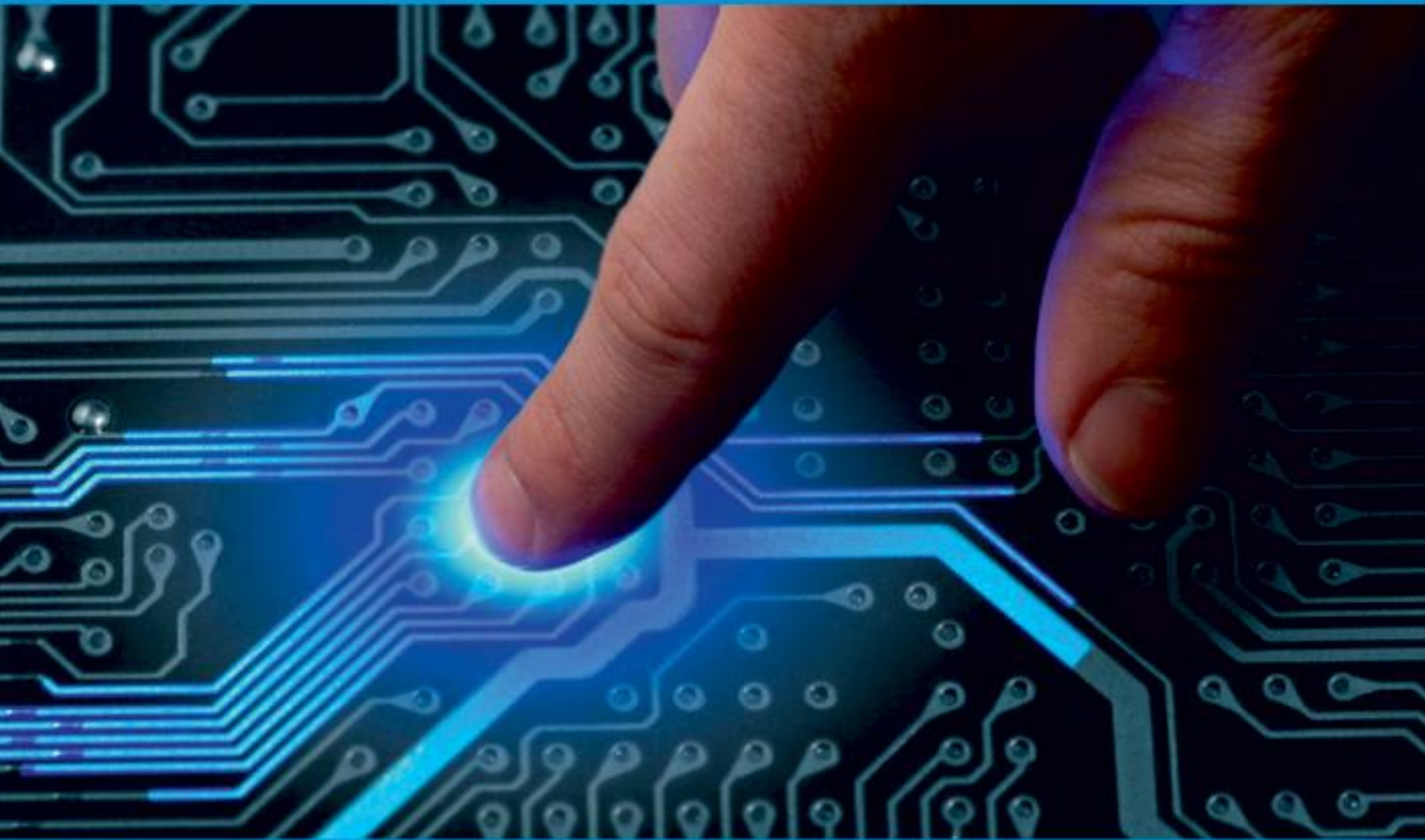
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 5, May 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105022 |

Bus Location Tracking System

Prof. Sushil R. Sankpal[1], Mr. Nilay A. Joshi[2], Mr. Pramod B. Kesarkar[3]

Dept. of E&TC., SGMCOE, Shivaji University, Kolhapur, Maharashtra, India[1]

UG Student, Dept. of E&TC., SGMCOE, Shivaji University, Kolhapur, Maharashtra, India[2]

