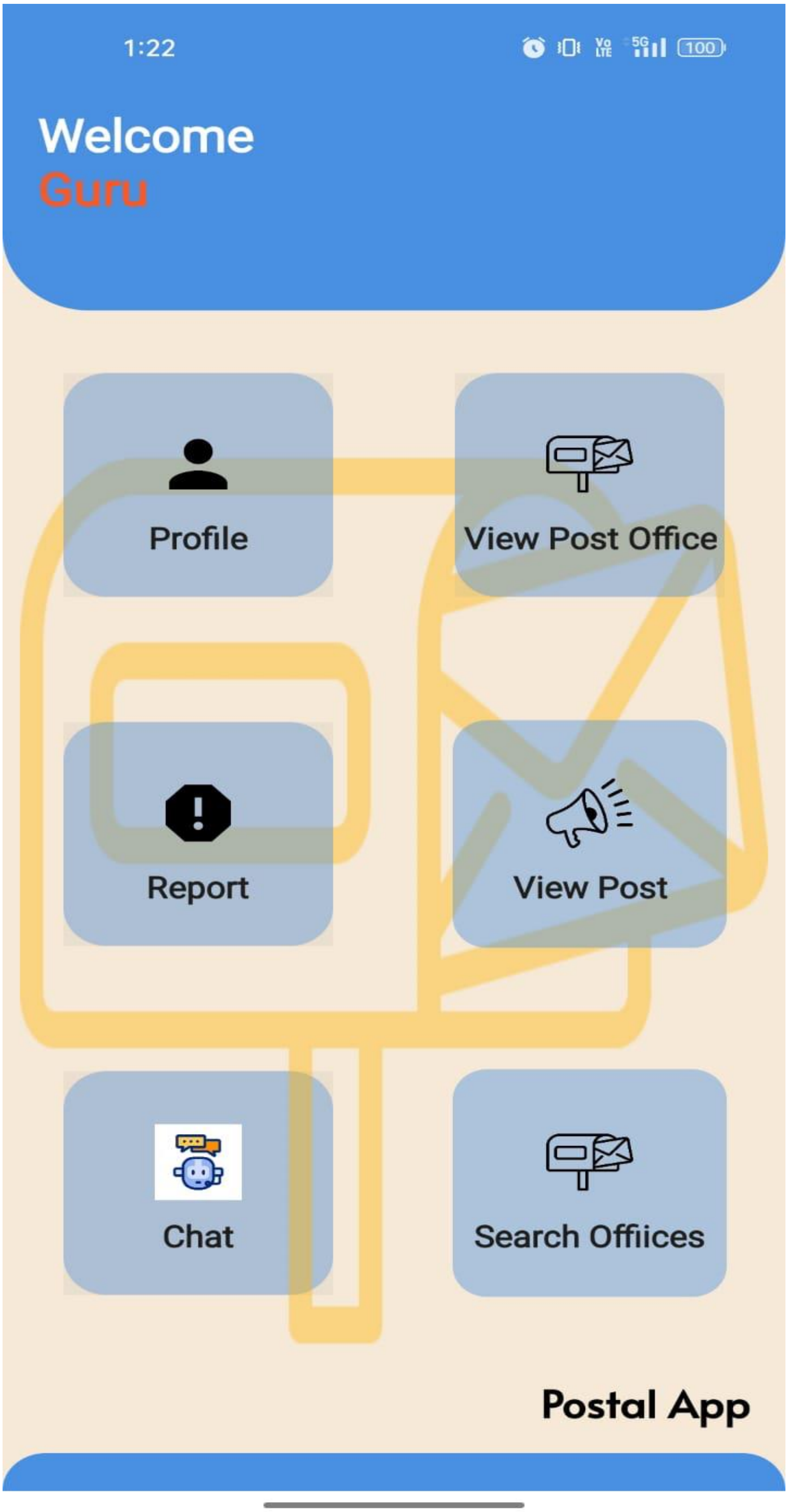


Figure Appendix B 1.2 User