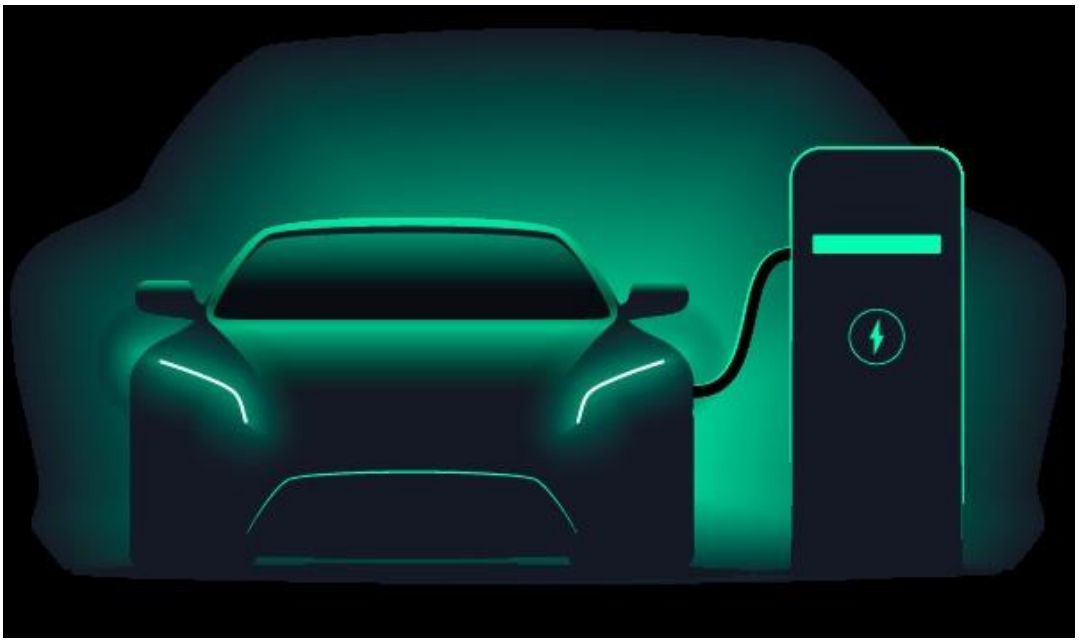
April 2024

“EV Car Hire Reservation Web Application”

Final Report



Name: Damien Driver

Student ID: 20099851

Supervising Lecturer: TJ McDonald

Course: Higher Diploma in Computer Science

Module: Project

Contents

Table of Figures

Declaration

I declare that the work which follows is my own, and that any quotations from any sources (e.g. books, journals, the internet) are clearly identified as such using ‘single quotation marks’, for shorter excerpts and identified italics for longer quotations. All quotations and paraphrases are accompanied by (date, author) in the text and a fuller citation is the bibliography. I have not submitted the work represented in this report in any other course of study leading to an academic award.

Student: DAMIEN DRIVER

Date: 01/04/2024

Abstract

EV Car Hire is a user-friendly web application to allow users book electric vehicle rentals. The React frontend invites users to sign up, browse the available fleet and easily filter by dates and locations. The platform simplifies the process of booking an eco-friendly ride with an integrated stripe payment facility. The backend of the application is built using Node and Express with data stored in MongoDB. An API from openchargemap provides information on the nearest available charge points for user convenience.

Acknowledgements

I would like to express my sincere gratitude to all the course lecturers in SETU who have contributed to my development over the past two years and assisted in enabling me to complete this project. Their support and encouragement have been invaluable.

I extend my heartfelt appreciation to my family for their unwavering support and understanding during the course duration. Their encouragement and belief in me have been my constant motivation.

Introduction

Project Background

My proposal was to develop an EV car hire reservation web application. The web app will streamline the booking process of electric vehicles and improve the customer experience. The project will include a vehicle booking engine, a pricing system with a fleet management integration and reservation management facility.

Motivation

The idea for the project came from observations of problems potential customers have finding EV cars on old-style car hire websites. EV's are currently hidden in the existing traditional rental fleet. EV rentals have unique terms and conditions in relation to battery charging that differ from standard internal combustion engine vehicles. The design of many existing car hire websites does not allow efficient display of these EV terms and conditions. Example shown below of Hertz website (Hertz Rent A Car, 2024).

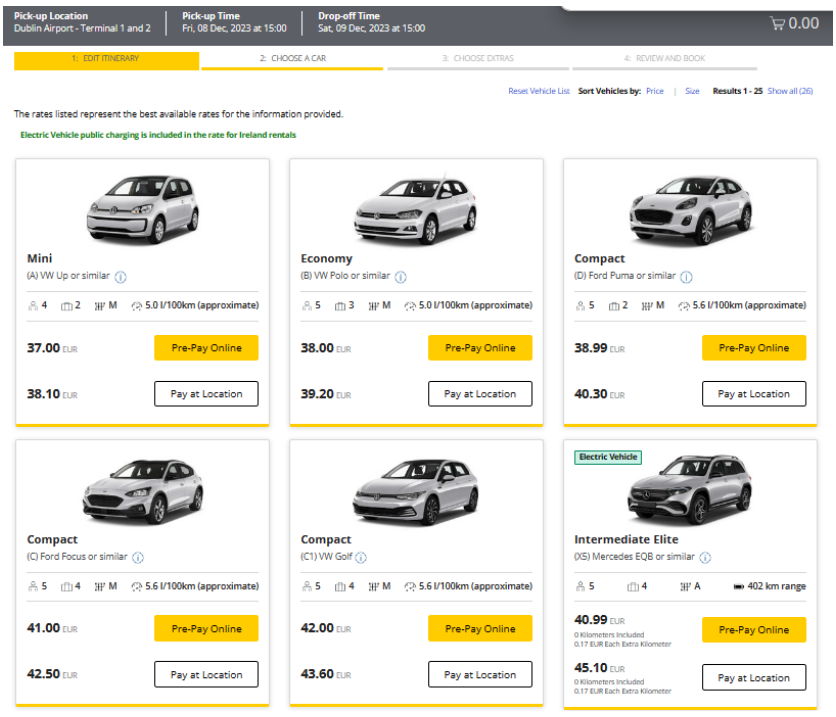


Figure 1 Example of Hertz Website

Objectives

- To build a web application that provides a user-friendly interface, allowing users to easily view and select available EV cars and enhance the overall user experience.
- Design a robust reservation system that enables users to select rental dates, select car models and confirm bookings. A calendar feature to be implemented to facilitate easy date selection.
- Build a secure and scalable backend system. Develop API endpoints for handling user authentication, car availability, reservations, etc.
- Database integration to store user information, car details and reservation data securely. Ensure proper indexing and data structure for efficient data query.
- Implement a secure user authentication system to protect user accounts and reservation data. Include security features such as password hashing and session management.
- Develop an admin dashboard to manage car inventory, view reservations and fleet management. Implement roles and permissions to differentiate between regular user and administrators.
- Integrate a secure payment gateway to facilitate online payment for reservations. Ensure compliance with industry standards for secure payment processing.
- Create an email service to send confirmation notification of reservation and payments.
- Implement a mapping API to enable users view charging points on an interactive map in their local area.
- Building of a web-scraping functionality to canvass competitor pricing to ensure competitiveness in the marketplace.

Project Methodology

I choose to adopt an agile scrum software development methodology for my project. Dividing the development into smaller iterations called sprints. This allow changes to the plan to be easily absorbed into the build process.

Agile scrum methodology is a project management system that relies on incremental development. It's one of the most popular and widely used frameworks in agile development, and with good reason. It offers a wide range of benefits with several common variations that can adapt to different project and organization needs (Inflectra, 2024).

Research & Analysis

Introduction

Market analysis was undertaken on both traditional car rental companies (Hertz, Budget, Enterprise) and newer car sharing companies (GoCar, Yuko, Ufodrive) to identify current market offerings. Potential strengths and weakness of each companies' platforms were detailed to try to help create a best-in-class platform.

Front-end Frameworks

EV car hire is a web application developed using JavaScript. The front-end was built using the React framework. There were multiple other frameworks to choose from. Three other libraries I considered were Svelte, Angular and Vue.

React lets you build user interfaces out of individual pieces called components then combine them into entire screens, pages, and apps (React, 2024). It is highly regarded and well documented and its use of virtual DOM makes it easy to work with. React has a large ecosystem of libraries and tools which make it easy to find solutions for development needs.

Svelte is a relatively new framework that shifts the work from the browser to build step allowing for highly efficient code. It uses simple syntax and minimal boilerplate. Svelte is a radical new approach to building user interfaces. Whereas traditional frameworks like React and Vue do the bulk of their work in the browser, Svelte shifts that work into a compile step that happens when you build your app (Twilio, 2024). The disadvantage for me was that because it is newer there are fewer third-party libraries available.

Angular is a well-established framework maintained by Google. It allows for complex developments. It has a strong support structure with many built in tools for a variety of tasks. Angular uses TypeScript for development (Joshi, 2023). The learning curve seems to be steep and with time constraints learning a new framework from scratch was a negative.

Vue is a progressive JavaScript framework that is adaptable and is known for ease of integration. It is lightweight with good documentation. It's a framework that combined the best approaches to frontend development from frameworks like Angular and React with other features that made writing web apps faster, easier, and more pleasant (Nowak, 2023). The support community is smaller than React which usually leads to fewer third part resources.

Design Framework

Bootstrap was integrated to ensure a responsive design across many browsers and devices. Bootstrap is a Powerful, extensible, and feature-packed frontend toolkit (Mark Otto, Jacob Thornton, and Bootstrap contributors, 2024).

Ant Design library was also used to integrate the calendar date selector due to it flexibility and ease of use.

Some alternatives that were considered were Material-UI and tailwind CSS. Material-UI is a React component library that implements Google material design principles. It has built in support for theming and custom navigation bars. Tailwind CSS is a utility first framework. It enables fast UI development by styling HTML elements. It allows for unique design features.

Back End Framework

EV car hire is a web application developed using JavaScript. The back end utilises node.js with the Express framework to build a secure and scalable project.

Node.js with Express allows for use of JavaScript on both the front and back ends, promoting code sharing and reduced need to switch languages. Node.js is highly scalable with a vast, active ecosystem. The wide range of NPM packages enable rapid development and easy integration of third-party libraries. Node.js is fast making it suitable for real time applications. Express is a minimalistic and flexible web framework built on top of Node.js, providing essential features for reliable web applications. Express allows middleware functions making it easy to add additional functions such as authentication and error handling. This makes Express ideal for building RESTful APIs in a structed way.

Other back-end technologies that were considered include Django, Spring Boot, NestJS and Hapi.

Django is a high-level Python framework. It offers a comprehensive set of web application building features. It follows the Don't Repeat Yourself (DRY) principle. With many built in features like authentication and automatic URL routing it enables developers to build applications quickly. It has strict traits, and hence, offers limited flexibility (Patel, 2024).

Spring Boot is a Java based framework for building enterprise grade applications. It has extensive community support and simplifies the development of Java applications. Development speed can be slower due to verbose syntax of Java language. Spring Boot is more suited for large-scale enterprise applications (Zanini, 2023).

NestJS is a Node.js framework inspired by Angular's architecture and TypeScript features. NestJS makes page navigation smoother by enabling client-side route transitions (Jayabal, 2024). It provides for a modular structure with built in support for dependency injection and middleware. It simplifies complex development by enforcing best practices with support for database integration and authentication.

Hapi is a powerful tool built on Node.js. It offers flexibility with its plugin system. Hapi offers a configuration-centric structure that comes with pre-built caching and authorization features (Hiren, 2021). It is suitable for a vast range of applications from APIs to web servers. There is an extensive range of third-party integrations.

Planning

Trello

I am using Trello for managing the Project development. As part of the agile workflow, I have divided each sprint into a list and then added each piece of work onto a card. At the beginning of a sprint, I have a planning session to breakdown each piece of work and create a checklist to tick off each piece as it is completed. Adjustment is a core principle of scrum workflows and I have needed this flexibility as some features have taken far longer to implement than I first planned for.

