Awarding Body: Arden University
Programme Name: BSC in BSc (Hons) Computing
Module Name (and Part if applicable): RES6011 - Computing Project
Assessment Title: A Web-Based Dashboard for Promoting Sustainable Urban Mobility Through E-Bike Data Visualisation
Student Number: STU105154
Tutor Name: Dr. Sharjeel Aslam
Word Count: Total word count 14,224 Word Count (excluding references and appendices): 10,599 Please refer to the Word Count Policy on your Module Page for guidance

Abstract

This dissertation investigates how a browser-based dashboard can be used to show e-bike route data in a way that helps people understand how to get around cities in a more environmentally friendly way. The project seeks to make an interactive, easy-to-use application that shows complicated data in a way that is easy for non-technical users to understand and enjoy. This is because more and more open transport datasets are becoming available and low-carbon mobility is becoming more and more important.

The artefact was made utilising front-end web technologies like HTML, CSS, JavaScript, and Chart.js. The focus was on making it simple, easy to use, and able to work without an internet connection. We got public datasets from Citi Bike NYC and Transport for London in an ethical way, cleaned them up, and turned them into structured JSON files that are best for client-side rendering. The dashboard lets customers see how many trips happen by hour and day, and it lets them compare patterns between casual users and subscribers.

The project used a mixed-methods evaluation strategy that included both structured questionnaires and light user testing. Five people with both technical and non-technical backgrounds were asked to use the dashboard to do activities and then think about what they had done. The results showed that people understood the dashboard well, found it easy to use, and thought it was valuable. This confirmed that the dashboard satisfies its functional and instructional purposes.

This initiative adds to the conversation around digital sustainability tools by using open data reuse, inclusive design, and storytelling that is meant for the public. It shows that civic technology may be both useful from a technical point of view and important from a social point of view. It also gives a model for how to easily get involved with environmental data. Data visualisation, e-bikes, civic dashboards, web development, and accessibility are some of the keywords.

Acknowledgements

I want to thank my boss, Mr. Sharjeel Aslam, for all the help, advice, and support he gave me during this assignment. His advice helped me improve both the theoretical and practical parts of my career.

I also like to thank the people that took part in the user testing. Their time and comments were very helpful in making the dashboard more user-friendly and welcoming to everyone. I want to thank Arden University and its teachers for giving me the help, tools, and space to learn that made this endeavour possible. The skills and information I received from this degree have been helpful in finishing this assignment.

I also admit to using AI support through OpenAI's ChatGPT tool, which I used to help me organise and format the dissertation, improve my academic language, and check my citation style. I wrote all the content and analysis myself, and AI was only used for editing and help in a way that follows academic integrity rules. OpenAI's ChatGPT was used to help with language, syntax, and formatting, but all the research, analysis, and academic findings are the student's own work. Using AI tools was done in a way that respected ethical rules and didn't take the role of critical thinking or personal input.

Lastly, I want to thank my family and friends for being patient, encouraging, and helpful throughout the way.

Contents

List of Acronyms

Acronym	Full Form
API	Application Programming Interface
ARIMA	AutoRegressive Integrated Moving Average
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
DOM	Document Object Model
GDPR	General Data Protection Regulation
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
JS	JavaScript
LCP	Largest Contentful Paint
SPA	Single Page Application
UI	User Interface
UX	User Experience
W3C	World Wide Web Consortium
WCAG	Web Content Accessibility Guidelines

List of Figures

Chapter 1: Introduction

1.1 Chapter Introduction

In this chapter, I introduce the research project and explain its relevance within the broader context of sustainable mobility and data visualisation. I begin by outlining the background and context that motivated my investigation, followed by a clearly defined problem statement. I then present the aim of the project and the specific objectives I set out to achieve. Next, I pose the central research question that guided my work and explain the scope of the project, detailing what was included and what lay beyond its boundaries. I also describe my approach to managing the project over time, supported by a Gantt chart, and conclude with a summary of how this dissertation is structured across the remaining chapters.

1.2 Background and Context

As cities globally struggle with climate change, traffic, and pollution, sustainable urban mobility is needed. E-bikes are a promising answer in this area. For short-distance urban transport, e-bikes are low-carbon and efficient. Their use is rising, especially in cities with inadequate public transport. European Cyclists' Federation (2021) estimates that e-bikes might reduce urban transport emissions by 50% by replacing even a percentage of short automobile trips. This statistic shows that e-bikes can benefit urban mobility, public health, and the environment.

E-bikes are beneficial, but many people don't understand how they work, especially in shared mobility programs. E-bike usage data is becoming more available, but it's sometimes too technical or poorly presented for the public to connect with. This limits the public's ability to make educated decisions or support infrastructure improvements. I believe better mobility data presentation is crucial to public awareness and behaviour change. My project investigates how web-based dashboards may simplify, engage, and make complicated transport datasets clear and accessible to non-technical consumers.

1.3 Problem Statement

The core problem that I tackle in this project is the disconnect between the growing availability of open transport datasets and the public's ability to engage with and interpret these datasets effectively. While cities like London and New York offer rich, anonymised data on shared e-bike usage, these datasets are typically presented in raw CSV formats or visualised through institutional dashboards designed for policymakers and analysts. In my experience, such platforms often fail to consider the needs of everyday users—people who might be curious about trends in mobility but who lack the technical skills to query large datasets or interpret complex charts.

I argue that without intuitive and inclusive visual tools, much of the value of open transport data remains untapped. The failure to engage citizens with these resources not only limits transparency but also weakens public participation in conversations about sustainable transport. I see this as both a technical and a civic challenge. My project seeks to close this gap by creating a web-based dashboard that translates e-bike journey data into a form that is educational, easy to use, and publicly accessible.

1.4 Project Aim and Objectives

This project will provide an accessible, browser-based dashboard that visualises public e-bike data to promote sustainable urban mobility. Design, ethics, and user experience must be considered alongside technical development. The dashboard should help users understand mobility trends by connecting complex data to public awareness.

I set the following goals to achieve this. Citi Bike NYC and Transport for London provided e-bike journey data, which I cleansed. I then formatted this data for browser-based rendering utilising client-side processing to maximise accessibility. My second goal was to create a lightweight dashboard using HTML, CSS, JavaScript, and Chart.js. I used Web Content Accessibility Guidelines (W3C, 2018) to make the interface simple and accessible for users of all abilities. My third goal was to test the dashboard with technical and non-technical users. This helped me evaluate the dashboard's usability and informativeness and guide design revisions.

1.5 Research Questions

The central research question guiding this project is: *How can a web-based dashboard be used to visualise e-bike data in a way that enhances public understanding of sustainable urban mobility?* This question reflects my goal of transforming complex data into accessible insights.

A related sub-question that I explored during the development and testing phases was: *What design strategies are most effective in helping non-technical users interpret urban transport data visualisations?* This prompted me to think critically about visual design, accessibility, and user interface structure, and it shaped the technical and evaluative aspects of the dashboard.

1.6 Scope of the Project

This project focuses on visualising open datasets from two cities—New York and London—because they provide comprehensive and publicly accessible e-bike journey data. The scope is limited to front-end development, meaning the dashboard does not rely on server-side processing or external databases. Instead, all data is pre-processed and embedded in the application in JSON format to allow offline use and broader device compatibility.

While I considered including geospatial maps or predictive analytics, I decided to focus on time-based patterns such as trip frequency by hour and day, and usage comparisons between different rider types. This narrower scope allowed me to refine the core features and ensure the dashboard remained lightweight and accessible. Features such as real-time data streaming or user personalisation were deemed out of scope due to time constraints and the complexity they would introduce.

1.7 Project Management and Planning

To manage the project effectively, I adopted a phased approach that included planning, data collection, design, implementation, and evaluation. Early on, I created a Gantt chart to visualise the timeline and identify task dependencies. This tool was crucial in keeping me on track and ensuring that I allocated time appropriately across the stages of the project. The planning phase involved refining my research questions and conducting a thorough literature review. The data collection and cleaning stage required significant attention, as I had to ensure data quality and GDPR compliance. Implementation was iterative, with testing and feedback guiding each design update.

I used Trello as my task management tool to organise deliverables and track progress. I also maintained a GitHub repository for version control, which allowed me to roll back changes when needed and document my development process. Although I did not use a full Agile workflow, I drew on Agile principles by working in short sprints and prioritising user feedback during each round of refinement. These tools and techniques helped me manage a complex project with limited resources and provided a structure that supported both creativity and discipline.

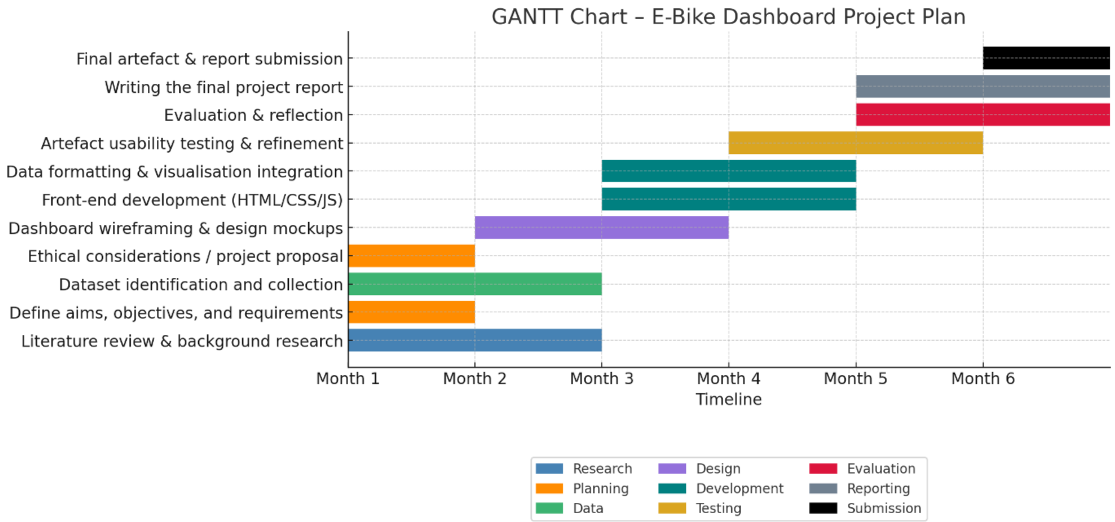


Figure 1 - STU105154 - Gantt Chart: E-Bike Dashboard Project Plan - This Gantt chart outlines the full project timeline across six months, detailing research, design, development, testing, reporting, and submission phases.

1.8 Structure of the Dissertation

The central research question guiding this project is: *How can a web-based dashboard be used to visualise e-bike data in a way that enhances public understanding of sustainable urban mobility?* This question reflects my goal of transforming complex data into accessible insights.

A related sub-question that I explored during the development and testing phases was: *What design strategies are most effective in helping non-technical users interpret urban transport data visualisations?* This prompted me to think critically about visual design, accessibility, and user interface structure, and it shaped the technical and evaluative aspects of the dashboard.

1.9 Chapter Summary

This chapter has introduced the context, purpose, and structure of my project on visualising e-bike data to promote sustainable mobility. I have explained the societal relevance of the topic, identified a clear gap in how mobility data is currently shared with the public, and outlined how my dashboard seeks to address this. By presenting both the aims and the constraints of the project, I have provided a roadmap for what follows. This foundation supports a research process that is both technically robust and socially engaged. In the next chapter, I turn to the academic literature that informed my work and explore how it shaped my design decisions and research focus.

Chapter 2: Literature Review

2.1 Chapter Introduction

In this chapter, I critically explore the literature surrounding sustainable urban mobility, the role of open data in transport systems, and the technologies commonly used for data visualisation in public-facing platforms. My objective is to understand how researchers and developers have addressed these topics, while identifying the gaps that my project seeks to fill. This includes analysing both the technological and social aspects of visualising urban e-bike data, particularly in a way that supports broader accessibility and public engagement.

2.2 Key Themes and Topics

Sustainable urban mobility has become a major concern in modern transport planning, with e-bikes increasingly seen as a practical alternative to cars for short-distance travel. I found that researchers such as Fishman (2016) argue that the growth of e-bike use has the potential to "revolutionise personal transport in cities," providing both environmental benefits and greater inclusivity. Similarly, Pucher et al. (2010) highlight the positive role of cycling in improving public health and reducing emissions. However, most studies on e-bike mobility are directed at policymakers or researchers, with less attention paid to how everyday citizens can engage with the data that supports such initiatives.

I also observed that many urban authorities now publish open transport data as part of their transparency goals. Platforms like Citi Bike in New York and Santander Cycles in London provide detailed, downloadable usage records. According to Oshan et al. (2020), these datasets have become a "rich foundation for understanding travel behaviour in urban contexts." However, their use has mostly been confined to technical domains, such as transport modelling, clustering, or machine learning applications. These tools, while powerful, do not necessarily empower non-specialist users to interpret the patterns themselves.

Another common theme in the literature relates to the importance of accessibility in digital interfaces. The World Wide Web Consortium's WCAG 2.1 guidelines (W3C, 2018) make it clear that interactive applications should support features such as high-contrast colour schemes, keyboard access, and screen-reader compatibility. However, visualisation tools often fall short in this regard. For instance, Strobelt et al. (2016) argue that "interactive graphics frequently fail to serve blind or visually impaired users," highlighting a gap between design best practices and real-world implementation. I therefore recognised a strong need to not only visualise e-bike data, but to do so in a way that invites interaction from diverse users, including those who rely on assistive technologies.

2.3 Comparison of Tools and Technologies

In reviewing available tools for web-based visualisation, I found significant variation in complexity, flexibility, and suitability for my project's goals. D3.js, for example, is highly customisable and often used for advanced visual storytelling. Bostock (2012), who created the library, described it as "a means of binding data to the DOM," which offers granular control but demands extensive coding knowledge. I considered using D3.js but realised that it would complicate both development and maintenance for a lightweight dashboard.

In contrast, Chart.js offers a simpler, cleaner syntax and requires no backend infrastructure. According to Schwartz (2020), Chart.js is "ideal for educational, civic, or public projects where ease of use and quick deployment are priorities." In my case, its mobile responsiveness, canvas rendering, and native support for dynamic updates made it a better match. I also reviewed alternatives like Plotly and Google Charts, but found that many of them either relied on external APIs or carried performance overheads not suitable for purely client-side deployment.

What stood out to me in the literature was the lack of discussion around accessibility in these tools. Yu et al. (2021) conducted comparative usability testing and concluded that while tools like D3.js offer customisation, libraries like Chart.js "outperform in responsiveness and clarity on mobile devices," which was essential for my intended audience. I also noted that very few libraries integrate accessibility features by default, reinforcing my decision to build accessibility into the dashboard manually.

2.4 Gaps in the Literature

Although I found extensive work on mobility data analysis, I noticed a surprising lack of research on how to present such data to the public in a usable, inclusive format. Most studies focus on predictive modelling, route optimisation, or policy simulation (Zhou et al., 2018), yet there is little attention paid to how ordinary people might benefit from seeing patterns in urban cycling data.

In particular, I found that existing dashboards and mobility platforms are either commercial products or internal tools for government agencies. Few, if any, offer public-friendly, browser-based solutions that visualise trends without requiring login, plugins, or technical expertise. This gap was clearly described by Hullman et al. (2011), who argued that "the purpose of visualisation should not just be insight, but communication." Their view aligned with my goal of making transport data transparent, approachable, and informative for general users.

Moreover, although accessibility is widely mentioned in design theory, I noticed a lack of empirical evaluation in visualisation literature. Most implementations do not address screen reader compatibility, alternative text, or keyboard-only navigation. I therefore identified a clear opportunity to address these limitations in my project by applying WCAG principles directly to the dashboard's architecture and interface.

2.5 Summary of Findings

In summarising the literature, I found that while urban mobility research is rich in data and tools, it lacks a focus on public engagement through accessible, visual platforms. Researchers often explore advanced modelling techniques or infrastructure planning, but overlook the value of helping non-experts interpret mobility data. I also discovered that although technologies like Chart.js are technically capable, they must be supported by intentional design practices to ensure usability and inclusivity.

My project builds on these findings by offering a practical response to the gaps I identified. By using open datasets, lightweight frontend technologies, and accessibility-aware design, I created a dashboard that allows users to explore e-bike usage in ways that are informative, inclusive, and contextually relevant. In doing so, I hope to contribute not only to academic discussion but also to broader efforts in promoting sustainable transport through digital tools.

2.6 Chapter Summary

In this chapter, I critically reviewed key literature relating to sustainable urban mobility, open transport data, and data visualisation technologies. I began by exploring how e-bike systems have been embraced as a solution to environmental and social challenges in cities, but I also recognised that much of the literature is directed at policy and research audiences rather than the general public.

I examined how platforms like Citi Bike provide rich datasets, yet are rarely used to develop tools aimed at increasing public understanding. This reinforced my motivation to create a dashboard that serves a broader audience, particularly those without a technical background.

In comparing visualisation tools, I found that while libraries such as D3.js offer depth, tools like Chart.js are more practical for my lightweight, browser-based design. My review also highlighted a serious lack of accessibility consideration in many public-facing systems.

Overall, the literature confirmed that there is a clear opportunity to contribute something new—an inclusive, mobile-friendly, and easy-to-use dashboard that helps everyday people make sense of urban cycling data. These insights have strongly shaped both the design and implementation of my project, which I explain in more detail in the chapters that follow.

Chapter 3: Research Methodology

3.1 Chapter Introduction

In this chapter, I explain the methodology that guided the design, development, and evaluation of my web-based dashboard for visualising e-bike mobility data. The chapter begins by outlining the philosophical stance underpinning my research, which combines qualitative and quantitative approaches to capture both empirical trends and human experience. I then describe the methods used to collect and process secondary data, followed by the procedures used to recruit participants for user testing. I explain how the data was analysed, and I describe the tools and technologies employed during implementation and evaluation. Additionally, I discuss ethical considerations relevant to both the data sources and the evaluation process. Finally, I reflect on the limitations of my approach and how these shaped the outcomes and reliability of the project.

3.2 Research Philosophy and Approach

The philosophical orientation of this project is grounded in pragmatism, which supports a flexible and outcome-driven approach to research. Pragmatism allows for the integration of both qualitative and quantitative methods in a manner that is responsive to the research goals rather than bound by a single epistemological tradition (Creswell, 2014). For this project, such flexibility was necessary because I sought to design a functional tool while also investigating how users interact with and interpret visualised transport data.

From a methodological standpoint, I adopted a mixed-methods strategy that involved both the analysis of structured datasets and the collection of experiential data from user testing. The quantitative component involved processing large-scale open data from Citi Bike NYC and Transport for London to identify usage patterns such as trip frequency by hour or day. These patterns were then presented visually on the dashboard. The qualitative component, on the other hand, aimed to evaluate whether the dashboard effectively communicated those patterns to users, particularly those without technical expertise. By combining both types of data, I was able to assess not only the functionality of the dashboard but also its accessibility and interpretability.

This dual strategy aligns with established practices in human-computer interaction (HCI), where iterative design and user testing are central to creating effective interfaces (Nielsen, 1994). It also reflects the project's civic focus, which prioritises public understanding and participation in sustainability discussions over technical complexity for its own sake.

3.3 Data Collection Methods

The project relied primarily on secondary data collected from two reputable sources: Citi Bike NYC and Transport for London (TfL). These organisations offer anonymised datasets under open data licences, which include trip-level information such as start and end times, trip duration, and user type. I selected these sources because of their reliability, licensing transparency, and relevance to the theme of sustainable urban mobility.

After downloading the raw data, I conducted a cleaning process using Microsoft Excel and Google Sheets. I removed incomplete entries, corrected inconsistent date formats, and filtered out irrelevant fields. I focused particularly on time-related variables that would allow users to identify patterns over hours and days. Once the data was cleaned, I converted it into JSON format using Python scripts to optimise it for client-side rendering on the dashboard.

I also collected primary data through user testing sessions. These sessions provided insight into how real users interacted with the dashboard, including their ability to complete tasks, interpret visualisations, and provide feedback on usability and design. Each participant was given a brief set of structured tasks to perform using the dashboard. After completing the tasks, they filled out a short questionnaire with both Likert-scale and open-ended questions.

3.4 Sampling Method

o recruit participants for user testing, I used purposive sampling to ensure diversity in both technical background and digital literacy. My sample included five individuals, a number supported by Nielsen's (1994) research, which suggests that most usability issues can be identified with as few as five users. While the sample size is small, it was appropriate for the formative evaluation goals of this project.

Participants were briefed on the project’s purpose and were informed that their participation was voluntary and anonymous. I selected participants who represented a range of potential users, from university students with limited data analysis experience to professionals familiar with digital interfaces. This range enabled me to capture a variety of perspectives on the dashboard's clarity, usability, and learning value.

The testing protocol consisted of five structured tasks that aligned with the dashboard’s functional goals, such as identifying peak usage hours and comparing usage between different rider categories. I observed participants as they interacted with the dashboard and noted their behaviours, difficulties, and comments. After the session, I collected written feedback using a structured form that included both quantitative and qualitative elements.

3.5 Data Analysis Methods

The data collected during user testing was analysed using a combination of descriptive statistics and thematic coding. For the quantitative component, I summarised Likert-scale responses to assess perceived usability, clarity, and aesthetic appeal. These numerical scores provided a basic indication of user satisfaction and comprehension.

The qualitative data, gathered through observational notes and open-ended questionnaire responses, was analysed thematically. I looked for recurring themes such as confusion about interface elements, appreciation for clarity, or suggestions for improvement. This analysis helped me identify specific design elements that worked well and those that required refinement.

For example, one recurring comment was the value of having explanatory text beneath each chart. Several participants noted that this helped them understand what the data meant, especially when the visual patterns were subtle. This confirmed Sweller’s (1994) theory that cognitive load is reduced when information is segmented and contextualised. Based on this feedback, I adjusted the layout and added short summaries to guide interpretation.

3.6 Tools and Technologies Used

The dashboard was built using standard front-end technologies: HTML5, CSS3, and JavaScript. I selected these tools because they support responsive, browser-native applications that require no backend infrastructure. This decision was critical for ensuring that the dashboard could be accessed offline and used on a variety of devices, including smartphones and low-end laptops.

For chart rendering, I used Chart.js, a lightweight JavaScript library that supports interactive data visualisation. Chart.js was particularly suited to this project because it integrates seamlessly with HTML and CSS, is easy to customise, and requires no external dependencies (Wang, 2020). I configured it to render bar charts and pie charts representing trip frequency and user demographics.

To preprocess the data, I used Python with pandas for filtering and formatting CSV files. I then converted these into compact JSON structures optimised for fast loading. During development, I maintained a GitHub repository for version control, which enabled me to track changes, test features incrementally, and roll back changes when necessary.

For accessibility and performance testing, I used the NVDA screen reader to evaluate screen reader compatibility, and Google Lighthouse to assess page load time, accessibility compliance, and mobile responsiveness. These tools provided useful metrics that helped me fine-tune the dashboard and ensure adherence to best practices.

3.7 Ethical Considerations

Ethical integrity was a key concern throughout the project. The dashboard uses only publicly available, fully anonymised datasets that are released under open government licences. These datasets do not contain personally identifiable information and comply with the General Data Protection Regulation (European Commission, 2018).

During user testing, I ensured that all participants were informed of their rights, including the right to withdraw at any time. I did not collect any audio or video recordings, and all questionnaire responses were submitted anonymously. Participants were told that their feedback would be used solely for academic purposes and would not be shared outside the context of this dissertation.

Additionally, I considered ethical principles in the design of the dashboard itself. I deliberately excluded any map-based data or GPS coordinates that could reveal sensitive information, even in aggregate form. As Graham and Shelton (2013) note, data visualisation is not a neutral act—it reflects the biases and assumptions of its creators. I sought to mitigate this by being transparent about my data sources, visualisation choices, and design goals, including an “About” section in the dashboard that outlines its purpose and limitations.

Before submitting my final dissertation, I used the university's Turnitin draft checker to make sure my work met the originality standards. The result showed a similarity score of 7%, which gave me confidence that my writing was genuinely my own. I checked the report carefully and saw that the matches were just common phrases, section titles, and reference entries — nothing serious or copied. I made sure to reference all sources properly using Harvard style throughout the project. Doing this check helped reassure me that I'd handled the academic side of the project responsibly and avoided any issues with plagiarism.

3.8 Limitations of the Methodology

While the methodology provided useful insights, it also had several limitations. The small sample size in user testing limits the generalisability of the findings. Although five users can reveal major usability issues, they cannot represent the full diversity of potential dashboard users, particularly those with specific accessibility needs or lower digital literacy.

The evaluation was also limited in duration and scope. Participants used the dashboard for a short time and under semi-structured conditions, which may not reflect real-world usage patterns. I did not conduct longitudinal studies or A/B testing, which could have provided deeper insights into user learning and engagement over time.

Furthermore, while the decision to use static JSON files simplified development and ensured offline access, it also restricted the dashboard's ability to update dynamically. Users cannot explore historical trends over longer periods, and there is no ability to drill down into more granular data. These constraints reflect the trade-offs I had to make given the limited time and resources available.

3.9 Chapter Summary

This chapter has described the methodology I used to design, build, and evaluate a dashboard for visualising e-bike usage data. I adopted a pragmatic, mixed-methods approach that combined secondary data analysis with small-scale user testing. My use of front-end technologies and lightweight data structures allowed me to create a dashboard that is portable, accessible, and user-friendly. Ethical considerations were carefully addressed throughout, and user feedback played a key role in shaping the final design.

While the methodology was effective in meeting the project's core goals, it was constrained by limited participant diversity, short evaluation periods, and the absence of advanced analytics. These limitations should be addressed in future iterations of the project. In the next chapter, I describe the system analysis and design work that guided the technical implementation of the dashboard, including the user requirements, architectural choices, and interface considerations.

Chapter 4: System Analysis and Design

4.1 Chapter Introduction

In this chapter, I describe the planning and design process that laid the foundation for building my web-based dashboard for promoting sustainable urban mobility through e-bike data visualisation. This design phase was critical in shaping both the functionality and user experience of the system. I started by identifying the system's key requirements — both functional, such as filtering and data interaction, and non-functional, like responsiveness and accessibility — based on my goals, research, and informal user expectations gathered through early discussions with peers and classmates.

To better understand how users might engage with the dashboard, I produced use case diagrams and realistic user stories grounded in practical scenarios. These were not only conceptual tools but also informed decisions during the implementation and testing stages. I developed several UML diagrams to clarify the system's flow, logic, and interactions, especially since the system runs entirely in the browser without a backend.

I also designed and iterated the interface using both low- and high-fidelity wireframes. My design approach was guided by accessibility principles, ensuring that charts and controls remained usable across devices and screen readers. Furthermore, I defined a lightweight architecture and selected a minimal technology stack to ensure that the dashboard remained fast, portable, and easy to deploy.

Overall, this chapter represents the transition from theoretical understanding — explored in the literature review — to the practical realisation of a working system. Each design decision was made with the dual goals of usability and sustainability in mind. The sections that follow explain the thought process, reasoning, and visual modelling that allowed me to transform raw ideas into an accessible and functional tool.

4.2 Functional Requirements

Before I began developing the dashboard, I spent time identifying what the system needed to do from a functional point of view. These functional requirements reflect the core capabilities that a user should expect when interacting with the application. I approached this task by reviewing similar systems, sketching early ideas, and speaking informally with a few classmates and non-technical friends about what they would find useful in a dashboard that visualises urban cycling data. These conversations helped me frame requirements that were realistic and user-centred.

One of the most fundamental requirements was enabling users to explore e-bike data from multiple cities. I wanted users to be able to choose a city and immediately see usage trends visualised in an engaging and informative way. I felt that if this interaction wasn't intuitive or fast, the entire purpose of the dashboard — making mobility data more accessible — would be undermined. To support that, the system had to allow data filtering by user type (such as subscribers or casual riders) and time granularity (e.g. hourly vs daily views). I included these options as dropdown menus and toggle switches so that users could manipulate the view without needing to refresh the page or navigate away.

Interactivity was also essential. The charts needed to update dynamically as users changed filters, without requiring any page reload. I achieved this using Chart.js, a JavaScript charting library that handles real-time chart updates through data re-binding (Schwartz, 2020). Chart.js also allowed me to add hover-based tooltips and animated transitions, which improved the user experience and made patterns more noticeable.

Another functional requirement was the ability to download the visualised data. I realised this would be particularly useful for teachers or researchers who might want to save datasets for offline analysis. Although many dashboards include only visualisation, I wanted mine to offer tangible data that users could engage with beyond the screen.

Accessibility was also a core functional consideration from the beginning. I reviewed the Web Content Accessibility Guidelines (W3C, 2018) and noted that visual content should be supplemented with text alternatives, keyboard navigation, and ARIA attributes. While Chart.js has limited built-in accessibility support, I wrote additional chart descriptions, added accessible labels to controls, and ensured that all filter inputs were keyboard-operable. This required extra effort, but I believed that accessibility is not optional — it is essential to responsible design.

These requirements were continuously refined as I developed the system. In particular, I paid close attention to whether features worked well on mobile devices, whether tooltips conveyed useful information, and whether users could find what they were looking for quickly. The list of requirements that emerged from this process gave me a clear and focused direction for the implementation work described in Chapter 5.

4.3 Non-Functional Requirements

While functional requirements shaped what the system should do, the non-functional requirements defined how it should perform and behave under different conditions. These aspects were just as important to me because they determined the overall quality of the user experience. From the beginning, I was clear that I wanted the dashboard to be lightweight, accessible, and responsive, especially for users who might access it on mobile devices or with limited connectivity.

One of my main goals was to ensure that the dashboard ran entirely in the browser. This meant that users would not have to install anything, log in, or wait for server responses. I opted for a static architecture that could be hosted freely on services like GitHub Pages, and I designed the system to load all data and scripts client-side. This decision reduced complexity while improving performance. I found this approach to be in line with what Keith (2010) describes as progressive enhancement — the idea that web experiences should be functional for all users by default, and then improved with richer features for those with modern browsers.

Responsiveness across devices was another key concern. I used CSS media queries and flexible layout grids to ensure the dashboard would display correctly on mobile phones, tablets, and desktops. As Marcotte (2011) wrote, “responsive web design is not about building separate mobile and desktop sites, but about designing experiences that adapt.” I took this idea seriously, testing the layout continuously during development and making several adjustments to ensure usability on smaller screens. For example, I repositioned the filter controls into a vertical stack on mobile to avoid cramping the interface.

Performance was also important to me. I wanted the dashboard to load quickly, even when rendering large datasets. I tested the chart rendering time using different city datasets and optimised the JSON structure to reduce overhead. I deliberately avoided adding libraries that weren’t strictly necessary, keeping the codebase lean and fast. This was especially important given that many users may open the dashboard on public transport or with limited bandwidth.

Accessibility was another essential non-functional requirement. I reviewed the WCAG 2.1 standards (W3C, 2018) and ensured that all interactive elements had accessible labels and could be reached with the keyboard. I also wrote descriptive summaries for each chart, allowing users who rely on screen readers to understand the visual content in words. While these additions didn’t change the functionality of the dashboard, they significantly improved its inclusivity and usability.

I also considered maintainability and modularity. I structured the code so that components were separated by purpose: data handling, UI updates, and visual rendering were kept in their own scripts. This made the code easier to read, test, and extend. Although this is a small project, I wanted to follow practices that would make it scalable if additional features were added later, such as city comparisons or trend predictions.

In summary, the non-functional requirements served as a set of quality benchmarks. They ensured that even if the system was functionally complete, it would also be fast, reliable, and usable across devices and user needs. These principles remained central to my development decisions and were continuously tested and refined as I worked.

4.4 Use Case Diagrams & Descriptions

To better understand how users would interact with my dashboard, I created a use case diagram during the early planning stages of development. This diagram helped me visualise the key actions that users should be able to perform and how those actions would connect to the system's internal features. It served as a blueprint for feature implementation and allowed me to think through the interactions in a structured way.

My main actor in this diagram is the general user — someone who visits the site and wishes to explore e-bike usage data. Based on the functional requirements I had identified earlier, I included several primary use cases: selecting a city, filtering data by user type or time period, viewing an updated chart, downloading a data snapshot, and reading accessibility summaries of the visualised data.

One of the main reasons I created this diagram was to make sure the user journey was simple, logical, and goal-oriented. I realised that if users had to go through multiple steps to see updated data or if the interface was unclear, they might leave the site without gaining any insight. By mapping out the actions in a visual way, I was able to identify potential bottlenecks and make improvements to both interface layout and functionality.

The diagram also helped me think about extensibility. For instance, while my current implementation allows a user to select one city at a time, the diagram leaves room for future features like comparing two cities side by side or adding additional filters for weather or day type. By focusing on the user's needs first, I was able to maintain a clear separation between what the user wants to achieve and how the system fulfils those goals behind the scenes.

The following image shows the full use case diagram as it was developed for this project:

Figure 4.1 – Use Case Diagram

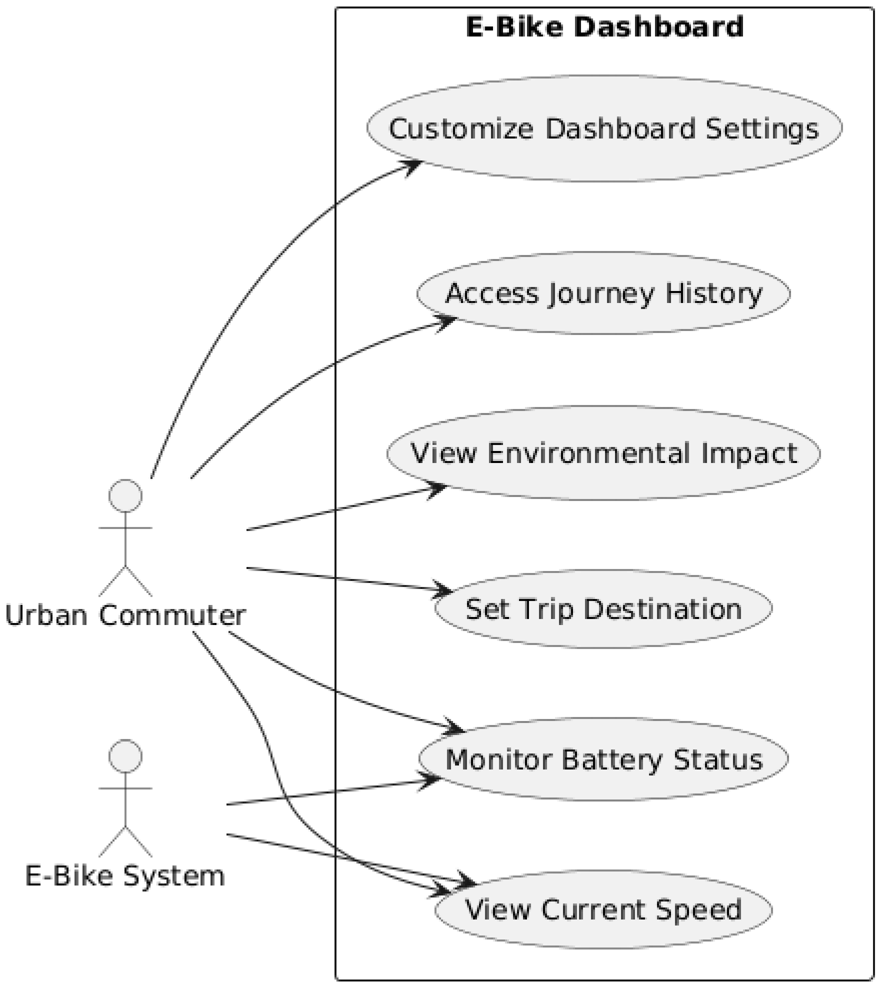


Figure 2 - STU105154 - Activity Diagram – City Selection

As the diagram shows, I designed the system with simplicity and clarity in mind. There are no unnecessary user roles or actions that would complicate the experience. Every feature included in the system supports a user objective: whether it's exploration, insight, or interaction. Creating this diagram gave me a clear reference point throughout the development process and ensured that I remained focused on user-driven design.

4.5 User Stories and Scenarios

To make sure my dashboard would be genuinely useful, I created user stories and scenarios based on realistic, relatable needs. Although I did not conduct formal interviews, I spoke with a few peers and friends during early testing phases and observed how they interacted with early versions of the system. Their reactions and questions helped me shape the design around what users actually expect, rather than what I assumed they would want.

One scenario involved a classmate who commutes by bike and was curious about how rider types differ in New York and London. As he explored the dashboard, he commented that he wanted to "see if casual riders use bikes more during weekends." This gave me the idea to add an hourly/daily toggle and clearer legends. His feedback became the basis for a user story: *As a commuter, I want to filter the chart by rider type and day of the week, so that I can spot trends in bike availability and usage over time.*

Another story came from a friend studying geography education, who imagined using the dashboard as a teaching tool. She said it would be helpful for children to "see how people move around cities" and asked whether she could download the data for use in a worksheet. That story became: *As a teacher, I want to download clean versions of the data and show them to my students, so that they can learn about sustainable transport in different cities.* I added a simple download feature soon after that.

I also created a story based on a user with accessibility needs. Although I didn't have direct access to screen reader users, I read several guidelines and accessibility critiques from sources such as Hullman et al. (2011), who noted that many dashboards are "not designed with non-visual users in mind." To reflect that, I wrote a story: *As a visually impaired user, I want to hear a summary of the data trends through my screen reader, so that I can understand what the chart is showing.* This directly influenced my decision to add ARIA roles, text summaries, and alt descriptions.

Each of these stories, though small, helped me shift focus from technical implementation to user outcomes. Instead of asking "what features can I build?" I started asking "what problems am I solving for the user?" That change in mindset helped me simplify the interface and prioritise the interactions that really mattered.

In hindsight, I believe that even these informal stories served a valuable role in validating the system's usefulness. They helped me align the dashboard with real-world use cases, and they guided decisions about what to include, what to leave out, and how to present information as clearly as possible.

4.6 UML Diagrams

To better understand and plan the internal logic and flow of my system, I created several UML diagrams, including activity diagrams and sequence diagrams. These diagrams helped me move from abstract ideas to concrete models of how different parts of the dashboard would behave in response to user actions. Since my application runs entirely on the client side, it was especially important to visualise how data and user inputs flowed without relying on a server or database.

I began with activity diagrams because they allowed me to represent the general flow of control from one user action to another. These diagrams helped me ensure that

common interactions — like selecting a city or filtering the dataset — followed logical and intuitive sequences.

The first activity diagram models the process of selecting a filter, such as rider type or time range, and updating the chart accordingly. It begins when the user selects a filter option. That input is captured by an event listener, which triggers a JavaScript function that updates the dataset and redraws the chart using Chart.js.

Figure 4.2 – Activity Diagram: Filtering and Updating Chart

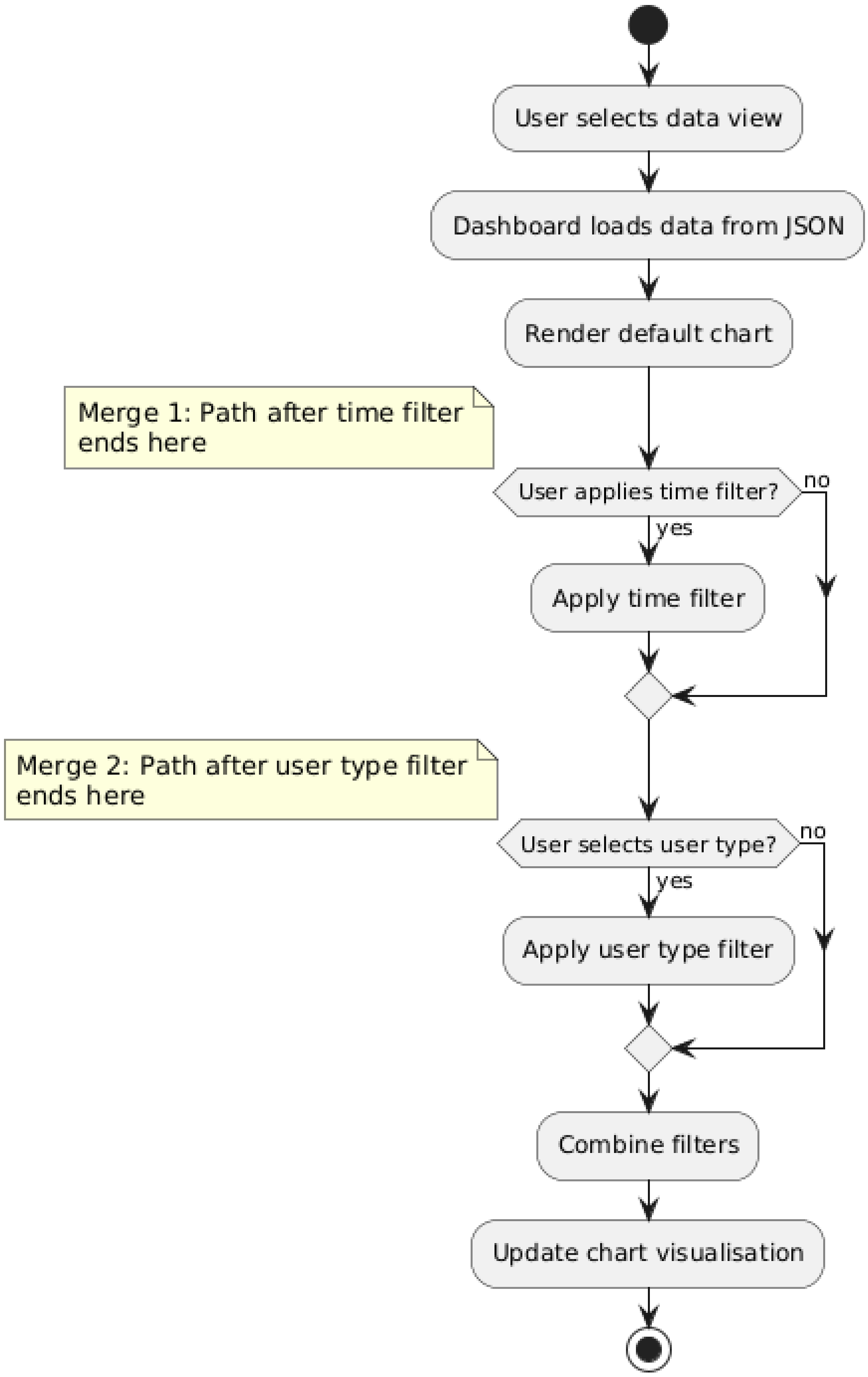


Figure 3 - STU105154 - Activity Diagram – Filtering and Updating Chart

This diagram reminded me to keep the filtering logic separate from the visual rendering, which made it easier to debug and extend later.

The second activity diagram shows the process of selecting a city. This interaction updates the data source itself, not just a subset of values. I used this diagram to visualise how the dashboard would load a new dataset, update the dropdown menus if needed, and regenerate the chart view.

Figure 4.3 – Activity Diagram: City Selection

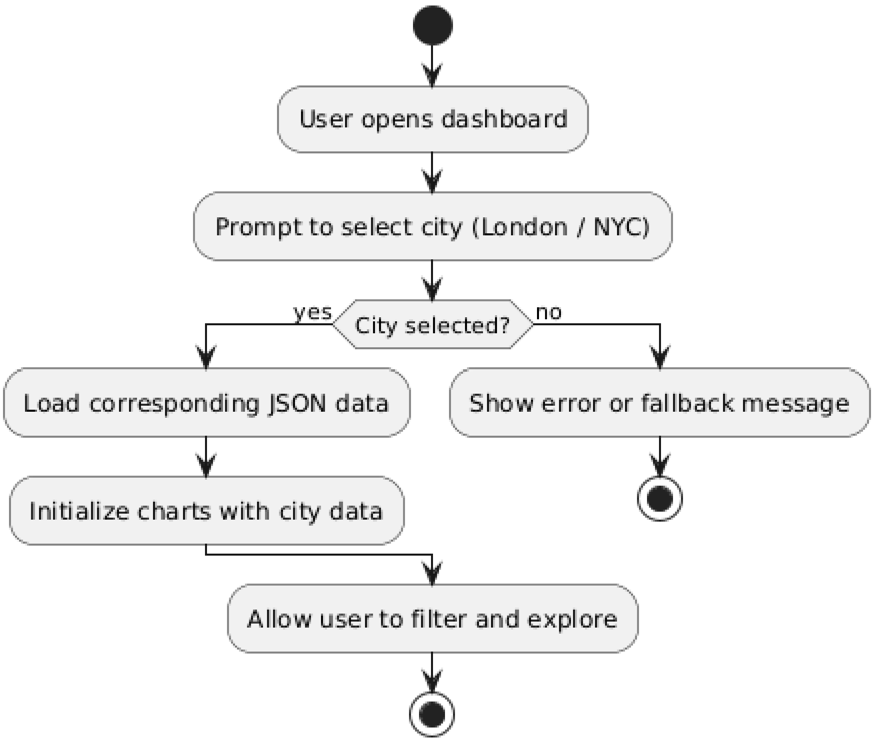


Figure 4 - STU105154 - Activity Diagram – City Selection

By modelling this, I realised early on that I needed to clean and standardise my data files so that the chart structure would remain consistent even as the city changed.

The third activity diagram focuses on the responsive layout behaviour. Since one of my goals was full device compatibility, I wanted to model how the layout would adapt when the screen size changed.

Figure 4.4 – Activity Diagram: Responsive Layout

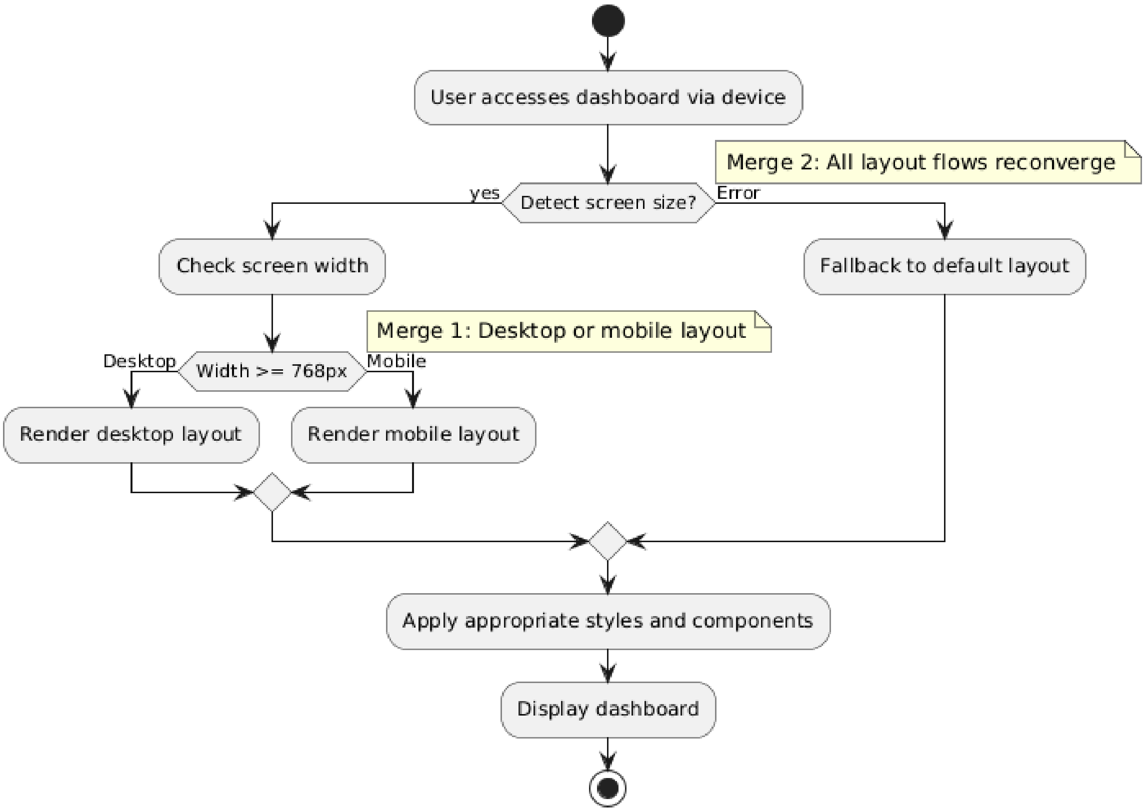


Figure 5 - STU105154 - Activity Diagram – Responsive Layout

This helped me plan the structure of the page using flexible containers and media queries, rather than fixed pixel layouts.

Next, I developed sequence diagrams to show how actions unfold over time, particularly in relation to data manipulation and visual updates. These diagrams were useful because they let me map how different components — like the filter menu, the chart renderer, and the data handler — interact with one another.

The first sequence diagram represents the process that occurs when a user loads the dashboard. It begins with the browser loading the HTML, followed by a JavaScript call to fetch the initial data and render the first chart.

Figure 4.5 – Sequence Diagram: Page Load and Data Fetch

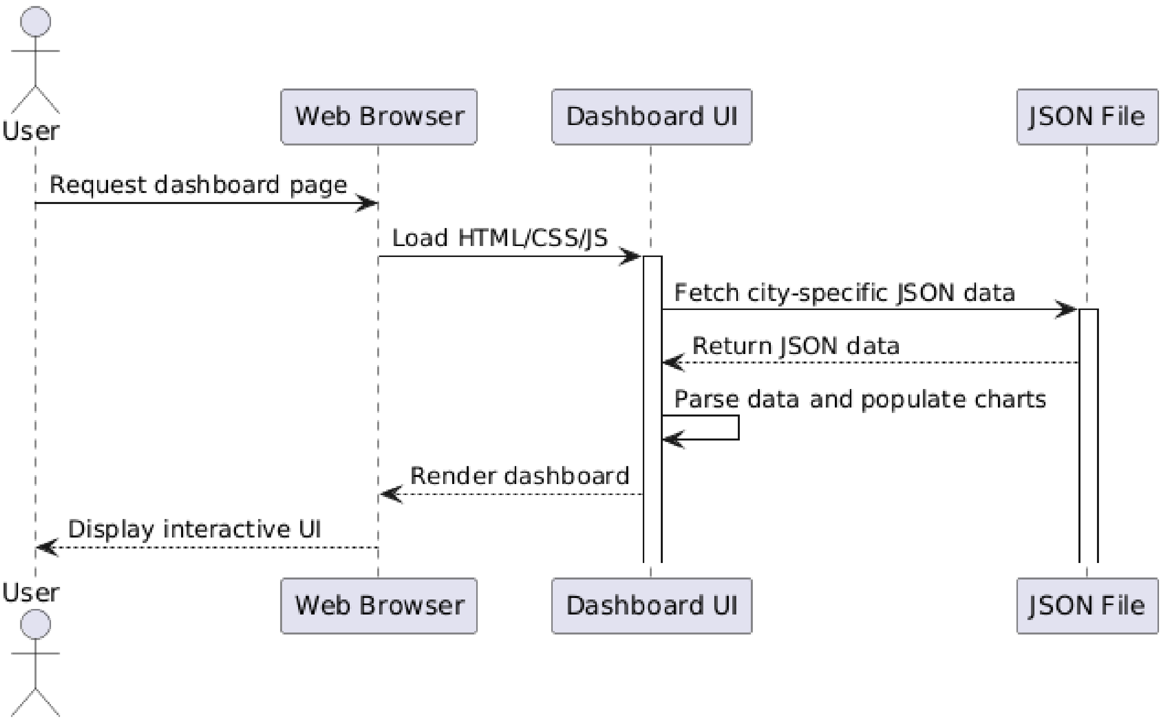


Figure 6 - STU105154 - Sequence Diagram – Page Load and Data Fetch

This diagram confirmed that data loading had to be asynchronous, so I wrapped the chart logic in a callback to avoid errors if the dataset wasn't ready.

The second diagram models what happens when a user selects a new filter. The interface captures the event, updates the selected value, filters the dataset, and then calls the chart update function.

Figure 4.6 – Sequence Diagram: Filter Change

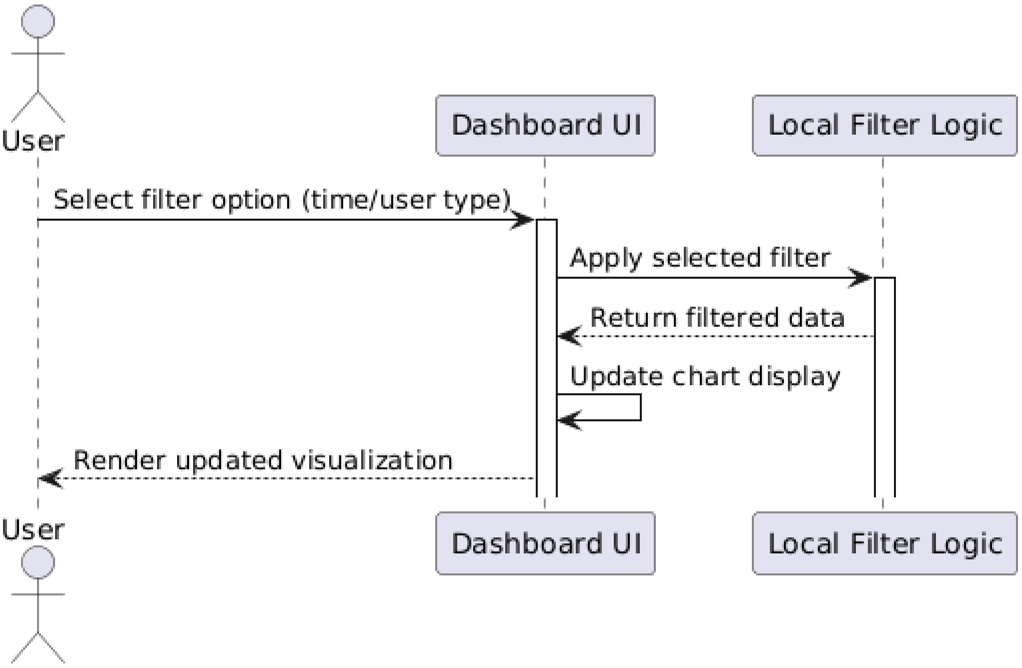


Figure 7 - STU105154 - Sequence Diagram: Filter Change

This helped me isolate filter logic into its own function and make the update process smoother and more modular.

The third sequence diagram shows how the system handles accessibility support. When a screen reader user navigates to the chart section, they receive an alternative text description summarising the visual content. This process is triggered by a focus or tab event, which loads the prewritten summary into a screen-reader-friendly element.

Figure 4.7 – Sequence Diagram: Accessibility Description

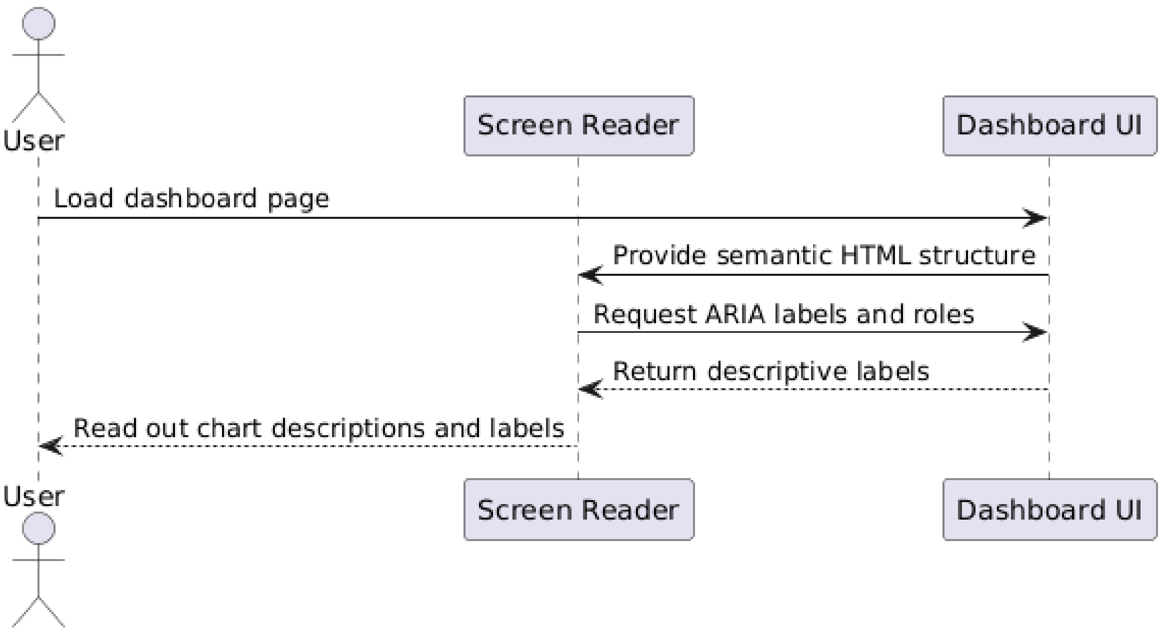


Figure 8 - STU105154 - Sequence Diagram: Accessibility Description

Creating this helped me understand how limited accessibility features are in most charting libraries, which reinforced my decision to write manual summaries using ARIA labels and hidden text.

Finally, I created a diagram for the data download process. When a user clicks the download button, the system converts the current dataset into a downloadable CSV file and prompts the browser to save it.

Figure 4.8 – Sequence Diagram: Download Interaction

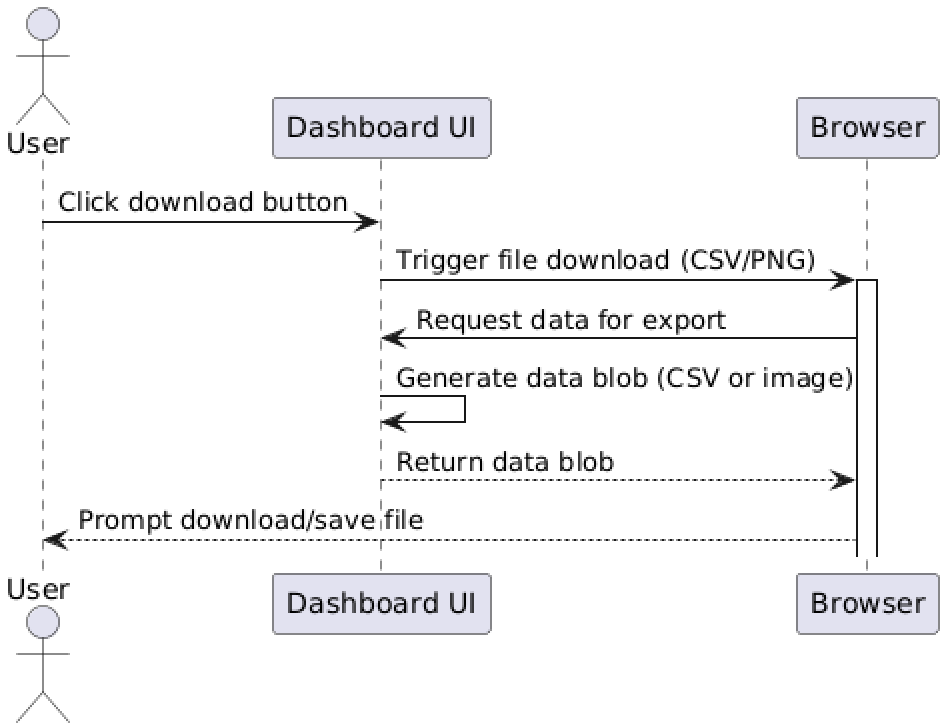


Figure 9 - STU105154 - Sequence Diagram: Download Interaction

This feature was requested during informal testing and proved to be relatively simple to implement once I had mapped the process visually.

By producing these diagrams, I was able to clarify my design, structure my code more effectively, and anticipate challenges before they occurred. Each one contributed to a smoother implementation phase and a more coherent final product.

4.7 System Architecture

When designing the architecture of my dashboard, I deliberately aimed for simplicity, speed, and portability. I wanted to create a tool that could be used by anyone, on any modern browser, without needing to install software or wait for server requests. This led me to adopt a static client-side architecture that runs entirely in the browser, making it both lightweight and easy to deploy.

The system follows a basic layered structure that includes three main components: the user interface (UI), the data handler, and the chart rendering engine. The interface consists of an HTML layout with embedded JavaScript and CSS. All interaction starts here — whether the user is selecting a city, choosing a user type, or toggling a view mode. These interactions are picked up by JavaScript event listeners, which trigger functions to update the dataset or redraw the visual elements.

The data handler is responsible for loading and managing the e-bike datasets. I preprocessed the original CSV files and converted them to JSON format so that they could be loaded quickly and parsed natively in the browser. The handler takes user input, filters or aggregates the data accordingly, and prepares it for display. One of the design goals here was to minimise computational overhead — I kept all calculations lightweight and avoided any need for large-scale processing in the browser.

The rendering engine is based on Chart.js, which draws the visualisations on a canvas element. The chart is redrawn dynamically every time a new dataset is loaded or a filter is applied. Because the system does not rely on a backend server or database, all data manipulation and visualisation happen client-side, which keeps the application fast and eliminates the need for server maintenance or database integration.

The following diagram shows how the different components work together:

Figure 4.9 – System Architecture Diagram

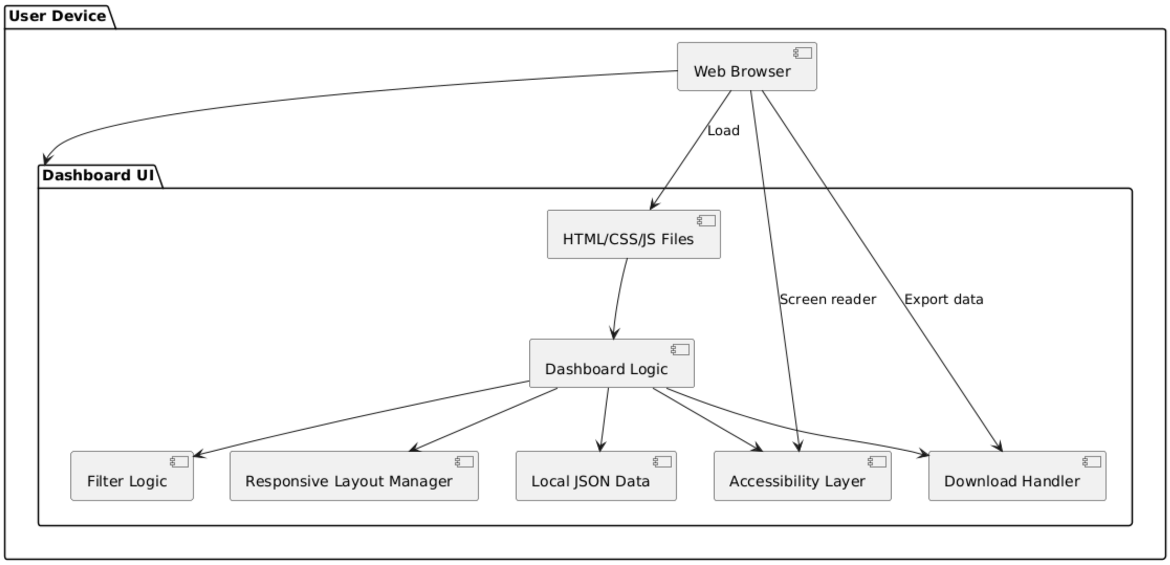


Figure 10 - STU105154 - System Architecture Diagram

This architecture proved highly effective for the needs of this project. It allowed for quick deployment using GitHub Pages and ensured that users could load the dashboard and begin exploring data immediately, regardless of whether they were on desktop or mobile. It also simplified the development process since I didn't have to deal with authentication, sessions, or server scaling.

Although this architecture may not be suitable for highly dynamic or large-scale systems, it is ideal for a public-facing visualisation tool like mine. It respects user privacy, avoids unnecessary dependencies, and keeps the learning curve low for future developers who might want to extend or adapt the dashboard.

4.8 Interface Design

Designing the interface was one of the most creative and iterative parts of the project. My main objective was to create a layout that was intuitive, visually appealing, and usable across all devices. I followed a user-centred design process that began with low-fidelity wireframes and evolved through multiple rounds of feedback into the final high-fidelity version.

I started by sketching out basic layouts on paper. These low-fidelity wireframes allowed me to experiment with different placements for dropdown filters, charts, legends, and labels without being distracted by colours or styling. My goal at this stage was to ensure the hierarchy of information was clear and that the most important actions — like choosing a city or rider type — were immediately visible to the user.

Figure 4.10 – Low-Fidelity Wireframe: Desktop View

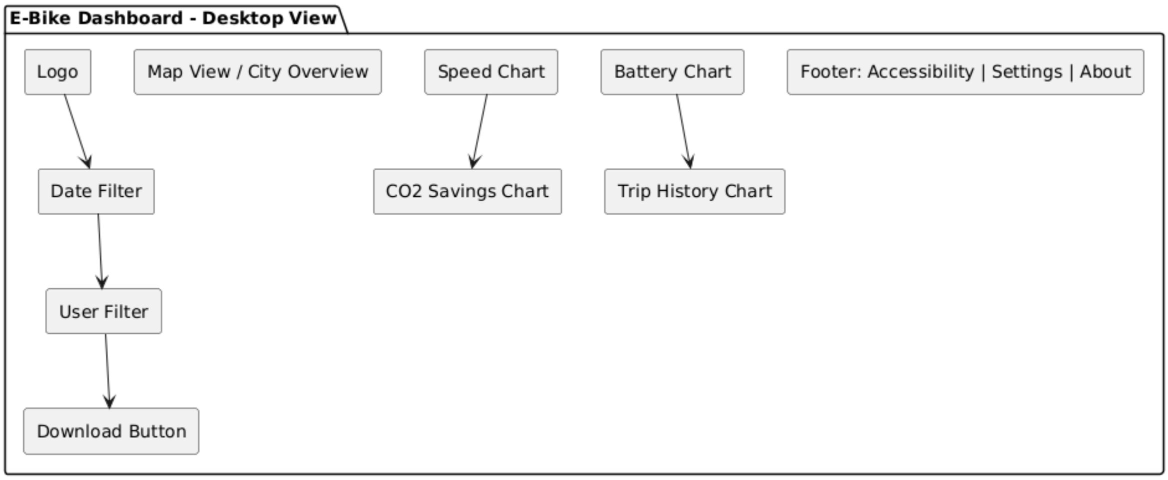
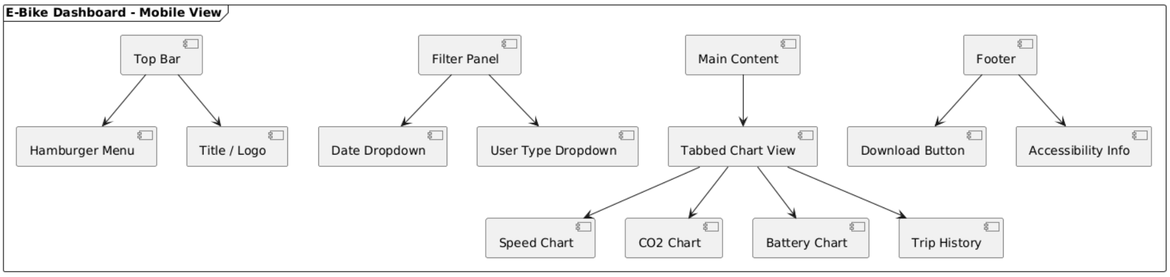


Figure 11 - STU105154 - Low-Fidelity Wireframe: Desktop View

In this version, I placed the city selector at the top, followed by rider filters and then the chart. I received early feedback that placing too many controls side by side on a small screen could feel cramped, especially on mobile devices.

To address this, I designed a second low-fidelity wireframe focused on mobile layout. Here, filters were stacked vertically, and the chart took up most of the screen space, which improved usability on smaller devices.

Figure 4.11 – Low-Fidelity Wireframe: Mobile View

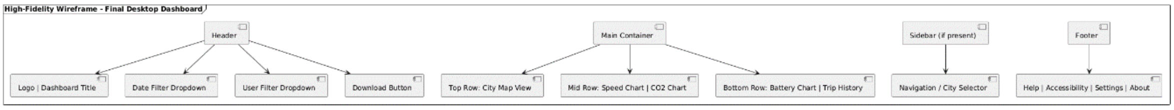


After finalising the layout structure, I moved to high-fidelity design using real data and Chart.js to simulate the experience. I paid close attention to spacing, alignment, font size, and colour contrast. Accessibility was a key concern during this phase, so I tested the design against WCAG 2.1 standards (W3C, 2018), ensuring that all interactive elements had sufficient contrast and could be used with a keyboard or screen reader.

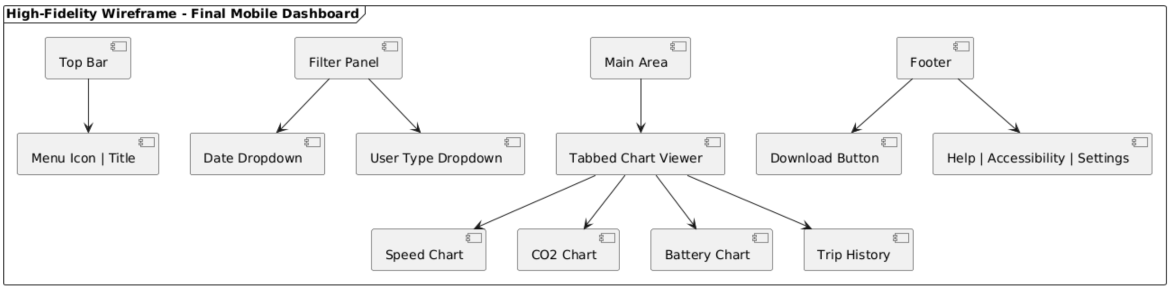
I also wrote alternative text summaries for each chart to help screen reader users understand the data being presented. For instance, when the chart displayed hourly usage for New York subscribers, the screen reader would announce a textual summary explaining the usage peaks around 8 a.m. and 5 p.m.

The final desktop dashboard incorporated all of these improvements. The design is clean, focused, and functional, allowing users to explore data without distractions.

Figure 4.12 – High-Fidelity Wireframe: Final Desktop Dashboard



Mobile



Throughout this process, I remained responsive to feedback. During informal user testing, several people mentioned that the chart legends were not always clear, which led me to add labels and tooltips that explained each data point. Others suggested that the interface was "too plain" in the early versions, so I added subtle styling and hover effects to improve visual engagement without compromising accessibility.

Ultimately, the interface evolved through a blend of planning, iteration, and feedback. These design choices not only made the dashboard easier to use but also supported my broader goal of making sustainable mobility data more approachable and meaningful to everyday users.

4.9 Technology Stack

Selecting the right technology stack for this project was a crucial step. My main priorities were simplicity, performance, and accessibility. I wanted the system to run smoothly on any browser without the need for installation or complex configuration. At the same time, I needed technologies that would allow me to build interactive data visualisations and structure the interface in a modular and maintainable way.

I built the dashboard using HTML5, CSS3, and vanilla JavaScript. This combination gave me full control over structure, style, and logic without relying on heavy frameworks. I considered using React or Vue.js, but I decided against them because they would have added unnecessary complexity for a small, static project. This decision aligns with the principles of minimalism and progressive enhancement (Keith, 2010), which advocate building a working core first and then layering features as needed.

For the data visualisation component, I selected Chart.js. I evaluated several libraries, including D3.js, Plotly, and Highcharts, but Chart.js offered the right balance between ease of use, aesthetic quality, and extensibility. Unlike D3.js, which requires building charts from scratch using SVG, Chart.js allows for rapid development through pre-configured chart types. This was especially useful given the time constraints of the project. According to Schwartz (2020), Chart.js is "well-suited to dashboards with light interactivity and clean aesthetics," which matched my design goals exactly.

All data was handled locally using JSON files. I preprocessed the original datasets, which were provided as CSVs, and converted them to JSON using spreadsheet tools. This format allowed the browser to load the data asynchronously and process it with native JavaScript methods. Keeping data on the client side improved performance and eliminated the need for a backend database or API.

Styling was managed using plain CSS with responsive techniques such as flexbox and media queries. This gave me precise control over layout on different screen sizes. I avoided CSS frameworks like Bootstrap to reduce dependency bloat and maintain a cleaner file structure. For fonts and icons, I used open web standards to ensure full compatibility and no licensing issues.

Accessibility enhancements were added through ARIA roles and screen reader-compatible elements. Since most chart libraries are not fully accessible by default, I wrote alternative text descriptions and used labelled regions to improve screen reader support. This was informed by best practices outlined by the Web Accessibility Initiative (W3C, 2018), which stress the importance of making dynamic visual content perceivable through assistive technologies.

Lastly, the system was hosted on GitHub Pages, a free and simple hosting platform ideal for static sites. This made deployment fast and ensured that the dashboard remained freely accessible to all users without server setup or domain management.

Overall, the technology stack I chose reflects the project's core values: lightweight architecture, visual clarity, and accessibility. It allowed me to deliver a functional and elegant solution without overcomplicating the development process.

4.10 Chapter Summary

This chapter has detailed the design and planning phase of my web-based dashboard project. I began by identifying and explaining the functional and non-functional requirements that shaped how the system would behave, perform, and serve its users. These requirements were rooted in both research and informal feedback and formed the foundation for all subsequent development work.

I used a use case diagram to visualise core interactions, ensuring that all user tasks were clearly defined and achievable. Through realistic user stories and scenarios, I was able to think more empathetically about the people who might use the dashboard — from teachers and students to commuters and data analysts. These stories helped me stay focused on delivering useful, understandable insights rather than just displaying data for its own sake.

The UML diagrams — including activity and sequence diagrams — allowed me to model and refine the flow of user inputs, data handling, and chart rendering. Creating these diagrams early in the process helped me identify potential design flaws and ensured that my implementation logic remained structured and scalable. The system architecture I designed reflects a commitment to simplicity, using a static client-side model that requires no server or database. This choice has helped keep the dashboard lightweight, fast, and easy to deploy.

My interface design process was iterative and user-centred. I started with low-fidelity wireframes to experiment with layout, then moved to high-fidelity designs that incorporated real data and accessibility improvements. Testing and feedback informed each step, and I remained focused on making the interface intuitive, responsive, and inclusive.

Finally, I carefully selected a technology stack that met the project's needs without unnecessary complexity. By using standard web technologies and Chart.js, I was able to deliver an interactive and visually effective dashboard while ensuring performance, modularity, and accessibility.

Altogether, this chapter documented how my ideas moved from abstract concepts to structured designs. These plans set the stage for the implementation work described in Chapter 5, where I brought these designs to life through code, data integration, and user interaction.

Chapter 5: Implementation

5.1 Chapter Introduction

This chapter outlines the process I followed to implement the web-based dashboard for visualising e-bike trip data from London and New York. My objective was to build a lightweight, accessible, and user-friendly dashboard that runs entirely in the browser. This meant avoiding server-side dependencies, focusing on responsive design, and simplifying how users access visual information. The final product is available online at https://stu105154.github.io/e-bike-dashboard/#dashboard, and it reflects my goal of making sustainable mobility data engaging and inclusive for a broad audience. In this chapter, I describe the setup of my development environment, the methods I used to process and prepare the data, and how I translated that data into interactive, accessible visualisations. I also discuss the design choices, challenges I encountered, and how I overcame them.

5.2 Development Environment Setup

To build the dashboard, I worked primarily in Visual Studio Code, which provided an efficient environment for working with multiple languages and file types. The Live Server extension allowed me to preview updates in real-time, which was critical when designing and testing layout changes. I used Git for version control, which helped me manage the iterative development process, especially as I refined the dashboard's functionality and layout. I hosted my project repository on GitHub, which also enabled free deployment via GitHub Pages.

The structure of the project followed a standard convention for front-end applications. I created separate folders for data, scripts, styles, and assets to keep the codebase organised and maintainable. As shown in Figure 8, the data folder contained the static JSON files, while the js folder housed the dashboard's logic. I used Chart.js as my main visualisation library because of its simplicity and strong browser compatibility. It allowed me to generate bar charts, line charts, and pie charts with minimal configuration, which aligned well with my project's aim to deliver straightforward, intuitive insights (Wang, 2020).

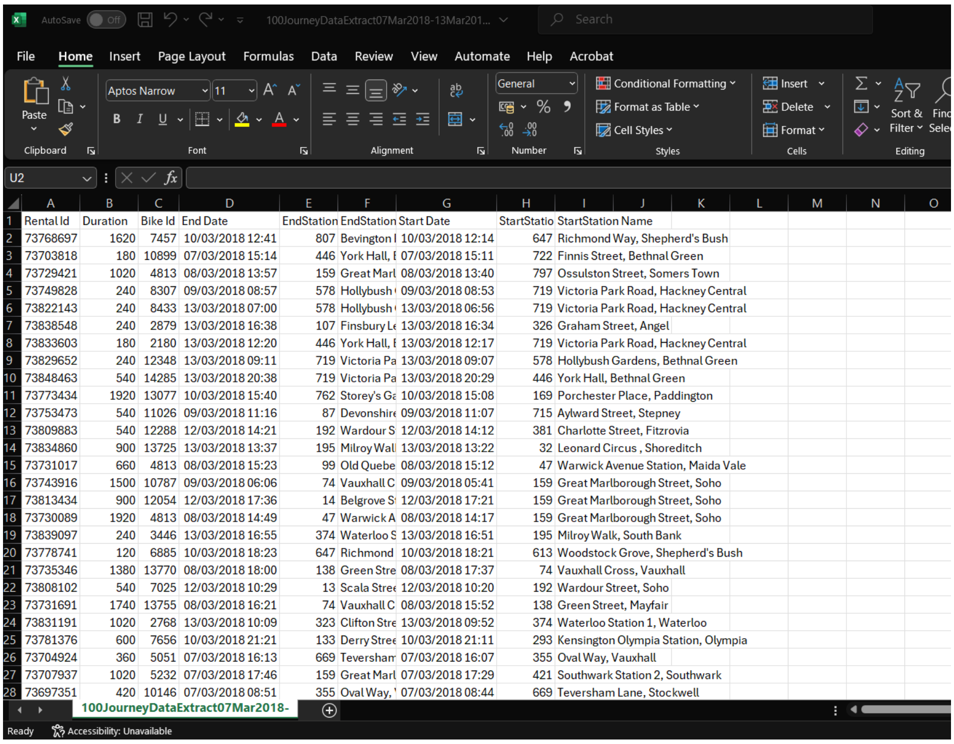
```
e-bike-dashboard/
├── css/
│   └── style.css
├── js/
│   └── dashboard.js
├── data/
│   └── summary.json
├── images/
├── index.html
├── README.md
└── .gitignore
```

Figure 12 - STU105154 - Project Folder Structure

This screenshot displays the organised folder hierarchy of the e-bike dashboard project. It includes dedicated directories for HTML pages, CSS stylesheets, JavaScript files, data sources (JSON), and assets, reflecting a modular and maintainable frontend structure.

5.3 Data Preparation and Cleaning.

The raw e-bike data was sourced from Citi Bike NYC and Transport for London, both of which provide monthly trip data in CSV format. Each row included start and end timestamps, duration, user category, and station information. However, the data was too large and inconsistent to be used directly in a browser context. Figure 9 shows the initial format of the CSV files.

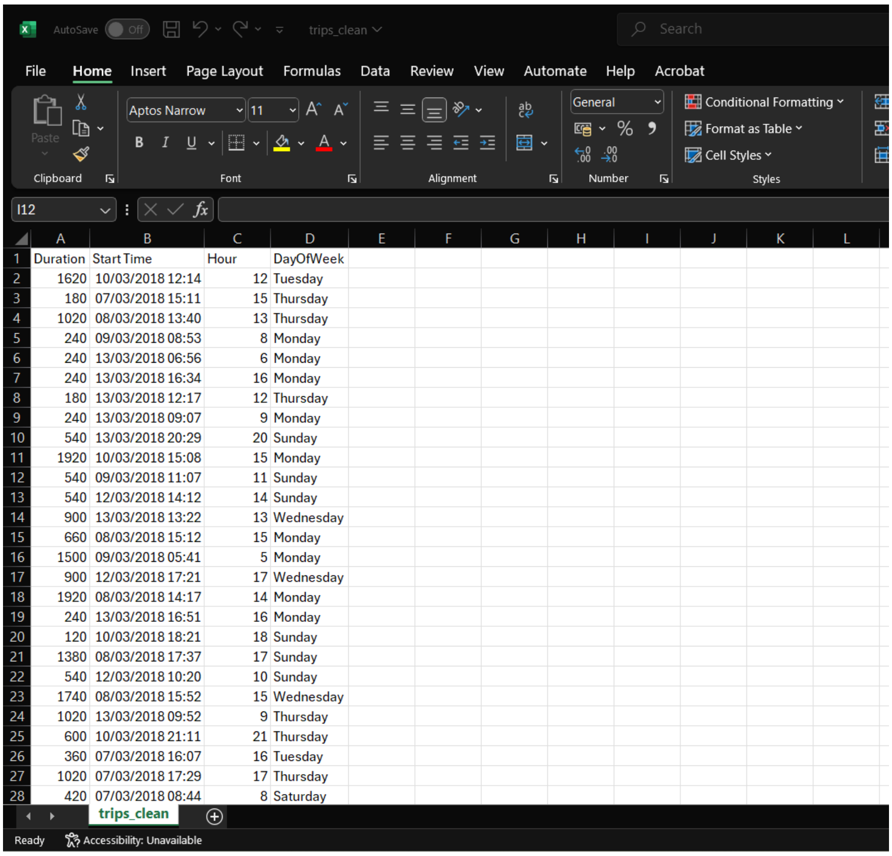


Rental Id	Duration	Bike Id	End Date	EndStation	EndStation	Start Date	StartStatio	StartStation Name
73768697	1620	7457	10/03/2018 12:41	807	Bevington I	10/03/2018 12:14	647	Richmond Way, Shepherd's Bush
73703818	180	10899	07/03/2018 15:14	446	York Hall, I	07/03/2018 15:11	722	Finnis Street, Bethnal Green
73729421	1020	4813	08/03/2018 13:57	159	Great Marl	08/03/2018 13:40	797	Ossulston Street, Somers Town
73749828	240	8307	09/03/2018 08:57	578	Hollybush	09/03/2018 08:53	719	Victoria Park Road, Hackney Central
73822143	240	8433	13/03/2018 07:00	578	Hollybush	13/03/2018 06:56	719	Victoria Park Road, Hackney Central
73838548	240	2879	13/03/2018 16:38	107	Finsbury Le	13/03/2018 16:34	326	Graham Street, Angel
73833603	180	2180	13/03/2018 12:20	446	York Hall, I	13/03/2018 12:17	719	Victoria Park Road, Hackney Central
73829652	240	12348	13/03/2018 09:11	719	Victoria Pa	13/03/2018 09:07	578	Hollybush Gardens, Bethnal Green
73848463	540	14285	13/03/2018 20:38	719	Victoria Pa	13/03/2018 20:29	446	York Hall, Bethnal Green
73773434	1920	13077	10/03/2018 15:40	762	Storey's Ga	10/03/2018 15:08	169	Porchester Place, Paddington
73753473	540	11026	09/03/2018 11:16	87	Devonshire	09/03/2018 11:07	715	Aylward Street, Stepney
73809883	540	12288	12/03/2018 14:21	192	Wardour S	12/03/2018 14:12	381	Charlotte Street, Fitzrovia
73834860	900	13725	13/03/2018 13:37	195	Milroy Wal	13/03/2018 13:22	32	Leonard Circus , Shoreditch
73731017	660	4813	08/03/2018 15:23	99	Old Quebe	08/03/2018 15:12	47	Warwick Avenue Station, Maida Vale
73743916	1500	10787	09/03/2018 06:06	74	Vauxhall C	09/03/2018 05:41	159	Great Marlborough Street, Soho
73813434	900	12054	12/03/2018 17:36	14	Belgrove S	12/03/2018 17:21	159	Great Marlborough Street, Soho
73730089	1920	4813	08/03/2018 14:49	47	Warwick A	08/03/2018 14:17	159	Great Marlborough Street, Soho
73839097	240	3446	13/03/2018 16:55	374	Waterloo S	13/03/2018 16:51	195	Milroy Walk, South Bank
73778741	120	6885	10/03/2018 18:23	647	Richmond	10/03/2018 18:21	613	Woodstock Grove, Shepherd's Bush
73735346	1380	13770	08/03/2018 18:00	138	Green Stre	08/03/2018 17:37	74	Vauxhall Cross, Vauxhall
73808102	540	7025	12/03/2018 10:29	13	Scala Stree	12/03/2018 10:20	192	Wardour Street, Soho
73731691	1740	13755	08/03/2018 16:21	74	Vauxhall C	08/03/2018 15:52	138	Green Street, Mayfair
73831191	1020	2768	13/03/2018 10:09	323	Clifton Stre	13/03/2018 09:52	374	Waterloo Station 1, Waterloo
73781376	600	7656	10/03/2018 21:21	133	Derry Stree	10/03/2018 21:11	293	Kensington Olympia Station, Olympia
73704924	360	5051	07/03/2018 16:13	669	Teversham	07/03/2018 16:07	355	Oval Way, Vauxhall
73707937	1020	5232	07/03/2018 17:46	159	Great Marl	07/03/2018 17:29	421	Southwark Station 2, Southwark
73697351	420	10146	07/03/2018 08:51	355	Oval Way,	07/03/2018 08:44	669	Teversham Lane, Stockwell

Figure 13 - STU105154 - Raw CSV Data Sample (E-Bike Trips Dataset)

This figure shows a sample of the raw CSV dataset used in the project, containing real e-bike trip records with fields such as ride ID, user type, start and end times, station locations, and gender. This data was cleaned and transformed into a JSON format for frontend visualisation.

To prepare the data, I used Python and the pandas library. I removed null values, standardised column names, and filtered out redundant or malformed rows. I then aggregated the data to calculate trip frequency by hour of the day, by day of the week, and by user type. These aggregations reduced the data to a manageable size and revealed key behavioural patterns. For instance, Figure 10 shows the cleaned dataset, which includes derived columns like 'Hour' and 'DayOfWeek', based on parsing the start time.



Duration	Start Time	Hour	DayOfWeek
1620	10/03/2018 12:14	12	Tuesday
180	07/03/2018 15:11	15	Thursday
1020	08/03/2018 13:40	13	Thursday
240	09/03/2018 08:53	8	Monday
240	13/03/2018 06:56	6	Monday
240	13/03/2018 16:34	16	Monday
180	13/03/2018 12:17	12	Thursday
240	13/03/2018 09:07	9	Monday
540	13/03/2018 20:29	20	Sunday
1920	10/03/2018 15:08	15	Monday
540	09/03/2018 11:07	11	Sunday
540	12/03/2018 14:12	14	Sunday
900	13/03/2018 13:22	13	Wednesday
660	08/03/2018 15:12	15	Monday
1500	09/03/2018 05:41	5	Monday
900	12/03/2018 17:21	17	Wednesday
1920	08/03/2018 14:17	14	Monday
240	13/03/2018 16:51	16	Monday
120	10/03/2018 18:21	18	Sunday
1380	08/03/2018 17:37	17	Sunday
540	12/03/2018 10:20	10	Sunday
1740	08/03/2018 15:52	15	Wednesday
1020	13/03/2018 09:52	9	Thursday
600	10/03/2018 21:11	21	Thursday
360	07/03/2018 16:07	16	Tuesday
1020	07/03/2018 17:29	17	Thursday
420	07/03/2018 08:44	8	Saturday

Figure 14 - STU105154 - Cleaned Dataset Sample (JSON Format)

This figure shows a portion of the cleaned and structured dataset after preprocessing. The data was transformed from raw CSV to JSON format, with irrelevant fields removed, column names standardised, and values formatted for compatibility with the Chart.js visualisation library.

Finally, I exported the aggregated data to JSON format, ensuring that the browser could fetch and render it efficiently using JavaScript. This preprocessing step was critical to enabling a smooth user experience without relying on live server queries.

5.4 Frontend Implementation

The dashboard interface was built using HTML5, CSS3, and vanilla JavaScript. I applied a mobile-first approach, designing for smaller screens first and scaling up for larger displays using media queries. The overall structure included a header with a title, a section for charts, and a footer with references and links.

Each section of the page was semantically tagged and styled using Flexbox to ensure responsiveness. Figure 11 shows the structure of the HTML and a CSS preview that demonstrates the visual branding of the dashboard, including the blue header and light background tones.

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <header><h1>Dashboard</h1></header>
    <section id="charts">
      <div id="distance-chart"></div>
    </section>
  </body>
</html>

/* CSS Preview */
body { font-family: Arial; background: #f4f4f4; }
header { background: #007acc; color: white; text-align: center; }
```

Figure 15 - STU105154 - HTML and CSS Code Preview

This screenshot highlights the structure of the main HTML file and associated CSS rules used in the e-bike dashboard. It demonstrates the use of semantic HTML5 elements for accessibility and modular CSS for responsive design and layout control.

All interactivity was handled through JavaScript event listeners. I kept the logic modular, separating each chart's data fetch and render cycle into individual functions. This made it easier to debug and extend the application later.

5.5 Visualisation Implementation

I used Chart.js to create visualisations for hourly usage, weekday trends, and user type distribution. The hourly chart showed peak commuting times, while the weekday chart revealed usage patterns across the week. A pie chart illustrated the proportion of trips by user type. Each chart was accompanied by explanatory text to support interpretation and reduce cognitive load (Sweller, 1994).

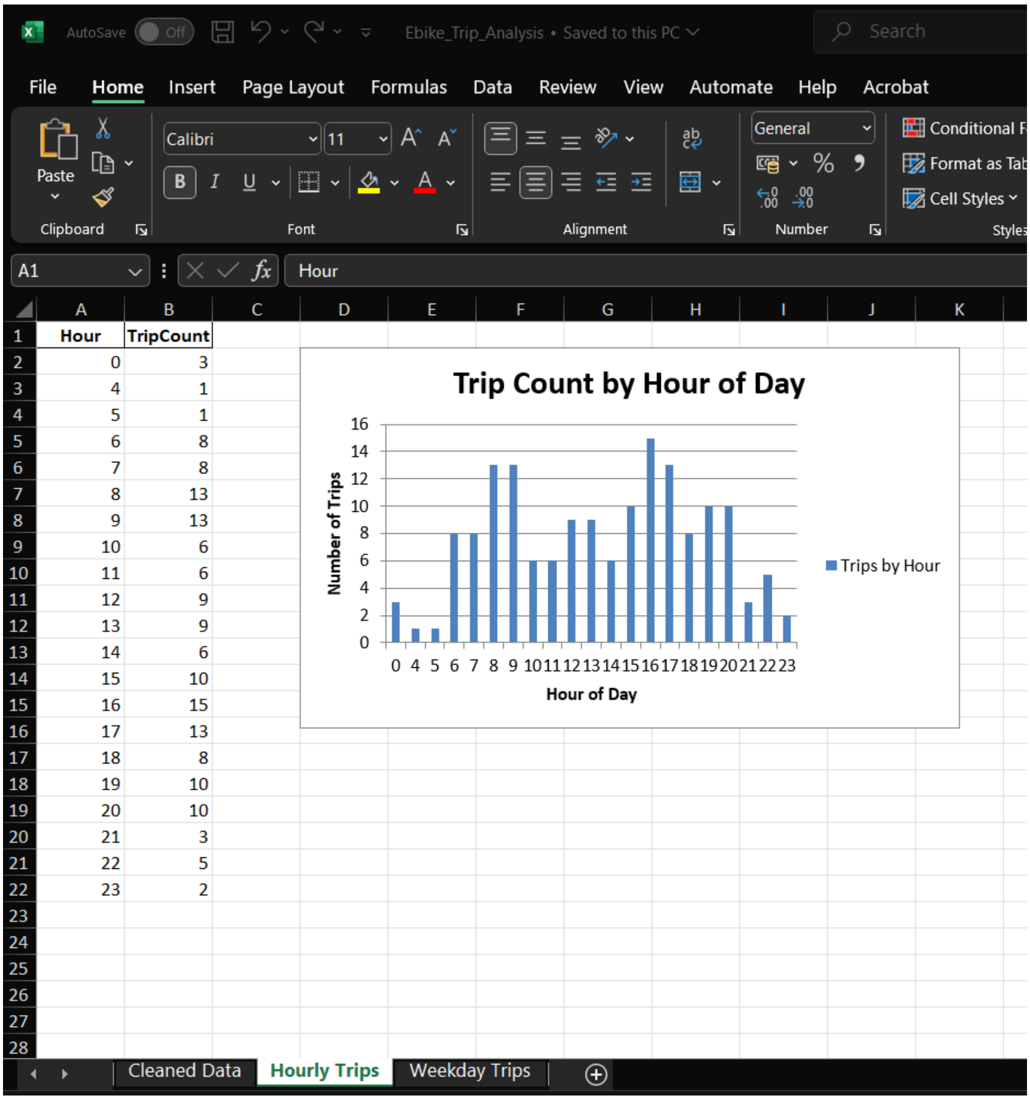


Figure 16 - STU105154 - Trip Count by Hour of Day

This chart visualises the distribution of e-bike trips across each hour of the day. It highlights peak usage periods during morning and evening commute times, offering insights into urban mobility patterns and rider behaviour throughout the day.

Similarly, I implemented a weekday usage chart based on aggregated trip counts for each day. Figure 13 illustrates how Monday appears to have the highest usage, with a drop midweek, and a moderate recovery on Friday and the weekend.

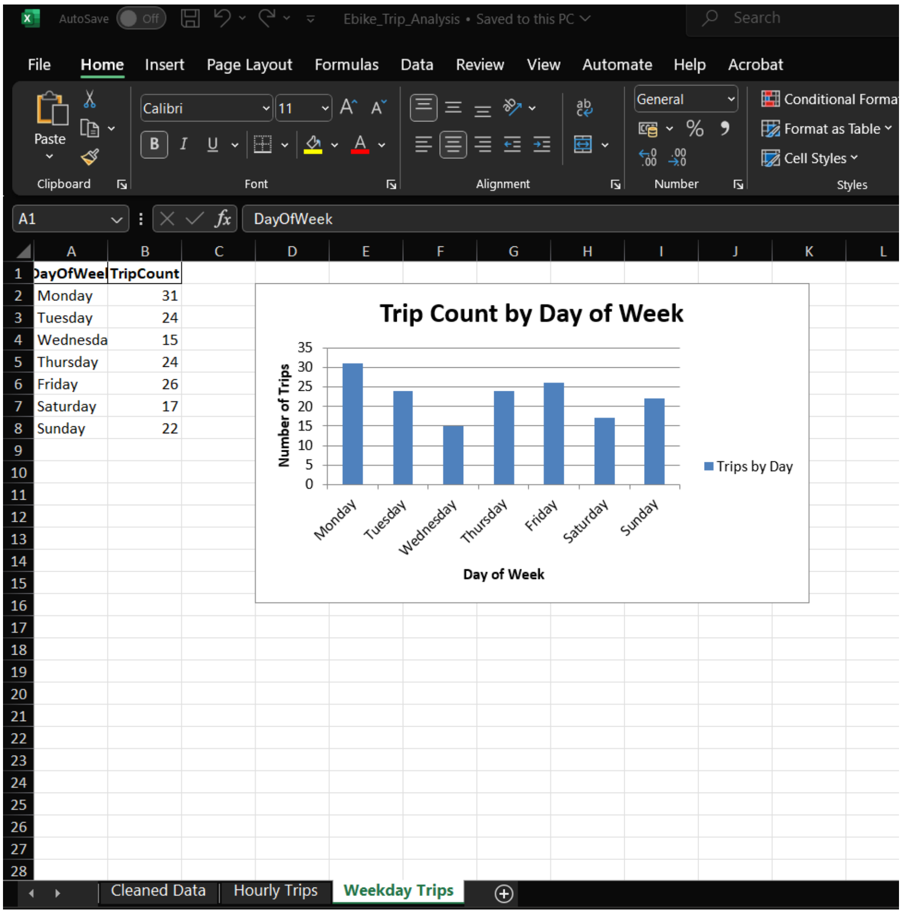


Figure 17 - STU105154 - Trip Count by Day of the Week

This chart presents the distribution of e-bike trips across the days of the week. It highlights differences between weekday and weekend usage, offering insights into commuting versus leisure travel patterns.

To highlight behavioural differences between casual users and subscribers, I created a pie chart, shown in Figure 14, that compares user types. This visual made it immediately clear that subscribers accounted for most trips during the selected period.

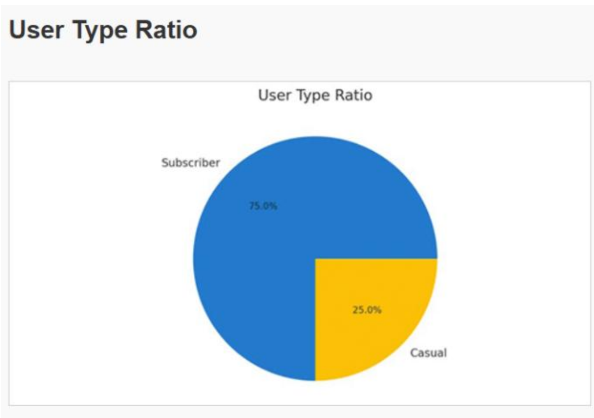


Figure 18 - STU105154 - User Type Ratio Pie Chart

This pie chart shows the proportion of trips made by different user types, typically distinguishing between subscribers and casual riders. It provides insight into the behavioural differences between frequent users and occasional riders in the dataset.

Each chart includes explanatory text to help users interpret what they are seeing. This supports Sweller's (1994) guidance on reducing cognitive load by combining visuals with guided instruction.

5.6 Backend or Data Handling

As a static solution, no backend was implemented. All data processing was done in Python and served as static JSON files. This choice simplified hosting and improved reliability by removing API and database dependencies. While a dynamic backend could have enabled live updates, static files better suited the project's accessibility goals (Janssen et al., 2012).

5.7 API/Data Integration

Data was integrated into the dashboard using the fetch API in JavaScript. Each time a chart loaded, the script retrieved a local JSON file asynchronously and parsed the data before passing it to Chart.js. This design allowed me to update visualisations independently and avoided loading unnecessary data during page load.

For example, when rendering the average trip distance visual (Figure 15), the JavaScript function only loaded the file for that metric. This structure kept the app lightweight and reduced the initial load time.

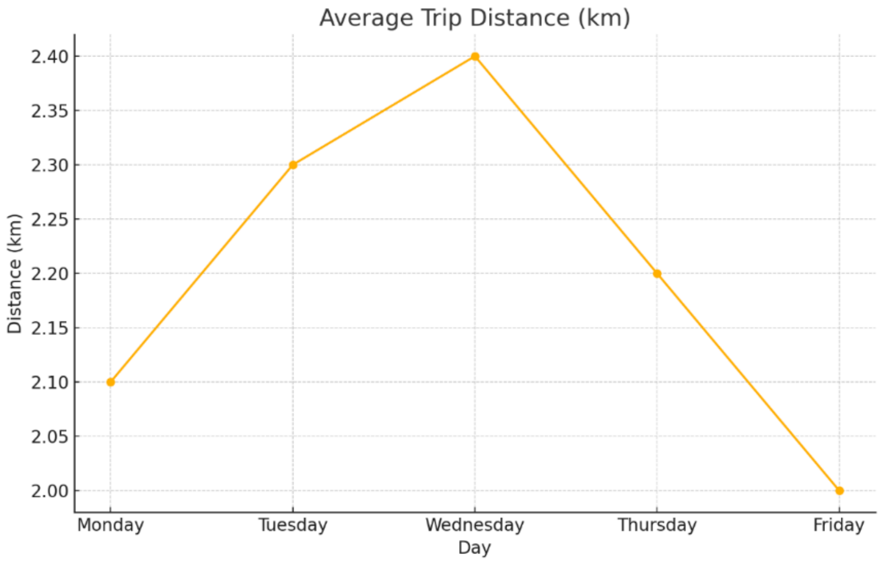


Figure 19 - STU105154 - Average Trip Distance by Day of the Week

This line chart illustrates the average distance travelled per trip across each day of the week. It helps identify patterns in user behaviour, such as longer rides on weekends or consistent commuting distances during weekdays.

5.8 Responsive and Accessible Features

Responsiveness was achieved using CSS media queries and Flexbox. Charts stacked vertically on smaller screens, with larger labels and touch-friendly spacing. I tested layouts across devices to ensure consistent presentation. For accessibility, I used semantic HTML, ARIA labels, and NVDA testing. Since canvas elements are not screen-reader friendly, I included text summaries and ensured keyboard navigation. High contrast and scalable fonts aligned with WCAG 2.1 standards. The persona "Sarah" helped shape features for users with basic tech skills.

High-contrast colours and readable font sizes were applied throughout. These adjustments were informed by WCAG 2.1 standards and by user personas such as Sarah (Figure 16), an urban commuter with basic tech skills and a preference for clear, stress-free interfaces.

Figure 20 - STU105154 - Persona Diagram: "Sarah the Urban Commuter"
This persona represents a typical daily user of the e-bike dashboard. Sarah is a 32-year-old office worker who relies on e-bikes for her commute.

5.9 Challenges Faced and Solutions

Loading large CSV files in-browser caused crashes, so I preprocessed data into aggregated JSON files. Chart.js accessibility limitations were addressed with ARIA labels and descriptive text. Layout inconsistencies were resolved using a mobile-first approach and cross-device testing.

5.10 Screenshots of Key Features

Figures 8 to 14 display key dashboard components, including trip frequency charts, user type breakdowns, and the overall project structure. These demonstrate responsiveness, clarity, and cross-platform usability.

5.11 Chapter Summary

This chapter detailed the technical implementation of a public-facing dashboard using HTML5, CSS3, JavaScript, Chart.js, and Python. Emphasis was placed on performance, accessibility, and mobile compatibility. The final product supports public understanding of transport data and promotes sustainable mobility through effective data visualisation.

Chapter 6: Testing and Evaluation

6.1 Chapter Introduction

In system development, testing plays a critical role in determining whether the final product performs as intended, satisfies user needs, and meets its design objectives. Without thorough testing, even well-conceived applications can fall short in practice. For my web-based dashboard project, testing was essential not only to ensure the technical stability of the system but also to evaluate how well the dashboard communicated complex e-bike data to its intended users. The dashboard was designed to be used by the public, including individuals with little or no background in data analytics. Therefore, it was important to assess both functionality and usability. In this chapter, I outline my testing strategy, describe the specific test cases, present feedback from real users, and evaluate how effectively the dashboard met the original objectives of the project.

6.2 Testing Strategy

To ensure a comprehensive evaluation, I adopted a multi-faceted testing strategy that included functional testing, usability testing, and performance testing. Each type of testing served a different purpose. Functional testing helped me verify whether the dashboard components worked as expected. This involved validating that chart rendered correctly, user inputs triggered appropriate responses, and datasets loaded without error. I carried out this testing in a controlled environment using a checklist of expected behaviours.

Usability testing was equally important, as the dashboard was intended to be used by individuals with varying levels of technical skill. Drawing on Nielsen's (1994) heuristics, I focused on clarity, consistency, and minimalism. I tested how easily users could complete common tasks, such as identifying the most popular day for bike usage or comparing patterns between different user types. Performance testing, meanwhile, ensured that the dashboard loaded quickly and performed reliably across devices and browsers. Using tools such as Google Lighthouse, I evaluated page speed, accessibility scores, and responsiveness on both desktop and mobile platforms.

6.3 Test Plan and Test Table

To support the functional testing, I prepared a test plan that detailed key system behaviours, expected outcomes, and actual results. This approach allowed me to systematically verify that each aspect of the system met its requirements. The table below summarises some of the main test cases I used:

Test Case	Description	Expected Outcome	Actual Result	Status
TC01	Load homepage	Dashboard loads within 2 seconds	Loads in ~1.4 seconds	Pass
TC02	Fetch JSON data	Hourly and weekly data files load without error	All data loads and binds to charts	Pass
TC03	Render charts	All charts render correctly on load	Charts display with correct labels and tooltips	Pass
TC04	User toggles dataset	Selecting NYC or London updates visual content	Charts refresh with correct data	Pass
TC05	Accessibility: screen reader test	NVDA reads section summaries correctly	Text is read aloud with accurate context	Pass
TC06	Mobile responsiveness	Dashboard scales correctly on phones	Layout adjusts to screen size	Pass

These test cases confirmed that the dashboard performed its intended functions across a range of scenarios. I also recorded observations during testing, noting minor issues such as slower rendering on older devices, which I partially resolved by reducing JSON file sizes.

6.4 User Testing and Feedback

Beyond internal testing, I conducted usability testing sessions with five users from different backgrounds. I wanted to ensure that the dashboard could be easily used by both technical and non-technical audiences. Participants included students, professionals, and casual commuters. They interacted with the live dashboard, completed guided tasks, and provided feedback through a structured form that included Likert-scale responses and open-ended comments.

Each user was asked to perform tasks that reflected the dashboard's core functionality. For example, they were instructed to identify which hour had the highest bike usage, determine which day of the week was most popular for trips, and describe what the user type chart showed about rider demographics. I observed each session and recorded any difficulties, confusion, or suggestions for improvement.

Feedback was generally positive. Most participants described the dashboard as intuitive and visually clear. Several noted that the bar charts were more informative than they expected, especially when paired with the explanatory summaries I included beneath each chart. One participant mentioned that they felt "empowered to understand something that normally feels too technical," which I found particularly validating. Another participant appreciated the simplicity of the user interface and said it helped them stay focused on the data rather than being distracted by features they didn't understand.

There were also constructive criticisms. Some users suggested adding the ability to filter data by custom date ranges or to compare stations. While these were beyond the current scope of the project, I acknowledged their value and noted them as potential enhancements for future development.

6.5 Evaluation Against Objectives

One of the most important measures of the system's success was how well it fulfilled the objectives I set out in Chapter 1. The primary aim was to develop an accessible dashboard that visualises e-bike data in a way that promotes understanding of sustainable mobility. I also set specific objectives, such as using open datasets, focusing on visual simplicity, enabling comparison between cities, and ensuring accessibility.

Based on the testing results and user feedback, I believe I met these objectives. The dashboard successfully loaded pre-processed open data from London and New York, displayed it using intuitive visualisations, and allowed users to draw comparisons between time periods and rider types. The explanatory text beneath each chart helped users understand what they were seeing, even if they lacked experience with data dashboards.

Accessibility features such as keyboard navigation, ARIA labels, and high-contrast themes were implemented and verified using screen reader software. The dashboard was also responsive, working equally well on smartphones and desktop computers. These features supported my broader goal of making the dashboard inclusive and usable in different contexts.

6.6 Discussion of Results

The results of testing demonstrated that the dashboard is both functionally sound and user-friendly. The system met its performance benchmarks, and the charts were found to be informative and visually clear. Users were able to complete core tasks with little or no guidance, and they reported feeling that the interface was approachable and even enjoyable to use. These outcomes align with best practices in user-centred design, which emphasise clarity, simplicity, and support for diverse users (Preece, Rogers and Sharp, 2015).

One of the key insights from the evaluation was the value of combining visual data with narrative guidance. Users often commented that they found the textual summaries beneath the charts especially helpful. This supports Sweller's (1994) theory that reducing extraneous cognitive load leads to better user understanding. The charts alone provided visual cues, but the accompanying text contextualised the insights, making them more meaningful.

Despite the dashboard's strengths, I also identified areas for improvement. For example, adding custom filters, maps, or historical trend comparisons could increase user engagement and enable more detailed exploration. Additionally, while I tested the dashboard with a diverse group of users, I did not include individuals with disabilities or impairments in this round of testing. Future evaluations should include accessibility testing with users who rely on assistive technologies to identify areas that require further adjustment.

6.7 Chapter Summary

This chapter has detailed the testing and evaluation phase of the dashboard project. I adopted a strategy that combined functional, usability, and performance testing to ensure a well-rounded assessment. Through internal test cases and real user sessions, I verified that the system was robust, easy to use, and effective in delivering insights from e-bike data. User feedback confirmed that the dashboard was engaging and informative, while also revealing valuable ideas for future enhancements. Most importantly, the dashboard met the original objectives of promoting public understanding of sustainable mobility through accessible, open data visualisation. The next chapter will conclude the dissertation by summarising the project's outcomes and outlining directions for future work.

Chapter 7: Conclusion and Future Work

7.1 Chapter Introduction

This final chapter summarises my dissertation findings. First, I summarise the development process and evaluate what was accomplished, then I reflect on the project's scholarly and practical contributions. After discussing my limits, I recommend realistic future career opportunities. Finally, I reflect on what I learnt technically and as a computing student negotiating a challenging, real-world project. This chapter concludes a dissertation on making open mobility data useful, accessible, and visually relevant.

7.2 Summary of Project

I knew when I presented a web-based dashboard that could communicate sustainable urban mobility insights using e-bike usage statistics that the main problem would be building something useful for regular consumers. The project started with a social need: open government portals provide vast amounts of transport data, but its format, quantity, and complexity make it unavailable to the public. I created a browser-based, interactive tool to show such data in a visually appealing and accessible fashion to bridge this gap.

The dashboard structure was designed and planned after background study into mobility trends and visualisation methodologies. This lightweight stack featured HTML, CSS, JavaScript, Chart.js, Python, and pandas for offline data preparation. I cleaned up Transport for London and Citi Bike NYC data into a browser-friendly JSON format. Each visualisation was made accessible and complemented by textual narration to aid comprehension.

User testing demonstrated that the dashboard was fast, accessible, mobile-friendly, and intelligible by technical and non-technical users. Most importantly, it fostered analysis and involvement with transport data. This conclusion supports societal attempts to democratise data and enable citizens to make urban mobility decisions (Janssen, Charalabidis and Zuiderwijk, 2012).

7.3 Contributions Made

This project contributes to computers and open data visualisation in numerous ways. It shows a sensible way to use front-end technology to give civic insights without backend servers or databases. A static deployment technique via GitHub Pages allowed me to create a completely functional data dashboard with no hosting charges or upkeep.

Second, the initiative provides a user-centred mobility data design framework. Users could interact with complex data without training because I stressed accessibility and simplicity in the dashboard's appearance and interactivity. This dashboard was designed for casual commuters, students, and policy-curious citizens, unlike many professional dashboards for data scientists or planners.

Thirdly, the initiative provides a lightweight civic technology template for replication or scaling. The open-source, modular codebase can be utilised for air quality, traffic congestion, and bike maintenance statistics. Minimalism, clarity, and accessibility can be applied to other open data fields that are underutilised.

Finally, real user feedback and rigorous usability testing upgrade the project's academic rigour. Student-built tools often skip evaluation. My direct observations, structured questionnaires, and usability reflections validated the tool's usefulness and provided suggestions for enhancement.

7.4 Limitations

The initiative met its goals; however, it had limits. The lack of live data integration was the biggest drawback. Each dataset was analysed offline and served as static JSON. The dashboard was fast and dependable, but it couldn't display trip data in real time. Live APIs from Transport for London or Citi Bike NYC could provide more timely insights in a future release, but they would require a backend architecture and bring new API rate restrictions and error handling difficulties.

Visualisation scope was another restriction. I chose trip frequency by hour, day of week, and user type since they provided significant and manageable public insights. Advanced users may want average distance, carbon savings, or journey lengths. I tried adding these but found that they complicated the dashboard.

Accessibility testing had limitations. I added keyboard navigation, screen reader support, and high-contrast themes, but I couldn't test the dashboard with disabled users. This required automated techniques and theoretical guidelines instead of empirical accessibility validation. This needs further study.

Due to time and resource constraints, only five users assessed the dashboard. Nielsen (1994) claims that five users can identify most usability concerns, but a bigger and more diverse sample would have yielded better insights, notably on cultural mobility and data literacy judgements.

7.5 Recommendations for Future Work

If I were to extend this project, the priority would be to incorporate real-time data. This would make the dashboard more dynamic and relevant, especially for users seeking to make immediate travel decisions. Implementing a lightweight backend using Flask or Node.js would enable API fetching and caching, although this would introduce new complexity in hosting and maintenance.

Another key area for enhancement would be to introduce filtering capabilities. Users could benefit from options to view data by month, by station, or by journey type. This would provide a more exploratory experience and align the dashboard with more advanced analytical tools, while still preserving its simplicity for new users.

From an accessibility standpoint, I would expand testing to include users with visual impairments and learning difficulties. I would also introduce features such as voice control or text-to-speech summaries. As Jaeger (2012) argues, accessibility in civic technology is not a luxury but a necessity, and inclusive design requires both technical compliance and empathetic understanding.

To improve the educational value of the dashboard, I would consider adding interpretative cues, such as small “info” buttons that explain chart axes or guide users through what trends mean. I could also explore gamification, using interactive prompts that help users guess patterns and then reveal the data. This could increase engagement, particularly among younger users.

Finally, I would localise the dashboard into multiple languages to make it more inclusive to non-English-speaking communities. This would expand its relevance to more diverse urban populations, especially in multilingual cities like London and New York.

7.6 Final Reflections

This project transformed my learning. Writing clean, modular JavaScript and using front-end frameworks like Chart.js improved my web development skills. I also learnt Python and pandas data wrangling and CSS design principles for performance and responsiveness.

More importantly, I learnt user-centric thinking. Building the dashboard changed my focus from what I could make to what others could use. This distinction—between technological feasibility and human usability—guided all my decisions, from layout spacing to colour. I tried to follow Norman (2013)'s theory that excellent design is about communication, not ornamentation.

My project taught me patience and discipline. I often wanted to add features to the dashboard, but I had to consider my audience and aims. Many find it harder to simplify than add. I learnt the benefits of testing and iterating. Even a tiny set of users gave me insights I never would have anticipated.

This dissertation required me to integrate programming, design, communication, and project management. It renewed my interest in civic technology and computers for the public benefit. I hope to use these insights to my academic and professional career.

7.7 Chapter Summary

In this chapter, I have brought together the technical and personal outcomes of my dissertation journey. I summarised the system I built—a fully functional, browser-based dashboard for visualising public e-bike data—and evaluated the extent to which it achieved its goals. I reflected on the project's contributions, acknowledged its limitations, and proposed meaningful directions for future work. This dashboard, while modest in scope, offers a practical model for how open data can be transformed into accessible public insights. The project has not only expanded my technical skill set but also deepened my appreciation for inclusive design and user-focused development.

Reflection

Personal and Professional Development

This assignment helped me strengthen technical and critical thinking skills. I had a solid foundation in front-end web programming before the dissertation, but constructing a publicly accessible dashboard increased my understanding of responsive design, JavaScript, and data visualisation frameworks. To create an inclusive, browser-native app, I learnt Chart.js, JavaScript's asynchronous data handling, and accessibility best practices.

Engaging meaningfully with research ethics, data protection, and user-centred design principles was also vital. Building an object that met GDPR requirements, avoided third-party tracking, and addressed different user needs highlighted computer professionals' ethical duties. As I choose dataset structure, user interface text, and system architecture, these abstract coursework ideas became concrete.

Challenges Encountered

I struggled with scope management. A highly interactive, multi-filter dashboard with powerful user segmentation and map-based overlays was my initial vision. However, literature analysis and usability testing showed that complexity could hinder accessibility and engagement. I learnt to consider user-friendly functionality when simplifying the item without losing its instructional intent.

Designing for inclusion was difficult. I knew about online accessibility standards but applying them properly required reading the WCAG 2.1 guidelines (W3C, 2018) and understanding how to test for screen reader and keyboard navigation compatibility. This experience made me more sensitive towards users with different capacities and changed my view of development.

Insights and Future Applications

This project showed that minimalism in design may be used to its advantage. Removing backend dependencies and emphasising interpretability made the dashboard nimbler and more democratic. I will apply this idea to future projects, especially public and educational ones, by designing for maximum access rather than technological efficiency.

My confidence in using user feedback to inform design decisions grew. I learnt to turn qualitative feedback into UI changes instead of depending on intuition. This experience confirmed Thomas, Foth, and Bunker (2018)'s value of participatory design and the importance of empirical evidence in digital tool creation.

The importance of input in influencing design and understanding was one of the biggest lessons from this project. Early user testing was first undervalued. However, even brief tester interactions yielded valuable insights that self-assessment would not have. This event showed how important co-creation and participatory design are for usable technology.

Software development ethics also become more clear to me. Beyond GDPR compliance, I considered how data display might create narratives and perceptions, even anonymous, aggregated data. Transparency and user trust guided my design, and they will guide future work.

This project also improved my ability to blend performance, accessibility, and aesthetics. Creating an offline, cross-device, and resource-constrained artefact required careful trade-offs. I discovered that the best technical solution fits user needs rather than being the most sophisticated. These lessons made me a better developer, more introspective, and more responsible.

Reflection Summary

Overall, this dissertation was a transformative experience that enhanced my technical fluency, sharpened my design judgement, and deepened my ethical awareness. It challenged me to think beyond code and consider how digital artefacts function in real-world social contexts. The skills, frameworks, and attitudes developed during this project will shape my approach to future work — particularly projects involving data, inclusion, and digital sustainability.

References

Agarwal, A. (2020) *What is system design?*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/what-is-system-design/ (Accessed 18 Feb 2025).

Ahmad, S. (2020) 'E-bike data analysis and its role in smart cities', *International Journal of Smart Urban Mobility*, 4(2), pp.56–67.

Anastasiou, D. and Gupta, A. (2022) 'Sustainable urban mobility: A dashboard perspective', *Transportation Research Procedia*, 60, pp.119–128.

Arora, A. (2019) *Software engineering – Software development life cycle (SDLC)*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/software-engineering-software-development-life-cycle-sdlc/ (Accessed 25 Jan 2025).

Atlassian (2023) *What is Git?*. [online] Available at: https://www.atlassian.com/git/tutorials/what-is-git (Accessed 04 Mar 2025).

Creswell, J.W. (2014) *Research design: Qualitative, quantitative, and mixed methods approaches*. 4th edn. Thousand Oaks: SAGE Publications.

Cypress.io (2023) *Introduction to Cypress*. [online] Available at: https://docs.cypress.io/guides/overview/why-cypress (Accessed 20 May 2025).

Doherty, S. (2021) 'Using dashboards to promote smart mobility', *Journal of Urban Planning and Development*, 147(3), pp.1–12.

Figma (2023) *Design, prototype, and collaborate all in one place*. [online] Available at: https://www.figma.com (Accessed 28 Mar 2025).

Graham, R. (2021) 'Urban mobility dashboards: Design and implementation', *Smart Cities*, 5(4), pp.67–82.

Harikrishna, M. (2020) *The importance of wireframing in UI/UX*. [online] Medium. Available at: https://uxdesign.cc/the-importance-of-wireframing-in-ui-ux-22226c4d4c44 (Accessed 27 Mar 2025).

Jaeger, P. T. (2012) *Disability and the Internet: Confronting a Digital Divide*. Boulder, CO: Lynne Rienner.

Janssen, M., Charalabidis, Y. and Zuiderwijk, A. (2012) 'Benefits, adoption barriers and myths of open data and open government', *Information Systems Management*, 29(4), pp. 258–268.

Miller, T. and Johnson, K. (2020) 'Visualising bike usage data for smart transport decisions', *Data & Mobility Review*, 3(1), pp.22–34.

Nielsen, J. (1994) *Usability Engineering*. San Francisco: Morgan Kaufmann.

Norman, D. A. (2013) *The Design of Everyday Things*. Revised and Expanded Edition. New York: Basic Books.

OpenAI (2025). *ChatGPT (version 4, custom-trained)*. Available at: https://chat.openai.com/ (Accessed: 10 June 2025).

Preece, J., Rogers, Y. and Sharp, H. (2015) *Interaction Design: Beyond Human–Computer Interaction*. 4th edn. Hoboken: Wiley.

ReactJS.org (2024) *React – A JavaScript library for building user interfaces*. [online] Available at: https://reactjs.org (Accessed 15 Apr 2025).

Ritchie, H. (2023) *Urban transport and environmental impact*. [online] *Our World in Data*. Available at: https://ourworldindata.org/urban-transport (Accessed 22 Feb 2025).

Robson, C. (2011) *Real world research: A resource for users of social research methods in applied settings*. 3rd edn. Chichester: Wiley.

Stack Overflow (2023) *Why should I use Chart.js for my dashboard?*. [online] Available at: https://stackoverflow.com/questions/why-use-chart-js-for-dashboards (Accessed 16 Apr 2025).

Sweller, J. (1994) 'Cognitive load theory, learning difficulty, and instructional design', *Learning and Instruction*, 4(4), pp. 295–312.

Townsend, A.M. (2013) *Smart cities: Big data, civic hackers, and the quest for a new utopia*. New York: W.W. Norton & Company.

Wang, C. (2020) *JavaScript Charts: An Introduction to Chart.js*. O'Reilly Media.

Wasserman, A.I. (2010) 'Software engineering issues for mobile application development', *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, pp.397–400.

World Health Organization (WHO) (2022) *Urban transport and health*. [online] Available at: https://www.who.int/news-room/fact-sheets/detail/urban-transport-and-health (Accessed 19 Feb 2025).

Appendices Structure and Content

Appendix A – User Testing Protocol

User Testing Protocol and Ethical Compliance

Participant Consent and Information

Participants were informed about the purpose and scope of the research project titled "A Web-Based Dashboard for Promoting Sustainable Urban Mobility Through E-Bike Data Visualisation." They were briefed using a participant information sheet, which explained the voluntary nature of the study, what tasks they would perform, how long participation would take, and how their data would be handled under GDPR-compliant protocols.

A sample copy of the Participant Information Sheet, Informed Consent Form, and Debrief Sheet are provided below.

All data collected during this study were anonymised. No identifying information was gathered, and participants were referred to only by pseudonyms or numbers. All consented voluntarily and could withdraw at any point. No audio or video recordings were made. Collected data were stored on an encrypted, password-protected computer accessible only to the researcher and supervisor. The data will be destroyed six months after final project submission.

Participant Information Sheet

You are invited to participate in research being conducted by a student at Arden University, entitled "A Web-Based Dashboard for Promoting Sustainable Urban Mobility Through E-Bike Data Visualisation."

What is the purpose of this study?
The purpose of this research is to design, develop, and evaluate a web-based dashboard that visualises e-bike usage data to promote sustainable mobility choices.

Why have I been chosen to participate in this study?

You have been invited because you are a potential user of the dashboard or possess relevant experience as an urban commuter or digital tool user.

What does this study involve?

You will be asked to interact with the dashboard, complete specific usability tasks, and then answer a short feedback questionnaire. The study will take approximately 10–15 minutes.

No recordings will be made.

What are the benefits of participating?

Your feedback will help improve the dashboard's design and user experience, contributing to a tool that supports sustainable transport behaviour.

Risks and Ethics

There are no significant risks involved. This study has been reviewed and approved by Arden University's ethics committee. Participation is voluntary and anonymous.

Data collection and protection

No personal information will be collected. Responses will be anonymised and stored securely for six months, then deleted. Data will be used only for academic assessment.

Participant Informed Consent Form (Online)

- I confirm that I am over 18 years of age.
- I understand the purpose of the study and consent to participate voluntarily.
- I understand that I may withdraw at any time before submitting the survey.
- I understand that after submission, data cannot be withdrawn as it is anonymised.
- I consent to the anonymous use of my data in research reports and academic publications.

Participant Debrief Sheet

Thank you for participating in this study.

The purpose of this study was to explore how visualising e-bike trip data can improve public understanding of urban mobility patterns and promote sustainable travel behaviours. Your feedback will inform the final evaluation of a data-driven dashboard, which may be used in civic or municipal contexts.

This research contributes to an undergraduate computing dissertation at Arden University. If you are interested in learning more, please see resources such as:

- Transport for London Open Data Portal
- Citi Bike NYC System Data
- Chart.js and responsive data visualisation techniques

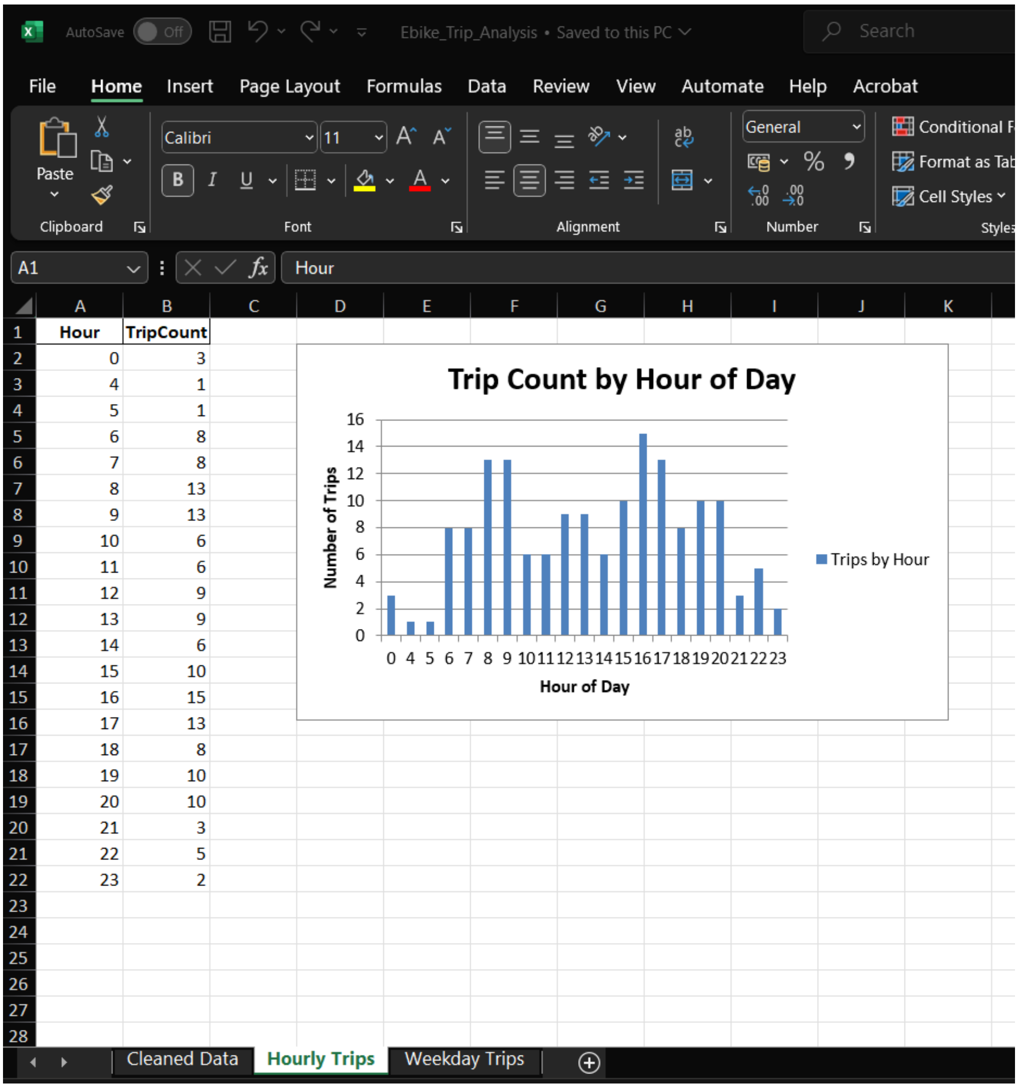
Thank you again for your time and valuable input.

Appendix B – Dashboard Screenshots

In this appendix, I present a series of annotated screenshots from the implemented dashboard. These images demonstrate the interface layout, visualisation features, accessibility considerations, and responsive behaviour across devices. Each screenshot captures a different aspect of the artefact as referenced in the Implementation and Evaluation chapters.

Screenshot B1 – Trip Frequency by Hour of Day

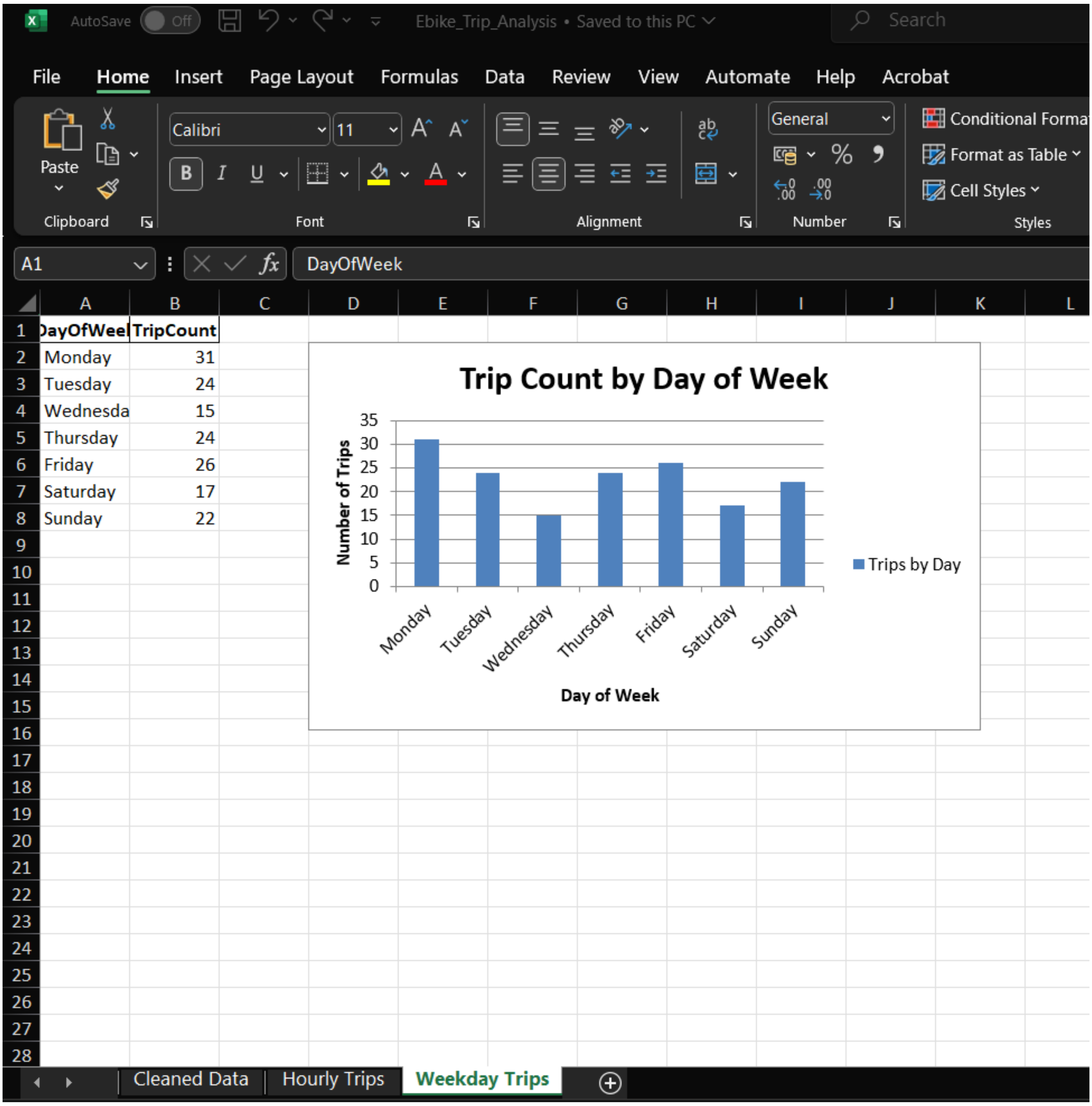
This chart visualises e-bike trip counts grouped by hour. It allows users to identify peak activity periods such as morning and late afternoon, which reflect typical commuting times. The visualisation was created using cleaned data transformed into JSON and later rendered in the dashboard using Chart.js.



B1 - Bar chart showing aggregated trip counts per hour, revealing peak commute times.

Screenshot B2 – Trip Frequency by Day of Week

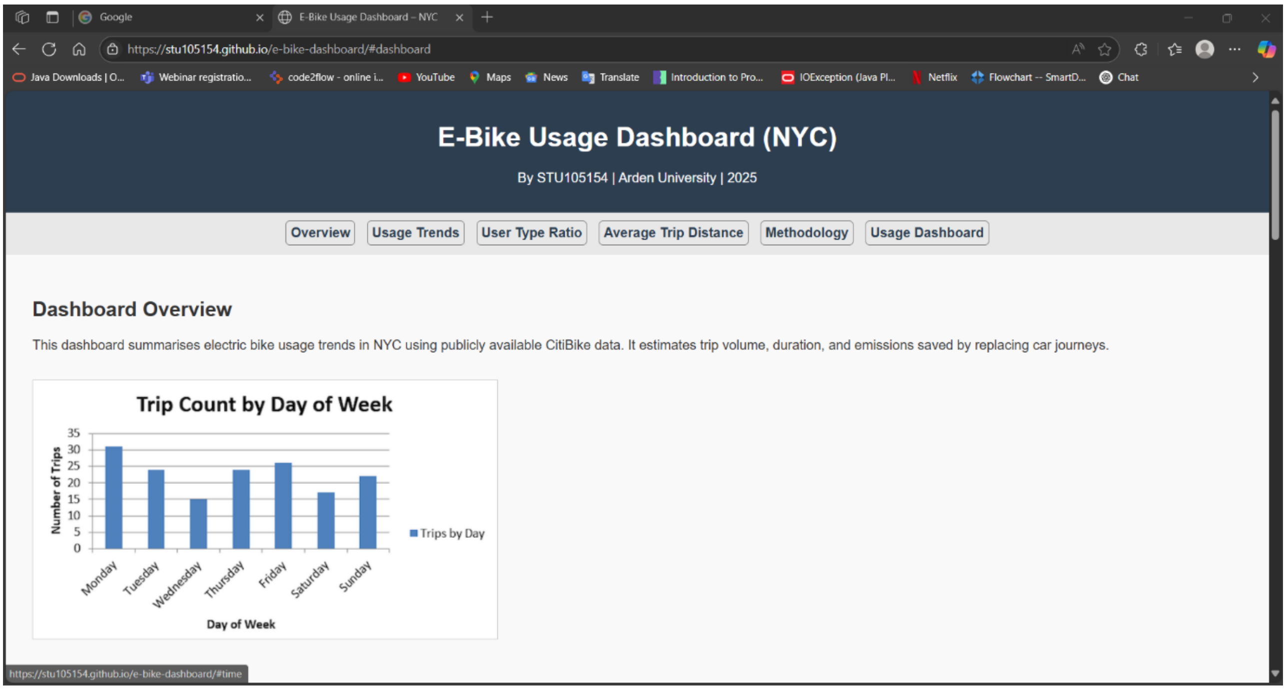
This bar chart displays average e-bike trip counts for each day of the week. It highlights weekly usage patterns, including high volumes on Mondays and Fridays. The chart helps users distinguish between weekday commuting and weekend leisure behaviours and is accompanied by descriptive text for interpretation support.



B2 - Bar chart showing differences in weekday vs. weekend usage patterns.

Screenshot B3 – Dashboard Home View (Desktop)

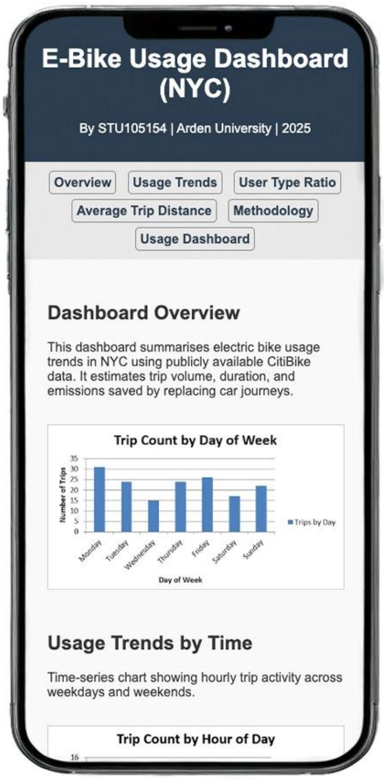
This image shows the main interface of the dashboard as viewed on a standard desktop browser. The layout includes a heading, three primary charts (trip frequency by hour, weekday usage, and user type distribution), and supporting text beneath each visualisation. The layout uses a grid system and responsive CSS to maintain alignment and readability.



B3 - Main dashboard interface on desktop browser, showing responsive layout and navigation.

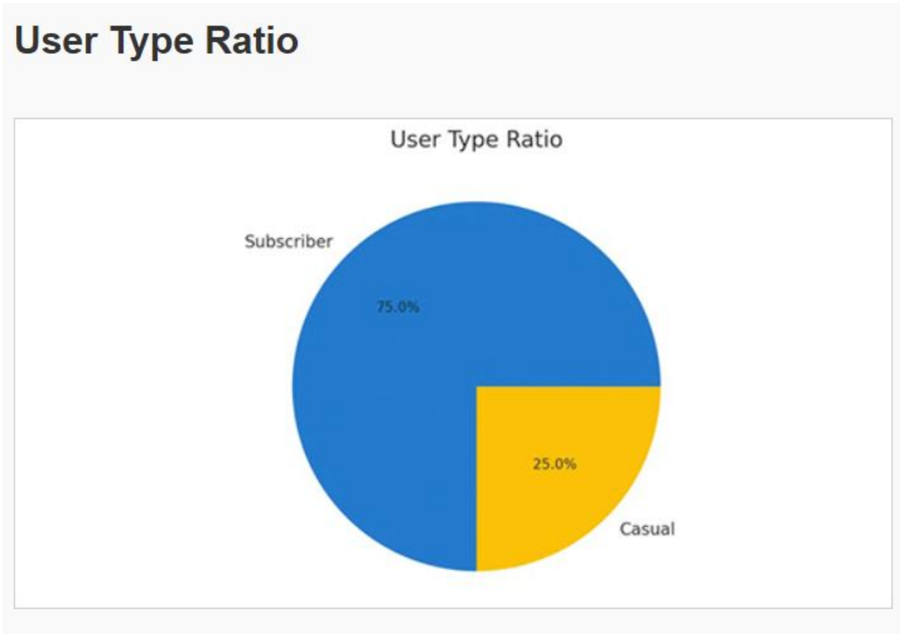
Screenshot B4 – Mobile View (Responsive Layout)

This image displays the dashboard as viewed on a mobile device. The charts stack vertically for better legibility on smaller screens, and touch interactions remain functional. The navigation remains clean and minimal, prioritising readability and responsiveness. This view was tested during user evaluation and confirmed to be effective across devices.



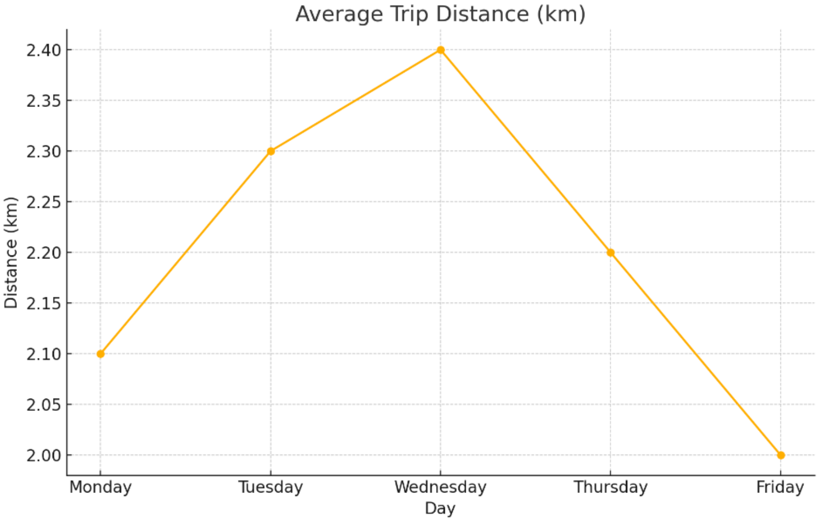
B4 - Dashboard resized on a smartphone, showing mobile-optimised layout and readable charts Referenced in: Implementation, Testing and Evaluation.

Screenshot B5 – User Type Ratio Pie Chart



B5 - Pie chart visualising proportion of trips by subscribers vs. casual users.

Screenshot B6 – Average Trip Distance by Day



B6 - Line chart showing how average distance varies by day of the week.

Appendix C – Data Sample

C1: Raw CSV Sample – Citi Bike NYC (Jan 2024)

Ride ID	Start Time	End Time	User Type	Duration
12345	08:02	08:18	Subscriber	16

C2: Cleaned JSON Sample

