UE22CS352B - Object Oriented Analysis & Design

Mini Project Report

CRISIS MANAGEMENT SYSTEM

Submitted by:

Namritha Lasyapriya Maddali: PES1UG22CS374
Nikhil M : PES1UG22CS384
Pradhaan S Bhat : PES1UG22CS420

Sixth Semester G-Section

Faculty--Bhargavi Mokashi

January - May 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

Problem Statement:

The Crisis Management System (CMS) is software that intends to bridge the gap between the affected individuals and relief providers in times of crisis and disasters. It intends to provide a platform so that the actors such as affected individuals, relief providers, government agencies and other authorized personnel to interact seamlessly to resolve all help requests raised by people in crisis as soon as possible to avoid unwanted consequences

Key Features:

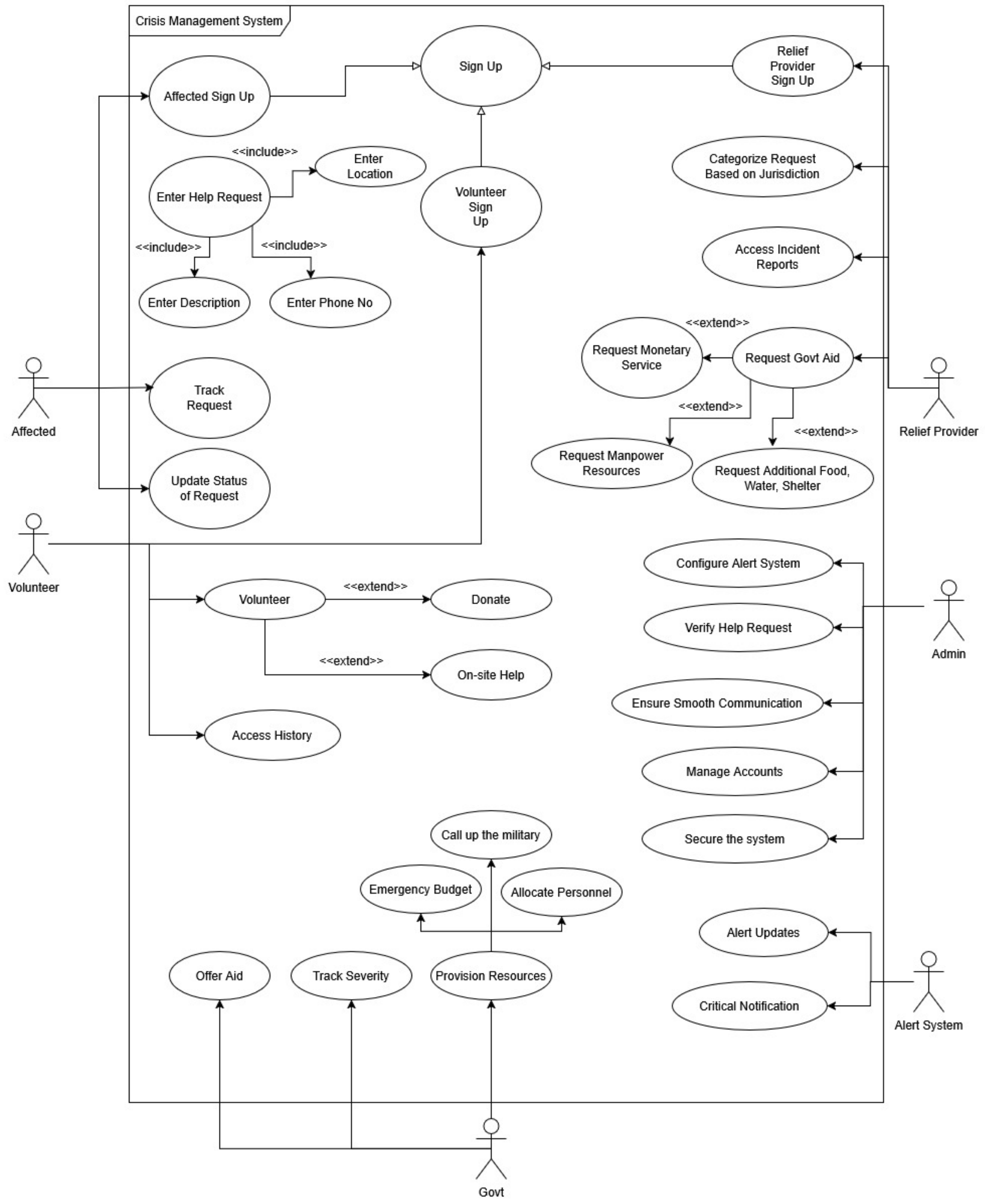
- Help Request Submission
- Track Help Request
- User-specific dashboards
- Real-time tracking and status updates
- Alert System configured to show alerts
- Volunteer coordination
- Resource management
- Relief provider management
- Emergency Response Tracking
- Government Agency funded resource allocation

Architecture:

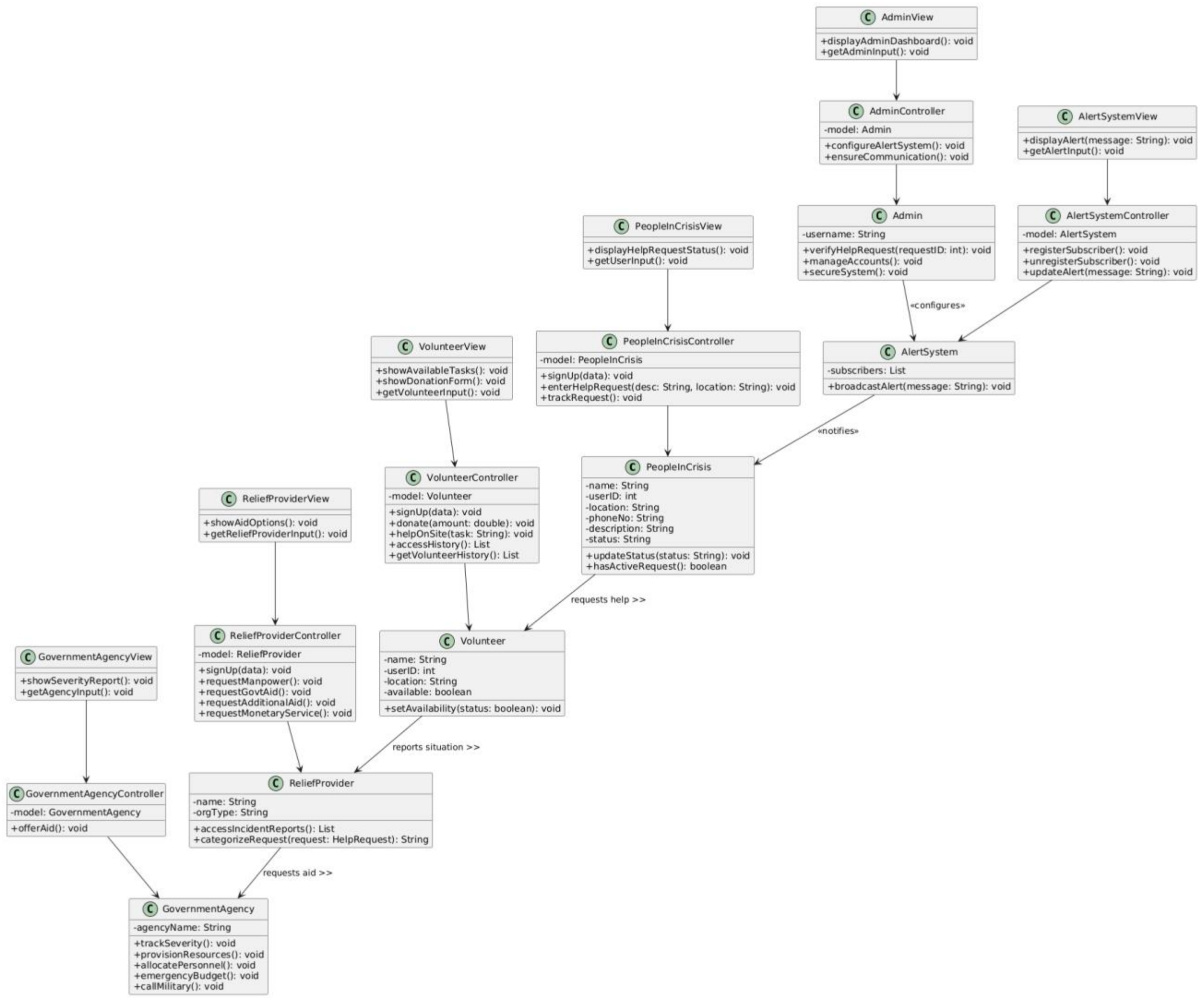
- The system makes use of the MVC architecture which separates the components into model, view and controller. The view part is integrated as frontend in the code and the model and controller are a part of the src directory
- **Frontend** : Streamlit
- **Backend** : POCO + C++
- **Database** : SQLite

Models:

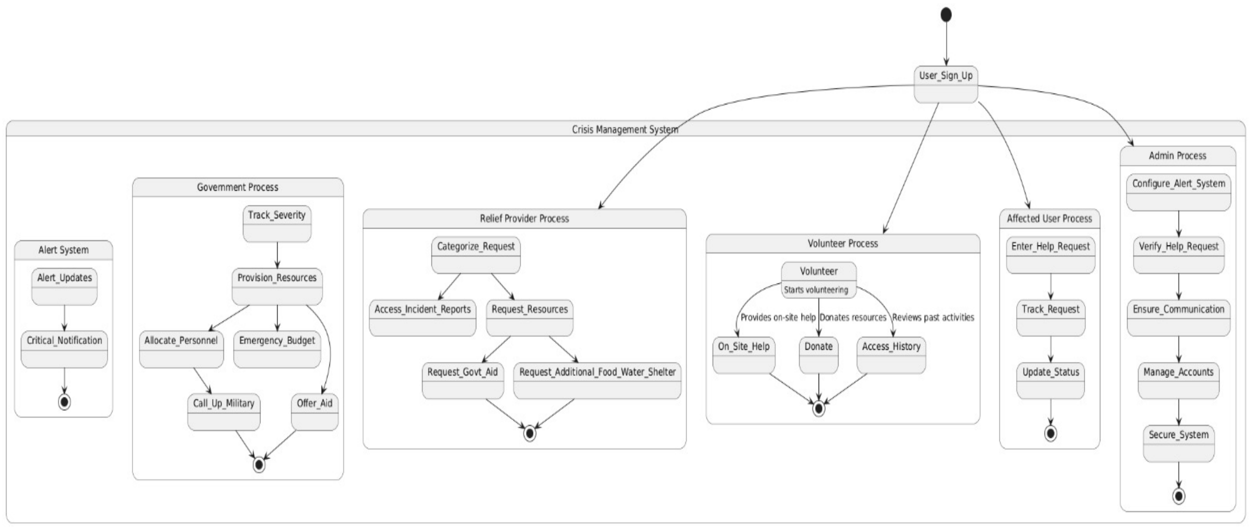
Use Case Diagram:



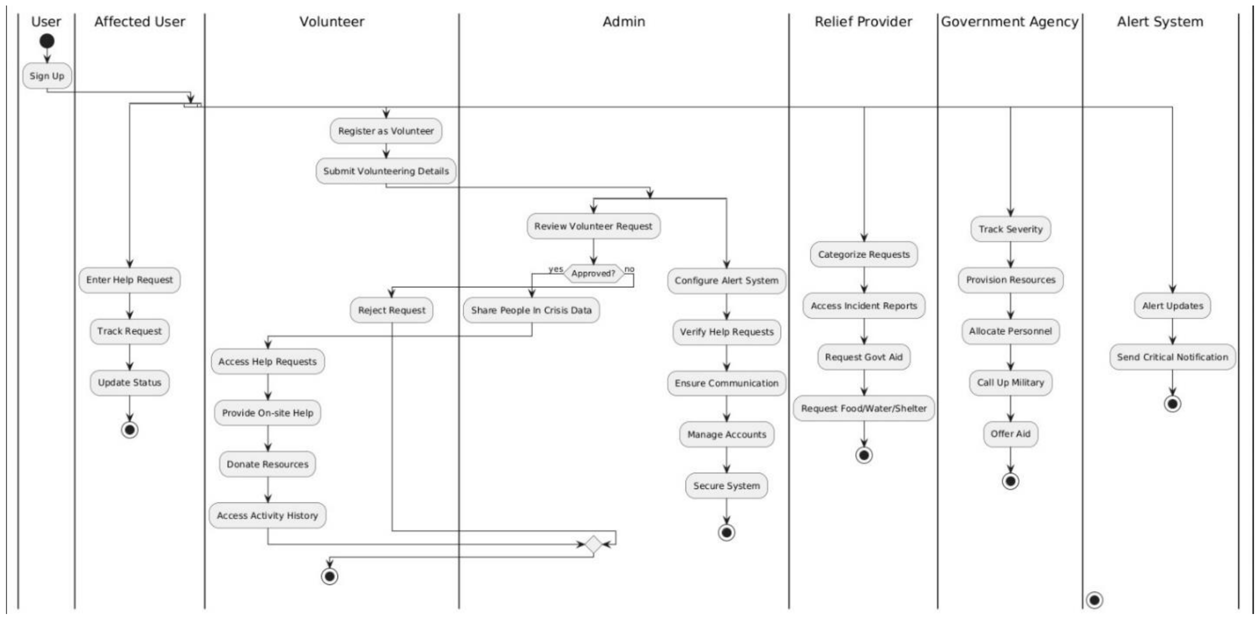
Class Diagram:



