# Opportuneer (Explore to Excel)

## Overview

Opportuneer is a specialized job portal designed to address the challenges students and graduates face in securing part-time jobs and sponsorship opportunities. This platform aims to bridge the gap between skilled job seekers and employers by providing tailored features that simplify recruitment workflows.

## Features

• Job Posting: Employers can create and manage job listings with detailed descriptions and requirements.  
• Applicant Tracking: Employers can review applications and track candidates effectively.  
• User-Friendly Interface: A seamless and intuitive experience for job seekers and employers.  
• Tailored Opportunities: Focused on part-time jobs and sponsorships for students and recent graduates.  
• Secure Data Handling: Implements GDPR-compliant practices to ensure data security.

## Technologies Used

• Frontend: React.js (Dynamic forms, responsive design, and state management with React hooks)  
• Backend: Spring Boot (RESTful APIs, JWT authentication, and scalable architecture)  
• Database: MySQL (Optimized schema with query indexing and pagination)  
• Testing: JUnit (Backend unit testing) and Postman (API testing)

## System Architecture

The platform is built on a three-tier architecture:  
1. Frontend: React.js for an interactive user interface.  
2. Backend: Spring Boot for business logic and API management.  
3. Database: MySQL for storing job postings, user profiles, and application data.

## Installation and Setup

### Prerequisites  
• Java Development Kit (JDK) installed  
• MySQL Database installed  
• Node.js installed  
• Git installed

### Steps  
1. Clone the repository:  
 ```bash  
 git clone https://github.com/adhivishnu121/Opportuneer.git  
 cd Opportuneer  
 ```  
2. Setup Backend:  
 - Navigate to the backend directory.  
 - Update `application.properties` with your MySQL credentials.  
 - Build and run the Spring Boot application.  
3. Setup Frontend:  
 - Navigate to the frontend directory.  
 - Install dependencies:  
 ```bash  
 npm install  
 ```  
 - Start the React application:  
 ```bash  
 npm start  
 ```  
4. Run the Application:  
 - Access the platform at `http://localhost:3000`.

## Testing

• Backend: Run JUnit tests included in the `test` folder.  
• API Testing: Postman test cases are provided in the project (see Appendix B of the report).  
• Frontend: Validate UI and API integration through browser testing.

