

# Project Report

Team ID	PNT2022TMID14431
Project Name	CONTAINMENT ZONE ALERTING APPLICATION

## 1. INTRODUCTION

### 1.1 Project Overview:

Currently there are several research works undergoing in the country to prevent Covid-19 cases from rising. Previously our country was importing medical kits like PPE (Personal Protection Kits), mask from outside, but now it has been successful in developing these kits. Along with taking initiatives to fight this disease, our country has also taken steps to make people aware of the disease. The news and media have a great part in creating this awareness by informing the public about the preventive measures that can keep them away from infection. Awareness among the people to carry out all the preventive measures can immensely help to reduce spread of the virus. The country has created containment zones throughout the cities wherever Covid-19 cases have been reported to prevent further spread of the virus. These containment zones have been kept isolated from the outside public to ensure no contamination occurs outside. After more than 2 months of the lockdown, the government has relaxed some of the lockdown rules and has permitted reopening of government offices, bus and other road transportation facilities and shopping markets. People can move inside the city for work and other purposes. But the containment zones are still being kept isolated, and new containment zones are being formed wherever Covid-19 cases have been reported. These zones are highly contagious as droplets with virus coughed out from an unscreened asymptomatic patient can travel up to 8 m (Bahl et al. 2020). Though these containment zones are guarded by policemen, still there remains a chance that people might unknowingly step into them. In this situation where people can move in the city, these containment zones pose a risk of infection to these city dwellers. Therefore, informing people about the location of the containment zones can help them bypass and avoid these zones and thereby reduce the chance of community transmission. In this paper, we focus on developing a mobile based application to provide information regarding the Covid-19 containment zones in West Bengal. The application further tracks the user's location and provides notification alert if the user has entered a containment zone. The application also provides daily Covid-19 case statistics to the users to keep them updated. The application is developed on Android SDK and uses Firebase Cloud Firestore to store the location data. Android's geofencing client is used to create geofences around the containment zones and notification manager is used to provide notifications. The application also uses RESTful web services to show the Covid-19 cases in West Bengal. We have tested our application with different users in different locations across West Bengal and it works efficiently and is able to attain our target.

### 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



## 2. LITERATURE SURVEY:

### 2.1 Existing problem:

People doesn't have proper knowledge about containment zones since they do change daily and hard to keep updated and if they are not updated properly, they will lead to wide spread of disease.

### 2.2 References:

1. <https://ieeexplore.ieee.org/document/9711880>
2. <https://ieeexplore.ieee.org/document/9432254>
3. <https://ieeexplore.ieee.org/document/9356316>
4. <https://ieeexplore.ieee.org/document/9388625>
5. <https://ieeexplore.ieee.org/document/9609407>
6. <https://ieeexplore.ieee.org/document/9356316>

### 2.3.Problem Statement Definition:

Problem Statement (PS)	I am (USER)	I'm trying to	But	Because	Which makes me feel
PS-1	Is it belonging all the age groups	Share my status to Everyone	Is it any compulsion to do that	It makes me embarrassed	As an influence person by spreading an irrelevant content to my native areas
PS-2	If a mobile app user whether we give all personal details what the app expecting from us.	The Given details should be verified by an admin side.	There are many apps cannot secure the data in a proper manner	The updating of an app cannot be prolonged.	Unsecured to give my personal data's.
PS - 3	Whether the updating of containment zone is plotted in a really affected area.	There is any way to share the alerted zone to others	It has an any constraint to share the location within the particular distance.	I am trying to reach beyond the areas.	Niche intimate to our others.
PS - 4	The alert messages are received by where only entered into the alerted zone	To stay awake from the alerted zone	Usage of mobile phone are taken out frequently to check out the messages	Just to check out whether I entered in to the containment zone or not.	Phone Holder for the frequent usage.
PS - 5	There is any feature to connect the Emergency Service.	When there is a helpless situation appeared.	The connections to helpline response as much earlier	On that hectic moment it was an only hope.	Pulled down into the Blinded mind



