CLOUD APPLICATION DEVELOPMENT

CONTAINMENT ZONE ALERTING APPLICATION

A MINI-PROJECT REPORT

Submitted by

DHANASHEKAR B - AC19UCS022

DHANUSH T - AC19UCS023

HARISH KUMAR V - AC19UCS034

KALAIYARASAN N - AC19UCS045

In partial fulfilment of the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

ADHIYAMAAN COLLEGE OF ENGINEERING

DR.M.G.R NAGAR, HOSUR-635130

ANNA UNIVERSITY: CHENNAI 600025

NOVEMBER 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this mini-project report **"CONTAINMENT ZONE ALERTING APPLICATION"** is the bonafide work of **"DHANASHEKAR B (AC19UCS022), DHANUSH T (AC19UCS023), HARISH KUMAR V(AC19UCS034), KALAIYARASAN N(AC19UCS045)"**who carried out the project under my supervision.

SIGNATURE

Dr. G. FATHIMA, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR,

Department of CSE,

Adhiyamaan College of Engineering,

(Autonomous)

Dr. M.G.R. Nagar,

Hosur - 635 130

SIGNATURE

MR. HARISHANKAR A , M.E

ASSISTANT

PROFESSOR,

Department of CSE,

Adhiyamaan College of Engineering,

(Autonomous)

Dr. M.G.R. Nagar,

Hosur – 635 130

Submitted for the Mini project VIVA-VOICE Examination held on …… at **Adhiyamaan College of Engineering (Autonomous), Hosur – 635 130**.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

It is one of the most efficient tasks in life to choose the appropriate words to express one's gratitude to the beneficiaries. We are very much grateful to God who helped us through the project and molded us into what we are today.

We are grateful to our beloved Principal **Dr. G. RANGANATH, M.E., Ph.D.,** Adhiyamaan College of Engineering (Autonomous), Hosur for providing the opportunity to do this work on the premises.

We acknowledge our heartful gratitude to **Dr. G. FATHIMA, M.E., Ph.D.,** Professor, and Head of the Department, Department of Computer Science and Engineering, Adhiyamaan College of Engineering (Autonomous), Hosur, for her guidance and valuable suggestions and encouragement throughout this project.

We are highly indebted to **Dr. B. GOPINATHAN, M.E., Ph.D.,** Professor, Department of Computer Science and Engineering, Adhiyamaan College of Engineering (Autonomous), Hosur, whose immense support, encouragement, and valuable guidance made us complete this project successfully.

We also extend our thanks to the Project Coordinator and all Staff Members for their support to complete this project successfully.

Finally, we could like to thank our parents, without their motivation and support would not have been possible for us to complete this project successfully.

CONTENT

1. INTRODUCTION

1.1 Project Overview

This paper mainly focuses on development of an Android application which can inform people of the COVID-19 containment zones and prevent trespassing into these zones. This Android application updates the locations of the areas in a Google map which are identified to be the containment zones. The application also notifies the users if they have entered a containment zone and uploads the user's IMEI number to the online database. With this IMEI number, the police can keep an eye on the people who are frequently violating the lockdown rules. Therefore, this application can be used as a tool for creating further social awareness about the arising need of precautionary measures to be taken by the people of India.

1.2 Purpose

The Android application shows the location of the containment zones to the users. It also notifies the user when he or she trespasses the boundary of a Containment Zone or stays in the containment zones. All these functionalities are achieved by the help of Firebase and Geo fencing tools from Google.

2. LITERATURE SURVEY

2.1 Existing Problem

It is Indian COVID-19 tracking mobile application. It is developed by the National Informatics Centre which comes under the Ministry of Electronics and Information Technology. To spread awareness of COVID-19. To connect essential COVID-19 – related health services to the people of India. It uses the smartphone's GPS and Bluetooth features to track the coronavirus infection.

Disadvantages

- The closed source architecture of the app violates transparency principles.
- There are no safeguards against data theft and other breaches.
- The app is useless for the low-income non-smartphone users. There exists no alternative for them as well

2.2 References

1. Development of an Android Application for viewing covid-19 containment zones and monitoring:

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7328652/

2. Aarogya Setu(COVID-19):

https://play.google.com/store/apps/details?id=nic.goi.aarogyasetu&hl=en

3. Application for covid-19 real time counter:

https://www.ijres.org/papers/Volume-10/Issue-5/Ser-10/B10050918.pdf

4. Tracking the covid zone through Geo-fencing technique:

https://www.emerald.com/insight/content/doi/10.1108/IJPCC-06-2020-0057/full/html

2.3 Problem Statement Definition

Cloud based discourse for Containment zone alerting

- This application is intended to provide information about containment zones in a particular region by alerting people, through continuous monitoring of an individual's location.
- Key benefits of the application are monitoring people's activity and alerting them of their safety movements.
- Based on the location a Geo fence will be created within a 100 m radius. They should be able to see how many people are visiting that zone.
- After the user logged into the app it will track the user's location and update the database with the current location. If the user is visiting the containment zone he will get an alert notification.

Problem Statement (PS)	I am	I'm trying to	But	Because	Which makes me feel
PS-1	Patient	Get medicines from hospital and medical.	I couldn't get medicine	Due to hospital located area is containment zone.	Severe illness and tiredness.
PS-2	Patient	Buy disease related medicine	I am not aware of buying medicine without doctor prescription.	Various fertilizers are available for a single plant.	Dependent

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users.

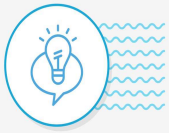
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

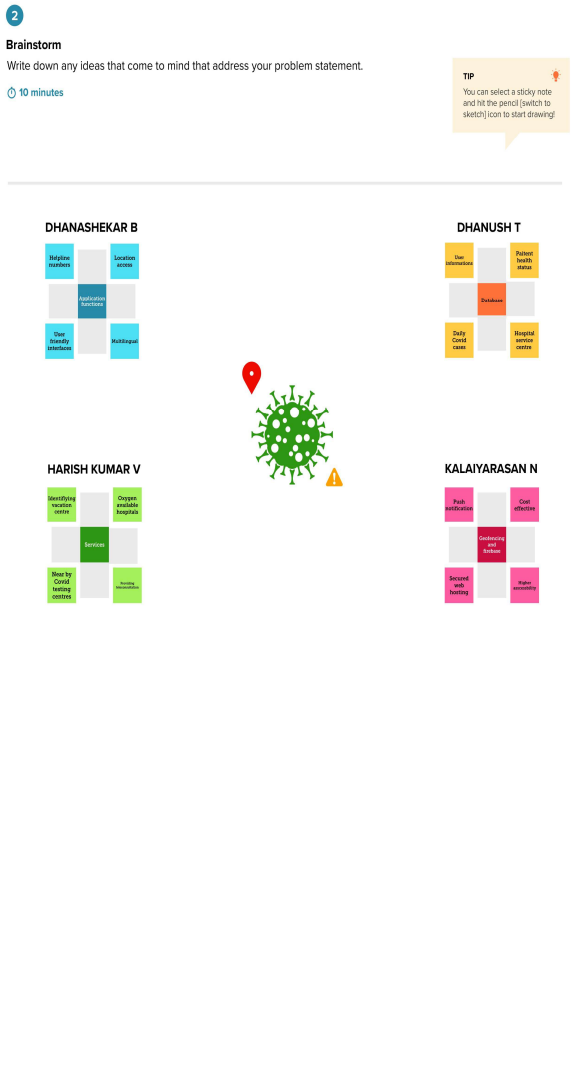
Mike is travelling to certain location and he wants to know a containment zones and updates of covid cases

