VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi: 590 018

A Mini Project Report
On
"resQSquad"

Submitted in partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering in Computer Science and Engineering

Submitted by

Ankit Kumar(1VE21CS014)
Ankit Singh(1VE21CS015)
Atul Yadav(1VE21CS024)

Under the Guidance of

Lokesh M
Assistant Professor
Dept. of CSE,
SVCE, Bengaluru.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

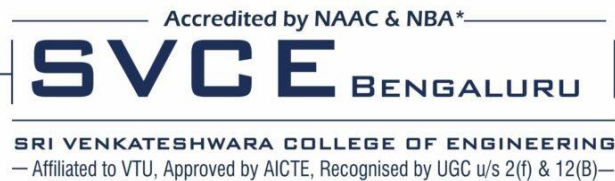# SRI VENKATESHWARA COLLEGE OF ENGINEERING

Affiliated to VTU Belgaum & Approved by AICTE New Delhi) an ISO 9001:2008 Certified, Kempegowda International Airport road, Vidyanagar, Bengaluru, Karnataka, India-562157

2023– 2024

SRI VENKATESHWARA COLLEGE OF ENGINEERING

Vidyanagar, Bengaluru, Karnataka, India-562157

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the Mini Project entitled **"resQSquad"** is submitted by **ANKIT KUMAR [1VE21CS014], ANKIT SINGH [1VE21CS015], ATUL YADAV [1VE21CS024]**, on partial fulfillment of sixth semester, Bachelor of Engineering in Computer Science and Engineering, Visvesvaraya Technological University for the academic year 2023-2024.

..…………..………	…………………………	..………………....
Signature of Guide	**Signature of Coordinator**	**Signature of HOD**
Lokesh M	**Suresh P**	**Dr. Hema MS**
Assistant Professor	Assistant Professor	Assistant Professor
Dept. of CSE,	Dept. of CSE,	Dept. of CSE,
SVCE, Bengaluru	SVCE, Bengaluru	SVCE, Bengaluru

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without complementing those who made it possible, whose guidance and encouragement made our efforts successful.

I sincerely thank the highly esteemed institution SRI VENKATESHWARA COLLEGE OF ENGINEERING for grooming me into a software engineer.

I sincerely thank **Dr. Nageswara Guptha**, Principal, SVCE, Bengaluru for providing the required facility.

I am extremely thankful to **Dr. Hema M S**, HOD of CSE, SVCE for providing support and encouragement.

I am grateful to **Mr. Lokesh M**, Asst. Professor, Dept. of CSE, SVCE who helped me to complete this project successfully by providing guidance, encouragement and valuable suggestion during entire period of the project. I thank all my computer science staff and others who helped directly or indirectly to meet my project work with grand success.

Finally, I am grateful to my parents and friends for their invaluable support guidance and encouragement.

ANKIT KUMAR [1VE21CS014]

ANKIT SINGH[1VE21CS015]

ATUL YADAV[1VE21CS024]

INDEX

Abstract

This mini-project explores the critical role of mental health services in disaster response and recovery. By examining the literature on emergency and disaster medicine, critical incident stress management, and the psychological impact of disasters, this study aims to understand the complex interplay between physical and mental health challenges in disaster-affected populations. The research investigates the effectiveness of various mental health interventions, identifies protective factors, and highlights the importance of preparedness and community resilience. Ultimately, this project contributes to developing more comprehensive and effective disaster response plans that prioritize the mental well-being of affected individuals, fostering long-term recovery and resilience within communities.

.

Chapter-1

Introduction

Disasters unleash chaos, leaving communication in tatters. Rescue teams, racing against time, find themselves grappling with the challenge of transmitting critical messages through unreliable channels. Traditional methods often prove slow and inefficient, hindering their ability to request backup and resources urgently. This delay can be catastrophic, as every second counts in a disaster zone.

resQSquad emerges as a beacon of hope, transforming disaster response through seamless communication. This innovative mini-application connects rescue teams with NGOs and authorized users, creating a real-time network for swift information exchange. By enabling instant alerts and requests for backup or specialized equipment, resQSquad empowers rescue teams to coordinate their efforts effectively, ultimately saving lives.