State Diagram:



Activity Diagrams:



Architecture Patterns, Design Principles, and Design Patterns:

Architecture Patterns

Model – View – Controller Pattern (MVC)

Models:
- **Admin Model**
- **AlertSystem Model**
- **GovernmentAgency Model**
- **PeopleInCrisis Model**
- **Volunteer Model**
- **Relief Provider Model**
- **Help Request Model**

Controllers:
- **Admin Controller**
- **GovernmentAgency Controller**
- **PeopleInCrisis Controller**
- **Volunteer Controller**
- **Relief Provider Controller**
- **Help Request Controller**
- **PeopleInCrisis Controller**

View:
- **Admin Dashboard**
- **Volunteer Dashboard**
- **GovernmentAgency Dashboard**
- **Relief Provider Dashboard**
- **PeopleInCrisis Dashboard**

Design Principles

GRASP Principles:

a) **<u>Creator :</u>** There are different classes for each entity which is responsible for the creation of respective objects.
b) **<u>Information Expert :</u>** Some overarching classes know information about multiple objects and are responsible for them. Ex : The AlertSystem class knows about subscribers and alert delivery. The Database Manager knows about database operations. The UserDecorator knows about user permissions.
c) **<u>Low Coupling :</u>** The AlertStrategy interface used decouples alert delivery methods. UserDecorator decouples user types from permission logic. HelpRequestCommand decouples request operations from their execution.
d) **<u>High Cohesion :</u>** Each class is solely responsible for its own functionalities and completely owns a use case thus providing high cohesion. Ex : AlertSystem only focuses on alert management. PeopleInCrisis only focuses on people in crisis functionalities to add,track help requests. DatabaseManager only focuses on database operations.
e) **<u>Controller :</u>** Each class has its own respective controller that is responsible for handling the operations and act as an interface between the model and the view.

<u>SOLID Principles</u>

a) **<u>Single Responsibility :</u>** The classes are only responsible for their own functionality and they completely own them. Ex : Volunteer only handles volunteer related functionalities and GovernmentAgency only handles GovernmentAgency functionalities
b) **<u>Open/Closed :</u>** New api handlers can be added through concrete factories. New commands can be added through helprequestcommands.
c) **<u>Liskov Substitution :</u>** All user decorators can be used where base User is expected. All commands can be used where base HelpRequestCommand is expected
d) **<u>Interface Segregation :</u>** Each class has its own clean interface segregated from other classes and implements only its own functionalities
e) **<u>Dependency Inversion :</u>** Wherever possible, abstractions are used so that it depends on the abstractions rather than the implementations

<u>Design Patterns</u>

<u>A) Singleton Pattern</u>

a. **<u>Database manager :</u>** A single instance of the database is used by all operations and the method to access it is made static

b. **AlertSystem :** A single instance of the alert system exists so that it can be used by all other operations

B) Observer Pattern

a. **NotificationSystem:** The alert messages make use of a simple observer pattern in order to maintain loose coupling between the different users and the alert messages and the ability to add and delete users easily

C) Factory Pattern

a. **ApiServer :** All API calls are made with the help of the abstract HTTPRequestFactory which is extended by the different concrete request factories

D) Command Pattern

a. **HelpRequests:** The command pattern is used to encapsulate help request operations as objects. It supports the undo/cancel of requests and a history of requests is maintained. The creation of a command is separated from its execution

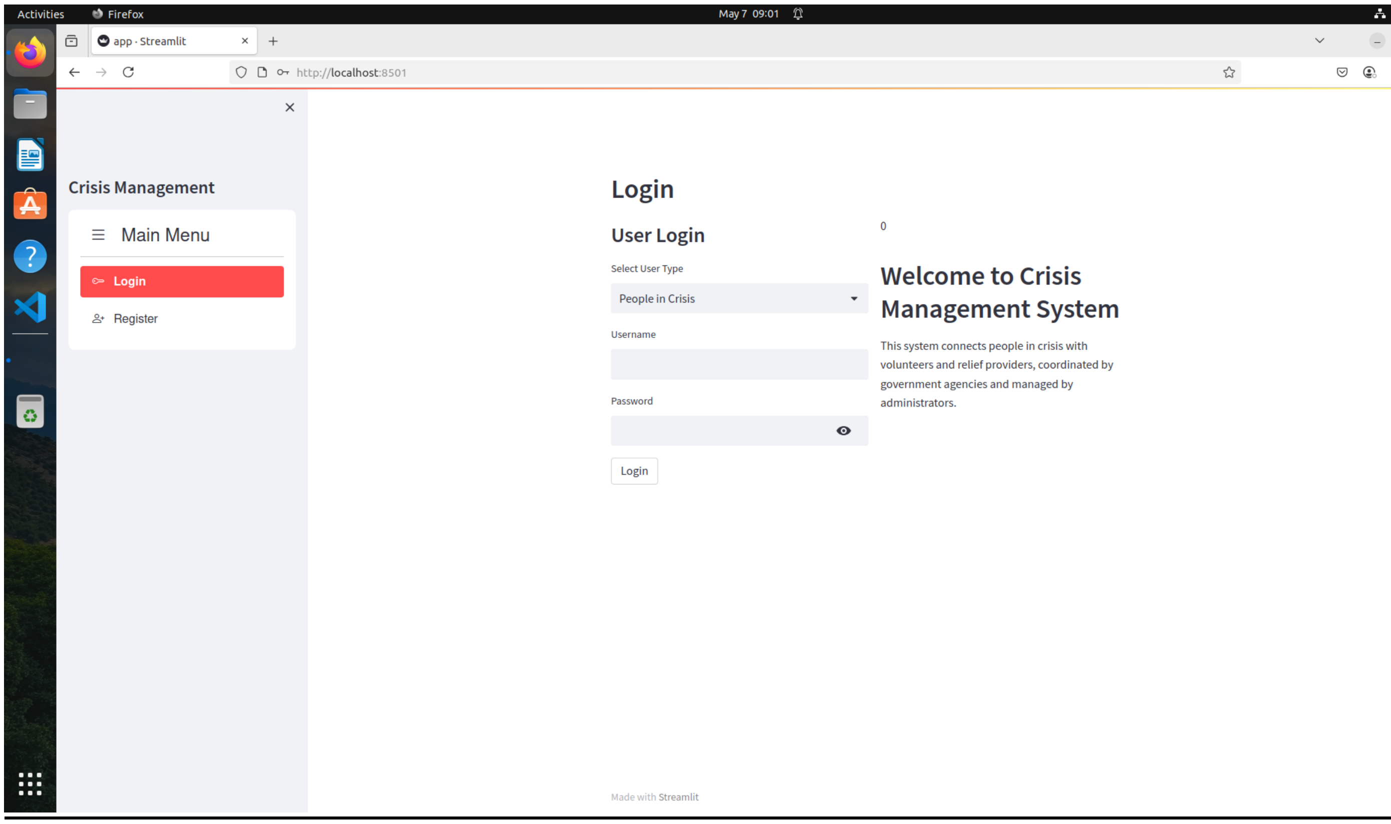
E) Decorator Pattern

a. **UserTypes:** The decorator pattern is used to dynamically add responsibilities to user objects. It implements different permission levels for different user types and allows for flexible combination of user roles and permissions while keeping base class simple for extending functionality.

Github link to the Codebase:
https://github.com/PES1UG22CS420/CMS

Screenshots

UI:



Login

User Login

Select User Type

People in Crisis

Username

Password

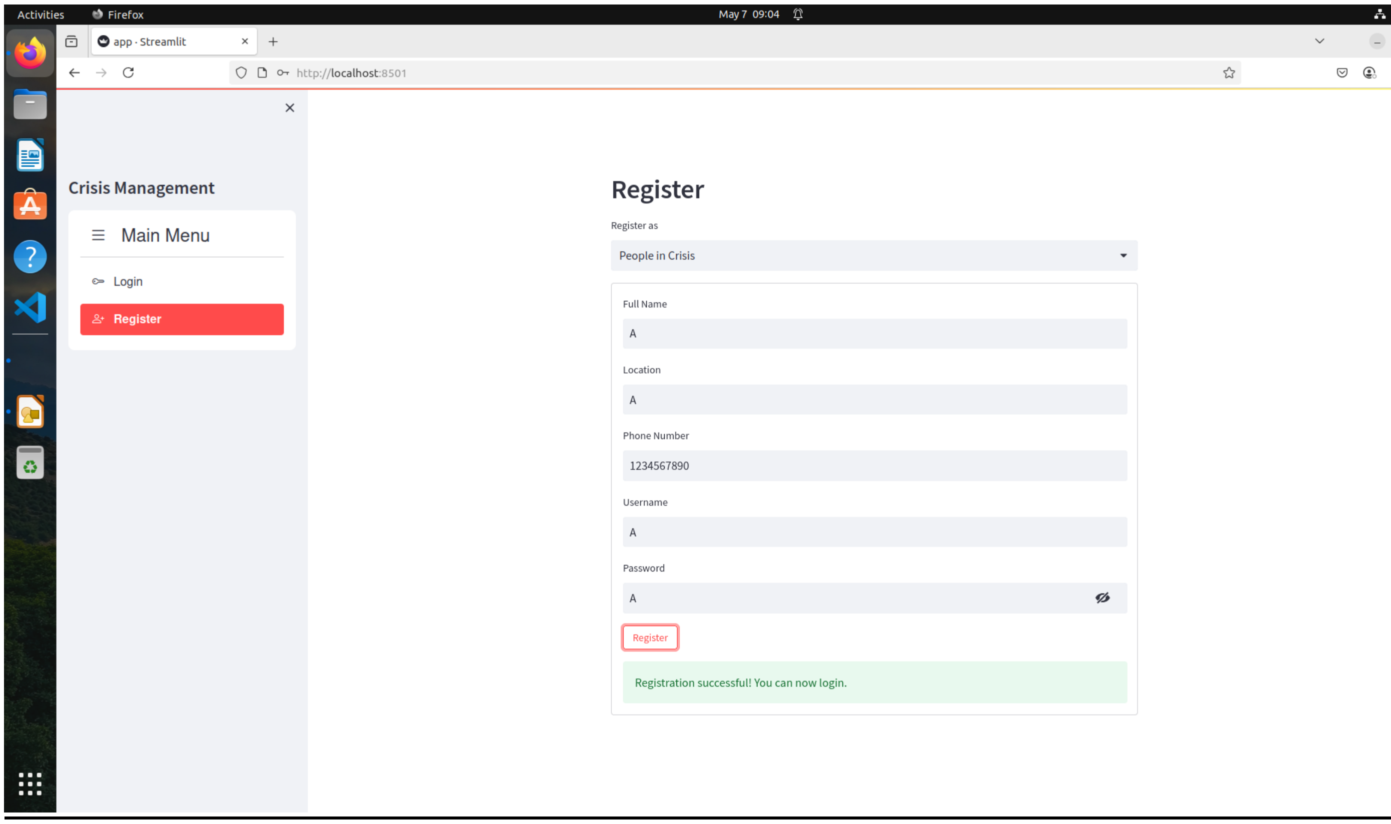
Login

0

Welcome to Crisis Management System

This system connects people in crisis with volunteers and relief providers, coordinated by government agencies and managed by administrators.

