Ollscoil Teicneolaíochta an Atlantaigh

Atlantic Technological University

TrafficVision

Rohan Sikder and Conor Murphy.

GitHub - https://github.com/ConorPadraigMurphy/FYP

Introduction

TrafficVision, a new web application developed to show bus system reliability and punctuality for public transport. Consistent bus schedules result in issues for commuters, the infrastructure and the environment of cities in urbanized areas where efficient transportation is a core element of everyday life. TrafficVision is built for these challenges with computer vision and web technologies, providing real-time information about bus service schedules and quality.

Urban mobility relies on public transportation, particularly bus systems. Their effective operation impacts the environmental sustainability and the daily life of the population. However, bus services often have delays and unpredictability that inconvenience passengers and lead to higher carbon emissions and congestion. Such inefficiencies hinder sustainable urban living. TrafficVision hopes to ease these issues by offering real time bus info so commuters, transportation authorities and urban planners can make the best choice possible.

Objectives

The primary **objectives** of TrafficVision are:

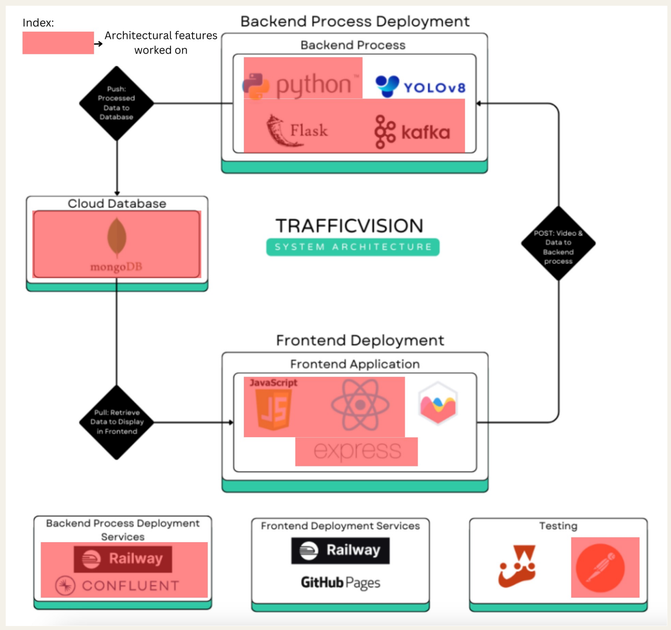
• To develop a web-based platform that offers real-time tracking and analysis of bus services.

• To enhance the punctuality and reliability of public transportation through data-driven insights.

• To improve the daily commuting experience by minimizing uncertainty and wait times for bus passengers.

• To assist in reducing traffic congestion and carbon footprint through efficient bus service management.

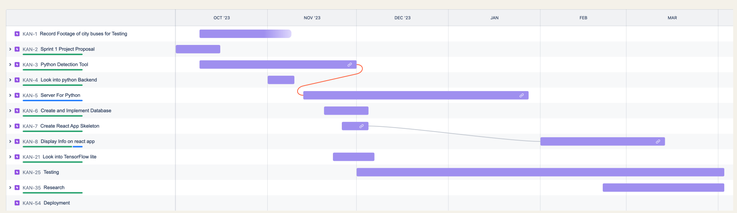
System Architecture



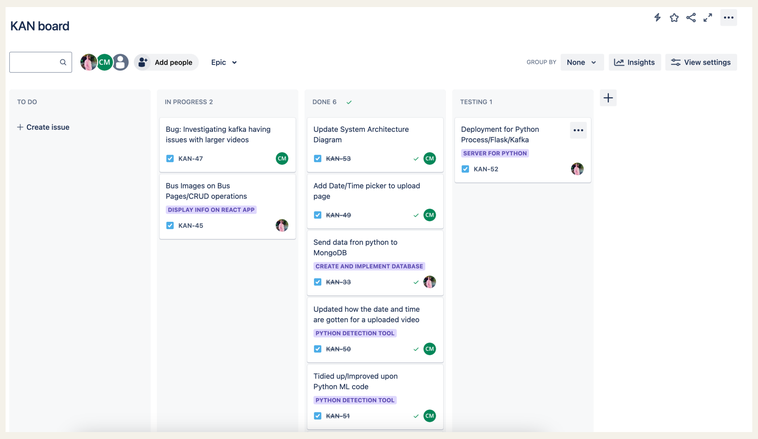
The system architecture of TrafficVision is devised to be modular, scalable, and responsive. By employing a serverless processing server, TrafficVision ensures efficient utilization of computing resources. Kafka's data streaming capabilities enable robust real-time processing and communication between the system's components.

