# 

# ParkMe | MU Interim Report

Student: Redon Ferizi (21358591) ([Redon.ferizi.2022@mumail.ie](mailto:Redon.ferizi.2022@mumail.ie))

Supervisor: Dr Aidan Mooney

# Overall Project Objectives

The primary objective of this project is to develop “ParkMe | MU,” a comprehensive web application designed to streamline parking management at Maynooth University. The application aims to provide real-time parking availability across campus, facilitate easy check-ins and check-outs for users, and offer navigation assistance to available parking spots. Additionally, it seeks to deliver analytics and insights to help users plan their commutes more effectively, enhance the overall parking experience for students, staff, and visitors, and reduce parking congestion to improve the efficiency of campus parking.

# Description of Work Completed

Substantial progress has been achieved during the initial phases of the project, adhering to the established schedule and objectives. Key milestones include the establishment of the overall framework, the setup of both frontend and backend infrastructure, and the development of core features that enable real-time functionality and user interactivity.

# Evidence of Work Completed and Issues Encountered

## Requirements Gathering:

A comprehensive requirements document detailing both functional and non-functional requirements was created to guide development. However, initially capturing all necessary requirements proved challenging due to the complexity of real-time parking data. Gathering input from potential users and stakeholders took time, and adjusting the requirements to meet practical expectations while maintaining a realistic scope was difficult.

## System Design:

System diagrams outlining the architecture of frontend and backend components were developed, along with user interface wireframes in Figma to illustrate layout and design (see Appendix B). Designing a seamless user interface that balanced functionality with simplicity was challenging. Translating wireframes into interactive features required numerous revisions to avoid clutter and ensure a user-friendly layout. Coordinating frontend and backend integration was complex, necessitating multiple design iterations to ensure smooth data flow across the system.

## Frontend Development:

The user interface was built using HTML5, CSS3, and JavaScript, with responsive design implemented using Bootstrap 5 to ensure compatibility across various devices. The codebase comprises approximately 4,500 lines across HTML, CSS, and JavaScript files (see Appendix A). Integrating Bootstrap effectively while maintaining unique design elements presented styling challenges. Ensuring consistency across different screen sizes was difficult, and managing CSS conflicts, especially with the interactive map, required extensive debugging to maintain the layout.

## Backend Development:

A server was set up using Node.js and Express for efficient request handling, and RESTful APIs were developed to manage parking data retrieval, check-ins, check-outs, and real-time updates (see Appendix B). Configuring Express routing and managing API requests effectively, particularly for real-time data updates, was initially problematic. MongoDB connection errors and port configuration issues also slowed development, as debugging these errors required familiarity with server environments and fine-tuning configurations for stability.

## Database Setup:

A MongoDB database was configured to store user data, parking spot information, and log management. Schemas for users, parking spots, and transactions were designed to standardise data handling across the system (see Appendix C). Designing efficient schemas was initially challenging, particularly in balancing flexibility with database efficiency. Implementing relationships between data types, such as users and parking spots, added complexity to the database structure, leading to performance issues when querying large datasets. These issues were addressed by optimising queries and fine-tuning schema configurations.

## Feature Implementation:

Key features implemented include real-time availability, check-in/check-out systems, an interactive map, navigation assistance, and a user feedback system. Real-time availability was integrated using Chart.js for dynamic data visualisation, providing real-time insights. The check-in/check-out system updates parking availability based on user input, while Leaflet was utilised to allow users to view real-time parking availability on a campus map. Navigation assistance was integrated using the Google Maps API to guide users to available parking spots. A feedback form was added to gather user feedback and drive improvements.

# Literature Review

## Background and Motivation:

Inspired by the challenges of finding parking at Maynooth University, the project aims to improve parking management for students, staff, and visitors.

## Prior Work in Web Development and Parking Systems:

Experience gained during an internship at Electric Ireland, where expertise in CMS and web systems was developed, is directly applicable to this project. Additionally, academic studies in web development modules provided essential skills in HTML, CSS, JavaScript, and modern frameworks.