Made with Streamlit



Register

Register as: People in Crisis

Full Name: A

Location: A

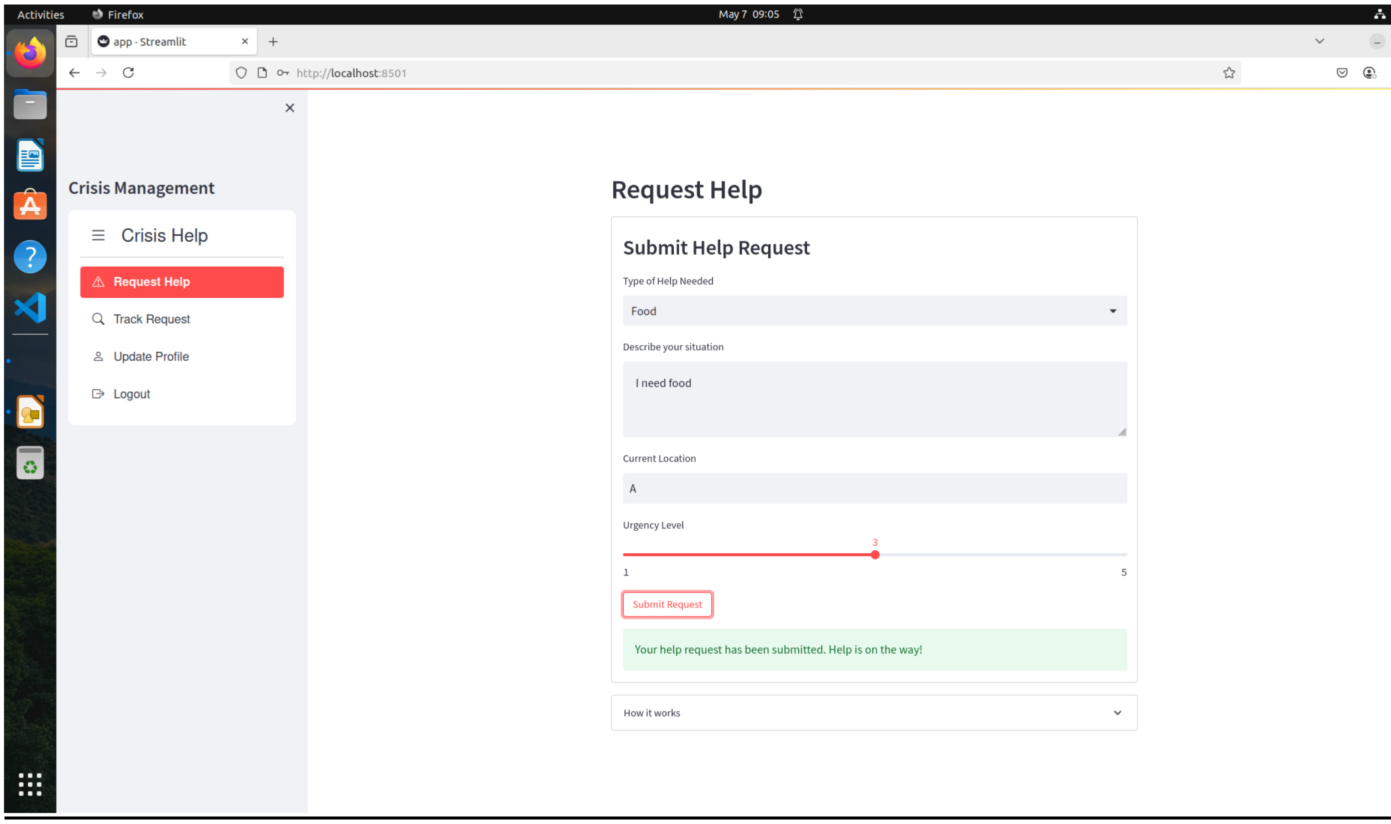
Phone Number: 1234567890

Username: A

Password: A

Register

Registration successful! You can now login.



Crisis Management

Crisis Help

- Request Help
- Track Request
- Update Profile
- Logout

Request Help

Submit Help Request

Type of Help Needed

Food

Describe your situation

I need food

Current Location

A

Urgency Level

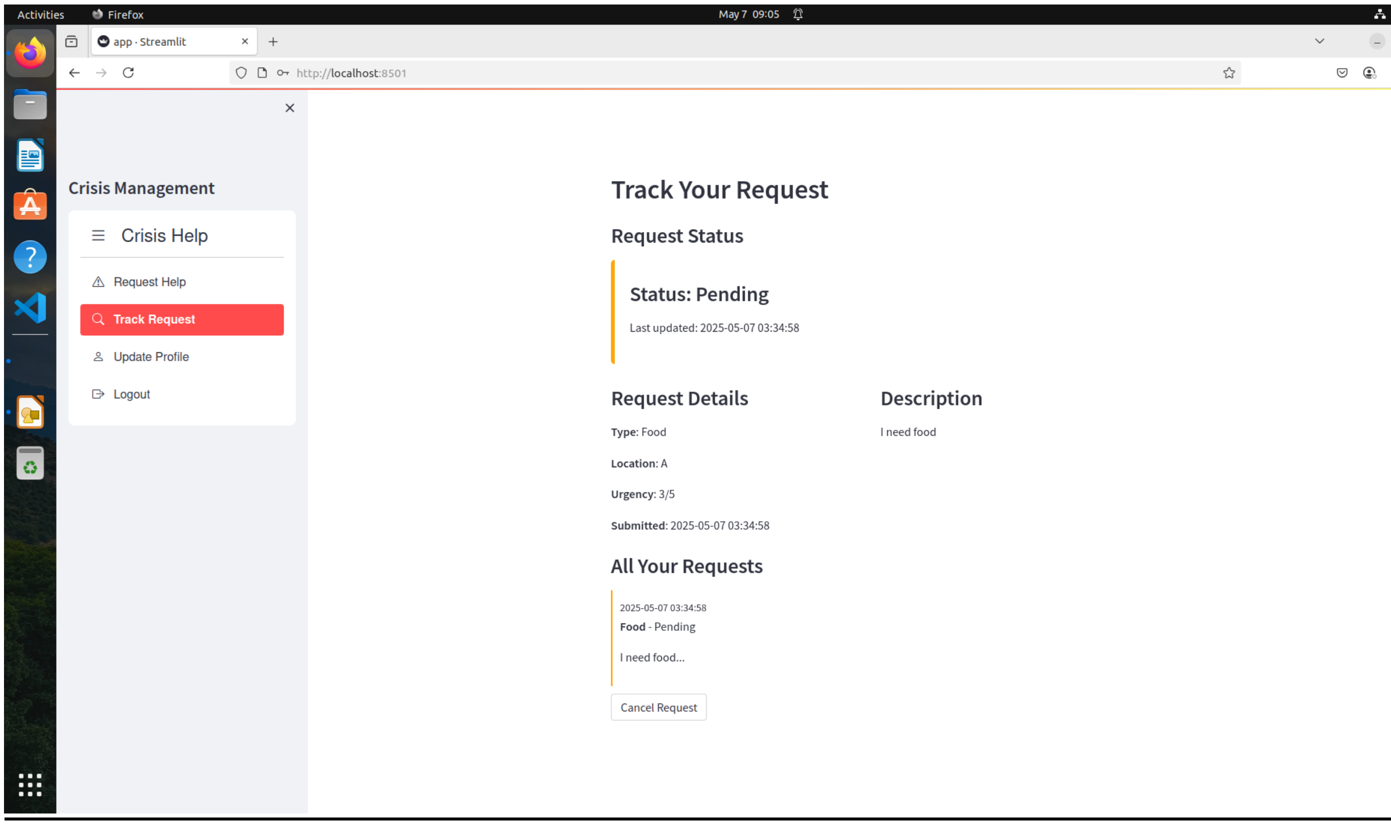
3

1 5

Submit Request

Your help request has been submitted. Help is on the way!

How it works



Crisis Management

Crisis Help

- Request Help
- Track Request
- Update Profile
- Logout

Track Your Request

Request Status

Status: Pending

Last updated: 2025-05-07 03:34:58

Request Details

Type: Food

Location: A

Urgency: 3/5

Submitted: 2025-05-07 03:34:58

Description

I need food

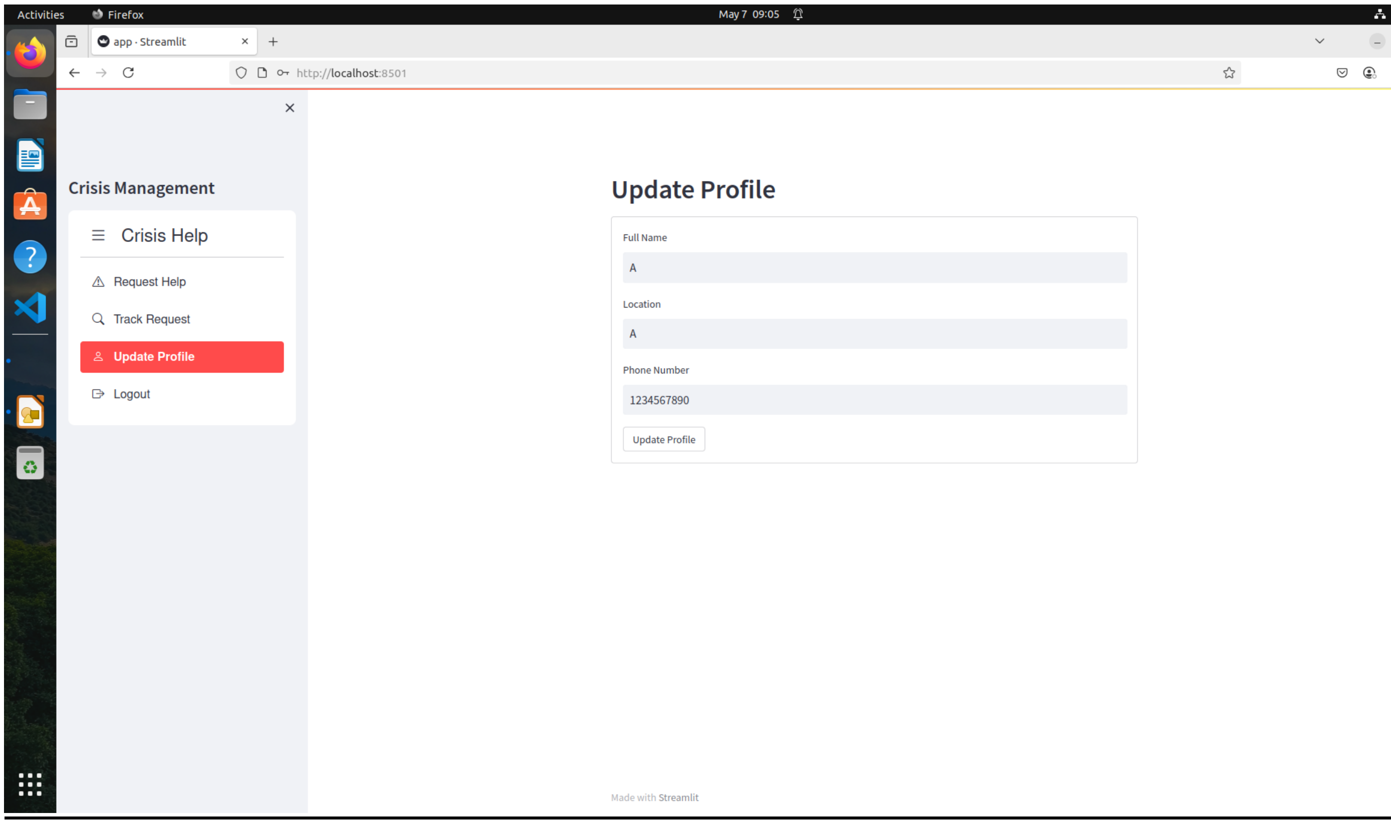
All Your Requests

2025-05-07 03:34:58

Food - Pending

I need food...

