

***Ghaith   
Mobile Application***

Project Team

Gamal Barakat Gamal

1

Ebrahem Mohamed

2

Ahmed mostafa

3

4

**Engineer Esraa Fathy**

Supervised by

Youssef Mohamed

## Table of Contents:

1. [Introduction](#introduction)
2. [Project Overview](#project-overview)
3. [Features](#features)
4. [System Requirements](#system-requirements)
5. [Technical Architecture](#technical-architecture)
6. [User Interface (UI)](#user-interface-ui)
7. [Donation Categories](#donation-categories)
8. [Volunteering Campaigns](#volunteering-campaigns)
9. [Payment Integration](#payment-integration)
10. [Development Environment Setup](#development-environment-setup)
11. [API Documentation](#api-documentation)
12. [Conclusion](#conclusion)

## 1. Introduction

**Ghaith** is an application designed to streamline and facilitate donations for those in need, especially during times of natural disasters. Unlike traditional donation platforms, Ghaith allows donors to contribute essential items like clothes, food, and medicine, not just money. It aims to make charitable work more accessible and tailored to actual needs.

## 2. Project Overview

* **Project Name**: Ghaith
* **Objective**:
  + Facilitate donations to help people during natural disasters.
  + Provide essential items rather than just monetary aid.
  + Enhance the work of charities and speed up the response during crises.
* **General Goal**: To create a user-friendly platform where donations can be made in various forms, and volunteer efforts can be organized and directed towards the most pressing needs.

## 3. Features

1. **User-Friendly Interface**: Simple and intuitive design for easy navigation.
2. **Multiple Donation Options**: Donate items like clothes, food, medicine, etc.
3. **Cash or Online Payment**: Secure payment gateways offering cash and online payments.
4. **Volunteer Opportunities**: Facilitate volunteering through various campaigns.
5. **Flexible Donations**: Donate to any category without being tied to specific causes.
6. **Disaster Response Support**: Accelerates donations and responses during crises.

## 4. System Requirements

* **Client-Side Requirements**:
  + Android 7.0 or later
  + iOS 11.0 or later
  + Stable internet connection
* **Server-Side Requirements**:
  + Node.js 14.x or later
  + MongoDB 4.x or later
  + Flutter SDK for mobile app development
  + Payment gateway integration (e.g., PayPal, Stripe)

## 5. Technical Architecture

Ghaith's architecture consists of three primary layers:

1. **Frontend**: Built using Flutter, providing cross-platform support for Android and iOS.
2. **Backend**: Node.js-based server managing user authentication, donations, payments, and volunteering campaigns.
3. **Database**: MongoDB for storing user data, donation records, volunteer info, and campaign details.

## 6. User Interface (UI)

The user interface of the Ghaith application is:

1. **Simple**: Clean, minimalistic design with easy navigation.
2. **Accessible**: Designed for users of all ages, including those less familiar with technology.
3. **Responsive**: Adapts to various screen sizes for both smartphones and tablets.

**UI Components**:

* **Homepage**: Displays current campaigns, donation categories, and recent requests.
* **Donate Page**: Allows users to choose donation categories.
* **Volunteering Page**: Lists active volunteer campaigns with signup options.
* **Payment Page**: Handles secure transactions via cash or online payment methods.

## 7. Donation Categories

Ghaith offers several categories of donations, including:

1. **Clothes**: New or used clothing.
2. **Food Meals**: Non-perishable food items and cooked meals.
3. **Medicine**: Essential medicines and medical supplies.
4. **Other Essentials**: Bedding, hygiene products, etc.

Users can donate to a specific category or contribute to a general fund distributed based on need.

## 8. Volunteering Campaigns

Ghaith provides opportunities for users to volunteer during natural disasters or emergencies. Campaigns are announced within the app, and users can sign up to participate.

### Volunteering Features:

1. **Volunteer Page**:
   * Displays a list of registered volunteers.
   * **Components**:
     + **BlocProvider and BlocBuilder**: Manage data state and update the UI based on data availability.
     + **Container**: Sets a background color for the page.
     + **ConditionalBuilder**: Shows either a list of volunteers or a circular indicator if no data is available.
   * **Functions**:
     + Displays a list of volunteers using ListView.separated with separators.
     + Each volunteer is shown in a CardViewVolunteer.
2. **Volunteer Registration Page**:
   * Allows users to register as a volunteer.
   * **Components**:
     + **TextEditingControllers**: For controlling text input for name, address, age, occupation, and experience.
     + **GlobalKey<FormState>**: Validates form inputs.
     + **Scaffold**: Creates the page structure with an AppBar and main content.
     + **BlocProvider and BlocConsumer**: Manage data state, updating the UI based on data changes.
   * **Functions**:
     + Validates and submits volunteer data.
     + Shows success or failure messages via Toast.

## 9. Payment Integration

Ghaith integrates secure payment gateways to facilitate donations. Payment methods include:

1. **Credit/Debit Card** (via gateways like Stripe or PayPal).
2. **Bank Transfer**.
3. **Cash Payment**.

All transactions are encrypted for donor information protection.

## 10. Development Environment Setup

**Prerequisites**:

* **Flutter SDK**: Install from [flutter.dev](https://flutter.dev).
* **Node.js**: Install version 14.x or later.
* **MongoDB**: Set up a MongoDB database.
* **Payment Gateways**: Integrate Stripe or PayPal.

**Steps to Set Up**:

1. Clone the project repository from GitHub.
2. Set up Flutter by running flutter doctor.
3. Configure the backend with a MongoDB connection.
4. Install backend dependencies with npm install and frontend dependencies with flutter pub get.
5. Configure payment gateway environment variables.
6. Run the backend server with npm start and the Flutter app using flutter run.

## 11. API Documentation

**Authentication**:

* POST /auth/login: Logs in a user and returns a JWT token.
* POST /auth/register: Registers a new user.

**Donations**:

* GET /donations: Fetches available donation categories.
* POST /donations: Submits a new donation.

**Volunteering**:

* GET /volunteer/campaigns: Fetches active volunteer campaigns.
* POST /volunteer/signup: Signs a user up for a campaign.

**Payments**:

* POST /payments/donate: Processes donation payments.

## 12. Conclusion

Ghaith aims to revolutionize donations during times of natural disaster and humanitarian crises. By allowing the donation of essential items rather than just money, Ghaith addresses the real needs of individuals. The platform also fosters community involvement through volunteer opportunitie