

# **Heatmap Project — User Manual**

Version: 1.0

Date: 2025-11-19

Author: Heatmap Project Team

Template reference: Structure inspired by GitHub and Microsoft user manual layouts; content is original and specific to this project.

- <https://docs.github.com/en/repositories/managing-your-repository-settings-and-features/customizing-your-repository/about-readmes>
- <https://github.com/othneildrew/Best-README-Template>
- <https://learn.microsoft.com/en-us/style-guide/welcome/>

Team Members:

E/21/113

E/21/141

E/21/193

E/21/407

## **Overview:**

This User Manual helps you install, understand, and operate the Heatmap Project — a web application that visualizes foot traffic data on a map-like heatmap and lets administrators manage exhibits for each zone.

Audience: This manual is written for non-technical users and administrators who need to run, operate, and troubleshoot the system on a Windows development or deployment machine.

What this manual covers:

- System components and high-level architecture
- Step-by-step installation and configuration on Windows (PowerShell commands included)
- How to run the backend and frontend on your machine
- How to use the application: viewing the heatmap, adding/removing exhibits, search, and admin actions
- Troubleshooting and frequently asked questions
- How to convert this manual to PDF and share it

## **1. System Components (What you will run)**

The project contains two main components:

1. Backend (Node.js + Express) — provides REST APIs, connects to a PostgreSQL database, and serves data to the frontend.

- Location: `backend/` folder
- Key files: `server.js`, `zoneInfoAPI.js`, `db.js`, `createTables.js`

2. Frontend (React + Vite) — user interface that shows the heatmap and admin modals.

- Location: `frontend/` folder

- Key files: `src/components/Heatmap.jsx`, `index.html`

Database: PostgreSQL. The SQL schema used by the app is in `backend/sql/create\_zone\_info.sql`.

Ports used by default:

- Backend: 2000 (configurable via `PORT` environment variable)
- Frontend (Vite dev server): 5173 (default for Vite)

Prerequisites (what you need installed):

- Node.js (LTS 18.x or 20.x recommended)
- npm (comes with Node.js) or pnpm
- PostgreSQL (a running server instance, locally or accessible remotely)
- (Optional for PDF conversion) Pandoc and a PDF engine like wkhtmltopdf or use VS Code built-in Print to PDF

## **2. Quick start (Overview steps)**

This section gives the short, single-machine walkthrough for a user who wants to run the system locally.

1. Prepare PostgreSQL and create a database. Remember connection details (host, port, user, password, database name).
2. Open a PowerShell terminal and configure backend environment variables.
3. Install backend dependencies and run the backend server.
4. Install frontend dependencies and run the frontend.
5. Open the frontend URL in your browser (usually `http://localhost:5173`) and use the app.

Detailed commands are in the next section.

### 3. Installation and configuration (step-by-step)

This walkthrough assumes you are on Windows and using PowerShell.

#### 3.1. Clone or obtain the project

If you already have the code, open a PowerShell window and change directory to the project root (the folder containing `backend/` and `frontend/`):

```
```powershell
cd /d D:\2yp_project_final_codes_DiniraTuesday\heatmap-project````
```

#### 3.2. Prepare the PostgreSQL database

1. Install PostgreSQL if you don't have it

<https://www.postgresql.org/download/windows/>

2. Create a database for the project. For example, using psql:

```
```powershell
# Open the psql shell or use a GUI like pgAdmin
psql -U postgres
# inside psql
CREATE DATABASE heatmap_db;
-- create a user if needed
CREATE USER heatmap_user WITH PASSWORD 'your_password';
GRANT ALL PRIVILEGES ON DATABASE heatmap_db TO heatmap_user;
````
```

#### 3.3. Configure backend environment variables

In `backend/` the code reads configuration from environment variables. Create a `.env` file in `backend/` (do not commit secrets to version control). Example `.env` contents:

```
```PORT=2000
LOCAL_DB_URL=postgresql://heatmap_user:your_password@localhost:5432/heatmap_db
or DATABASE_URL=postgresql://... depending on your setup
````
```

```
DATABASE_URL=postgresql://postgres:Dinira123@localhost:5432/heatmap_final
PORT=2000
LOCAL_DB_URL=postgresql://postgres:Dinira123@localhost:5432/heatmap_final

GEMINI_API_KEY=AIzaSyDQF6qps0Avt3jplAkZSvWL1CBrLnk7v00
USE_POSTGIS=true
FRONTEND_ORIGIN=http://localhost:5173
SEED_COUNT=1000
```

### *3.4. Install and run the backend*

From the project root or `backend/` folder in PowerShell:

```
```powershell
cd /d D:\2yp_project_final_codes_DiniraTuesday\heatmap-project\backend
npm install
npm run dev
````
```

Notes:

- `npm run dev` should start the Express backend and listen on the configured port (default 2000). If you see errors, check the console output for missing environment variables or database connection errors.
- If `npm install` fails, check your Node.js version and network settings. Run `npm -v` and `node -v` to verify.

### *3.5. Apply database schema (if needed)*

If the database is empty, create the tables using the SQL script in `backend/sql/create\_zone\_info.sql`.

You can run it from `psql` or a GUI like pgAdmin:

```
```powershell
psql -U heatmap_user -d heatmap_db -f
"D:\2yp_project_final_codes_DiniraTuesday\heatmap-
project\backend\sql\create_zone_info.sql"
````
```

### *3.6. Install and run the frontend*

Open a second PowerShell window and run:

```
```powershell
cd /d D:\2yp_project_final_codes_DiniraTuesday\heatmap-project\frontend
npm install
npm run dev
````
```

Vite should start and show a local development URL, usually `http://localhost:5173`.

### *3.7. Open the app in a browser*

Open **http://localhost:5173** in your browser. If the frontend cannot reach the backend, confirm the backend is running on `http://localhost:2000` and CORS is enabled in the backend.

## 4. Using the application (user workflows)

### 4.1. Main screen — Heatmap overview

After opening the frontend, the main page shows a heatmap visualization of zones. The map overlays color-coded activity intensity per zone. Typical elements:

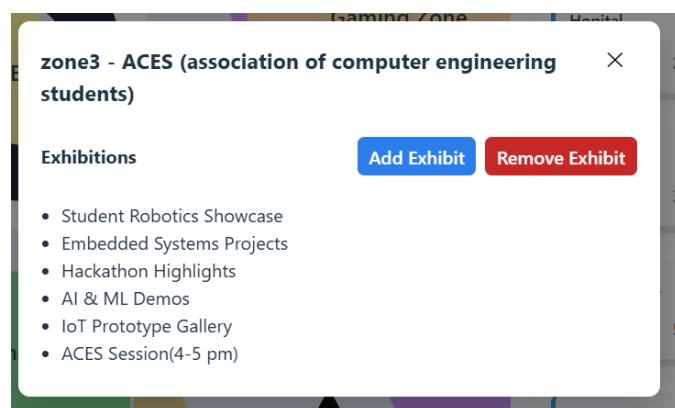
- Zone list and navigation
- Search box for local keyword search
- Buttons to open zone details or admin panels



### 4.2. Viewing zone details and exhibits

Click a zone (or use the list) to open its modal. The modal shows:

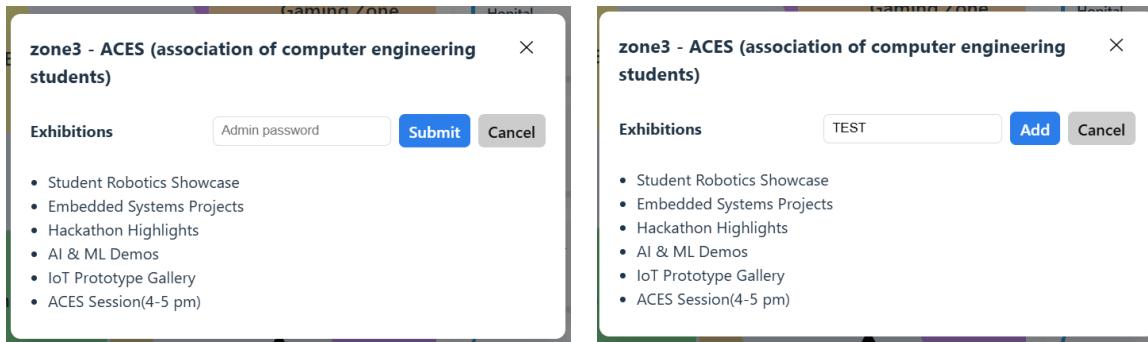
- List of exhibits in that zone
- Visitor counts (if available)
- Buttons to add or remove exhibits (admin only — password-protected)



### 4.3. Adding an exhibit (step-by-step)

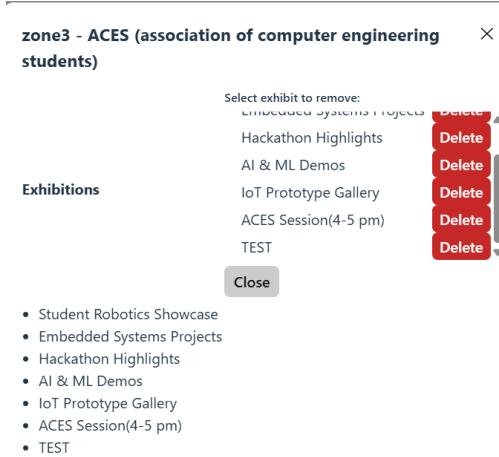
1. In the zone modal, click the Add Exhibit button.

2. If prompted, enter the admin password (the system may require authentication). Follow on-screen instructions to enter exhibit name and optional attributes.
3. Submit the form. The frontend will call the backend API to store the exhibit for that zone.
4. After a successful add, the modal will refresh to show the new exhibit.



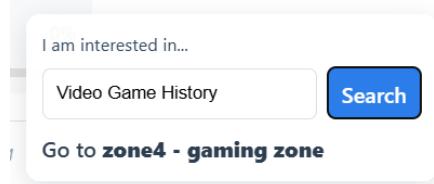
#### *4.4. Deleting an exhibit (step-by-step)*

1. In the zone modal, find the exhibit you want to remove.
2. Click the Delete (or Remove) button next to the exhibit.
3. Confirm the deletion in the confirmation dialog.
4. After confirmation, the system will call the backend API to remove the exhibit. The list will refresh after success.



## [4.5. Search](#)

Use the search box to find exhibits by keywords. The search uses a local algorithm and returns matching exhibits or zones. Enter a keyword and press Enter or click Search; results will appear below the search box or in the modal.



## [4.6. Admin notes](#)

- Admin actions (add/delete) may require a password. Ensure only authorized users have access.
- When performing database-affecting operations, the UI performs a refresh to show up-to-date information.

# **5. Administration and maintenance**

## [5.1. Changing backend configuration](#)

Set environment variables in `backend/.env` or the host environment:

- `PORT` — backend port
- `LOCAL\_DB\_URL` or `DATABASE\_URL` — PostgreSQL connection string

Restart the backend after making config changes.

## [5.2. Backups and DB maintenance](#)

Regularly backup your PostgreSQL database using `pg\_dump`:

```
```powershell
pg_dump -U heatmap_user -h localhost -p 5432 -F c -b -v -f
"heatmap_db_backup.backup" heatmap_db
````
```

## [5.3. Logs](#)

Backend logs appear in the console where you run `npm run dev`. For production deployment, configure a log aggregator or use process managers (PM2) and redirect logs to files.

## **6. Troubleshooting (common problems and fixes)**

Problem: Backend fails to start or shows `ECONNREFUSED` for the database

- Cause: Incorrect `LOCAL\_DB\_URL` or PostgreSQL not running
- Fix: Verify Postgres is running and `LOCAL\_DB\_URL` is correct. Test connection with psql.

Problem: Frontend cannot reach the backend (CORS or network error)

- Cause: Backend not running, wrong port, or CORS misconfiguration
- Fix: Confirm backend is running on the configured port. Check console for CORS errors and ensure backend allows requests from `http://localhost:5173`.

Problem: `npm install` fails with network or permission errors

- Fix: Ensure Node.js and npm are installed. Try clearing npm cache: `npm cache clean --force` and run `npm install` again. If behind a corporate proxy, configure npm proxy settings.

Problem: Changes in the UI do not reflect in the database

- Cause: Backend API endpoint not being called or failing server-side
- Fix: Open the browser developer tools (Network tab) to see the API call and response. Check backend logs for inserted debugging messages.

## **7. Frequently Asked Questions (FAQ)**

Q: Can I run the app without PostgreSQL locally?

A: You need a database to persist exhibits. For temporary testing, you can use a mocked data file if present in the frontend (see `frontend/src/mock/`). However, persistent add/delete operations require PostgreSQL.

Q: Where do I change the admin password?

A: Admin authentication logic is in the backend. If the project uses a simple password prompt, it may be stored or validated server-side. Contact your system administrator or developer to change admin authentication securely.

Q: How do I change the default ports?

A: Update `PORT` in `backend/.env` for backend. For Vite (frontend), set `PORT` in `frontend/package.json` or use Vite environment variables.

## 8. Security and best practices

- Do not commit ` `.env` files with plain credentials to version control.
- Use strong admin passwords and limited-access accounts for production databases.
- Use TLS for any production deployment (HTTPS) and secure Postgres access via network rules or VPN.

## 9. Appendix — Helpful commands

**PowerShell snippets for common tasks (copy/paste):**

*Install backend dependencies and run:*

```
```powershell
cd /d D:\2yp_project_final_codes_DiniraTuesday\heatmap-project\backend
npm install
npm run dev
````
```

*Install frontend dependencies and run Vite:*

```
```powershell
cd /d D:\2yp_project_final_codes_DiniraTuesday\heatmap-project\frontend
npm install
npm run dev
````
```

*Run SQL script to create tables (psql):*

```
```powershell
psql -U heatmap_user -d heatmap_db -f
"D:\2yp_project_final_codes_DiniraTuesday\heatmap-
project\backend\sql\create_zone_info.sql"
````
```

*Backup database with ` pg\_dump`:*

```
```powershell
pg_dump -U heatmap_user -h localhost -p 5432 -F c -b -v -f
"heatmap_db_backup.backup" heatmap_db
````
```

## 10. Contact and support

If you need help beyond this manual, provide the following to your developer or support team:

- Which OS and Node.js version you used (run `node -v` and `npm -v`)
- Exact error messages or screenshots
- Steps you followed and the point of failure

---

## ## Revision history

- v1.0 (2025-11-19) — Initial manual created and added to repository. Template structure referenced GitHub and Microsoft manual patterns; content is original to this project.

End of manual.