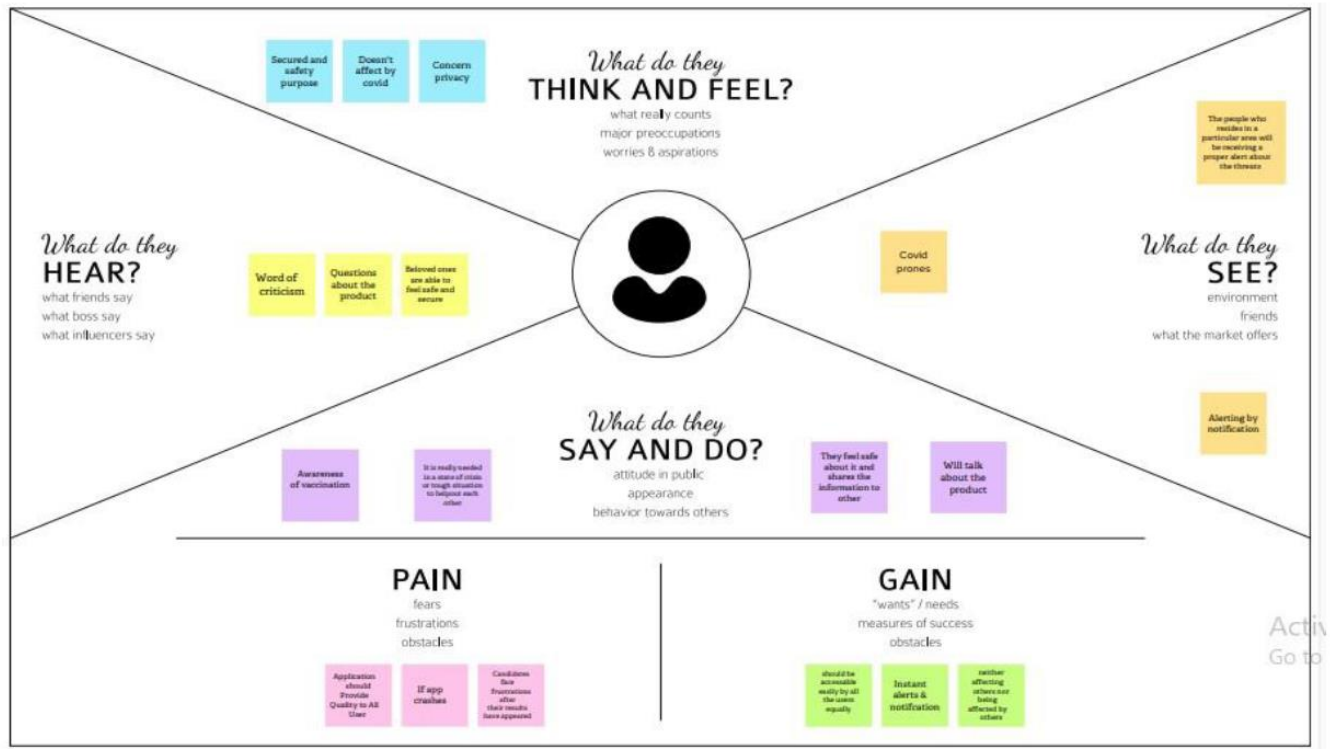
**Proposed Solution:**

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The users of the Android application can see where the containment zones are located. Additionally, it alerts the user if they remain inside containment zones or cross their boundaries.
2.	Idea / Solution description	To contain infectious disease outbreaks, contact tracing and case isolation are utilized, as they were with the coronavirus epidemic in 2019. (COVID-19). The features of the disease and the response from public health organizations will determine if this technique will be successful in achieving control. Long intervals between the onset of symptoms and isolation, a dearth of cases discovered through contact tracing, and rising transmission prior to symptoms all reduce the likelihood of control.
3.	Novelty / Uniqueness	Due to its crucial role in the creation of new vaccinations, scientific novelty is significant throughout the pandemic. In order to broaden teams' search operations for a larger scope of resources needed to handle the global challenge, parachuting collaboration and international collaboration are key channels. A pandemic call for the preservation of a cooperative scientific community that goes beyond established networks and nationalism, according to findings. Evidence from COVID-19 suggests that pandemics spur scientific innovation. The search for new vaccinations and treatments during the epidemic is driven by scientific novelty. The expansion of teams' search efforts for a greater impact requires international and first-time collaboration, which are crucial channels.
4.	Social Impact / Customer Satisfaction	The 2019 Coronavirus 1 disease (Covid-19 2) pandemic 3 has grown to be the highest total demand for immediate assistance in the twenty-first century. The World Health Organization (WHO) identified Covid-19 2 to be a Public Health immediately needed of international business place on January 30, 2020. (PHEIC). The Covid-19 2 pandemic 3 have greater size, range, degree of readiness to move limitations, travel stops, and edge close to stone edging than those in preceding PHEICs during the past hundred years

5.	Business Model (Revenue Model)	<p>Scale-copy refers to many experiments, thinking through how to create value for clients and other interested companies on the back end, and capturing value through the organization's mechanisms for converting value into money.</p> <p>The company's sector is the first mathematical factor that has influence over this. Higher degrees of inclusivity have typically been accompanied by an improvement in the ability to bounce back.</p> <p>The likelihood that someone receiving assistance or commodities will continue their relationship with the same organization providing goods increases with the offering's size and distance down.</p>
6.	Scalability of the Solution	<p>The Covid-19 pandemic's confluence with the accompanying lockdown and many businesses' need to operate entirely remotely. In February, our Upscale programme for mid-stage, scaling businesses had just begun. What was it like for businesses that were in the process of rapidly scaling? ODDBOX doubled their predicted growth prediction and saw growth of 600% year over year.</p> <p>Farewill had to adjust to entirely remote service at the same time that demand increased.</p>

### 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 help 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



### 3.3 Ideation & Brainstorming

**Define your problem statement**

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.

