ParkMe | MU

Interim Report

Student: Redon Ferizi (21358591) (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

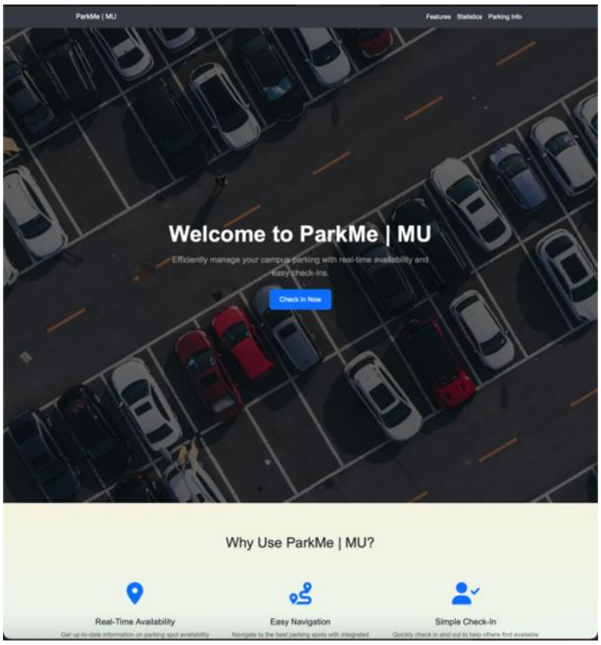
The next phases of the project will focus on completing feature implementation, including push notifications, a "where is my car" feature, and the ability to plan parking before arrival. Additionally, the application will incorporate cookie functionality to enhance user experience. Thorough unit, integration, 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*



Welcome to ParkMe | MU

Efficiently manage your campus parking with real-time availability and easy check-ins.

Why Use ParkMe | MU?

Real-Time Availability

Get up-to-date information on parking spot availability

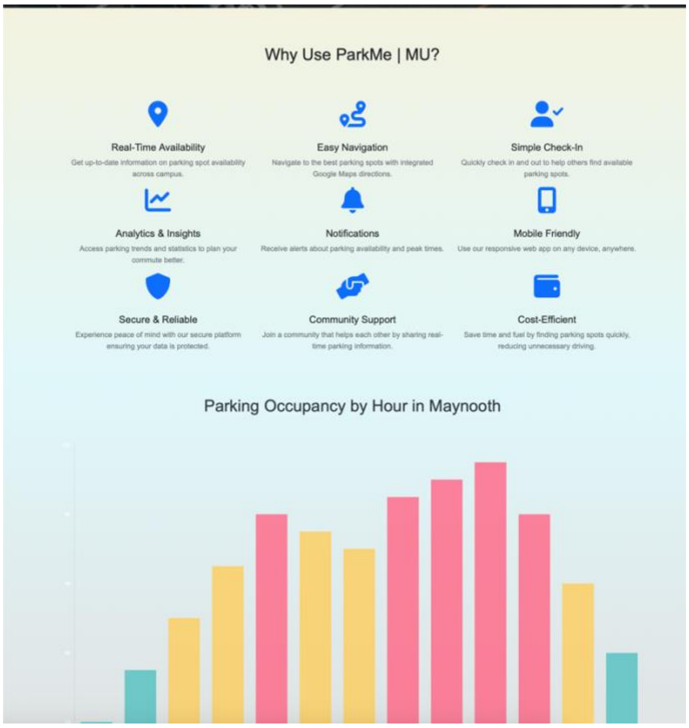
Easy Navigation

Navigate to the best parking spots with integrated

Simple Check-In

Quickly check in and out to help others find available

Figure 1: *Features under landing*



Why Use ParkMe | MU?

Real-Time Availability

Get up-to-date information on parking spot availability across campus.

Easy Navigation

Navigate to the best parking spots with integrated Google Maps directions.

Simple Check-In

Quickly check in and out to help others find available parking spots.

Analytics & Insights

Access parking trends and statistics to plan your commute better.

Notifications

Receive alerts about parking availability and peak times.

Mobile Friendly

Use our responsive web app on any device, anywhere.

Secure & Reliable

Experience peace of mind with our secure platform ensuring your data is protected.