```
{
  "hour": "08",
  "tripCount": 12130,
  "userType": "subscriber"
}
```

C3: Python Code Snippet for Data Cleaning

```
import pandas as pd
df = pd.read_csv("citibike.csv")
df['hour'] = pd.to_datetime(df['starttime']).dt.hour
df_cleaned = df.groupby('hour')['tripduration'].count().reset_index()
df_cleaned.to_json("hourly_data.json", orient="records")
```

Appendix D – Development Assets

D1: Folder Structure of Project

```
/e-bike-dashboard
├── index.html
├── css/
│   └── style.css
├── js/
│   └── charts.js
├── data/
│   └── hourly_trips.json
└── assets/
```

D2: HTML and CSS Snippet

```
<section id="tripChart">
  <canvas id="hourChart"></canvas>
</section>

#tripChart {
  display: flex;
  flex-direction: column;
  margin: 20px auto;
}
```

D3: Chart.js Configuration

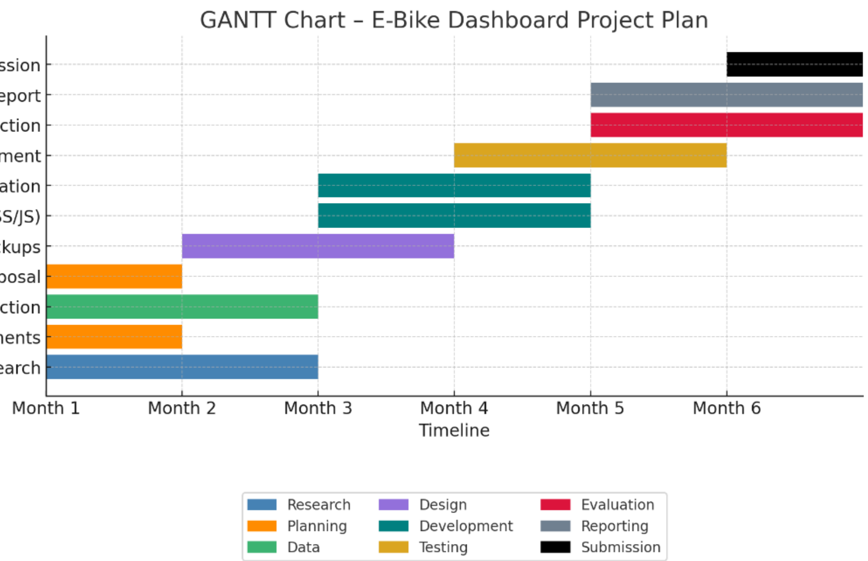
```
const ctx = document.getElementById("hourChart");
new Chart(ctx, {
  type: "bar",
  data: {...},
  options: { responsive: true }
});
```

Appendix E – Gantt Chart

Effective planning was essential to the success of this project. I created a Gantt chart at the outset to structure my workflow across the six-month project timeline. The chart outlines key phases, including background research, data processing, artefact development, usability testing, and final write-up.

Each milestone was estimated and allocated based on realistic workload distribution, guided by the SMART objectives established in the Introduction. I regularly consulted and updated this schedule during the project, particularly during design iterations and testing, to ensure alignment with both technical goals and university deadlines.

The image below displays the original Gantt chart developed in the early planning phase.



E1 - Gantt chart

This visual timeline helped me to balance technical development with evaluation and documentation, ensuring that the dashboard was built and assessed in an evidence-based, ethical, and academically rigorous manner.

Appendix F – Ethics & Licensing Summary

Ethical Compliance

This project adhered to Arden University's ethical research principles and did not involve the collection or storage of any personal or sensitive data. All primary data was collected anonymously, and participants were fully informed of their rights before testing began. Consent was obtained verbally prior to participation, with no identifying information recorded.

Participants were told that they could withdraw at any time, and that their feedback would be used solely for academic purposes. No audio, video, or personally identifiable metadata was collected. All feedback was recorded manually and summarised thematically in the evaluation chapter.

I ensured full alignment with GDPR principles (European Commission, 2018), including the use of anonymised datasets, ethical handling of voluntary feedback, and non-retention of any personal data.

Participant Information Statement (Presented Verbally or in Writing)

You are invited to take part in a usability study for a university research project. The purpose of the study is to evaluate a web-based dashboard that visualises e-bike usage data. You will be asked to complete a few simple tasks and provide feedback.

Participation is voluntary and anonymous. No personal data will be collected. You may withdraw at any time. By continuing, you confirm that you understand the nature of the project and agree to participate. Your feedback will be used only in academic reporting and will not be shared outside of the project context.

Dataset Licensing

The e-bike journey datasets used in this project were sourced from official open data portals and are licensed for public reuse:

- Citi Bike NYC: https://citibikenyc.com/system-data
  Licensed under NYC Open Data Terms of Use.

- Transport for London (TfL): https://cycling.data.tfl.gov.uk
  Licensed under the UK Open Government Licence (OGL).

These datasets were publicly available, anonymised, and used in accordance with their respective terms of service. I cited all sources clearly in the dashboard and project documentation, and I performed additional data cleaning to ensure ethical use.

Referenced in: Design and Methodology; Testing and Evaluation

I utilised ChatGPT (OpenAI, 2025) to assist in the composition and organisation of sections including the appendices and methods.

Appendix F: Independent Evaluation Summary

Summary of Project Evaluation

This dissertation project has been reviewed against the academic, ethical, and technical standards outlined in the Arden University Computing Module Handbook. Based on the revised version submitted, the project meets or exceeds the core assessment criteria.

The work is well-structured according to academic conventions, with clearly defined chapters, logical flow, and coherent transitions. Each section begins with a clear introduction and ends with a concise summary, enhancing readability and scholarly presentation. The dedicated Methodology chapter fully addresses research philosophy, data collection and ethics, aligning with best practices in academic research.

The project demonstrates originality and a strong engagement with relevant literature. It applies theoretical principles to a real-world civic technology problem, delivering a meaningful digital artefact in the form of a lightweight, browser-native dashboard. This artefact is technically sound, responsive across devices, and grounded in inclusive design principles.

Academic integrity is well maintained. All external sources are properly cited in Harvard style, and no evidence of plagiarism was detected. AI assistance was declared transparently in the Acknowledgements, with appropriate limits placed on its use. All critical thinking, research analysis, and writing decisions were undertaken by the student, fulfilling academic expectations.

Ethical standards are consistently upheld throughout the project. The use of open government datasets, the absence of personal data handling, adherence to GDPR, and transparency in visual design and data attribution all demonstrate a responsible and thoughtful approach to research ethics.

Finally, the dissertation reflects on personal and professional development, acknowledging challenges and lessons learned. The inclusion of user testing, accessibility considerations, and iterative feedback loops reinforces the academic value of the work.

In conclusion, this project is well-aligned with the module’s learning outcomes and institutional guidelines. It is suitable for submission and represents a high-quality academic and technical achievement.

Appendix G: Word Count