UG Student, Dept. of E&TC., SGMCOE, Shivaji University, Kolhapur, Maharashtra, India[3]

ABSTRACT: The globe has produced a variety of technologies to improve and simplify people's lives on a daily basis. The newest and fastest-growing technology on the market today is called Android, and it is accessible to all users. In the last few years, there has been a significant rise in end-user adoption. The initiative is built on the most recent GPS technology, providing the bus management team with a better way to monitor bus operation, manage the timetable, and give real-time bus location. The proposed system is a completely integrated online bus tracking system using a database. It provides the facility of tracking the particular bus's location in Google Maps. They can also view the bus details such as the bus schedule and they reach the bus on time. By adding a few minor changes to the hardware and software, this bus location tracking system can also be utilized as an accident detection alert system, soldier tracking system, and many more applications. The method of tracking a vehicle involves keeping track of its position using latitude and longitude (GPS coordinates). The value of a location is its GPS coordinates. This method works quite well for outdoor applications. This type of vehicle monitoring system project is frequently used to track taxis, stolen cars, school buses, and other vehicles.

KEYWORDS: Buskaro; Real-time location; GPS location; Bus management system; Neo6 GPS module; Raspberry Pi

I. INTRODUCTION

The purpose of this project is to develop a web application for tracking bus locations and providing real-time information on bus timings and seat reservations. The web application is using Next.js and utilizes Google Maps for tracking the location of the bus. The project also involves the use of a Raspberry Pi and a NEO6 GPS module to send data to a server, which is then fetched through an API to display the bus location on the website.

The primary objective of this project is to provide a solution to the problem of unreliable public transportation systems. This project aims to make public transportation more efficient and convenient for commuters by providing accurate real-time information on bus timings and locations. The web application is designed to be user-friendly, with a simple interface that allows users to easily track buses and seat reservations online.

To achieve the project goals, the team followed a methodology that involved several steps. The first step was to identify the problem and research existing solutions. The team then decided to use Next.js for building the web application and Google Maps for tracking bus locations. The team also identified the need to use a Raspberry Pi and a NEO6 GPS module to send data to a server and fetch it through an API.

The project was implemented successfully, and the results showed that the web application is effective in tracking bus locations and providing real-time information on bus timings. The project also demonstrated the use of modern technologies such as Next.js, Google Maps, and Raspberry Pi in building a reliable and efficient transportation system.

In conclusion, this project provides a practical solution to the problem of unreliable public transportation systems. By providing real-time information on bus timings and locations, the web application developed in this project can significantly improve the efficiency and convenience of public transportation for commuters.

II. RELATED WORK

1. The proposed system aims to minimize the waiting time for remote bus users by tracking the bus's location using a system that operates round the clock. The system stores all current information on a server, which remote users can access via a web-based application. While web-based systems have been the norm, their use has become somewhat inconvenient for regular bus commuters, who prefer to use more portable Android apps, given the widespread use of smartphones. As such, it is highly recommended to create an Android app version of the system, which will be much more user-friendly and convenient for bus commuters.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105022 |

2. By developing an Android app, bus commuters can obtain real-time bus tracking and information, which can be easily accessed on their smartphones while waiting at the bus stop. In addition, an Android app would enable the system to take advantage of the features available on smartphones, such as push notifications, GPS tracking, and real-time alerts, which would improve the system's efficiency and effectiveness. Ultimately, an Android app version of the proposed system will provide an accessible, user-friendly, and efficient solution to reduce the waiting time for remote bus users.
3. The proffered system uses Artificial intelligence with the help of an RFID module which is exercised in order to reduce the homemade work carried out in the Bus- Management & Monitoring System. In this RFID is exercised to track a machine when it crosses the machine stop. Hence the exact situation of the machine is not shown off, only an approximate situation is shown off predicated on the machine stops. In the moment's world, accuracy is very important and hence this was the restriction of this project.
4. A machine tracker operation to track a machine using a GPS transceiver has been proposed in this paper. The thing of this work is to develop a system that manages and controls the transport using a shadowing device to know the listed vehicle and the current place of the vehicle via SMS using a GPS tracking device.

