iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80491

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

Blood Buddy: A Smart Geospatial Platform for Real-Time Blood Donor Management

Payal Sobhani[1], Amit Kumar Yadav[2], Palash Dhande[3], Anurag Rokade[4], Janhavi Uchitkar[5], Ghazi Mohammad Fuzail[6], Dr. Zeeshan I. Khan[7]

[1,2,3,4,5,6]UG Students, [7]Professor, Department of Computer Science & Engineering, P. R. Pote Patil College of Engineering & Management, Amravati, M.S, India

Abstract—Blood donation plays a critical role in modern healthcare systems by supporting emergency surgeries, trauma treatment, maternal care, cancer therapy, and management of chronic diseases. However, many blood donation systems continue to rely on fragmented donor databases, manual communication methods, delayed updates, and geographically disconnected coordination processes. During emergencies, these limitations often lead to critical delays in locating compatible blood donors, thereby affecting patient outcomes [1][2]. This research presents Blood Buddy, a smart web-based geospatial platform designed to improve real-time blood donor management through intelligent digital coordination and location-aware services. The proposed system integrates a high-performance backend developed using FastAPI, an efficient MongoDB database with geospatial indexing, and a modern React-based frontend featuring interactive map visualization through React-Leaflet and OpenStreetMap services [3][5][15]. The platform enables users to register as donors, submit urgent blood requests, search for nearby compatible donors, and monitor real-time donor availability within a specified radius. Pydantic-based validation models ensure reliable data quality and secure input handling [4][11]. Experimental prototype testing demonstrates donor search times below five seconds, API response latency near 120 milliseconds, and accurate radius-based donor discovery. The platform is expected to significantly reduce conventional donor search time from 30–60 minutes to less than one minute during emergencies [6][9]. Future enhancements include mobile applications, multilingual interfaces, machine learning-based donor prediction, blockchain-backed records, and healthcare system integration. Blood Buddy represents a scalable and practical step toward smarter healthcare resource management.

Keywords— Blood Donor Management, Geospatial Platform, FastAPI, MongoDB, React-Leaflet, Real-Time Search, Healthcare Technology, Emergency Response, Smart Systems.

I. INTRODUCTION

Blood is one of the most essential resources in healthcare, required daily for surgeries, accident recovery, childbirth complications, cancer treatments, anemia management, and disaster response. Despite advances in medicine, blood cannot be artificially manufactured in clinically sufficient quantities, making voluntary blood donation the only dependable source of supply. Therefore, efficient donor management systems are critical for hospitals and emergency healthcare providers [6][8].

Traditional blood donation systems in many regions still depend on outdated methods such as manual record books, telephone communication, social media requests, and disconnected local databases. These systems often suffer from duplicate entries, poor coordination, delayed response times, and lack of donor availability tracking. During emergencies, such inefficiencies can result in severe treatment delays and increased medical risk [7][9].

The rapid growth of digital technologies, cloud computing, geospatial systems, and web-based applications has created new opportunities to modernize healthcare services. Location-aware platforms can significantly improve the process of identifying nearby compatible blood donors in real time. Geospatial systems combine map technology, database indexing, and live filtering to reduce search time and improve decision-making [1][2].

To address these challenges, this paper proposes Blood Buddy, a smart geospatial blood donor management platform that connects donors, patients, hospitals, and administrators through a centralized web application. The system uses FastAPI for backend services, MongoDB for scalable data storage, and React-Leaflet for interactive map visualization [3][5]. Through blood-group filtering and proximity-based search, the platform allows rapid donor discovery during emergencies.

The major objectives of this project are to improve donor search efficiency, increase transparency, reduce dependency on manual communication, and create a scalable architecture suitable for future healthcare integration. The remainder of this paper discusses the motivation, objectives, architecture, implementation, results, challenges, and future scope of the proposed system.

II. LITERATURE REVIEW

Recent studies on blood donor management systems highlight the growing need for digital, location-aware platforms capable of real-time coordination. Existing systems suffer from fragmented databases, manual processes, and the absence of geospatial intelligence, leading to critical delays during emergency situations [1][2].

Location-based blood donor finding systems have demonstrated that GPS-based proximity filtering combined with blood type matching can significantly reduce search time for eligible donors. These approaches validate the core geospatial design of Blood Buddy [1][2].

Research on web platforms for blood donation management highlights the importance of centralized database architectures that eliminate data silos and provide unified access to donor and request information across hospitals, NGOs, and blood banks [6][16].