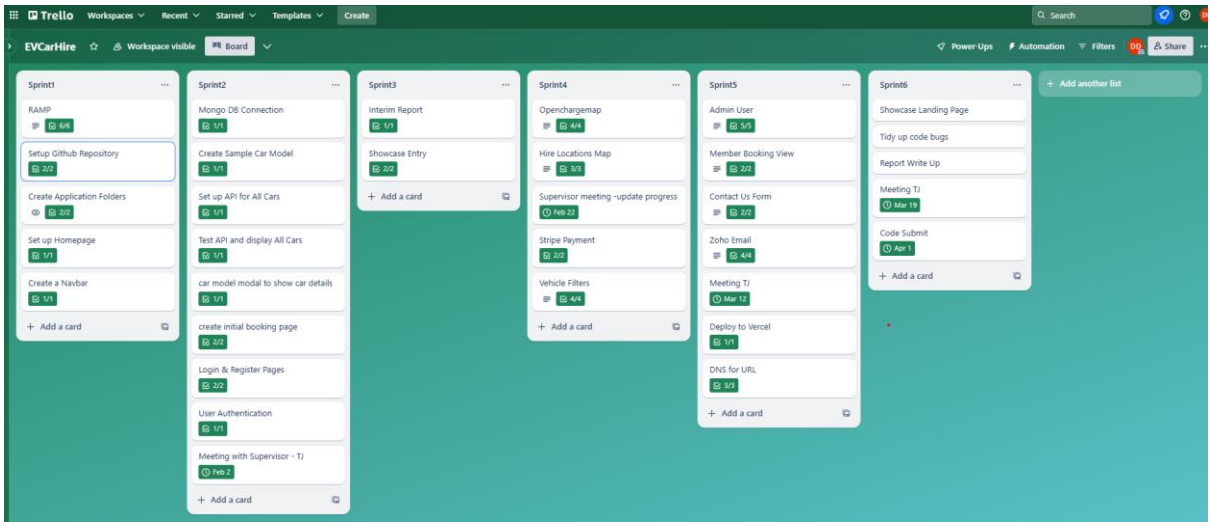


Figure 2 Trello Board

Development Plan

Backend Development:

The backend development process for my EV car hire application will begin with the installation of Node.js and Express, the initial technologies for building my web servers and APIs. Following this, I will establish a connection to MongoDB, a NoSQL database known for its flexibility and scalability.

With the database connection in place, I will proceed to define the data schema for my application, encompassing entities such as cars, users, locations and reservations. Each schema will be carefully structured to capture the essential attributes and relationships within the system. Armed with these schemas, I will start on creating rest API endpoints to facilitate CRUD operations, enabling users to interact with the application's data resources seamlessly.

As security is paramount, I hope to implement authentication and authorization mechanisms using JWT tokens, ensuring that only authorized users can access protected endpoints. To enhance the robustness of the system, I will integrate validation and error handling mechanisms, safeguarding against invalid data input and gracefully managing any unforeseen errors.

Finally, I will validate the functionality of my endpoints through thorough testing using Postman, verifying that they performed as intended and adhered to the specified requirements. By following this comprehensive approach, I hope to build a solid foundation for the backend of my EV car hire application.

Frontend Development:

In the frontend development process, the first step will be setting up the React project structure. This includes organizing directories and files for efficient development, laying the foundation for the application. Next, the UI will be designed using the CSS Bootstrap framework, providing a responsive and visually appealing layout.

Essential components such as the navbar, car availability, data selection, and user authorization will be created to establish the core functionality of the application. To navigate between different views, React Router will be implemented for efficient routing within the application. This enables users to move between the car options, reservation form, and other sections with ease, maintaining a smooth flow.

As user input is crucial, form validation and error handling will be implemented to ensure data accuracy and a smooth user experience. In case of errors, clear error messages will be displayed to guide users on corrections needed, enhancing usability.

This approach should result in an intuitive and visually appealing application that provides a seamless experience for users interacting with car listings, making reservations, and managing their user accounts.

Integration:

The integration process begins by seamlessly linking the backend APIs with the various frontend components, ensuring a cohesive interaction. Once connections are in place user authentication mechanisms will be implemented, safeguarding the system's security and privacy. This step establishes the foundation for secure user access, essential for protecting sensitive data.

Rigorous testing will be needed to examine the integration to validate the smooth flow of data between frontend and backend systems. This comprehensive testing phase aims to identify any potential bottlenecks or inconsistencies, and hopefully improving the reliability.

Through these steps, the integration between frontend and backend evolves into a system where seamless data exchanges are prioritized, providing dependable application ready to serve its intended purpose.

Features:

I hope to introduce a comprehensive set of features tailored to enhance the user experience and streamline the process of finding and renting electric vehicles. Users will benefit from a robust search and filter function for car listings, ensuring they can easily locate the perfect vehicle for their needs.

To make the rental process even more convenient, I'll be integrating maps to display rental locations and vehicle charging points, providing users with valuable geographical information at their fingertips.

Additionally, I understand the importance of a seamless payment process, I will be implementing payment integration to handle reservations effortlessly and securely. Users will have the ability to not only make reservations but also to view and manage them within the application, offering a centralized hub for all their rental needs.

On the administrative side, I will be creating a dedicated function for admins to efficiently manage cars, users, and reservations, simplifying the task of overseeing the system. With these features in place, I hope to deliver a user-friendly, efficient platform for all things related to EV car hire.

Deployment:

To deploy the application, the first step will be to create a GitHub repository. Following this, both the frontend and backend will be deployed to the Vercel platform (Vercel, 2024). From here all pushes to the GitHub repository will automatically be redeployed to the live applications.

The Domain Name System (DNS) will be configured to ensure seamless access to the deployed application. DNS is a critical component of the internet which will translate the IP addresses provided by Vercel into the readable domain name www.evcarhireireland.ie

Design

Wireframes

Initial designs were created in Figma. These designs are important as they are a space where ideas can be sketched without constraints. Styling and colour themes are not considered at this early stage.

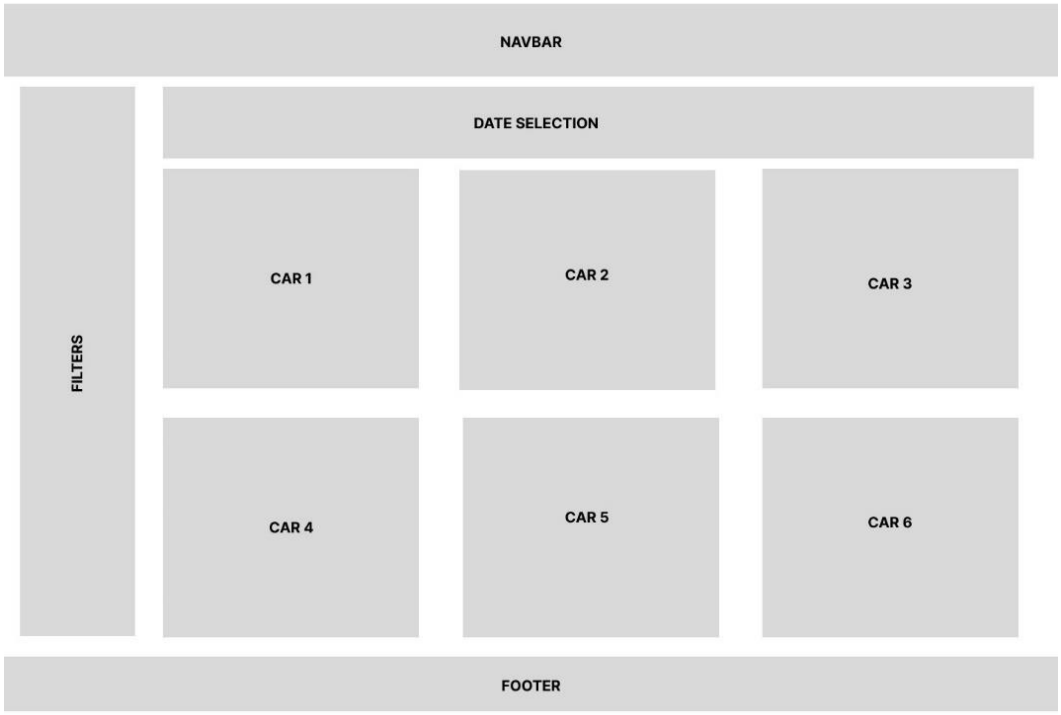


Figure 3 Wireframe of Homepage.

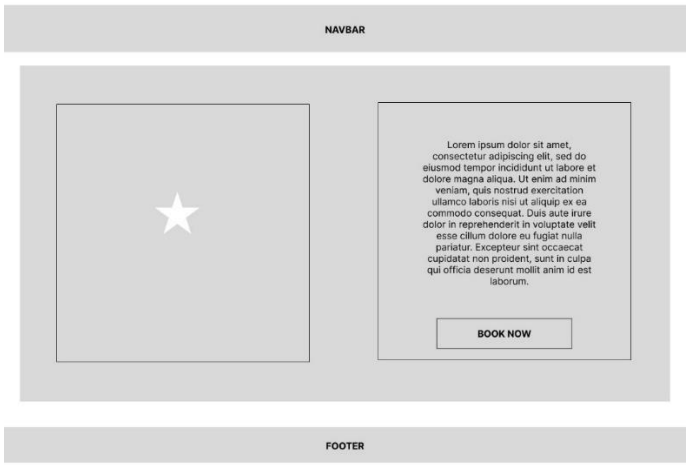


Figure 4 Wireframe of Booking Page.

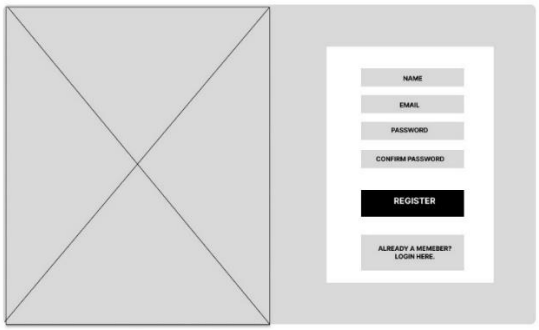


Figure 5 Wireframe of Register Page.

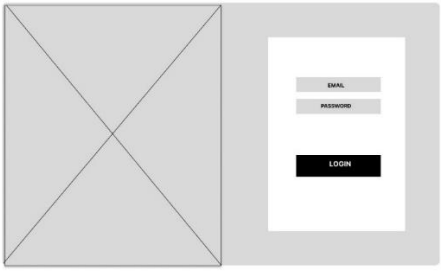


Figure 6 Wireframe of Login Page.

System Diagrams

Block Diagram

A simple block diagram that shows the principal pages of the application. The lines show the relationship between the pages.

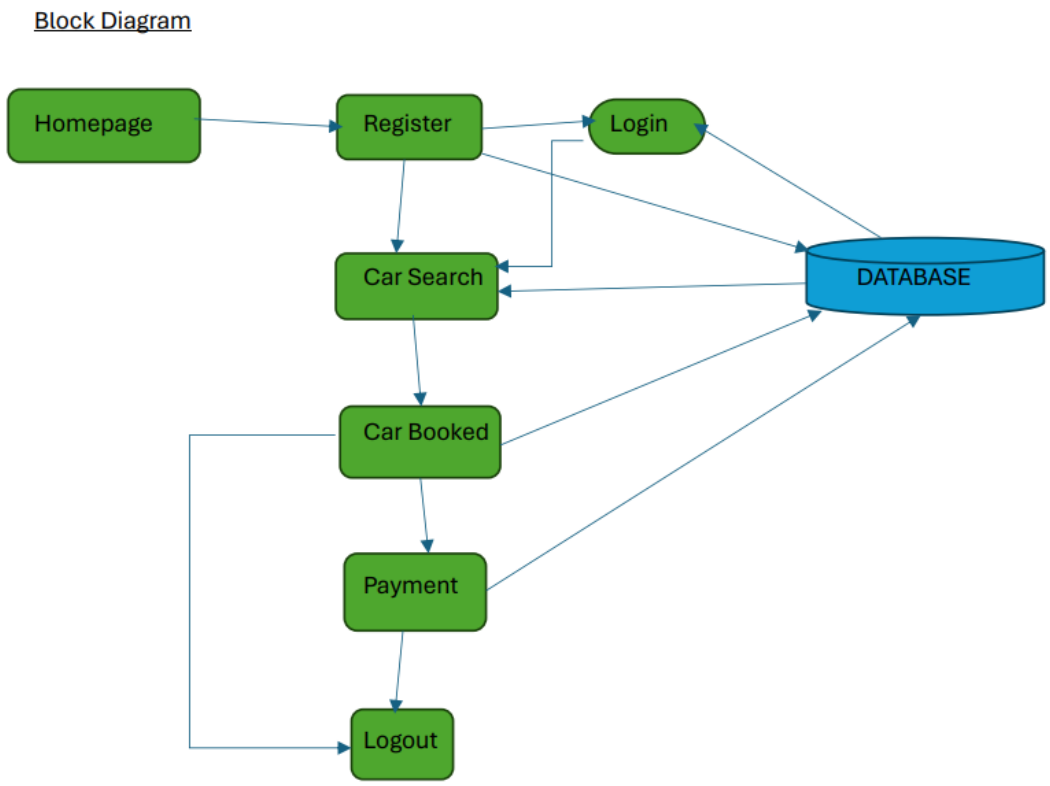


Figure 7 Block Diagram of system.