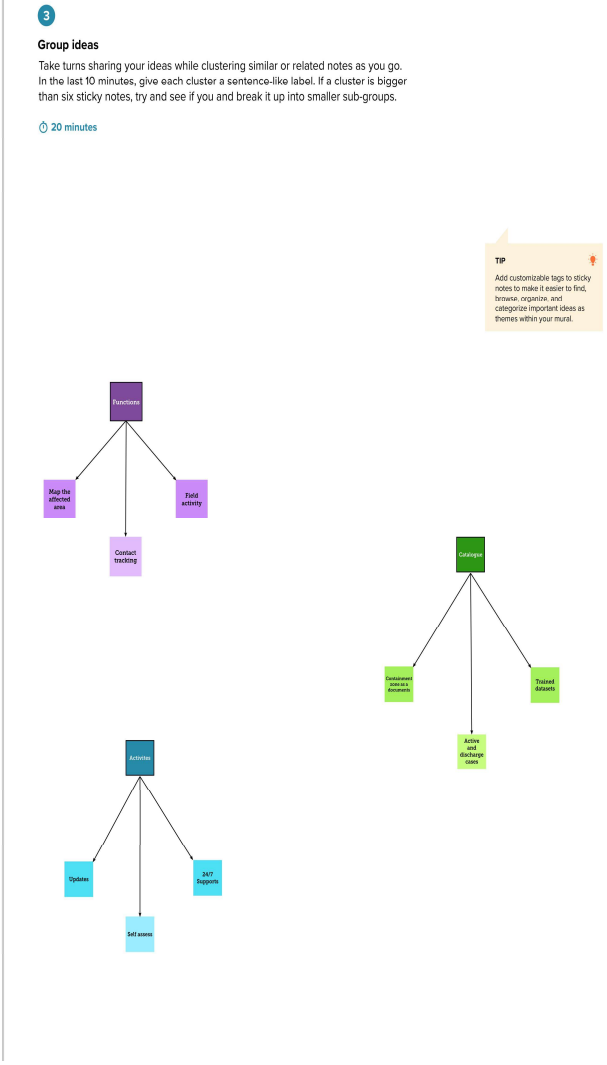
Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping





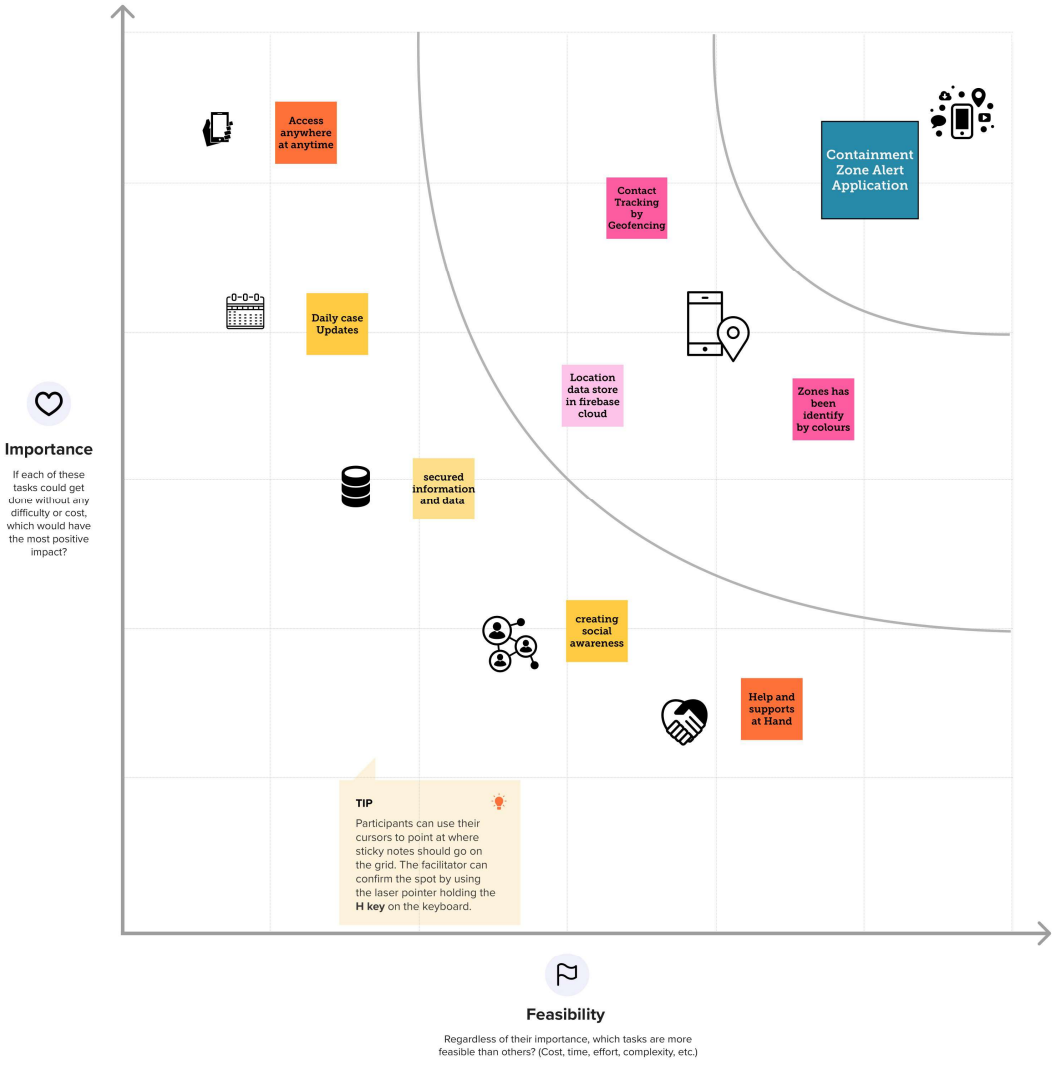
Step-3: Idea Prioritization

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



3.3 Proposed Solution

S. No	Parameter	Description
1	Problem Statement (Problem to be solved)	Containment Zone Alerting User is travelling to certain location and he wants to know a containment zones and updates of Covid cases.
2	Idea / Solution description	The proposed solution that classifies containment zone using Geo fencing technology and getting daily covid updates based on state and district cases.
3	Novelty / Uniqueness	For detecting the containment zone an API is used (Geo fencing and Firebase) which very efficient than other methods. When we need to create a user friendly for creating a social awareness.
4	Social Impact / Customer Satisfaction	By this application the user can get updates on covid cases and containment zone alerting through SMS. So that they can plan their day accordingly.
5	Business Model (Revenue Model)	• Input module • Data set training module • Automated alert message • Location access module • Containment zone level estimation API module • Suggestion module
6	Scalability of the Solution	The accuracy of the result for the covid cases and zone alerting notification is 99%. We can design user friendly interface to use all kind of users.

3.4 Problem Solution Fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioural patterns and recognize what would work
and why

Purpose:

- Solve complex problems in a way that fits the state of your customers.
- Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behaviour.
- Sharpen your communication and marketing strategy with the right triggers and messaging.
- Increase touch-points with your company by finding the right problem-behaviour fit and building trust by solving frequent annoyances, or urgent or costly problems.
- Understand the existing situation in order to improve it for your target group.

Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) — CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

People or user wants to travel the other state or district during pandemic time

6. CUSTOMER CONSTRAINTS — CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Due to lack of technology awareness ,they go away from the using of current technology

5. AVAILABLE SOLUTIONS — AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

- Automatic Notification and Alarms for Individual

-In past ,they Identified the number of cases are affected by Covid-19 in a certain Area

-Pros &Cons They easily identified the zones by using Individual location tracking

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS — J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

- People are unware of travelling the other area during pandemic situation

- People don't know about the safety measure of the Containment zones

9. PROBLEM ROOT CAUSE — RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

- Users without have enough awareness of covid19 pandemic

- User not aware of containment zones of covid-19

7. BEHAVIOUR — BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

To determine the zone characteristics and identify the zones ,Then form different containment zone depends on effect.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS — TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

safe precautions for people to be aware in pandemic

4. EMOTIONS: BEFORE / AFTER — EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

If they faced a problem ,they could use our technology to aware off pandemic containment zones

10. YOUR SOLUTION — SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Low cost cloud Based Application device that can be easily provides the Notification and Alarms based on users enter in containment zones

8. CHANNELS of BEHAVIOUR — CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

Promoting through social media ,with the help of social media Entrepreneurs

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Identification of the containment zone and take precautionary actions

Extract online & offline CH of BE

Problem-Solution it canvas is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 license
Created by Daria Nepriakhina / Amaltama.com