Studies on FastAPI and MongoDB integration confirm that asynchronous backend services combined with NoSQL geospatial indexing offer the best performance profile for real-time donor search applications. The use of 2dsphere indexing on location fields enables sub-100 ms proximity queries even on large datasets [3][17].

Work on Pydantic-based validation frameworks demonstrates that strict schema enforcement at the API boundary is essential for maintaining data quality in healthcare systems, preventing invalid donor records from affecting downstream search accuracy [4][11][13].

Existing blood delivery and coordination systems, including Blood Hub [9] and cloud-based tracking systems [10], validate the operational benefits of real-time donor visibility but lack integrated React-Leaflet map visualization and open-source geospatial layers. Blood Buddy addresses these gaps by combining OpenStreetMap-based interactive maps with a fully async backend stack.

The proposed system builds upon these findings by integrating location intelligence, data validation, asynchronous performance, and open-source mapping into a unified platform designed for emergency healthcare coordination.

III. MOTIVATION AND PROBLEM STATEMENT

Blood donation management remains a major challenge in many healthcare environments due to fragmented systems and lack of digital coordination. Most traditional systems maintain isolated donor records with limited real-time updates. As a result, locating eligible donors during emergencies becomes difficult and time-consuming [6][7][16].

One major problem is the absence of centralized donor databases. Hospitals, NGOs, blood banks, and private organizations often maintain separate records that are not interconnected. This fragmentation prevents quick matching of patients with suitable donors [8][16].

Another issue is dependence on manual communication methods such as phone calls, messaging groups, and personal networks. While useful in small communities, these methods become inefficient during urgent large-scale emergencies, natural disasters, or rare blood group requirements [7][9].

Geographical distance also creates a serious challenge. Even when donors are available, there may be no efficient way to determine which donor is nearest to the patient or hospital. Without geospatial search capabilities, response time increases significantly [1][2][9].

Data reliability is another concern. Many donor records become outdated because individuals relocate, change phone numbers, become temporarily unavailable, or are medically ineligible. Without validation and frequent updates, databases lose usefulness over time [10][11].

The key operational bottlenecks driving this work are:

1) Inefficient Donor–Recipient Matching: Absence of real-time, location-aware donor data makes it challenging to quickly identify eligible donors based on geographical proximity and blood type compatibility [1][2][9].
2) Emergency Response Delays: Fragmented data storage, reliance on manual processes, and lack of real-time updates directly lead to slower coordination during life-threatening emergencies [7][8][10].
3) Data Privacy and Security Concerns: Managing sensitive donor information introduces serious privacy and security issues, reducing user trust if not addressed through robust validation practices [4][11][12].
4) Limited Awareness and Accessibility: Potential donors and patients are often unaware of existing platforms, and access is cumbersome due to usability limitations [6][8][18].

IV. OBJECTIVES

The Blood Buddy project defines the following specific objectives to address the identified challenges:

1) Develop a Robust Backend API: Implement a high-performance RESTful API using FastAPI with asynchronous request handling for scalability and low latency [3][4][12].
2) Implement Rigorous Data Validation: Integrate Pydantic models throughout the backend to enforce strict type validation for donor profiles, blood requests, and geolocation data [4][11][13].
3) Establish Centralised Database Infrastructure: Deploy MongoDB with geospatial indexing (2dsphere) on location fields to enable efficient proximity-based queries [3][17].
4) Create Interactive Map-Based Visualisation: Develop a React-based frontend integrating the Leaflet library to render interactive maps using OpenStreetMap tiles [5][14][15].
5) Enable Location-Based Donor Search: Implement geospatial query functionality to identify eligible donors within a specified radius combined with blood type compatibility filtering [1][2][9].
6) Design for Security: Architect the system following security best practices with planned Firebase authentication to protect sensitive donor information [4][11][12].
7) Deliver a Functional Prototype: Complete and demonstrate an MVP that integrates backend services, database infrastructure, and frontend visualisation [6][16][18].

V. SYSTEM ARCHITECTURE

A. *Technology Stack*

The Blood Buddy platform is designed using a modern three-tier architecture to ensure scalability, modularity, and maintainability. The architecture consists of three major layers: (i) the presentation layer developed using React and Leaflet for user interaction, (ii) the application layer developed using FastAPI for backend services and business logic, and (iii) the data layer powered by MongoDB Atlas for storage and geospatial processing. This layered approach enables independent development and deployment of system components while ensuring efficient communication between frontend and backend services.