Portal

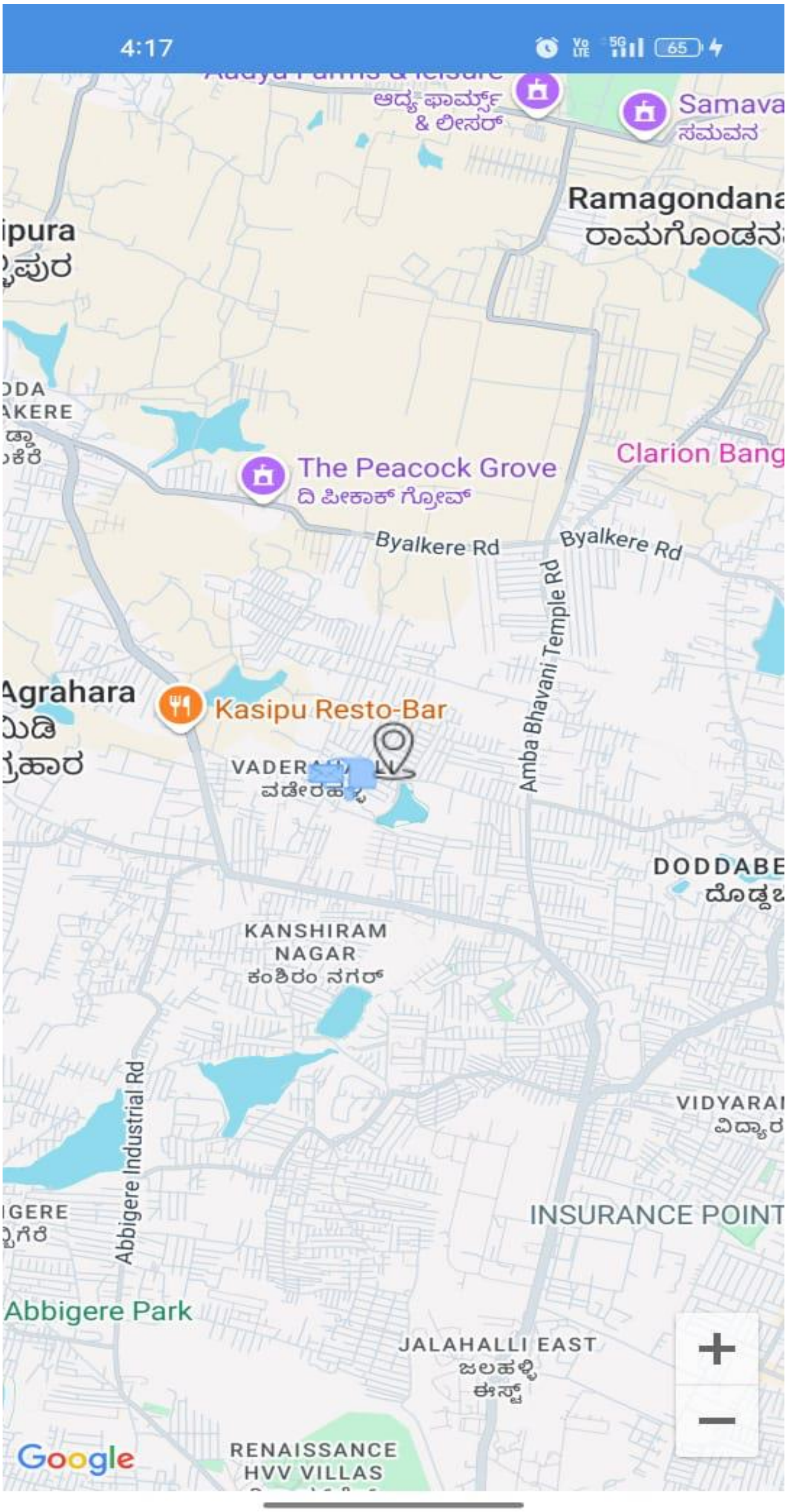