Cancel Request



Update Profile

Full Name

A

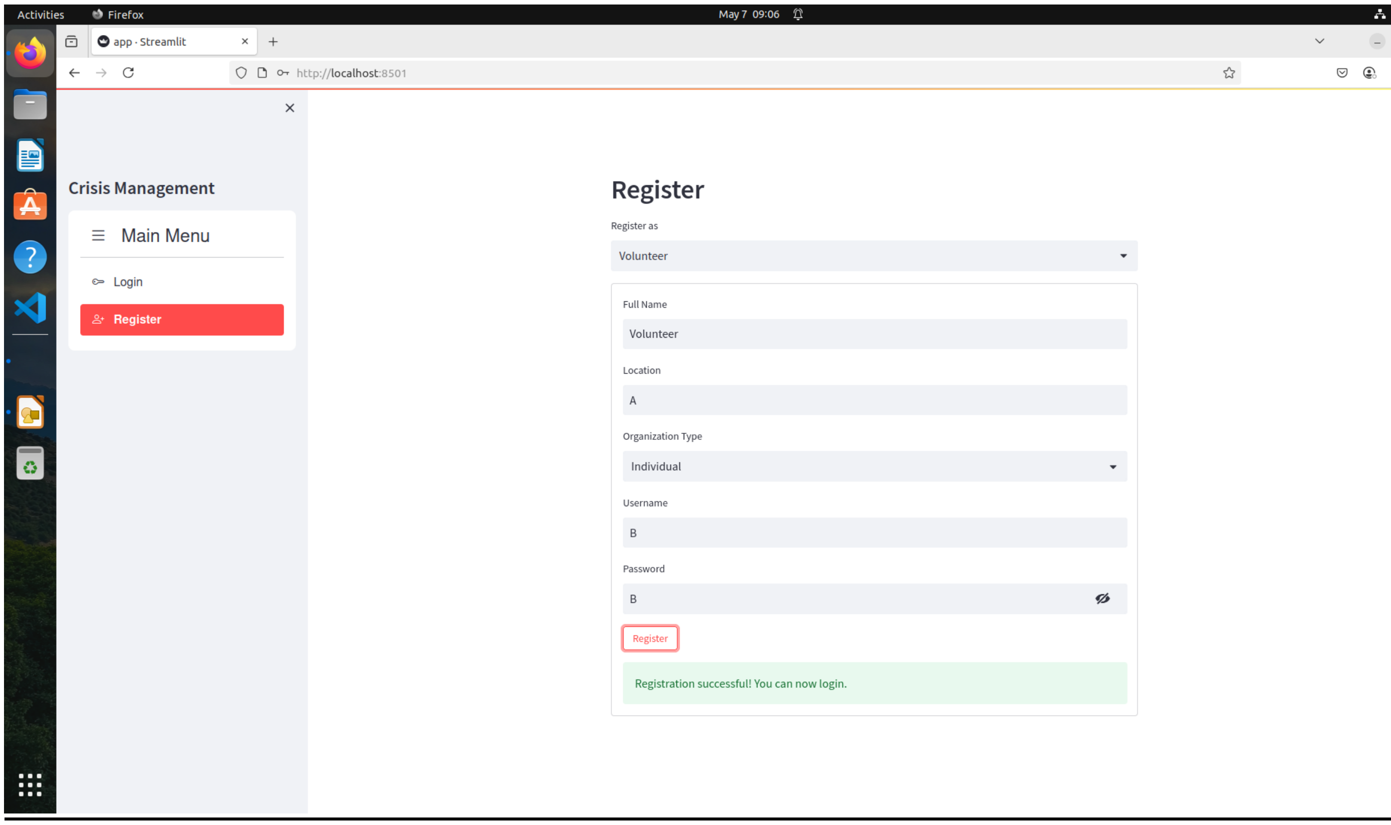
Location

A

Phone Number

1234567890

Update Profile



Register

Register as

Volunteer

Full Name

Volunteer

Location

A

Organization Type

Individual

Username

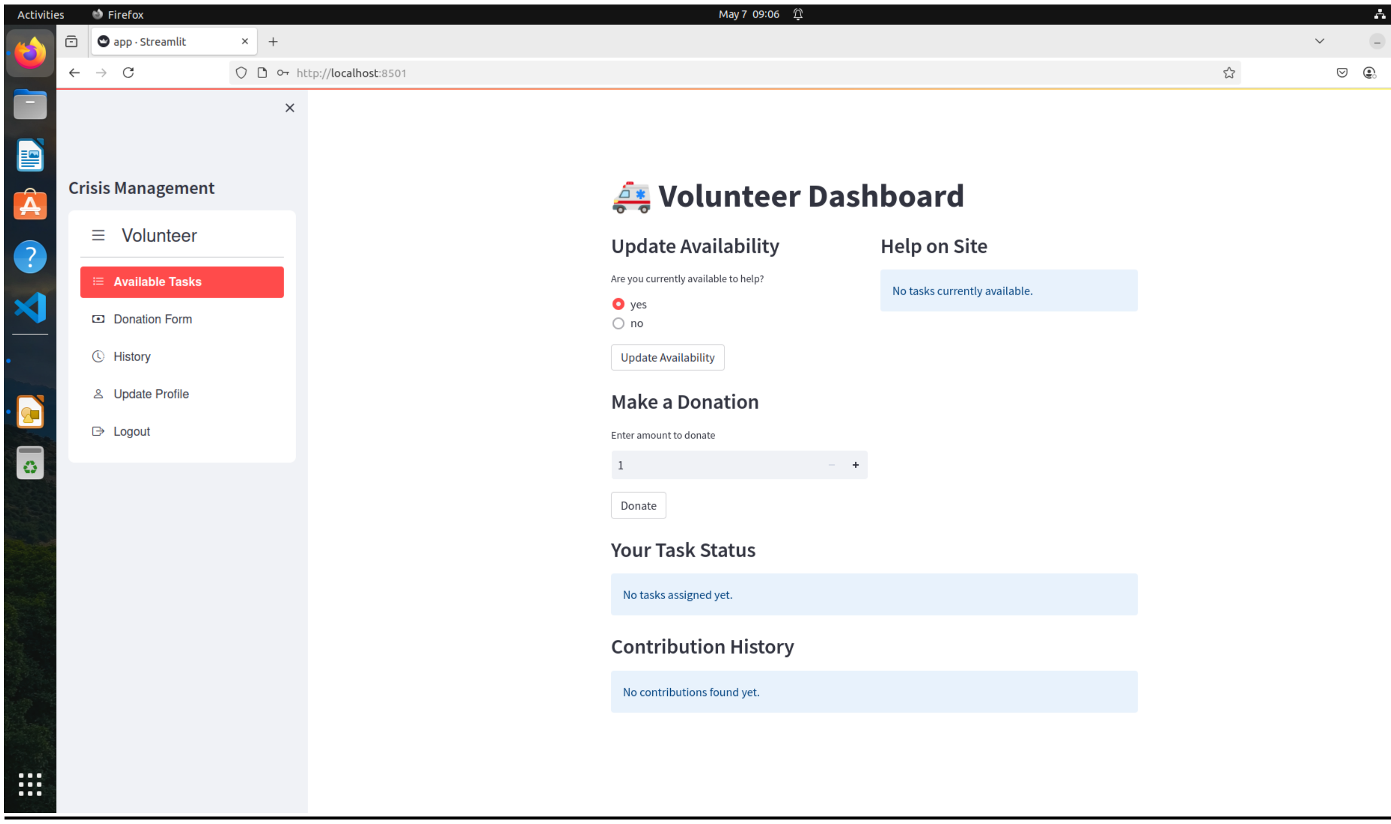
B

Password

B

Register

Registration successful! You can now login.



Volunteer Dashboard

Update Availability

Are you currently available to help?

[✓ yes] no

Update Availability

Help on Site

No tasks currently available.

Make a Donation

Enter amount to donate

1

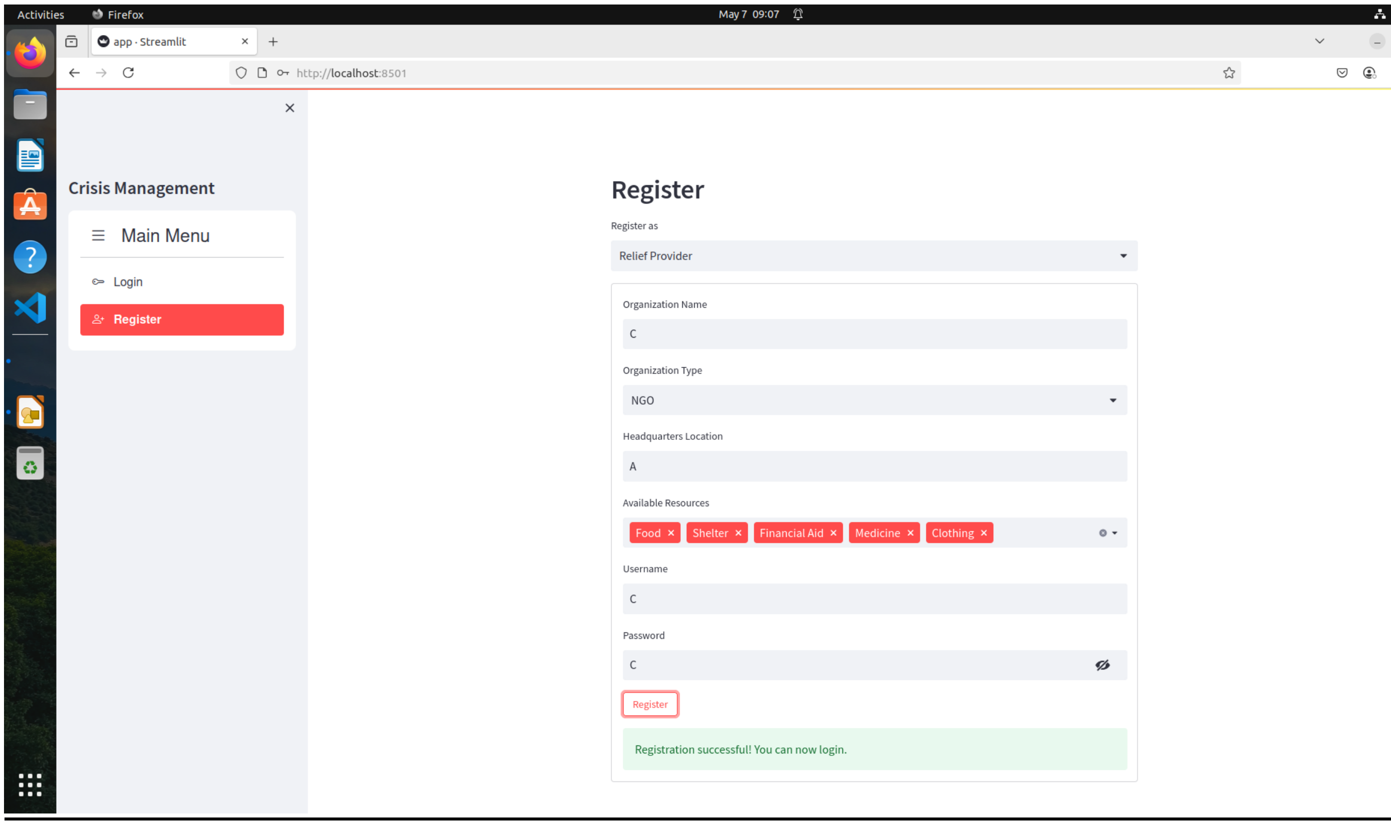
Donate

Your Task Status

No tasks assigned yet.

Contribution History

No contributions found yet.



Register

Register as: Relief Provider

Organization Name: C

Organization Type: NGO

Headquarters Location: A

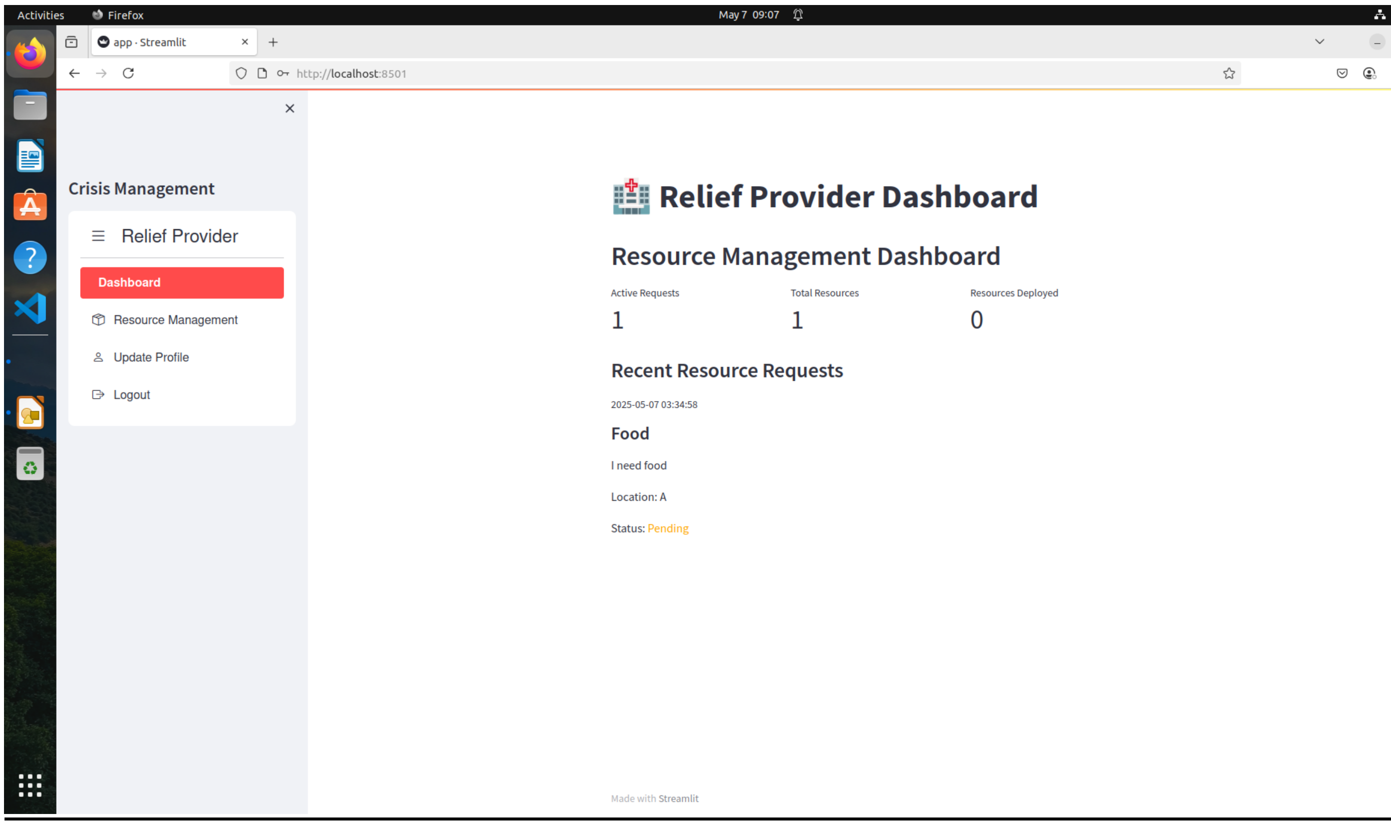
Available Resources: Food, Shelter, Financial Aid, Medicine, Clothing

Username: C

Password: C

Register

Registration successful! You can now login.