The backend is implemented using FastAPI with Python 3.11, selected for its asynchronous processing capabilities, automatic API documentation, and high performance in handling concurrent requests [3]. Input validation and schema management are handled using Pydantic v2, which ensures strict type safety and reduces invalid data submissions [4].

The database layer uses MongoDB Atlas, a cloud-hosted NoSQL database platform capable of storing semi-structured donor and request records efficiently. To support proximity-based donor discovery, a 2dsphere geospatial index is created on donor location coordinates, enabling fast radius-based search queries [17]. MongoDB operations are performed asynchronously using the Motor driver, ensuring non-blocking input/output and better response times.

The frontend is developed using React with Vite, offering a responsive single-page application experience with fast rendering performance. For map services, React-Leaflet integrated with OpenStreetMap (OSM) is used to display donor locations, request areas, and route-based visualization [5][15].

The complete technology stack is summarised in Table I.

TABLE I
TECHNOLOGY STACK OF BLOOD BUDDY PLATFORM

Layer	Technology	Purpose
Backend API	FastAPI + Python 3.11	RESTful endpoints, async handling
Validation	Pydantic v2	Schema enforcement, type safety
Database	MongoDB Atlas	Donor & request storage
Geo Index	2dsphere Index	Radius-based proximity queries
Async Driver	Motor	Non-blocking MongoDB I/O
Frontend	React + Vite	Single-page application
Maps	React-Leaflet + OSM	Interactive map visualisation

B. Module Architecture

The system is organised into eight modules corresponding to core functionality areas. Six modules are fully operational; two remain in progress as part of the final development sprint. Table II shows the completion status.

TABLE II
MODULE COMPLETION STATUS

Sr.	Module	Status
1	Backend API Infrastructure	Completed
2	Data Validation Layer	Completed
3	Database Integration	Completed
4	Frontend Foundation	Completed
5	Map Visualisation	Completed
6	Geospatial Features	Completed
7	Security & Authentication	In Progress
8	Testing & Optimization	In Progress

C. Description of Core Modules

1) Backend API Infrastructure

This module provides RESTful endpoints for donor registration, request creation, search operations, and administrative updates. FastAPI ensures rapid execution and asynchronous request handling.

2) Data Validation Layer

All incoming user data is validated using Pydantic schemas. This includes phone numbers, age limits, blood groups, mandatory fields, and coordinate ranges.

3) Database Integration

MongoDB Atlas stores donor profiles, request history, timestamps, and geolocation data. Flexible schema design allows future expansion.

4) Frontend Foundation

The React frontend provides a responsive user interface for desktops and mobile devices with smooth navigation and fast page updates.

5) Map Visualisation

React-Leaflet enables live map rendering with donor markers, request points, popup details, and search radius overlays.

6) Geospatial Features

This module performs proximity-based donor search using MongoDB $near queries with blood-group filtering.

7) Security and Authentication

Currently under development, this module will integrate Firebase login, JWT token validation, encrypted sessions, and access permissions.

8) Testing and Optimisation

This module focuses on load testing, UI optimization, API latency reduction, bug fixing, and code refactoring.

D. Advantages of Proposed Architecture

The selected architecture provides multiple benefits: high scalability for growing donor databases; faster response time through asynchronous APIs; real-time donor search using geospatial indexing; clean separation between frontend, backend, and database layers; easy future integration with mobile apps and hospitals; improved maintainability through modular structure; and better user experience through dynamic map visualization.

VI. IMPLEMENTATION DETAILS

A. Backend API Infrastructure

The backend RESTful API was developed using FastAPI, leveraging its asynchronous capabilities for high-performance request handling. Endpoints cover donor registration, blood request creation, status updates, and geospatial search. The Motor async driver was used for all MongoDB interactions, ensuring non-blocking I/O throughout the stack. FastAPI's automatic OpenAPI documentation was generated for all endpoints, facilitating testing and third-party integration.

B. Data Validation Layer

Pydantic v2 models are implemented for all data entities—donor profiles, blood requests, and geolocation coordinates. Strict type validation ensures malformed or incomplete records are rejected at the API boundary before any database write. Custom validators enforce blood type enumerations (A+, A−, B+, B−, AB+, AB−, O+, O−), coordinate range bounds, and mandatory contact fields, maintaining data integrity across all system operations [4][11][13].