AMALTAMA

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

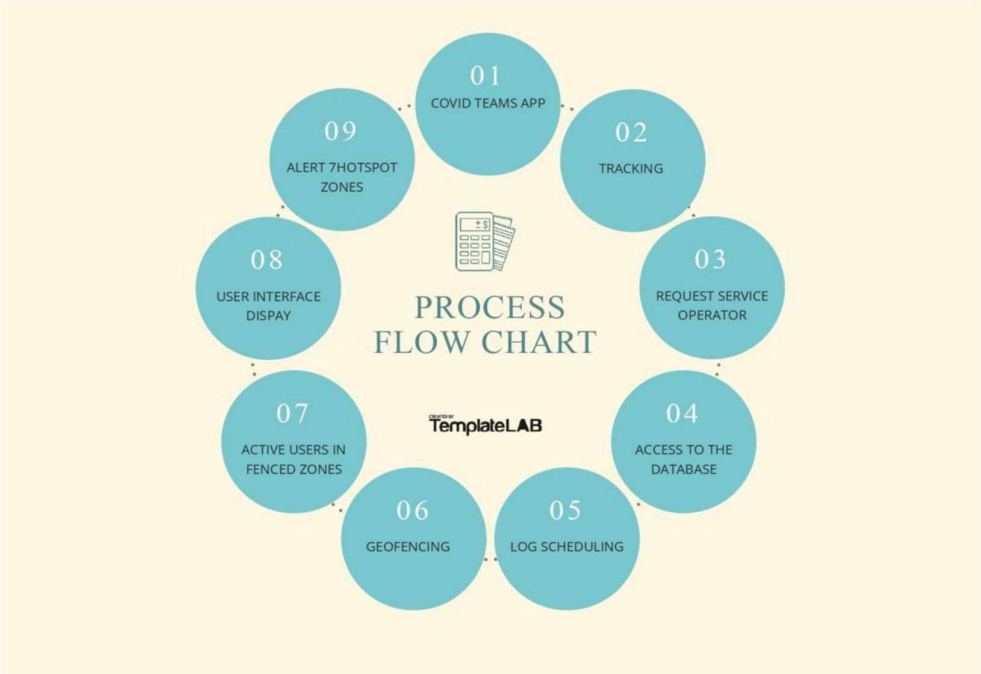
S. NO	Functional Requirement (Epic)	Sub Requirement (Story/Sub-Task)
1	Admin Login	• Separate Portal for Administrator to login and make use of.
2	Admin Dashboard	• A responsive dashboard that shows all statistics of the app and gives the admin full control over the application. • Controls to perform CRUD operations on the application's database.
3	User Registration	• A simple and clear interface for user interface that requires the necessary details to be input by the user. • All fields should be validated according to the required detail.
4	User Login	• One time login, required whence installing using the app for the first time.
5	Primary Specifics	• Highlight of containment zones in a map interface is to be shown to the user. • Listing of details of Containment zones. • Instant notification delivery on user-entry into a containment zone.
6	Additional Features	• COVID statistics to be shown to the user. • Covid news from NEWSAPI

4.2 Non-Functional Requirements

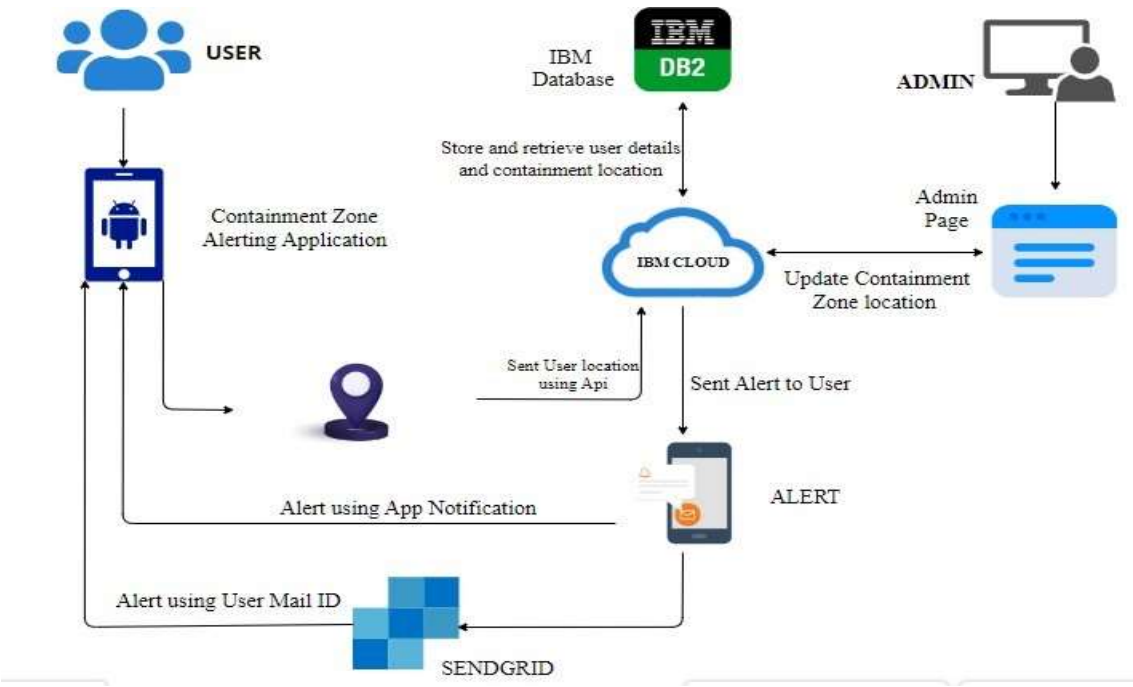
S.NO	Non-Functional Requirement(Epic)	Sub Requirement (Story/Sub-Task)
1	Usability	• The application should provide a good UI/UX design with seamless navigation across sections of the app.
2	Security	• Passwords should not be stored by the application. • The passwords must be hashed and only the hashed values should be stored in the database. • Privacy and security of the user must be maintained.
3	Reliability	• The data provided by the application should be reliable. • At any point, data corruption should not exist.
4	Performance	• The system's performance should not be affected by a change in the number of users.
5	Availability	• The application and its resources must be available at all times.
6	Scalability	• The application must be extendable to a larger scale of users.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Covid team app)	Registration	USN-1	As a user, I can register for the application by entering my email address, password, and password confirmation.	I can access my account/dash board.	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once I have registered for the application.	I can receive a confirmation email & click confirm.	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook.	I can register & access the dashboard with Facebook Login.	Low	Sprint-4
		USN-4	As a user, I can register for the application through Gmail.	I can register & access the dashboard with Google Login.	Medium	Sprint-1
		USN-5	As a user, I can register for the application through Twitter.	I can register & access the dashboard with Twitter Login.	Low	Sprint-4
	Login	USN-6	As a user, I can log into the application by entering my email & password.	I can access it whenever I want its access.	High	Sprint-1
	Dashboard	USN-7	As a user, I need to give permission to access My Contacts, Location, and Storage.	I get access to their services.	High	Sprint-2

		USN-8	As a user, I get access to the dashboard which shows a map with marked zones.	I can see the zone information on the dashboard.	High	Sprint-2
Hospitals Administrator	Registration	USN-9	As a management, I need to register my hospitals on the site.	I can see the registered hospital in the hospital dashboard.	High	Sprint-1
	Login	USN-10	As a management, I need to login into my dashboard with my given hospital id and password.	I can see my dashboard after login.	Medium	Sprint-1
	Dashboard	USN-11	As a management, I need to enter the case information of the patient that visits our hospital.	I can view the patient information on the dashboard.	High	Sprint-2
		USN-12	As a management, I need to store all the patient information on the cloud.	-	High	Sprint-3
Administrator	Services	USN-13	As an admin, I need to provide valid information about the pandemic out there.	I can get the pandemic updates out there.	High	Sprint-2
		USN-14	As an admin, I need to provide medical advice through a chat bot.	I get medicinal recommendations through a chat bot.	Medium	Sprint-3
		USN-15	As an admin, I need to provide medical recommendations by collaborating with top hospitals.	I get medical instruction through chief doctors.	Low	Sprint-3

		USN-16	As an admin, I need to alert the user when they enter pandemic zones.	I got a notification when I am in the pandemic area.	Medium	Sprint-4
		USN-17	As an admin, I need to provide preventive measures when they travel through it.	I got a remedies notification when I am in the pandemic area.	High	Sprint-3
		USN-18	As an admin, I need to provide special services for premium users by giving services like monitoring health by their smart bands.	I was treated special after becoming a premium member.	Low	Sprint 4
	Data collections	USN-19	As an admin, I need to store all the user information on the cloud.	I can access my information when I needed.	Medium	Sprint-4
		USN-20	As an admin, I need to collect the list of viruses & bacteria present in this world.	-	Medium	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