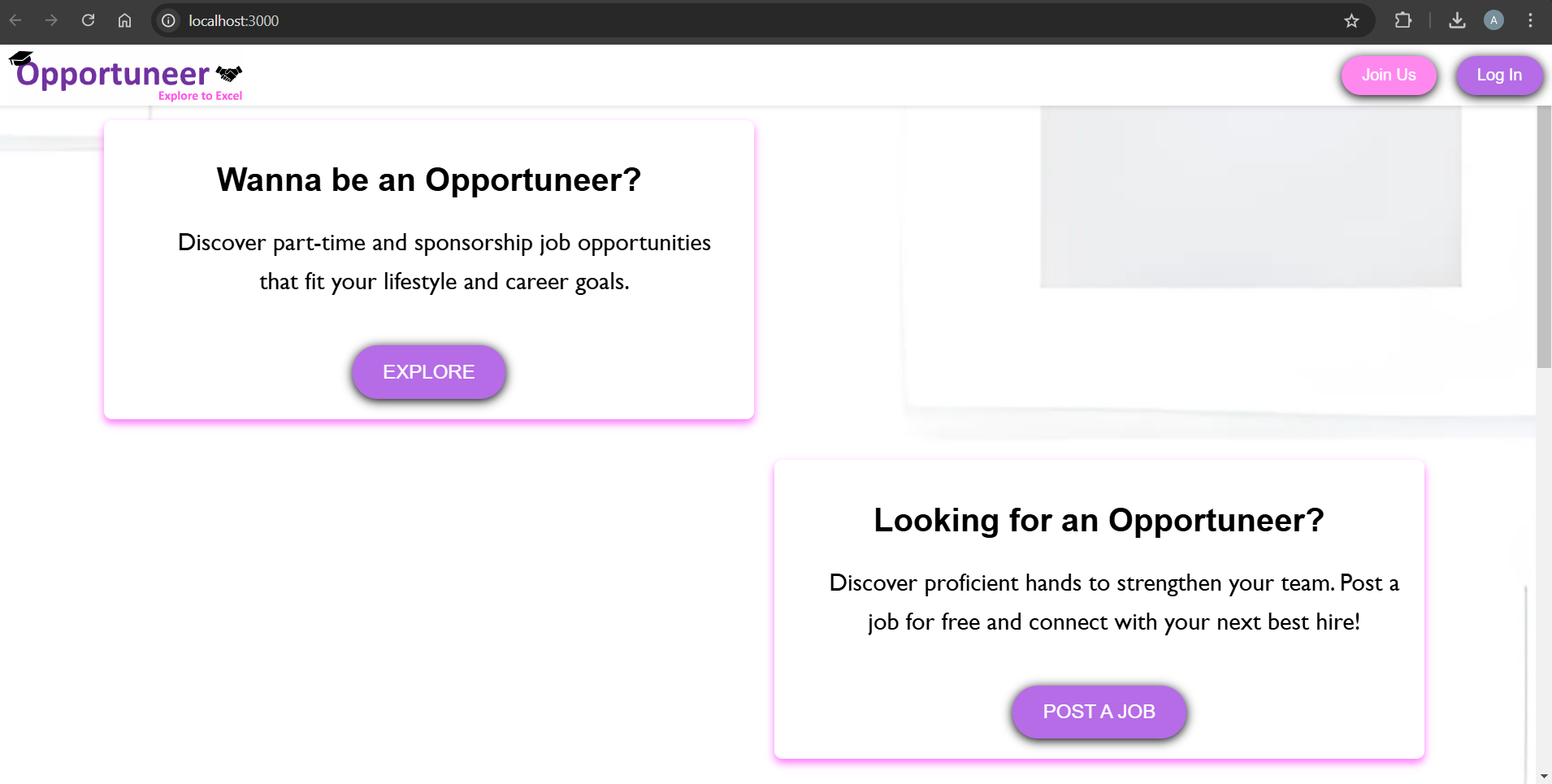
## Metrics and Results

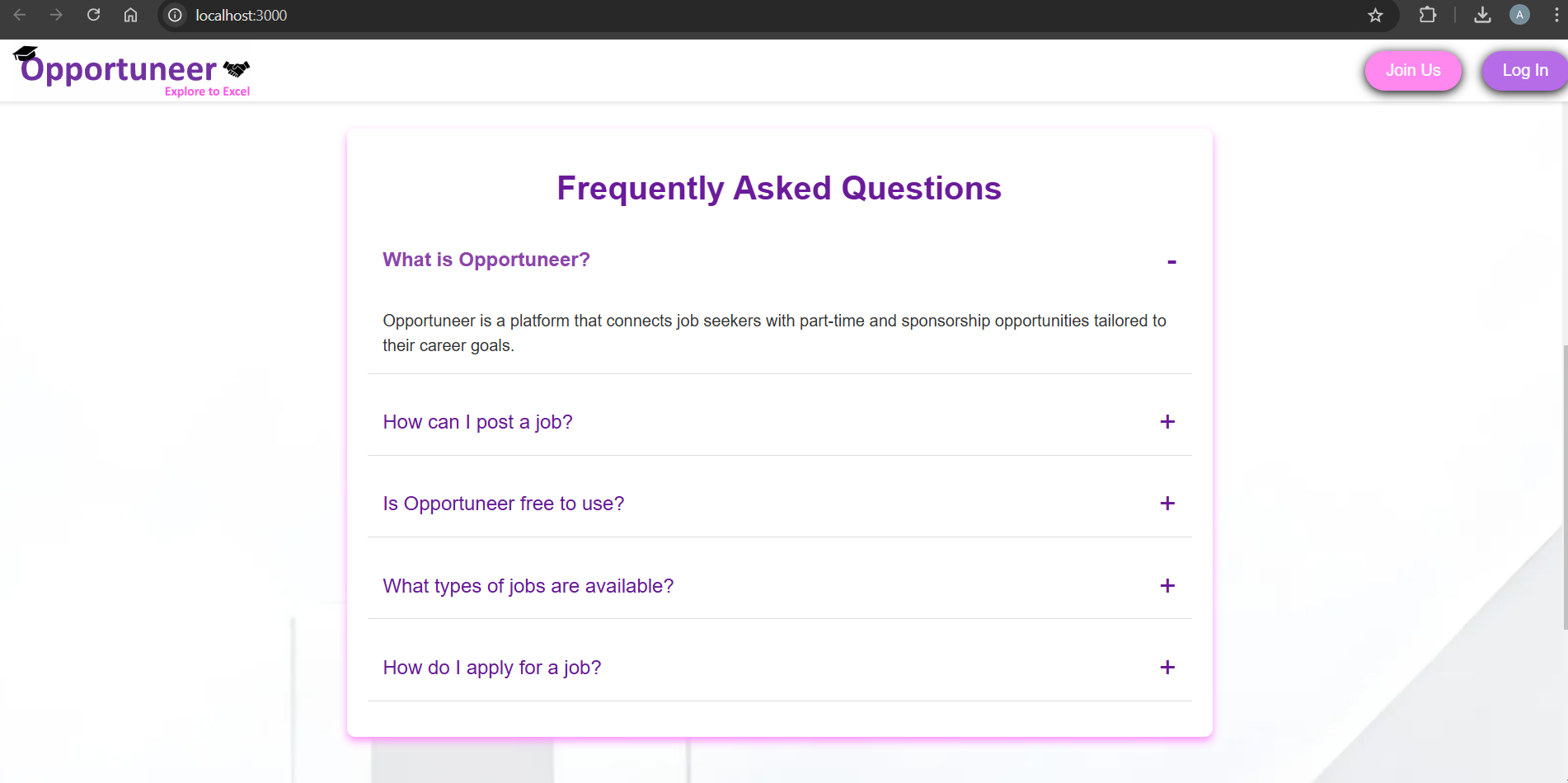
• Task Success Rate: 100% success in job posting and registration functionalities during testing.  
• System Response Time:  
 - Job posting: ~500ms  
 - Job search: ~400ms  
• Database Query Performance:  
 - Optimized queries with average execution time well below 200ms.