Community Support

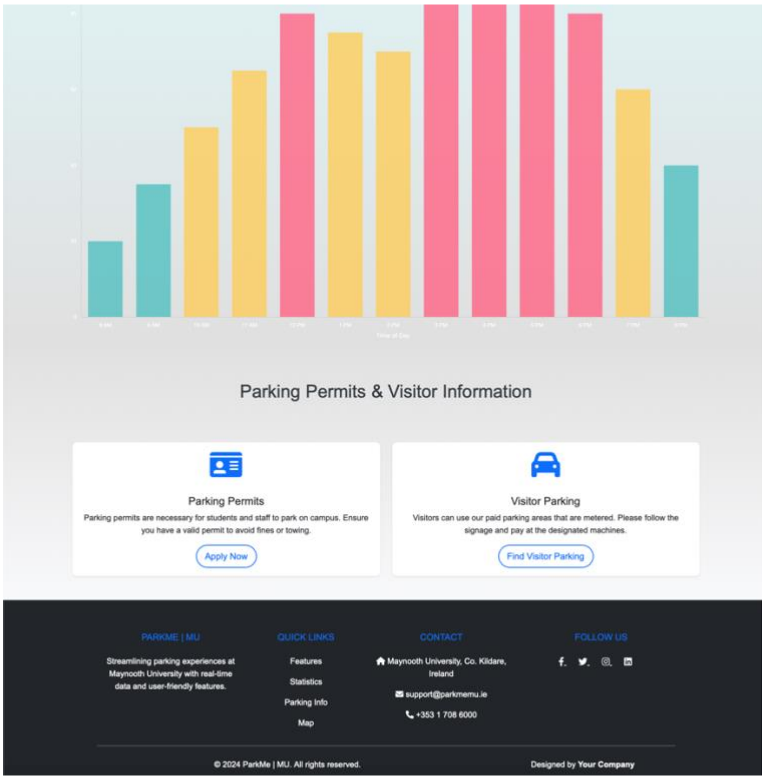
Join a community that helps each other by sharing real-time parking information.

Cost-Efficient

Save time and fuel by finding parking spots quickly, reducing unnecessary driving.

Parking Occupancy by Hour in Maynooth

Figure 2: *Occupancy Graph under features*



Parking Permits & Visitor Information

Parking Permits

Parking permits are necessary for students and staff to park on campus. Ensure you have a valid permit to avoid fines or towing.

Visitor Parking

Visitors can use our paid parking areas that are metered. Please follow the signage and pay at the designated machines.

PARKME | MU

Streamlining parking experiences at Maynooth University with real-time data and user-friendly features.

QUICK LINKS

CONTACT

Maynooth University, Co. Kildare, Ireland

support@parkmemu.ie

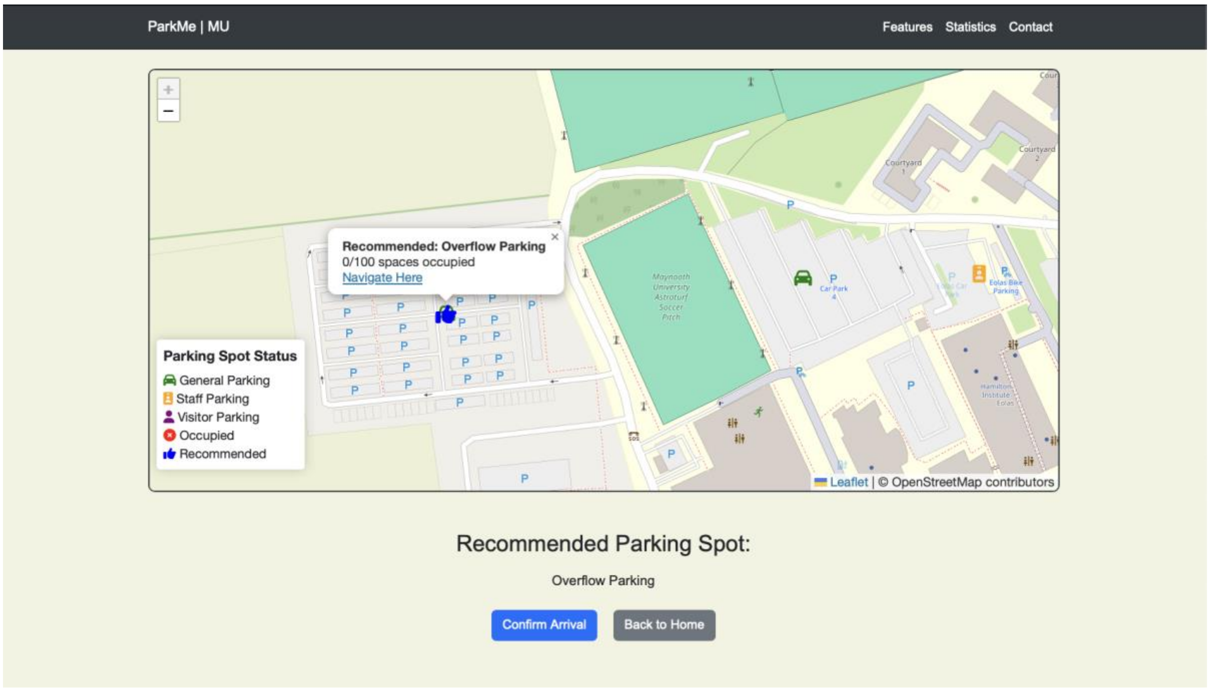
+353 1 708 6000

FOLLOW US

© 2024 ParkMe | MU. All rights reserved.

