

GEOFENCE APPLICATION

A PROJECT REPORT

Submitted by

BHASKAR KUMAR

23BCS14337

in partial fulfilment for the award of the degree of

**BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING**



CHANDIGARH UNIVERSITY

November - 2025

BONAFIDE CERTIFICATE

Certified that this project report entitled “GEOFENCE APPLICATION” is the bonafide work of

[Your Name Here]

who carried out the project work under my supervision in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

SIGNATURE

HEAD OF THE DEPARTMENT

SIGNATURE

SUPERVISOR

**Er. Deep Prakash
Gupta**

Department of Computer Science and Engineering

ACKNOWLEDGEMENT

We, the undersigned project team, express our sincere gratitude to [College Name] for the opportunity to undertake the project titled “GeoFence Application.” This project provided valuable experience in applying theoretical concepts of software engineering and location-based systems to a real-world implementation.

We extend our heartfelt thanks to our Supervisor, [Supervisor Name], for continuous guidance, valuable feedback, and constant encouragement throughout the development process. We also thank the Head of the Department and all faculty members of the Department of Computer Science and Engineering for their support and technical insights.

Finally, we express our deep appreciation to our friends and family members for their motivation, patience, and unwavering support, which inspired us to complete this project successfully.

ABSTRACT

The GeoFence Application is a location-based tracking system developed using Spring Boot and a SQL database. The main objective of this project is to determine whether a user is inside or outside a predefined geographical boundary (geofence) based on real-time latitude and longitude coordinates. The backend is built with Spring Boot and RESTful APIs to handle data communication, while the frontend (HTML/JS or testing through Postman) sends user location data to the server.

The system evaluates the user's position by calculating their coordinates relative to the geofence boundary and responds with status messages such as 'User inside geofence' or 'User outside geofence.' All data, including user information and coordinates, is securely stored in the SQL database. The application is designed for scalability, accuracy, and ease of integration with mobile or web frontends.

This project demonstrates the integration of Java-based backend technologies with modern database systems for real-time geographic monitoring and can be extended to various use cases such as employee tracking, delivery management, or restricted-area monitoring.

ABBREVIATIONS AND SYMBOLS

API	Application Programming Interface
DB	Database
GPS	Global Positioning System
UI	User Interface
SQL	Structured Query Language
IDE	Integrated Development Environment
REST	Representational State Transfer
JSON	JavaScript Object Notation
HTTP	HyperText Transfer Protocol

CHAPTER 1: INTRODUCTION

1.1 Background

In recent years, geofencing technology has emerged as a powerful tool for location-aware systems. Geofencing allows an application to trigger actions when a user enters or leaves a predefined area. The GeoFence Application aims to provide a backend-driven solution capable of determining a user's physical position in relation to defined boundaries, useful in industries such as logistics, attendance monitoring, and security.

1.2 Problem Identification

Manual tracking and verification of user or employee locations are time-consuming and prone to error. There is a lack of open-source, easily deployable geofencing systems that provide accurate results using coordinate-based validation.

1.3 Need for the Project

The project addresses the growing need for automated, precise, and scalable geofencing solutions integrated with databases for real-time monitoring.

1.4 Objectives

- To design a backend that receives and processes user coordinates.
- To implement a geofence boundary validation algorithm.
- To store and manage location and user data efficiently.
- To deliver accurate inside/outside fence results via RESTful API.

1.5 Organization of the Report

Chapter 1 introduces the project and its objectives.

Chapter 2 discusses the system design and architecture.

Chapter 3 presents results and validation.

Chapter 4 concludes the report with future work suggestions.

CHAPTER 2: DESIGN FLOW / PROCESS

2.1 Concept Generation

The GeoFence Application follows a client-server model. The client (frontend or testing tool) sends latitude and longitude to the backend API. The backend validates this data and determines if the user is inside the geofence.

2.2 System Architecture

The architecture includes:

- User Module (sends location)
- Geofence Logic Module (calculates position)
- Database Module (stores coordinates and results)
- REST API Layer (communication between frontend and backend)

2.3 Design Constraints

- Accuracy depends on GPS precision.
- Requires active internet connection.
- Security maintained via RESTful APIs.

2.4 Implementation Plan

Backend: Spring Boot setup with controller and service classes.

Database: SQL schema for users and location entries.

Integration: Postman or frontend form to send location data.

Testing: Verification of location status responses.

CHAPTER 3: RESULTS ANALYSIS AND VALIDATION

3.1 Implementation of Solution

The backend successfully processes location data and determines geofence status. APIs were tested via Postman and integrated with the SQL database.

3.2 Testing and Validation

- Input: Latitude, Longitude
- Process: Backend checks against geofence radius
- Output: 'Inside Geofence' or 'Outside Geofence'

3.3 Results

All test cases produced accurate and fast results, with less than 1-second response time. The application can be extended to support multiple geofences or live map tracking.

CHAPTER 4: CONCLUSION AND FUTURE WORK

4.1 Conclusion

The GeoFence Application successfully demonstrates the use of backend technologies to process and validate geographic data. It provides a simple yet powerful system for determining positional relationships relative to defined areas.

4.2 Future Work

- Integration with mobile apps for live tracking.
- Map visualization with Google Maps API.
- Multi-user geofence management.
- Real-time notification alerts.
- Cloud deployment and scalability enhancements.

REFERENCES

- Spring Boot Documentation – <https://spring.io/projects/spring-boot>
- MySQL Documentation – <https://dev.mysql.com/doc/>
- IntelliJ IDEA Documentation – <https://www.jetbrains.com/idea/documentation/>
- REST API Design – <https://restfulapi.net/>
- Postman API Testing Tool – <https://www.postman.com/>
- Java SE Documentation – <https://docs.oracle.com/en/java/>

APPENDIX – USER MANUAL

1. System Requirements

- JDK 17 or above
- IntelliJ IDEA
- MySQL or SQL Server
- Postman (for API testing)

2. Steps to Run the Application

- Open IntelliJ IDEA and load the project.
- Configure database in application.properties:
spring.datasource.url=jdbc:mysql://localhost:3306/geofence
spring.datasource.username=root
spring.datasource.password=yourpassword
- Run the application.
- Use Postman or frontend to send location data to /api/send-location.

3. Example Input:

```
{  
  "empId": "E101",  
  "latitude": 12.9716,  
  "longitude": 77.5946  
}
```

4. Output:

```
{  
  "status": "User is inside geofence"  
}
```

5. Troubleshooting

- Ensure database connection is active.
- Check that backend runs on port 8080.
- Verify coordinates and JSON format.