Entity Relationship Diagram

ER diagram that shows the relationship between the different people and objects within the application.

ER Diagram

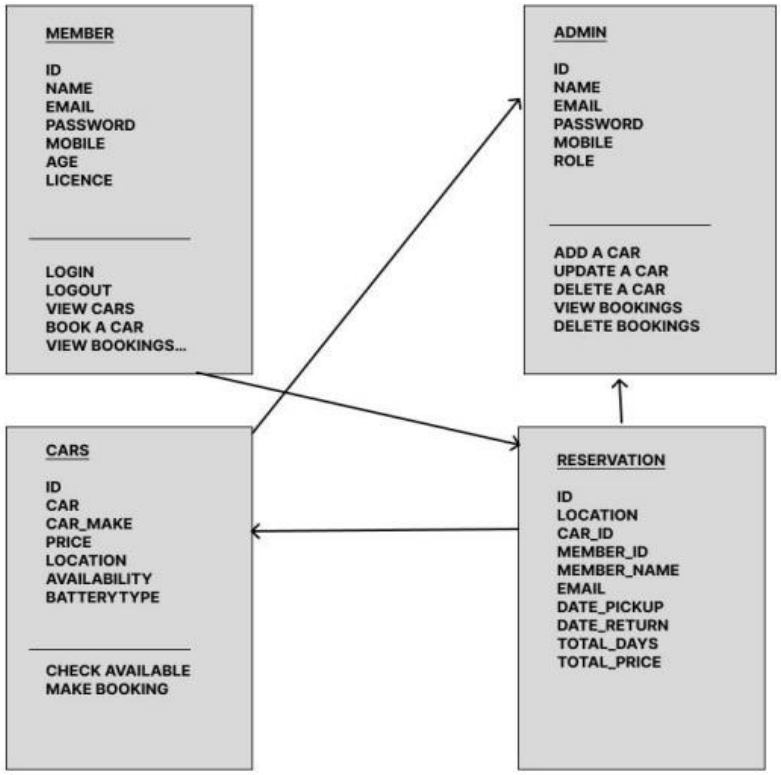


Figure 8 ER Diagram of system.

Flowchart Diagram

The flowchart is the roadmap of the actions within the application. It illustrates the entire process from landing page to booking.

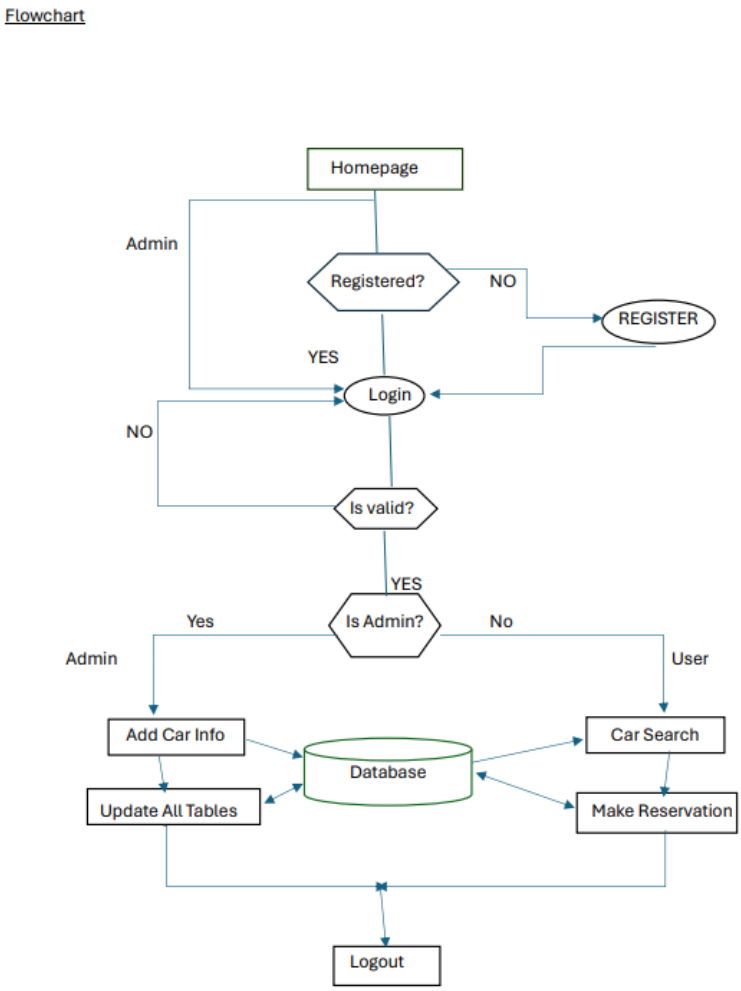


Figure 9 Flowchart Diagram of system.

Color Scheme

A green Color scheme theme was chosen for the application. EV car hire is rooted in the ethos of environmental consciousness. Green, symbolizing nature, growth, and sustainability, aligns seamlessly with the aim to promote eco-friendly transportation solutions.

After finding the image that is the main banner on the landing page, I used Adobe Color to extract the theme for the application. Based on the background Color I then used a font Color that was recommended by the Adobe Color checker (Adobe , 2024) .

The Lime green theme is a vibrant and energetic hue within the green spectrum. I feel it adds a touch of modernity and dynamism to the application's visual identity.

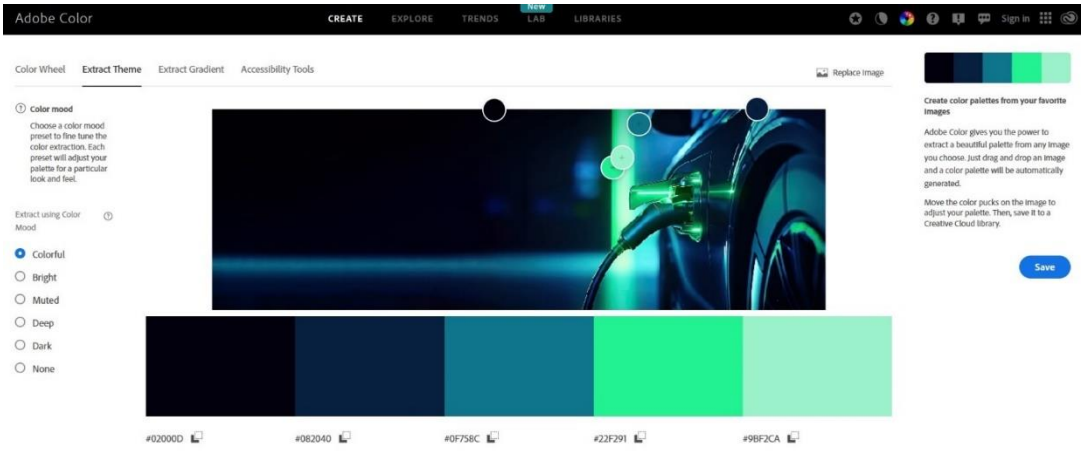


Figure 10 Adobe Color – Theme Extract

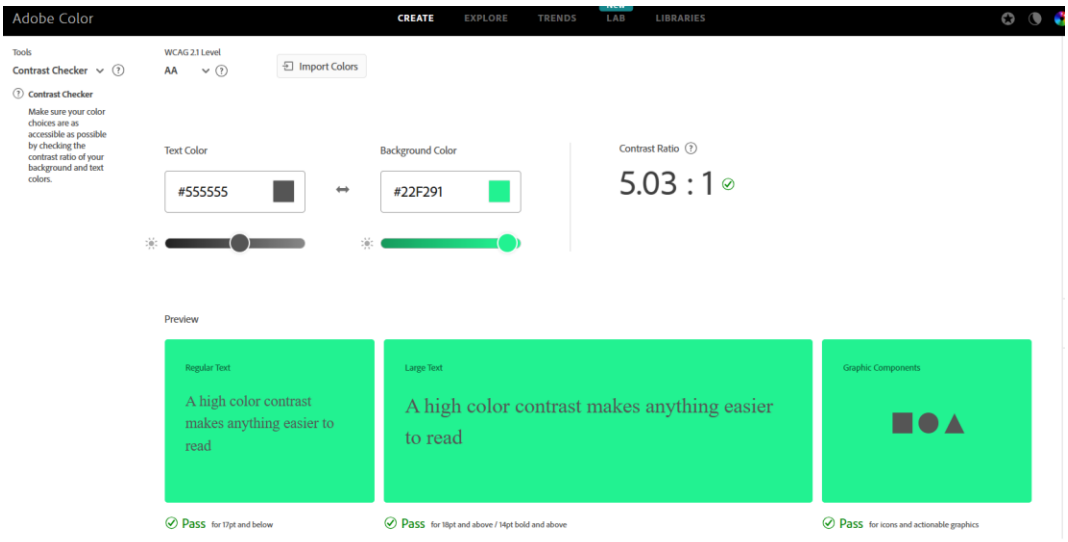


Figure 11 Color Contrast Checker.

Development – Project Implementation

Sprint 1

After the ramp phase was complete the aim for the first sprint was to get the application set up in its most basic form. The frontend would be built in React. React was chosen for its component-based architecture, which would allow for easier management of complex user interfaces later in the development. A simple, static homepage was designed with a navigation bar for easy access to other pages, though these would be placeholders until later in the development. Using HTML, CSS, and the Bootstrap framework, I ensured the page and especially the navbar had a responsiveness design. Learnings from YouTube were used to create a toggle button to open and close the menu and adjust the navbar for different screen sizes (Web Dev Creative, 2023).

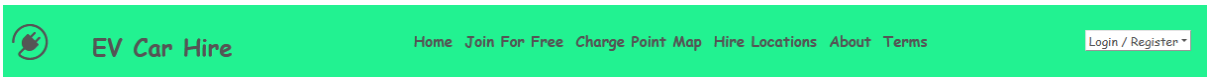


Figure 12 Navbar design.

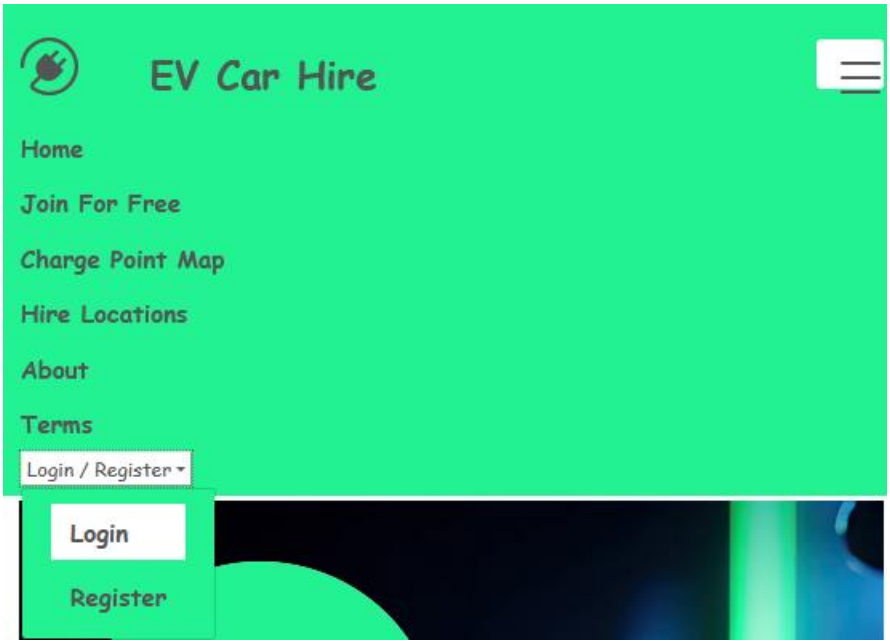


Figure 13 Example of Navbar Responsiveness

Sprint 2

The focus in sprint two turned to developing a backend API using Express.js and Node.js to deliver car options for the homepage. The backend car model and car routes were created to serve data to the frontend, I used Postman to test my rest HTTP API routes. Postman is a great tool as it allows testing in a safe environment and can help identify errors before implementation into frontend. Postman will continue to be an essential tool for testing data retrieval from both internal and external sources as development progresses. Locations, Users and charge point data would all be tested.