Designed by **Your Company**

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



Recommended: Overflow Parking
0/100 spaces occupied
Navigate Here

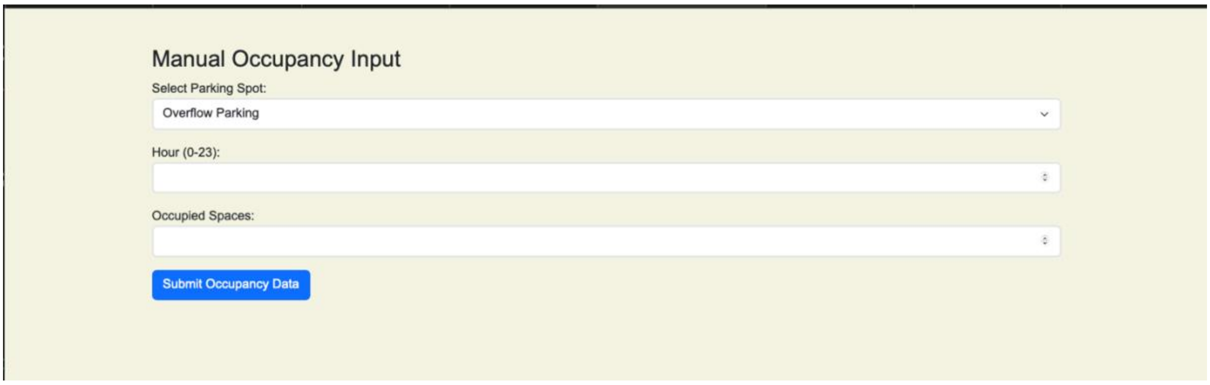
Parking Spot Status

- General Parking
- Staff Parking
- Visitor Parking
- Occupied
- Recommended

Recommended Parking Spot:

Overflow Parking

Figure 4: *Admin page*



Manual Occupancy Input

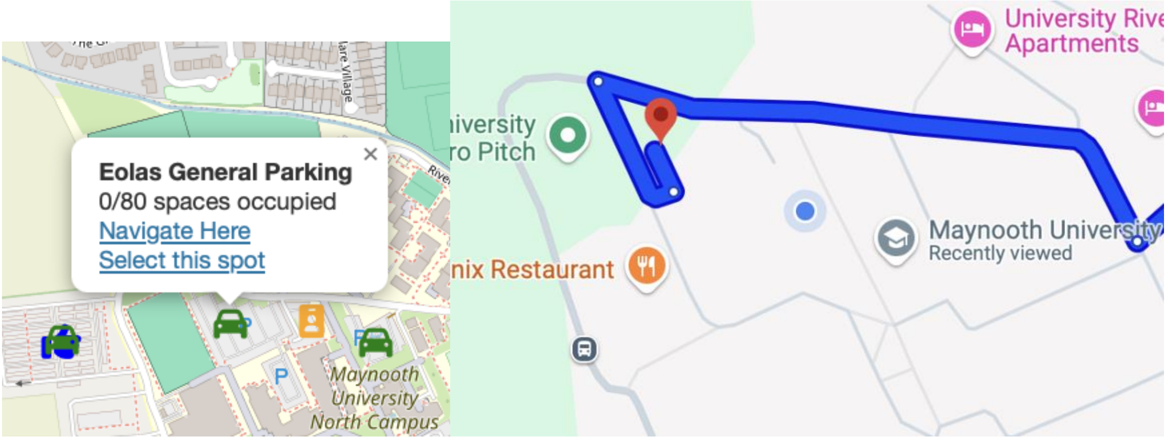
Select Parking Spot:

Overflow Parking

Hour (0-23):

Occupied Spaces:

Figure 5: *Google maps integration with different icons*

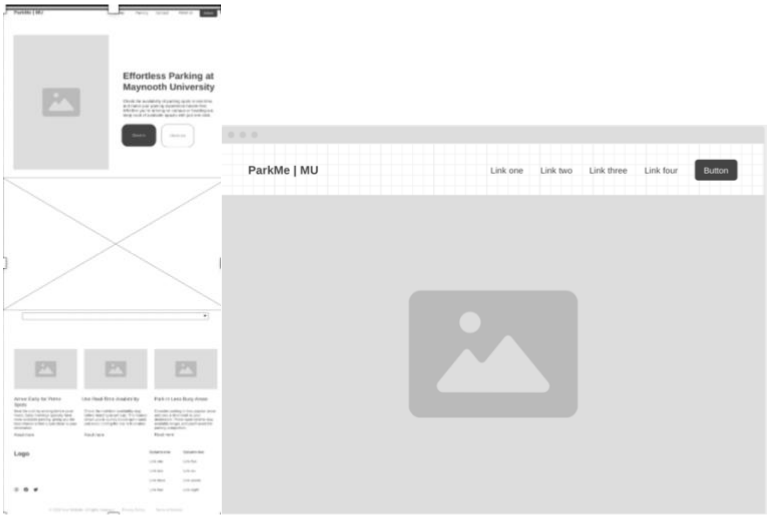


Eolas General Parking

0/80 spaces occupied

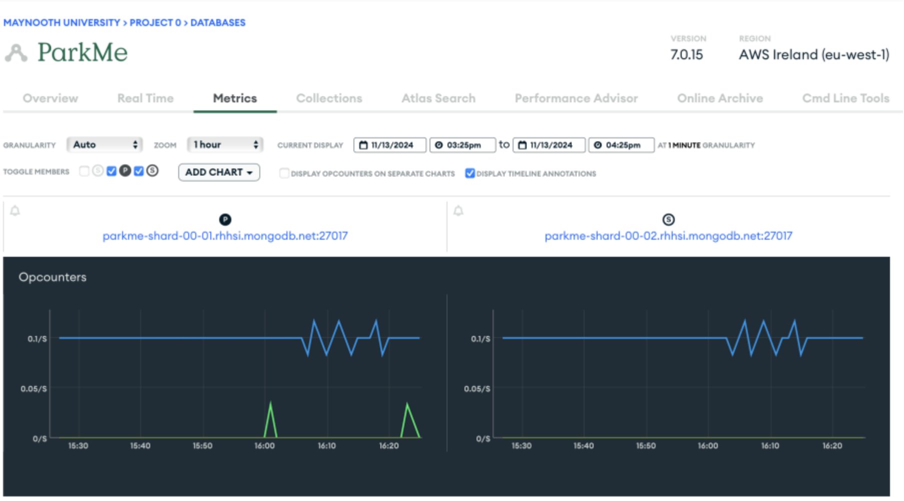
Appendix B: Wireframe

Figure 6: *Initial Wireframes*



Appendix C: Backend

Figure 7: *MongoDB Dashboard*



VERSION 7.0.15

REGION AWS Ireland (eu-west-1)

GRANULARITY Auto ZOOM 1 hour CURRENT DISPLAY 11/13/2024 03:25pm to 11/13/2024 04:25pm AT 1 MINUTE GRANULARITY