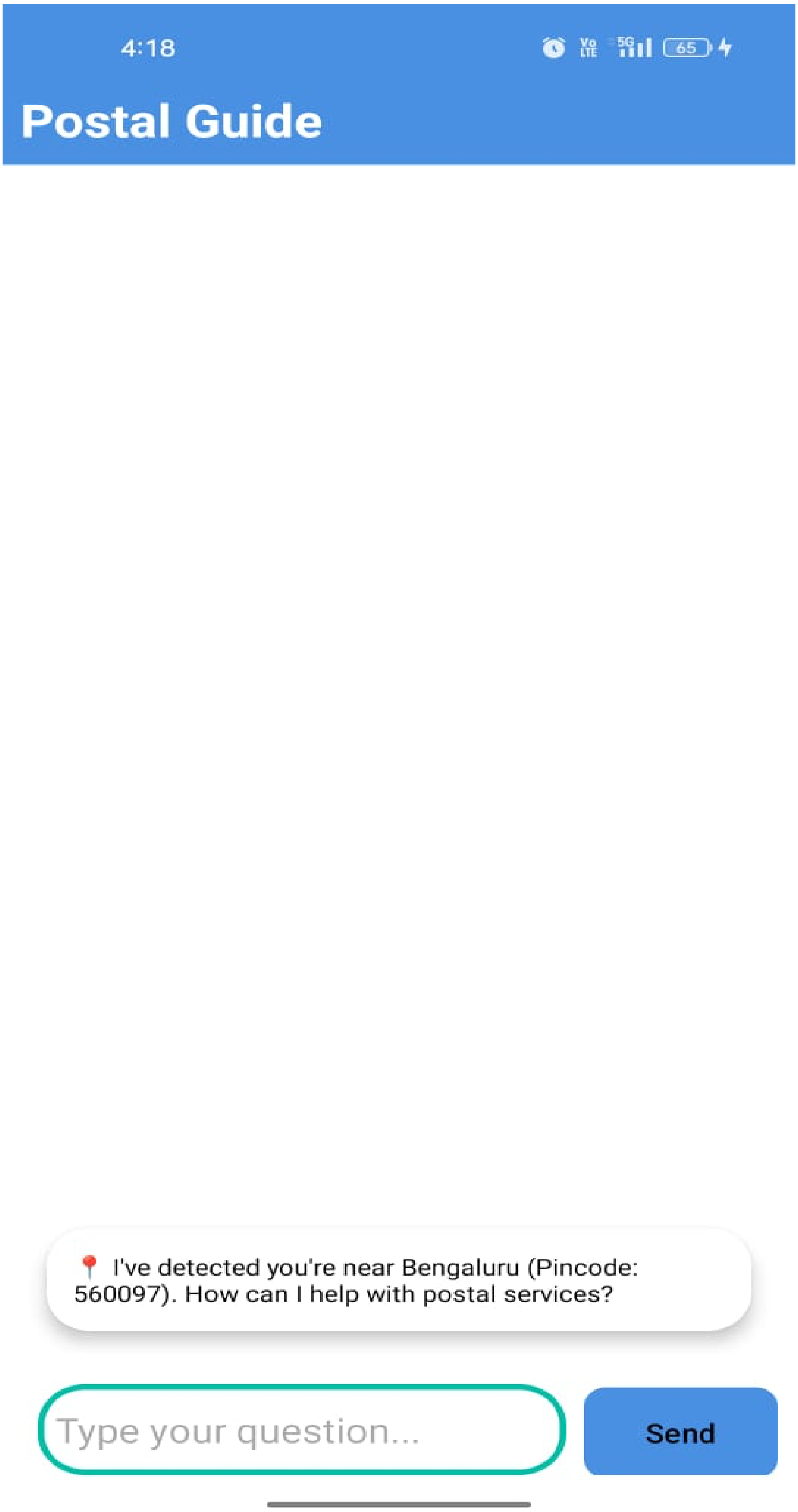
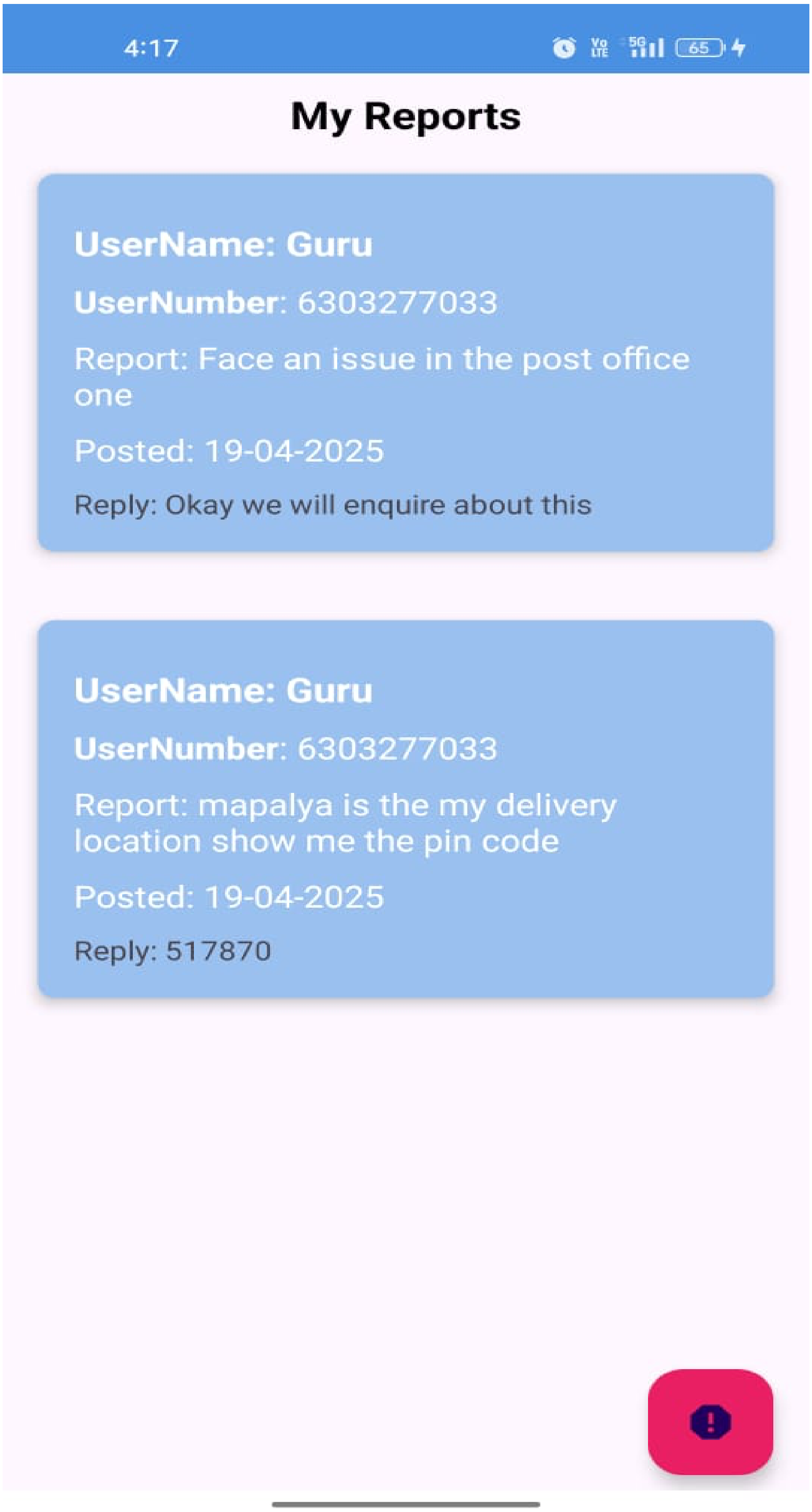