1.1 Background

The aftermath of a disaster paints a grim picture: collapsed buildings, devastated landscapes, and a desperate race against time. Amidst this chaos, communication often becomes a casualty. Traditional methods like radios and phone calls can be slow, unreliable, or overwhelming. This creates a nightmare scenario for rescue teams, who struggle to get critical messages through. Urgent requests for backup or vital resources like medical supplies or specialized equipment can go unheard, hindering their ability to reach and save those in need.

This is where the need for a more robust and efficient communication system becomes paramount. The concept behind resQSquad stems from this critical gap. Imagine a scenario where outdated methods fail, leaving rescue teams in a collapsed building desperately needing additional manpower or equipment.

1.2   Objective

- **To Reduce Response Times:** Streamlined communication leads to faster deployment of resources and personnel, allowing rescue teams to reach survivors in critical moments.
- **To Improve Rescue Efficiency:** Real-time requests and resource coordination minimize duplication of efforts and optimize team effectiveness.
- **To Save Lives:** Faster response times and improved coordination ultimately mean more lives saved during disaster situations.

Chapter-2

LITERATURE REVIEW

CISM

Critical Incident Stress Management (CISM) is a multi-phase intervention. It evolved from earlier, singular crisis response methods. CISM encompasses pre-incident preparation, immediate post-incident care, and ongoing support.

Key Features:

- CISM is a multi-phase intervention including pre-incident training, acute care, and post-incident follow-up.
- It addresses the psychological impact of critical incidents on individuals and groups.
- CISM emphasizes peer support, education, and symptom management.

Disaster mental health services

Disaster mental health services focus on addressing the psychological and emotional toll of disasters on individuals, families, and communities. Disasters, whether natural or human-made, can cause widespread trauma and disrupt lives, leading to a range of mental health challenges. These services aim to provide support, treatment, and prevention strategies to mitigate the long-term effects of disaster-related stress.

- Provides immediate and ongoing support to disaster survivors and responders.
- Offers a range of services including crisis counselling, trauma therapy, and community-based programs.
- Aims to reduce mental health symptoms, promote resilience, and facilitate recovery.
- Collaborates with other disaster relief organizations for comprehensive support.

Chapter-3

Design and Implementation

3.1 Requirements and Specifications

3.1.1 Software Requirement

- **Operating System:** Windows (any recent version should suffice)
- **Programming Language:** Python 3 (version 3.7 or later recommended)
- **Web Framework:** Django (version compatible with your chosen Python version)
- **Code Editor:** Visual Studio Code (highly recommended, but any code editor with Python support will work)
- **Optional:** Git (version control system for managing your project code.)

3.1.2 Hardware requirements

- Processor: 1.1 GHz or faster (multi-core processor recommended for better performance)
- Memory (RAM): 2 GB minimum (4 GB or more recommended for smoother development experience)
- Storage: Enough disk space to accommodate the operating system, development tools, and your project files (minimum not specified, but aim for at least 20 GB)
- Internet Connection (recommended): For downloading software, accessing online tutorials, and collaborating with others (if applicable)

3.2 Diagram

Fig 3.2.1: flow diagram of resQSquad

3.3 Implementation

3.3.1 FRONT END

resQSquad leverages the power of HTML and Django templates to build its user interface.

- **HTML Foundation:** Standard HTML provides the core structure and visual elements for the application, including forms for user input, buttons for actions, and informative displays.

- **Django Template Magic:** Django templates extend HTML functionalities through the Django Template Language (DTL). DTL empowers you to:

  - **Display Dynamic Data:** Integrate template variables to showcase real-time information retrieved from Python code within the HTML.

- **Simplify Complexities:** Implement logic and conditional statements directly in the templates for a more dynamic user experience.
- **Loop with Ease:** DTL facilitates looping through data lists retrieved from the database, presenting them in a user-friendly format.

This approach offers several advantages:

- **Separation of Concerns:** HTML handles presentation, while Python manages logic, promoting clear and maintainable code.
- **Dynamic Content:** Templates allow for content to adapt based on user interactions or real-time data, leading to a more engaging interface.
- **Security Focus:** Django's built-in mechanisms prevent security vulnerabilities like XSS by automatically escaping user input within templates.