🏥 Relief Provider Dashboard

Resource Management Dashboard

Active Requests	Total Resources	Resources Deployed
1	1	0

Recent Resource Requests

2025-05-07 03:34:58

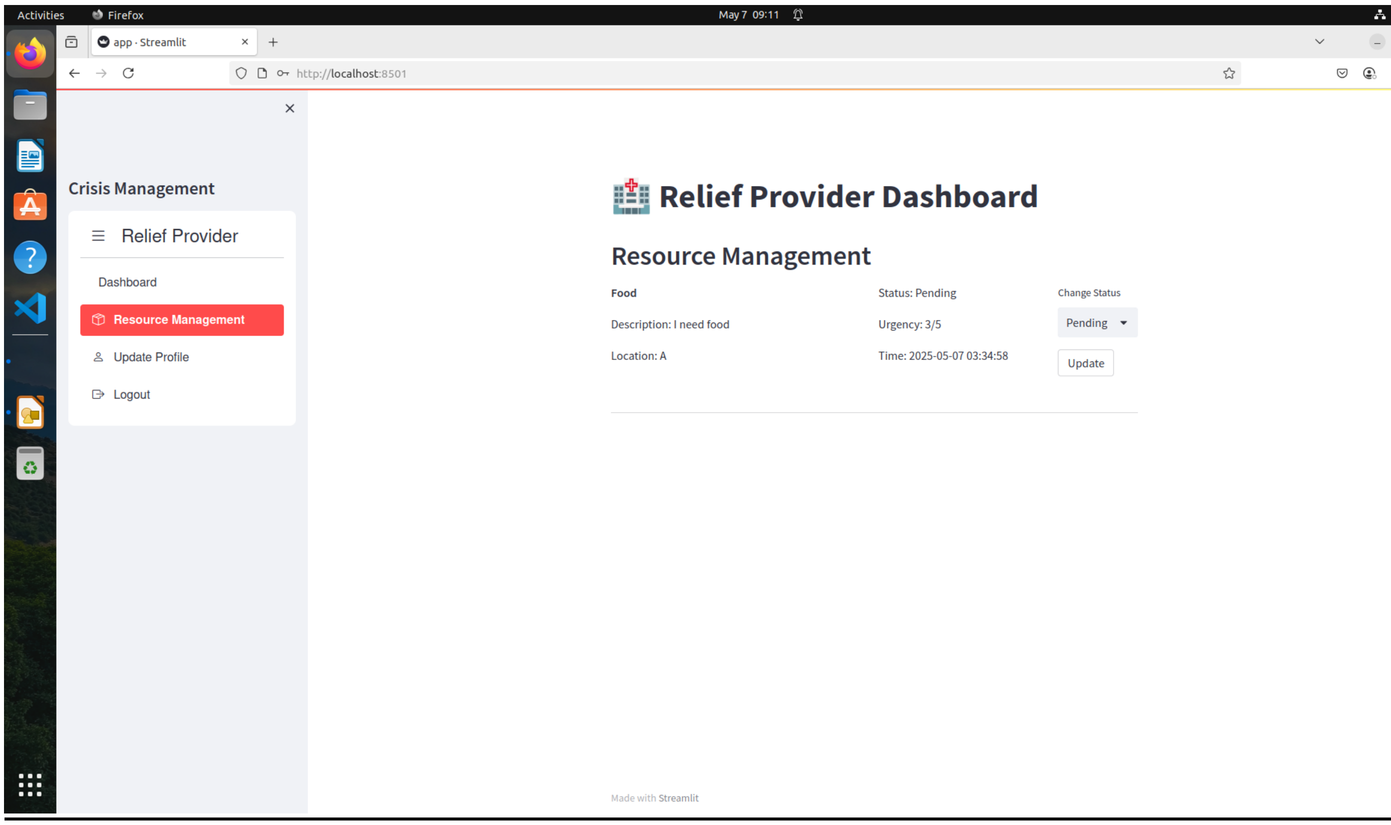
Food

I need food

Location: A

Status: Pending

Made with Streamlit

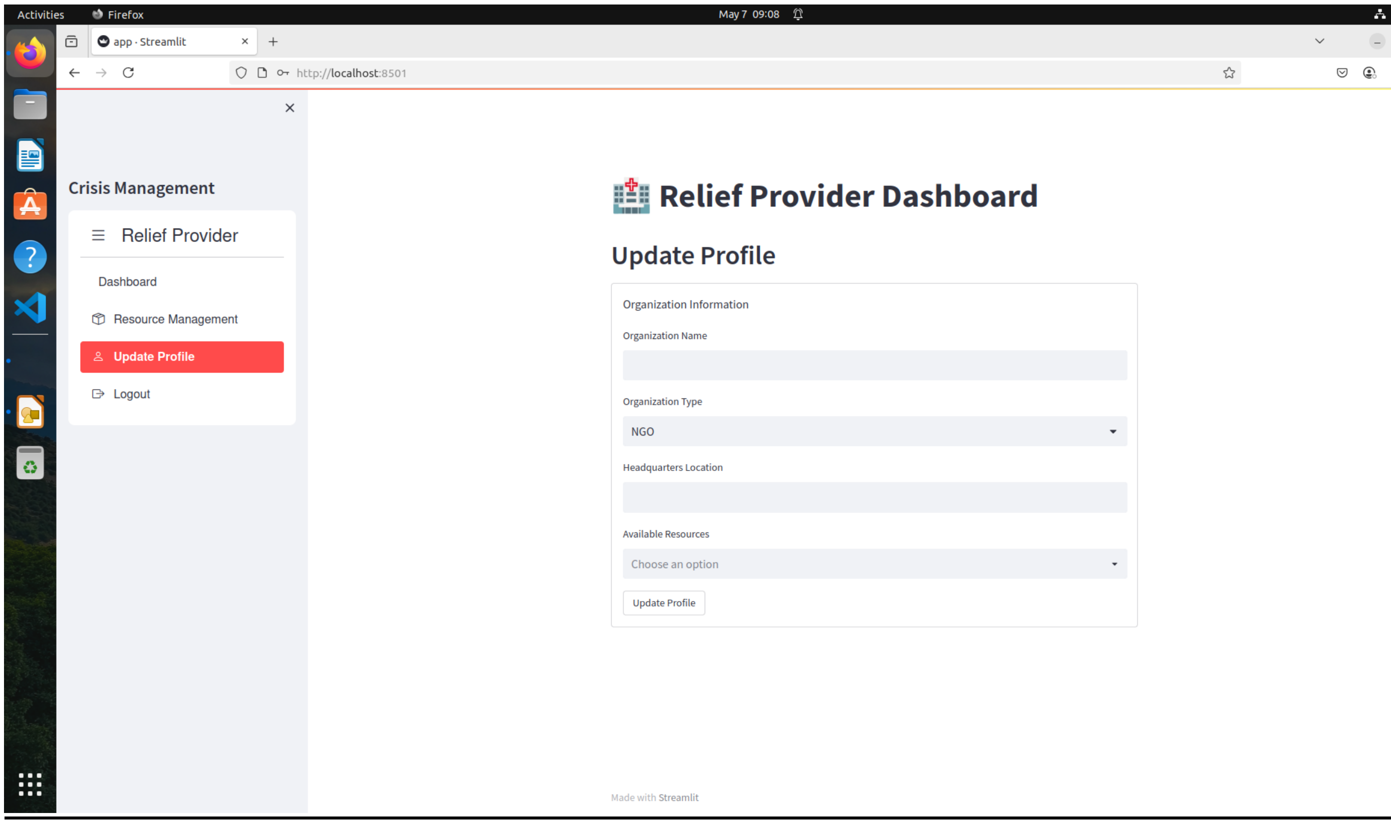


🏥 Relief Provider Dashboard

Resource Management

Food	Status: Pending	Change Status
Description: I need food	Urgency: 3/5	Pending
Location: A	Time: 2025-05-07 03:34:58	Update

