Project Proposal

Interactive Visualisation of Spatial Data

Team members: Tyler Gainey, Maddy Prazeus, Tal Rabani, and Feba Salsabila.

Word count: 9048

Table of Contents

1) Introduction

1.1) CS18

Our team is a non-for-profit based set of 4 developers located in Melbourne, Victoria. We were tasked with a project that spans a 3 month period, wherein each team member will dedicate a maximum of 12 hours per week. We recognised the importance of accessible weather information for various industries and the general public, and seek to present this data in a visually engaging and user-friendly format. Our focus is on simplicity and intuitive UI/UX design, ensuring that users can access essential weather data while also utilising temporal tools to gain insights into historical trends.

Our objective stems from the current challenges users face when navigating government websites to access weather trends. Our team conducted a business study on the usability of the Bureau of Meteorology's (BOM) website. This site presents a myriad of options that can confuse non-technical users, particularly when trying to acquire temporal data related to Australia's weather patterns. This situation highlights a significant opportunity to improve accessibility by developing an interactive data visualisation tool. We want to empower individuals, by eliminating the need for specialised technical expertise, to enhance their understanding of weather dynamics in any specified region.

Our project is subject to several critical assumptions and constraints. The budget constraint requires us to operate under a "no-cost" model, necessitating the use of free tools, limiting our data storage capacity. Additionally, we assume that pre-existing weather data will be readily accessible through an API or database, and that updates will continue to keep our tool current. We aim to create a solution with these assumptions and constraints in mind.

Our project aims to develop a mapping tool that visualises temperature and rainfall data from the BOM's site, and enable users to easily explore and analyse spatial data patterns across Australia. Through our business case, we found several options to create this tool, and our group chose to develop a self contained website akin to an interactive archive. This tool will support users in data procurement and filtering, as well as in the creation of intuitive visual representations to aid in understanding trends. In our business case, we developed the goals for our project:

1. **Develop a Web Application:** We will create an interactive spatial map that enables users to explore patterns in rainfall and temperature data in an engaging manner.

2. **Improve Accessibility:** The application will feature intuitive tools for data analysis and visualisation, specifically targeting non-technical users, such as teens and young adults.

3. **In-house Database and Web Scraper:** We will establish a database to store historical weather data from the BOM's site to minimise direct requests to their website, and develop a web scraper that regularly pulls the latest data to ensure that our website provides up-to-date information.

1.2) Report outline

This proposal includes a background section, a project management plan, external designs, proposed methodology, and test planning. The background reviews literature, and identifies opportunities that support our project goals. We detail the project management plan by defining our requirements, organisational model, risk, communication and monitoring management processes, scheduling model, and necessary resources. We developed a Requirement Traceability Matrix (RTM) and user acceptance criteria to define project scope. We chose a process model (PM) with assigned responsibilities. We identify and document risks in our risk register, analyse our stakeholder to develop a communication plan, and establish a maintenance protocol. The project schedule is organised using a Work Breakdown Structure (WBS), and further expanded into a Gantt chart. We also provide estimates for total resources. We include external designs where we represent the website's database and wireframe. Our proposed methodology details the softwares, tools, and algorithms for the website, and integration plans. Finally, a test planning schedule outlines methods for testing the main components of the website.

2) Background

2.1) Aims and synopsis

The BOM's website offers extensive datasets, however, information can only be viewed station by station, limiting users' ability to analyse broader geographic trends. This lack of spatial visualisation and the reliance on users for data processing creates barriers, particularly for non-technical demographics, such as teens and young adults. Our business case identified this usability issue, and identified and analysed 3 options to solve this problem.

The first option was to develop a self-contained website akin to an interactive archive. This website would allow full control over design and functionality, utilise our developer's existing web development skills, and increase accessibility for users via a web browser. However, some features may take significant development time to implement, and maintenance and updates may require ongoing commitment from developers. The second option was to develop a cross-platform application for interactive data visualisation. This offers a native user experience on multiple platforms, leveraging device-specific features like notifications and local data storage, and providing more design flexibility. However, development for multiple platforms increases complexity, requires expertise that our team lacks, and complicates maintenance due to the need to support multiple application versions. The third option was to implement a site affiliated with pre-existing specialised interactive visualisation software. This would allow for quick implementation and access to advanced features and professional support, requiring minimal in-house development, but incurs a high budget, and depends on an external company for updates and maintenance. Ultimately, we decided on the first option; to develop an interactive mapping tool.

We created preliminary project requirements in the development of our tool. Our web application must be hosted at a publicly accessible domain with an easy-to-remember domain name. In our site, we must implement a user-friendly interface that prioritises usability. The interactive component must include a map interface with spatial and temporal filtering options, and displayed rainfall and temperature data. For our design to be accessible, we will follow the Web Content Accessibility Guidelines (WCAG) 2.1 (2023) to accommodate users with disabilities. In our back end, we aim to

establish a database to house historical weather data, and develop an interface with the APIs or web scrapers to fetch the data from the BOM site. Additional research to improve on prior work and handle project management was conducted to implement this project's preliminary requirement.

2.2) Previous work

In recent years, there has been significant advancement in data visualisation tools, especially in the context of making complex datasets more accessible to non-expert users. With improvements in web technologies, the field of data visualisation has evolved, enabling the creation of interactive, web-based visualisations which can be accessed by a broader audience (Hillery, 2020). Below is a synthesis of relevant literature, tools, and techniques that will inform the development of this project.

Geographic Information Systems (GIS) is a technology that is used to gather, manage and analyse spatial and geographic data. GIS allows users to visualise spatial relationships and patterns by presenting data in the form of maps, enabling a more intuitive understanding of trends and relationships within spatial datasets (U.S. Geological Survey, 2004). In the *GIS and Society* chapter (Nyerges, 2009), "Initiative 19" at the National Center for Geographic Information Analysis (NCGIA) establish a criteria for public participation GIS (PPGIS) as various methods of "spatial decision-making tools" available to individuals with a "stake in official decisions". A major aspect of PPGIS is the accessibility of GIS services, where the main barrier is the transition of GIS from professionals to the general public (Aranda et al., 2023). These barriers involve "data quality", and "user-friendliness" rather than technological. Modern GIS software such as ArcGIS and Google Earth, is widely used for research grade spatial analysis, but these platforms are still less suited for casual or exploratory data interaction, due to their difficulty of use for non-expert users, such as the use of jargon. It is recommended clear documentation of all users so that users can complete their tasks efficiently (Haklay, 2010). In a case study testing the usability of Google Maps, MSN Maps & Directions, MapQuest, and Multimap, by 8 cartographers, 8 engineers, and 8 ordinary users, found overall 403 problems, mostly due to "search operations" (Nivala et al., 2008). This highlights the problem with filtering options for mapping tools in addition to better UI components.

In response to this, some GIS technologies have evolved to incorporate tools, such as Mapbox (Mapbox, n.d.), to allow developers to create web-based interactive maps that are more accessible, and easier to customise. These tools enable users to interact with spatial data in an interactive, simplified, and visually appealing manner. They are particularly well-suited for developers aiming to bridge the usability gap between professional GIS users and the general public.

Another significant area of development in recent years has been web-based data visualisation libraries like D3.js. This library has become one of the most influential tools for creating custom, highly interactive visualisation on the web (Fireship, 2021). D3.js provides developers with the ability to bind data to the Document Object Model (DOM), and apply data-driven transformations to create dynamic visualisations. A significant advantage that D3.js offers is the introduction of dynamic properties, where graphical components could be set to a function, allowing for data-driven updates, and interactive transitions in real-time visualisations.

2.3) Opportunities

Despite significant advancements in data visualisation tools and technologies, notable gaps persist in their application, particularly regarding usability and accessibility. These gaps highlight opportunities for improvement and innovation, especially in government data platforms such as the Bureau of Meteorology (BOM) in Australia. Despite the development of tools, such as D3.js and Mapbox, to aid in data visualisation, many government websites remain outdated and challenging to navigate. Raut and Singh (2024) identify persistent usability issues, emphasising that many government websites do not adequately consider the needs of non-technical users. Some of these challenges include, inaccessible to users with disabilities, "unclear navigation paths, lengthy forms, and inconsistent layout". While BOM provides extensive weather data, the site also faces these challenges. The map displaying weather data is outdated and unresponsive, creating difficulties for user interactivity. Moreover, reviewing historical data requires searching through individual stations and datasets, which is inefficient for users who need to access broader trends across different regions.

Our team will adopt user-centred design (UCD) principles and continuous usability testing to create more intuitive interfaces to address these issues. UCD focuses on understanding the needs, preferences, and behaviours of our target audience, ensuring that the final product meets their expectations (Interaction Design Foundation - IxDF, 2024; Norman & Draper, 1986). This approach is an "iterative design" that allows for improvements based on user feedback to increase satisfaction and engagement with the application. Dalangin (2022) emphasises the importance of involving users throughout the design process, from initial concept development to usability testing. In a case study by Corry et al. (1997) to improve Indiana University's website, there were 2 phases of usability testing in paper and computer prototypes to develop the main components before going online after a year. The site was reported to be in the top 50 most frequently visited websites accessed by home computers in 1996, and was used for orientation of new students. Since our team has been unable to conduct our own survey on our demographic, we have developed our scope through preliminary research, and will gather insights from real users when prototyping. As such, we made our requirements following a persona based on our demographic. In gathering user experience during the development, we can evaluate our design and identify areas of improvement to make our visualisations more accessible and user-friendly. We will incorporate UCD principles throughout the development process, so that our tool satisfies a heuristic evaluation of accessibility to complex datasets. This way we can address the usability challenges currently faced by platforms like BOM.

Windy (Windyty, SE, n.d.) is an excellent example of an interactive weather map that showcases the potential of data visualisation tools through combining a user-friendly interface with real-time meteorological data, allowing users to explore wind patterns, temperature, precipitation, and other various weather phenomena globally. Our website will contain a similar style interactive map with addition to historical data. This functionality will enable users to analyse changes in climate patterns, such as shifts in rainfall or temperature, which can be valuable for educators, students, and the general public interested in historical weather data. By providing an intuitive platform for accessing and visualising this information, we aim to improve user engagement, and enhance understanding of long-term weather trends, ultimately addressing the usability challenges present in existing government platforms such as the BOM's site.

By Kerzner's (2017) criteria of a project, we have defined our specific goal constrained by the 3 month period in our Scope and Requirements section, where our resources, and performance metrics are also listed. Our project life cycle follows a process model defined in our Organisation section, and our stakeholders are identified in the Communication section. In developing our platform, this project

will undergo several planning stages. Project Management Institute's (PMI) *PMBOK Guide* (2021) book defines a planning process group, wherein our team developed a min map that outlined our scope, timeframe, resources, communication methods, risks, quality assurance, and finance. Each of these components are elaborated in the project management plan. It is important to define these processes because any updates can incur significant impacts on parts of the project. Moreover, the initial effort of the project's involvement of all stakeholders in the planning stage are defined in this proposal.

Ethical concerns for the rapport of our team can be standardised by following PMI's *Code of Ethics and Professional Conduct* (Ethics Standards Development Committee (ESDC), 2006). This code of conduct highlights responsibility, respect, fairness, and honesty. Each of our developers must have ownership of their tasks, and are accountable for their contributions. Where tasks have been completed, acknowledging input regardless of experience, and having an open communication channel is imperative for creating an inclusive atmosphere. In assigning responsibilities, the distribution of workload must be agreed upon and equal, and conflicts are resolved through addressing disputes in open discussion. In our development process, it is imperative to have transparency and integrity with progress updates to allow for timely milestone advancements.

3) Project management plan

3.1) Overview

As stated in our introduction, the primary objectives of our project are as follows: Create a website with intuitive UI, create a database to house historical weather data from the BOM, create functionality for users to search and view this data via custom graphical visualisations, finally save and share the aforementioned data and visualisations.

Our developers will focus on building core HTML and CSS components to use throughout the website, implementing express js server framework, designing a relational database structure, building a web scraper for the BOM's site, setting up a secure login system, and developing custom visualisation generation tools. Our major milestones will be the initial setup of the website and back end structure, completion of back end and database, and integration of all back end methods into front end UI.

3.2) Scope and requirements

We developed a RTM (table A.2) that describes the user's requirements, categorised as a Functional (FR) or Non-Functional (NFR) requirement, and acceptance criteria.

Requirement 1 is a FR that asks for a main web page that enables users to access primary interactive widgets. This requires the inclusion of a structured layout that adheres to best practices for effective interface design, such as maintaining consistent functionality across repeated components, ensuring accessible sizing for readability, minimising duplicate actions for the same functionality, and organising similar elements and components together.

Requirement 2 is a FR that asks the website to give users the ability to visualise historical weather data on a map of Australia to gain an intuitive understanding of data trends. This should include the use of heat maps to overlay the historical data onto geospatial locations, with appropriate pop ups for data lacking specific locality, and a sliding scale for users to view temporal changes.

Requirement 3 is a FR that asks for the user's ability to navigate the web page smoothly. Our website should ensure that loading data and interacting with the map is intuitive and seamless through straightforward navigation through a well-structured layout and supporting visual elements.

Requirement 4 is a FR that asks for the user's ability to download data as both a graphical image and a CSV file for manipulation and display. This means data from a selected region can be downloaded in CSV, PNG, and JPG formats, where each field should be clearly defined with labelled graph axes to indicate the content of the CSV. Additionally, users who sign up for an account should be able to save and load their created graphs.

Requirement 5 is a FR that asks for the user's ability to intuitively interact with the map to view trends in temperature and rainfall through visualisations. Users should be provided options to switch between heatmaps, scatter plots, line charts, and histograms for flexible data interpretation. Interactive map layers should allow toggling between temperature and rainfall datasets. Tooltips and pop ups must display detailed information when users hover over specific map points or regions.

Requirement 6 is a NFR that asks for the user's ability to pan and zoom on the map. Users should be able to pan the map by clicking, holding, and dragging, and zooming in can be achieved using the mouse scroll wheel.

Requirement 7 is a FR and NFR that asks for all map actions to be smooth so eye strain is minimised for users. This means that map panning is smooth, especially on low-power devices, and that dark mode, and other visually aiding colour schemes, are compatible with the design.

Requirement 8 is a NFR that asks for all visualisations to be easy to interpret. Visualisations must be colour-blind friendly, well-labelled with axes and titles, use appropriate idioms for their respective data (e.g., nominal data should not be placed on an interval x-axis), and avoid clutter by not displaying more than six unique categories of data.

Requirement 9 is a NFR that asks for a website load time of under 3 seconds (assuming normal internet speed > 10Mb/s).

Requirement 10 is a FR that asks for the user's ability to filter and display specific types of data for more targeted analysis. The website should include a search bar that enables users to quickly locate and zoom into specific locations by name or coordinates, as well as an advanced search option that allows filtering areas based on certain conditions (e.g., only displaying areas with rainfall or temperature above a specified threshold).

3.3) Organisation

3.3.1) Framework

Our team will follow the Agile framework (Beedle et al., 2001) due to its flexibility and iterative nature that allows swift adaptability to schedule changes. This is due to the limited time frame and evolving nature of the project may require frequent reassessments and adjustments, thus choosing Agile is ideal for accommodating evolving requirements, and ensuring timely delivery of key features. Additionally, Agile supports regular feedback from stakeholders, ensuring that the final product aligns with the users' needs and expectations. This also helps the team efficiently manage potential risks by breaking down the project into manageable sprints, promoting both communication and quality delivery throughout the development process. Furthermore, our team has held weekly meetings, and successfully integrated some Agile principles in the completion of previous assignments such as regular collaboration and communication, and iterative development. This was done by splitting up tasks into smaller, manageable increments, allowing members to focus on individual components step by step, then iterating on those components based on feedback, thus evolving project requirements.

3.3.2) Process model

We are choosing to implement the Scrum software development model as our process model (PM) methodology. Scrum's use of sprints (Rehkopf, n.d.-b) allows us to manage the project in cycles, where each sprint includes planning, development, review, and retrospective stages. This cycle fosters regular team synchronisation, adjustment to feedback, and allows risk management through regular reassessment. Each sprint's progress will be monitored in weekly meetings, ensuring any roadblocks are addressed promptly and the team remains aligned with project goals. The scrum team consists of clearly defined roles to ensure each project responsibility is effectively managed (West, n.d.).

Product Owner: The voice of the stakeholders that prioritises features defining user stories based on needs and project requirements. They are responsible to set the vision of the project and ensure that development stays aligned with stakeholder expectations and overall project goals.

Scrum Master: Facilitator for the Agile process, ensuring that team members adhere to Agile principles. They are responsible to keep the team focussed and running on schedule. The Scrum Master also monitors sprint progress, organises regular Scrum ceremonies and handles obstacles to project development (Rehkopf, n.d.-a).

Development Team Members: Developers responsible for implementing project features according to user stories and acceptance criteria. They handle the technical aspects of the project, including coding, designing interfaces, collecting data, testing, and integration, and collaborate closely with the Product Owner to ensure alignment with the project's vision.

3.4) Management process

3.4.1) Risk Identification and Analysis

When identifying risks, our team organised brainstorming sessions to generate a list of potential risks. During our brainstorming sessions, we will be carefully analysing a variety of segments of our project to identify where risks may arise. This may include analysing, testing and researching software we intend to use. For example, our developers may conduct research on Node.JS packages to familiarise themselves with known vulnerabilities that may incur a security risk, or limitations in the server's free tier. However, we will also attempt to address social factors as they can introduce risks to our project. For example, we may want to ask if developers feel overworked or demotivated to contribute to the project, due to its ability to incur delays on the project timeline.

We included different categorical risks that included technical, project management, and organisational. After identification, our team made a qualitative assessment to evaluate the likelihood of each risk occurring and its potential impact on the project, categorising these as high, medium, or low (table 3.4.1). This prioritisation will guide the team in focusing on the most critical risks. For significant risks that may pose a considerable threat to project timelines, quantitative analysis can be employed, using statistical methods or simulations to gauge potential scheduling impacts.

3.4.2) Risk Response Planning

After analysing the risks, we developed a risk response plan to effectively resolve complications in the event of any risk occurring. This involves using the TARA(E) strategy for each identified risk (Adcyber, 2023). The team can choose to avoid certain risks by modifying project plans, mitigate risks by implementing additional testing protocols or fallback options, or accept minor risks by acknowledging their potential impact while preparing contingency plans. Each member was expected to contribute to developing these strategies, to ensure practical response plans were put in place.

3.4.3) Risk Monitoring and Control

Ongoing risk monitoring and control will be scheduled by regular check-in meetings to review the status of identified risks, and discuss any new risks that may have emerged during project execution. We will maintain a risk register (table A.1) to document each identified risk, its analysis, response strategies, and current status, with regular updates reflecting any changes. Establishing key performance indicators (KPIs) will help the team track the effectiveness of risk management strategies, and make adjustments as needed. Furthermore, our communication section will elaborate monitoring methods with all stakeholders regarding significant risks and the measures taken to address them.

Low, Medium, High, Extreme

Probability upper limit of risk (<%)	**Impact score**				
	0 - 1	**2 - 3**	**4 - 6**	**7 - 9**	**10**
90			R1 (540)		
70		R3 (210)		R2 (420)	
40			R5 (180), R6 (160), R8 (120)	R4 (200)	
20				R7 (140)	R10 (70)
5			R15 (30), R16 (24)	R9 (90), R12 (45), R13 (40), R14 (40)	R11 (50)

Table 3.4.1: Qualitative assessment taken of the evaluated risks (see Table A.1), where the ID is associated with a score in brackets.

3.4.1) Communication

Our communication plan is designed to keep all stakeholders informed of our current progress through frequent meetings and detailed reports. A communication table has been outlined below to indicate the frequency of meetings, their objectives as well as who will be present.

Stakeholder	Objective	Channel	Schedule
All stakeholders	- Introduce project - Outline expectations - Present deliverables - Define requirements	Proposal document Design analysis	Once
Team	- Report progress alignment with milestones - Review issues to solve - Implement supervisor feedback - Conduct internal audits	Discord In person	Weekly (before supervisor meeting)
Supervisor	- Review project status - Review project alignment with requirements - Monitor performance - Gather feedback	Zoom Studio session	Weekly (allocated day tbc.)

Table 3.4.2: Communication table outlining stakeholder expectations.

3.4.2) Weekly developer meetings

Developers will meet weekly using Discord, after our supervisor meeting. The purpose of this meeting will be to review our progress on the outcomes against our milestones outlined in our Gantt chart. Developers will perform an internal audit, reviewing code committed to Github, assessing task completion as well as their adherence to requirements. Task completion will be assessed through tracking of our Gantt chart. Developers can use this meeting to discuss any rollbacks to be performed through GIT Version Control, if any errors have been identified through an internal audit. It is also very important that developers refer to the risk register during these meetings, to identify if there are unresolved risks. If risks have been resolved since the previous developer meeting, developers can use this time to mark the risks as resolved. The weekly frequency of these meetings will also ensure developers do not remain isolated during the development of the project.

3.4.3) Weekly supervisor meetings

Every week, our team will meet with our project supervisor during our studio session or through Zoom. The project supervisor has a comprehensive understanding of the requirements for our project, making them a valuable source of feedback. During this meeting, we will have the opportunity to have our current progress reviewed by our supervisor through our weekly status reports, thus ensuring any adherence to the requirements are rectified early, minimising delays.

3.4.4) Weekly status reports

After each developer meeting, our developers will create a status report. Using data from our Github commits, internal audits, and discussion among developers, the project leader can create a generalised overview of what has been completed, and how carefully we are adhering to the project timeline. Our

risk register outlines any potential risks or delays. This ensures all stakeholders are informed of our current progress with the project, and are able to collectively make decisions moving forward.

3.4.5) Monitoring methods

Our project relies heavily on frequent monitoring to ensure our team is completing our project in adherence to our schedule, as well as the requirements. We will be employing multiple methods to perform this effectively, including Gannt chart tracking, GIT Version Control, and internal audits.

3.4.6) Gantt chart tracking

It is important that our project outcomes are being completed in accordance with our project schedule. In following our predefined schedule, there may be delays, and concerns in regards to our competency from stakeholders. To ensure this does not occur, we will monitor progression toward our project milestones with our Gantt chart. Our project leader will evaluate, on a weekly basis, if our developers have completed work in alignment with our predefined Gantt chart schedule.

3.4.7) GIT Version Control

To allow for version control of our project, we will also be using Github. This ensures any changes can be tracked. Tasks will be developed in their own branches, keeping them isolated, and deterring potential conflicts. Furthermore, Github allows for rollbacks. This means that if any errors or conflicts occur in our code, we can safely return to a stable version of our code, thus maintaining project integrity.

Measuring progress of our project may be subjective and difficult to determine. Therefore, through use of Github, tasks can be committed to the cloud where developers can easily identify which outcomes and subtasks have been completed, thus providing an accurate understanding of our progress towards a milestone. Furthermore, as commits to our Github repository include descriptions, developers can clearly state what has been completed in each commit, which can be further compared to our Gantt chart milestones. Commit descriptions will also provide a means of documentation for our project that is accessible to all our developers. Prior to beginning development, training will need to be conducted on how to perform internal audits.

3.4.8) Internal audits and requirements adherence

Internal audits must be conducted where developers will analyse the codebase in our Github repository, as well as commit descriptions, to ensure quality of the outcomes that are produced, identification of errors, and adherence to requirements. Without performing internal audits, our developers may not identify misalignments with the requirements, which may result in a more extensive rework of a future version in the project.

3.4.9) Training

Developers must understand the purpose of internal audits, which is to ensure completed outcomes align with the predefined requirements for the project. Developers will be required to familiarise themselves with the requirements, and thoroughly assess both the commit descriptions, and the codebase, to determine their compliance to the requirements. This will further ensure that any inaccuracies between the completed outcomes and the requirements are addressed early, since modifying the completed outcomes later in our project timeline may be difficult.

3.5) Schedule

Our project agreement will be achieved via 3 overarching features. Namely, the Website, Backend and Database. These 3 objectives and their sub tasks are defined in our WBS (see figure A.1 in the appendix), which is expanded in our Gantt chart (see figure 3.5). We have then organised these subtasks into weekly sprint allocations, in such a format that minimises the total project length. This is primarily achieved by minimising dependencies or blocking features, and maximising opportunities for parallel development (our Proposed Methodology will elaborate on this). There are also 3 key milestones for this project, namely at the end of sprint 4, sprint 9 and sprint 12.

3.5.1) Sprint 4 Milestone

By this milestone our front end development will consist of a main HTML page, some basic CSS styling, and an empty Mapbox widget. In the back end we will have created js files to house the server functionalities and the Express.js endpoints. There will be minimal user functionality available for demonstration at this stage, but it is an important point for intra-team analysis, specifically in identifying any issues in our work style, communication, or organisation procedures.

3.5.2) Sprint 9 Milestone

After sprint 9's completion we will have completely finished the back end and the database. We will conduct an immersive technical demonstration with the client. This will be an important opportunity to get feedback on the state of key features such as data fetching, graphical or visualisation generation, and the overall experience from an unfamiliar user point of view. If any functionalities are not up to our clients standards and expectations, we will need to find alternative solutions to be implemented in the final milestone. At this point in time, we should keep in mind the fatigue and workload of the development team. Thus, we will require time to assess if any feelings of burnout have begun to creep into the group, and take appropriate countermeasures.

3.5.3) Sprint 12 Milestone

We will now have completed the front end implementation of all back end functionalities. We expect our product to be fully functional with a high degree of polish. We will conduct a final technical demonstration with the client, and complete our development, regardless of the result in the sign off of our project that is dependent on the client's satisfaction. As such this marks a key point to develop handover documentation for future development teams who may take over this project. This handover documentation will focus on financial maintenance (eg. any ongoing bills for hosting, or domain name registration), code maintenance (eg. regular updates of packages, and check ins of webscraper functionality), and finally ongoing features yet to be implemented that can be recorded in the features section of our Github repository.

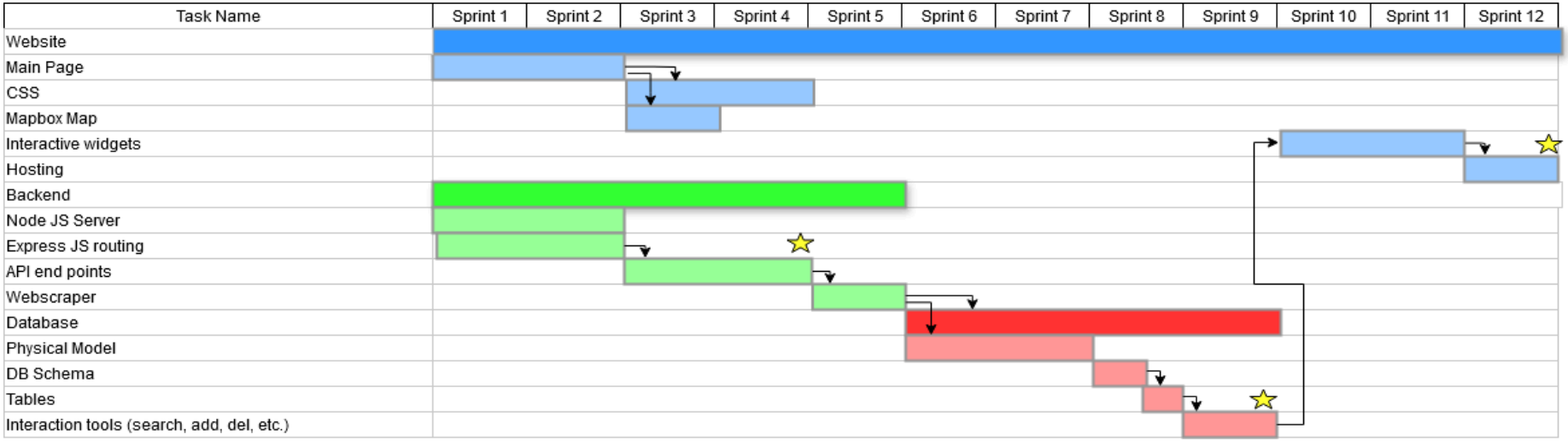

Figure 3.5: Gantt chart representation of the project schedule.

3.6) Resource

Table 3.6 below provides an estimate of all resources.

Resource	**Type**	**Function**	**Description**
Azure	Server	Cloud storage	Our initial analysis found 20,000 stations from the BOM whose lifelong record files are about 100 KB each, giving us a minimum amount of 2GB required storage.
Bcrypt	Algorithm	Hash algorithm	Stores encrypted user information in JSON format on our server.
Bootstrap	Library	Cross platform front end integration	Provides responsive, modern design features for visually appealing interfaces across devices.
CSS	Language	Front end	Stylesheet language for visual presentation and layout of HTML documents.
D3.js	Library	Data visualisation	Creates complex and interactive data visualisations, such as graphs from the data shown on the map.
Docker	Container	OS execution platform	Containerises the application to standardise the development environment across various systems using Linux-based containers.
duck dns	Domain sourcer	Creating our DNS	Sources our domain name, and allows us to set up SSL and other core features.
Express.js	Language	Middleware	Builds the APIs, and provides an environment for managing and delivering data requests to the front end.

GitHub	Shared repository	Version control	Incorporates software version control to track changes made through commits and branches. Handles automatic testing after every commit through a collection of unit tests through built-in CI/CD pipelines.
Handlebars.js	Library	Front end templating	Templating engine for rendering dynamic content on the front end.
HTML	Language	Front end	Standard front end language used to structure content on the web.
Javascript	Language	Front end	Enables interactive and complex functionalities on web pages to create responsive user experiences through real-time updates and event handling.
Mapbox	API	Map functionality	Modification and customisation of our map, and offers a highly customizable interface to implement dynamic layers for rainfall and temperature data, and interactive zoom controls.
Mocha & Chai	Testing framework	Javascript maintenance	Integrated into CI/CD pipeline, and provides a framework supporting synchronous and asynchronous testing for API calls and database interactions.
Node.js	Language	Back end logic	The back-end to handle API requests.
Personnel	Developer	Makes the website	4 members available from the team project.
pg-promise	Library	Data query library	Simplifies interactions with PostgreSQL.
PostgreSQL	Language	Database	Reliable and supports complex queries for handling large datasets. Includes PostGIS extension to manage spatial data, and enable advanced geographical queries.
Postman	API	API maintenance	Testing and debugging API endpoints during development. Provides features for sending HTTP

	platform		requests, automating tests, managing environments, and collaborating on API development.
Puppeteer	Library	Data processing	Interacts with js and html objects without the need of wrapper functions.

Table 3.6: All resources used in the application of the website.

4) External design

From our preliminary designs, our team created two representations of our website application.

4.1) Conceptual Database

For this project, we are implementing a relational database to effectively manage the data components required for our interactive spatial map application. The architecture divides the data into two distinct databases: one for spatial weather data and another for user interaction data. This separation enhances performance by isolating frequently accessed weather data (figure 4.1.1), such as rainfall and temperature records, from user-specific information (figure 4.1.2), such as saved maps, and preferences. By partitioning these datasets, we minimise data dependencies, optimise query efficiency, and improve scalability to accommodate a growing user base and increasing data volume. The design ensures that all tables maintain necessary relationships to preserve database integrity.

4.1.1) Weather Information Database

STATION catalogues each weather station across Australia. This foundational table links temperature and rainfall data to geographic points. Temperature, and rainfall data is organised into 6 tables: TEMPERATURE_DATA_DAILY, TEMPERATURE_DATA_MONTHLY, TEMPERATURE_DATA_YEARLY, RAINFALL_DATA_DAILY, RAINFALL_DATA_MONTHLY, and RAINFALL_DATA_YEARLY. Each table captures data records at varying time granularities linked to the specific weather station through *station_id*. Monthly and yearly data is stored directly in the database for efficient retrieval without recalculating from daily records, thereby reducing server load during user queries.

4.1.2) User Interaction Database

USER maintains user accounts for the application, including usernames and encrypted passwords. Each user is assigned a unique *user_id*, which is referenced by other tables. The USER_TOWN_BOOKMARK table enables users to bookmark specific towns, linked through *user_id,* and to the town via its name and state. SAVED_MAP stores user-specific map configurations, including the centre coordinates, zoom levels, and applied filters, allowing users to return to previously saved maps easily. Each saved map is also associated with the user who created it, along with a timestamp. When developing the website, we will likely choose to enter the applied filters as a JSON object into the database instead of individual fields in the SAVED_MAP table. This is because filter data should not need to be searched for in the database and therefore, it would increase efficiency by having it as one field in the table. Lastly, the GRAPH table allows users (linked through *user_id*) to save and revisit graphs they have generated.

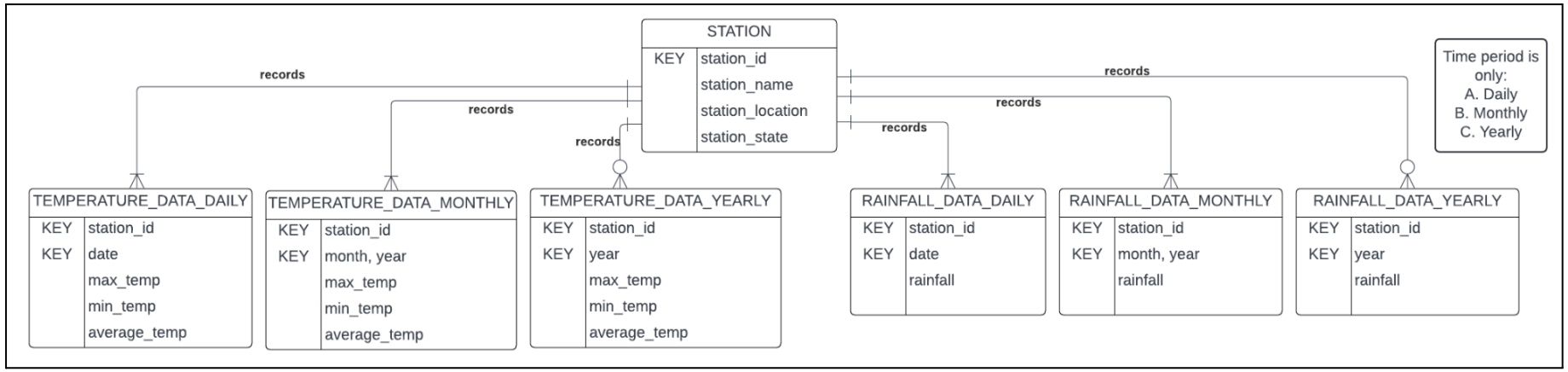

Figure 4.1.1: Planned database model of the website's weather information.

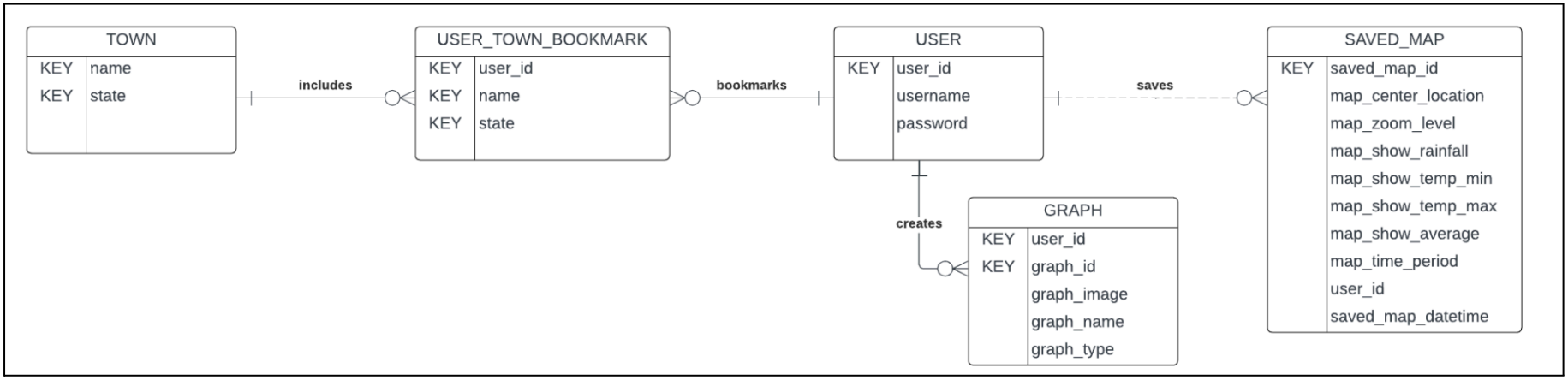

Figure 4.1.2: Planned database model of the website's user end.

4.2) Wireframe

Our website follows the laws of UX (Yablonski, 2018) and principles of visual design (Gordon, 2024) to effectively enhance user engagement through intuitive approaches. This wireframe (figure 4.3) is structured to align with the visual elements outlined in the RTM (see table A.2).

4.2.1) Home page

The home page (frame 1 of figure 4.2) features two primary components: an interactive map, and a tutorial section. The design incorporates filtering options that allow users to select temporal and spatial elements to improve user experience. This addresses requirement 10 of the RTM. This organised structure reduces cognitive load by categorising data intuitively, enabling users to visualise and analyse information without feeling overwhelmed, which is noted in requirement 8. A clear visual hierarchy is established for ease of navigation through the filtering system. This is done by varying font sizes and weights for titles, subtitles, and headers. Large, clickable map features and zoom buttons are designed to comply with Fitts' law, allowing for quick user interaction.

4.2.2) Line chart visual type

After users input their filters, the home page transforms to display their selected graphical output (frame 2 of figure 4.2). The tutorial section becomes less prominent, allowing users to focus on their results. This layout promotes a sense of connection among elements, adhering to the law of common regions. New "download" and "share" buttons enhance user engagement to satisfy requirement 4. The exclusion of these options prior to map creation is a constraint to prevent users from downloading or sharing null outputs. This approach follows Jakob’s law, and Hick’s law since they use common icons across applications, and ensures users avoid non-functional interactables.

4.2.3) Heat map visual type

Per requirement 5, users can also select regions on the map or switch to a heat map visualisation (frame 3 of figure 4.2). The graph includes temperature levels represented in degrees Celsius, and, per requirement 2, a time slider that enables users to observe changes over time. The time slider interaction sequence is shown in frame 4 of figure 4.2. The slider’s contrast improves usability by guiding users toward engaging with the features.

4.2.4) User map loading and saving

Additionally, the website offers "save map" and "load map" options (frame 5 of figure 4.2). An overlay prompts the users when they attempt to save or load a map without logging in. Consistency in visual design is maintained through a monochromatic colour scheme, and buttons indicated through rounded elements, reinforcing the intuitive nature of the site as outlined in requirements 1 and 3.

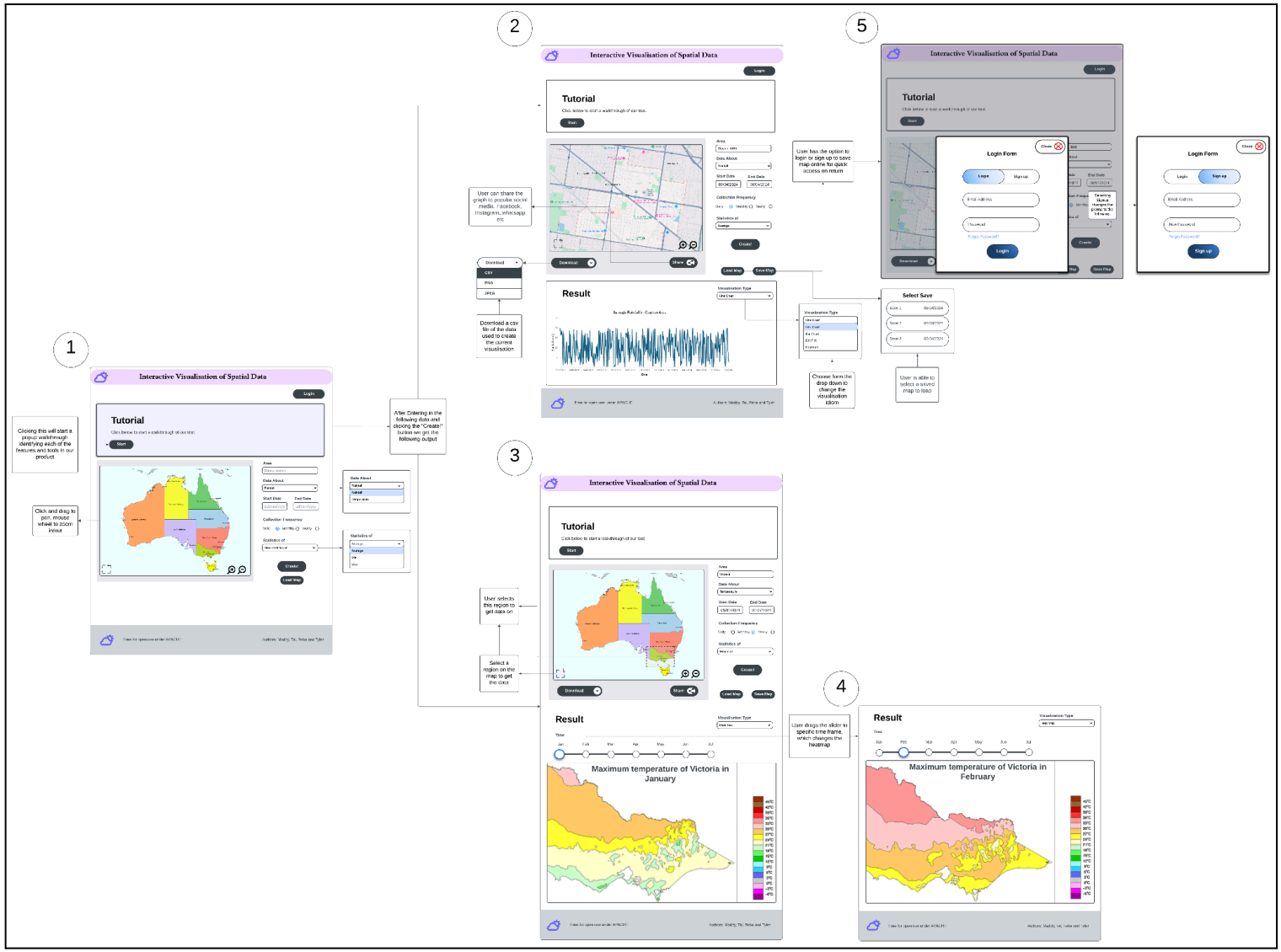

Figure 4.2: Labelled wireframe representation of the website's main features with corresponding frame IDs.

5) Proposed methodology

We will implement a full stack development approach, encompassing both front-end and back-end development to build a seamless and interactive web application with an emphasis on usability, performance, and scalability. This section will describe and justify our chosen tools and the methodologies.

5.1) Toolset

5.1.1) Front-End Components

Our main front-end languages will be the standard HTML, CSS, and Bootstrap. Bootstrap is a responsive front-end framework that will enable cross-platform integration by adjusting the screen size to the device. Handlebars.js will be used as our templating engine for rendering HTML content dynamically based on user inputs and data retrieved from the back-end. The decision to use Handlebars is motivated by its simplicity and the team's familiarity with it.

JavaScript will be used for dynamic content updates, and D3.js will provide graphical data visualisations. D3.js will allow us to create dynamic graphs that represent trends and changes in weather data visually. Map-based visualisations will be powered by Mapbox, enabling us to create interactive maps with zoom and pan functionalities and integrate rainfall and temperature layers.

5.1.2) Back-End Components

The back-end will be developed using Node.js along with Express.js as the middleware for handling API requests. Node.js was chosen for its widespread use, performance and scalability. Express.js will provide a robust environment for managing incoming requests, and delivering data to the front-end. PostgreSQL was chosen for our database system specifically due to the team’s experience with SQL. We will use the PostgreSQL query library, pg-promise, to simplify the integration with our database for efficient interactions with the data. This application will be containerised using Docker to ensure a consistent development environment across the team and make deployment to cloud services straightforward.

5.1.3) Data Collection and Preprocessing

There are 17936 weather stations located across Australia that record rainfall and temperature daily. The BOM website will be scraped using an automated script developed in Puppeteer to collect all the rainfall and temperature data for each station.

Puppeteer simulates user interaction with websites by automating web page navigation, form submission, and data retrieval. It will navigate to each weather station's page, and access 4 data sections: daily rainfall, monthly rainfall, daily temperature, and monthly temperature. Puppeteer will automate the download of the CSV files for each data section, where these files will undergo a comprehensive processing phase before being stored in our database.

Each weather station will have 4 CSV files downloaded, 2 for daily rainfall and temperature data, and 2 for monthly rainfall and temperature data. We will illustrate an example of our data processing using raw data from a specific weather station.

1	Product code	Bureau of	Year	Month	Day	Rainfall amount (millimetres)	Period over which rainfall was measured (days)	Quality
2540	IDCJAC0009	86018	1893	12	13	0		Y
2541	IDCJAC0009	86018	1893	12	14	0		Y
2542	IDCJAC0009	86018	1893	12	15	3.8	1	Y
2543	IDCJAC0009	86018	1893	12	16	6.6	1	Y
2544	IDCJAC0009	86018	1893	12	17	0		Y
2545	IDCJAC0009	86018	1893	12	18	1.8	1	Y
2546	IDCJAC0009	86018	1893	12	19	7.6	1	Y
2547	IDCJAC0009	86018	1893	12	20	0		Y
2548	IDCJAC0009	86018	1893	12	21	0		Y
2549	IDCJAC0009	86018	1893	12	22	0		Y
2550	IDCJAC0009	86018	1893	12	23	0		Y
2551	IDCJAC0009	86018	1893	12	24	0		Y
2552	IDCJAC0009	86018	1893	12	25	5.8	1	Y
2553	IDCJAC0009	86018	1893	12	26	16.5	1	Y
2554	IDCJAC0009	86018	1893	12	27	0		Y
2555	IDCJAC0009	86018	1893	12	28	10.4	1	Y
2556	IDCJAC0009	86018	1893	12	29	15	1	Y
2557	IDCJAC0009	86018	1893	12	30	2.3	1	Y

Figure 5.1: Raw daily rainfall data downloaded from Caulfield (Racecourse) weather station, station number 086018 (BOM, 2024).

From this raw data, the following preprocessing steps will be applied:

1. **Data Cleaning:** The raw data may contain missing or erroneous entries (e.g., incomplete records, outliers). We will employ filtering techniques on missing data points, and handle any inconsistencies to ensure the dataset is reliable.

2. **Data Formatting:** As the data comes in a CSV format we will standardise the data into a unified format suitable for PostgreSQL injection. This step will involve organising the data into well-defined schemas to simplify queries and management. Here yearly data tables will also be pre computed using the monthly data, and stored in the database.

3. **Geospatial Conversion:** We will store the location of the weather station in a spatial format using PostGIS for efficient querying of geospatial information. This will be done by converting the latitude and longitude into spatial data types such as POINT.

5.2) Algorithms

Our project leverages 2 main algorithms to process, display, and interact with large sets of spatial data.

5.2.1) Hashing Algorithm

Bcrypt will be used to hash user passwords securely before storing them in the database. It is based on the Blowfish cipher (Schneier, 1994; Provos et al., 1999). This algorithm comprises of these characters

"./ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789$.",

which results in a hash that is 60 characters long in the format

"$[algorithm]$[cost]$[salt][hash]".

This format contains 2 chars hash algorithm identifier prefix, "$2a$" or "$2b$" indicating BCrypt, cost-factor, n, which represents the exponent to determine how many iterations 2^n, 16-byte, salt, and 24-byte, hash, that is base64 encoded to 22 and 31 characters, respectively (Salazar Cardozo et al., n.d.).

5.2.2) User Management Algorithms

Users must login to the site to be able to access their saved maps and graphs. Therefore, an algorithm must be created to allow users to enter their username and password as input, and have this information verified by the server. As each password is encrypted using Bcrypt, the algorithm must first de-hash the user's password saved on the server, then compare the de-hashed password to the password input by the user. Pseudocode for this algorithm located in Node.JS, with inputs retrieved via HTTP, is presented below.

```
function verifyUserDetails(username, password) {
        // retrieve stored user via username
        user = findUserByUsername(username)

        // compare input password with stored hashed password
        passVerified = bcrypt.compare(password, user,hashedPassword)

        // password matches, login
        if passVerified:
                login
        else:
                reject
}
```

5.2.3) Spatial Algorithms

Data displayed based on user interactions with the map is retrieved from weather stations within particular geographic areas. This will involve spatial indexing and range queries using PostGIS. We will use a bounding box query to do this to return all weather stations within a selected rectangular region defined by the top left and bottom right corners.

A user may want to click on a spot on our map, to determine the weather data at that specific location. One of our spatial algorithms will determine the longitude and latitude values of this click, and use PostGIS's geospatial capabilities to query the station with longitude and latitude values minimum to the click. It is then possible for the weather data from that station to be sent back to the requesting user. Pseudocode for this algorithm located in Node.JS, with inputs retrieved via HTTP, is presented below.

```
function getClickWeatherData(clickLong, clickLat) {
        // find the nearest weather station to the click location
        nearestStation = findNearestStation(clickLong, clickLat)

        // retrieve weather data from the nearest station
        clickWeatherData = getWeatherDataFromStation(nearestStation)

        // return the weather data to user
        return clickWeatherData
}
```

5.3) Software architecture and application

Integration between the front-end and back-end will be achieved through RESTful APIs developed from Express.js. We can illustrate this application through a model, view, and controller architecture, and 2-Tier system application. In the client's view, user interactions in the front-end will be processed with Javascript logic to the controller, where API requests are sent to the back-end. When users create graphs, the back end will retrieve rainfall and temperature data for specific regions displayed on the interactive map. When users login or signup, the back end will process user authentication. The data will be queried from the model via PostgreSQL database and pg promise library, processed in Node.js, where user logins are encrypted via bcrypt, and returned to the client's view. Mapbox will be used for user-end mapping interactables, D3.js will be used to create the graphs, and Handlebars will render dynamic content on the website.

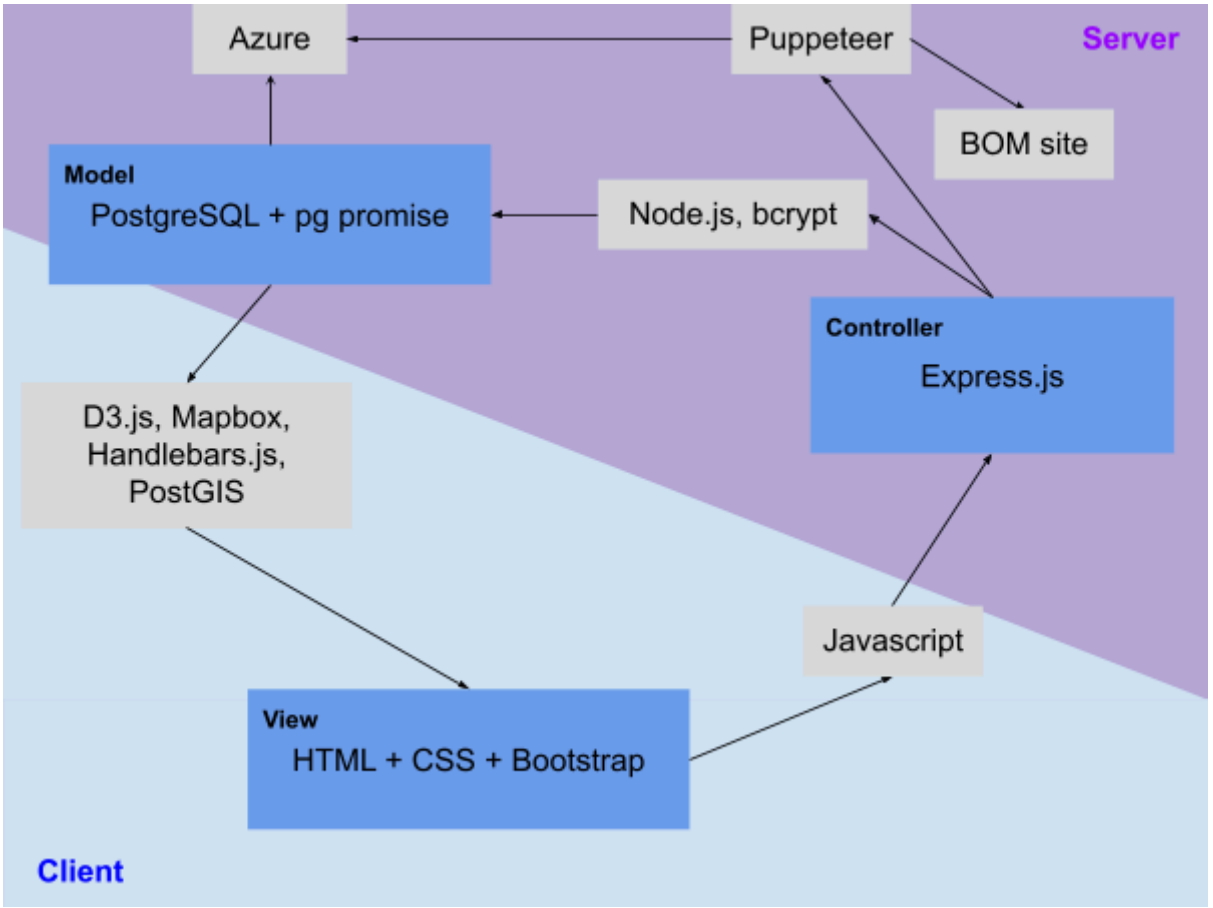

Figure 5.3: Software architecture and application of the tool set with MVC architecture, and 2-Tier system.

5.4) OS Execution

Throughout our development process, we will be using Docker. Dockers allows us to work inside containers, allowing developers to contribute to the project regardless of their host operating system. Our designated server host Azure allows for deployment of Docker containers. This means the container will remain exactly the same from the development process to deployment, minimising any server compatibility or configuration issues. Our application will make use of the following two Docker containers, whose environments have already been extensively vetted and determined to be stable for their specific functions.

5.4.1) Front-end container

The front-end container will host the application server, built with Node.js and Express. Here it will serve its primary duties of querying weather data from the PostgreSQL database and serving the data over HTTPS. Furthermore, user data will be stored in this container in JSON format.

5.4.2) Back-end container

The back-end container will host the PostgreSQL database, where BOM data will be stored. Our Node.JS application will have the ability to query data from this container when a request is made. As this PostgreSQL container environment has already been extensively used by other developers, it ensures that our database is secure and free of any potential vulnerabilities. Furthermore, Docker allows for efficient scalability of this container, providing confidence that increases and changes in complexity of data will not cause concern. FigureB in the appendix illustrates how our Docker containers will exist on the server.

5.5) Intellectual property

For this project we will licence our code as open source. More specifically, the GNU General Public License (GPL) v3 licence. While GPL is incompatible with proprietary software all the tools and software that we have mentioned are also open source. We also do not expect to need to work with open source software in the future. Furthermore, by making our code open source we make it possible for other savvy developers who have feature requests or suggestions to fork and improve our project. This is particularly important when we consider the intended life cycle of this project. Since we intend to stop development after semester 1 of 2025, if our project was open source, it would be henceforth dead in the water. No reality for it to be further developed by our team, and any bugs that inevitably arise as dependencies change could never be fixed. As such, by choosing to make our project open source we give it the opportunity to continue to live and grow beyond our initial group.

6) Test planning

6.1) Testable Components

Our testing suite will be focused on 3 main areas. Namely, front end UI, backend API endpoints and our BOM web scraper. With regards to the front end, we will be testing website sign in, and UI navigation and redirection through <a> tags. However, this will not be initialised immediately as front end testing is complex and unnecessary during early development stages. Back end testing will focus on functionality and veracity of our API endpoints. This will be straightforward to implement, and immediately beneficial as our back end is the area most likely to have bugs or crashes while we expand our project. Finally, web scrapers are prone to breaking or retaining errors because constant updates to the website (ie. HTML changes) may result in the scraper trying to reference elements that no longer exist. As such it is important to have regular tests for our web scraper so that changes to the BOM's site that affects our website should be identified at the earliest possible stage.

6.2) Testing Methodology

The back end API will be covered through unit tests by performing valid and invalid GET requests to the respective API endpoint, and comparing the result to an "expected output". The webscraper can be run on a known link to check that it actually runs without an error, and scrapes the expected data. Finally, the front end can be tested using Puppeteer's expansive interaction tool kit, which allows us to mimic user interaction paths.

Our testing will be integrated with Github's CI/CD pipelines and version control to run on commit and pull requests. Our CI/CD pipeline will run tests automatically using Mocha and Chai after each commit to maintain code quality and detect issues early in the development process. This ensures that we get regular feedback of the current project state after every modification. More granular testing and feedback can be obtained during tech demos with our client at the end of each milestone. Additionally, Postman will be used to test and debug our API endpoints during development, ensuring that front end and back end components work in harmony.

All unit tests that are run will be documented in a unit tests document. This will serve as an entrypoint to get a big picture understanding of the testing suite, what it does capture and what it does not. This will be represented in a table with the following columns: test id, test area (front end, back end etc), component to be tested (function name etc), test description, test inputs, test expected outputs and date of creation. This document will also provide a historical viewpoint of our testing philosophies and standards. Which will be valuable to any new developers who join the project, ensuring they are quickly brought up to speed and quality testing contributions.

7) Conclusion

This proposal outlines a comprehensive framework for our project focused on creating a website for the interactive visualisation of spatial data, specifically rainfall and temperature information across Australia taken from the BOM’s site. We provided background of thorough literature review on project management concepts, established the project's rationale in comparison to similar websites, and justified the need for a user-friendly platform tailored for non-technical users. Our project management plan defined our objectives, scope, through an RTM and user acceptance criteria, while detailing our structured approach to risk management and stakeholder communication. We adopted the Scrum software PM under the Agile framework, and assigned specific responsibilities to ensure accountability throughout the project lifecycle. Additionally, we presented a detailed schedule of project execution through a WBS and Gantt chart, and provided estimates on our resource requirements. Our external design included our database conceptual model and wireframe representations. Our methodology outlines our front end and back end tools, algorithms, and integration strategies to be employed in the development process. Lastly, we highlight our testing plan, focusing on front end UI, backend API endpoints and our BOM web scraper.

8) References

Adcyber. (2023, June 22). Mitigating Cyber Threats With Tara's Risk Response Approach - Cyber Insight. *Cyber Insight*. https://cyberinsight.co/what-is-tara-approach-to-risk-response/

Aranda, N. R., De Waegemaeker, J., & Van De Weghe, N. (2023). The evolution of public participation GIS (PPGIS) barriers in spatial planning practice. *Applied Geography*, *155*, 102940. https://doi.org/10.1016/j.apgeog.2023.102940

Beedle, M., Beck, K., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile software development*. https://agilemanifesto.org/

Corry, M. D., Frick, T. W., & Hansen, L. (1997). User-centered design and usability testing of a web site: An illustrative case study. *Educational Technology Research and Development,* 45(4), 65–76. https://doi.org/10.1007/bf02299683

Dalangin, M. A. (2022, April 14). *Continuous user testing | Userpeek.com*. Userpeek.com. https://userpeek.com/blog/continuous-user-testing/

Ethics Standards Development Committee (ESDC). (2006, October). *PMI Code of Ethics and Professional Conduct*. Project Management Institute.

Fireship. (2021, September 21). *D3.js in 100 Seconds* [Video]. YouTube. https://www.youtube.com/watch?v=bp2GF8XcJdY

Gordon, K. (2024, February 8). *5 Principles of Visual Design in UX*. Nielsen Norman Group. https://www.nngroup.com/articles/principles-visual-design/

Haklay, M. (2010). *Interacting with Geospatial Technologies*. John Wiley & Sons.

Hillery, A. (2020, October 19). *The evolution of data visualization*. Chartio. https://chartio.com/blog/the-evolution-of-data-visualization/

Interaction Design Foundation - IxDF. (2024, April 25). *What is User Centered Design (UCD)?* https://www.interaction-design.org/literature/topics/user-centered-design

Kerzner, H. (2017). *Project management: A Systems Approach to Planning, Scheduling, and Controlling*. John Wiley & Sons.

Mapbox. (n.d.). *About MapBox | Location Intelligence for Business*. https://www.mapbox.com/company

Nivala, A., Brewster, S., & Sarjakoski, T. L. (2008). Usability evaluation of web mapping sites. *The Cartographic Journal*, *45*(2), 129–138. https://doi.org/10.1179/174327708x305120

Norman, D. A., & Draper, S. W. (1986). *User centered system design: New Perspectives on Human-computer Interaction*. Hillsdale, N.J. : L. Erlbaum Associates.

Nyerges, T. (2009). GIS and Society. In *International Encyclopedia of Human Geography* (pp. 506–512). https://doi.org/10.1016/b978-008044910-4.00694-5

Project Management Institute. (2021). *A Guide to the Project Management Body of Knowledge (Pmbok(r) Guide) - Seventh Edition*. Pmbok(r) Guide.

Provos, N., Mazières, D., & The OpenBSD Project. (1999). A Future-Adaptable password scheme. In *Proceedings of the FREENIX Track: 1999 USENIX Annual Technical Conference*. https://www.usenix.org/legacy/events/usenix99/full_papers/provos/provos.pdf

Radigan, B. D. (n.d.). *Four agile ceremonies, demystified | Atlassian*. Atlassian. https://www.atlassian.com/agile/scrum/ceremonies

Raut, P., & Singh, V. P. (2024). Enhancing accessibility and usability of government websites. *International Journal for Research in Applied Science and Engineering Technology*, *12*(3), 151–155. https://doi.org/10.22214/ijraset.2024.58751

Rehkopf, B. M. (n.d.-a). *Scrum Master Roles and Responsibilities | Atlassian*. Atlassian. https://www.atlassian.com/agile/scrum/scrum-master

Rehkopf, B. M. (n.d.-b). *Scrum Sprints: Everything You need to know | Atlassian*. Atlassian. https://www.atlassian.com/agile/scrum/sprints

Salazar Cardozo, A., Glow, B., Nguyen, V., Trejo, D., Westerveld, A., Côté-Roy, V., Hilaiel, L., Shtylman, R., Graboys, V., Noordhuis, B., Rajlich, N., McArthur, S., Oosthuysen, F., Mahapatra, A. S., & Del Gobbo, N. (n.d.). *BCrypt*. Npm. https://www.npmjs.com/package/bcrypt

Schneier, B. (1994). Description of a new variable-length key, 64-bit block cipher (Blowfish). In *Lecture notes in computer science* (pp. 191–204). https://doi.org/10.1007/3-540-58108-1_24

U.S. Geological Survey. (2004, November 18). *What is a geographic information system (GIS)?* https://www.usgs.gov/faqs/what-geographic-information-system-gis

Web Content Accessibility Guidelines (WCAG) 2.1. (2023, September 21). https://www.w3.org/TR/WCAG21/

West, B. D. (n.d.). *A Deep Dive into Scrum Team Roles | Atlassian*. Atlassian. https://www.atlassian.com/agile/scrum/roles

Windyty, SE. (n.d.). *Windy*. Windy.com/. https://www.windy.com/-Waves-waves

Yablonski, J. (n.d.). *Laws of UX*. Laws of UX. https://lawsofux.com/

9) Appendix

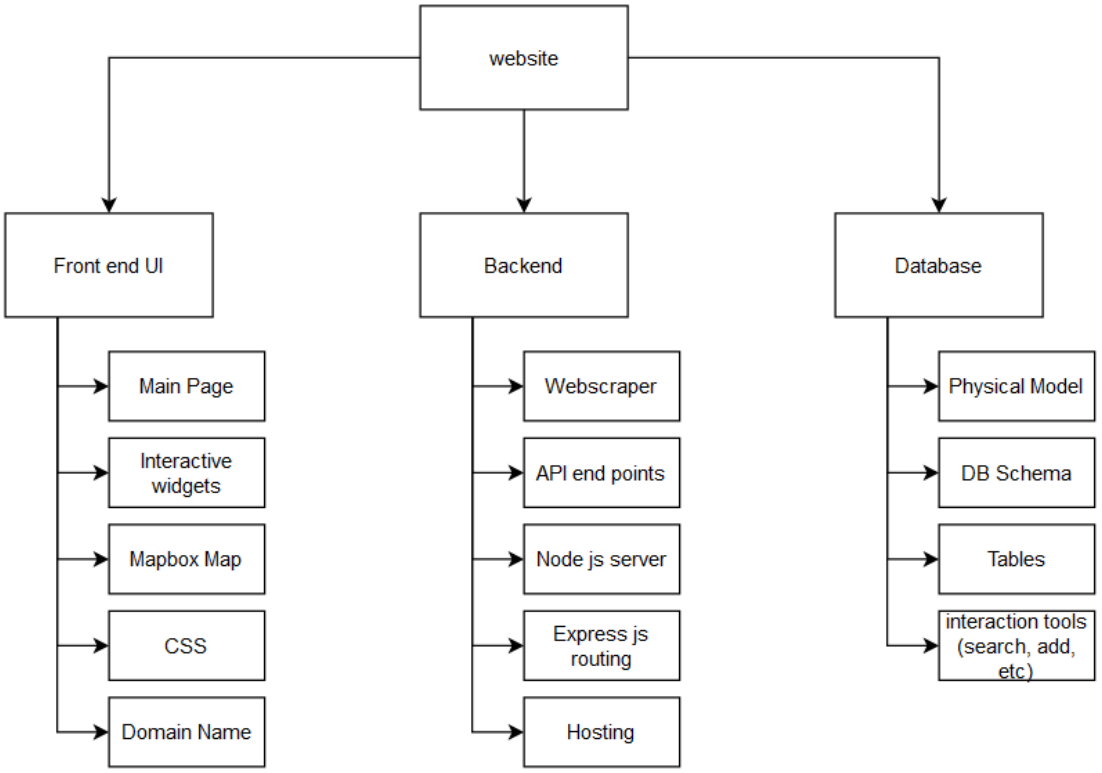

Figure A.1: Developed WBS for developing the website.

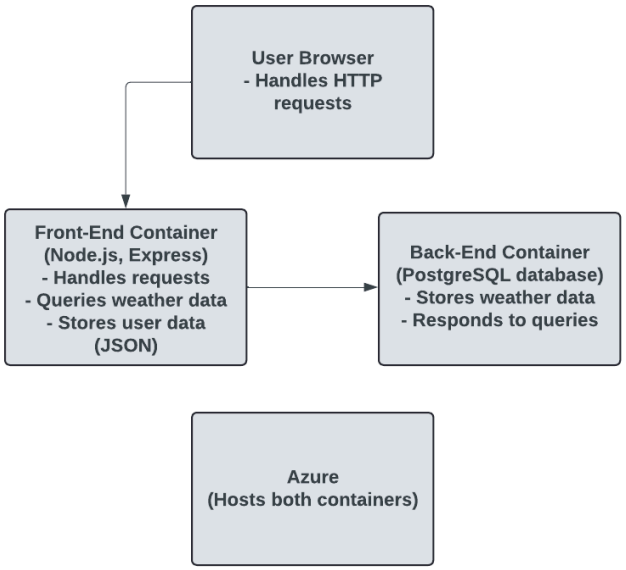

Figure A.2: Visualisation of the Docker environment.

ID	Information			Response		Owner	Risk Assessment				Last Updated
	Description	**Reason**	**Trigger**	**TARA(E)**	**Action**		**%**	**I****	**Total**	**Stat***	
1	Evolving web standards	Website standards changed based on user needs to enhance accessibility, ensure compatibility across devices, and address security concerns. Standards are updated to improve practices, tools, and frameworks.	Emergence of new technologies, changes in user behaviour, or regulatory requirements (eg. WGAC). Feedback from users or developers regarding usability issues can prompt a reevaluation of existing requirements.	Transference. Stakeholder collaboration to assess the new standards, and identify areas for improvement. Research new technologies, gathering user feedback, and examining practices in similar sites. Acceptance once the development process is over.	Revise technical documentation and guidelines to reflect new standards. Provide training for developers on new technologies and standards to implement changes. Integrate updated standards into the website, such as adopting responsive design, enhancing accessibility features, or utilising performance optimisation techniques.	Product owner	90%	6	540	M	28/10/2024 (Feba)
2	Missed deadline	Lack of experience	Missed milestone	Transference. Provide training, and assign tasks within skill sets.	Reallocate workload	Project manager	60%	7	420	M	26/10/2024 (Feba)

3	Feature does not meet client's expectations	Communication breakdown with client	Find out during technical demonstration.	Avoidance, Primary way to circumvent this risk is improving client - developer communication. We will frequently engage with the client, conducting a tech demo at the end of each milestone. Project owner will consult with client before every project increment, taking notes on their requirements and getting them signed off + confirmed by another attendee.	Take stock of how big of a differential there is between client expectation and project reality. Utilise a SPIKE to get quick feedback from the client about alternative potential solutions. Otherwise delegate to project owner to reevaluate approach for upcoming project increment. Where it can be added to backlog.	Project Owner	70%	3	210	M	27/10/2024 (Maddy)
4	Data accidentally deleted.	A developer accidentally deletes files from the server.	Users can not log in. Users notice their saved maps or graphs no longer exist. Weather data not appearing on the site.	Reduction. Consistently store backups of data so that in the unfortunate event that data is deleted, recent backups can be used to at least partially rectify the deletion.	Attempt to retrieve the lost data from the server. If the data has been permanently lost, restore users data back to the most recent backup, rescrape lost old BOM data from BOM site and place an announcement on the site informing users of the mistake.	Product owner	25%	8	200	M	27/10/2024 (Tyler)