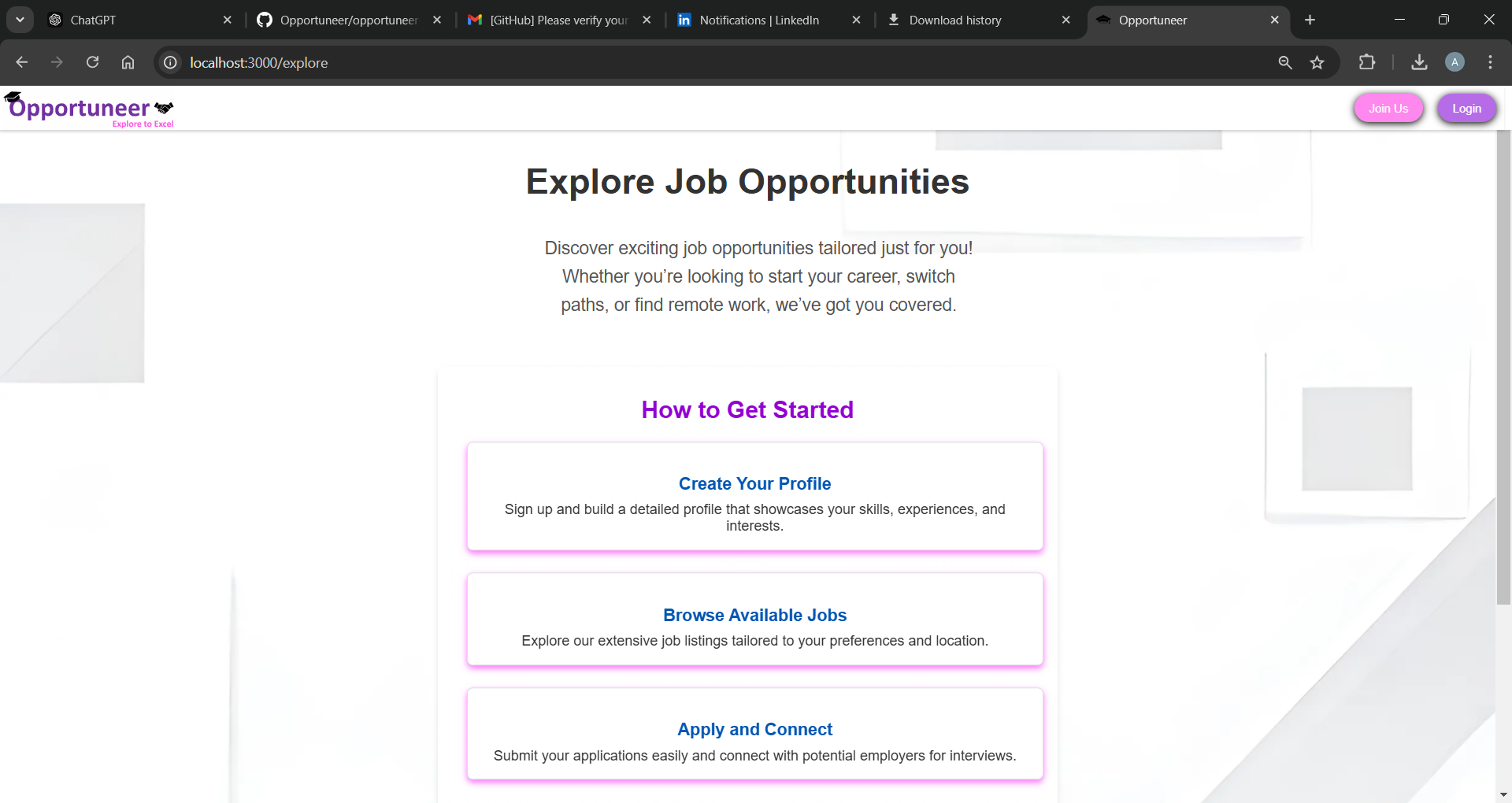
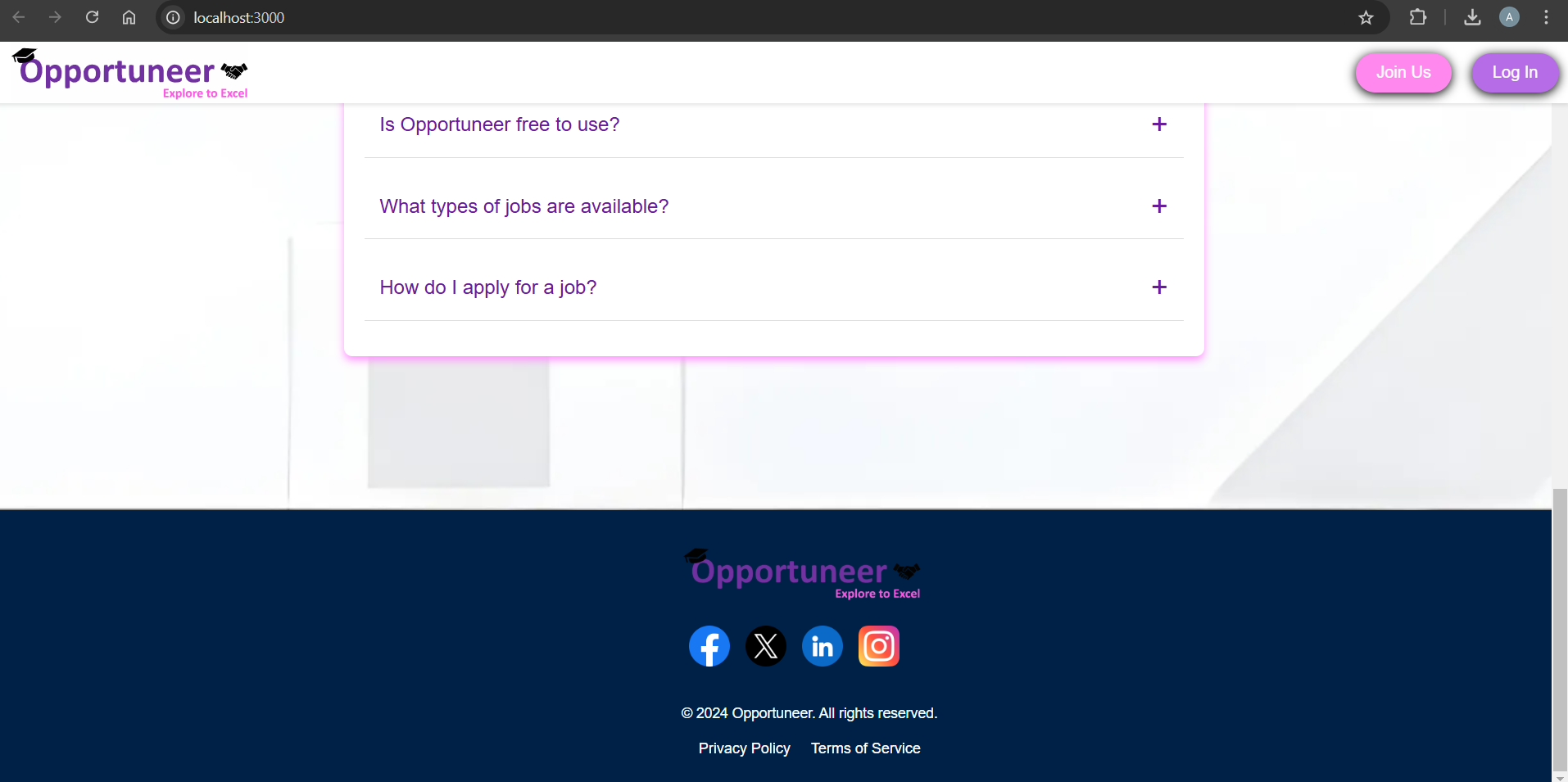
## Future Work