**Brainstorm**

Write down any ideas that come to mind that address your problem statement.

10 minutes

**Sai Siddhartha Narasimha**

For the precise use of location and get prior alert messages

We need to access the location while to see the alerted zones

And they provide correct information native zones

**S V S Dhwanaj Kumar**

We can send grid type of alert messages

If the person encountered into the marked zones the alert messages are sent via mail

Users can see the updated alert zones before moving to the destination point

**Sai Reddy Dadda**

How can we push the notification to the user

Can we send the alert messages through App notification

User can share their status as public whether they affected or not

User must keep updated of their app for better reliable of data

**Sonu Sampath Reddy**

If we want push the notification we need to provide the alerted zones

User can access the nearby pharmacy with limited no. of tablets without Doctor prescription

**Ujjini Vijay Kumar**

We might face lot of challenges

We must develop both web & mobile application for the various kind of users

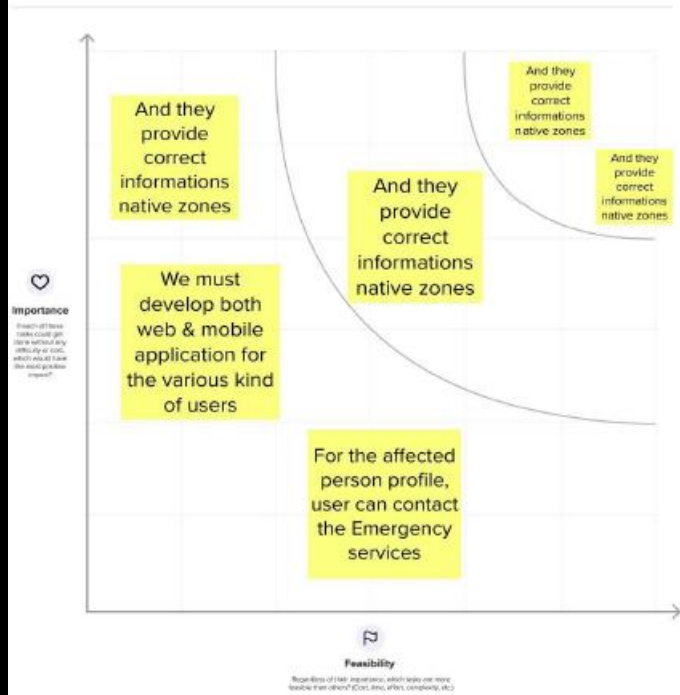
The alert messages should be some interactive

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





## 3.4 Problem Solution fit

### Problem-Solution Fit canvas

Purpose / Vision

Version:

Define CS, fit into CL	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> Our clients are individuals who have travelled from one location to another	<b>6. CUSTOMER LIMITATIONS</b> <span>CL</span> <small>EG. BUDGET, DEVICES</small> Avoid flying with an economy problem financial turbulence	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> <small>PLUSES &amp; MINUSES</small> The software only displays a containment zone for the Covid 19 virus when it has spread. There is a safe route displaying app available, but not everyone can use it. It just displays the safe path to go.	Explore AS, differentiate
	<b>2. PROBLEMS / PAINS</b> <span>PR</span> <small>+ ITS FREQUENCY</small> Moving to a new place without first learning about the local climate. Find a solution to any issues they may be having if they are stuck.	<b>9. PROBLEM ROOT / CAUSE</b> <span>RC</span> Ignorance of the pandemic not keeping up with the necessary precautions. inadequate direction on pharmaceutical inputs	<b>7. BEHAVIOR</b> <span>BE</span> <small>+ ITS INTENSITY</small> The virus's behavior cannot be predicted with precision. Based on the length of time it has been incubating, its intensity or vulnerability is determined.	
Identify strong TR & EM	<b>3. TRIGGERS TO ACT</b> <span>TR</span> People in this culture have developed a phobia of the fake news that circulates on social media and prompts people to act inappropriately during this pandemic.	<b>10. YOUR SOLUTION</b> <span>SL</span> Developing a cloud-based programme that links patients and doctors. connecting hospitals and obtaining case information by providing doctors with distinct logins constructing an algorithm that evaluates information provided by hospitals and produces a list of dangerous viruses that propagate. constructing a user-friendly interface that sends signals of caution to users when they visit polluted areas .	<b>8. CHANNELS of BEHAVIOR</b> <span>CH</span> <b>ONLINE</b> Users on the internet will learn more about the virus that is now circulating.	Extract online & offline CH of BE
	<b>4. EMOTIONS</b> <span>EM</span> <small>BEFORE / AFTER</small> People would experience a sense of security and joy as a result of learning about the pandemic's spread and being informed prior to contracting it.		<b>OFFLINE</b> The company's service will be available to offline consumers.	