In essence, this combination of HTML and Django templates allows for a secure, dynamic, and user-friendly front-end experience for ResQSquad.

3.3.2 BACK END

The core logic and functionality of ResQSquad reside in its Python-powered back-end. This is where the magic happens! Built upon the robust Django framework, the back-end provides a foundation for user management (authentication and authorization), database management (storing and retrieving critical information), URL routing (directing user requests), and even an administrative interface for managing application data and users. Beyond Django, additional Python libraries may be employed to enhance specific functionalities, such as identifying the nearest authorized agency based on a rescue team's location or sending real-time alerts requesting backup. This powerful combination allows ResQSquad to establish a user management system, ensure data persistence through a database, and facilitate real-time communication between rescue teams and authorized agencies, ultimately promoting a more coordinated and efficient response during critical disaster situations.

3.3.3 DATABASE

While PostgreSQL plays a crucial role in ResQSquad's back-end, it's important to clarify that PostgreSQL is not directly related to the application's core functionality. Postgre is a separate mail filtering software, and it's not typically used in the context of Django web applications. The Python-powered back-end of ResQSquad utilizes the robust Django framework. This framework streamlines development by providing functionalities like user management, database interaction, and an administrative interface. This allows the project to focus on core functionalities like real-time communication and location-based features. ResQSquad leverages PostgreSQL, a powerful open-source database management system, for secure and reliable data storage. PostgreSQL's features like data integrity and scalability ensure the critical information ResQSquad uses remains secure and accessible. Together, Python and PostgreSQL create a robust foundation for resQSquad, enabling efficient data management and supporting the smooth operation of this communication platform for disaster response.

Chapter-4

SAMPLE CODE

1 FRONT END

HOME.html

```
{% extends 'base.html' %} {% block main %} {% load static %}
  <div class="h-[60dvh] overflow-hidden bg-base-300 relative">
    <img
      src="{% static 'img/rescue_bg_1.jpg' %}"
      alt="Rescue Background"
      class="opacity-20 absolute top-0 h-[60dvh] w-full sm:h-auto pointer-events-none" />
    <div class="flex flex-col sm:ml-8 mx-2 items-center sm:items-start text-center sm:text-start justify-center h-full">
      <h1 class="text-4xl font-bold z-20">ResQSquad</h1>
      <p class="text-xl z-20">Uniting Heroes, Saving Lives – One Umbrella, Infinite Hope.</p>
      <a class="btn btn-accent mt-4 w-full max-w-md z-20 opacity-80" href="{% url "user_report" %}">In Need of Help?</a>
    </div>
  </div>


  <div class="flex flex-col items-center justify-center my-8 py-12">
    <h2 class="text-3xl font-bold text-center my-4">Our Mission</h2>
    <p class="text-[min(1.25rem,3.5vw)] text-center max-w-4xl mt-4 px-4 mb-8 leading-relaxed">
      Our goal at <span class="font-bold">ResQSquad</span> is to bring together the unsung heroes of different
      rescue organizations under one strong roof, becoming a formidable force committed to offering quick, dependable,
      and life-saving reactions during emergencies. When a disaster happens, we think that relief and assistance
      should be given right away. <span class="underline">Every moment counts</span> because lives are on the line. We
```

have therefore set out to coordinate the operations of the rescue organizations, bringing them all together to

create a strong network of first responders. Our organization acts as the

lighthouse that directs various organizations, making sure they are organized

and prepared to effectively manage any emergency crisis.

</p>

</div>

<div class="flex flex-col items-center justify-center bg-base-200 my-12 py-16">

<h2 class="text-3xl font-bold text-center my-4">Agencies We Work With</h2>

<p class="text-xl text-center my-4 leading-relaxed mx-8 max-w-5xl">

We work with a number of agencies to ensure that we are able to provide the best possible service to our

customers. We are always looking to expand our network of agencies and are open to working with new agencies.