Made with Streamlit



Relief Provider Dashboard

Update Profile

Organization Information

Organization Name

Organization Type

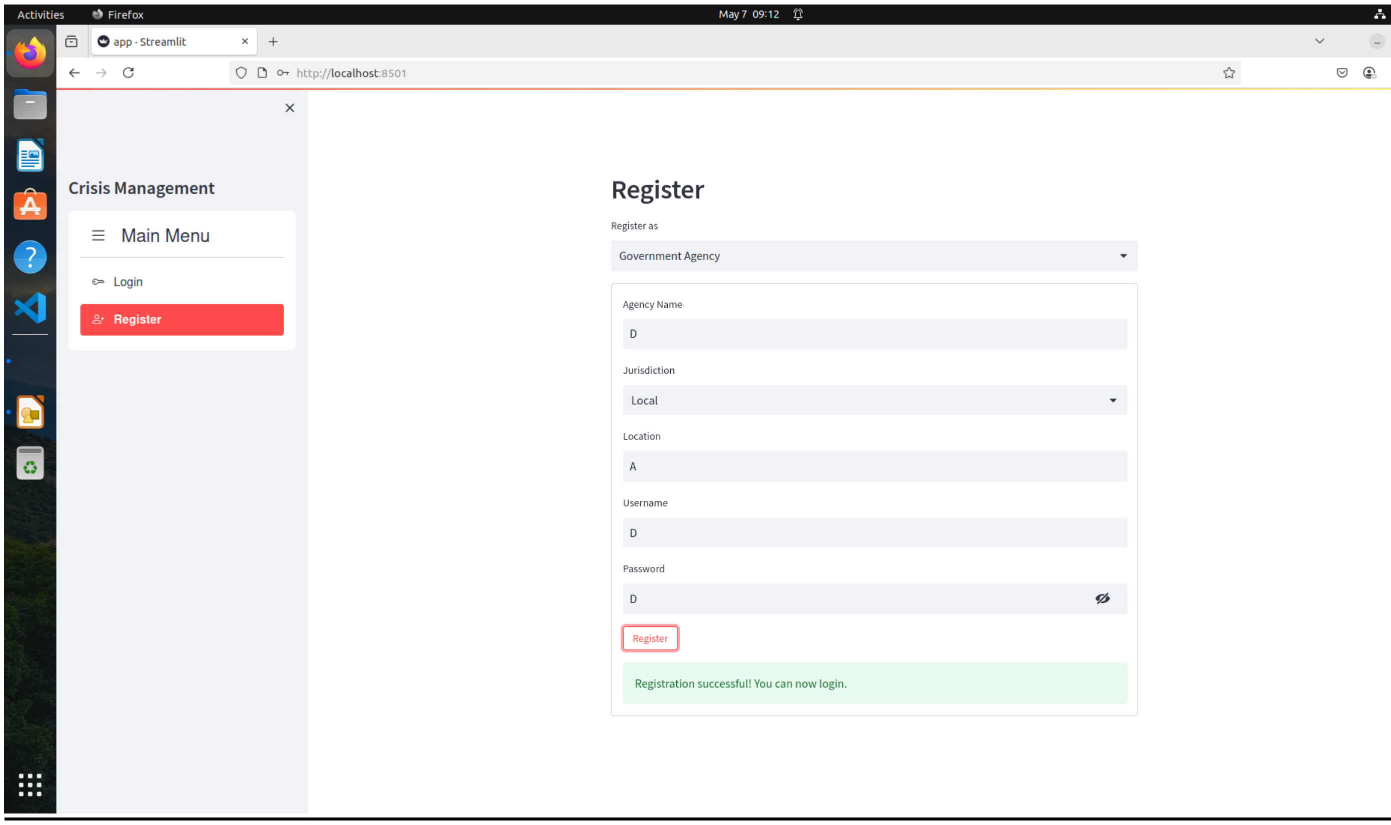
NGO

Headquarters Location

Available Resources

Choose an option

Update Profile



Register

Register as

Government Agency

Agency Name

D

Jurisdiction

Local

Location

A

Username

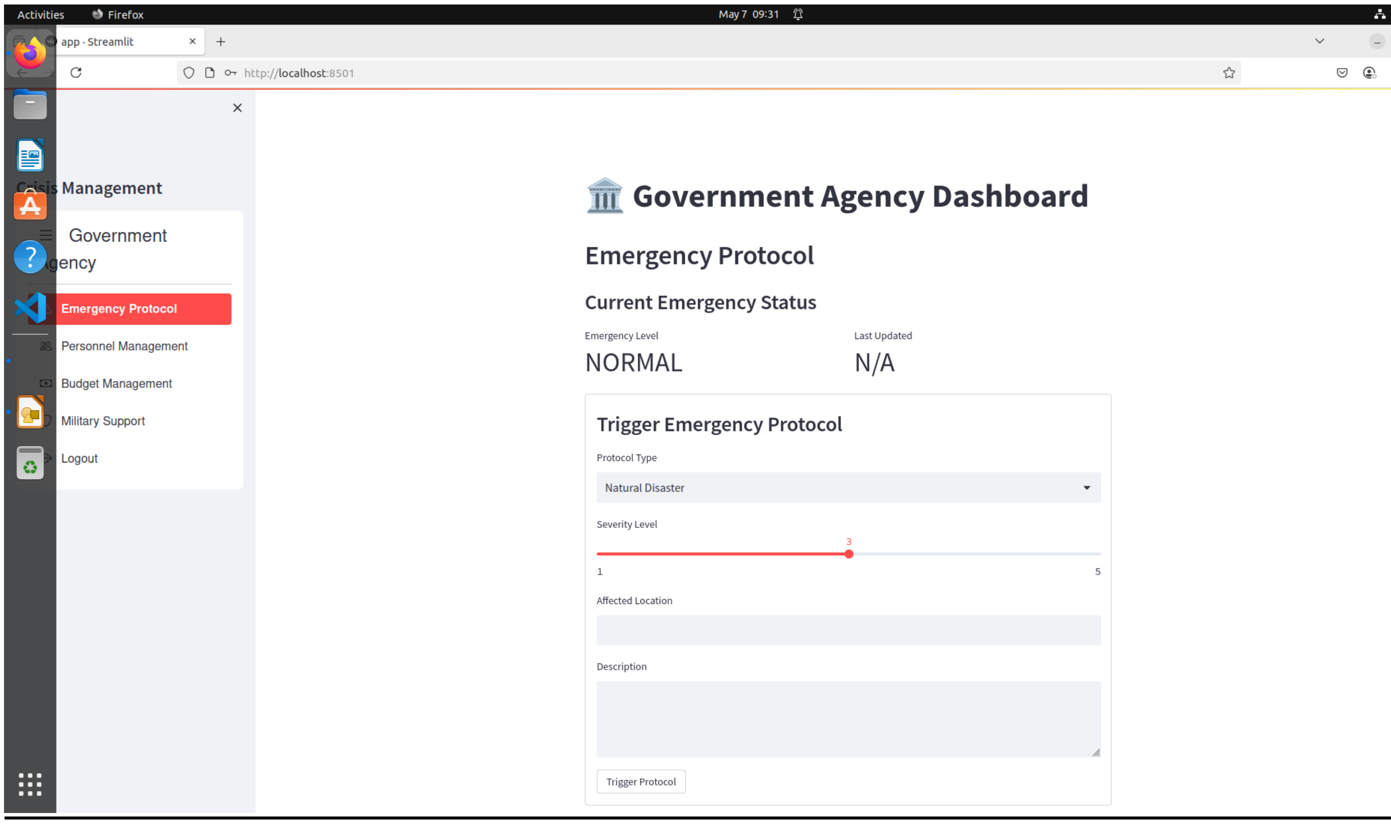
D

Password

D

Register

Registration successful! You can now login.


Activities Firefox May 7 09:31

app · Streamlit

http://localhost:8501

Crisis Management

Government Agency

- Emergency Protocol
- Personnel Management
- Budget Management
- Military Support
- Logout

🏛️ Government Agency Dashboard

Emergency Protocol

Current Emergency Status

Emergency Level: NORMAL

Last Updated: N/A

Trigger Emergency Protocol

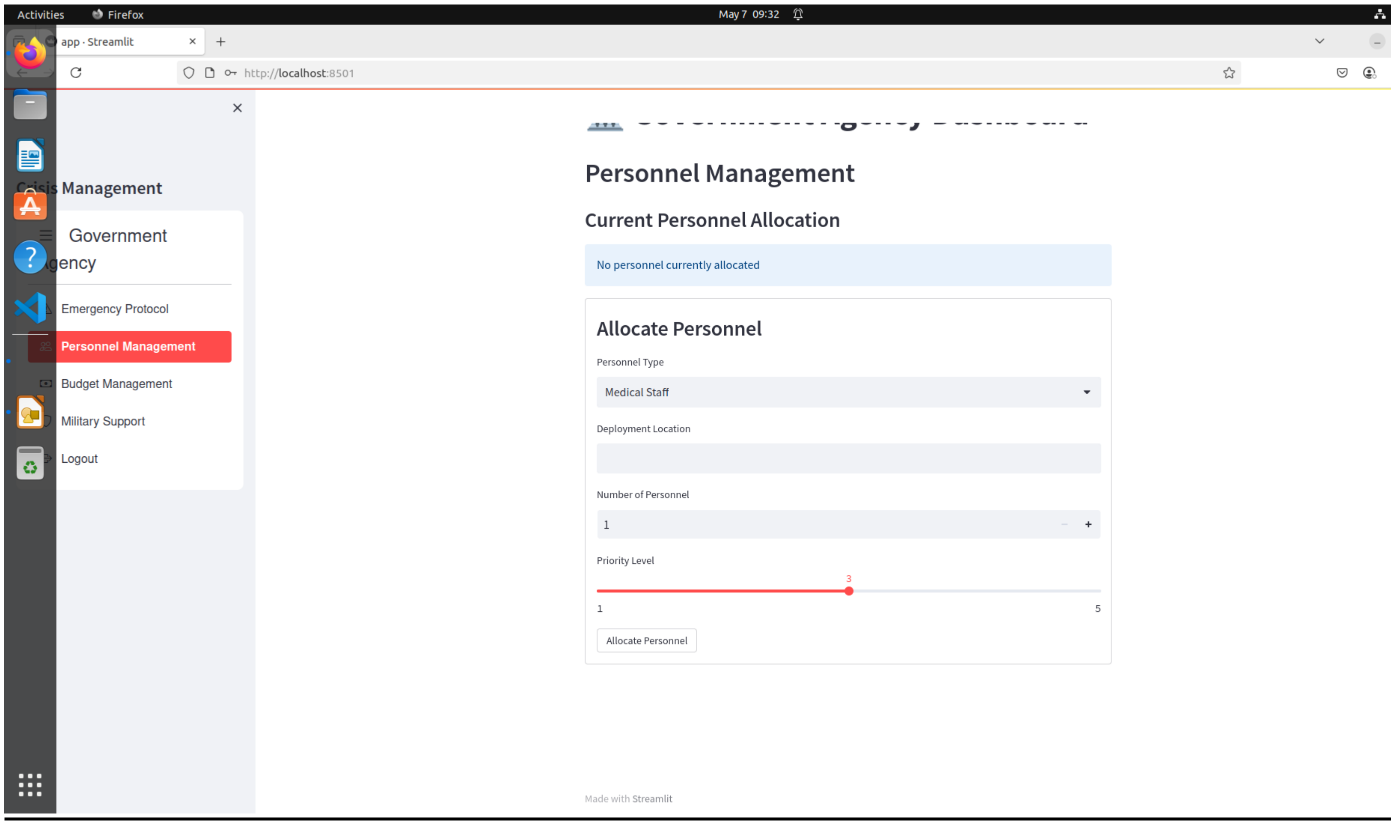
Protocol Type: Natural Disaster

Severity Level: 3 (1–5)

Affected Location:

Description:

Trigger Protocol


Activities Firefox May 7 09:32

app · Streamlit

http://localhost:8501

Crisis Management

Government Agency

- Emergency Protocol
- Personnel Management
- Budget Management
- Military Support
- Logout

Personnel Management

Current Personnel Allocation

No personnel currently allocated

Allocate Personnel

Personnel Type: Medical Staff

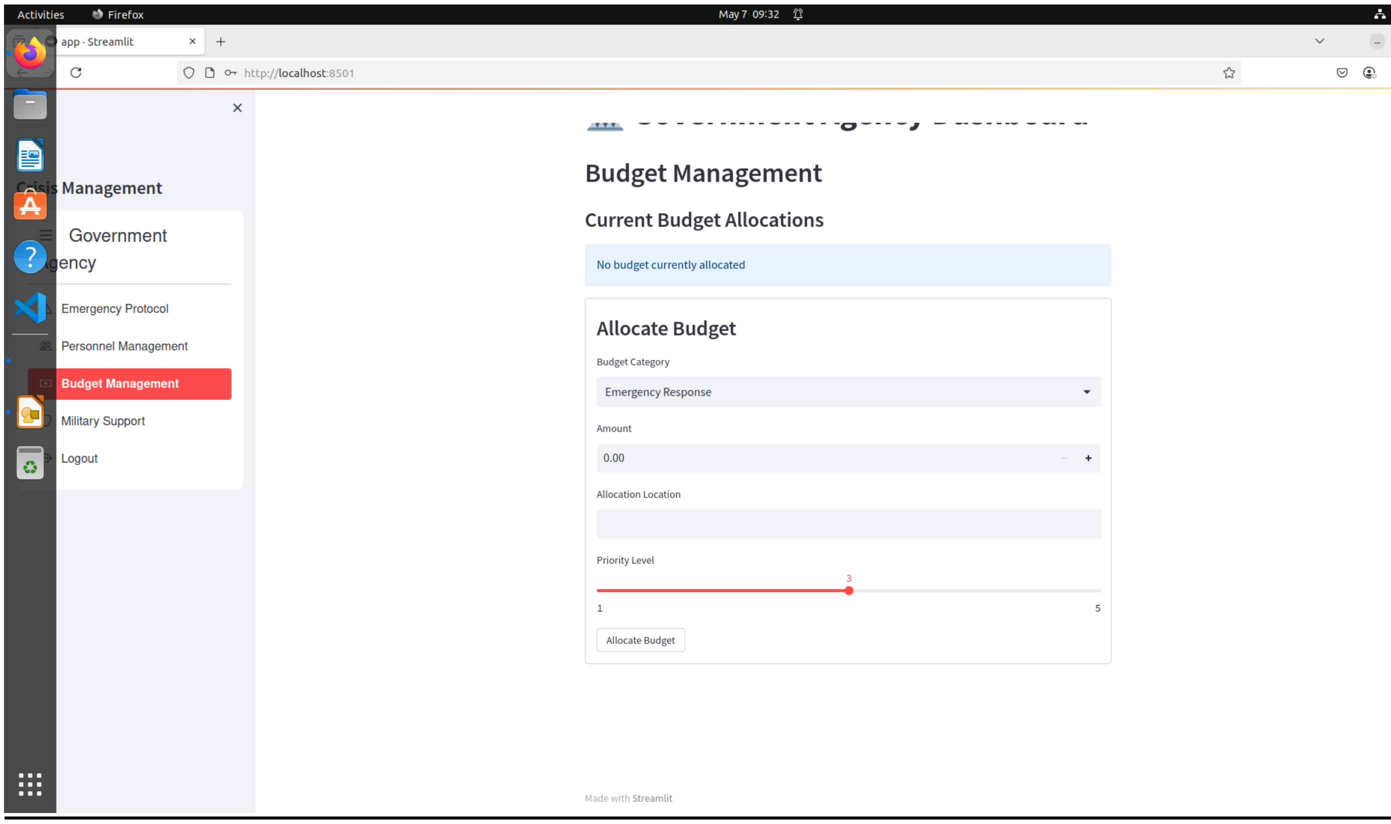
Deployment Location:

Number of Personnel: 1

Priority Level: 3 (1–5)

Allocate Personnel

Made with Streamlit


Activities Firefox May 7 09:32

app · Streamlit

http://localhost:8501

Crisis Management

Government Agency

- Emergency Protocol
- Personnel Management
- Budget Management
- Military Support
- Logout

Budget Management

Current Budget Allocations

No budget currently allocated

Allocate Budget

Budget Category: Emergency Response

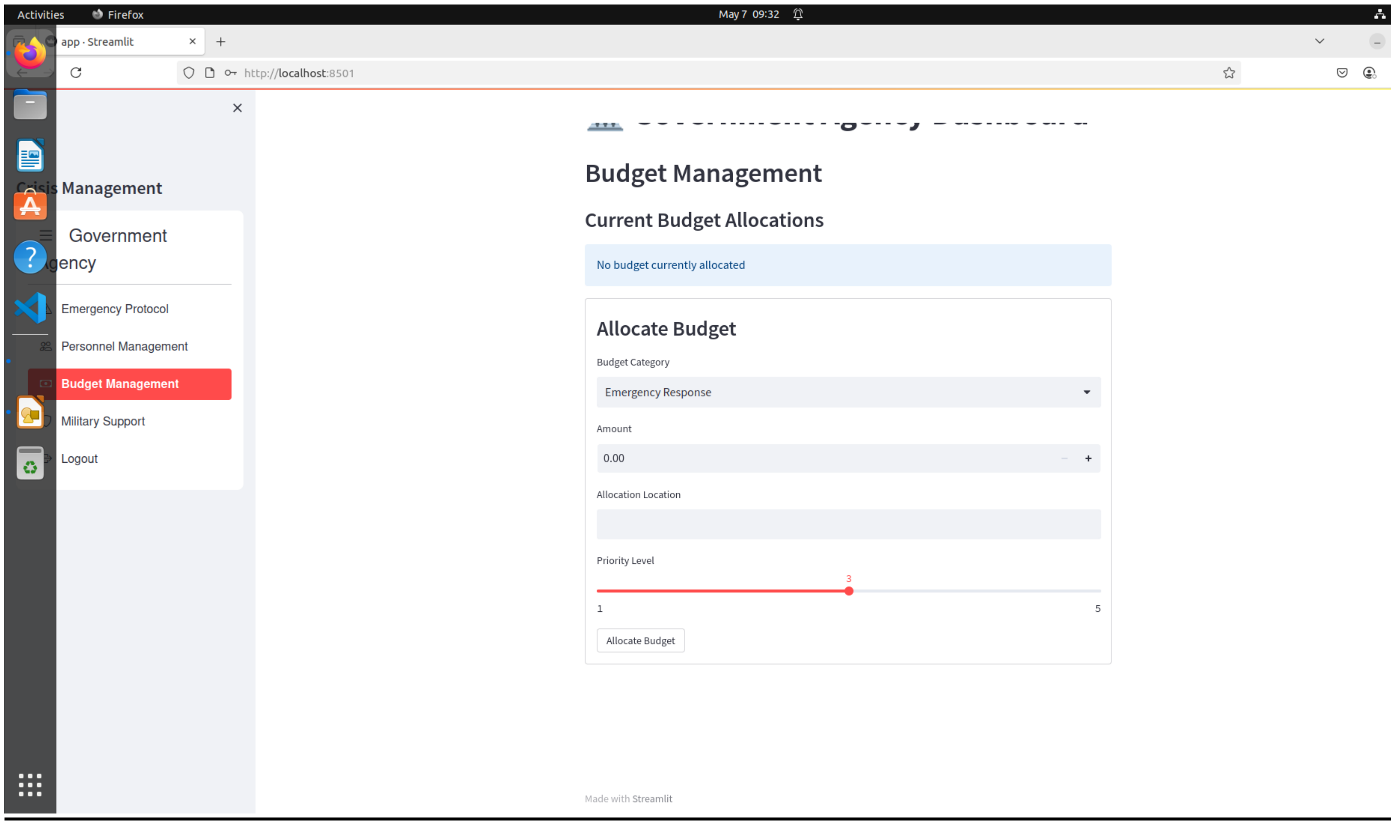
Amount: 0.00

Allocation Location:

Priority Level: 3 (1–5)

Allocate Budget

Made with Streamlit



Activities Firefox May 7 09:32

app · Streamlit

http://localhost:8501

Crisis Management

Government Agency

- Emergency Protocol
- Personnel Management
- Budget Management
- Military Support
- Logout

Budget Management

Current Budget Allocations

No budget currently allocated

Allocate Budget

Budget Category: Emergency Response

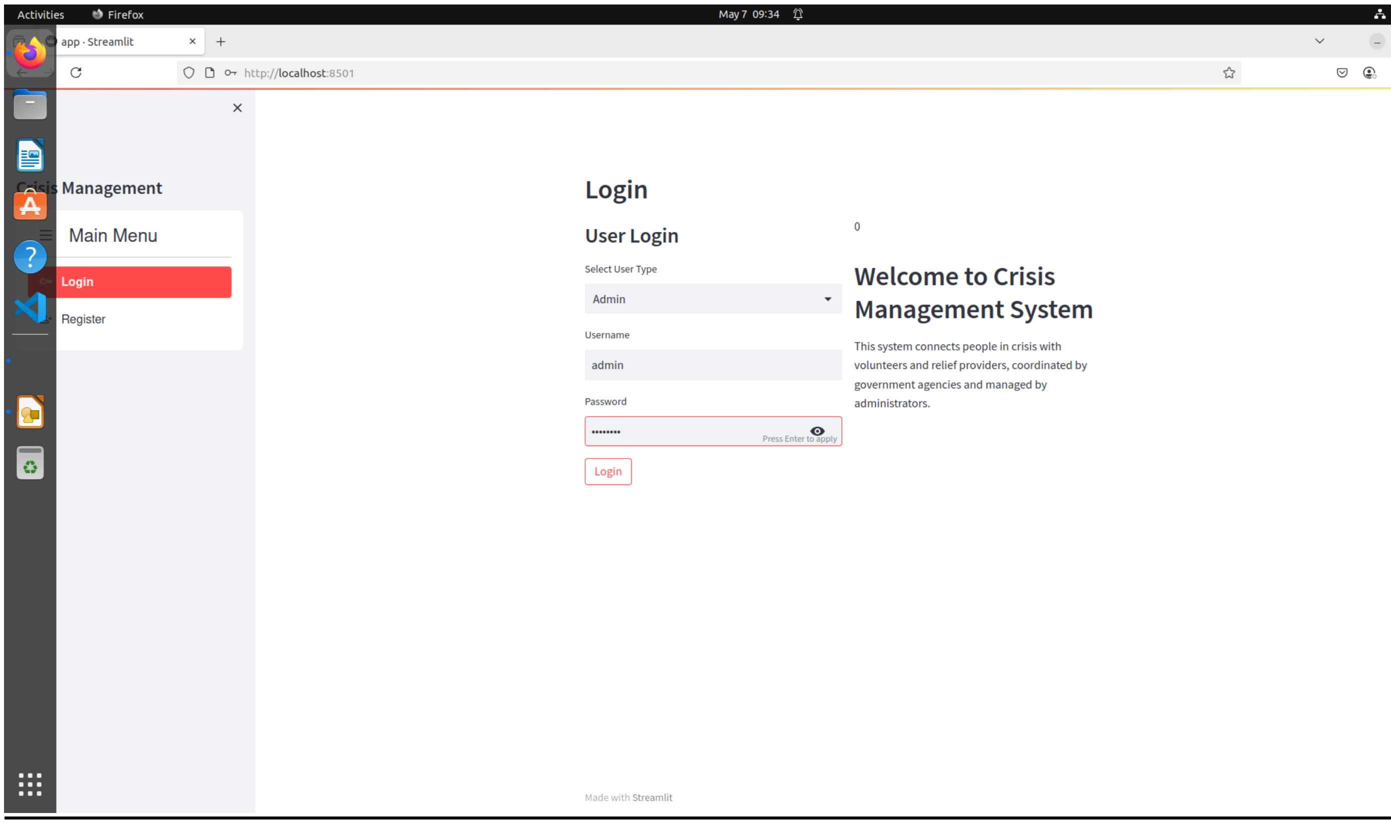
Amount: 0.00

Allocation Location:

Priority Level: 3 (1–5)

Allocate Budget

Made with Streamlit


Activities Firefox May 7 09:34

app · Streamlit

http://localhost:8501

Crisis Management

Main Menu

Login

Register

Login

User Login

Select User Type

Admin

Username

admin

Password

••••••••

Press Enter to apply

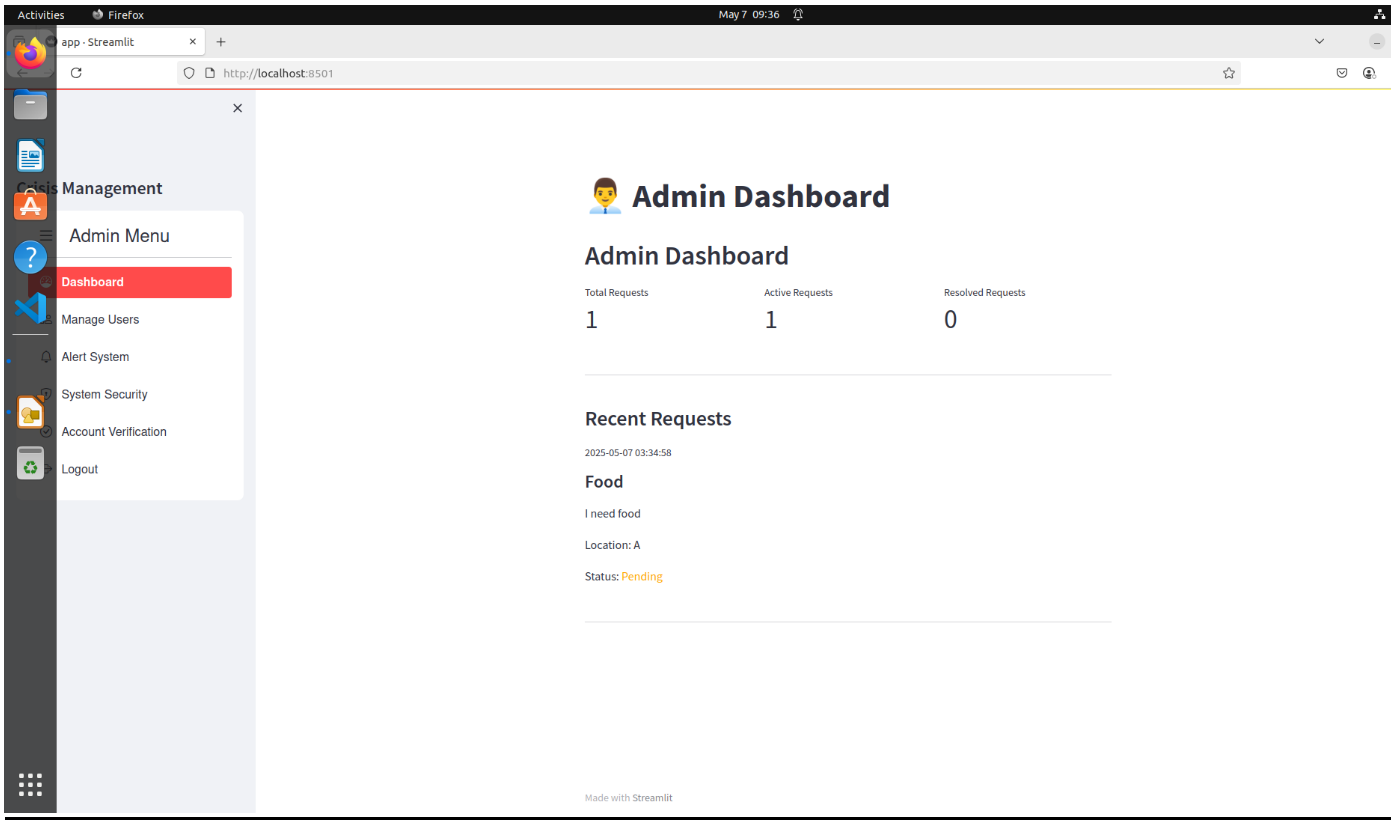
Login

0

Welcome to Crisis Management System

This system connects people in crisis with volunteers and relief providers, coordinated by government agencies and managed by administrators.