Figure Appendix B 1.3 User Nearest Post Office



📍 I've detected you're near Bengaluru (Pincode: 560097). How can I help with postal services?

Figure Appendix B 1.4 Postal Guide Chatbot



My Reports

UserName: Guru

UserNumber: 6303277033

Report: Face an issue in the post office one

Posted: 19-04-2025

Reply: Okay we will enquire about this

UserName: Guru

UserNumber: 6303277033

Report: mapalya is the my delivery location show me the pin code

Posted: 19-04-2025

Reply: 517870

Figure Appendix B 1.5 User Reports

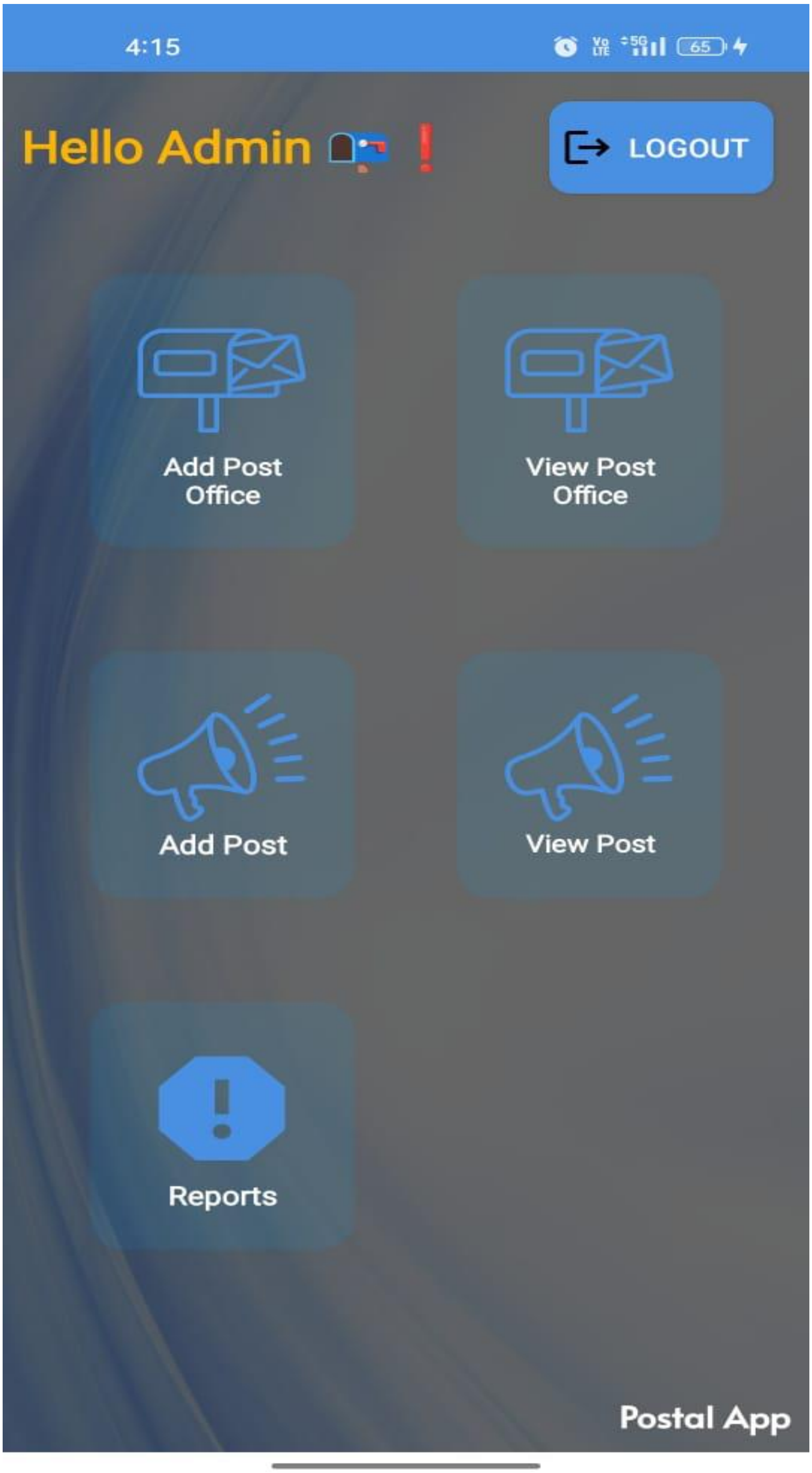


Figure Appendix B 1.6 Admin Portal

APPENDIX-C

ENCLOSURES

Sustainable Development Goals (SDGs)

1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE AND JUSTICE STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
THE GLOBAL GOALS
For Sustainable Development

Figure Appendix C 1.1 Sustainable Development Goals

SDG 9 – Industry, Innovation, and Infrastructure

This project uses modern technologies like AI, Kotlin, and geo-mapping to build a smart postal service platform. By creating a digital infrastructure that connects users and post offices, it fosters innovation in traditional public services. The app's integration of real-time maps and AI chat support shows how industry and infrastructure can evolve through software, making postal services faster and more reliable.

SDG 10 – Reduced Inequalities

The app offers equal access to postal information and services for everyone, including people in under-served or remote areas. Its simple interface and AI support make it usable by older adults, people with disabilities, or those unfamiliar with digital tools. In this way, it bridges the gap between different social groups and reduces inequality in access to essential services.

SDG 11 – Sustainable Cities and Communities

By showing nearby post offices on an interactive map and reducing the need for long travel, this app helps build smarter, more connected communities. Whether in busy cities or remote villages, users can quickly locate services, submit complaints, and receive updates.

This improved access supports sustainable urban and rural living by cutting down on wasted time and vehicle use.

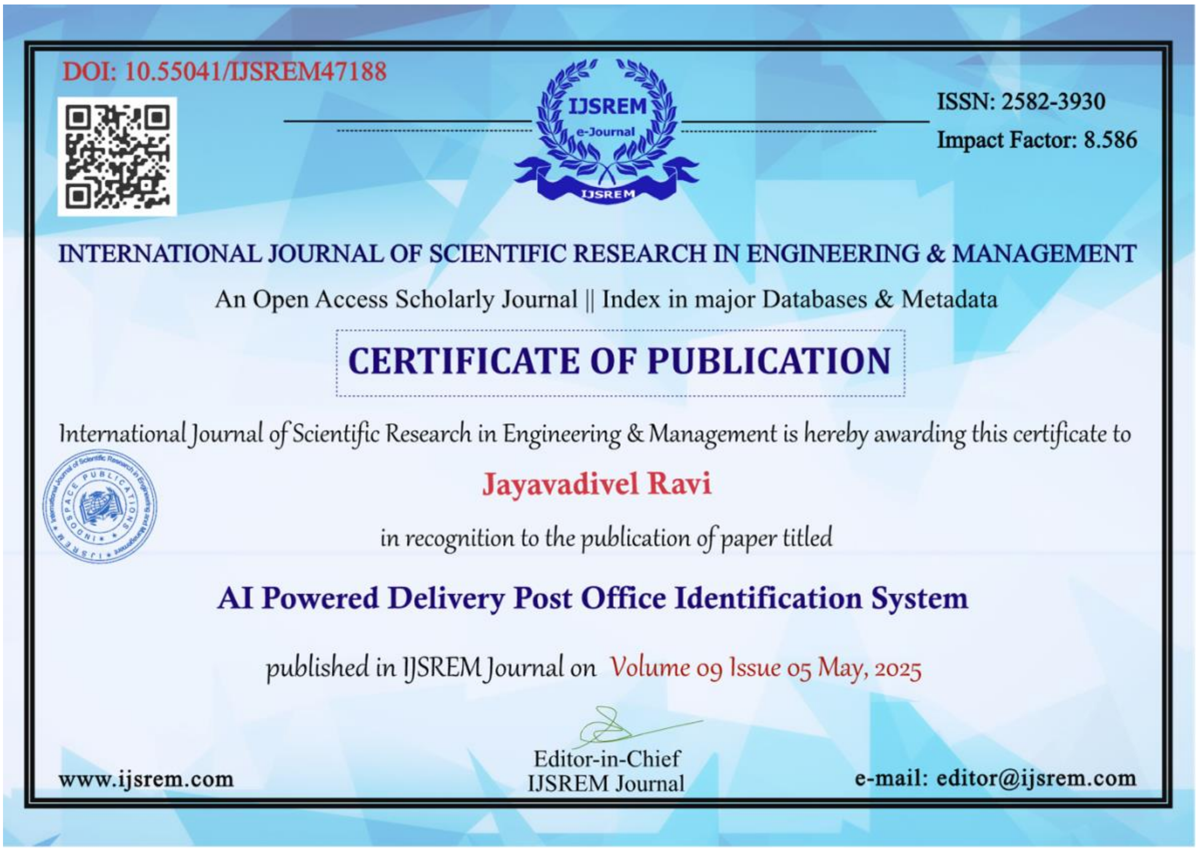
SDG 16 – Peace, Justice, and Strong Institutions

With built-in complaint and feedback features, the system encourages transparent dialogue between citizens and postal authorities. Users can lodge issues and receive responses directly through the app, which strengthens trust in public institutions. This openness and accountability promote fair treatment and better governance in postal services.