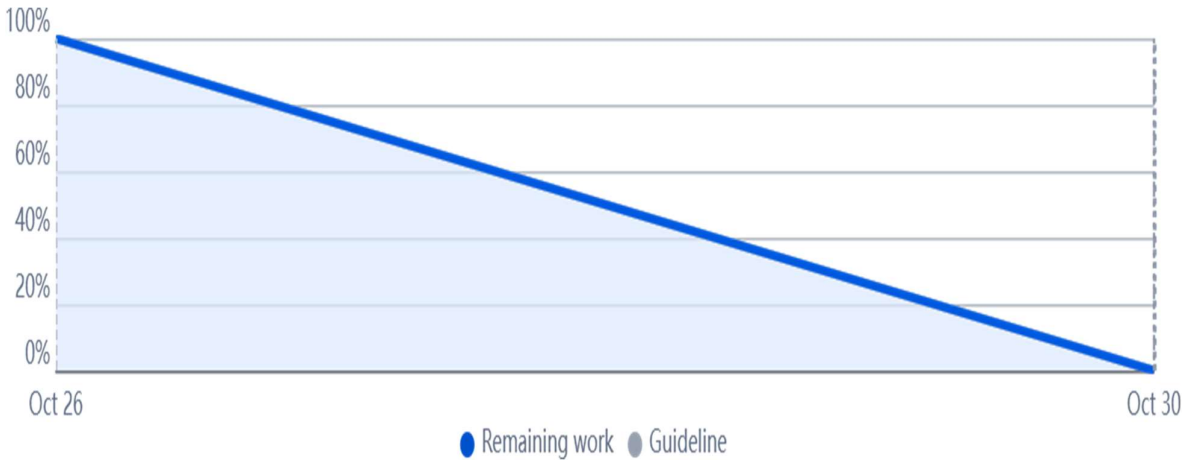
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	User: I can register by providing information like username, email, password ,mobile number.	3	High	Dhanashekar
		USN-2	User: Once I have registered ,I will receive the verification mail through the given mail id.	2	High	Dhanush
		USN-3	Management: we need to register hospitals available near to their surroundings.	2	High	Kalaiyarasan
	Login	USN-4	User: I can login to the application by entering my username, email & password	3	High	Harish Kumar
		USN-5	Management: I may store the Personal information of User in the cloud.	5	Medium	Dhanashekar
	Dashboard	USN-6	User: I have to give permission to access my Location	5	High	Dhanush

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		USN-7	User: After Login Page the dashboard which shows a map with containment zones will appear.	5	High	Harish Kumar
		USN-8	Management: I have to update the daily cases based on the information Provide by government.	5	High	Kalaiyarasan
	Services	USN-9	Admin: I need to provide valid information about the pandemic situation.	5	High	Dhanush
Sprint-3	Dashboard	USN-10	Management: It is necessary to store information of user in cloud for safety measures.	5	High	Dhanshekar
	Services	USN-11	Admin: I can provide medical Advice and suggestions through a chatbot.	5	Medium	Kalaiyarasan
		USN-12	Admin: I need to provide medical advices to the patients based on the consultation provided by doctors.	5	Low	Harish kumar
		USN-13	Admin: I need to provide precautionary measures for the users.	5	High	Dhanashekar
	Services	USN-14	Admin: I have to alter the trespasser by sending notification about 500meter before the containment zones.	3	Medium	Dhanush
	Data Collection	USN-15	Admin: I need to store all the user information on the cloud	5	Medium	Kalaiyarasan
		USN-16	Admin: I need to collect the recent List of symptoms, death total and daily affected cases.	5	Low	Harish Kumar

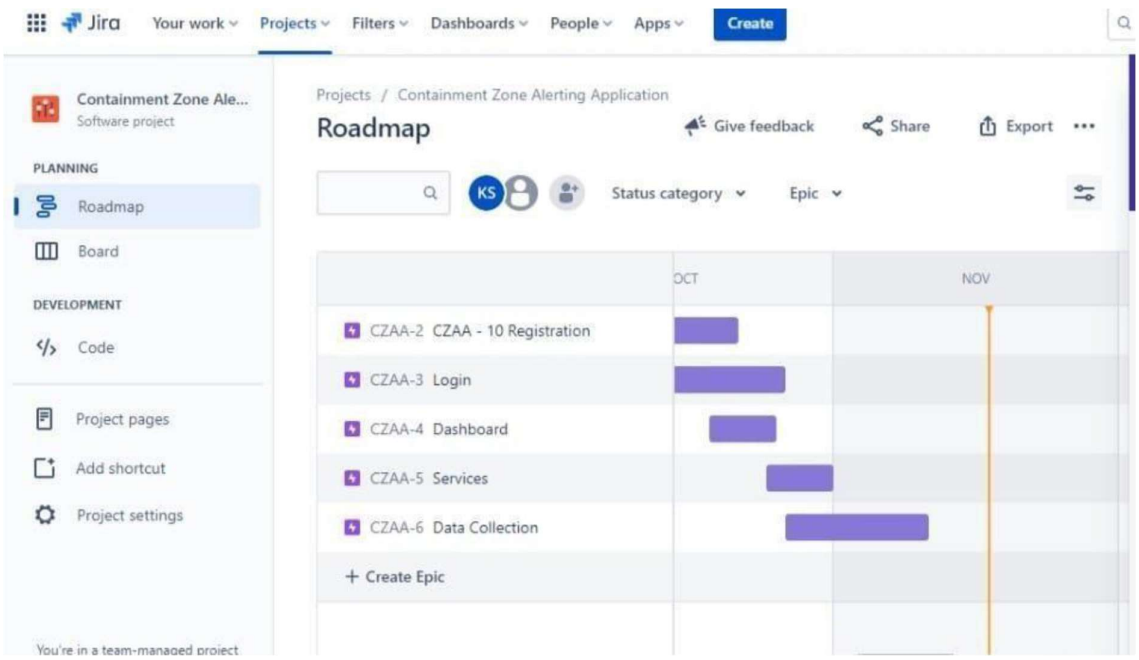
6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$



6.3 Reports from JIRA



7. CODING & SOLUTIONING

7.1 Features 1

In this page, the admin can register, login, remove zone and add zone.

CODING:

Login:

```
<html lang="en" xmlns="http://www.w3.org/1999/html">
<head>
<! DOCTYPE html>
<meta charset="UTF-8">
<title>Login</title>
<link rel="stylesheet" href="../static/adminlogin.css">
</head>
<body>
<video src="../static/video/background.mp4" autoplay loop playsinline
muted></video> <div class="box">
```

```
<form action="/login" method="POST" autocomplete="off">
<h2>LOGIN</h2>
<div class="inputBox">
<input type="text" name="email" id="email" required="required">
<span>Email</span>
<i></i>
</div>
<small></small>
<div class="inputBox">
<input type="password" required="required" name="password"
id="password"> <span>Password</span>
<i></i>
</div>
<small></small>
<div class="links">
<a href="#">Forgot Password ?</a>
<a href="{{url_for('adminRegistration')}}"><b>REGISTER</b></a>
</div>
<input type="submit" value="Login">
<h4 class="text-danger" id="pass-warning" style="color: white"></h4>
</form>
</div>
<script>document.getElementById("pass-warning").innerHTML =
'{{data}}';</script> <script src="../static/js/adminlogin.js">
</script>
</body>
</html>
```

Register:

```
<! DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/html">
<head>
<meta charset="UTF-8">
<title>Registration</title>
<link rel="stylesheet" href="../static/adminreg.css">
<link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css"
integrity="sha384-
AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJvOvKxVfELGr
oGkvsg+p" crossorigin="anonymous"/>
</head>
<body>
```

```
<video src="../static/video/background.mp4" autoplay loop playsinline
muted></video>
<form action="/" method="POST">
<div class="user">
<i class="fas fa-user"></i>
<input type="text" placeholder="Enter your username" name="username"
id="username"> <small></small>
</div>
<div class="user">
<i class="fas fa-user"></i>
<input type="email" placeholder="Enter your mailid" name="email"
id="email"> <small></small>
</div>
<div class="password">
<i class="fas fa-lock"></i>
<input type="password" placeholder="Password" name="password"
id="password">
<br>
<small></small>
</div>
<div class="password">
<i class="fas fa-lock"></i>
<input type="text" placeholder="Confirm password"
name="confirm_password" id="confirm_password">
<br>
<small></small>
</div>
<div class="button">
<button class="button" type="submit"><b>REGISTER</b></button>
<button><a href="/login">LOGIN</a></button> </div>
<h4 class="text-danger" id="pass-warning" style="color: white"></h4>
</form>
<script>document.getElementById("pass-warning").innerHTML =
'{{data}}';</script>
<script src="../static/js/adminreg.js"></script>
</body>
</html>
```