• Deploy the platform to gather real-world feedback.  
• Introduce machine learning for personalized job recommendations.  
• Develop a mobile application for enhanced accessibility.  
• Add advanced security features like two-factor authentication.

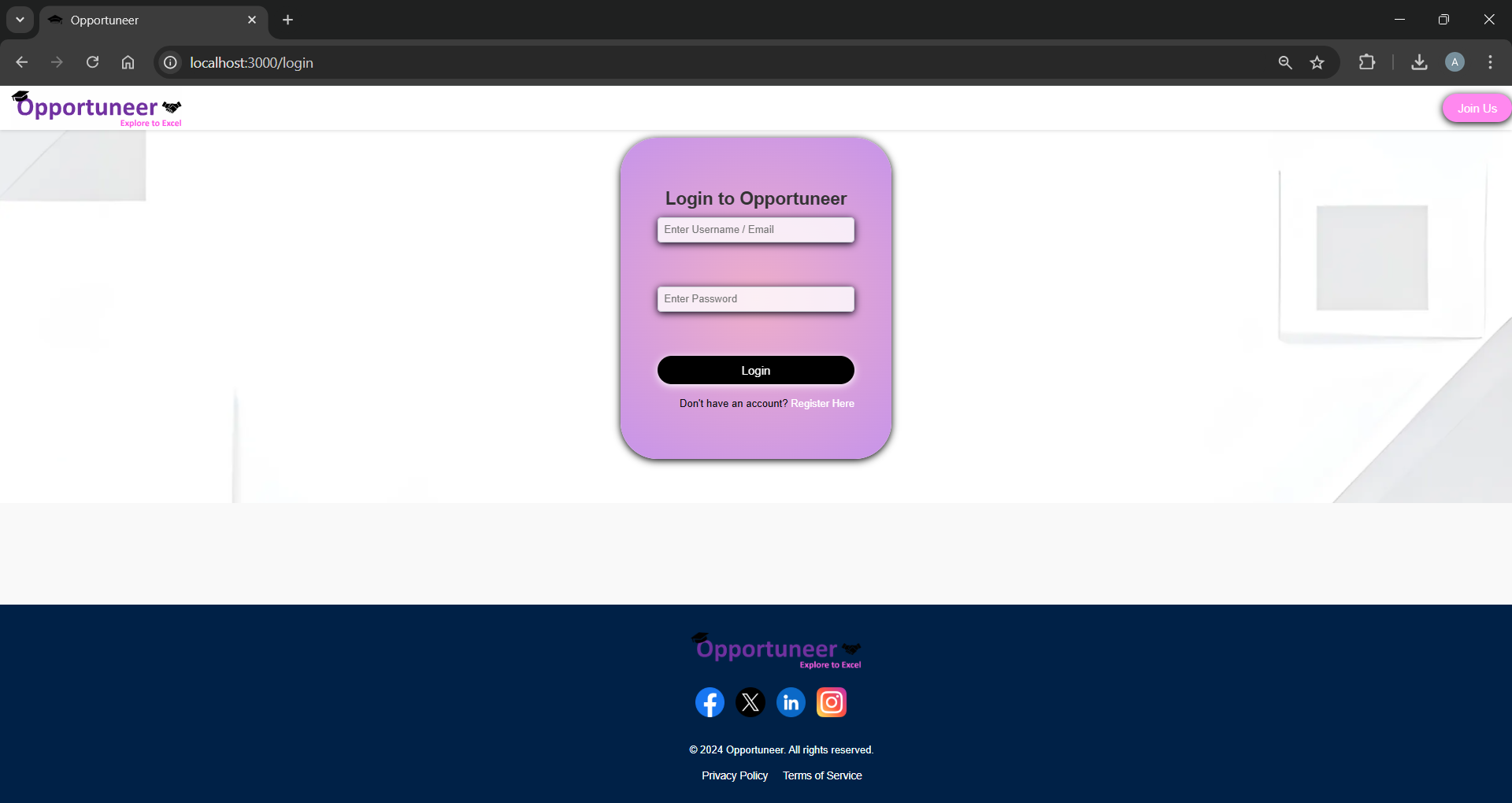