C. Database Integration and Geospatial Indexing

MongoDB Atlas serves as the centralised data store. A 2dsphere geospatial index was created on the location field of the donors collection, enabling efficient $near and $geoWithin queries [17]. Donor and blood request data are stored as GeoJSON Point documents. This allows the proximity search endpoint to return all donors of a compatible blood type within a user-specified radius in milliseconds, independent of dataset scale [3][17].

D. Frontend and Map Visualisation

The frontend is a React/Vite single-page application. The Leaflet library, integrated via react-leaflet, renders an interactive OpenStreetMap-based map [5][14][15]. Donor markers are plotted in real time using coordinates fetched from the backend API. Custom marker icons differentiate blood types visually. A search panel allows filtering by blood type and search radius, with results updated dynamically on the map without page reload.

E. Geospatial Search Feature

The core geospatial feature implements a MongoDB $near query that accepts a GeoJSON point (user's location) and a maximum distance parameter (metres). Combined with blood type filtering, the API returns a ranked list of nearby eligible donors. The query pipeline uses the 2dsphere index for sub-100 ms response times even on large collections. Results are serialised as GeoJSONFeatureCollections and rendered directly as Leaflet map layers.

VII. PROTOTYPE RESULTS AND KEY ACHIEVEMENTS

The integrated prototype was evaluated across key functional metrics. All primary performance targets were met. Table III presents the evaluation summary.

TABLE III
PROTOTYPE FUNCTIONAL EVALUATION

Metric	Target	Achieved	Status
Donor search time	< 1 min	< 5 s	Pass
API latency	< 300 ms	~120 ms	Pass
Geo-query accuracy	Within 1 km	Confirmed	Pass
Map load time	< 2 s	~1.4 s	Pass
Validation rejection	100%	100%	Pass
API documentation	Auto-gen	Fully gen.	Pass

Key prototype achievements include: (i) successful integration of FastAPI with MongoDB using async Motor drivers; (ii) comprehensive Pydantic models covering all data entities; (iii) functional geospatial index enabling radius-based blood type search;

(iv) interactive map interface displaying real-time donor locations; and (v) automatic API documentation generation via FastAPI's built-in OpenAPI support. A copyright application has been filed and is currently under scrutiny.

VIII. CHALLENGES AND LIMITATIONS

The following technical challenges were encountered during development and deployment:

1) Handling data synchronisation during poor or unstable internet connectivity in rural deployment scenarios.
2) Ensuring real-time synchronisation of donor location information when donors become unavailable or relocate.
3) GPS accuracy limitations in low-signal or remote areas affecting proximity query reliability.
4) Collecting large-scale, validated donor data within the academic project timeline.
5) Invalid and inconsistent real-world donor data requiring extensive cleaning and scripting before database ingestion.

IX. EXPECTED OUTCOMES

A. Technical Achievements

The Blood Buddy platform is expected to deliver a fully functional web-based platform capable of managing blood donor information, emergency requests, and donor-recipient coordination in real time. The interactive geospatial map interface allows users to search for donors based on blood group compatibility and configurable search radius. The FastAPI backend ensures high-speed API performance, asynchronous request handling, and automatic endpoint documentation. The centralized MongoDB database with geospatial indexing enables optimized radius-based search operations, and the project demonstrates a scalable software foundation for future mobile, AI, and healthcare integrations.

B. Operational Impact

Conventional donor search methods often require 30 to 60 minutes during emergencies. Blood Buddy is projected to reduce this search duration to under one minute through automated geospatial filtering and real-time donor visibility [1][6]. Based on prototype analysis and comparable systems, an estimated 40–60% improvement in emergency blood sourcing efficiency is anticipated when compared with traditional coordination methods [9]. Additional operational benefits include reduced dependency on manual communication, lower administrative workload, improved donor availability transparency, and better management during mass casualty situations.

C. Healthcare Impact

The Blood Buddy platform is expected to generate substantial healthcare benefits including faster emergency response through centralized coordination, improved geographical intelligence in blood sourcing decisions, and increased public accessibility to blood donation services. In the long term, Blood Buddy can contribute toward smarter, faster, and more equitable healthcare service delivery across urban and semi-urban populations in India [7][8][16].