III. PROPOSED WORK

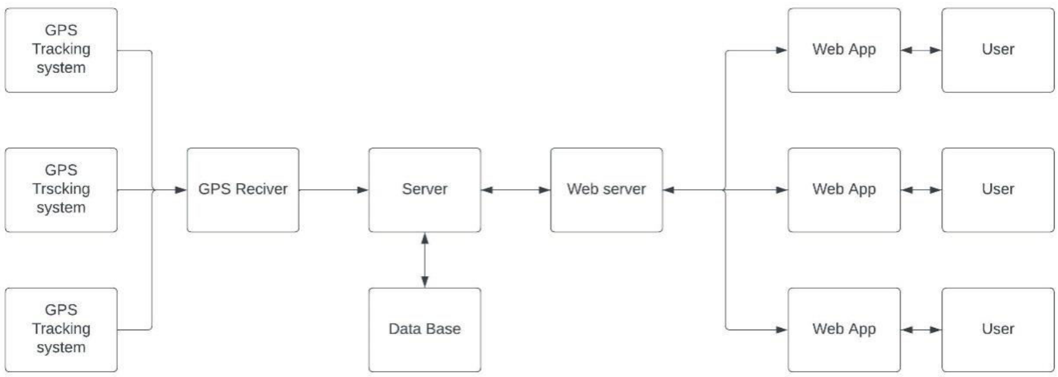


Fig.1. Block Diagram of Proposed Work

HARDWARE PART:

Raspberry Pi: We chose the Raspberry Pi 4 as the main hardware component of the project due to its low cost, small size, and compatibility with various operating systems.
Neo6 GPS module: The GPS module was selected to track the location of the bus in real-time.
Power supply: A 5V/2.5A power supply was used to power the Raspberry Pi and GPS module.
Display screen: A display screen was used to show the bus timings and location on the website.
In addition, we also made sure to follow proper safety procedures while working with the hardware components to avoid any potential damage or injury.

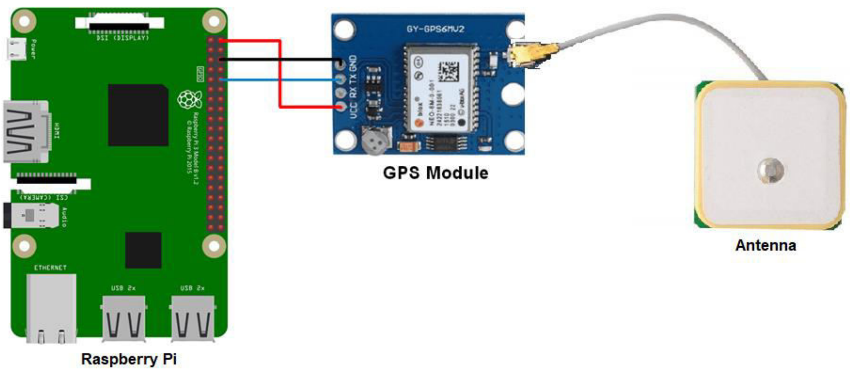


Fig.2. Hardware Requirement

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |
|| Volume 11, Issue 5, May 2023 ||
| DOI: 10.15680/IJIRCCE.2023.1105022 |

SOFTWARE PART:

A. Website

Next.js: We used the Next.js framework to build the web application as it is a fast and efficient framework that allows for server-side rendering and easy deployment.

Node.js: We selected Node.js to develop the server-side application as it is a lightweight, cross-platform, and scalable JavaScript runtime environment that can handle a large volume of data.

Google Maps API: We used the Google Maps API to display the location of the bus on the website as it is a widely used and well-documented API that provides accurate and reliable location data.

Payment gateway: We integrated a payment gateway that was secure, reliable, and easy to use for online ticket booking and payments.

We also considered other options for each component, such as different GPS modules, payment gateways, and display screens. However, after careful consideration of the project requirements and budget, we selected the above-mentioned components for the project.

In addition, we ensured that the selected technologies were up to date and met the project requirements. We also made sure to follow the best practices and guidelines for each technology to ensure the project's success.

B. Android Application

WebView can be added to an Android app by creating a WebView object in the layout file or programmatically in the code. Once added, developers can use methods like loadUrl() to load web pages or loadData() to display HTML content directly.WebView also provides several configuration options to customize its behavior, such as enabling JavaScript, handling URL redirects, and setting custom user-agent strings.