SDG 17 – Partnerships for the Goals

The platform is designed to grow through collaboration with other government departments and private partners, such as courier companies or utility services. By offering APIs or shared data models, it can integrate with electricity, water, or transport services, fostering partnerships that support wider development goals and smarter, more cohesive public service delivery.

CERTIFICATES



DOI: 10.55041/IJSREM47188

ISSN: 2582-3930

Impact Factor: 8.586

IJSREM e-Journal

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Jayavadivel Ravi

in recognition to the publication of paper titled

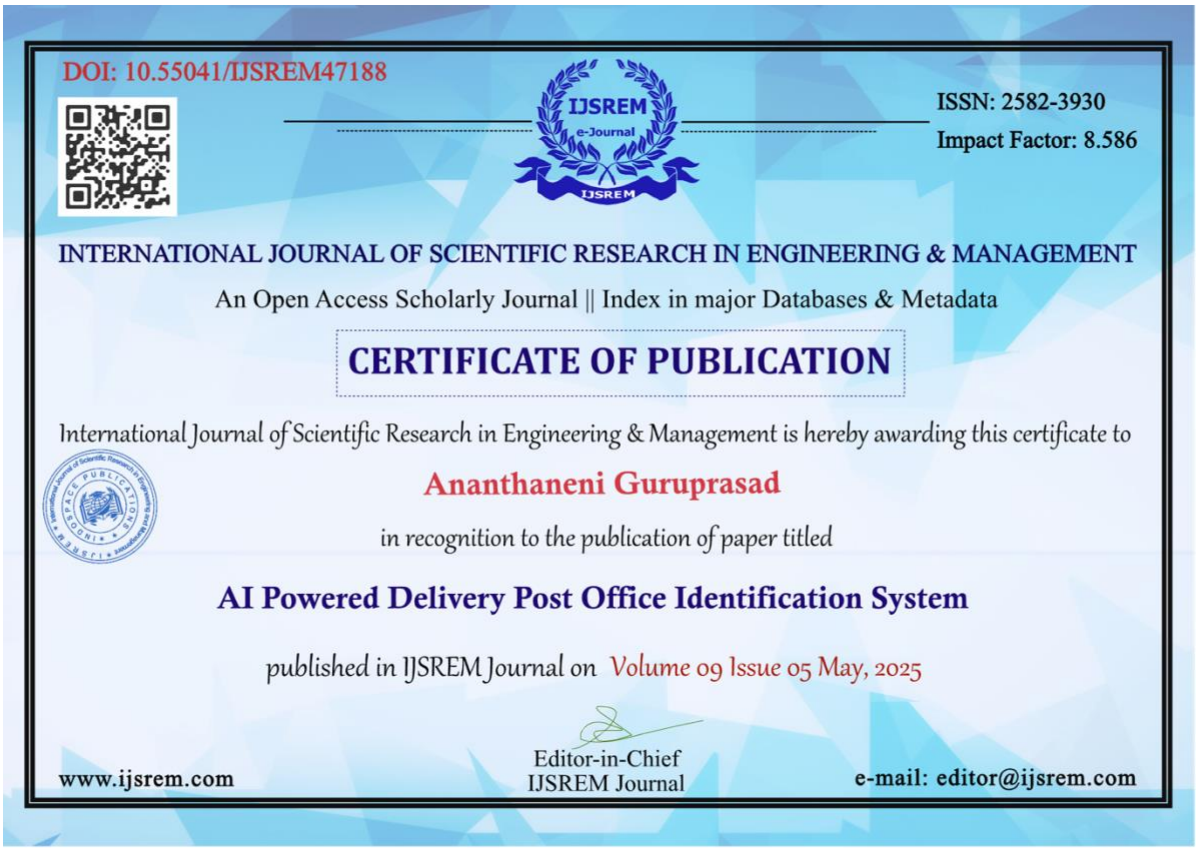
AI Powered Delivery Post Office Identification System

published in IJSREM Journal on Volume 09 Issue 05 May, 2025

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com



DOI: 10.55041/IJSREM47188

IJSREM e-Journal IJSREM

ISSN: 2582-3930
Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Ananthaneni Guruprasad