</p>

<div class="grid grid-cols-2 md:grid-cols-4 gap-8 mt-8">

<div class="flex flex-col items-center justify-center">

<img

src="{% static 'img/agency_1.png' %}"

alt="Agency 1"

class="w-32 h-32 object-cover"

data-aos="flip-up" />

<p class="text-xl mt-4">NDMA</p>

</div>

<div class="flex flex-col items-center justify-center">

<img

src="{% static 'img/agency_2.png' %}"

alt="Agency 2"

class="w-32 h-32 object-cover"

data-aos="flip-up" />

<p class="text-xl mt-4">NDRF</p>

```
      </div>
      <div class="flex flex-col items-center justify-center">
        <img src="{% static 'img/agency_3.png' %}" alt="Agency 3" class="w-32 h-32" data-aos="flip-up" />
        <p class="text-xl mt-4">NIDM</p>
      </div>
      <div class="flex flex-col items-center justify-center">
        <img src="{% static 'img/agency_4.png' %}" alt="Agency 4" class="w-32 h-32" data-aos="flip-up" />
        <p class="text-xl mt-4">CDRI</p>
      </div>
    </div>
  </div>


  <div class="flex w-full flex-col items-center justify-center my-12 py-12">
    <h2 class="text-3xl font-bold text-center mb-4">Join Us</h2>
    <p
      class="text-xl text-center px-8 italic leading-relaxed max-w-5xl"
      data-aos="fade-up"
      data-aos-anchor-placement="top-bottom"
      data-aos-duration="1000"
      data-aos-easing="ease-in-out">
      "Alone, we can do so little; together, we can do so much." — Helen Keller
    </p>
    <p class="text-xl text-center px-8 my-4 mb-8 leading-relaxed max-w-5xl">
      We are always looking to expand our network of agencies and are open to working with new agencies. If you are
      interested in working with us, please register your agency by clicking the button below.
    </p>
    <div class="w-full px-8 flex items-center justify-center">
      <a href="{% url "account_signup" %}" class="w-full max-w-md"><button class="btn btn-accent w-full max-w-md">Register Your Agency</button></a>
    </div>
```

```
    </div>

{% comment %} Footer {% endcomment %}
  <footer class="bg-base-300 px-8 py-12">
    <div class="grid grid-cols-1 text-center md:text-start md:grid-cols-4 gap-8">
        <h3 class="text-2xl font-bold whitespace-nowrap row-span-3 my-auto
md:my-0 md:row-span-1">ResQSquad</h3>
      <ul
        class="flex flex-col justify-start items-center md:items-start text-gray-500
text-sm space-y-2 whitespace-nowrap">
        <li>
          <a class="hover:underline" href="{% url 'about' %}">About Us</a>
        </li>
        <li>
          <a
            href="https://github.com/RescueRadar"
            class="hover:underline"
            target="_blank"
            rel="noopener noreferrer"
          >The Team</a
          >
        </li>
        <li>
            <a class="hover:underline" href="{% url 'contact' %}">Need
Help?</a>
        </li>
      </ul>
      <ul
        class="flex flex-col justify-start items-center md:items-start text-gray-
500 text-sm space-y-2 whitespace-nowrap">
        <li>
          <i class="fa-brands fa-facebook"></i>
          <a href="#" class="hover:underline">Facebook</a>
        </li>
```

```
          <li>
            <i class="fa-brands fa-x-twitter"></i>
            <a href="#" class="hover:underline">Twitter</a>
          </li>
          <li>
            <i class="fa-brands fa-instagram"></i>
            <a href="#" class="hover:underline">Instagram</a>
          </li>
          <li>
            <i class="fa-brands fa-linkedin"></i>
            <a href="#" class="hover:underline">LinkedIn</a>
          </li>
        </ul>
        <ul
          class="flex flex-col justify-start items-center md:items-start text-gray-
500 text-sm space-y-2 whitespace-nowrap">
          <li>
            <a href="#" class="hover:underline">Privacy Policy</a>
          </li>
          <li>
            <a href="#" class="hover:underline">Terms of Service</a>
          </li>
        </ul>
      </div>
      <p class="md:text-start text-center text-gray-500 text-xs mt-8">&copy;
2023 ResQSquad. All rights reserved.</p>
    </footer>

{% endblock main %}

          - 50-60 people affected
        </p>
      </div>
      <div class="text-sm">
```

```
<p class="mb-1">
  <i class="fa-solid fa-user mr-2"></i>
  Ankit Singh
</p>
<p class="mb-1">
  <i class="fa-solid fa-phone mr-2"></i>
  9638527410
</p>
<p>
  <i class="fa-solid fa-map-marker mr-2"></i>
  Chennai, Tamil Nadu
</p>
      </div>
    </div>


    <div
      class="bg-base-200 rounded-xl shadow p-6 hover:shadow-lg border-base-
200">
      <div class="flex mb-2 items-center">
        <h3 class="font-bold text-xl">
          Flood
        </h3>
        <p class="text-sm">
          - 10-20 people affected
        </p>
      </div>
      <div class="text-sm">
        <p class="mb-1">
          <i class="fa-solid fa-user mr-2"></i>
          Ankit Kumar
        </p>
        <p class="mb-1">
          <i class="fa-solid fa-phone mr-2"></i>
          9638527410
```

```
            </p>
            <p>
              <i class="fa-solid fa-map-marker mr-2"></i>
              Goa City, Goa
            </p>
          </div>
        </div>
      </div>
    </div>


    {% endblock %}
```