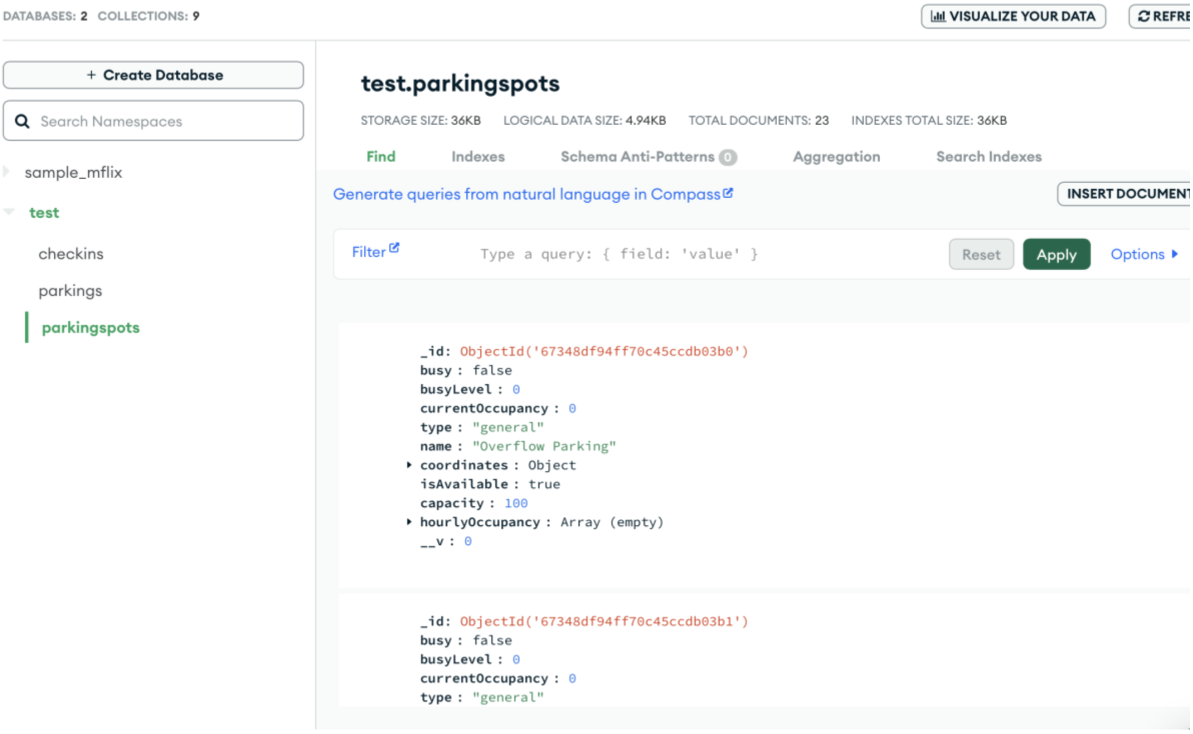
TOGGLE MEMBERS □ Ⓢ ☑ P ☑ Ⓢ

□ DISPLAY OPCOUNTERS ON SEPARATE CHARTS ☑ DISPLAY TIMELINE ANNOTATIONS

parkme-shard-00-01.rhhsi.mongodb.net:27017

parkme-shard-00-02.rhhsi.mongodb.net:27017

Figure 8: *MongoDB Collection Sample*



DATABASES: 2 COLLECTIONS: 9

test.parkingspots

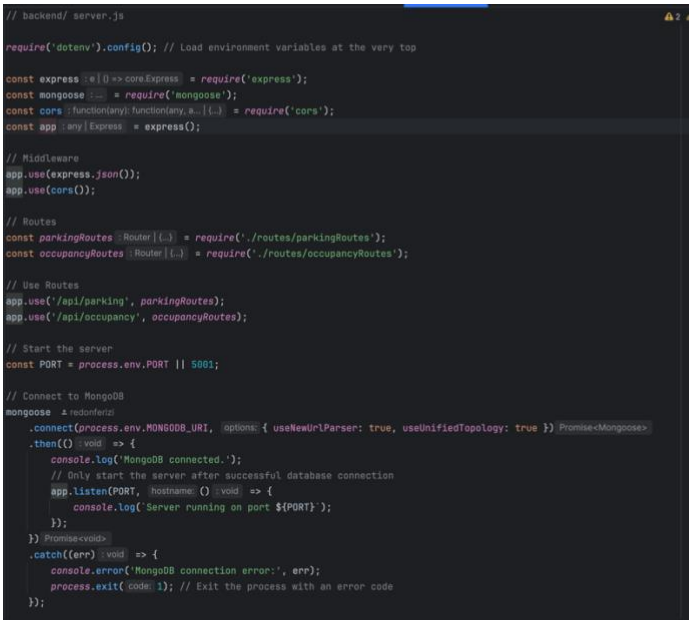
STORAGE SIZE: 36KB LOGICAL DATA SIZE: 4.94KB TOTAL DOCUMENTS: 23 INDEXES TOTAL SIZE: 36KB

Filter Type a query: { field: 'value' }

```
_id: ObjectId('67348df94ff70c45ccdb03b0')
busy : false
busyLevel : 0
currentOccupancy : 0
type : "general"
name : "Overflow Parking"
coordinates : Object
isAvailable : true
capacity : 100
hourlyOccupancy : Array (empty)
__v : 0
```

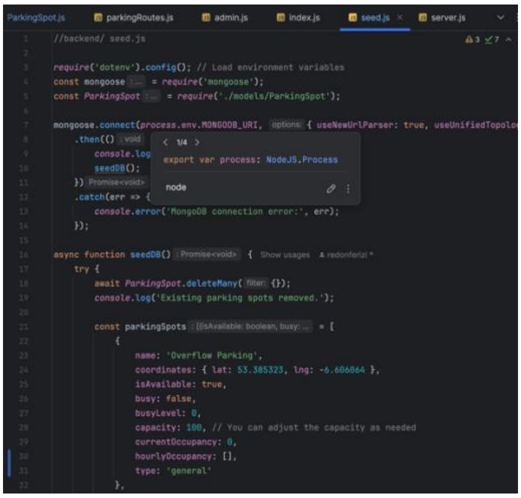
```
_id: ObjectId('67348df94ff70c45ccdb03b1')
busy : false
busyLevel : 0
currentOccupancy : 0
type : "general"
```

