

ElectroSense: React + Vite + Firebase Monorepo Guide

Welcome to your new React-based shop! This guide will walk you through building and deploying two separate applications from one project folder:

1. `shop-frontend` : The customer-facing shop (React + Vite)
2. `shop-admin` : The admin dashboard (React + Vite)

We will deploy them to `shop.sandunsiwantha.com` and `admin.sandunsiwantha.com` using one Firebase project.

Part 1: Firebase Project Setup

1. **Create ONE Firebase Project:** Go to the [Firebase Console](#) and create one new project (e.g., `electrosense-shop`).
2. **Enable Services:**
 - **Authentication:** Go to "Sign-in method" -> Enable **Email/Password** and **Google**.
 - **Firestore:** Create a database. Start in **Test Mode** for now.
3. **Create Firestore Collections:**
 - Go to your Firestore Database.
 - Create a collection named `products` .
 - Add your first product document. Click "Auto-ID" and add these fields:
 - `name` (string) - e.g., "Arduino Uno"
 - `price` (number) - e.g., 3500
 - `stockQuantity` (number) - e.g., 50
 - `imageUrl` (string) - e.g., "https://www.google.com/search?q=https://placeholder.co/400x400/0d9488/white%3Ftext%3DArduino"
 - `category` (string) - e.g., "Microcontrollers"
 - `description` (string) - "The classic Arduino board."
 - Create a collection named `users` . (Authentication will use this).
 - Create a collection named `orders` . (The app will add to this).
4. **Set Up Multi-Site Hosting:**
 - Go to the **Hosting** page in your Firebase project.
 - Click "Get Started".
 - Click **"Add custom domain"**. Follow the steps to add and verify `shop.sandunsiwantha.com` .

- Click "Add custom domain" **AGAIN**. Follow the steps to add and verify `admin.sandunsiwantha.com`.
- You will now have two sites listed in your Hosting dashboard. This is perfect.

Part 2: Local Project Structure (The Monorepo)

In VS Code, you will have one main folder, and inside it, your two React apps.

1. Create your main project folder:

```
mkdir electrosense-monorepo
cd electrosense-monorepo
```

2. This `electrosense-monorepo` folder is now your project root. All `firebase` commands will be run from here.

Part 3: Create the Customer Shop (`shop-frontend`)

1. **Create the Vite App:** While inside `electrosense-monorepo`, run:

```
npm create vite@latest shop-frontend -- --template react-swc
```

2. **Install Dependencies:**

```
cd shop-frontend
npm install
npm install firebase react-router-dom zustand tailwindcss postcss autoprefixer @heroicons/react
cd ..
```

- `react-router-dom` : For multi-page navigation.
- `zustand` : For simple and powerful cart state management.
- `tailwindcss` : For styling.
- `@heroicons/react` : For icons.

3. **Configure Tailwind:**

- Run this command inside the `shop-frontend` folder:

```
cd shop-frontend
npx tailwindcss init -p
```

- This creates `tailwind.config.js` and `postcss.config.js`.
- **Action:** Copy the contents of `shop-frontend-tailwind.config.js` (which I provided) and paste it into your new `tailwind.config.js`.
- **Action:** Copy `shop-frontend-index.css` (which I provided) and paste its contents into `src/index.css` (overwrite the existing content).

4. Add Your React Code:

- **Action:** Create a new file: `shop-frontend/src/firebaseConfig.js`. Copy the code from my `firebaseConfig.js` file into it. **Remember to add your Firebase SDK keys!**
- **Action:** Create a new file: `shop-frontend/src/store.js`. Copy the code from my `shop-store.js` file into it. This is your cart logic.
- **Action:** Open `shop-frontend/src/App.jsx`. Delete everything in it. Copy the *entire contents* of my `shop-app.jsx` file and paste it in.
- **Action:** Open `shop-frontend/src/main.jsx`. Replace its contents with this to enable the router:

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.jsx'
import './index.css'
import { BrowserRouter } from 'react-router-dom'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>,
)
```

5. Run the Shop:

```
cd shop-frontend
npm run dev
```

Your customer shop is now running on `http://localhost:5173` !

Part 4: Create the Admin Dashboard (`shop-admin`)

1. **Create the Vite App:** Go back to your root folder `electrosense-monorepo`.

```
cd ..
npm create vite@latest shop-admin -- --template react-swc
```

2. Install Dependencies:

```
cd shop-admin
npm install
npm install firebase react-router-dom tailwindcss postcss autoprefixer @heroicons/react
cd ..
```

3. Configure Tailwind:

- Run this inside the `shop-admin` folder:

```
cd shop-admin
npx tailwindcss init -p
```

- **Action:** Copy `shop-admin-tailwind.config.js` into your new `tailwind.config.js`.
- **Action:** Copy `shop-admin-index.css` into `src/index.css` (overwrite).

4. Add Your React Code:

- **Action:** Create `shop-admin/src/firebaseConfig.js`. You can copy the *exact same file* from `shop-frontend/src/firebaseConfig.js` (it works for both).
- **Action:** Open `shop-admin/src/App.jsx`. Delete everything and paste in the contents of my `admin-app.jsx` file.
- **Action:** Open `shop-admin/src/main.jsx`. Replace its contents (same as the shop's `main.jsx`):

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.jsx'
import './index.css'
import { BrowserRouter } from 'react-router-dom'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>,
)
```

5. Run the Admin Panel:

```
cd shop-admin  
npm run dev
```

Your admin panel is now running (likely on `http://localhost:5174`)!

Part 5: How to Split `App.jsx` into Separate Files

My `shop-app.jsx` and `admin-app.jsx` files contain *all* components. This is for delivery. You should split them up.

Example for `shop-frontend` :

1. Create a folder `shop-frontend/src/pages` .
2. Create a folder `shop-frontend/src/components` .
3. Go into `shop-app.jsx` . Cut the *entire* `HomePage` component (from `function HomePage() { ...` } to the end).
4. Create a new file `shop-frontend/src/pages/HomePage.jsx` .
5. Paste the code. At the top, add `import React from 'react';` and any other imports it needs (like `Link` or `useCart`). At the bottom, add `export default HomePage;` .
6. In `shop-app.jsx` , at the top, add `import HomePage from './pages/HomePage';` .
7. Repeat this for `ProductsPage` , `CartPage` , `Navbar` , `Footer` , etc.
8. Your `App.jsx` will become very clean, containing only the main router.

Part 6: Deployment

1. **Initialize Firebase:** Go to your root `electrosense-monorepo` folder.

```
cd ..  
firebase login  
firebase init hosting
```

2. **Answer the CLI Questions:**

- **Project:** Select your `electrosense-shop` project.
- **Public Directory:** This doesn't matter, we will overwrite it. Type `shop-frontend/dist` .
- **Configure as SPA?:** Yes.
- **Set up GitHub?:** No (for now).

3. **Configure `firebase.json` :**

- **Action:** A `firebase.json` file now exists in your root folder.

- Delete its contents and **paste in the code from my `firebase.json` file**. This is the multi-site config.

4. **Set Deploy Targets:** This links your local folders to your live domains.

- **Run this ONCE:**

```
firebase target:apply hosting shop shop.sandunsiwantha.com
```

- **Run this ONCE:**

```
firebase target:apply hosting admin admin.sandunsiwantha.com
```

5. **Deploy Your Website!:**

- **Step 1. Build the Customer App:**

```
cd shop-frontend  
npm run build  
cd ..
```

- **Step 2. Build the Admin App:**

```
cd shop-admin  
npm run build  
cd ..
```

- **Step 3. Deploy Both!:**

```
firebase deploy
```