Containment Zone List:

```
<! DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-
scale=1.0">
   <title>Zones</title>     <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1
/css/bootstrap.min.css"
integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9Muh
Of23Q9Ifjh" crossorigin="anonymous" />
   <style>        body {
padding-top: 30px;
paddingbottom: 30px;
backgroundcolor:
#0C1017;
          color : #F5F4F4;


}
a
,
t
d
,
t
h
{
          color: #F5F4F4;
```

```
        }
    </style>
</head>

<body>
    <div class="m-4 container">
        <h1 style="font-family : monospace">Location data and
Visited People</h1>     </div>
    <div class="m-4 container">
        <table class="table">
            <thead>
                <tr>
                    <th scope="col">S.No</th>
                    <th scope="col">Latitude</th>
                    <th scope="col">Longitude</th>
                    <th scope="col">No_Visited</th>
                </tr>
            </thead>
            <tbody>

                {%- for row in data %}
                <tr>
                    <th scope="row">{{loop.index}}</th>
                    <td>{{row['LOCATE_LAT']}}</td>
<td>{{row['LOCATE_LANG']}}</td>
                    <td>{{row['VISITED']}}</td>
                {%- endfor %}
            </tbody>
```

```
        </table>
    </div>
    <div class="m-3 float-right">
        <button type="button" class="btn btn-danger"><a
href={{url_for("home")}}>Go to location update Page</a></button>
    </div>
 </body>
</html>
```

Home/Remove/Add Zone:

```
<! DOCTYPE html>
<html>
    <head>
        <title>Dashboard</title>
        <link rel="stylesheet" href="static/style.css">
        <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Open+Sans&display=swa
p" rel="stylesheet">         <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.
css"
integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9Muh
Of23Q9Ifjh" crossorigin="anonymous" />
```

```
   <style>

       body {
padding-top: 30px;
padding-
bottom: 30px;            background-color:
#0C1017;
           color : #F0F6FC;
font-family: 'Open Sans',
sans-serif;
padding: 30px;
       }

       .m-3 float-right {
background-color :
#0C1017;



}

a
{
           color: #F0F6FC;

       }
<!--       design the whole html page to beautify-->


   </style>
   </head>
   <body>
    {% if success %}
```

```
    <script>
        alert("Location Uploaded Successfully");
    </script>
     {% elif not success %}
    <script>
        alert("Enter Proper Location data");
    </script>
    {% endif %}
    <div class="logout">
        <button type="button" class="btn btn-primary"><a
href={{url_for("logout")}}>Log Out</a></button>
    </div>
    <div class="logout">
        <h1>Declare Containment Zone</h1>
    </div>

        <h2>Welcome: {{name}}</h2>

    <form method="POST" action="/home">
        <div class="container">
            <div class="form-group row">
                <div class="col-sm-6">
                    <label class="control-label">Lat.:</label>
                    <input type="text" class="form-control" id="lat"
name="lat" />                </div>
                <div class="col-sm-6">
                    <label>Long.:</label>
                    <input type="text" class="form-control" id="lon" name="lon"
/>
```

```
            </div>
            <div class="col-sm-6">
                <label>Get current Location:</label>
                <button type="button" class="btn btn-warning"
onclick="getLocation()">Current  Location</button>
                <label>(Click this first)</label>
            </div>
        </div>

        <!-- map -->
        <div id="map_disp" style="height: 400px;width: 500px;"></div>
        <div class="m-3 float-right">
            <button type="submit" class="btn btn-danger">Declare
Containment Zone</button>
        </div>
        <div class="m-3">
            <button onclick="toggleTips()" type="button" class="btn
btnsecondary">Tutorial</button>                <div id="tips" class="m-3">
                <ol>
                    <li>Select The Location By Clicking the Current Location
Button</li>
                    <li>Drag the Pin to change the location</li>
                    <li>Click on Declare Containment Zone to save the location
to the database </li>
                </ol>
            </div>
        </div>
        <div class="m-3 float-right">
            <button type="button" class="btn btn-warning"><a
href="{{url_for('data')}}">Click Here To View The
```

Containment Zones and Number of people visited</button>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.min.js" integrity="sha384-+YQ4JLhjyBLPDQt//I+STsc9iw4uQqACwlvpslubQzn4u2UU2UFM80nGisd026JF" crossorigin="anonymous"></script>

<script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>

<script src="https://maps.google.com/maps/api/js?sensor=false&libraries=places"></script>

<script src="https://rawgit.com/Logicify/jquery-locationpickerplugin/master/dist/locationpicker.jquery.js"></script>

<script>
function getLocation() {
if (navigator.geolocation) {
navigator.geolocation.getCurrentPosition(showPosition);
} else {
alert("No location");
}
}
function showPosition(position) {
$('#map_disp').locationpicker({
location: { latitude: position.coords.latitude,

```
                longitude: position.coords.longitude
            },
            radius: 0,
inputBinding: {
latitudeInput: $('#lat'),
                longitudeInput: $('#lon'),
            },
            enableAutocomplete: true,
            onchanged: function (currentLocation, radius,
isMarkerDropped) {                    // Uncomment line below to show
alert on each Location Changed event                    // alert("Location
changed. New location (" + currentLocation.latitude + ", " +
currentLocation.longitude + ")");
            }
        });
    }
    function toggleTips() {
var x =
document.getElementById("tips"
);
if (x.style.display === "none") {
x.style.display = "block";
        } else {
            x.style.display = "none";
        }
    }
  </script>

</form>
</body>
</html>
```

7.2 Features 2

The users get alerted from entering the contaminated zone by geofencing the location and sending it as notification.

CODING:

```
# create a flask app

import re import ibm_db

from flask import Flask, flash, render_template, request, redirect, url_for, session from

sendgrid import SendGridAPIClient

from sendgrid.helpers.mail import Mail

app = Flask(__name__, static_url_path='/static', static_folder='static', template_folder='templates')

app.secret_key = 'sus'

conn = ibm_db.pconnect('DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-

7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;S
SLServerCertificate=DigiCertGlobalRootCA.crt;UID=tyb34892;PWD=Qq5GdhZKREQl1Vrc', '', '')


# print("Connected to database", conn)


session = {}



# route for
sending email
@app.route('/se
ndemail') def
sendemail():
```

```
id=session["id"
]
   sql="select email from users1 WHERE id=?;"
stmt =
ibm_db.prepare(con
n, sql)
ibm_db.bind_param
(stmt, 1, id)
ibm_db.execute(stm
t)     data =
ibm_db.fetch_assoc(
stmt)      print(data)


   message =
Mail(from_email='anandmic06@gmail.com',to_emails=data['EMAIL'],subj
ect='CONTAINMENT
ZONE ALERT!!!',html_content='<strong>OPPS!<br>
YOU HAVE ENTERED A CONTAMINATED ZONE!
<br> PLEASE STAY SAFE.</strong>')     try:          sg =
SendGridAPIClient('SG.9nbY1OfbSTeYnWdfSEBzXw.e0rYMvlTWmlfc
LdwnFVP7hOH7HwSsV
Oxs-GSc_AXMtU')
       response = sg.send(message)
print(response.status_code)
print(response.body)
print(response.headers)     except
Exception as e:         print(e)


   return render_template('home.html')




# create a route for the home page
```

```
@app.route('/',
methods=['GET', 'POST'])
def register():     message =
''     if request.method ==
'POST':         # get the data
from the form         name =
request.form['username']
email =
request.form['email']
password =
request.form['password']
        confirm_password = request.form['confirm_password']
        # if nothing is entered in the form         if not
name or not email or not password or not
confirm_password:
            message = 'Please fill all the fields!'             return
render_template('register.html',
message=message)         # if the password and
confirm password do not match         elif
password != confirm_password:
message = 'Passwords do not match!'
            return render_template('register.html', message=message)


        #  password length must be 8 or above
if len(password) < 8:             message =
'Password must be 8 or more characters'
return render_template('register.html',
message=message)
        # check if the email is valid         if
re.match(r"[^@]+@[^@]+\.[^@]+", email):
            # insert the data into the database
            # check if email already exists in the database
            sql = "SELECT * FROM users1 WHERE email = '" + email + "'"
            stmt = ibm_db.exec_immediate(conn, sql)
```

```
        # print("stmt", stmt)
result = ibm_db.fetch_assoc(stmt)
# print("result", result)            if result:
message = 'The username or email
already exists!'

else:

            sql = "INSERT INTO users1 ( username, email, password, type)
VALUES (?, ?, ?, ?)"

stmt = ibm_db.prepare(conn, sql)


            ibm_db.bind_param(stmt, 1, name)

ibm_db.bind_param(stmt, 2, email)

ibm_db.bind_param(stmt, 3, password)

ibm_db.bind_param(stmt, 4, "ssss")              ibm_db.execute(stmt)

            message = 'You have successfully registered!'

return redirect(url_for('login'))       else:
message = 'The email is invalid!'
return render_template('register.html',
message=message)


@app.route('/login',
methods=['GET', 'POST']) def
login():    message = "     if

request.method == 'POST':
# get the data from the form
email = request.form['email']
password =
request.form['password']
# if nothing is entered in the
form        if not email or not
password:

        message = 'Please fill all the fields!'

        return render_template('login.html', message=message)
```