Made with Streamlit


Activities Firefox May 7 09:36

app · Streamlit

http://localhost:8501

Crisis Management

Admin Menu

Dashboard

Manage Users

Alert System

System Security

Account Verification

Logout

👨‍💼 Admin Dashboard

Admin Dashboard

Total Requests	Active Requests	Resolved Requests
1	1	0

Recent Requests

2025-05-07 03:34:58

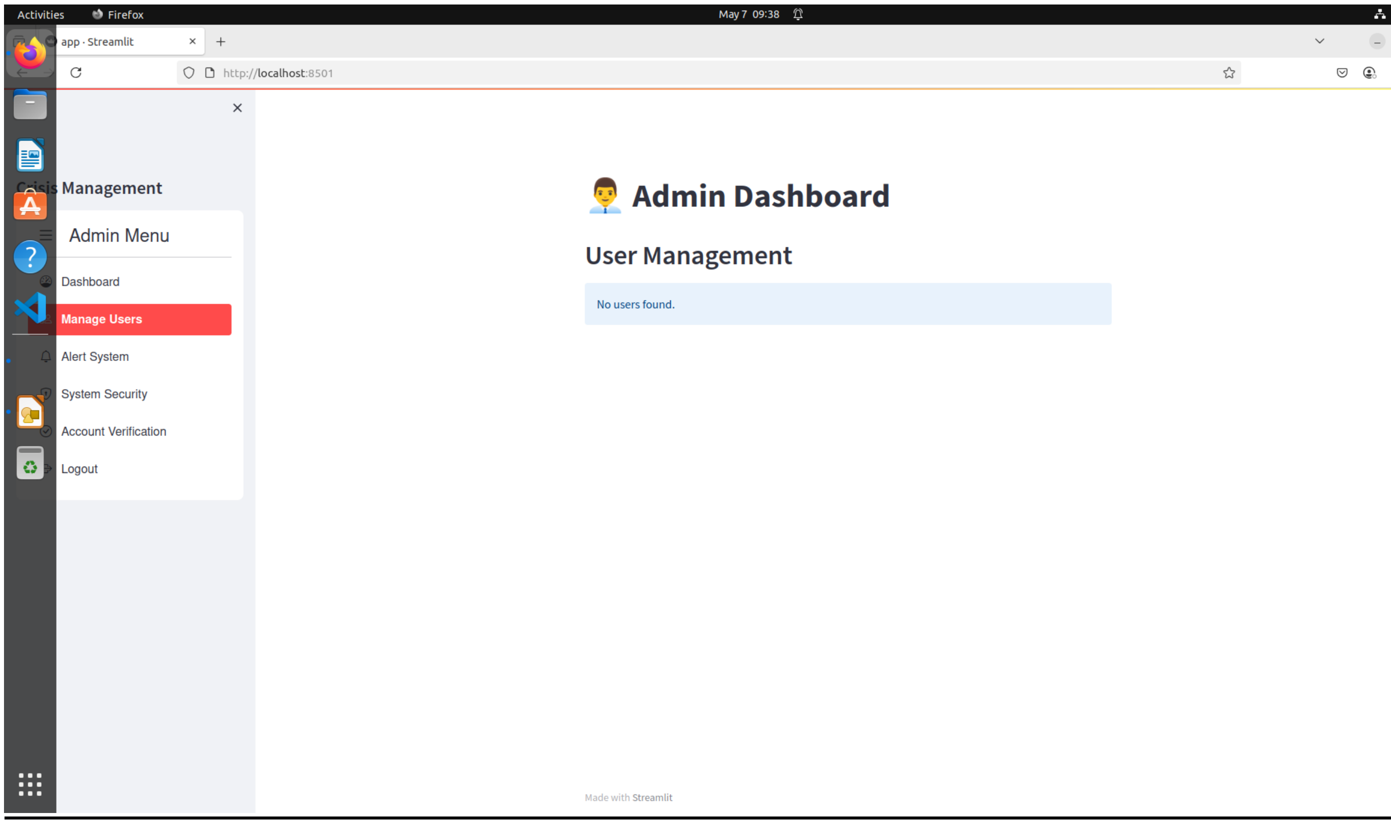
Food

I need food

Location: A

Status: Pending

Made with Streamlit



Activities Firefox May 7 09:38

app · Streamlit

http://localhost:8501

Crisis Management

Admin Menu

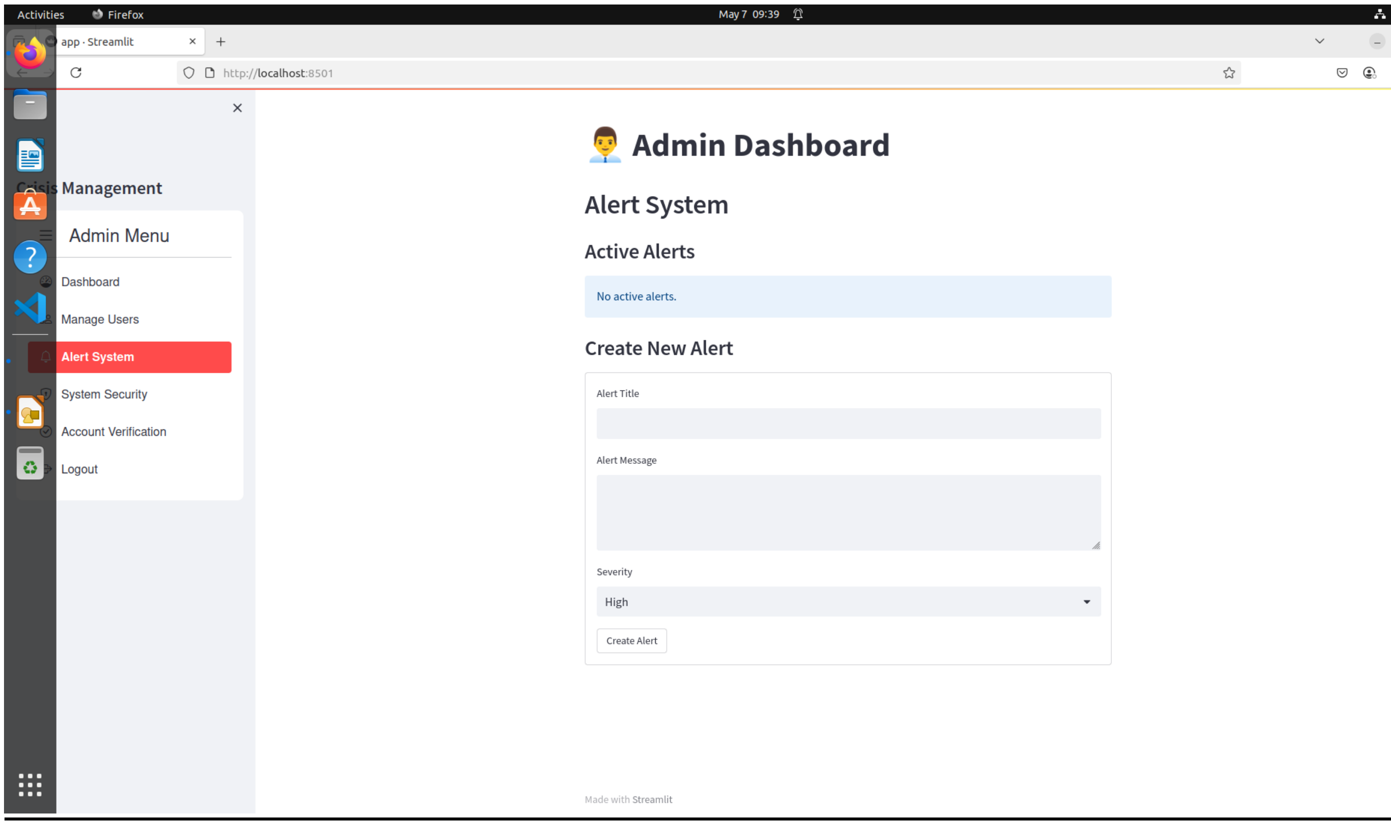
- Dashboard
- Manage Users
- Alert System
- System Security
- Account Verification
- Logout

👨‍💼 Admin Dashboard

User Management

No users found.

Made with Streamlit



Activities Firefox May 7 09:39

app · Streamlit

http://localhost:8501

Crisis Management

Admin Menu

- Dashboard
- Manage Users
- Alert System
- System Security
- Account Verification
- Logout

👨‍💼 Admin Dashboard

Alert System

Active Alerts

No active alerts.

Create New Alert

Alert Title

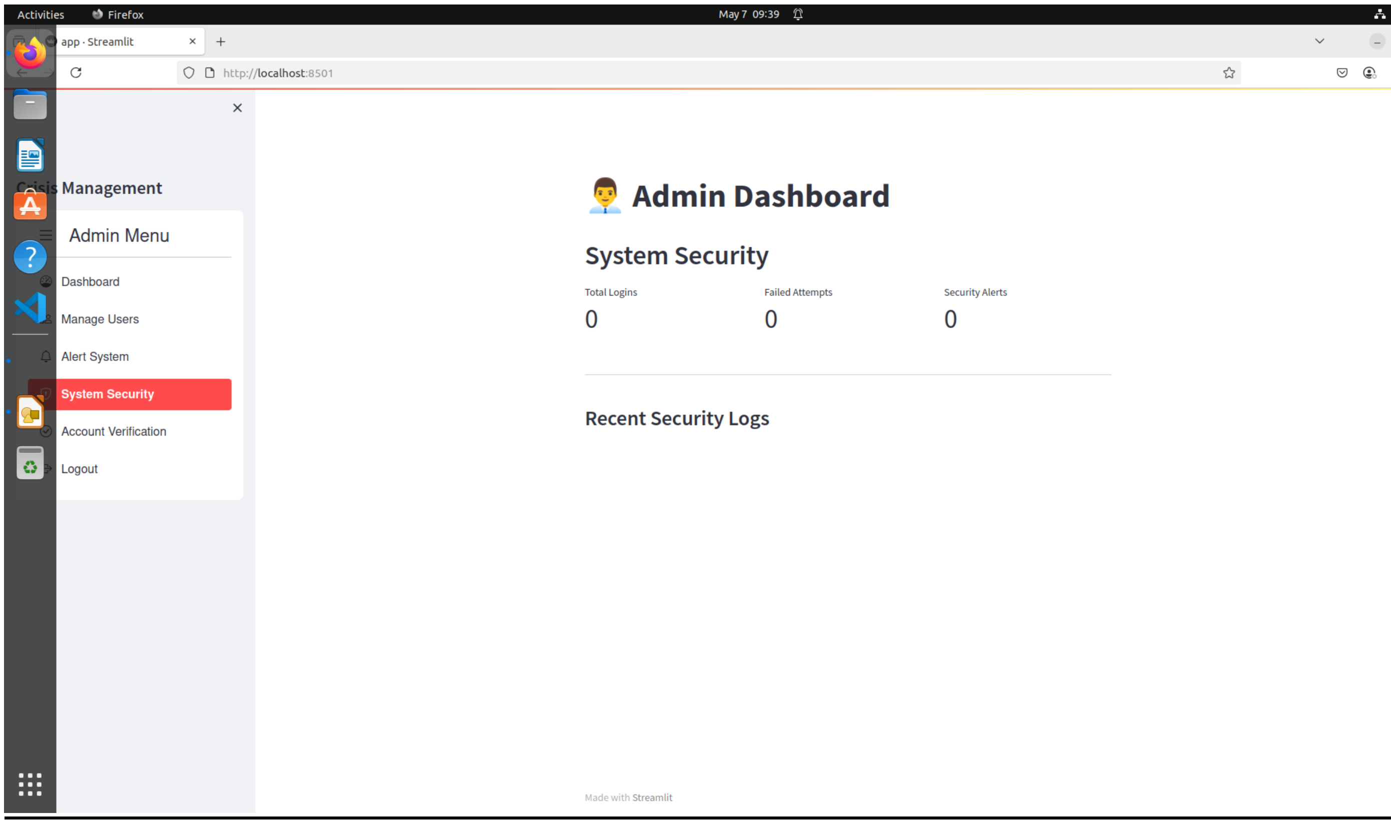
Alert Message

Severity

High

Create Alert

Made with Streamlit



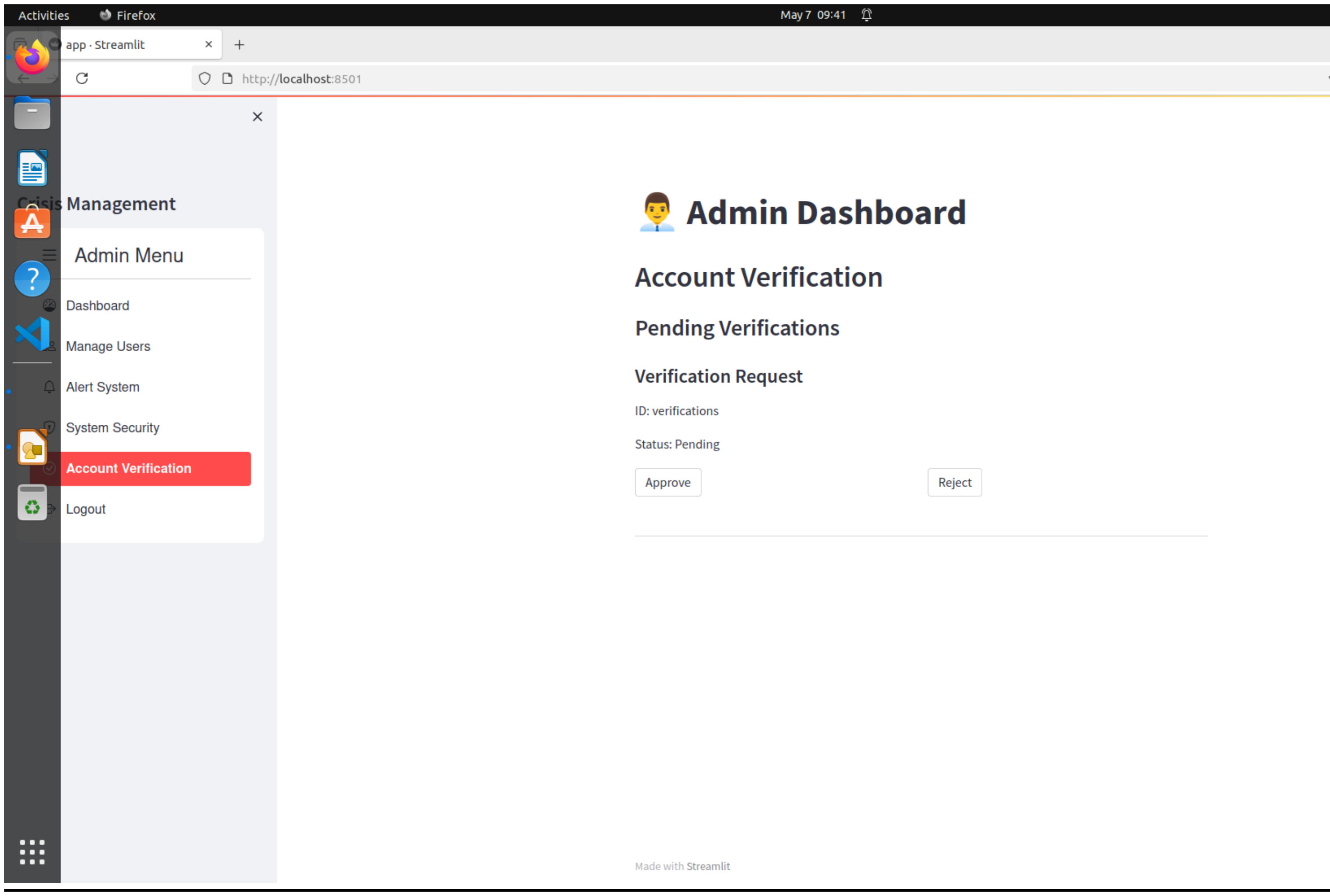
👨‍💼 Admin Dashboard

System Security

Total Logins	Failed Attempts	Security Alerts
0	0	0

Recent Security Logs

Made with Streamlit



👨‍💼 Admin Dashboard

Account Verification

Pending Verifications

Verification Request

ID: verifications

Status: Pending

Made with Streamlit

Individual contributions of the team members:

Name	**Module worked on**
Namritha Lasyapriya Maddali	Volunteer + Admin
Nikhil M	Relief Provider + AlertSystem
Pradhaan S Bhat	PeopleInCrisis + Help Request + GovernmentAgency + Integration