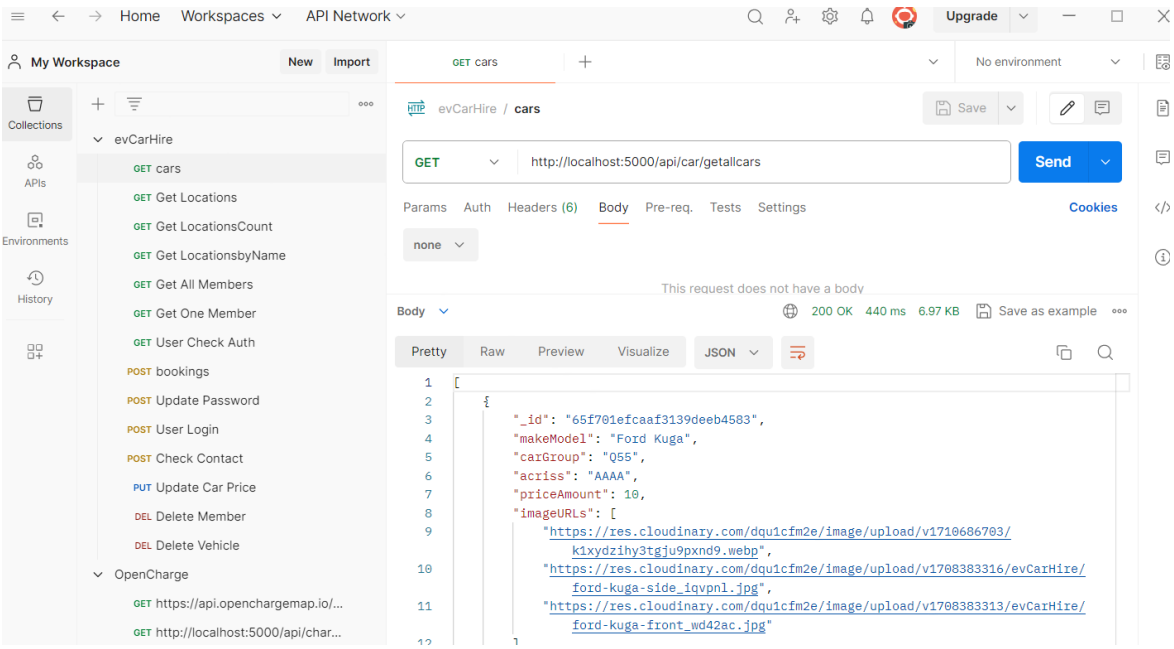


Figure 14 Example of Postman API testing.

The original plan was to have the cars displayed on the homepage. This was to allow all users to see available vehicles and seamlessly get to the vehicle booking page. However, this plan changed as I decided to make the application more of a member's style offering. This meant that I had to refactor to move the car display functionality to the vehicle options page and create a separate homepage / landing page for the application.

The vehicle options page is built using a grid system where each individual car is displayed on its own bootstrap card. As new cars are added the grid expands in a responsive manner to keep a stylish look and feel to the page. Each vehicle card has a "more car details" button which opens a modal which displays images in a carousal style to the user and has some additional information about the hire vehicle. The modal design was based on learnings from Hotel Booking System React - (Next Code, 2022)

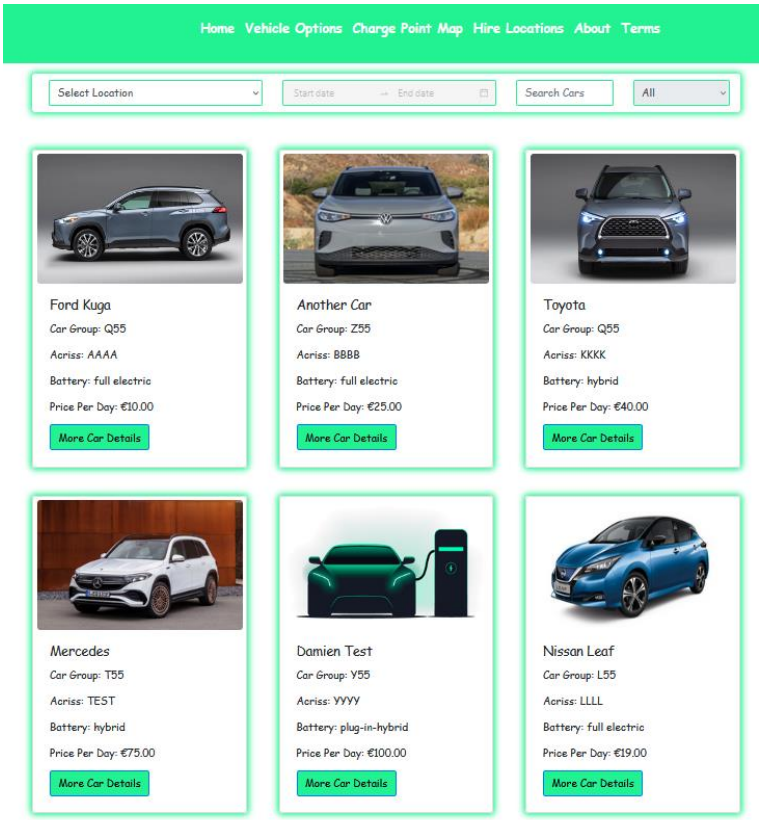


Figure 15 Example of Vehicle Options display.

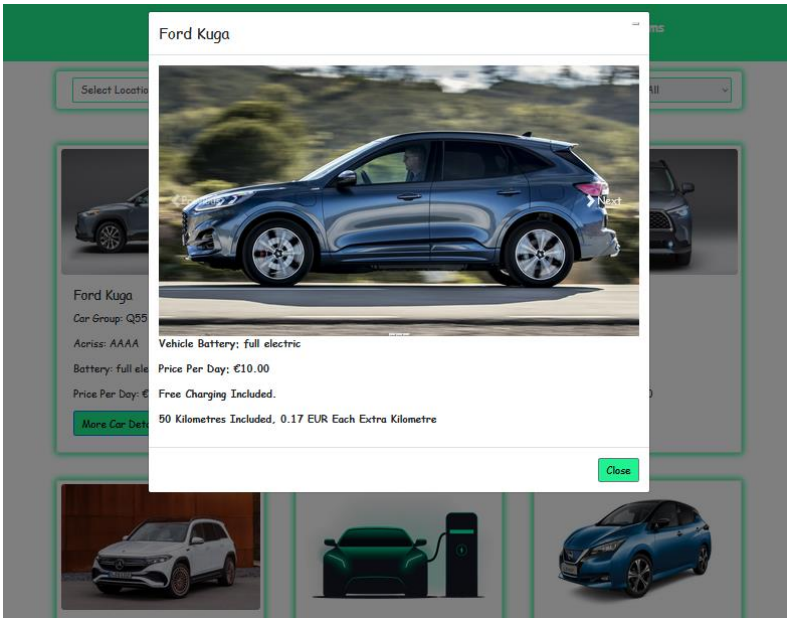


Figure 16 Example of the car modal.

The decision to make the vehicle options page view only available to logged in members meant that I needed a new landing page for the Homepage of the application. From here the call-to-action message was to direct interested users to Join for Free and link them to the Register page.

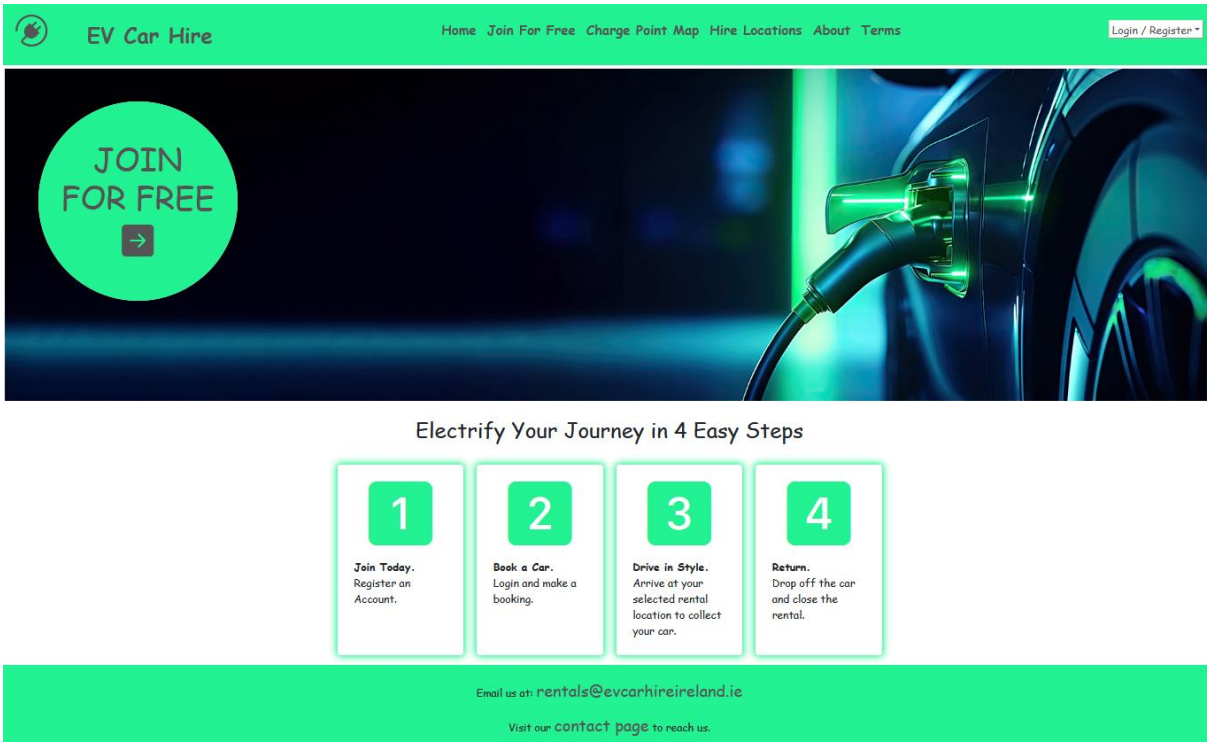


Figure 17 Application Homepage

The register and Login pages were created with simplicity in mind. The aim was to provide user friendly experience while ensuring data integrity. The user can register using their name, email and password. Once registered they are redirected to the Login page. There are simple checks to ensure that the email entered is in a valid format and that it doesn't already exist in the database. Regex is used to ensure that users only put alpha characters or spaces in the name field and an error displayed to advise the user if data validation rules are not adhered too.

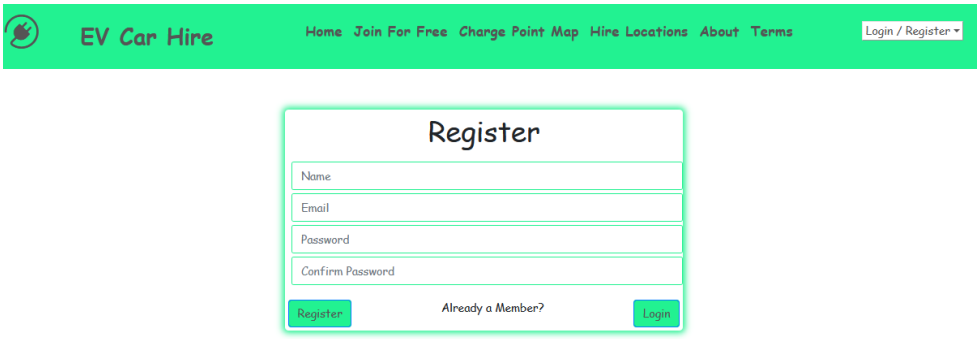


Figure 18 Register Page

Once redirected to the Login page the user simply enters their email and password. There is a facility to reset their password if they have forgotten it. This is done via an email with a unique token.