4.2 **BACKEND**

Views.py

```python
from django.shortcuts import render, redirect, get_object_or_404
from django.http import HttpResponse
import requests
from django.http import JsonResponse
from agency.models import agency, non_approved_agency,Post
from django.contrib import messages
from django.contrib.auth.decorators import login_required
from .decorators import agency_req
from .forms import ContactForm, PostForm
from django.core.mail import send_mail
import os
from django.template.loader import render_to_string
from django.core.mail import EmailMessage
import folium
from django.db.models import Q


# Create your views here.
def home(request):
    if request.user.is_authenticated:
        if agency.objects.filter(user=request.user).exists():
```

```python
        return redirect('dashboard')
    return render(request, 'home.html')


def about(request):
    return render(request, 'about.html')


def contact(request):
    if request.method == 'POST':
        form = ContactForm(request.POST)
        if form.is_valid():
            first_name = form.cleaned_data['first_name']
            last_name = form.cleaned_data['last_name']
            email = form.cleaned_data['email']
            message = form.cleaned_data['message']

            EmailMessage(
                'New contact form submission',
                render_to_string('email/contact_form.html', {
                    'first_name': first_name,
                    'last_name': last_name,
                    'email': email,
                    'message': message
                }),
                os.environ.get('EMAIL_HOST_USER'),
                [os.environ.get('EMAIL_HOST_USER')],
                reply_to=[email]
            ).send()

            return redirect('home')
        else:
            print(form.errors)

    form = ContactForm()
    return render(request, 'contact.html',{
```

```python
            'form': form
        })
def user_report(request):
    return render(request, 'user_report.html')


def error_message(request, error_message):
    return render(request, 'error_message.html', {'error_message': error_message})


def profileEdit(request):
    return render(request, 'profileEdit.html')


@login_required
@agency_req
def victims_portal(request):
    return render(request, 'victims_portal.html')


@login_required
@agency_req
def rooms(request):
    return render(request, 'rooms.html')


@login_required
@agency_req
def chatroom(request):
    return render(request, 'chatroom.html')


@login_required
def agencyPage(request):
    return render(request, 'agencyPage.html')


@login_required
@agency_req
def dashboard(request):
    q=request.GET.get('q') if request.GET.get('q') != None else ''
```

```python
    posts = Post.objects.filter(
            Q(title__icontains=q) |
            Q(content__icontains=q)
        )
    return render(request, 'dashboard.html', {'posts': posts})


@login_required
def user_profile(request):
    return render(request, 'user_profile.html')


def profile(request):
    form_submitted = False
    if request.method == "POST":
        name = request.POST.get('name')
        address = request.POST.get('address')
        phone = request.POST.get('phone')
        email = request.user.email
        website = request.POST.get('website')
        about = request.POST.get('about')
        city = request.POST.get('city')
        state = request.POST.get('state')
        manpower = request.POST.get('manpower')
        volunteers = request.POST.get('volunteers')

        form_submitted=True

        if not all([name, address, phone, website, about, city, state, manpower,
volunteers]):
            messages.error(request, "All fields must be filled.")
            return redirect('profile')


        non_approved_agency.objects.create(
            user=request.user,
```

```python
            name=name,
            address=address,
            phone=phone,
            email=email,
            website=website,
            about=about,
            city=city,
            state=state,
            manpower=manpower,
            volunteers=volunteers
        )
        messages.success(request, "Profile information saved successfully.")
        form_submitted=True
        return redirect('request_submitted', form_submitted=str(form_submitted))  #
Convert to string
    else:
        if non_approved_agency.objects.filter(user=request.user).exists():
            return redirect('request_submitted', form_submitted=form_submitted)
        elif agency.objects.filter(user=request.user).exists():
            return redirect('home')
        return render(request, 'profile.html', {'form_submitted': form_submitted})


def request_submitted(request, form_submitted):
        return render(request, 'request_submitted.html', {'form_submitted':
form_submitted})


@login_required
def all_agencies(request):
    all_agencies = agency.objects.all()
    return render(request, 'all_agencies.html', {'all_agencies': all_agencies})


@login_required
def post_detail(request, pk):
    post = get_object_or_404(Post, pk=pk)
```

```python
    return render(request, 'post_detail.html', {'post': post})


@login_required
@agency_req
def post_new(request):
    if request.method == "POST":
        form = PostForm(request.POST)
        if form.is_valid():
            post = form.save(commit=False)
            post.agency = agency.objects.get(user=request.user)
            post.save()
            return redirect('post_detail', pk=post.pk)
    else:
        form = PostForm()
    return render(request, 'post_edit.html', {'form': form})


@login_required
@agency_req
def post_edit(request, pk):
    post = get_object_or_404(Post, pk=pk)
    if post.agency.user != request.user:
        return error_message(request, 'You are not allowed to edit other agency posts!')
    if request.method == "POST":
        form = PostForm(request.POST, instance=post)
        if form.is_valid():
            post = form.save(commit=False)
            post.save()
            return redirect('post_detail', pk=post.pk)
    else:
        form = PostForm(instance=post)
    return render(request, 'post_edit.html', {'form': form})
```