Figure 9: *Server.js Config*



```
// backend/ server.js

require('dotenv').config(); // Load environment variables at the very top

const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
const app = express();

// Middleware
app.use(express.json());
app.use(cors());

// Routes
const parkingRoutes = require('./routes/parkingRoutes');
const occupancyRoutes = require('./routes/occupancyRoutes');

// Use Routes
app.use('/api/parking', parkingRoutes);
app.use('/api/occupancy', occupancyRoutes);

// Start the server
const PORT = process.env.PORT || 5001;

// Connect to MongoDB
mongoose
    .connect(process.env.MONGODB_URI, { useNewUrlParser: true, useUnifiedTopology: true })
    .then(() => {
        console.log('MongoDB connected.');
        // Only start the server after successful database connection
        app.listen(PORT, () => {
            console.log(`Server running on port ${PORT}`);
        });
    })
    .catch((err) => {
        console.error('MongoDB connection error:', err);
        process.exit(1); // Exit the process with an error code
    });
```

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



```
//backend/ seed.js

require('dotenv').config(); // Load environment variables
const mongoose = require('mongoose');
const ParkingSpot = require('./models/ParkingSpot');

mongoose.connect(process.env.MONGODB_URI, { useNewUrlParser: true, useUnifiedTopolo
    .then(() 
        console.log
        seedDB();
    })
    .catch(err => {
        console.error('MongoDB connection error:', err);
    });

async function seedDB() {
    try {
        await ParkingSpot.deleteMany({});
        console.log('Existing parking spots removed.');

        const parkingSpots = [
            {
                name: 'Overflow Parking',
                coordinates: { lat: 53.385323, lng: -6.606864 },
                isAvailable: true,
                busy: false,
                busyLevel: 0,
                capacity: 100, // You can adjust the capacity as needed
                currentOccupancy: 0,
                hourlyOccupancy: [],
                type: 'general'
            },
```

Appendix D: Gantt Chart

Figure 11: *YouTrack Gantt Chart*

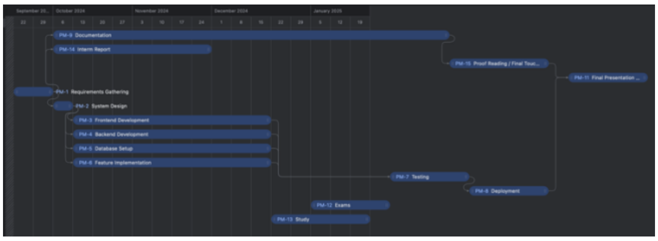
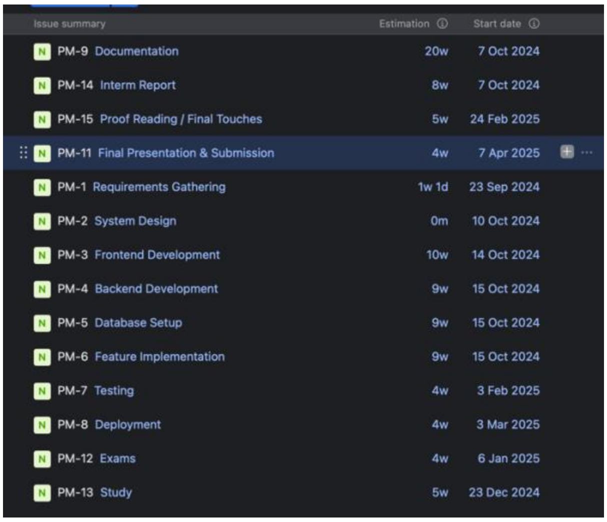


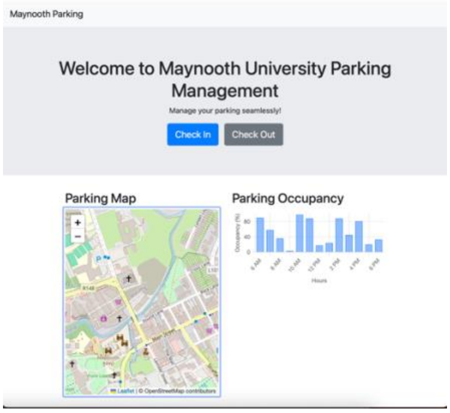
Figure 12: *You track gantt chart issues/items*