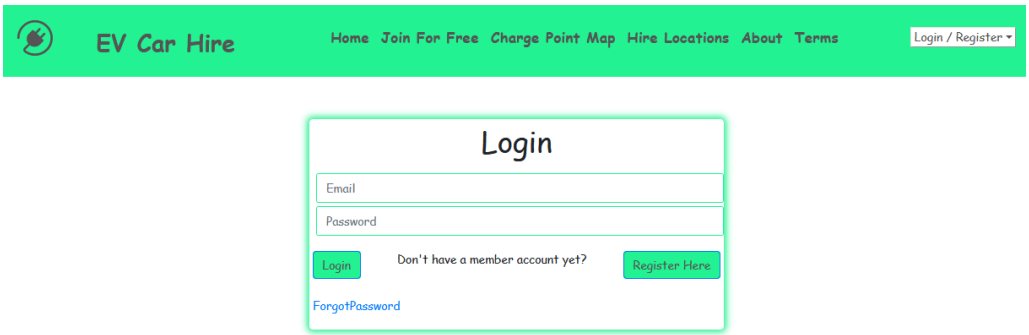


Figure 19 Login Page

When the member has logged in and has selected their car, they are directed to the booking page which provides a summary of the car, the pickup location and the time and dates, and most importantly the total price of the rental.

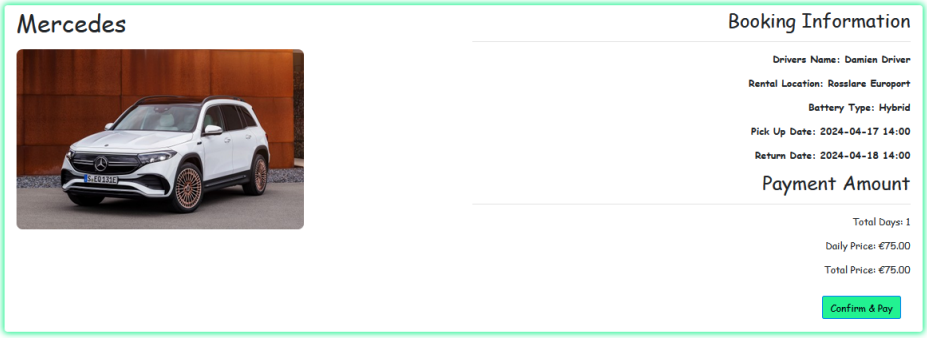


Figure 20 Example of Booking Page

Sprint 3 & 4

Little development work was done in sprint 3 as the focus was on the interim report deadline. In sprint 4 Stripe was integrated to allow for payment of the hire vehicle. The integration with stripe was very straight forward and developer friendly. Tutorial video from Hotel Booking System React was a great aid in the process (Next Code, 2022).

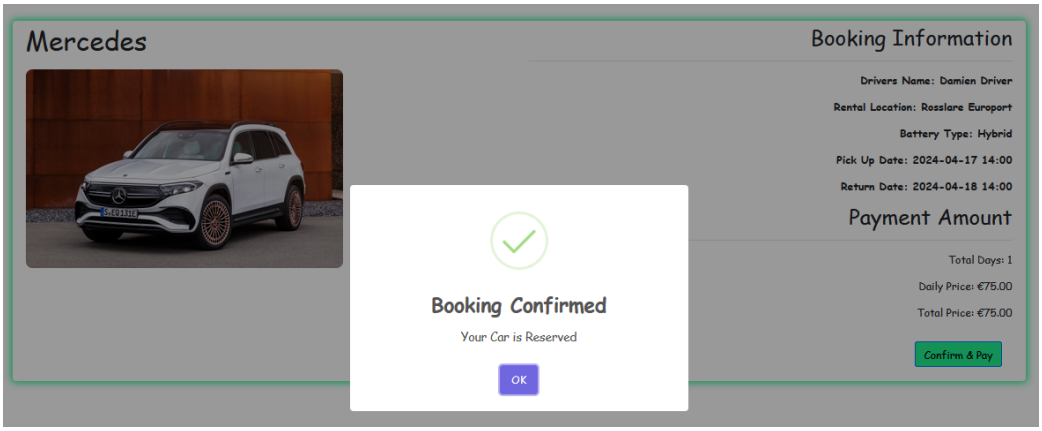


Figure 21 Example of confirmed booking view.

One of the main gripes and complaints from electric vehicle drivers is that there are not enough charging points in Ireland, or they do not know where to find public charge points if they drive to an area that is not familiar to them. To try assist renters, I wanted to provide data regarding charge points in the local area. After some investigation I found that openchargemap.io have a listing of registered charge points worldwide. Registration with them is free and they provide developers with access to their API. After some testing in Postman, I was able to tailor the API request to deliver a refined list that was relevant to potential renters. "Please see sample of the openchargemap.io API response in the Appendices."

Next, I needed to be able to map these charge points onto a map. I had some experience with Leaflets during the course work and decided this was the perfect way to display the charge points.

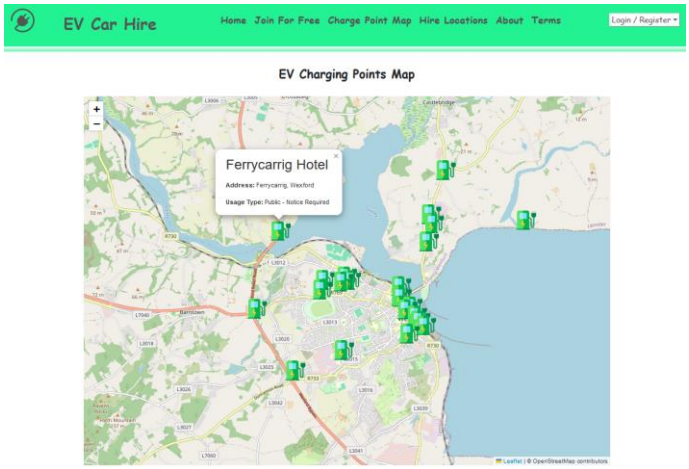


Figure 22 Leaflets Map with EV ChargePoint's

After implementing the charge points on the map, I thought it would also be a good idea to have a map that showed hire locations. The maps looked too busy when I had both the hire locations and charge points on the same map, so I created two map pages. The Hire location displays the location address and the numbers of cars currently in the fleet at that location. To display the number of cars in the location fleet we are checking the database to see the length of the array of cars available at that location.

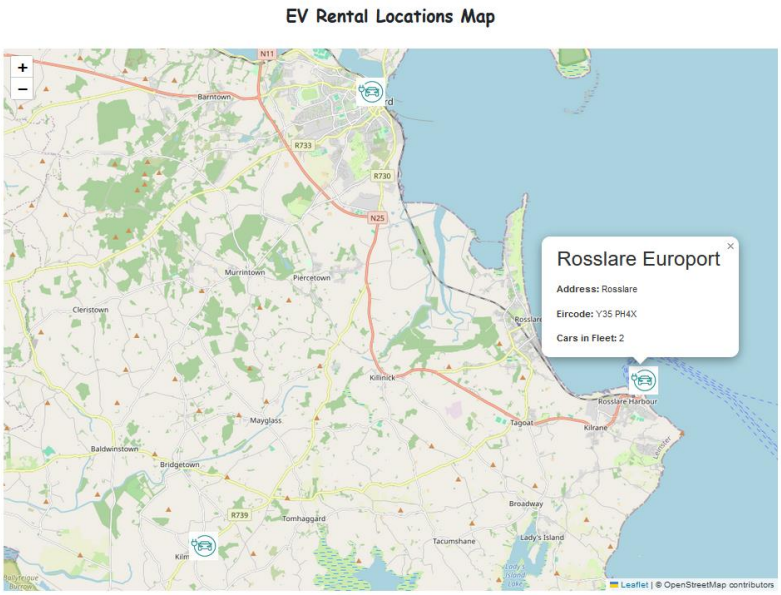


Figure 23 Leaflet Location Map

```
if (rentalLocations) {
  rentalLocations.forEach(location => {
    L.marker([location.latitude, location.longitude], { icon: rentalMarkerIcon })
      .addTo(map.current)
      .bindPopup(`
        <h3>${location.name}</h3>
        <p><strong>Address:</strong> ${location.town}</p>
        <p><strong>Eircode:</strong> ${location.eircode}</p>
        <p><strong>Cars in Fleet:</strong> ${location.carsAvailable.length}</p>
      `);
  });
}
```

Figure 24 Code to display Fleet at location on map.

The Vehicle Options display has four filters on the page. Filter by location, date, battery type and Car Name. They however did not work together and needed to be refactored. This was because in my initial build I did not have a location model, I only had cars. But when I added locations, I needed to reference the cars available at each location. This caused the existing filters to be confused. For example, if you selected a date span it would only show available cars for those dates but if you then selected a location all cars at the location were displayed regardless of the date filer. After a lot of trial and error I decided to disable the date filer until the location was selected. This meant that I needed to add some extra state variables to handle the different filters and a can book variable was created to check that all criteria were matched. I found this YouTube video very useful in achieving my goal with the location filter - (Lama Dev, 2022)

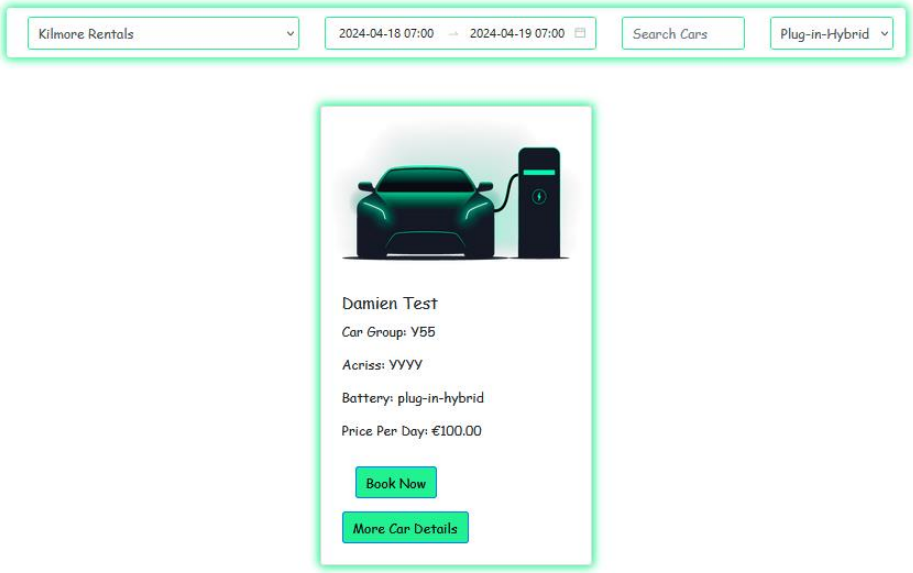


Figure 25 Example of filters applied to vehicle options.

```
function isCarAvailable(car, startDate, endDate) {
  if (car.currentbookings.length === 0) return true;

  for (const booking of car.currentbookings) {
    const bookingStartDate = moment(booking.pickupdate);
    const bookingEndDate = moment(booking.returndate);

    // Check if the selected date range overlaps with any booking
    if (
      (moment(startDate).isSameOrAfter(bookingStartDate) && moment(startDate).isBefore(bookingEndDate)) ||
      (moment(endDate).isAfter(bookingStartDate) && moment(endDate).isSameOrBefore(bookingEndDate)) ||
      (moment(startDate).isBefore(bookingStartDate) && moment(endDate).isAfter(bookingEndDate))
    ) {
      return false; // Car is not available for this date range
    }
  }
  return true; // Car is available for the selected date range
}
```

Figure 26 Code to check if car is available for selected date.

Sprint 5

I started to investigate how to send email from my application. I needed to do this for the password reset function, but I also wanted to add a contact us form and a booking confirmation email. I decided that Nodemailer was the best option to send smtp mail. The Nodemailer library simplifies the email-sending process in Node.js applications. Using npm, I installed the Nodemailer package, a straightforward process thanks to its popularity and documentation.

