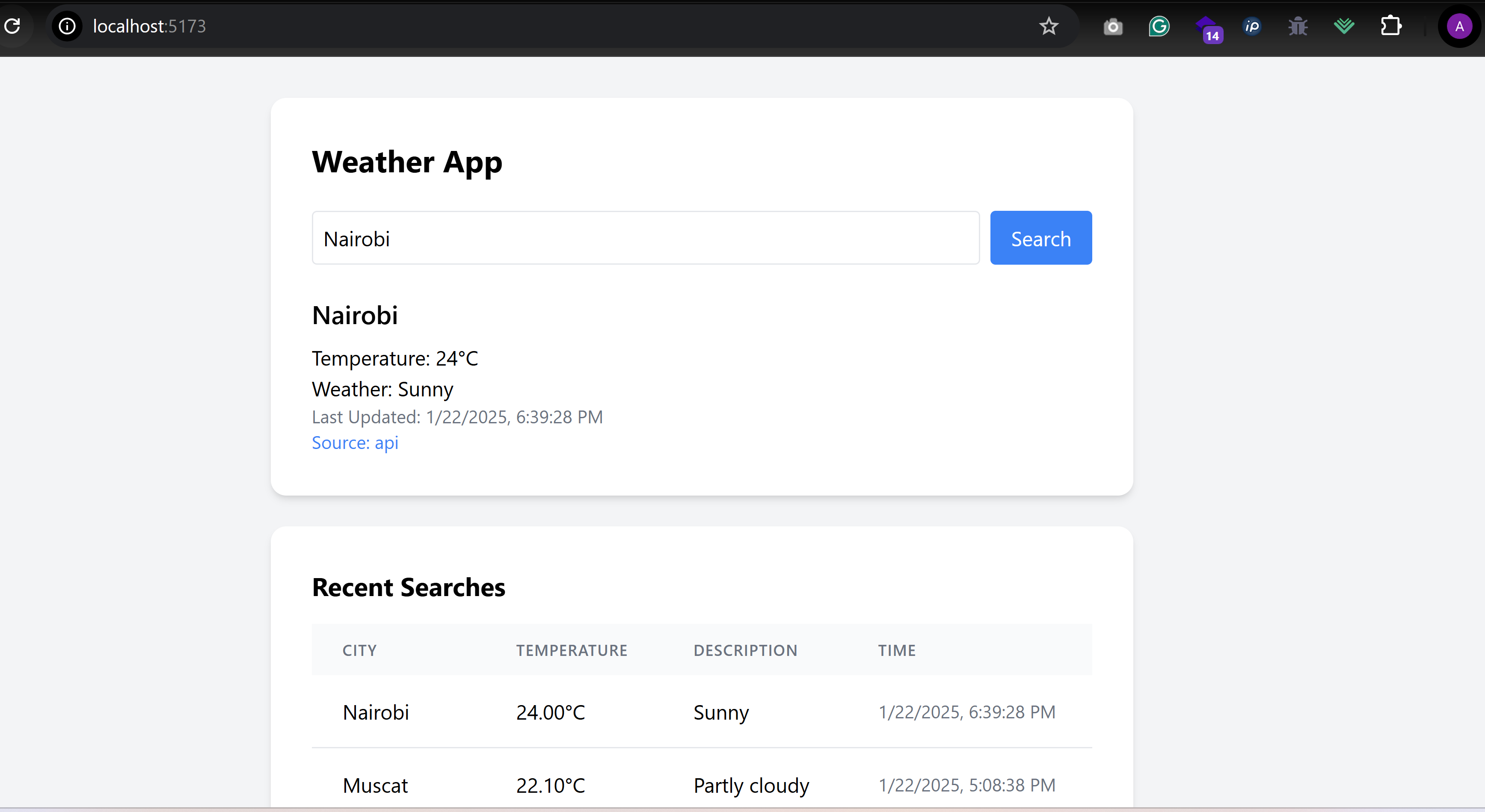
1. **Introduction: Weather Application**

A web application that fetches and displays weather information. The application uses WeatherAPI.com to get current weather data, implements caching to minimize API calls, and stores search history in a database.

1. **Key Features**
   1. Frontend:
      1. **Real-time weather data retrieval:** Instantly fetches current weather information for a searched city from WeatherAPI.com when a user submits a request.
      2. **Data caching (5 minutes):** Stores weather data for 5 minutes to reduce unnecessary API calls improving app performance and minimizing external API usage. This shown for the user.
      3. **Search history tracking:** Automatically saves and displays the last 10 city weather searches in a table allowing users to quickly revisit recent locations.
      4. **Form validation:** Ensures the city name input meets minimum requirements (at least 2 characters long) before submitting the search.
      5. **Error handling:** Manages and displays user-friendly error messages for scenarios like invalid input, API failures, or network issues.



Screenshot of the frontend.

* 1. **Backend:**
     1. **Handles weather data retrieval:** Fetches weather information from WeatherAPI.com based on user-submitted city names.
     2. **Implements caching mechanism:** Stores weather data for 5 minutes to reduce API calls and improve application performance.
     3. **Manages database interactions:** Saves and retrieves weather search history, creating and querying database records for each weather search.

1. **Tech Stack**

|  |  |
| --- | --- |
| **Use** | **Technology** |
| Backend | Laravel |
| Frontend | React with Vite + Tailwin CSS |
| Database | MySQL |
| API | weatherAPI.com |

1. **How to setup?**
   1. **Frontend.**

**Step 1: Clone the repository**

* git clone <your-repository-url>

**Step 2: Navigate to project directory**

* cd weather-app-frontend

**Step 2: Install dependencies**

* npm install

**Step 5: Start Development**

* Run “npm run dev”
  1. **Backend**

**Step 1: Clone Repository**

* Git clone <url>

**Step 2: Navigate to backend directory**

* Run “ cd weather-app”

**Step 3: Install Dependencies**

* Run “composer install”

Or setup is downloadable via <https://getcomposer.org/download/>

**Step 4: Environment Setup**

* Use Existing .env. File is Up to date.

In the case API Key is not functional, generate one from weatherAPI.com.

**Step 5: Generate application key**

* Run “php artisan key:generate”

**Step 6: Run DB Mitigation**

* Run “php artisan mitigate”

**Step 7: Start backend server**

* Run “php artisan serve”
  1. **DB Setup**

Run “php artisan migrate”