Issue summary	Estimation	Start date
N PM-9 Documentation	20w	7 Oct 2024
N PM-14 Interm Report	8w	7 Oct 2024
N PM-15 Proof Reading / Final Touches	5w	24 Feb 2025
N PM-11 Final Presentation & Submission	4w	7 Apr 2025
N PM-1 Requirements Gathering	1w 1d	23 Sep 2024
N PM-2 System Design	0m	10 Oct 2024
N PM-3 Frontend Development	10w	14 Oct 2024
N PM-4 Backend Development	9w	15 Oct 2024
N PM-5 Database Setup	9w	15 Oct 2024
N PM-6 Feature Implementation	9w	15 Oct 2024
N PM-7 Testing	4w	3 Feb 2025
N PM-8 Deployment	4w	3 Mar 2025
N PM-12 Exams	4w	6 Jan 2025
N PM-13 Study	5w	23 Dec 2024

Appendix E: UI Iterations

11th October

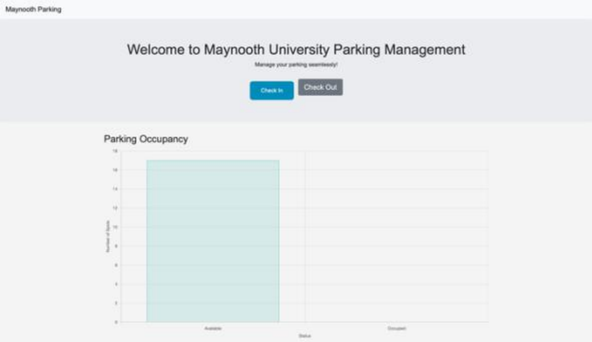


Maynooth Parking

Welcome to Maynooth University Parking Management

Manage your parking seamlessly!

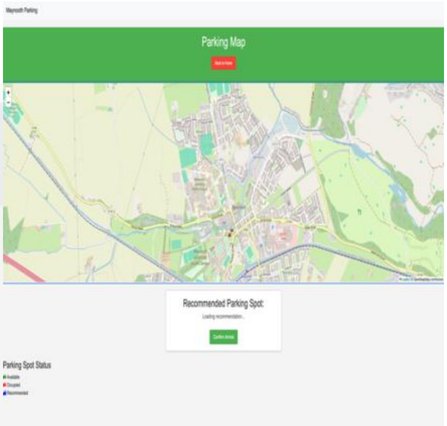
25th October



Maynooth Parking

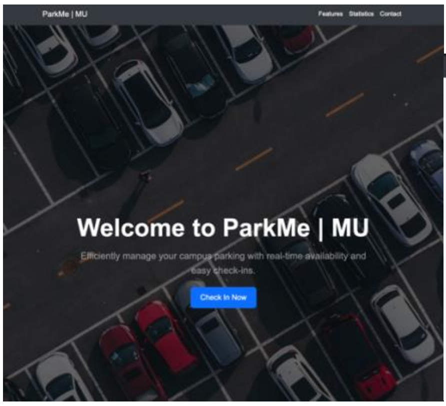
Welcome to Maynooth University Parking Management

Manage your parking seamlessly!



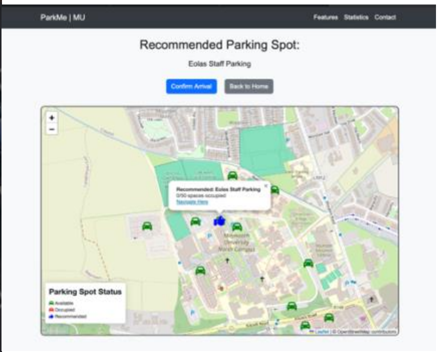
Maynooth Parking

6th November



Welcome to ParkMe | MU

Efficiently manage your campus parking with real-time availability and easy check-ins.


ParkMe | MU
Features Statistics Contact

Recommended Parking Spot:

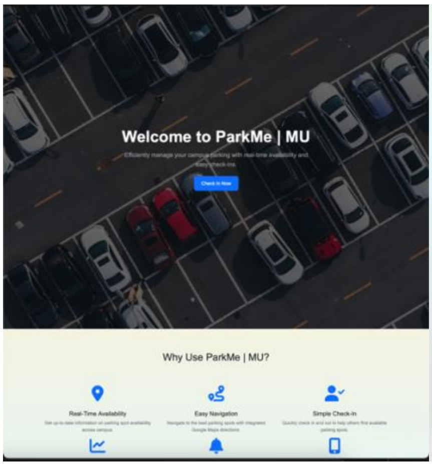
Eolas Staff Parking

Confirm Arrival
Back to Home
Recommended: Eolas Staff Parking
0/50 spaces occupied
Navigate Here

Parking Spot Status
- Available
- Occupied
- Recommended

Leaflet | © OpenStreetMap contributors

13th November


Welcome to ParkMe | MU

Efficiently manage your campus parking with real-time availability and easy check-ins.