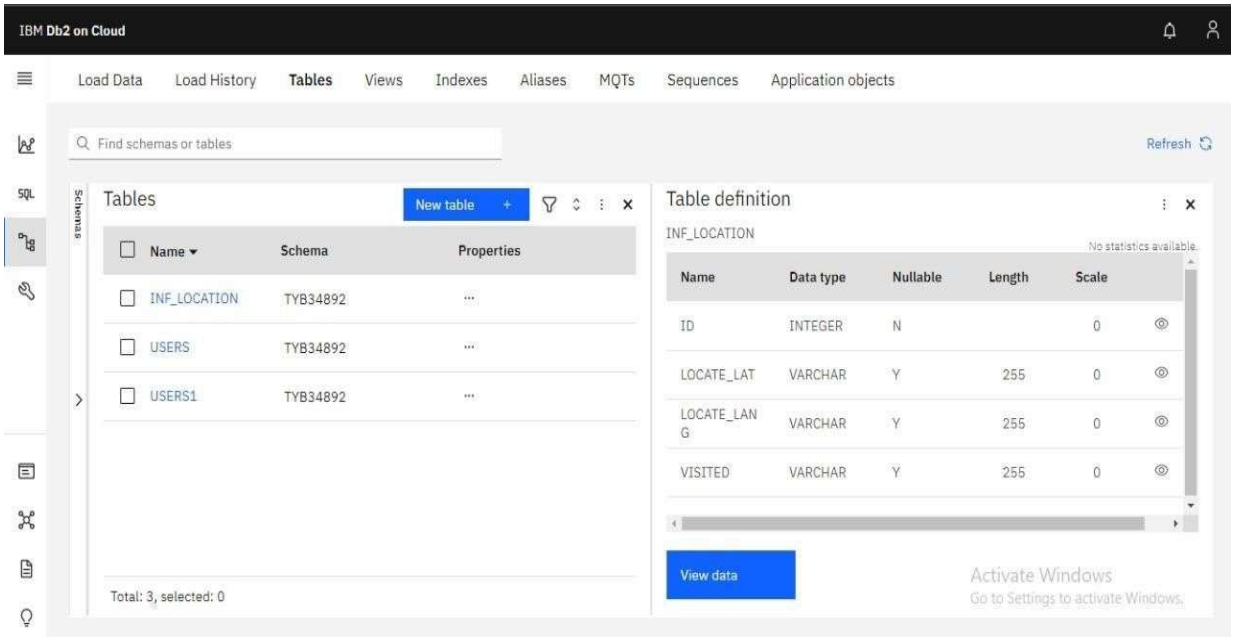
```
        # check if the username and password are valid          sql = "SELECT
* FROM users1 WHERE email = '" + email + "' AND password = '" +
password
+ """          stmt =
ibm_db.exec_immediate(co
nn, sql)
result =
ibm_db.fetch_ass
oc(stmt)          #
print("result",
result)          if
result:
            # message = 'You have successfully logged
in!'             session['id'] = result['ID']
            session['username'] =
result['USERNAME']
session['email'] = result['EMAIL']
# print("id ==", session['id'])
return render_template('home.html',
mail=email)         else:              message
= 'The email or password is incorrect!'
return render_template('login.html', message=message)



@app.route('/lo
gout') def
logout():
session.clear()
return
redirect(url_for('
login'))



# create a route for the home page and open only if the user is logged in
```

```
@app.route('/home',
methods=['GET', 'POST']) def
home():     # print(name)
     if 'id'
in session:
if
request.m
ethod ==
'GET':
return
render_te
mplate('ho
me.html',
name=ses
sion['user
name'])
if request.method == "POST":
          # get data             lat =
request.form["lat"]             lon
= request.form["lon"]
          sql = "SELECT * FROM inf_location WHERE locate_lat = ? AND
locate_lang = ?;"
          stmt = ibm_db.prepare(conn, sql)
          ibm_db.bind_param(stmt, 1, lat)
ibm_db.bind_param(stmt, 2, lon)
ibm_db.execute(stmt)
data =
ibm_db.fetch_assoc(st
mt)            if (data)!=0:
               flash("OOH!! You Have Entered A Contaminated Zone")
return redirect(url_for('sendemail'))
```

```
            else:
flash("You Are in Safe
Zone")
return redirect(url_for('home'))


        return render_template('home.html')
else:          return
redirect(url_for('lo
gin'))




# create a route for the data page and open only if the user is logged in
@app.route('/data')
def data():     if 'id'
not in session:
return
redirect(url_for('logi
n'))     else:
        # create a query to fetch the data from
the database         sql = "SELECT * FROM
inf_location"         stmt =
ibm_db.exec_immediate(conn, sql)
        # print("stmt", stmt)
        # fetch all the data from the database and store it in the
result dictionary         result = ibm_db.fetch_assoc(stmt)


        # create a list to store the data
        data = []
        # loop through the result dictionary and
append the data to the list              while result:
data.append(result)                          result =
ibm_db.fetch_assoc(stmt)
```

```
        # print(data)          return
render_template('data.html',
data=data)


if __name__ == '__main__':
app.run(host='0.0.0.0', port=5000, debug=True)
```

7.3 Database Schema



8. TESTING

8.1 TEST CASES

1. Login button click with wrong credentials entered.
2. Signup with already registered mail ID.
3. Signup with wrong form data entered.
4. Entering home page with logged out session.
5. Clicking home page buttons with logged out session.
6. Invalid data entered in change password page and requested for change in password.

8.2 User Acceptance Testing

S.NO	TEST CASE	REQUIRED OUTPUT	RESULT OUTPUT	STATUS
1	Login button click with wrong credentials.	Wrong credentials entered notification.	Wrong credentials entered notification.	ACCEPTED
2	Signup with already registered mail ID.	Email already registered notification.	Email already registered notification.	ACCEPTED
3	Signup with wrong form data entered.	Wrong credentials entered notification.	Wrong credentials entered notification.	ACCEPTED
4	Entering home page with logged out session.	Take user to login page.	Take user to login page.	ACCEPTED

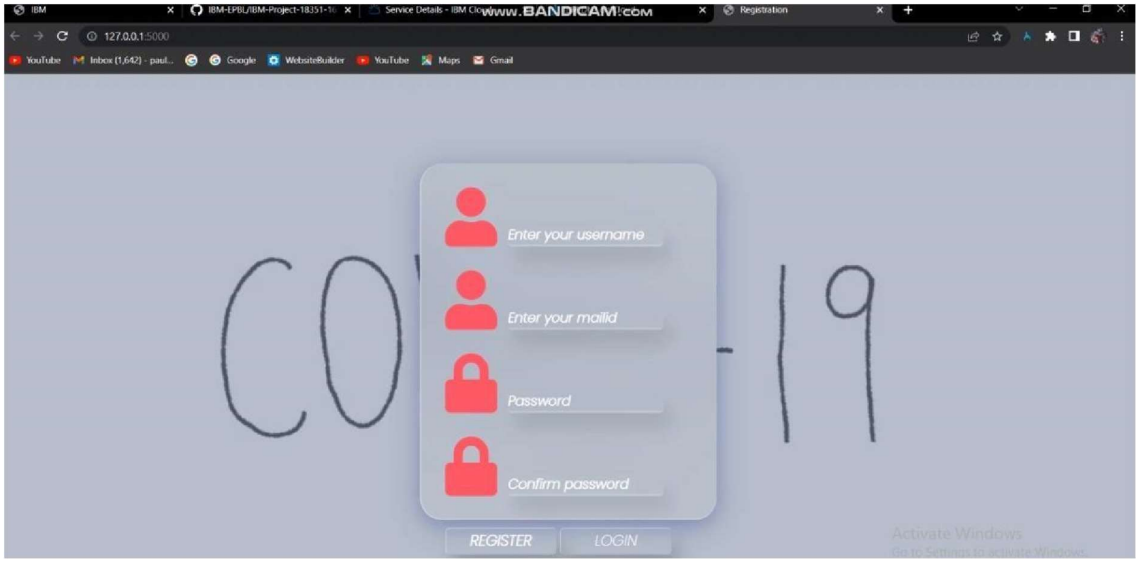
5	Clicking home page buttons with logged out session.	Take user to login page.	Take user to login page.	ACCEPTED
6	Invalid data entered in change password page and requested for change in password.	Wrong form data entered notification.	Wrong form data entered notification.	ACCEPTED