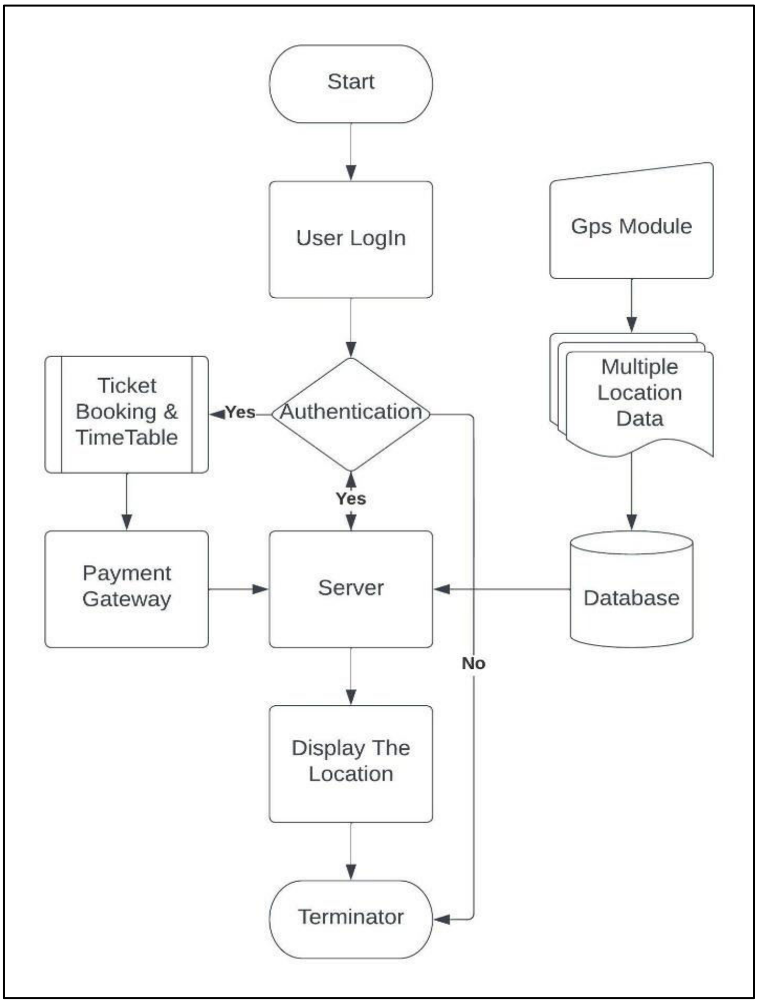


Fig.3. Planned Flow Chart

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105022 |

C. TESTING

We conducted several types of testing to ensure that the web application was error-free, met the project requirements, and was user-friendly. We used various testing frameworks and tools to test the front-end and back-end components of the application.

Unit Testing:

We used Jest and Enzyme to conduct unit testing of the front-end components of the web application. Jest is a popular JavaScript testing framework that is used for testing React components, while Enzyme is a testing utility for React that provides a set of convenient APIs for testing React components' output.Unit testing allowed us to test individual components of the front-end and ensure that they functioned as intended, improving the overall reliability of the application.

Integration Testing:

We used a Supertest to conduct integration testing of the back-end components of the web application. Supertest is a testing library for Node.js that allows us to test APIs' responses and endpoints, simulating HTTP requests and responses.Integration testing allowed us to test the integration of the back-end components of the application and ensure that they functioned together as intended.

User Acceptance Testing:

We conducted user acceptance testing to ensure that the web application was user-friendly and met the needs of the end-users. User acceptance testing involved getting feedback from users on the user interface, functionality, and overall user experience of the application. User acceptance testing allowed us to identify any user interface or functionality issues that were not identified during the development and testing phase.

Overall, testing was a crucial phase in ensuring that the web application was robust, reliable, and user-friendly. By conducting various types of testing, we were able to identify and address any issues or bugs that arose during development, ensuring that the application met the project requirements and specifications.

D. DEPLOYMENTS

After completing the development and testing phases, the next step was to deploy the web application. Deployment involves transferring the web application from the development environment to a production environment where it is accessible to users.

The following are the steps we followed to deploy the web application:

Infrastructure Provisioning:

We used Amazon Web Services (AWS) to provision the necessary infrastructure for our web application. We used Amazon Elastic Compute Cloud (EC2) to create and launch virtual servers to run the web application, Amazon Simple Storage Service (S3) to store static assets such as images and videos, and Amazon Elastic Load Balancer (ELB) to distribute incoming traffic to the web application.

