**Approach Overview**

**1. API Key Security**

To secure the OpenWeatherMap API key, the following steps were implemented:

* **Environment Variables**: The API key is stored in the .env file of the Laravel backend. This method prevents the key from being hard-coded in the source code, reducing the risk of exposure.

plaintext

Copy

OPENWEATHER\_API\_KEY=your\_api\_key\_here

* **.gitignore**: The .env file is included in the .gitignore file, ensuring it is not committed to version control. This helps keep the API key private and secure from public repositories.

**2. Error Handling**

Robust error handling mechanisms were integrated to enhance user experience and application reliability:

* **Backend Error Handling**:
  + In the Laravel controller, try-catch blocks are employed when making requests to the OpenWeatherMap API. If an error occurs (e.g., invalid city name or network issue), a meaningful error message is returned to the frontend.

php

Copy

try {

$response = Http::get("https://api.openweathermap.org/data/2.5/weather", [

'q' => $city,

'appid' => env('OPENWEATHER\_API\_KEY'),

'units' => 'metric',

]);

if ($response->failed()) {

return response()->json(['error' => 'City not found'], 404);

}

} catch (\Exception $e) {

return response()->json(['error' => 'An error occurred while fetching data.'], 500);

}

* **Frontend Error Handling**:
  + In the Vue.js component, error handling is implemented using try-catch blocks within the fetchWeather method. If an API request fails, the error is logged, and a user-friendly message can be displayed to the user.

javascript

Copy

async fetchWeather() {

try {

const response = await axios.get(`http://localhost:8000/api/weather/${this.city}`);

this.weather = response.data;

} catch (error) {

console.error("Error fetching weather data:", error);

alert("Could not retrieve weather data. Please check the city name.");

}

}

**Summary**

This approach effectively secures the API key and ensures that users receive clear feedback in case of errors. By implementing these practices, the application remains robust, user-friendly, and secure.