MongoDB offers a scalable and flexible data storage solution to accommodate the diversity and volume of data managed by TrafficVision. Lastly, the React frontend provides an accessible and user-friendly interface for TrafficVision's diverse user base.

Agile Methodology

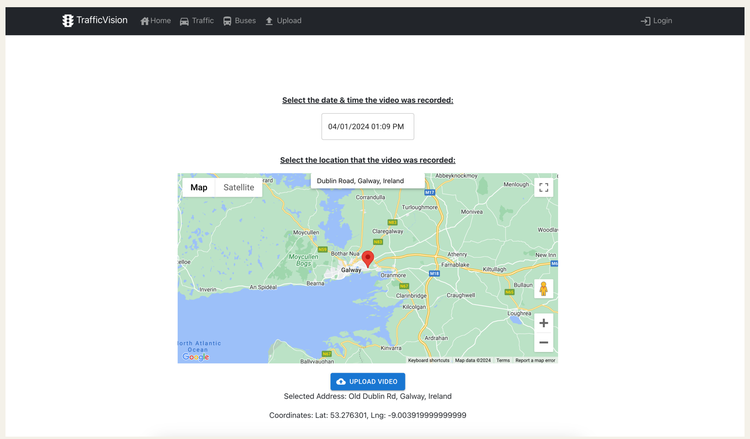


To complement the agile project management approach and enhance visibility into the project timeline, a Gantt chart was employed. This tool was essential in planning, coordinating, and tracking specific tasks against time.
The Gantt chart for TrafficVision, available at *https://rohansikder4.atlassian.net/jira/software/projects/KAN/boards/1/timeline*



The workflow was effectively managed and visualized using Kanban boards in the project. From initial development phases to final testing, each task was represented by a card on the Kanban board in columns labelled To Do," In Progress, "Testing," and Done."

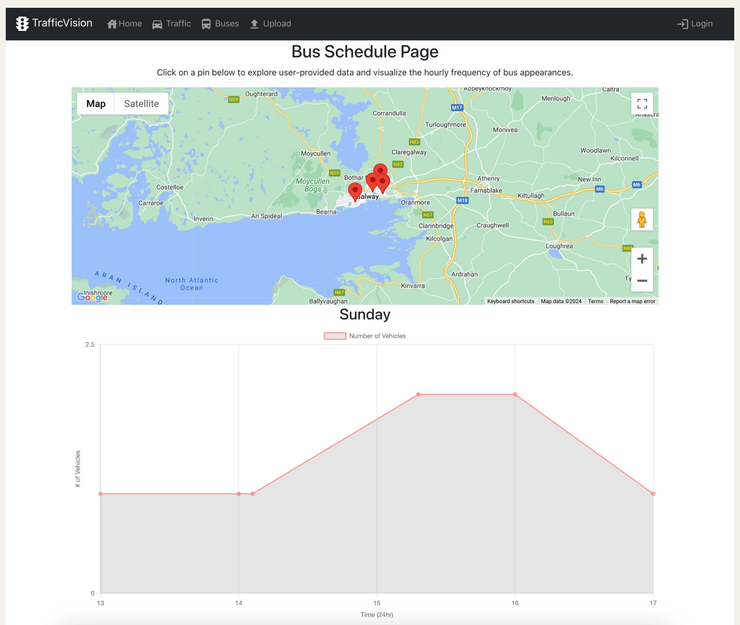
Application Preview



End-to-End Encryption with Node.js: Utilizes Node.js for both encryption and authentication, ensuring comprehensive security measures without relying on third-party services.

Secure Address Selection and Geo-tagging: Employs Google Maps API within React to allow users to select the location where the video was recorded, integrating geocoding functionality for precise address retrieval and geotagging of video uploads.

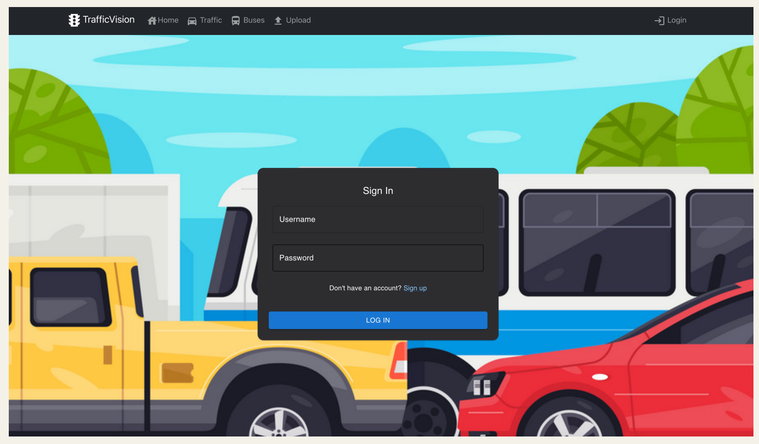
Efficient Video Upload Handling: Implements efficient video upload handling through Axios, enabling seamless transfer of video files along with accompanying metadata such as location coordinates and recording date-time stamps directly to the server for processing and storage.



Dynamic Data Visualization: Utilizes Chart.js library to dynamically visualize hourly frequency of bus appearances based on user-provided data, allowing users to explore and understand bus schedules effectively.

Interactive Map Display: Integrates Google Maps API within React to display pins representing bus locations, enabling users to click on pins and view detailed data for specific locations, enhancing user engagement and interactivity.

Efficient Data Filtering and Grouping: Implements efficient data filtering and grouping functionalities to organize bus data by location and day, enabling users to analyze bus schedules by location and time, facilitating better decision-making and planning.



Node.js Encryption Implementation: Utilizes Node.js for login page encryption, ensuring data security without relying on third-party services.

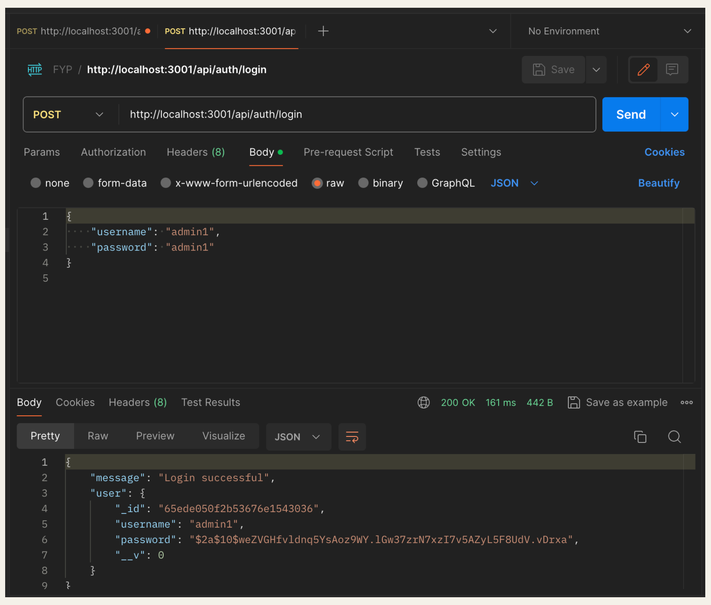
bcrypt Integration for Secure Authentication: Implements bcrypt within Node.js to convert passwords into highly secure hashes, enhancing login page security and integrity.

Independent Encryption Workflow: Ensures encryption and authentication processes are conducted solely within the TrafficVision system, eliminating reliance on external services for enhanced control and security.

Complete Control with No Third-Party Dependencies: By leveraging Node.js for encryption and authentication, TrafficVision maintains full control over user data protection without relying on external services, enhancing trust and security for users.

Backend Preview

Testing



Conclusion

TrafficVision was a success in achieving its initial goals due to its real-time traffic status reports and bus schedule compliance. This critical feature allows commuters and urban planners to make informed decisions that significantly improved commuting efficiency and traffic control. However, TrafficVision currently has a limit on the amount of crowdsourced data it processes. With more data collection the system will become more precise and useful in terms of deep insights and more accurate traffic forecasts.

Acknowledgements

I would like to express my deepest gratitude to Dr Brian McGinley whose expertise and guidance were invaluable during all stages of this project and Special thanks to my team member, Conor Murphy for their hard work and dedication.