Build and Deployment Pipeline:

We used AWS Code Pipeline to automate the build and deployment process of our web application. Code Pipeline automatically fetched the latest code from our Git repository, built the application, and deployed it to the virtual servers.

Continuous Integration and Continuous Deployment (CI/CD):

We used AWS Code Build to implement continuous integration and continuous deployment (CI/CD) of our web application.

Configuration Management:

We used Ansible to manage the configuration of our virtual servers. Ansible is an open-source automation tool that allows us to define and manage the infrastructure configuration as code. We defined the configuration of our virtual servers using Ansible playbooks and roles, which were automatically executed during the deployment process.

Monitoring and Scaling:

We used AWS CloudWatch to monitor the performance and availability of our web application. CloudWatch provided real-time metrics and log data that allowed us to detect and diagnose any issues that arose during deployment. We also used AWS Auto Scaling to automatically scale up or down the virtual servers based on the incoming traffic to the web application.

Overall, deployment was a crucial phase in ensuring that the web application was accessible and available to users. By using AWS services and tools, we were able to automate and simplify the deployment process, reducing the risk of errors and ensuring the web application was running efficiently and reliably.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105022 |

IV. Working of Project

1. The project involves creating a web app that allows users to track buses on a map, view bus timings, and book tickets online with payment. The system uses a Raspberry Pi and a NEO-6 GPS module to send bus location data to a server. The web app fetches the location data from the server using an API and displays it on a Google Map.
2. To start with, the Raspberry Pi is connected to the NEO-6 GPS module via wires. The GPS module reads the location data and sends it to the Raspberry Pi in the form of serial data. A Python script is used to read the serial data and send it to the server using the Firebase Realtime Database.
3. The server stores the location data in the database and provides an API for the web app to fetch the data. The web app is built using Next.js, a framework for building server-side rendered React applications. It uses the Google Maps API to display the bus locations on a map and displays the bus timings using data from the server.
4. The web app also has a feature for booking tickets online with payment. Users can create an account and log in using JWT authentication. The web app uses MongoDB to store user data such as name, email, and booking history. The payment system is implemented using Firebase's Payment Gateway, which allows users to pay for their tickets securely online.

In summary, the project involves the following steps:

- Gathering the required hardware and software resources
- Selecting appropriate technologies for the project
- Designing and developing the web app with features such as bus tracking, bus timings, and seat reservations.
- Testing the web app to ensure its functionality and performance
- Deploying the web app on a server for public access
- Maintaining the web app to fix bugs and update features as needed.