Chapter-5

User Interface

5.1 **Screen Shot**

Home Page
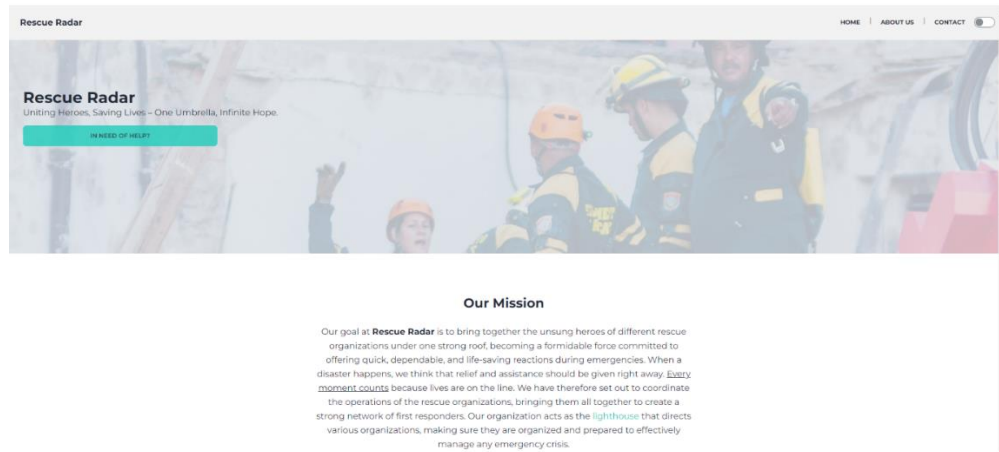
Emergency Situation report

Sign-Up page(For agency)

Agency Registration

Sign-in Page

Sign in

Username:

Username

Password:

Password

LOGIN

Forgot Password / Sign Up

Dashboard

resQSquad   HOME | ABOUT US | CONTACT | PROFILE

- DASHBOARD
- ALL AGENCIES
- ROOMS
- VICTIMS PORTAL
- MAPS PORTAL
- SAMPLE IPINFO

Search your feed...