9. RESULTS

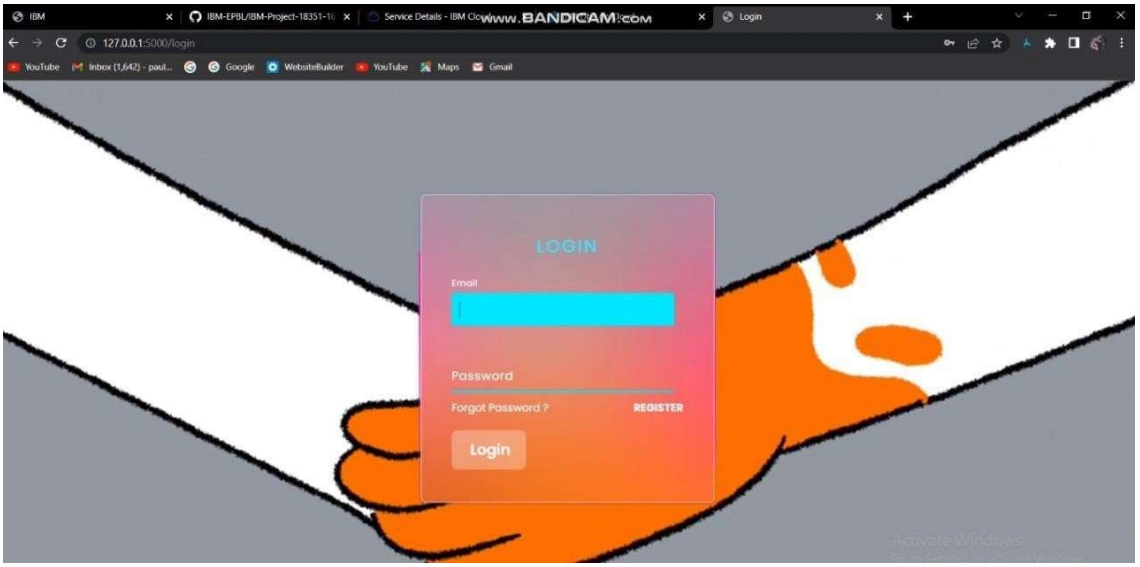
9.1 Performance Metrics

This app service monitors the location and provide information about the contaminated zones near a particular user and send notification to the user. It displays the contaminated zone area by geofencing the particular location.

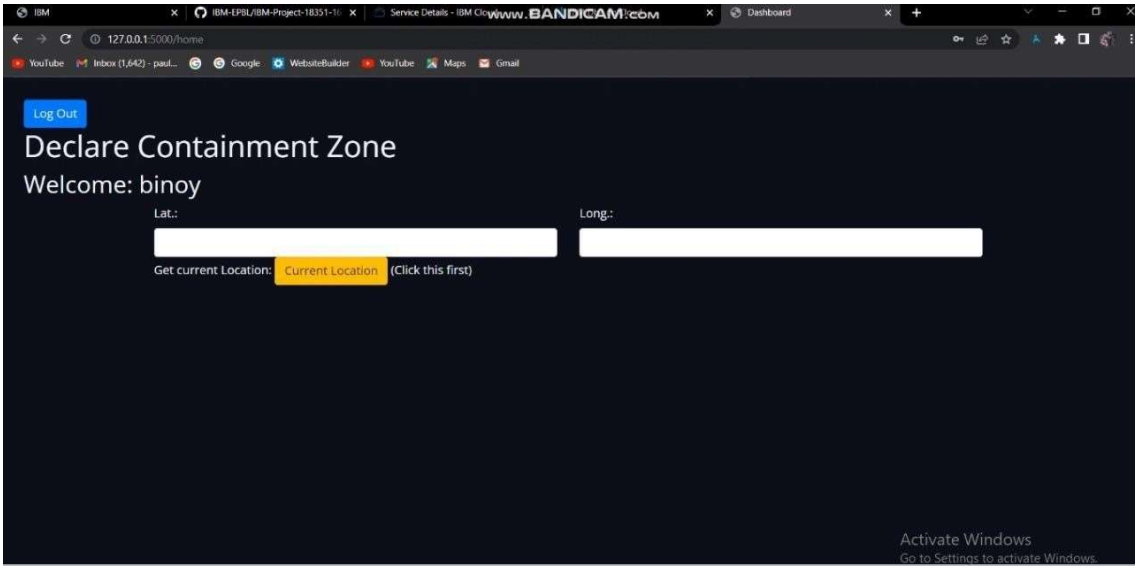
Registration Form

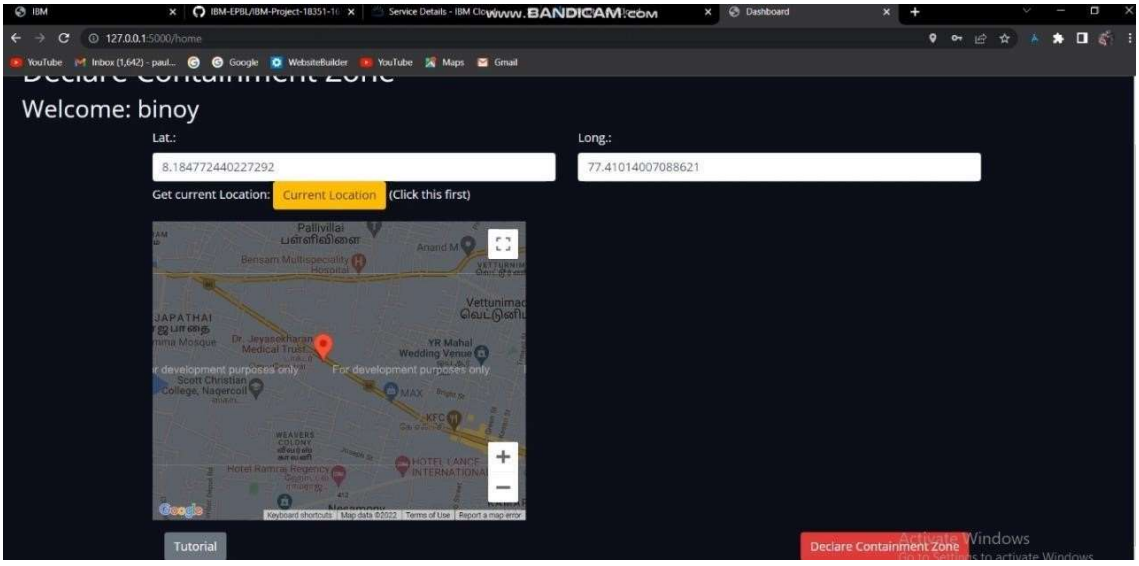


Login Form

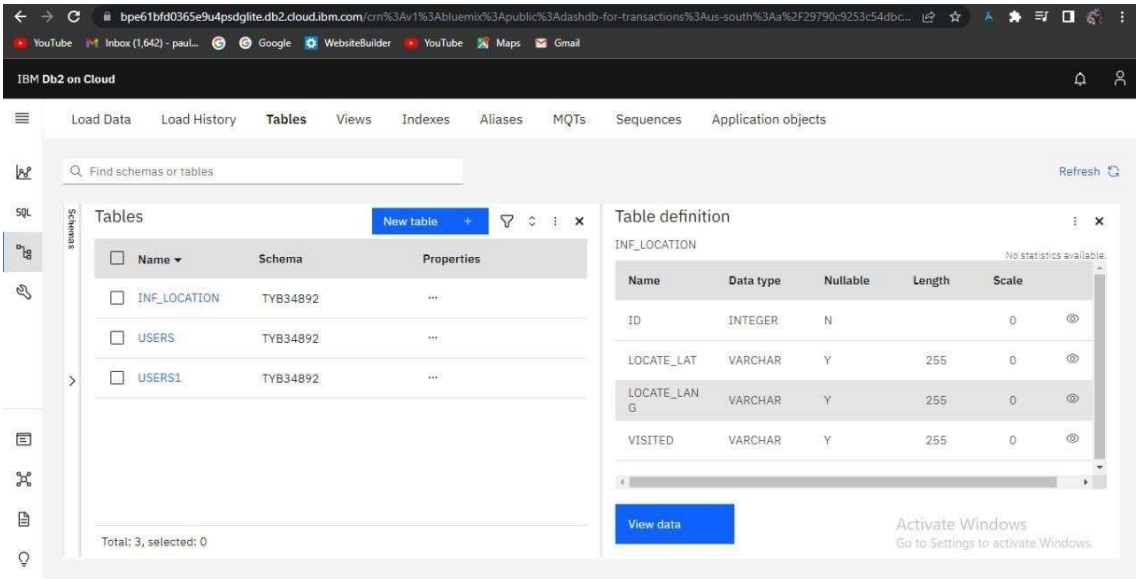


Home Page


www.BANDICAM.com
127.0.0.1:5000/home
Log Out
Declare Containment Zone
Welcome: binoy
Lat.:
Long.:
Get current Location:
Current Location
(Click this first)
Activate Windows
Go to Settings to activate Windows.


www.BANDICAM.com
127.0.0.1:5000/home
Welcome: binoy
Lat.:
8.184772440227292
Long.:
77.41014007088621
Get current Location:
Current Location
(Click this first)
For development purposes only
Tutorial
Declare Containment Zone

IBM Database



10. ADVANTAGES & DISAVANTAGES

The main advantages of containment zone alerting applications is that they can help to prevent the spread of diseases by alerting people to areas where there is a risk of infection. However, there are also some disadvantages to these, including the potential for false alarms and the possibility that people may ignore the warnings. This application is intended to provide information about containment zones in a particular region by alerting people, through continuous monitoring of an individual's location. Key benefits are monitoring people's activity and alerting them of their safety movements.

