# **Setting up the Front End**

This guide helps you set up the front end of the wv-temperature-map project (demo1 branch) on your local machine. You’ll learn how to:

* Clone the repository and open it in VS Code.
* Understand the project structure under frontend/src.
* Install dependencies and run the development server.
* Configure the connection to your backend (login/registration functionality).
* Troubleshoot common issues.

## **1. Prerequisites**

Before you begin, make sure you have the following installed:

* **Node.js and npm:** Download and install from [nodejs.org](https://nodejs.org/).
* **Git:** Download and install Git from [git-scm.com](https://git-scm.com/).
* **VS Code:** Download Visual Studio Code from [code.visualstudio.com](https://code.visualstudio.com/).

## **2. Cloning the Repository (skip if already done!)**

1. **Open VS Code.**
2. **Open the Command Palette:** Press Ctrl+Shift+P

**Clone the Repo:** Type Git: Clone and press Enter.  
 When prompted, paste the repository URL for your project.  
  
**Open the Front End Folder:** Navigate to the frontend folder in the repository and open it in VS Code.

## **3. Understanding the Project Structure**

Inside the frontend/src folder, you’ll find a structure similar to the one below:

frontend/

└── src/

├── components/ // Reusable UI components (buttons, form inputs, etc.)

├── pages/ // Pages for different routes (e.g., Login, Register)

├── services/ // API service modules (e.g., authService.js)

├── App.js // Main application component with routing

├── index.js // Entry point of the application

└── styles/ // (Optional) CSS or styled components

**Key Points:**

* **pages:** Contains the login and registration pages.
* **services:** Includes modules that handle API calls to your backend.
* **App.js:** Sets up routes (using React Router) to navigate between pages.

## **4. Installing Dependencies**

**Open the Integrated Terminal:** In VS Code, go to **View > Terminal** or press Ctrl+`

**Install Packages:** Make sure you are in the frontend directory, then run:  
npm install

This will install all dependencies specified in the package.json file.

## **5. Configuring the Connection to the Backend**

Your front end uses a service module (for example, authService.js) to connect with the backend API. To ensure it targets the correct backend URL:

**Environment Variables:** In the front end, create (or update) a .env file at the root of the frontend folder with the following variable:  
  
REACT\_APP\_API\_URL=http://localhost:3000

(Adjust the URL and port if your backend runs on a different port or host.)

**Using the Environment Variable:** In your service module (e.g., src/services/authService.js), the API URL is read as:  
  
const API\_URL = process.env.REACT\_APP\_API\_URL || 'http://localhost:3000';

This value is used when making API calls (for example, POST requests for login and registration).

## **6. Running the Application**

**Start the React App:** In the terminal (make sure you’re inside the frontend directory), run:  
  
npm start

1. **Access the App in Your Browser:** The development server will start (by default on<http://localhost:3000> or<http://localhost:3001> if port 3000 is used by the backend).  
    If needed, check your package.json or terminal output for the correct URL.
2. **Navigate Between Pages:** Use the navigation links (or enter the URL paths directly) to visit the login page (/login) and registration page (/register). Test the forms by entering sample data.

## **7. Testing the Front End**

* **Check Network Requests:** Open your browser’s developer tools and inspect network requests to confirm that the front end is successfully sending requests to your backend.
* **Error Handling:** If you encounter issues (such as CORS errors), verify that your backend is configured to allow requests from your front end’s origin.
* **Iterative Development:** As you make changes to the UI or the API calls, the development server (with hot reloading) will automatically update, letting you see changes in real time.

## **8. Troubleshooting Common Issues**

* **404 Errors for API Calls:** Ensure that the REACT\_APP\_API\_URL variable is set correctly and that your backend server is running.
* **CORS Issues:** Make sure your backend includes CORS headers (for example, using the cors middleware in Express) so that your front end can communicate with it.
* **Port Conflicts:** Verify that the front end and backend are running on different ports. If they conflict, adjust your configurations accordingly.

## **Conclusion**

By following this guide, you have:

* Cloned the wv-temperature-map repository and switched to the demo1 branch.
* Installed all necessary dependencies.
* Understood the project structure and configured environment variables.
* Run the React development server and connected the front end to the backend.

This setup gives you a solid foundation to further develop and test the login and registration features. Happy coding and enjoy building your application!