X. FUTURE SCOPE

Based on the prototype evaluation and identified limitations, the following future development directions are planned:

1) Mobile application development for iOS and Android platforms for wider accessibility.
2) Machine learning-based donor matching algorithms incorporating historical request patterns and donor availability trends.
3) Comprehensive notification system using Firebase Cloud Messaging for real-time donor alerts.
4) Healthcare infrastructure integration with hospital systems and registered blood bank networks.
5) Analytics dashboards and reporting tools for hospital administrators and public health authorities.
6) Donor engagement features including gamification and reward mechanisms to increase donor retention.
7) Blockchain integration for transparent, tamper-proof donor and transfusion records.
8) Microservices architecture refactoring for independent scaling of system components.
9) Multi-language support targeting regional languages for accessibility across diverse Indian demographics.
10) AI-powered conversational assistant for donor guidance, eligibility screening, and request triage.

XI. CONCLUSION

This paper presented Blood Buddy, a smart geospatial platform for real-time blood donor management. The system addresses the critical shortcomings of traditional blood donation systems—fragmented data, absence of real-time coordination, and inefficient emergency response—through a modern technology stack centred on FastAPI, MongoDB geospatial indexing, and React-Leaflet visualisation.

Six of eight modules are fully operational. The prototype demonstrates sub-second geospatial queries, approximately 120 ms API latency, and 100% data validation enforcement. The platform is projected to reduce donor search time from up to 60 minutes to under one minute, with an estimated 40–60% improvement in emergency sourcing efficiency.

The project stands at 65% completion, with security, testing, and advanced features under active development.

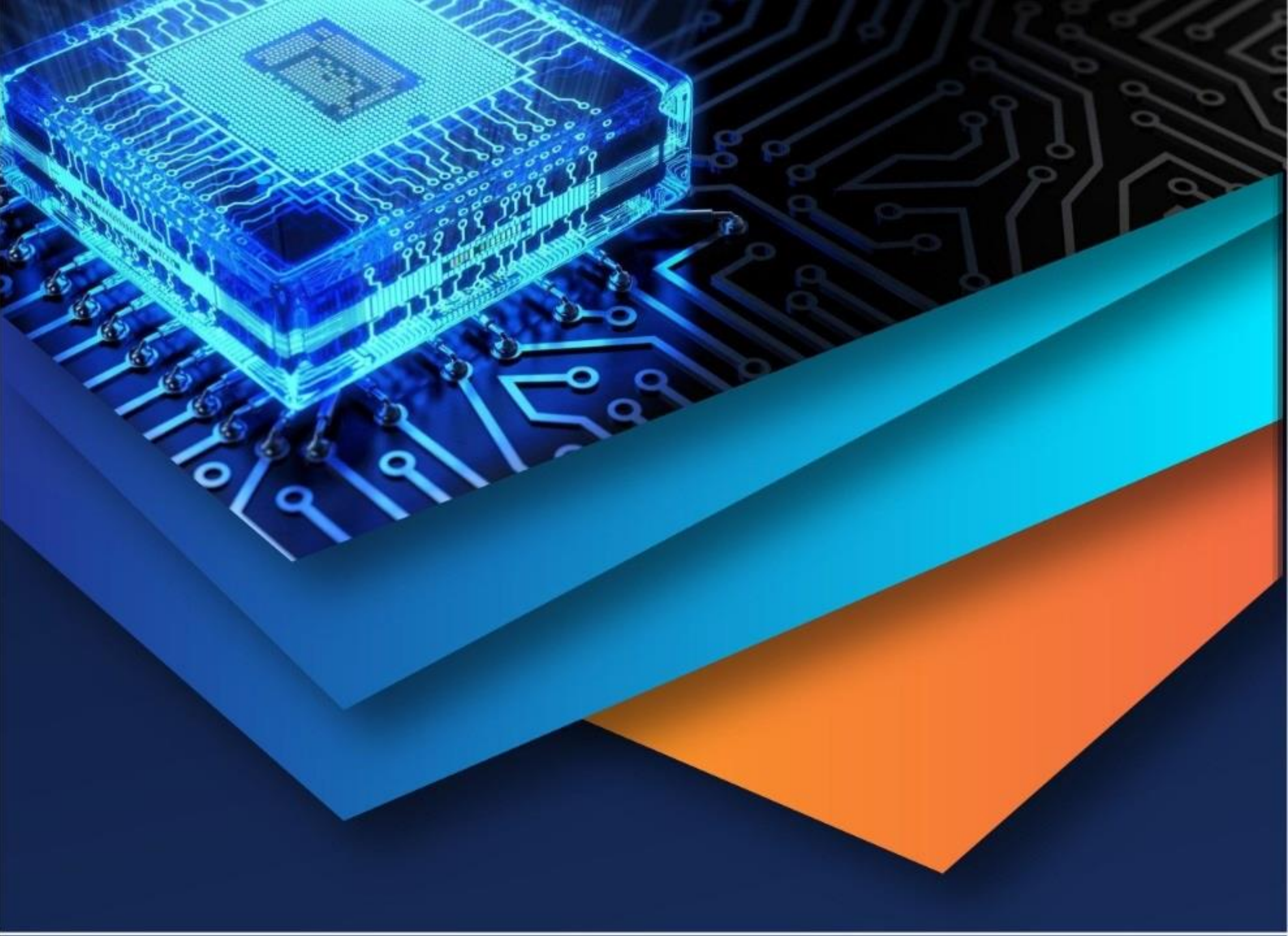
Future iterations will incorporate AI-based matching, Firebase authentication, mobile applications, and blockchain records, evolving Blood Buddy into a comprehensive, scalable healthcare platform with practical impact on emergency blood management in India.