POST

Emergency 1
- CodXThor
The team is deployed in Indra Nagar

Status
Location
Manpower

Emergency 2
- AyushSingh
Team is deployed in Mysore

Status
Location
Manpower

Emergency 3
- AnkitKumar
team is deployed in Mumbai

Status
Location
Manpower

Victim Portal

All Agency

Agency Location

Chapter-6

Conclusion

A Lifeline for Disaster Response ResQSquad emerges as a beacon of hope in the chaotic aftermath of disasters. This mini-application, built with Python and Django on the back end and utilizing HTML and Django templates on the front end, strives to revolutionize communication during critical response efforts.

By establishing a real-time communication platform, ResQSquad empowers rescue teams to swiftly request backup and coordinate resource deployment with authorized agencies. This streamlined communication eliminates delays and fosters a unified response, ultimately saving lives in those precious moments when every second counts.

Chapter-7

Looking Ahead:

While this initial project serves as a solid foundation, future iterations of ResQSquad can be further enhanced by:

1. **Integrating real-time mapping and location sharing:** This would allow for a more comprehensive situational awareness for both rescue teams and coordinating agencies.

2. **Developing mobile app functionality:** Extending ResQSquad to mobile devices would provide greater accessibility and convenience for rescue teams on the ground.

3. **Enhancing security measures:** Implementing robust security protocols will be crucial as the platform scales and handles increasingly sensitive data.

ResQSquad holds immense potential to transform disaster response by bridging the communication gap and fostering a more coordinated effort. With ongoing development and collaboration, this mini-application can evolve into a powerful tool, saving lives and minimizing damage in the face of unforeseen disasters.

Chapter-8

References

|	
Django documentation	https://docs.djangoproject.com/en/5.0/
Bootstrap	https://getbootstrap.com/docs/4.1/getting-started/introduction/
GitHub	https://docs.github.com/en
Ipinfo	https://ipinfo.info/documentation
Stackoverflow	https://stackoverflow.com/documentation/
Python Documentation	https://www.python.org/doc/
PostgreSQL	https://www.postgresql.org/docs/

Chapter-9

Plagiarism Report

DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

Author Name	Ankit Singh
Title	resQSquad
Paper/Submission ID	2188089
Submitted by	virupaksha.msc12@gmail.com
Submission Date	2024-07-31 15:01:08
Total Pages, Total Words	30, 3201
Document type	Project Work

Result Information

Similarity   **3 %**

Sources Type

Student Paper 3.0%

Report Content

Quotes 0.12%

Words < 14, 0.91%

Ref/Bib 0.75%
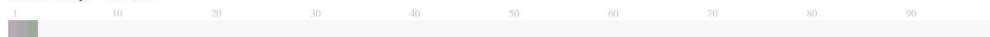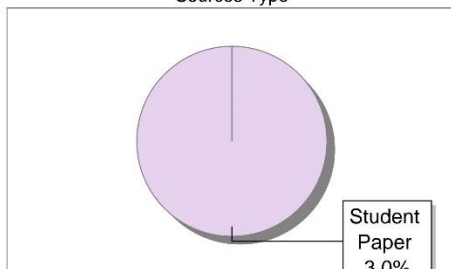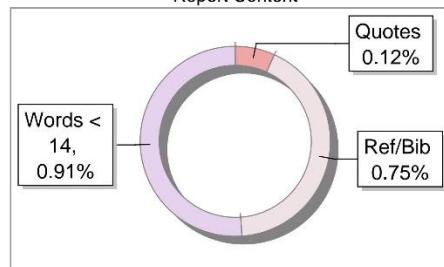
Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

			A-Satisfactory (0-10%)
3	**3**	**A**	B-Upgrade (11-40%)
			C-Poor (41-60%)
SIMILARITY %	MATCHED SOURCES	GRADE	D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	Submitted to Visvesvaraya Technological University, Belagavi	1	Student Paper
2	Submitted to Visvesvaraya Technological University, Belagavi	1	Student Paper
3	Submitted to Visvesvaraya Technological University, Belagavi	1	Student Paper