in recognition to the publication of paper titled

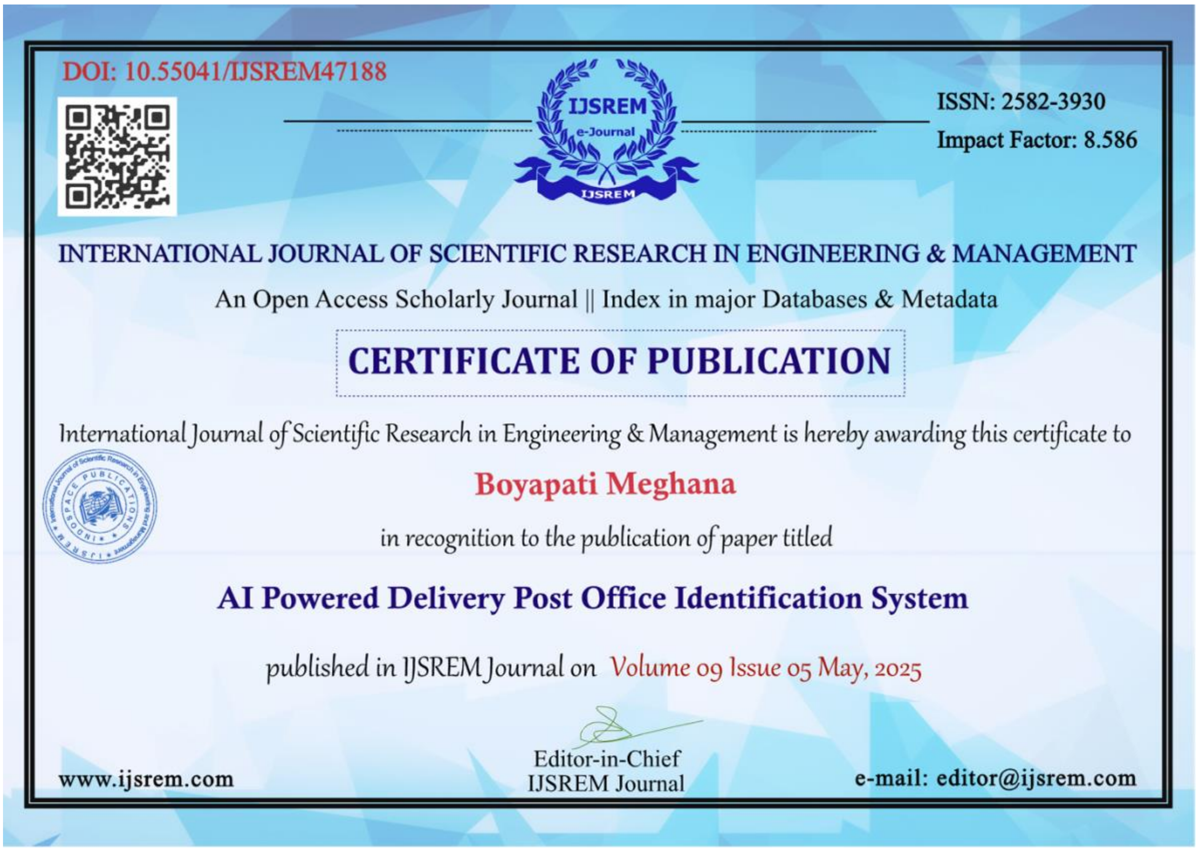
AI Powered Delivery Post Office Identification System

published in IJSREM Journal on Volume 09 Issue 05 May, 2025

www.ijsrem.com

Editor-in-Chief
IJSREM Journal

e-mail: editor@ijsrem.com



DOI: 10.55041/IJSREM47188

IJSREM e-Journal IJSREM

ISSN: 2582-3930
Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Boyapati Meghana

in recognition to the publication of paper titled

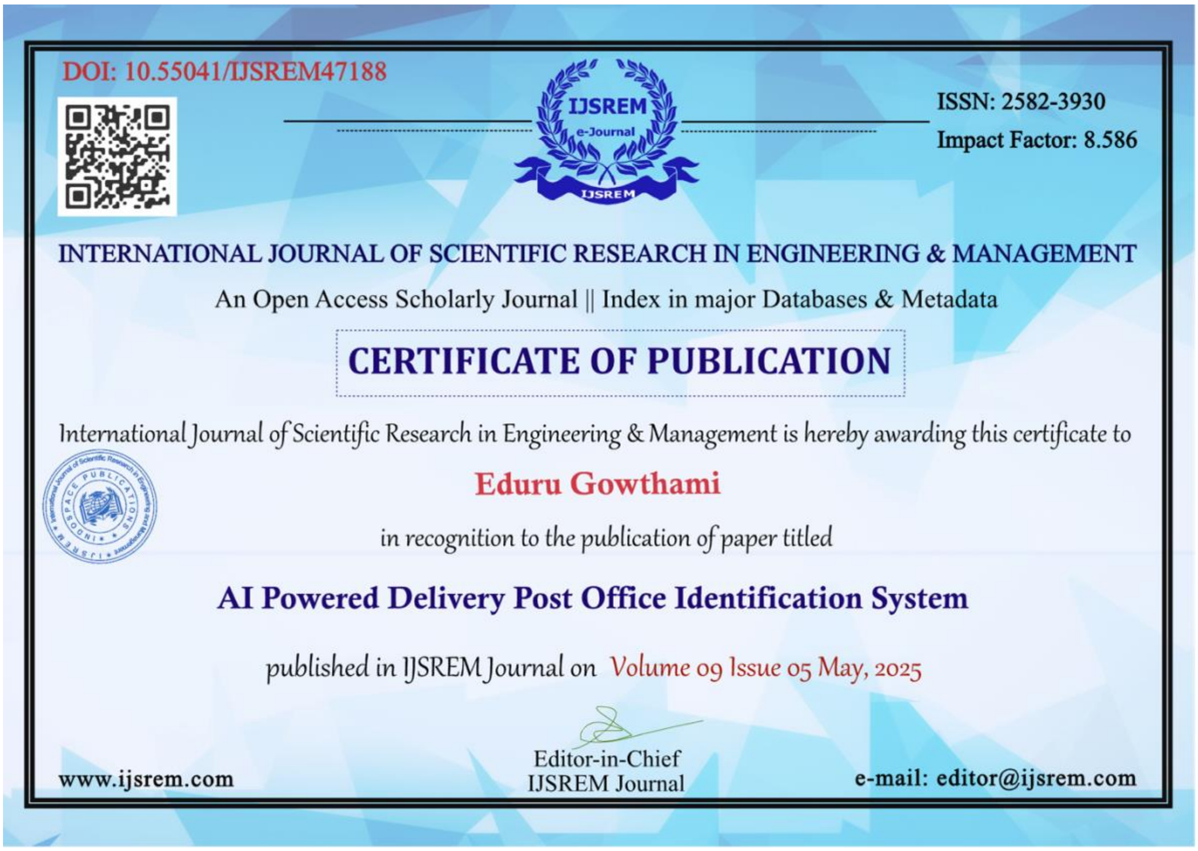
AI Powered Delivery Post Office Identification System

published in IJSREM Journal on Volume 09 Issue 05 May, 2025

www.ijsrem.com

Editor-in-Chief
IJSREM Journal

e-mail: editor@ijsrem.com



DOI: 10.55041/IJSREM47188

IJSREM e-Journal IJSREM

ISSN: 2582-3930
Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Eduru Gowthami

