Crowdsourced Disaster Relief Platform User Documentation

Course: CS 3354 Spring 2025 Group Number: 2 Group Members: Casey Nguyen, Kevin Pulikkottil, Andy Jih, Sawyer

**Table of Contents:**

1. Introduction
2. Installation Guide (Backend)
3. Flutter Integration Guide
4. Testing the Backend
5. Troubleshooting
6. Introduction

This user documentation provides step-by-step guidance for setting up and interacting with the Crowdsourced Disaster Relief Platform backend and integrating it with a Flutter frontend.

1. Installation Guide (Backend)

Prerequisites:

* Python 3.9 or newer
* PostgreSQL installed
* Docker (optional, for deployment)

Backend Setup:

1. Install Python packages:

pip install fastapi uvicorn sqlalchemy psycopg2-binary scikit-learn numpy

1. Create PostgreSQL database:

CREATE DATABASE disaster\_relief;

CREATE USER postgres WITH PASSWORD 'password';

ALTER DATABASE disaster\_relief OWNER TO postgres;

1. Launch backend API:

uvicorn main:app --reload

Backend API is accessible at http://localhost:8000.

1. Flutter Integration Guide

Example Flutter HTTP call:

import 'package:http/http.dart' as http;

import 'dart:convert';

Future<void> fetchMatchedVolunteers(int requestId) async {

final response = await http.get(

Uri.parse('http://localhost:8000/match/$requestId'),

);

if (response.statusCode == 200) {

final data = json.decode(response.body);

print(data); // handle/display data in Flutter UI

} else {

print('Request failed: ${response.statusCode}');

}

}

Ensure your Flutter app is set up to handle JSON responses and display the returned data appropriately.

1. Testing the Backend

Running Basic Tests:

1. Start backend API:

uvicorn main:app --reload

1. Run provided testing scripts:

python test\_matching.py

Expected results clearly indicate success or failure for each test case.

1. Troubleshooting

Common Issues:

* **Database connection failed**: Ensure PostgreSQL service is running and credentials match the DATABASE\_URL.
* **API inaccessible**: Verify the backend service is running with Uvicorn.
* **CORS errors**: Verify CORS middleware is correctly enabled in main.py.