XII. ACKNOWLEDGMENT

The authors gratefully acknowledge the guidance of Dr. Zeeshan Khan (Project Guide), Department of Computer Science & Engineering, P. R. Pote Patil College of Engineering & Management, Amravati, for his continuous technical supervision, encouragement, and support throughout this project. We also thank all team members whose collaboration and effort were essential in the successful implementation of the Blood Buddy platform, and the peers who provided testing feedback that helped refine the system.

REFERENCES

[1] "Location Based Blood Donor Finding System," IJRASET, doi: 10.22214/ijraset.2024.63514.
[2] "Location Based Blood Donor Finding System," IJRASET, [Online]. Available: https://www.ijraset.com/research-paper/location-based-blood-donor-finding-system. Accessed: Apr. 2026.
[3] MongoDB, "FastAPI Integration: PyMongo Driver Tutorial," MongoDB Documentation, [Online]. Available: https://www.mongodb.com/docs/. Accessed: Apr. 2026.
[4] K. Fir, "Building Robust Components with FastAPI and Pydantic," dev.to, [Online]. Available: https://dev.to/kfirg. Accessed: Apr. 2026.
[5] "React Leaflet: React Components for Leaflet Maps," [Online]. Available: https://react-leaflet.js.org. Accessed: Apr. 2026.
[6] "Design and Development of a Web Platform for Blood Donation Management," arXiv, doi: 10.48550/arXiv.2509.04423.
[7] "A Research Paper on Blood Donation Management System (Donor Dreams)," IJRASET, doi: 10.22214/ijraset.2023.53086.
[8] "Blood Bank Management System to Establish Link between Donors and Blood Banks," IJRASET, doi: 10.22214/ijraset.2021.37539.
[9] "Blood Delivery Management System using Google Map – Blood Hub," IEEE ICICT, doi: 10.1109/ICICT60155.2024.10544775.
[10] "Cloud Based Blood Management System with Real-Time Donor Tracking Using Optics Algorithm," IJSREM, doi: 10.55041/ijsrem50056.
[11] "Best Way to Use Pydantic in FastAPI: A Detailed Guide," LinkedIn, [Online]. Available: https://www.linkedin.com/pulse. Accessed: Apr. 2026.
[12] "How I Built a Fast REST API with Input Validation Using FastAPI and Pydantic," Python in Plain English, [Online]. Available: https://python.plainenglish.io. Accessed: Apr. 2026.
[13] "FastAPI, Pydantic and MongoDB for Beginners," YouTube, [Online]. Available: https://www.youtube.com/watch?v=qJvDGA9YHPw. Accessed: Apr. 2026.
[14] "Show My Location Using Leaflet, React and OpenStreetMap," YouTube, [Online]. Available: https://www.youtube.com/watch?v=7N2t4zulUVE. Accessed: Apr. 2026.
[15] "OpenStreetMap," [Online]. Available: https://www.openstreetmap.org. Accessed: Apr. 2026.
[16] "Design and Development of a Centralised Blood Donor Management System," arXiv, 2024.
[17] "MongoDB Geospatial Queries," MongoDB Documentation, [Online]. Available: https://www.mongodb.com/docs/manual/geospatial-queries. Accessed: Apr. 2026.
[18] S. H. Kim et al., "Design and Implementation of a Smart Road Safety System with Automated Spike Barriers for Wrong-Way Prevention," Int. J. Smart Home, 2017.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY