SUPPORT DOCUMENT

Support document for bushfire brigade website.

Team TA30
Next Gen Innovators

Contents

1 Introduction

The Bushfire Brigade Support Document aims to provide essential information for support personnel, IT staff, and other technical teams to resolve any technical issues that may arise during the operation of the website. This document offers sponsors a step-by-step guide to independently manage and sustain the project after handover.

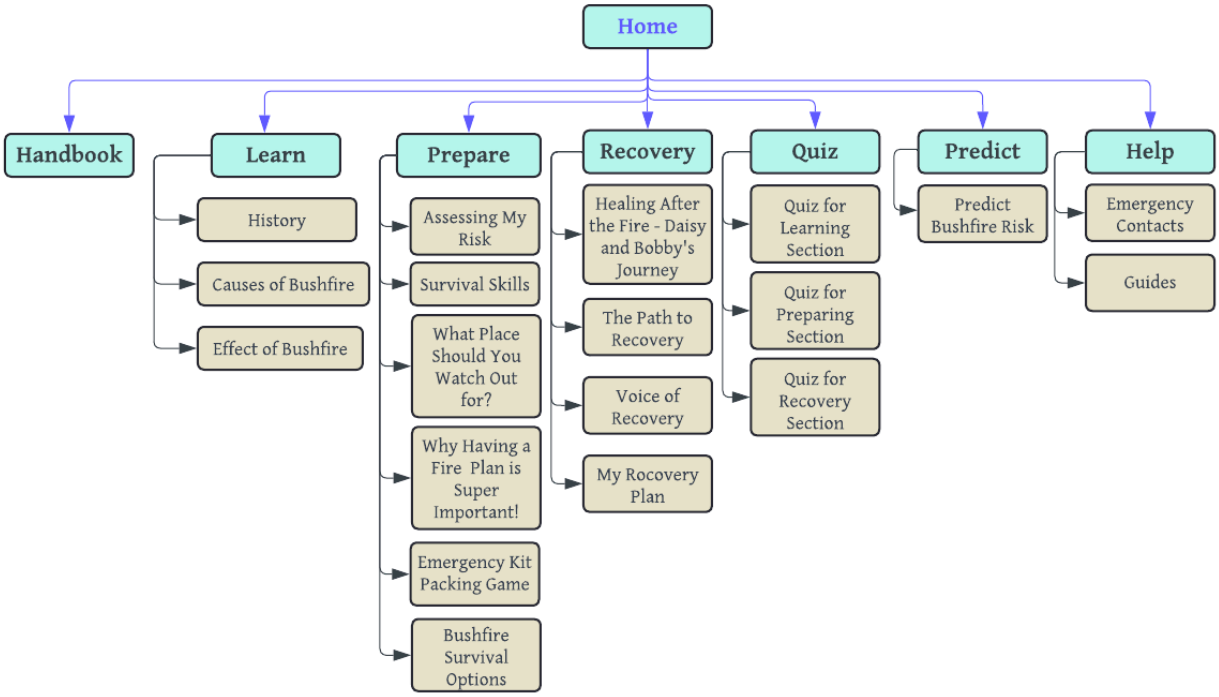
Next-Gen Innovator is confident that the future owner of the Bushfire Brigade website will use it to further enhance bushfire education for children, equipping them with the knowledge and skills to build resilience against the growing bushfire risks, not only in Victoria but across Australia.

2 Installation

Step	Description	Command/Action
Step 1: Clone the Repository	Clone the bushfire repository from GitHub to your local machine.	• Open a terminal or command prompt on your machine. • Navigate to the folder where you want to clone the project. • Run the following command to clone the repository: git clone https://github.com/PoomSira/bushfire.git • Change directory into the project: cd bushfire
Step 2: Install Dependencies	Install all necessary project dependencies using npm	• Inside the project directory, run the following command to install the necessary dependencies using npm: npm install
Step 4: Start the Development Server	Start the Next.js development server locally.	• To start the development server, run the following command: npm run dev • Once the server starts, you should see output indicating the app is

		running. The default URL is: http://localhost:3000 • Open a browser and navigate to http://localhost:3000 to see the application running locally.

3 Product Sitemap



Please refer to the user manual for a detailed explanation of the User Manual.

4 Support

The platform is hosted on **Vercel**, with the **PostgreSQL** database managed via **Neon.tech**. This setup ensures a highly scalable and secure infrastructure, optimized for performance and reliability.

4.1 Website Hosting on Vercel

Hosting Location	The platform is hosted in the Sydney, Australia region, ensuring low latency for users in the Asia-Pacific area.
Domain Management	The primary domain for the platform is

	bushfire-brigade.me, which is configured on Vercel with a permanent redirect to www.bushfire-brigade.me for better SEO and consistent user experience.
Deployment & Branching	The platform uses Vercel's seamless integration with GitHub, allowing for continuous deployment. Any new commits pushed to the GitHub repository are automatically deployed, ensuring that the latest features and bug fixes are live.

4.2 Database Management with Neon.tech PostgreSQL:

Endpoint	The database is hosted at the endpoint **ep-frosty-sea-a71uc8or-pooler.ap-southeast-2.aws.neon.tech**, which facilitates secure and optimized access to the PostgreSQL database.
Database Backup	Regular backups of the database are automated through Neon.tech. This ensures that data can be recovered in the event of any incident. The backups are retained for a defined period, typically allowing for recovery from any point in time.
Storage & Compute Usage	The current database size is 34MB, with a compute time of approximately 3.49 hours. The data transfer remains minimal, ensuring optimized usage of resources.
PostgreSQL Access	Secure access to the PostgreSQL database is provided through Prisma ORM, which ensures that the database queries are optimized, and SQL injections are prevented by using parameterized queries.

4.3 Performance Monitoring

Vercel Dashboard	Vercel provides real-time insights into the platform's performance, including website load times, server response times, and resource usage.

Database Performance	The database performance, including data transfer rates and compute time, is continuously monitored via the Neon.tech dashboard to ensure that it operates within optimal parameters.

5 Back up

The platform utilizes **GitHub** for version control and **Vercel** for database backups, ensuring that both code and data are securely stored and can be restored efficiently in case of issues.

5.1 GitHub for Version Control

- **Code Backup**: All versions of the platform's code are stored and managed on GitHub repositories. Each code commit is tracked, enabling seamless version control. This means that any update or feature change can be rolled back to previous stable versions in case of bugs or failures.
- **Branching Strategy**: A **feature branching** approach is used in GitHub to manage updates. Main branches, such as main or develop are reserved for stable code, while new features are developed in separate branches to avoid conflicts.
- **Continuous Integration/Continuous Deployment (CI/CD)**: GitHub Actions is used to automate testing and deployment processes, ensuring that only thoroughly tested code is deployed to the production environment on Vercel.

5.2 Vercel for Database Backup

- **Database Backup on Vercel**: The PostgreSQL database is hosted and backed up regularly on Vercel. Automated database backups are scheduled at regular intervals (e.g., daily or weekly), ensuring that no data is lost, even in the case of a server or system failure.
- **Backup Retention**: Vercel's database backup policy is configured to retain backups for a certain period (e.g., 30 days), allowing recovery of data from different points in time.
- **Data Export**: Periodic database exports can be made from Vercel's database management interface. These exports allow the project administrators to download the database contents as SQL files, which can be re-imported to a different database or used for disaster recovery.

5.3 Rolling Back Changes with GitHub and Vercel

- **Git Rollback**: If any bugs or issues are introduced during an update, the team can quickly revert to a previous version using GitHub's version control features. All previous commits and pull requests are tracked, allowing the team to roll back any unwanted changes.
- **Database Rollback**: In case of data corruption or database-related issues, backups from Vercel can be restored to a previous stable state. The restore process is streamlined through Vercel's database management interface, ensuring minimal downtime.

5.4 Regular Security Audits

- **Code Audits**: Regular security audits of the codebase are conducted on GitHub using automated tools such as Dependabot, which checks for vulnerabilities in the code dependencies.
- **Database Audits**: Vercel provides logging and monitoring tools that track database access and suspicious activity, ensuring the database is protected from unauthorized access or security breaches.

5.5 Patch Management and Updates

- **Web Server and PostgreSQL Updates**: Both the Vercel platform and PostgreSQL database are automatically patched to the latest versions, ensuring they are protected from newly discovered vulnerabilities. Vercel provides regular updates to its infrastructure without requiring manual intervention.
- **Codebase Updates**: For updates to the codebase, all new code passes through a CI/CD pipeline on GitHub. Unit and integration tests are automatically run before the deployment to prevent introducing bugs into the production environment.

5.6 Disaster Recovery Plan

- **Disaster Scenarios**: In the event of major incidents like server failure, security breach, or accidental data loss, the team can rely on both GitHub's version control and Vercel's database backups to restore the platform quickly and efficiently.
- **Recovery Procedure**: The recovery process involves rolling back the code to the last stable commit on GitHub and restoring the latest database backup from Vercel. The downtime is minimized due to the cloud-based nature of the platform, and user data is safeguarded through redundant backups.

6 Security

The security of the platform is prioritized to safeguard the personal information of users and protect against threats such as **SQL injection**, **Cross-Site Scripting (XSS)**, and **data breaches**. Additionally, **HTTPS** is enforced across the platform to ensure secure communication between the server and clients.

6.1 SQL Injection Protection

- The platform utilizes Prisma ORM with parameterized queries to ensure that all database interactions are secure. This mitigates the risk of SQL injection attacks by ensuring that malicious inputs are not executed as SQL commands.

6.2 Cross-Site Scripting (XSS) Protection

- Input validation and output encoding are implemented to prevent XSS attacks. A Content Security Policy (CSP) is also in place, further reducing the risk of harmful scripts being executed in user-generated content such as blog posts and comments.

6.3 Database Security

- Least privilege principles are applied to database access, ensuring that only authorized users can access and modify data. Additionally, the database is protected with robust firewall rules and is continuously monitored for suspicious activity.

6.4 HTTPS & SSL Certificates

- The platform is secured with HTTPS, ensuring that all communication between the users and the server is encrypted using SSL/TLS protocols. This prevents eavesdropping, tampering, and man-in-the-middle attacks, ensuring the confidentiality and integrity of user data.
- SSL Certificates are automatically managed and renewed by Vercel. The platform uses HTTP Strict Transport Security (HSTS) headers, forcing browsers to connect using HTTPS and preventing any attempts to downgrade the connection to an insecure HTTP version.

6.5 Content Filtering

- The blog feature, where users can post without logging in, is secured by filtering content to ensure that inappropriate or malicious content is not displayed. All user inputs are validated and sanitized before being stored or displayed.

6.6 Real-Time Monitoring and Audits

- Regular audits of the codebase and database are conducted to identify potential vulnerabilities. Real-time monitoring and alert systems are in place to detect and respond to suspicious activity, ensuring prompt action in case of a security breach.

6.7 Data Encryption in Transit and at Rest

- All sensitive data is encrypted in transit using SSL/TLS encryption, ensuring that data is protected as it moves between the client and server. The database also employs encryption at rest, ensuring that stored data remains protected from unauthorized access.

7 Data Management

7.1 Data Overview

No	*Dataset*	*Source*	*Physical Access*	*Granularity*	*License*
1	*School locations in Victoria*	*https://discover.data.vic.gov.au/dataset/school-locations-2023/resource/92fdd072-4666-4cc6-a28a-749c826297a7*	*CSV*	*High - Geocodes for School location available*	*Creative Commons Attribution 4.0 International*

2	*Enrollments in schools in Victoria*	*https://discover.data.vic.gov.au/dataset/all-schools-fte-enrolments-feb-2023-victoria/resource/64c14b8a-dbe9-4c28-9f75-c5689a727b80*	*CSV*	*High - Number of student enrollments for each grade in each school available*	*Creative Commons Attribution 4.0 International*
3	*Bushfire risk school registry*	*https://www.vic.gov.au/bushfire-risk-register-barr*	*CSV*	*High - Risk category for each high-risk school within Victoria available*	*Creative Commons Attribution 4.0 International*
4	*Bushfire scar history*	*https://discover.data.vic.gov.au/dataset/fire-history-records-of-fires-across-victoria-showing-the-fire-scars1*	*shapefile*	*High - Bushfire by date and location available*	*Creative Commons Attribution 4.0 International*
5	*Weather Station Dataset*	*http://www.bom.gov.au/vic/observations/map.shtml*	Text or pdf	Medium - The weather station dataset contains information on 17,935 stations, including their operational status. Only the relevant stations will be queried and processed into a CSV file.	*Creative Commons Attribution 4.0 International*
6	*Weather Dataset*	*https://webarchive.nla.gov.au/awa/20040803104956/http://www.bom.gov.au/climate/dwo/IDCJDW0300.shtml*	CSV	High-The dataset contains daily weather parameters collected by identified stations, with 20 parameters observed from year 2014-2023.	*Creative Commons Attribution 4.0 International*

7.2 Data Usage

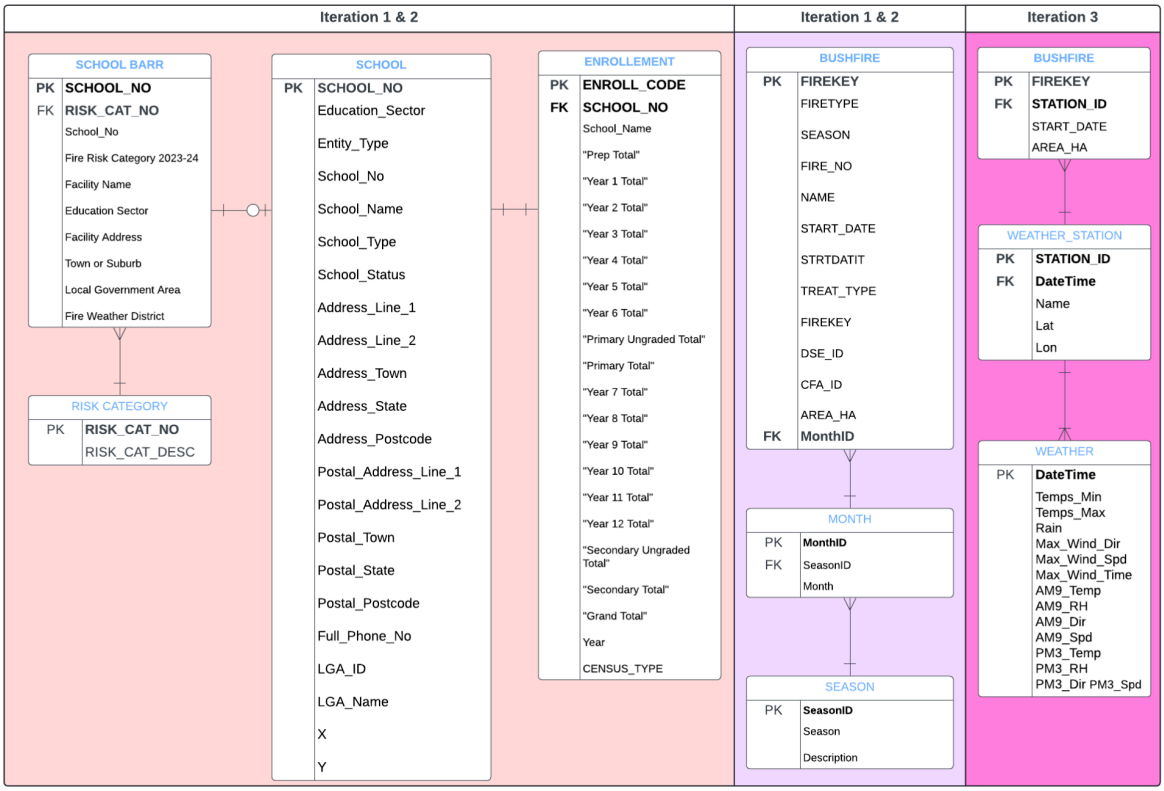
Learn Page: The datasets for the visualizations in the "Effects of Bushfires" section aim to help children understand the trends and effects of bushfires on schools across Victoria. The *Bushfire History Timeline* section shows the changing trend of bushfires from 2000 to 2023, with an aggregated count of bushfires by year to illustrate the number of occurrences over time. In the *Schools Affected by Bushfires* section, a pictograph bar plot displays the number of schools in different risk categories, with the schools also grouped by these categories. Finally, in the *Bushfire Risk Areas* section, a drill-down bubble plot allows children to explore schools grouped into bubbles, which can be further drilled down to individual schools.

Prepare Page: In the *Assessing My Risk* section, most datasets are reused to show the number of bushfires in different seasons and the locations of schools in relation to bushfire scar areas. The map highlights areas that were previously affected by bushfires. The bushfire scar area dataset is stored as a separate GeoJSON file in the database for use in map visualizations.

Predict Page:Building the machine learning model for prediction required a new relational dataset. This dataset was created by merging bushfire scar history centroids with weather conditions, based on the date and weather station name. The weather parameters used for training the model went through feature selection, resulting in the most relevant and important features, as shown in the database schema under the entity *weather*.

7.3 Database Schema

The database schema below illustrates all the datasets utilized throughout each iteration of the project. Note that Iteration 1 and Iteration 2 primarily use the same datasets. However, for the map integration, the underlying bushfire scar polygons are stored as separate entities in a GeoJSON file.



7.4 Data Preparation

The details of the data preparation process can be found in the Data Management Plan. The dataset and Jupyter Notebook are accessible via the links below:

- Learn Page: This page is part of iteration 1, focusing on the learning about bushfire themes. It aims to simplify scientific concepts for children through engaging visualizations.

- Prepare Page: This page is part of iteration 2, focusing on preparation for the bushfire theme. Under the epic that helps children assess their risks by identifying potential bushfire hazards.
- Predict Page: This page is part of iteration 3, under the epic "Weather-Based Bushfire Simulation." It allows children to observe how different weather conditions can influence the likelihood of bushfire occurrences.

Datasets and Jupyter Notebook link:

Datasets:https://drive.google.com/drive/folders/1orErT3fh3eiEgBVuwAx445nOTBvymOzk?usp=sharing

Data Processing Code:
https://drive.google.com/drive/folders/1QvTfjkRkxbJDGASuIWqNZk2eN91504ll?usp=sharing

Machine Learning:
https://drive.google.com/drive/folders/1iVJYupcrNmC62GMe2vQoiXm1nauIRXop?usp=sharing