## Technical Resources and Inspirations:

The project utilises several state-of-the-art technologies and practices. Mapping systems such as Leaflet and the Google Maps API are used for dynamic maps and navigation. Chart.js is employed for real-time data representation, while Node.js, Express, and MongoDB provide a scalable backend for real-time data management. The application is designed for inclusivity with colourblind-friendly features and an intuitive user interface.

# Overview of Learning Gained Through GenAI:

Generative AI tools like ChatGPT and GitHub Copilot have been invaluable throughout the development of the ParkMe | MU project. These tools provided insights into industry-standard best practices for both frontend and backend development, facilitated the exploration and integration of new technologies and APIs relevant to the project’s requirements, and enhanced UI/UX design skills by offering suggestions on layout, colour schemes, and responsive design techniques.

# Future Work

The next phases of the project will focus on completing feature implementation, including push notifications, a “where is my car” feature, and the ability to plan parking before arrival. Additionally, the application will incorporate cookie functionality to enhance user experience. Thorough unit, integration, and user testing will be conducted to identify and address any bugs or performance issues. Preparation for deployment will involve optimising the application for a scalable hosting platform such as Heroku or AWS and ensuring that security best practices are in place. Comprehensive documentation, including user manuals, API documentation, and technical guides, will be developed. The final presentation and submission will include preparation of presentation materials, such as slides and a live demo.

One of my goals, should I complete a significant portion of the work ahead of schedule, is to transform the app into a CMS-style platform. This would allow different users to customise the app based on their specific needs, such as creating ParkMe Limerick or a food location app. Additionally, I aim to establish contact with the university to promote the app and gain insights for further feature development from them.

# Appendix - Additional Evidence

## **Appendix A: Screenshots**

**Figure 0:** *landing page*

A screenshot of a parking lot

Description automatically generated

**Figure 1:** *Features under landing*

A screenshot of a screen

Description automatically generated

**Figure 2:** *Occupancy Graph under features*

A screenshot of a parking permit

Description automatically generated

**Figure 3:** *Interactive App with Realtime data with legend*

A screenshot of a map

Description automatically generated

**Figure 4:** *Admin page*  A screenshot of a phone

Description automatically generated

**Figure 5:** *Google maps integration with different icons*

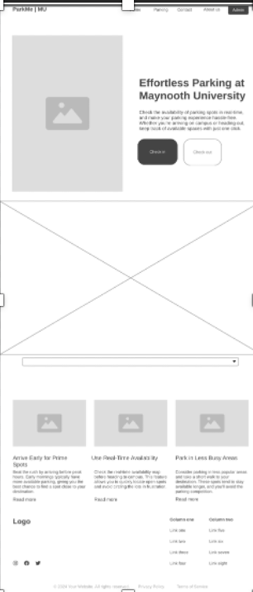
A screenshot of a map

Description automatically generatedA map with a blue line

Description automatically generated

## **Appendix B: Wireframe**

**Figure 6:** *Initial Wireframes*

A screenshot of a computer

Description automatically generated

## **Appendix C: Backend**

**Figure 7:** *MongoDB Dashboard*

A screenshot of a computer

Description automatically generated

**Figure 8:** *MongoDB Collection Sample*

A screenshot of a computer

Description automatically generated

**Figure 9:** *Server.js Config*

A screen shot of a computer program

Description automatically generated

**Figure 10:** *Seed.js with parking spot data for mongoDB*

A screenshot of a computer program

Description automatically generated

## **Appendix D: Gantt Chart**

**Figure 11:** *YouTrack Gantt Chart*

A screenshot of a computer

Description automatically generated

**Figure 12:** *You track gantt chart issues/items*

A screenshot of a computer

Description automatically generated

## **Appendix E: UI Iterations**

### 11th October

A screenshot of a parking map

Description automatically generated

### 25th October

A screen shot of a parking management

Description automatically generatedA map of a parking spot

Description automatically generated

### 6th November

A screenshot of a parking lot

Description automatically generatedA screenshot of a map

Description automatically generated

### 13th November

A screenshot of a parking lot

Description automatically generatedA screenshot of a graph

Description automatically generated

A screenshot of a parking permit information

Description automatically generatedA screenshot of a map

Description automatically generated