in recognition to the publication of paper titled

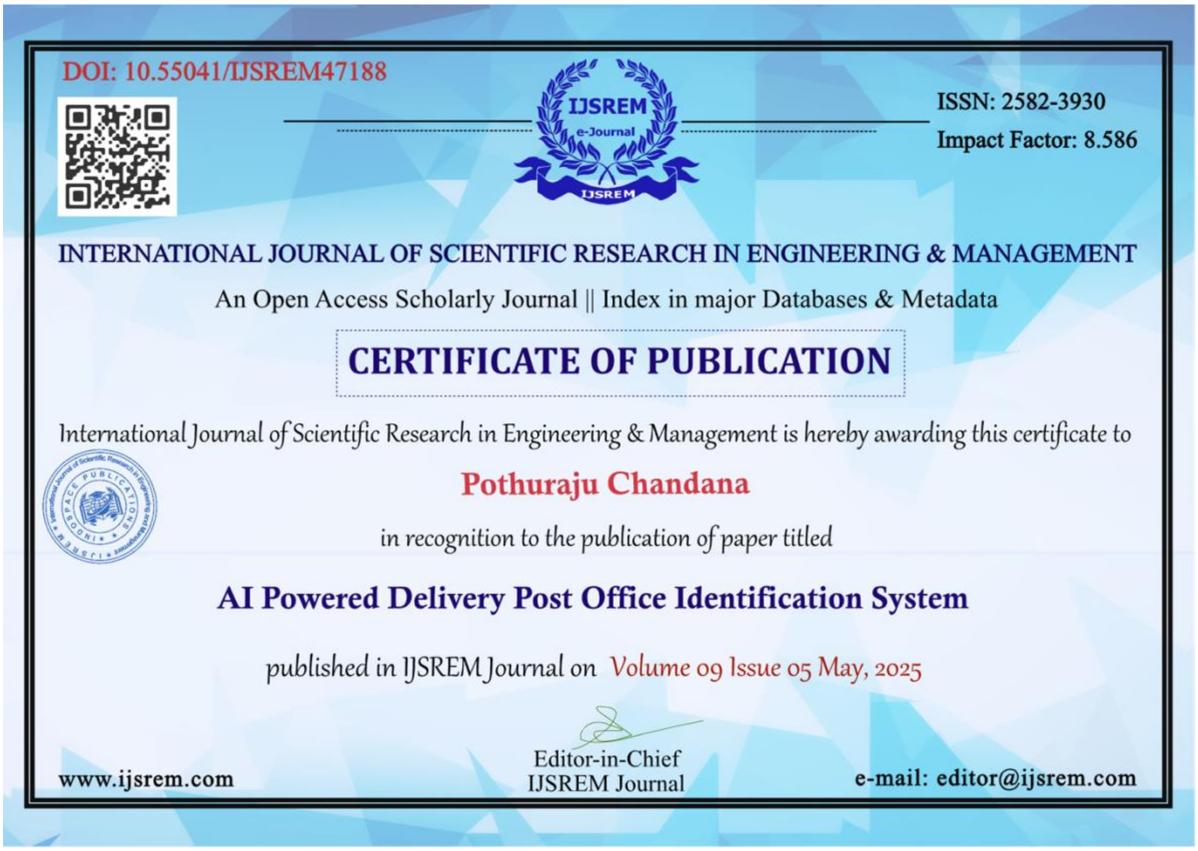
AI Powered Delivery Post Office Identification System

published in IJSREM Journal on Volume 09 Issue 05 May, 2025

www.ijsrem.com

Editor-in-Chief
IJSREM Journal

e-mail: editor@ijsrem.com



DOI: 10.55041/IJSREM47188

IJSREM e-Journal IJSREM

ISSN: 2582-3930
Impact Factor: 8.586

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Pothuraju Chandana

in recognition to the publication of paper titled

AI Powered Delivery Post Office Identification System

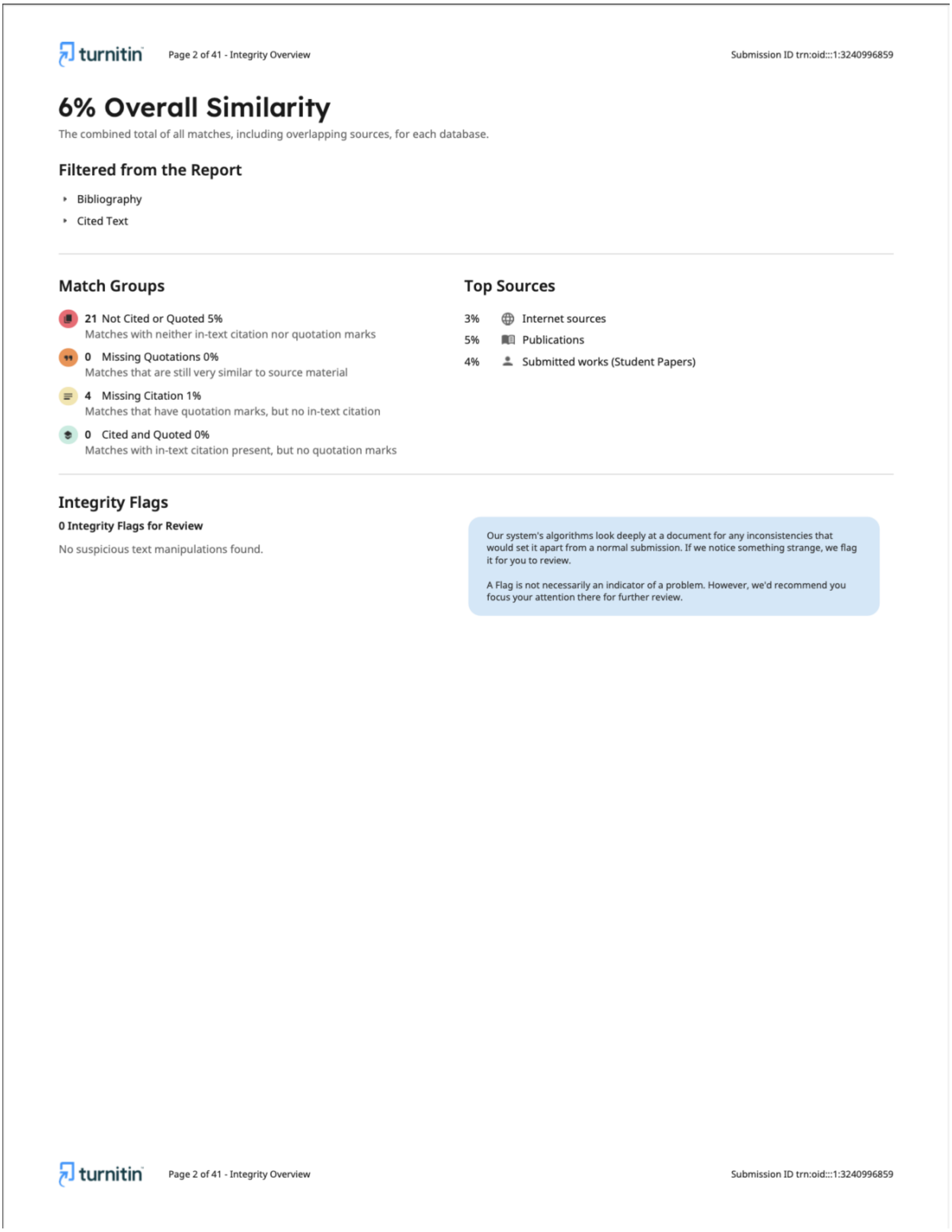
published in IJSREM Journal on Volume 09 Issue 05 May, 2025