5	Team member suffers from burn out	Overworked, lack of support or external factors reducing their ability to function.	Team member becomes more withdrawn in meetings, lacking enthusiasm. Code velocity drops. May become irritable / frustrated.	Avoidance, burn out typically comes about due to being overworked, lacking support or external pressures limiting ones capacity. As such we will focus on regular and open communication to nip these problems in the bud early. During each program increment we will utilise the 5 fingers confidence vote method to determine if everyone agrees with work allocation (only moving forward if everyone >= 4). Conducting in depth retrospectives at the end of each milestone. Asking developers if they believe work allocations were or were not appropriate? What other support could assist them in performing to their best ability.	Immediately reallocate majority of work away from affected team member. Take time to discuss what could be changed in future to avoid situation happening again.	Scrum Master	30%	6	180	M	27/10/2024 (Maddy)

6	Poor usability and UI choices.	Developers poorly adhered to design and usability principles.	Users having difficulty navigating the site UI.	Transference. Work with usability experts who are more familiar with UI that are mind-less to navigate.	Obtain feedback from users. Front-end developers may collaborate with a usability expert to modify the UI.	Scrum Master and Team Members	40%	4	160	M	27/10/2024 (Tyler)
7	Developer or intermediate family of developer falls ill for a prolonged period	Serious health conditions including chronic illnesses, significant surgeries, or other medical issues requiring extended recovery time.	Formal communication of illness. Noticeable changes in work performance.	Transference. Reallocate workload.	Redistribute work. Communicate with the developer for any updates. Monitor and support developers. Develop a reintegration plan into the team.	Team member	20%	7	140	M	28/10/2024 (Feba)
8	GIT merge conflicts.	Two developers with separate branches have merged on to the same segment of code.	Code does not function as intended.	Reduction. Review changes from other developers before committing to GIT.	Developers involved in the conflict will rollback changes and work together to merge their commits without conflict.	Team Member	30%	4	120	M	27/10/2024 (Tyler)
9	Azure free tier no longer has enough bandwidth / requests for website	Spike in website popularity or Azure strips back the free tier	Service disruptions, Long website load times + HTTP 500 errors on requests.	Mitigation, We will dockerise our application so in the case where Azure no longer provides the hosting capabilities we provide we can	Discuss potential of upgrading to a paid plan. If there is no budget to support this then begin research into alternative hosting providers. In the	Product owner	10	9	90	M	27/10/2024 (Maddy)

				easily migrate to another hosting service provider. (Hosting services often try lock you in to their ecosystem, making it hard to leave)	meantime we can limit computationally expensive functionality (such as creating heat/chloropleth maps), and focus on delivering quality uninterrupted core features (fetching data, creating line charts / dot plots).						
10	BOM is no longer scrapeable (i.e. updates robots.txt and actively blocks scrapers).	Lacking server infrastructure or personal choice.	Webscraper fails test cases. Daily weather data no longer being updated	Acceptance, due to low likelihood and lack of avoidance / mitigation strategies. Best thing we can do is react to this risk coming to fruition.	Update website to notify users that new data is not being collected. Delegate Maddy the role of finding a new weather data source and building a scraper for it (as she has most experience)	Scrum master	10%	7	70	M	26/10/2024 (Maddy)
11	Sensitive user information is maliciously leaked.	A security vulnerability is present in our application, allowing a malicious party to obtain sensitive user information.	User data published publicly from a source not within our team.	Transference. Outsource a security expert to identify vulnerabilities in our application before, or after they are exploited.	Temporarily shut down the site, preventing new user information from being leaked. Seek advice from a penetration tester to locate and fix the vulnerability. Relaunch the site with an announcement for users on the leak that has	Product owner	5%	10	50	M	27/10/2024 (Tyler)

					occurred.						
12	DNS spoofing	Attacker manipulates the DNS cache to redirect users from legitimate websites to malicious ones. Results in stolen sensitive, distribution of malware, or conduct phishing attacks. Consequence of weaknesses in DNS protocols, misconfigurations, or lack of security measures.	Users report issues accessing certain websites. Abnormal traffic patterns detected by security monitoring systems. Alerts from security tools that identify potential cache poisoning.	Mitigation. Assess the extent of the attack, identify compromised DNS servers, and gather data on affected users and systems.	Clearing DNS Cache on affected servers to remove the poisoned entries and restore correct mappings. Investigate how the spoofing occurred, and reconfigure DNS settings to prevent future vulnerabilities. Deploying DNSSEC to enhance security. Inform users of the incident, and recommending changes to DNS settings if necessary.	Product Owner	5%	9	45	M	28/10/2024 (Feba)
13	Team member drops out of FIT3161/FIT3162.	Likely personal reasons, has to defer etc.	Hopefully communication from respective team member. Otherwise radio silence + confirmation with teaching team.	Mitigation, Focus on minimising siloing. Can be done by providing detailed documentation for all code + setup / handover documentation of how to use a program	Triage tasks that were to be completed by absent member. Anything that would block future feature or is critical for the client will be immediately reallocated during the current sprint. Anything else	Scrum Master	5%	8	40	M	27/10/2024 (Maddy)

				/ tool. We will also run technical meetings to catch up members who are having difficulty understanding a piece of software.	that is not critical will be added to the project backlog and picked up next sprint.						
14	Site experiences DDOS attack.	A third party maliciously floods the site with network requests, to slow it down or crash it.	Slow site performance. Unusual number of network requests, potentially exceeding server capacity.	Avoidance. Implement some form of DDOS protection.	Implement some form of DDOS protection into Azure. If not possible with the current tier or server, we will consider alternative tiers or servers that offer this protection.	Product owner	5%	8	40	M	27/10/2024 (Tyler)
15	New weather data can not be retrieved from the database.	BOM may decide to change the way their .csv files containing weather data are structured.	New temperature and rainfall information not appearing on the site.	Acceptance. The likelihood that BOM data changes structure is unlikely, therefore it is not necessary to prepare for such an event.	For dates prior to the changes to BOM data structure, use the original database querying algorithm. For new BOM data, use an updated querying algorithm in accordance with changes to tables, variables, relationships etc.	Product owner	5%	6	30	M	27/10/2024 (Tyler)
16	Hashed user passwords are no longer	Technology improves in dehashing bcrypt	Concern arises over how effective bcrypt password	Acceptance. It is difficult to prepare for concerns over the	Attempt to implement a more secure password hashing package such as	Product owner	4%	6	24	M	27/10/2024 (Tyler)

	difficult to decrypt.	passwords.	hashing is.	strength of bcrypt until technology progresses to a point where our current hashing algorithm is vulnerable.	Argon2.						

Table A.1: Risk register of the documented risks identified. M - Monitored, T - Triggered, R - Resolved, L - Lapsed. ** I - Impact score*

ID	Category	NFR or FR	Source	Description	Acceptance criteria ("Definition of Done")	Status
1	Front End	FR	Business Proposal Document	As a user I want a main web page that will allow me to use the primary interactive widgets.	Structured layout following the best practices for effective interface. This includes: - Consistent functionality in repeated components. - Accessible sizing for ease of readability. - Duplicate actions are reduced for the same functionality. - Similar elements and components organised together.	Not started
2	Front End	FR	Project Description	As a user I want to be able to visualise the historical weather data on a map of Australia so I can get an intuitive understanding of data trends.	- Map utilises heat maps to overlay historical weather data onto geospatial location. For data without locality, appropriate popup idiom is used. - A sliding scale allows users to focus on specific points within the historical timeline.	Not started
3	Front End	FR	Business Proposal Document	As a user I want to be able to navigate the web page smoothly.	- Loading data and interacting with the map is smooth and intuitive to the user. - Navigating the website is also very straightforward through structure, and accompanying visual elements.	Not started

4	Front End	FR	Inferred from users	As a user I want to be able to download the data as a graphical picture and as a CSV file so that I can manipulate, and display the information as I please.	- Data from a selected region can be downloaded as a CSV, PNG, and JPG file from the website. - Each field is clearly defined by the labelled graph axes to indicate the content of the CSV to the user. - The graph created by the user can be saved and loaded by the user if they wish to sign up with their own account.	Not started
5	Back End	FR	Project Description	As a user I want to interact intuitively with the map to view trends in temperature and rainfall via visualisations.	- The platform provides the option to switch between heatmaps, scatter plots, line charts, and histograms for flexible data interpretation. - Interactive map layers enable toggling between temperature and rainfall datasets. - Tooltips and pop-ups display detailed information when hovering over specific map points or regions.	Not started
6	Front End	NFR	Inferred from project description	As a user I want to be able to pan and zoom on the map.	- Click, hold, and drag are used to pan map. - Mouse scroll wheel can be used to zoom in.	Not started
7	Front End	FR and NFR	Discussion with user	As a user I want all map actions to be smooth so that it causes minimal strain on my eyes.	- Map box panning is smooth (particularly on low power devices). - Dark mode, and other visually aiding colour schemes are compatible with the design.	Not started
8	Front End	NFR	Inferred from project description	As a user I want all visualisations to be easy to interpret.	All visualisations must - be colour blind friendly, - have well labelled with axes and titles, - use appropriate idioms for their respective data (ie. idioms with nominal data are not placed on an interval x axis), and - not be cluttered (ie. does not display more than 6 unique categories	Not started

					of data).	
9	Front End	NFR	Discussion with user	As a user I want the page to load within 3 seconds.	Page loads within 3 seconds (assuming normal internet speed > 10Mb/s.	Not started
10	Front End	FR	Inferred from project description	As a user, I want to filter and display specific types of data for more targeted analysis.	- A search bar allows users to quickly locate and zoom into specific locations by name or coordinates. - An advanced search option can filter areas that meet a certain condition. (e.g. only show areas above a certain amount of rainfall or temperature).	Not started

Table A.2: Requirement Traceability Matrix of the project as user centric specifications.