I created a free account with Zoho mail. To send email from my own domain I needed to verify domain ownership, so I registered the domain www.evcarhireireland.ie with Blacknight domain registry – Zoho mail provided smtp server name, port number and authentication details. Once configured I created a transport object within the Node.js application. With the aid of a YouTube tutorial about Nodemailer SMTP I was sending email from rentals@evcarhireireland.ie (OpenJavaScript, 2023).

```
// Setting up nodemailer transporter
const transporter = nodemailer.createTransport({
  host: "smtp.zoho.eu",
  port: 465,
  secure: true,
  auth: {
    user: process.env.ZOHO_EMAIL_HERE,
    pass: process.env.ZOHO_PASSWORD_HERE,
  },
});

// Email content
const mailOptions = {
  from: process.env.ZOHO_EMAIL_HERE,
  to: email,
  subject: "Reset Your Password",
  text: `Click the link to reset your password: ${resetLink}`,
};

// Sending the email
transporter.sendMail(mailOptions, function (error, info) {
  if (error) {
    console.log(error);
    return res.status(500).json({ status: "Error sending email" });
  } else {
    console.log("Email sent: " + info.response);
    return res.status(200).json({ status: "Email sent successfully" });
  }
});
```

Figure 27 Sample of code for sending email.

My attention now turned to building some new member only views, for this I used Ant Design. I created a profile page to display the member details and a list of their bookings. This was done using the tabs component feature from the Ant Design library. When a member makes a booking, they see the user-friendly confirmation message from sweet alert, a confirmation email with a summary of the bookings details is sent to the member and they are then redirected to the reservations tab in their profile page. From here they can cancel their reservation if required. The Booking Status will be updated in the database.

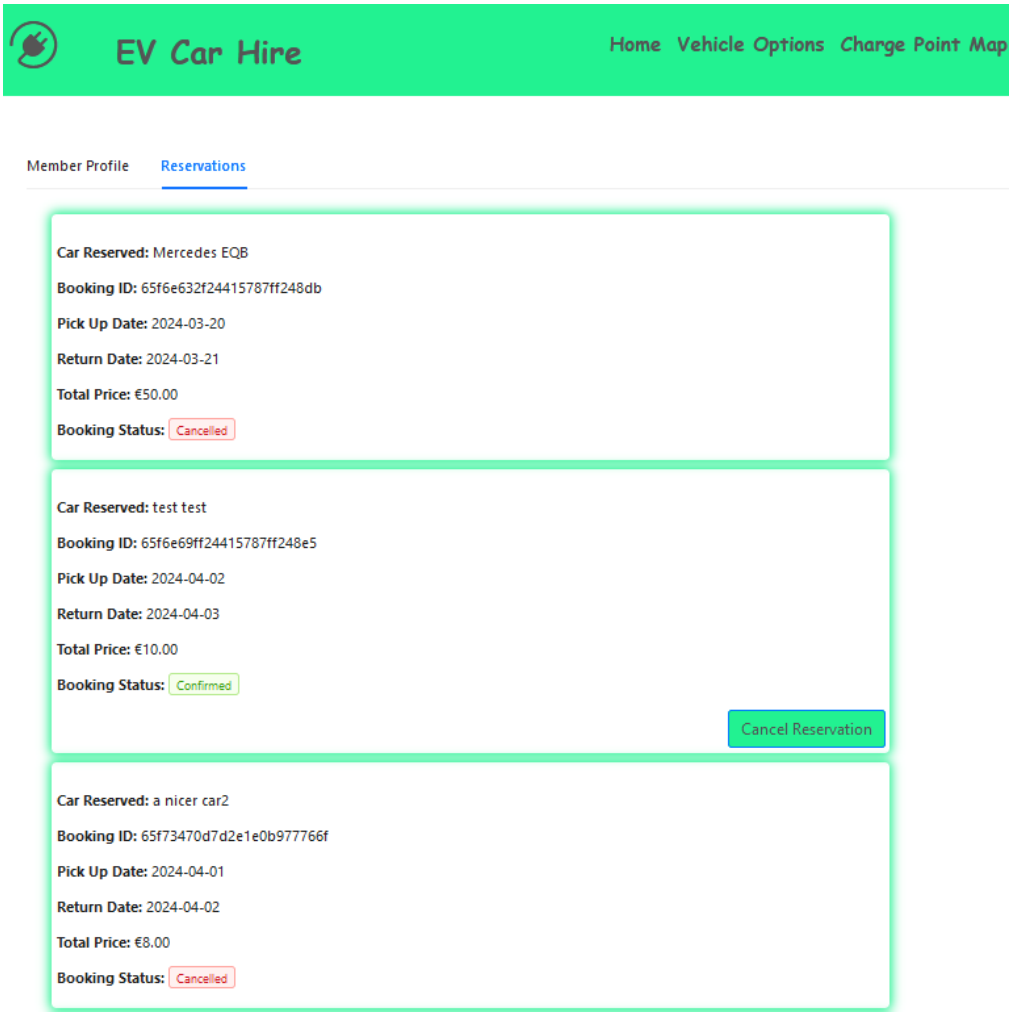


Figure 28 Member profile page.

Built using the same Ant Design library I developed an Admin section with the goal to create a detailed dashboard where admin users can manage various aspects of the application. A full list of all member bookings is displayed, admins can update car prices or delete existing car groups. They can add new cars and view and delete member profiles. The design is intuitive and responsive.

The admin dashboard prominently displays a list of all bookings in the database.

Figure 29 Admin dashboard – Reservation view.

Admin users have access to update car prices directly from the dashboard. They can also delete cars if no longer required. This is crucial for maintaining accurate pricing. Upon submission the changes are updated in real time in the database.

Figure 30 Admin dashboard – Cars update view.

The ability to add new fleet to the system is an essential feature of the admin section. The admin user is required to add seven fields, makeModel, carGroup, acriss, priceAmount, Location, images and batteryType.

The Location field looks up the database for associated locations so that the new car can be listed at a hire location. It pushes the car ID to the carsAvailable array of the associated location.

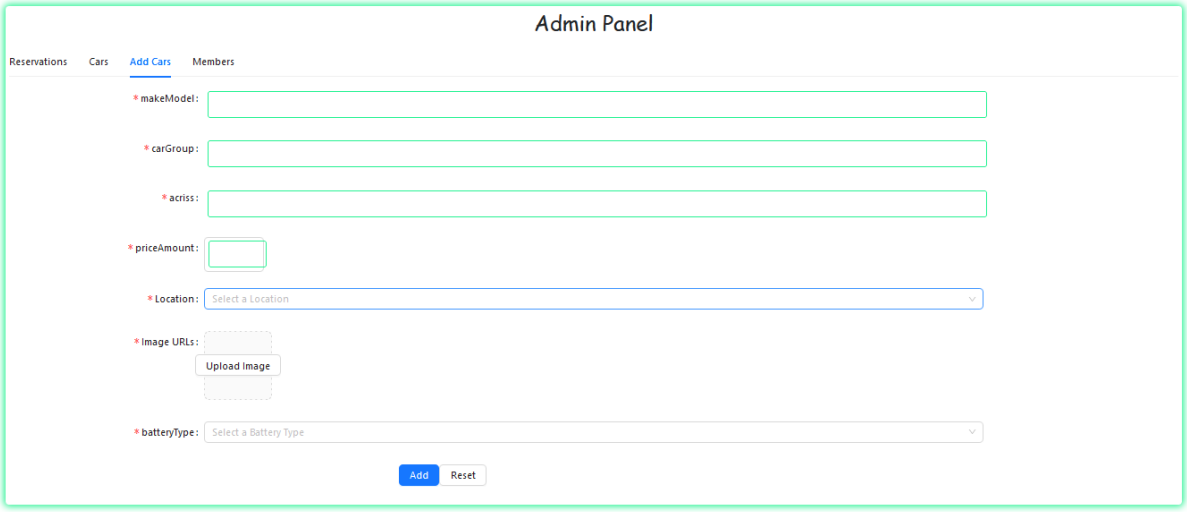


Figure 31 Admin dashboard – add new car view.

The form has validation rules to ensure correct input formats and required fields including at least one image are provided.

Figure 32 Admin dashboard – add new car form validation view.

One of the key features of the new car functionality is the integration of Cloudinary to upload and store images. Cloudinary is a cloud-based service that provides a solution to managing images. In the add car route the images are uploaded to the cloud, I wait for the secure URL of the image to be retrieved and then this secure URL is uploaded to the Mongo database. Multer middleware is used to process the files. The images are initially provided in a req.file that is converted into a base64 string and then uploaded to Cloundinary. To implement the required code for the upload function I followed the YouTube tutorial (Blakely, 2024).

```
// Upload images to Cloudinary
const uploadPromises = imageFiles.map(async (image) => {
  const b64 = Buffer.from(image.buffer).toString("base64");
  let dataURI = "data:" + image.mimetype + ";base64," + b64;
  const uploadedImage = await cloudinary.uploader.upload(dataURI, {
    folder: "evCarHire/",
  });
  return uploadedImage.secure_url;
});

const imageURLs = await Promise.all(uploadPromises);
```

Figure 33 Code for Cloudinary upload and URL retrieval.

The final Admin tab provide the ability to manage members. The admin can delete a member account if needed. There is a two-step process for deletion. The admin must click the delete button which requires confirmation before deletion. Ant design modal component offers a clear and user-friendly interface for the admin to action.

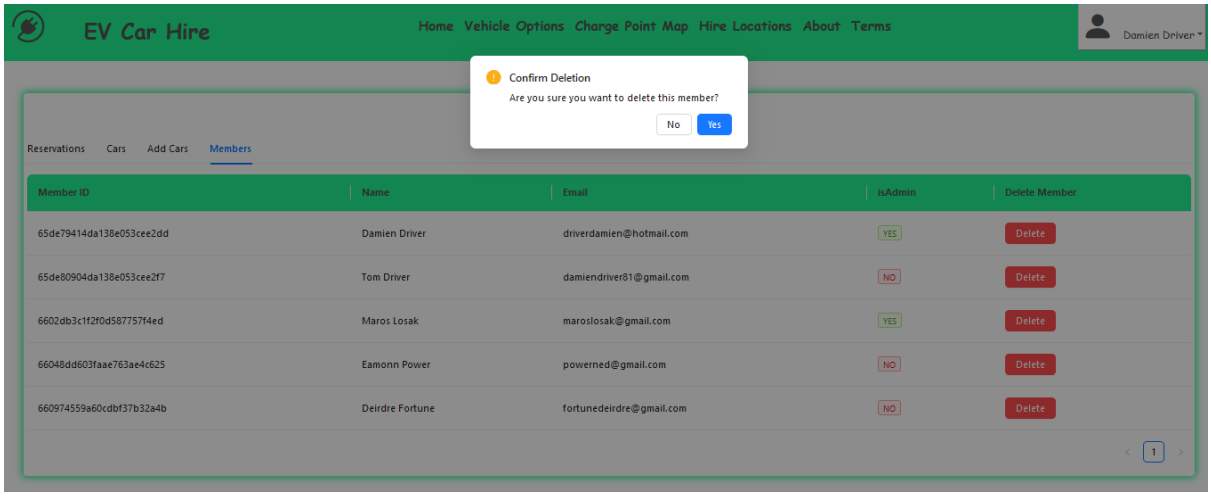


Figure 34 Admin dashboard – Member view.

At this stage of the development cycle, I felt that it was time to deploy the application. I wanted to have this done in plenty of time as previous experiences during the course program were that numerous things can you wrong when application moves from local environment to production.

I decided to deploy my application to the Vercel platform. Vercel's focus is on frontend deployment meaning that it is not the first choice for the backend deployment for most developers. However, I wanted to see if I could deploy both my frontend and backend on the same platform. Vercel deploys directly from Git repositories, so having my code in a Git repository was essential. The frontend and backend need to be set up as separate projects on the platform.

To facilitate the deployment, I created a Vercel json file in the root directory of the backend. This file is used to configure deployment settings for Vercel. I then added specific environment variables with necessary configurations for database connection strings into the Vercel project settings. Once deployed Vercel provides a URL for the backend component. I adapted a YouTube tutorial on Vercel backend deployment for my project, sourced from (Coding Garden, 2023).