## Screenshots

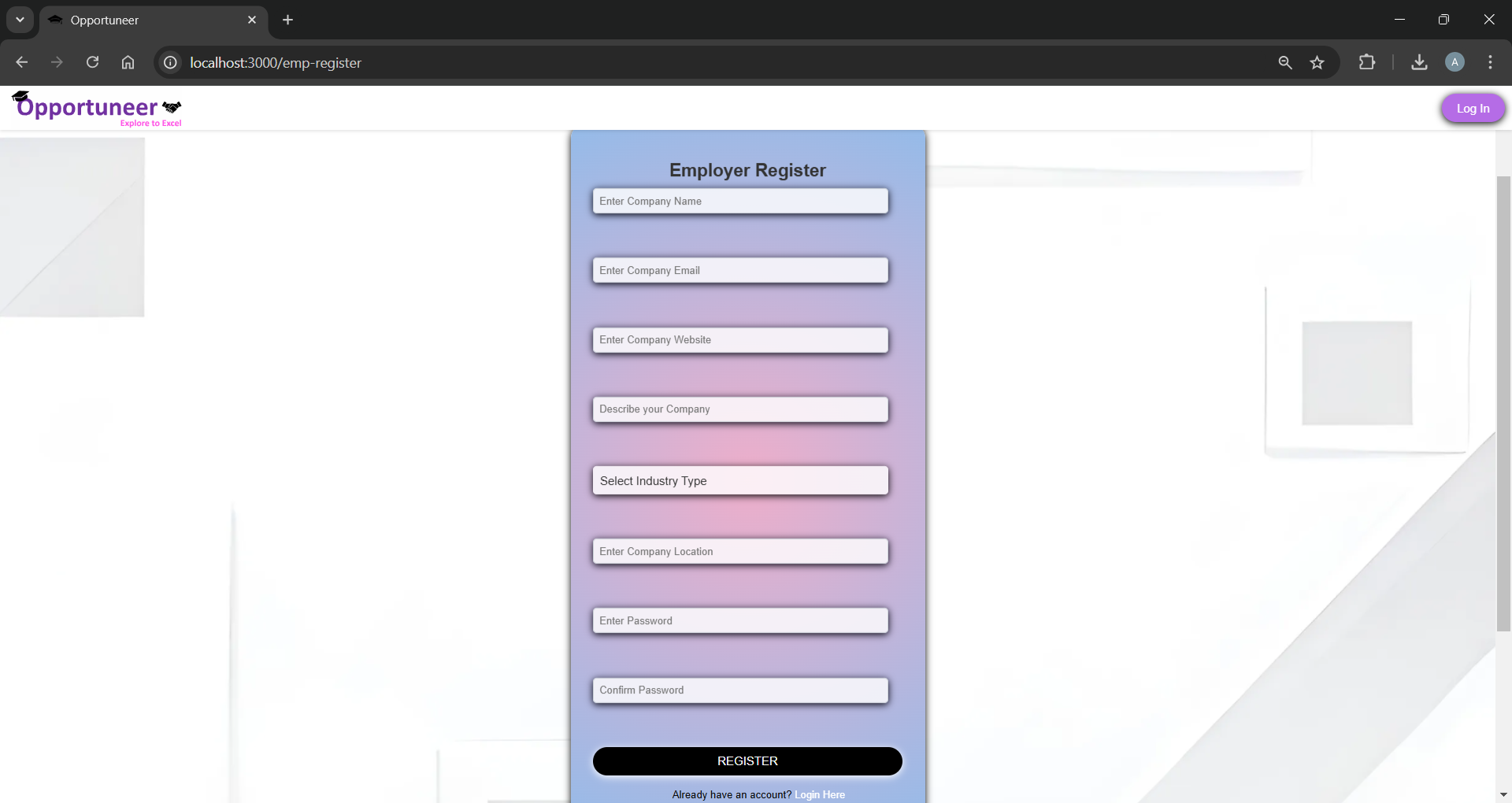


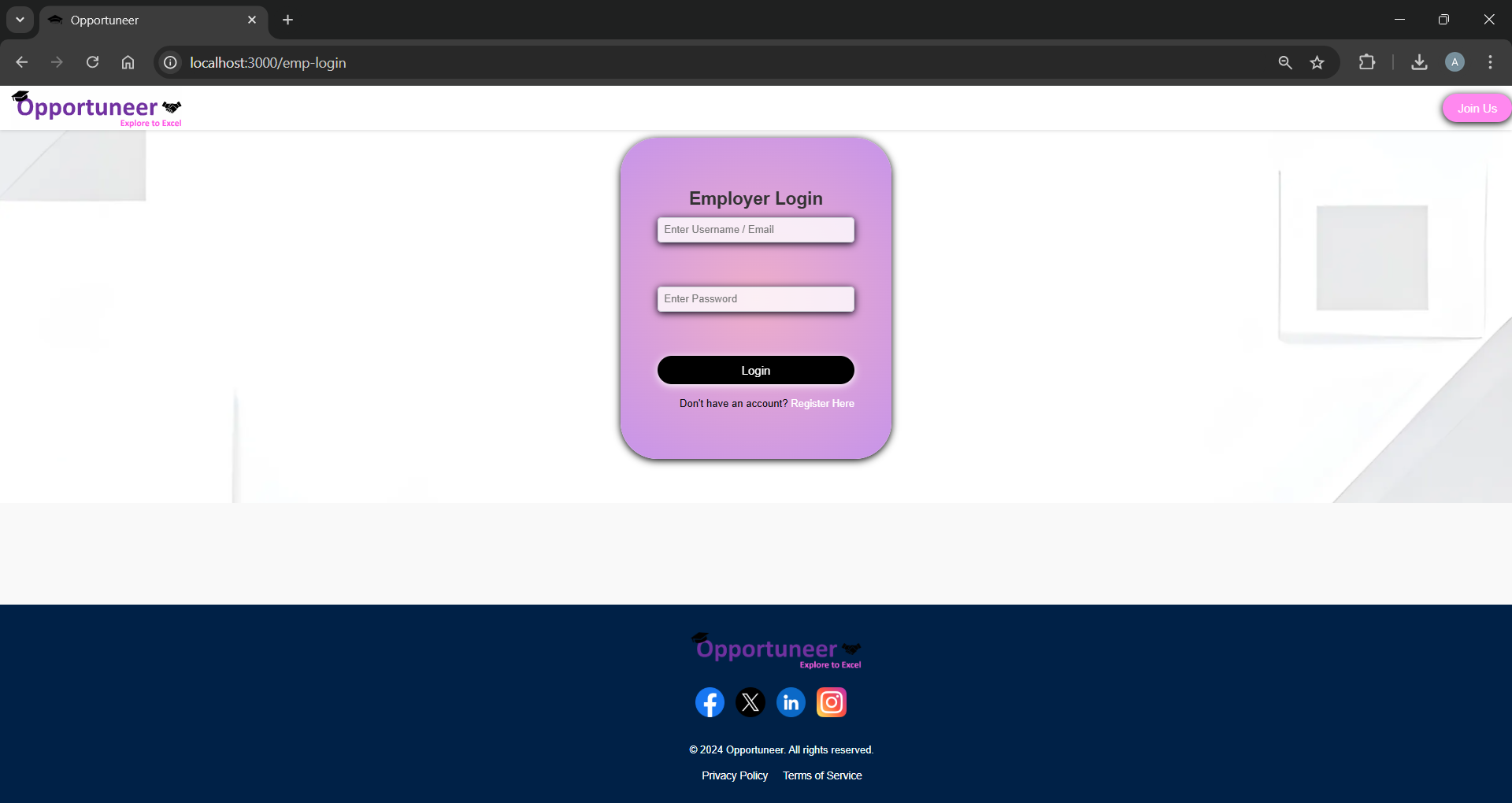


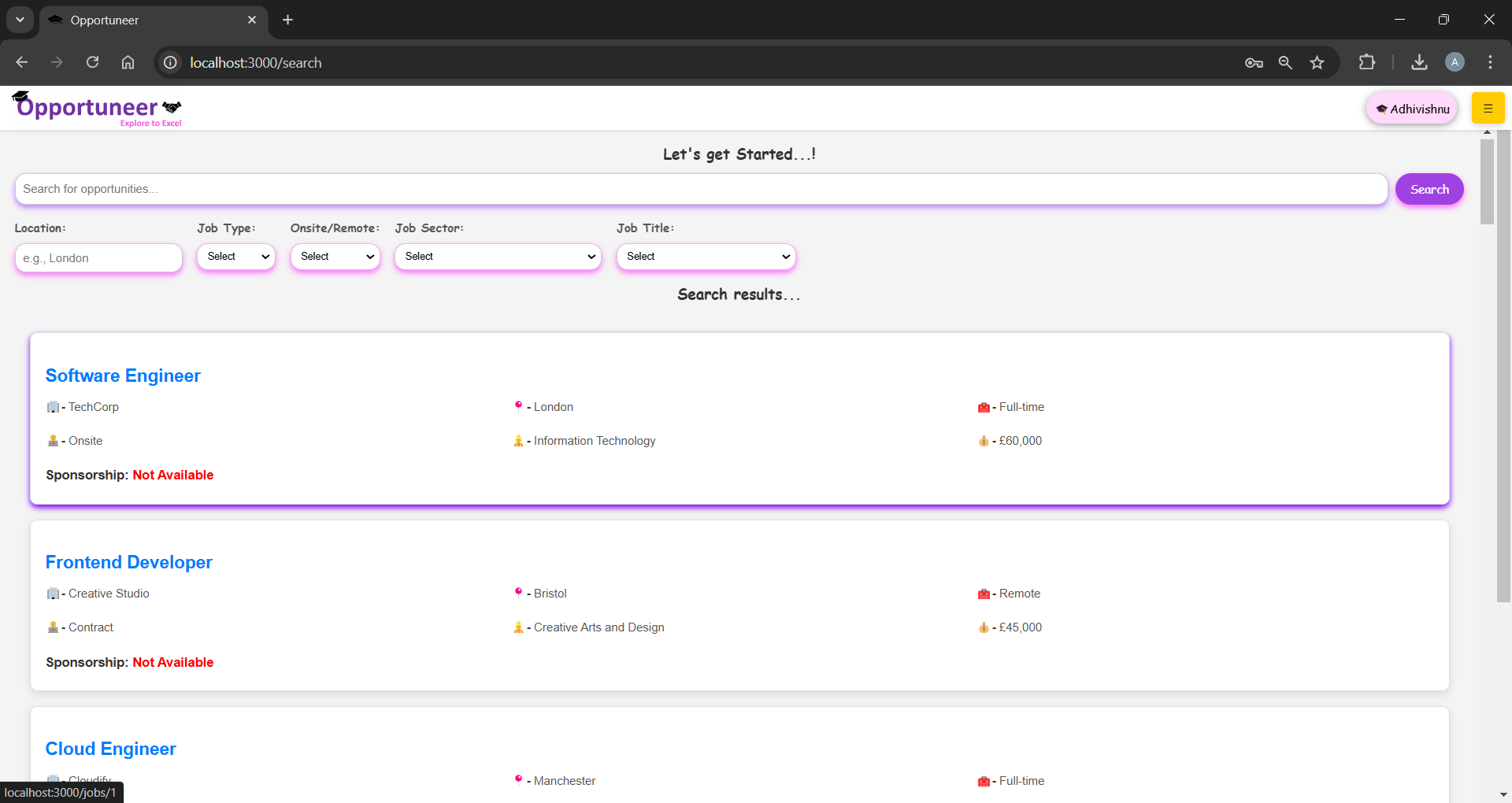


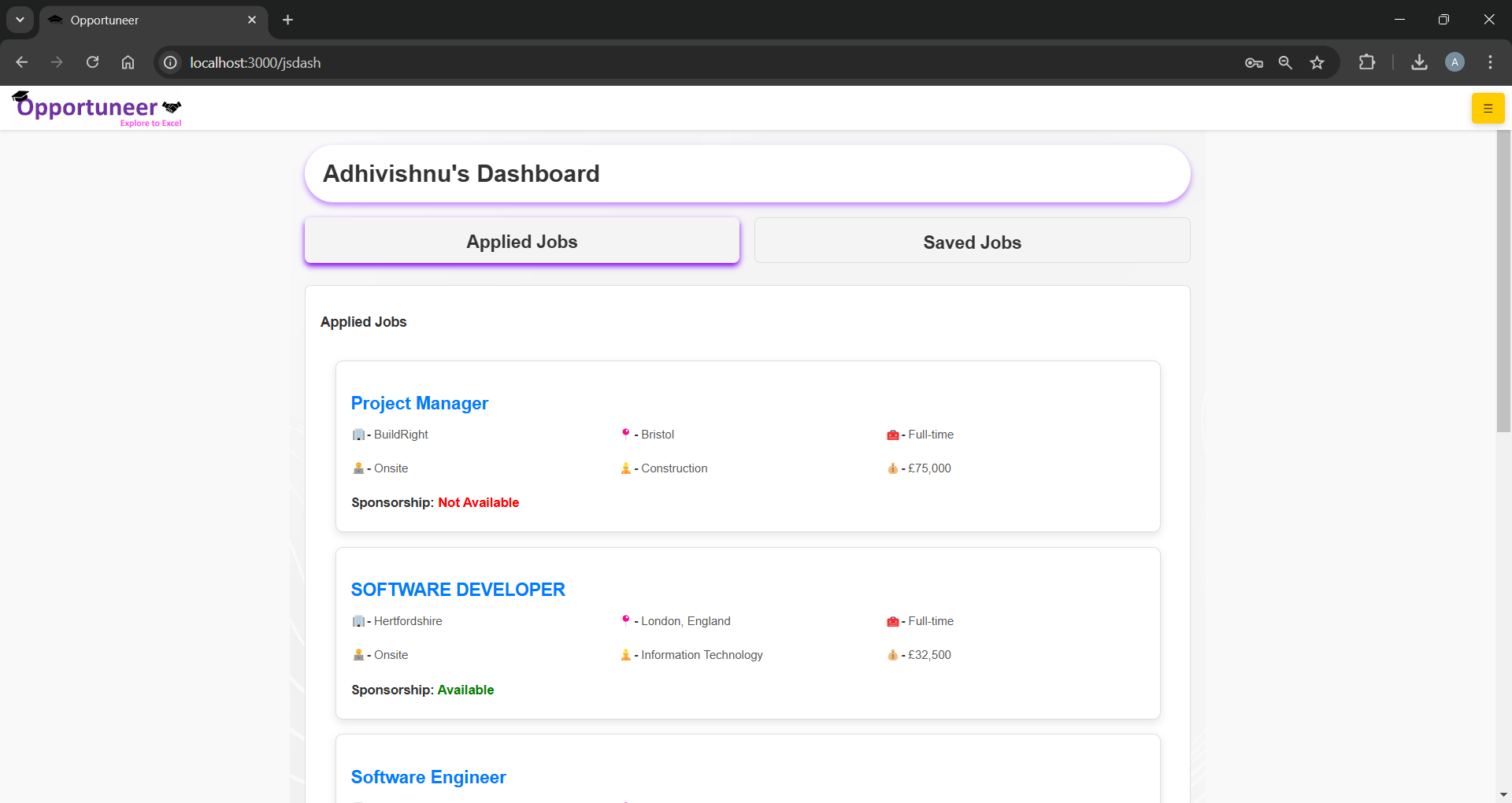