V. RESULTS

A. Implementation

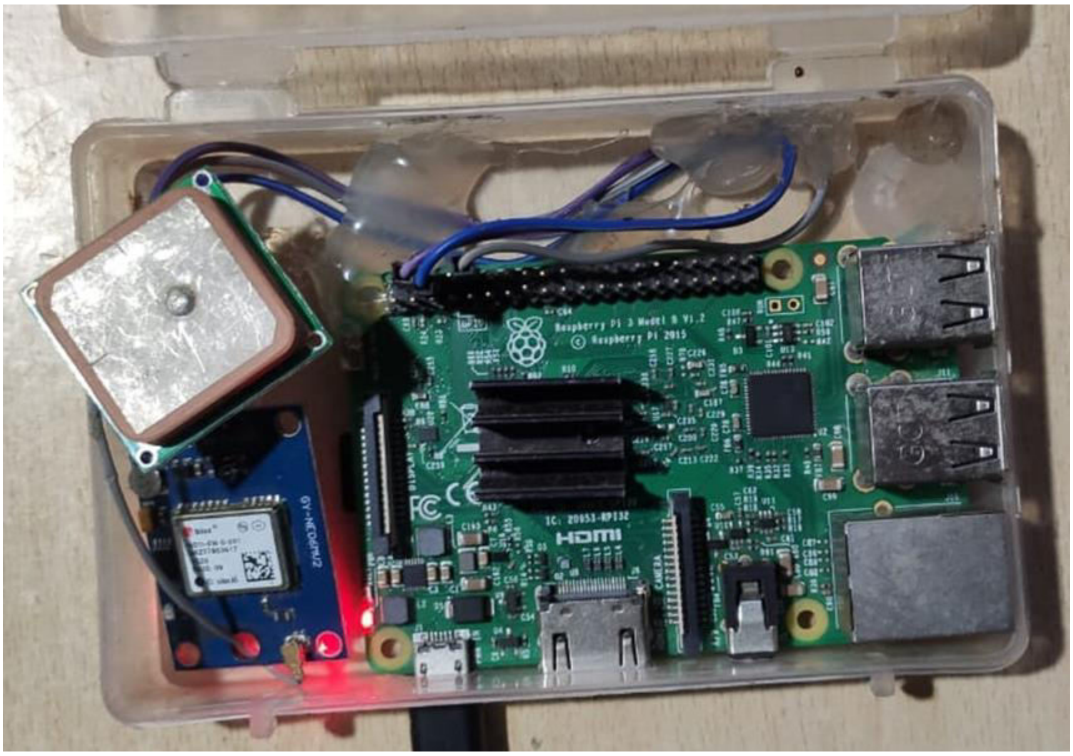


Fig.4. Hardware Implementation

B. WebView

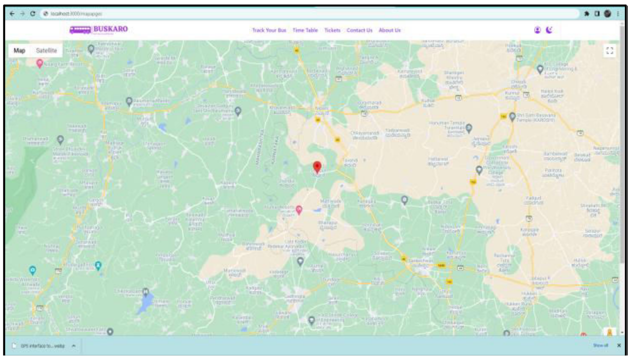


Fig.5.Implemented System Map View

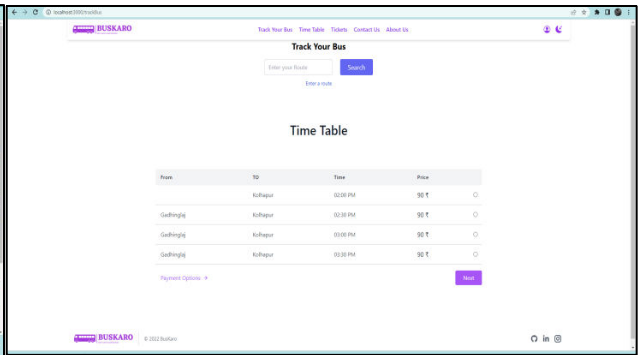


Fig.6.Timetable

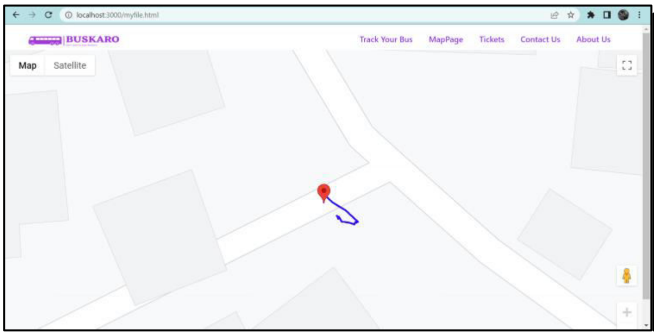


Fig.7. Tracking

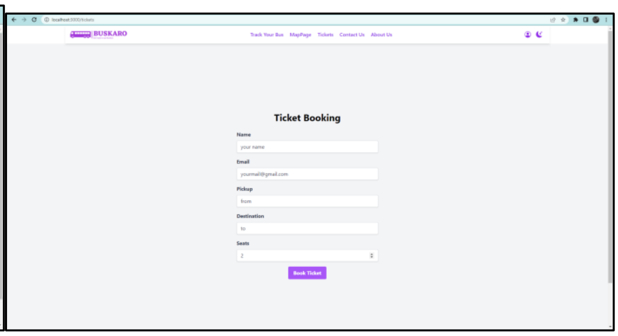


Fig.8. Ticket

C. Android Application View

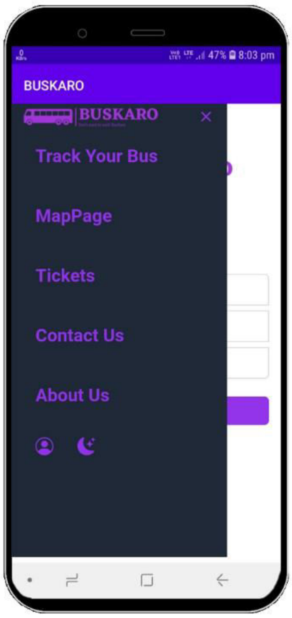


Fig.9. Main Menu

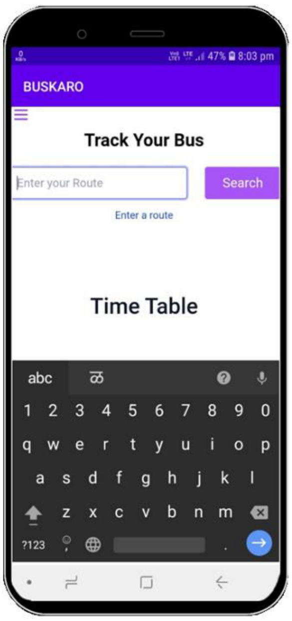


Fig.10. Tracking

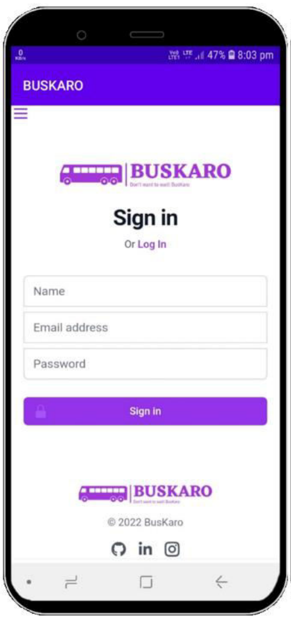


Fig.11. Account Signup

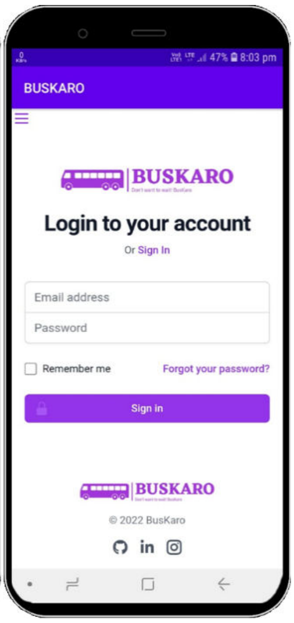


Fig.12. AccountLog in

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105022 |

VI. CONCLUSION AND FUTURE WORK

The use of GPS-based location tracking and API calls to transmit data from a Raspberry Pi to a centralized server has proved to be effective for the development of the system. The system's user-friendly interface and cross-device compatibility make it easily accessible to a wide range of users.The project has contributed to the field of smart transportation systems by demonstrating the potential of data-driven technologies in addressing real-world problems and improving the quality of life for people. The project has also highlighted the limitations of GPS-based location tracking and centralized systems, which may require further research and exploration.

REFERENCES

1. Shubham Jain; Adarsh Trivedi; Shweta Sharma, “Application-Based Bus Tracking System” Published in 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India.
2. Judy Thyparampil Raj; Jairam Sankar, “IoT based smart school bus monitoring and notification system”, Published in: 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)
3. Andrew J. Kotz; Eric Miller; Andrea Watson; Kenneth J. Kelly, “Transit Bus Electrification Evaluation from GPS Speed Traces”, Published in: 2020 IEEE Transportation Electrification Conference & Expo (ITEC)
4. Mohd Hakimi Bin Zohari, Mohd Fiqri Bin Mohd Nazri, “GPS Based Vehicle Tracking System”, Published in: 2021 International Journal of Scientific & Technology Research Volume 10
5. Ms.A.Deebika Shree, Ms.J.Anusuya, Dr.S.Malathy, “Real Time Bus Tracking And LocationUpdation System”, Published in:2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)
6. Pratap Misra, Per Enge, “Global Positioning System” Signals, Measurements, and Performance, Second Edition, Ganga-Jamuna Press.

INNO SPACE

SJIF Scientific Journal Impact Factor

Impact Factor: 8.379

doi crossref

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

Scan to save the contact details