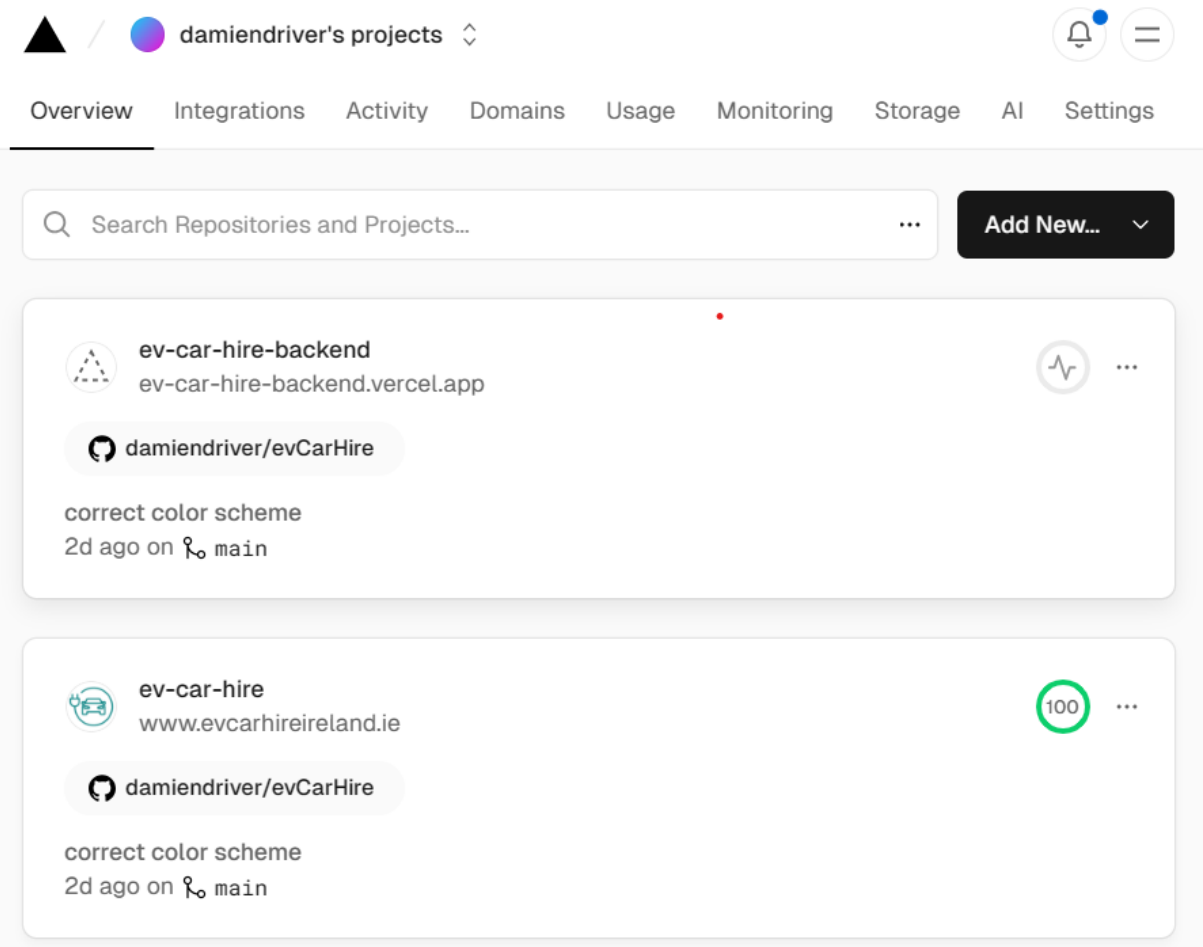


Figure 35 Vercel projects.

The frontend deployment is where Vercel excels. It is a very user-friendly system. You link up your GitHub repository in the Vercel dashboard. Trigger a deployment and Vercel will automatically build and deploy the frontend. Once deployed Vercel provides a unique URL for the frontend application. Vercel will redeploy your code on each push to your GitHub repository. The dashboard allows a custom domain to be configured. I pointed my application to www.evcarhireireland.ie and adjusted the domain name system (DNS) settings in my Blacknight account.

Some adjustments were need to the frontend code after deployment as the API routes stopped working. I needed to define the base URL for the Axios requests by adding the live backend URL to the request.

Sprint 6

The final sprint was all about fixing the little errors within the application. One example of this was the car price amounts did not have a currency indicator or uniformity of display throughout the application. To fix this I added a utils function that could be applied everywhere a price was displayed in the frontend of the application.

```
export const formatPrice = (price) => {
    return new Intl.NumberFormat("en-IE", {
      style: "currency",
      currency: "EUR"
    }).format(price);
  };
```

Figure 36 Price Format

A contact us form had been added in sprint 5 but did not work due to a bug where it was trying to send email from the members email address rather than from the application email address. This was corrected and the page reinstated.

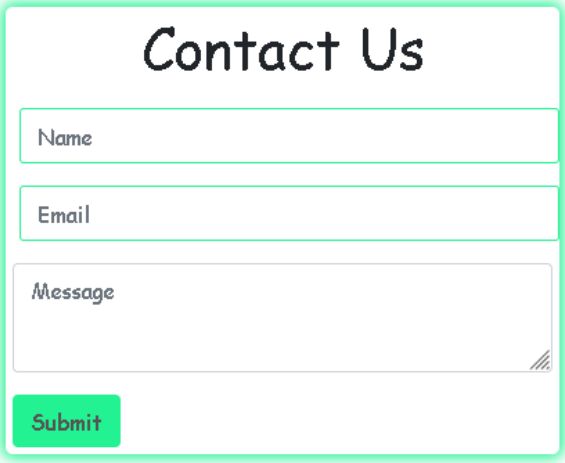


Figure 37 Contact Us Form

On the homepage I made some small modifications so that a logged in member saw a slightly different page and navbar than a user that was not a member. These were small details that I felt would improve the experience for a user and member alike. First the navbar options was updated on login. A new user would see the tab to “Join for Free” but a logged in member will see “Vehicle Options”.

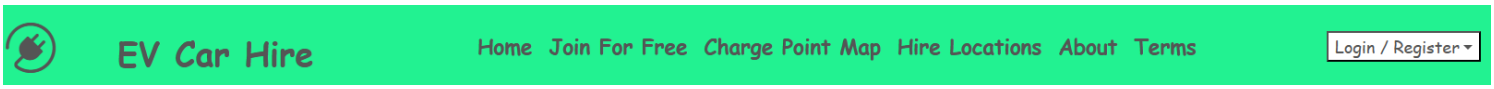


Figure 38 navbar logged out.

Figure 39 navbar member view.

The other amendment was to the home page message to “Join For Free” that a user see when landing on the homepage. This message was changed to “Vehicle Options” when a member logs in.

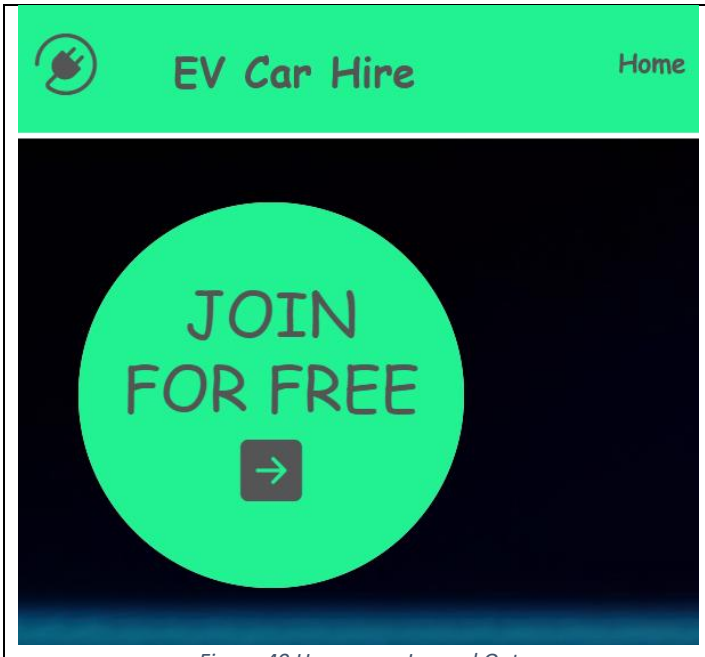


Figure 40 Homepage Logged Out

Figure 41 Homepage Member View

Future Development

Frontend

Some areas I would like to improve on if there was more time would be to allow members to modify their bookings in the application. Currently they can only cancel their booking. I would also add some rules to prevent bookings to be cancelled when the pickup time has passed.

I would improve the map functionality to allow the user to find the nearest charge point or hire location to their current location and provide directions.

I would also expand the homepage to provide more detail about cars available, hire locations and special offers.

Backend

I would like to provide full CRUD API function for all models in the database. Admin users would be provided with tools to update all aspects of the database.

I would document the API in Swagger. Swagger provides a standard formatting of API's making them easier to understand and use. It provides a reference guide to quickly view the available endpoints. Swagger also allows testing and validation of data improving the overall development process.

Finally, I would hope to build a comprehensive pricing system to allow for seasonality changes in price. Rather than having the same daily price 365 days of the year I would create a system where everyday could be priced differently if need. This would allow for seasonal demand and peak periods where demand creates an opportunity to yield better prices.

Conclusion

Developing this MERN (MongoDB, Express.js, React.js, Node.js) stack application has been an immensely rewarding journey. Frustrating at times but always providing new learning opportunities, and significant growth as a developer. I've gained a greater understanding of the full stack development process and valuable lessons in project management. At the start of this journey, I didn't know MERN was an established framework and here I am now with a functioning application.

The learning curve was steep at first, setting up the technologies and trying to understand the flow of data from the database to the frontend. This was my first experience going alone without class notes and lecturer guidance at every step of the journey. Through a lot of late nights, I have managed to enhance my knowledge and build an application that will give me a great foundation for future projects.

I now looking back see the value of the Trello boards and the RAMP phase of the development. I underestimated these at the time and saw them as unnecessary extra workload. The time spent on this phase can save hours down the line.

In conclusion, this project has been an invaluable experience that has improved my technical expertise and provided personal growth. I am proud of the functional application that has emerged from this learning experience, and I am excited to continue refactoring and expanding its capabilities in the future.

References

Adobe , 2024. *Adobe Color.* [Online]
Available at: https://color.adobe.com/create/color-contrast-analyzer
[Accessed January 2024].

Blakely, C., 2024. *Complete MERN Stack Project: Build a Hotel Booking App Like a Pro Developer Step-by-Step Course 2024.* [Online]
Available at: https://www.youtube.com/watch?v=YdBy9-0pER4
[Accessed March 2024].

Coding Garden, 2023. *Deploy an Express API to Vercel.* [Online]
Available at: https://www.youtube.com/watch?v=B-T69_VP2Ls
[Accessed March 2024].

Hertz Rent A Car, 2024. *Ireland Car Hire and Van Rental | Hertz Rent A Car.* [Online]
Available at: https://www.hertz.com/rentacar/reservation/#vehicles
[Accessed January 2024].

Hiren, 2021. *Express vs Hapi — Choosing the Best Node.js Framework.* [Online]
Available at: https://medium.com/@hiren_81825/express-vs-hapi-choosing-the-best-node-js-framework-5f151f148e0d
[Accessed February 2024].

Inflectra, 2024. *Agile Scrum Methodology: What To Know | Inflectra.* [Online]
Available at: https://www.inflectra.com/Solutions/Methodologies/Scrum.aspx
[Accessed February 2024].

Jayabal, A., 2024. *NextJS vs ExpressJS: Which JavaScript Framework To Choose?.* [Online]
Available at: https://www.mirrorfly.com/blog/nextjs-vs-expressjs/
[Accessed February 2024].

Joshi, M., 2023. *Angular vs React vs Vue: Core Differences | BrowserStack.* [Online]
Available at: https://www.browserstack.com/guide/angular-vs-react-vs-vue
[Accessed February 2024].

Lama Dev, 2022. *React Node.js Booking App Full Tutorial | MERN Stack Reservation App (JWT, Cookies, Context API).* [Online]
Available at: https://www.youtube.com/watch?v=k3Vfj-e1Ma4&t=18s
[Accessed March 2024].

Mark Otto, Jacob Thornton, and Bootstrap contributors, 2024. *Bootstrap · The most popular HTML, CSS, and JS library in the world..* [Online]
Available at: https://getbootstrap.com/
[Accessed February 2024].

Next Code, 2022. *Hotel Booking System React.* [Online]
Available at: https://www.youtube.com/playlist?list=PLflVqQLAWzC9qybF40gPXpzntMQIoYG5u
[Accessed Feb 2024].

Nowak, M., 2023. *Vue vs. React in 2023 - Comparison of Two Most Popular JS Frameworks.* [Online]
Available at: https://www.monterail.com/blog/vue-vs-react
[Accessed February 2024].