11. CONCLUSION

We proposed a framework for identifying the contaminated zone areas and store it in database for future use. Then using the database, information is provided to the user about contaminated zone areas and alerting them by sending notification and geofencing the location. From the above information, it can be concluded that the Containment zone Alerting Application, in which we have successfully developed is a web application that sends alerts to users when they enter or exit a containment zone. It uses GPS to track the user's location and sends an alert if the user enters or leaves a containment zone. It also allows users to set up alerts for specific containment zones. It has successfully demonstrated. In this project, we alert users about the containment zone area by that they are aware and realize of high containment zone area.

12. FUTURE SCOPE

The application provides an efficient way of showing the identified COVID-19 containment zones to the users in a Google map. With the alarming increase of COVID-19 affected cases throughout the world, it can be employed as a tool for creating further social awareness among the people. This application further tracks the user's location and checks whether it is present in the list of identified containment zones. It sends separate notification alerts to the user on entering and exiting the containment areas. The developed android application further extracts the IMEI Number of the trespasser in the containment zones which can be useful to the local police to track and identify people who are frequently trespassing the containment zones. Thereby this application identifies the containment zones and highlights the need for taking further precautionary measures for combating COVID- 19. The application can be further used for many purposes like maritime and forest safety to prevent users from entering restricted areas.

13. APPENDIX

The Containment zone alerting application is a web application that sends alerts to users when they are in close proximity to a containment zone. The app uses the user's location to determine if they are in close proximity to a containment zone, and if so, sends an alert to the user. The app also allows users to view a map of containment zones in their area, and provides information on how to avoid contracting the virus.

HOME.HTML

```
<! DOCUTYPE html>
<html>
   <head>
      <title>Dashboard</title>
      <link rel="stylesheet" href="static/style.css">
      <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Open+Sans&display=swa
p" rel="stylesheet">        <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-
scale=1.0">
   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.
css"
integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9Muh
Of23Q9Ifjh" crossorigin="anonymous" />
   <style>
```

```
        body {
padding-top: 30px;
padding-
bottom: 30px;            background-color:
#0C1017;
            color : #F0F6FC;
font-family: 'Open Sans',
sans-serif;
padding: 30px;
        }

        .m-3 float-right {
background-color :
#0C1017;

        }
         a
{
            color: #F0F6FC;

        }
<!--        design the whole html page to beautify-->


    </style>
    </head>
    <body>
     {% if success %}
    <script>
        alert("Location Uploaded Successfully");
```

```
    </script>
     {% elif not success %}
    <script>
        alert("Enter Proper Location data");
    </script>
    {% endif %}
    <div class="logout">
        <button type="button" class="btn btn-primary"><a
href={{url_for("logout")}}>Log Out</a></button>
    </div>
    <div class="logout">
        <h1>Declare Containment Zone</h1>
    </div>

        <h2>Welcome: {{name}}</h2>

    <form method="POST" action="/home">
        <div class="container">
            <div class="form-group row">
                <div class="col-sm-6">
                    <label class="control-label">Lat.:</label>
                    <input type="text" class="form-control" id="lat"
name="lat" />                </div>
                <div class="col-sm-6">

                    <label>Long.:</label>
                    <input type="text" class="form-control" id="lon" name="lon"
/>
                </div>
```

```
            <div class="col-sm-6">
                <label>Get current Location:</label>
                <button type="button" class="btn btn-warning"
onclick="getLocation()">Current  Location</button>
                <label>(Click this
first)</label>                </div>
        </div>

        <!-- map -->
        <div id="map_disp" style="height: 400px;width: 500px;"></div>
        <div class="m-3 float-right">
            <button type="submit" class="btn btn-danger">Check
Containment Zone</button>
        </div>
        <div class="m-3">
            <button onclick="toggleTips()" type="button" class="btn
btnsecondary">Tutorial</button>                <div id="tips" class="m-3">
                <h6>
                    Type Your Langitude and Longitude<br><br>
                   Click on Check Containment Zone to know the Status
                </h6>
            </div>
        </div>
        <div class="m-3 float-right">
            <button type="button" class="btn btn-warning"><a
href="{{url_for('data')}}">Click Here To View The
                    Containment Zones and Number of
people visited</a></button>
        </div>
      </div>
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.min.js"
integrity="sha384-
+YQ4JLhjyBLPDQt//I+STsc9iw4uQqACwlvpslubQzn4u2UU2UFM80nGi
sd026JF"          crossorigin="anonymous"></script>
<script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>
<script
src="https://maps.google.com/maps/api/js?sensor=false&amp;libraries=pla
ces"></script>
<script
src="https://rawgit.com/Logicify/j
query-
locationpickerplugin/master/dist/locationpicker.jquery.js"></script>

<script>          function getLocation() {
if (navigator.geolocation) {
navigator.geolocation.getCurrentPosition(show
Position);
        } else {
            alert("No location");
        }
    }
    function showPosition(position) {
$('#map_disp').locationpicker({
            location: {                latitude:
position.coords.latitude,
                longitude: position.coords.longitude
            },
radius: 0,
inputBinding: {
latitudeInput: $('#lat'),
```

```
                longitudeInput: $('#lon'),
            },
            enableAutocomplete: true,
            onchanged: function (currentLocation, radius,
isMarkerDropped) {
                // Uncomment line below to show alert on each Location
Changed event
// alert("Location changed. New location (" + currentLocation.latitude + ",
" + currentLocation.longitude + ")");
            }
        });
    }
    function toggleTips() {
var x =
document.getElementById("tips"
);
if (x.style.display === "none") {
x.style.display = "block";
        } else {
            x.style.display = "none";
        }
    }
</script>

</form>
</body> </html>
```

LOGIN.HTML :

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <title>Login</title>
    <link rel="stylesheet" href="static/style.css">
    <style>          body {          background-image:
url('https://cdn.dribbble.com/users/27417/screenshots/11017927/media/2a
0706f846edace3c4e81e0af
4e63807.gif');          background-
repeat: no-repeat;          background-
attachment: fixed;
          background-size: 100% 100%;

        }
        </style>
  </head>
  <body>
  <h1>Login</h1>
  <p>{{message}}</p>
  <div class="form_div">
    <form action="/login" method="POST">
        <input type="text" name="email" placeholder="Enter your
email"><br>
        <input type="password" name="password" placeholder="Enter
your password"><br>
        <input type="submit"
value="Login">      </form>
<!--      create a sign up button for register-->
      <form action="{{url_for('register')}}" method="GET">
        <p>To create a new account</p>
        <input type="submit" value="Sign Up">
```

```
    </form>
  </div>
  </body>
```

REGISTER.HTML :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Registration</title>
    <link rel="stylesheet" href="static/style.css">

<style>
body {
        background-image:
url('https://wallpaperaccess.com/full/1198250.gif');
background-repeat: no-repeat;          background-attachment: fixed;
        background-size: cover;
       }
       </style>
  </head>
  <body>
  <h1>Registration</h1>
  <p>{{message}}</p>
  <div class="form_div">
    <form action="/" method="POST">
      <input type="text" name="username" placeholder="Enter your
username"><br>
```

```
        <input type="text" name="email" placeholder="Enter your
email"><br>
        <input type="password" name="password" placeholder="Enter
your password"><br>
<input type="password" name="confirm_password" placeholder="Confirm
your password"><br>          <input type="submit" value="Register">
    </form>
<!--      create a login field for existing user-->
    <form action="{{url_for('login')}}" method="GET">
      <p>Already have an account?</p>
      <input type="submit" value="Login">
    </form>
  </div>
  </body>
</html>
```

GITHUB ACCOUNT: https://github.com/IBM-EPBL/IBM-Project-7797-1658899550