

Shopping List Web App - Project Documentation

A full-stack web-based shopping list app built with:

- **FastAPI** (backend server with Jinja2 templating)
- **TailwindCSS CLI** (mobile-first UI styling)
- **Firebase Auth & Firestore** (authentication and cloud data storage)
- **Jinja2** for server-side HTML templates
- **JavaScript** for frontend interactivity

Project Structure

```
shopping-list/
├── app/
│   ├── main.py           # FastAPI entry point
│   ├── services/
│   │   └── auth_service.py # Handles Firebase auth and user saving
│   ├── templates/
│   │   ├── base.html
│   │   ├── auth/
│   │   │   └── login.html
│   │   └── dashboard/
│   │       └── index.html
│   └── static/
│       ├── js/
│       │   └── app.js      # Firebase integration and UI handling
│       └── css/
│           └── output.css  # Tailwind CSS compiled file
├── .env                  # Environment variables
├── tailwind.config.js    # Tailwind config
└── src/
    └── input.css          # Tailwind input file
```

1. Backend Setup (FastAPI + Jinja + Firebase Admin SDK)

1.1 Create a virtual environment

```
python -m venv env
source env/bin/activate # or env\Scripts\activate on Windows
```

1.2 Install dependencies

```
pip install fastapi uvicorn jinja2 python-dotenv firebase-admin google-cloud-firestore
```

1.3 Set up `.env`

```
# Firebase Admin
FIREBASE_PROJECT_ID=shopping-list-e9cc7
FIREBASE_PRIVATE_KEY_ID=...
FIREBASE_PRIVATE_KEY="-----BEGIN PRIVATE KEY-----\n...\n-----END PRIVATE KEY-----\n"
FIREBASE_CLIENT_EMAIL=firebase-adminsdk@shopping-list-e9cc7.iam.gserviceaccount.com
FIREBASE_CLIENT_ID=...

# Firebase Web SDK
FIREBASE_API_KEY=...
FIREBASE_AUTH_DOMAIN=shopping-list-e9cc7.firebaseio.com
```

1.4 Firebase Admin Initialization (in `auth_service.py`)

- Load credentials from `.env`
- Initialize Firebase Admin SDK

1.5 User Authentication Route (`/auth`)

- Accepts a Firebase ID token
- Verifies it and saves the user (if new) to Firestore under `users` collection

2. Frontend Setup (TailwindCSS + Firebase Auth)

2.1 Install Tailwind CLI (one-time)

```
npm install tailwindcss @tailwindcss/cli
```

2.2 Tailwind CSS Build Script

```
npx tailwindcss -i ./src/input.css -o ./app/static/css/output.css --watch
```

2.3 Basic HTML Structure

Use `base.html` to define reusable layout.

2.4 Firebase Web SDK Setup in `app/static/js/app.js`

- Handles login/logout with Google
- Verifies user session via Firebase
- Fetches user profile
- Displays user info and their shopping lists

2.5 Dashboard Logic

- Displays list items from `/api/lists`
 - Includes logout and redirect logic
-

3. Firebase Setup

3.1 Enable Firebase Auth

- Go to [Firebase Console](#)
- Choose your project
- Enable **Authentication > Sign-in method > Google**
- Add `http://127.0.0.1:8000` to **Authorized domains**

3.2 Enable Firestore

- Go to **Firestore Database**
- Click **Create Database** (select test mode or locked mode)
- Your app will store users and future lists here

3.3 Enable Cloud Firestore API in Google Cloud

- Go to the [Google Cloud Console](#)
- Navigate to **APIs & Services > Library**
- Search for **Cloud Firestore API** and enable it

Why?: This is required so that the FastAPI backend (using the Firebase Admin SDK) can read/write data to Firestore. This API must be enabled by the **developer**, not by users.

4. Current Features

- Google Sign-in (Firebase Auth)
- Authenticated user storage (Firestore > `users` collection)
- Tailwind CSS layout (mobile-first)

- View sample shopping lists (static for now)
 - Dashboard with user photo, name, and email
-

Planned Features

- Add/edit/delete shopping lists
 - Share lists with other users
 - Add list items with checkboxes and categories
 - Real-time sync using Firestore snapshot listeners
 - Offline support with local storage fallback
 - PWA support for mobile install
-

To Run the App

```
# Activate venv
source env/bin/activate

# Compile Tailwind CSS (keep running)
npx tailwindcss -i ./src/input.css -o ./app/static/css/output.css --watch

# Start FastAPI server
uvicorn app.main:app --reload
```

Open: <http://127.0.0.1:8000>

Author

Built by John Paul Quintana for public use and collaboration.

Let me know if you want to add deployment instructions (e.g. Vercel, Firebase Hosting, or Render).