OpenJavaScript, 2023. *Send Emails in Node.js | NodeMailer Tutorial.* [Online]
Available at: https://www.youtube.com/watch?v=L46FwfVTRE0
[Accessed March 2024].

Patel, J., 2024. *Django Vs Express: The Key Differences To Observe in 2024.* [Online]
Available at: https://www.monocubed.com/blog/django-vs-express/
[Accessed February 2024].

React, 2024. *React.* [Online]
Available at: https://react.dev/
[Accessed February 2024].

Twilio, 2024. *React vs. Svelte: Comparing the Basics.* [Online]
Available at: https://www.twilio.com/en-us/blog/react-svelte-comparing-basics
[Accessed February 2024].

Vercel, 2024. *Vercel.* [Online]
Available at: https://vercel.com/
[Accessed 2024].

Web Dev Creative, 2023. *How To Create a Responsive Navbar Using HTML & CSS | Step By Step Tutorial.* [Online]
Available at: https://www.youtube.com/watch?v=GdrbE-s5DgQ
[Accessed January 2024].

Zanini, A., 2023. *https://blog.logrocket.com/express-js-adoption-guide/.* [Online]
Available at: https://blog.logrocket.com/express-js-adoption-guide/
[Accessed February 2024].

Appendix A:

Sample Response from https://api.openchargemap.io:


```
[
  {
    "DataProvider": {
      "WebsiteURL": "http://openchargemap.org",
      "Comments": null,
      "DataProviderStatusType": {
        "IsProviderEnabled": true,
        "ID": 1,
        "Title": "Manual Data Entry"
      },
      "IsRestrictedEdit": false,
      "IsOpenDataLicensed": true,
      "IsApprovedImport": true,
      "License": "Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0)",
      "DateLastImported": null,
      "ID": 1,
      "Title": "Open Charge Map Contributors"
    },
    "OperatorInfo": {
      "WebsiteURL": "https://easygo.ie/",
      "Comments": null,
      "PhonePrimaryContact": null,
      "PhoneSecondaryContact": null,
      "IsPrivateIndividual": false,
      "AddressInfo": null,
      "BookingURL": null,
      "ContactEmail": null,
      "FaultReportEmail": null,
      "IsRestrictedEdit": false,
      "ID": 1241,
      "Title": "EasyGO (aka Carcharger.ie)"
    },
    "UsageType": {
      "IsPayAtLocation": false,
      "IsMembershipRequired": true,
      "IsAccessKeyRequired": true,
      "ID": 4,
      "Title": "Public - Membership Required"
    },
    "StatusType": {
      "IsOperational": true,
      "IsUserSelectable": true,
      "ID": 50,
      "Title": "Operational"
    },
    "SubmissionStatus": {
      "IsLive": true,
```

```
    "ID": 200,
    "Title": "Submission Published"
},
"UserComments": null,
"PercentageSimilarity": null,
"MediaItems": null,
"IsRecentlyVerified": false,
"DateLastVerified": "2023-09-06T15:58:00Z",
"ID": 274274,
"UUID": "E8D01DDC-C1C2-4EF6-B2C6-156699C6F22F",
"ParentChargePointID": null,
"DataProviderID": 1,
"DataProvidersReference": null,
"OperatorID": 1241,
"OperatorsReference": null,
"UsageTypeID": 4,
"UsageCost": null,
"AddressInfo": {
    "ID": 274658,
    "Title": "Bolands",
    "AddressLine1": "Ferrybank",
    "AddressLine2": null,
    "Town": "Wexford",
    "StateOrProvince": "Wexford",
    "Postcode": null,
    "CountryID": 3,
    "Country": {
        "ISOCode": "IE",
        "ContinentCode": "EU",
        "ID": 3,
        "Title": "Ireland"
    },
    "Latitude": 52.35185795835778,
    "Longitude": -6.454382086886881,
    "ContactTelephone1": null,
    "ContactTelephone2": null,
    "ContactEmail": null,
    "AccessComments": null,
    "RelatedURL": null,
    "Distance": 0.12029974832508333,
    "DistanceUnit": 1
},
"Connections": [
    {
        "ID": 475394,
        "ConnectionTypeID": 25,
        "ConnectionType": {
            "FormalName": "IEC 62196-2 Type 2",
            "IsDiscontinued": false,
            "IsObsolete": false,
```

```
            "ID": 25,
            "Title": "Type 2 (Socket Only)"
         },
         "Reference": null,
         "StatusTypeID": 50,
         "StatusType": {
            "IsOperational": true,
            "IsUserSelectable": true,
            "ID": 50,
            "Title": "Operational"
         },
         "LevelID": 2,
         "Level": {
            "Comments": "Over 2 kW, usually non-domestic socket type",
            "IsFastChargeCapable": false,
            "ID": 2,
            "Title": "Level 2 : Medium (Over 2kW)"
         },
         "Amps": null,
         "Voltage": null,
         "PowerKW": 22,
         "CurrentTypeID": 20,
         "CurrentType": {
            "Description": "Alternating Current - Three Phase",
            "ID": 20,
            "Title": "AC (Three-Phase)"
         },
         "Quantity": 4,
         "Comments": null
      }
   ],
   "NumberOfPoints": 4,
   "GeneralComments": null,
   "DatePlanned": null,
   "DateLastConfirmed": null,
   "StatusTypeID": 50,
   "DateLastStatusUpdate": "2023-09-06T15:58:00Z",
   "MetadataValues": null,
   "DataQualityLevel": 1,
   "DateCreated": "2023-09-06T15:58:00Z",
   "SubmissionStatusTypeID": 200
}]
```

Appendix A:

Ethics Checklist

This Ethics Checklist must be completed for all final year undergraduate, taught postgraduate and research projects in the School of Science and Computing.

View your response(s)

Respondent: **Damien Driver** (Group: CM-HDIPCS) Submitted on: Saturday, 18 November 2023, 4:32 PM

Ethics Checklist for Undergraduate, Taught Postgraduate and Research Projects in the School of Science and Computing

All students in the School of Science and Computing who are either (1) in the final year of an undergraduate/BSc degree, or (2) on a taught postgraduate/MSc programme **must complete this Ethics Checklist before conducting their project** regardless of the project type or discipline. The Checklist should also be completed by anyone (whether staff member or student) conducting a **research project** (whether programmatic or not) within the School.

The purpose of this Ethics Checklist is to **identify projects that will require formal ethical approval** from the School Research Ethics Committee, or the SETU Research Ethics Committee, before they can proceed.

Students/applicants should note that this Ethics Checklist is a **formal declaration**, and great care must be taken to **answer all questions accurately**. Students should consult with their project supervisors/advisors regarding any aspects or questions that they are unsure of before completing and submitting the Ethics Checklist.

Students/applicants must **answer all questions** presented to them until the Checklist questionnaire is completed.

Feedback Report

No human experimentation issues (TPG/RP).

No animal experimentation issues (N/A).

No issues regarding the use of human tissues.

No animal tissue or biological fluids issues.

No ionising radiation issues.

No primary data collection issues (N/A).

?

No underage/vulnerable people issues (TPG/RP).

No issues regarding existing/secondary data use (N/A).

No controversial data issues.

No issues related to the collection of rare or protected plants.

No issues regarding the use of genetically modified (GM) plant material.

Instructions:

1. If the above feedback is **entirely green** then, based on your answers, there is **no need to apply for ethical approval** for your project.
2. If **any** part of the above feedback is **yellow/amber**, then there is at least one issue with your project that needs to be reviewed and **you must apply for ethical approval** to continue your project.
3. If **any** part of the above feedback is **red** then there is a serious ethical issue and **you cannot continue your project** as currently planned.

It is recommended that you print this Feedback Report to a PDF file for your records. You should also forward and discuss this Feedback Report PDF with your project supervisor. They will be able to advise if you have any further questions or if you need to apply for ethical approval.

1 * Are you a student on a **final year undergraduate** programme, a **taught postgraduate** programme, or are you conducting a **research project**?

- ○ Final Year Undergraduate
- ◉ Taught Postgraduate
- ○ Postgraduate Research Project
- ○ Other Research Project

2 * What is the **working title** of your project?

EV Car Hire Reservation Web Application

3 * Who are the project **supervisors/advisors/principal investigators**?

Colm Dunphy

4 * Does your project involve **human experimentation**?

○ Yes ◉ No

5 * Does your project involve **live animal experimentation**?

○ Yes ◉ No

(6) * Is the planned animal experimentation limited to **non-invasive procedures only** (such as feeding, weighing, or taking naturally voided faecal or hair samples), and does **not** involve any invasive procedures (such as taking rectal faecal samples or blood) from live animals?

○ Yes ○ No

7 * Does your project involve the use of **human** remains/cadavers/tissues/cells/biological fluids/embryos/foetuses?

○ Yes ◉ No

(8) * Do you intend to only use established **commercial human cell lines**, and no other **human** remains/cadavers/tissues/cells/biological fluids/embryos/foetuses in your project?

○ Yes ○ No

9 * Does your project involve the use of **animal cells, tissues** or **biological fluids**?

○ Yes ◉ No

(10) * Do you intend to only use (1) **established commercial animal cell lines**, or (2) **slaughterhouse-derived tissues/fluids**, or (3) **fluids collected as part of routine animal husbandry** (e.g. milk) and no other animal tissues or biological fluids in your project?

○ Yes ○ No

11 * Does your project involve the **collection of rare or protected plants**?

○ Yes ◉ No

12 * Does your project involve the generation or use of **genetically modified (GM) plant material**?

○ Yes ◉ No

(13) * Do you agree to (1) only use **established genetically modified (GM) plant cell lines, seeds, or plant products** in your project, (2) **not generate new plant mutations** using chemical or other means, and (3) follow specified SETU **containment and use protocols** for GM plant materials at all times?

○ Yes ○ No

14 * Does your project involve the use of **ionising radiation**? (e.g. use of gamma ray spectrometry)

○ Yes ◉ No

(15) * Do you agree to carefully **follow the instructions** of the SETU designated **Radiation Protection Officer (RPO)**, and **adhere to all legal requirements** as set out in the Radiological Protection Act 1991 (Ionising Radiation) Regulations (2019), regarding the use of ionising radiation materials and equipment?

○ Yes ○ No

16 * Does your project involve the **collection of any new (or primary) data** from **individual people or groups**?

○ Yes ◉ No

(17) * Does your project involve the **collection of any new (or primary) individual or group data** that is **personally or uniquely identifying**? (e.g. data about people or organisations/companies/groups that could be used to identify those individuals or groups; data collection might take any form, including internet and social media data, etc.)

○ Yes ○ No

(18) * Will you ensure that participants who you are collecting data from are provided with **fair warning** and must provide **explicit informed consent** for any data collected?

Yes No

(19) * Will you ensure that any project-related data collection, data storage, and data use is in **full compliance** with the **EU General Data Protection Regulation (GDPR)** and the **Data Protection Act (2018)**?

Yes No

(20) * Does any of the data that you intend to collect include **sensitive or private personal information** about individuals, or **commercially sensitive information** about organisations/companies/groups?

Yes No

21 * Does your project involve **persons under the age of 18 years** (i.e. minors), or **any vulnerable groups**? (e.g. prisoners, refugees, those in care, addiction service users, etc.)

Yes No

22 * Does your project involve the use of **existing (or secondary) human data**? (i.e. data originally collected for another purpose)

Yes No

(23) * Is the existing or secondary human data you intend to use either (1) **anonymous/non-personally identifying** and in the **public domain**, or (2) available with **explicit and specific informed consent or permission** for the data to be **legally** reused in the way you intend?

Yes No

(24) * Are any aspects of the primary/secondary data you intend to use for the project **controversial** in nature?

Yes No

25 * Before you submit the Ethics Checklist, you must **confirm all of the following**:

- I understand that the Ethics Checklist is a formal declaration.
- I have answered all questions on the Ethics Checklist carefully and truthfully.
- The supervisor/advisor (or principal investigator) for the project is present as the Ethics Checklist is being submitted, or they have given me explicit permission to submit it in their absence.
- I have had adequate ethics training and/or instruction prior to completing the Ethics Checklist.
- I understand, and agree to abide by, the general ethical principle of "do no harm" for this project.
- I will follow the instructions given in the Feedback Report.

26 * Authentication Code (ask your project supervisor/advisor for this code)

Enter Student Number:

Enter the Authentication Code below and click "Verify Code" Verify Code

Note: If an INVALID authentication code is used then this submission is NULL and VOID

4253