## 





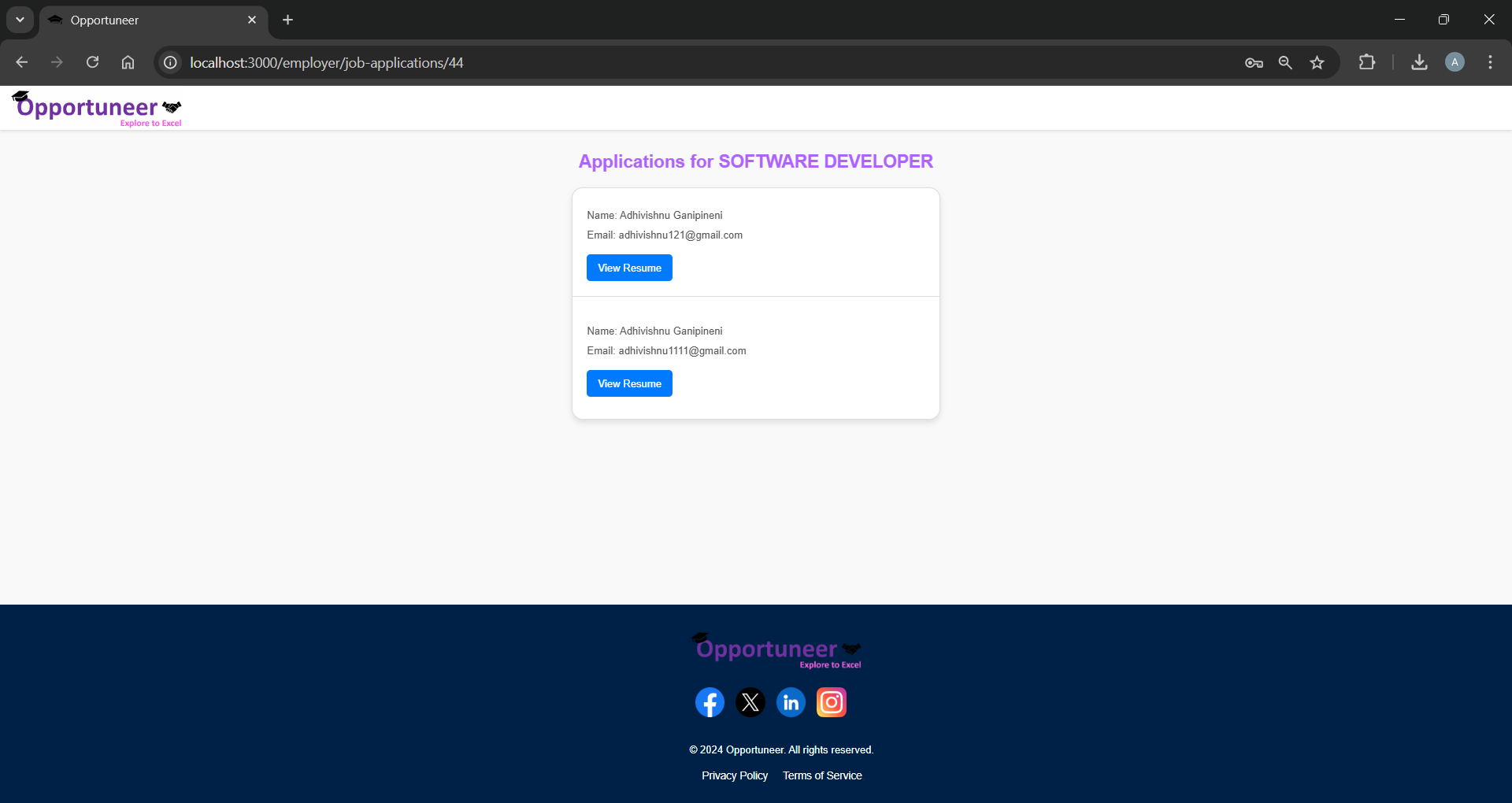




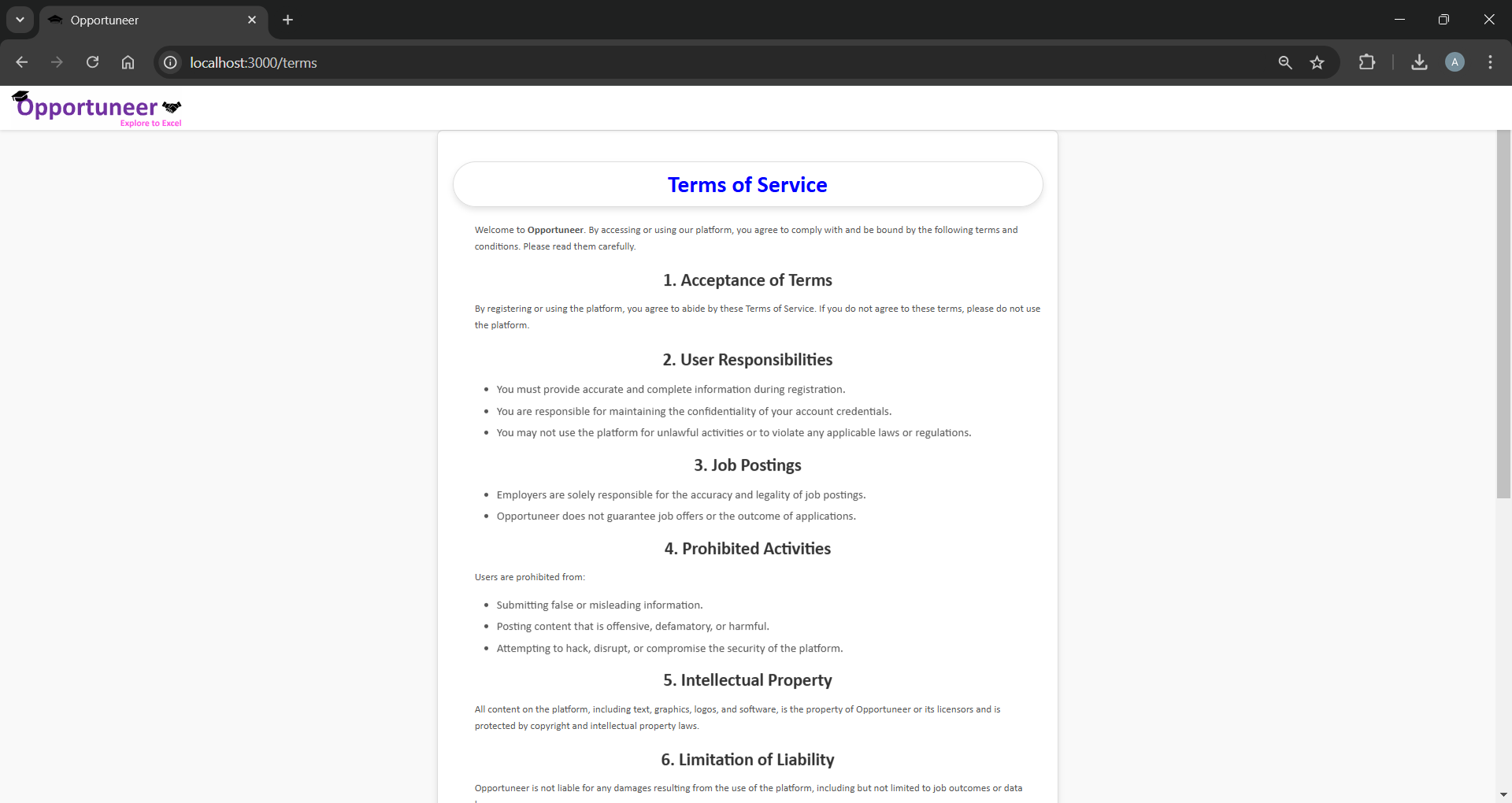


## 

## 



## 



## Contact

For any queries or suggestions, contact:  
• Name: Adhivishnu Ganipineni  
• Email: adhivishnu121@gmail.com  
• LinkedIn: https://www.linkedin.com/in/adhivishnu121/