www.ijsrem.com

Editor-in-Chief
IJSREM Journal

e-mail: editor@ijsrem.com

Plagiarism for Report



turnitin Page 2 of 41 - Integrity Overview Submission ID trn:oid:::1:3240996859

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Cited Text

Match Groups

21 Not Cited or Quoted 5%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

4 Missing Citation 1%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

3%	Internet sources
5%	Publications
4%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

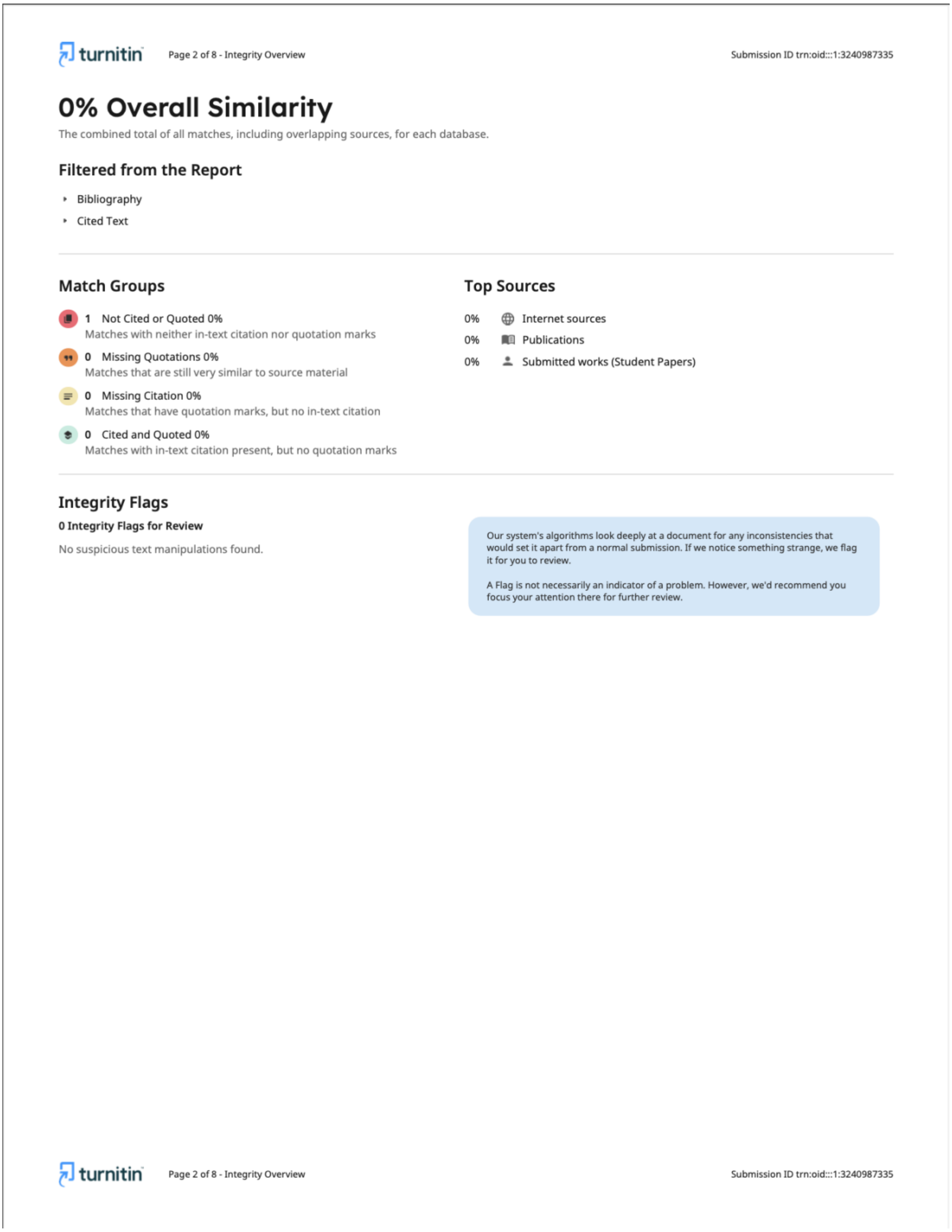
No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 41 - Integrity Overview Submission ID trn:oid:::1:3240996859

Plagiarism for Conference Paper



0% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Cited Text

Match Groups

1 Not Cited or Quoted 0%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

0%	Internet sources
0%	Publications
0%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.