Why Use ParkMe | MU?

Real-Time Availability

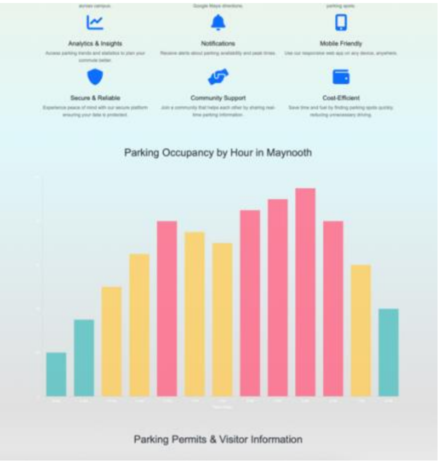
Get up-to-date information on parking spot availability across campus.

Easy Navigation

Navigate to the best parking spots with integrated Google Maps directions.

Simple Check-In

Quickly check in and out to help others find available parking spots.


across campus.

Google Maps directions.

parking spots.

Analytics & Insights

Access parking trends and statistics to plan your commute better.

Notifications

Receive alerts about parking availability and peak times.

Mobile Friendly

Use our responsive web app on any device, anywhere.

Secure & Reliable

Experience peace of mind with our secure platform ensuring your data is protected.

Community Support

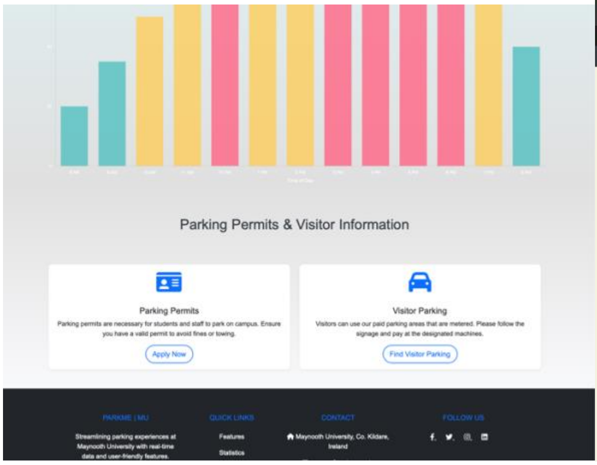
Join a community that helps each other by sharing real-time parking information.

Cost-Efficient

Save time and fuel by finding parking spots quickly, reducing unnecessary driving.

Parking Occupancy by Hour in Maynooth

Parking Permits & Visitor Information


Parking Permits & Visitor Information

Parking Permits

Parking permits are necessary for students and staff to park on campus. Ensure you have a valid permit to avoid fines or towing.

Visitor Parking

Visitors can use our paid parking areas that are metered. Please follow the signage and pay at the designated machines.

PARKME | MU

Streamlining parking experiences at Maynooth University with real-time data and user-friendly features.

QUICK LINKS

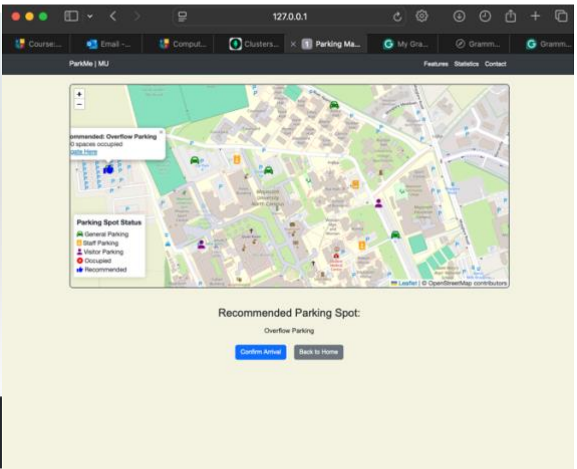
Features

Statistics

CONTACT

Maynooth University, Co. Kildare, Ireland

FOLLOW US



Parking Spot Status
- General Parking
- Staff Parking
- Visitor Parking
- Occupied
- Recommended

Recommended Parking Spot:

Overflow Parking