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Mobile Number Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Permission	The user needs to provide location access,in order to trace his/her location.
FR-4	Containment zones are shown	Containment zones are marked and trespassers were indicated by geofencing.
FR-5	Tracking the location	Trace the trespassers by using Google map API
FR-6	Alert message via notification	By tracking their location,a message will be send if they enter the containment zone.

### 4.1 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	People who travel can use this application to track the containment zone and pass through a safe route.
NFR-2	<b>Security</b>	Using blockchain technology for location and data encryption to protect user's data from getting into wrong hands.
NFR-3	<b>Reliability</b>	Fake news will be avoided and proper guidance is given in the application. The user can trust the result and navigate safely.
NFR-4	<b>Performance</b>	The geofencing is updated daily and shows the day- to-day updates of containment zones.
NFR-5	<b>Availability</b>	The application uses the network to load the google maps to retrieve containment zones.It is available for a good range of network bandwidth.
NFR-6	<b>Scalability</b>	This application can be accessed from anyplace and monitoring users movements in pandemic zones and alerts before they are affected.

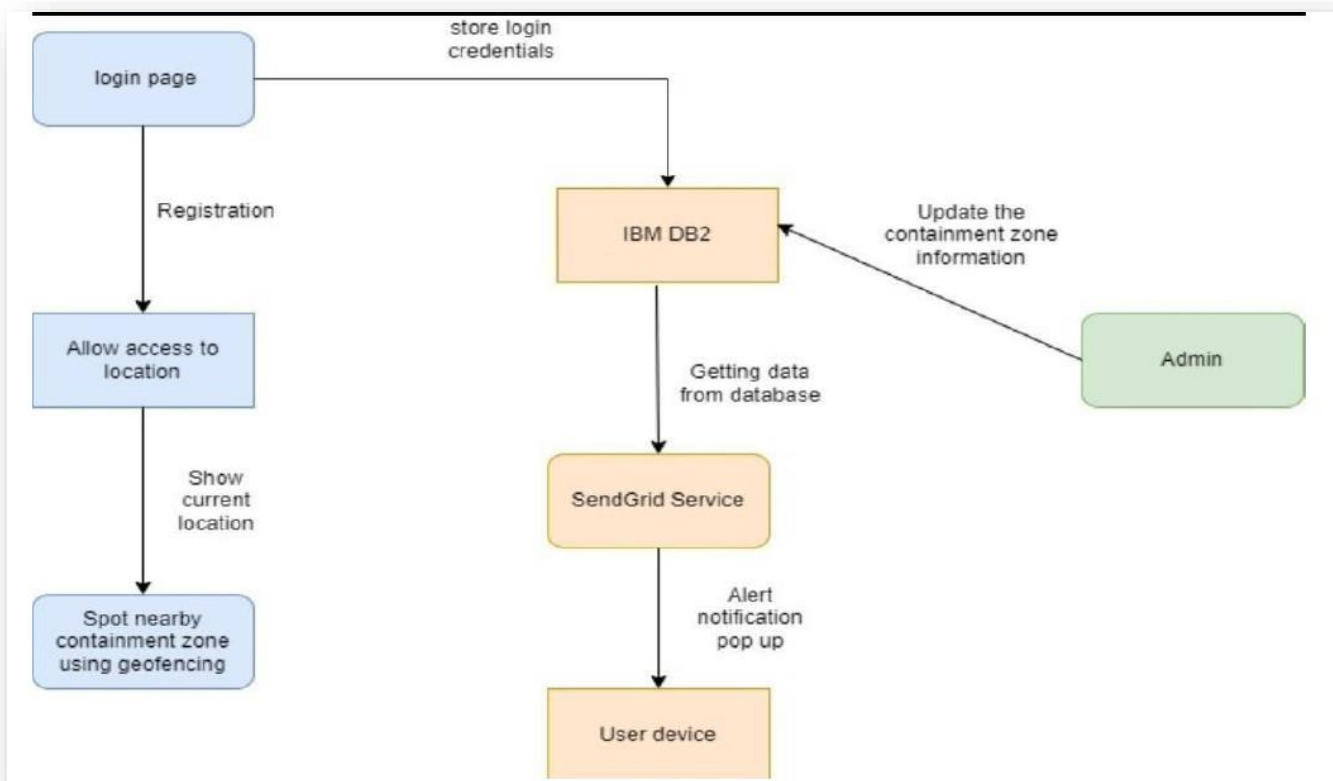
## 5.PROJECT DESIGN

### Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically.

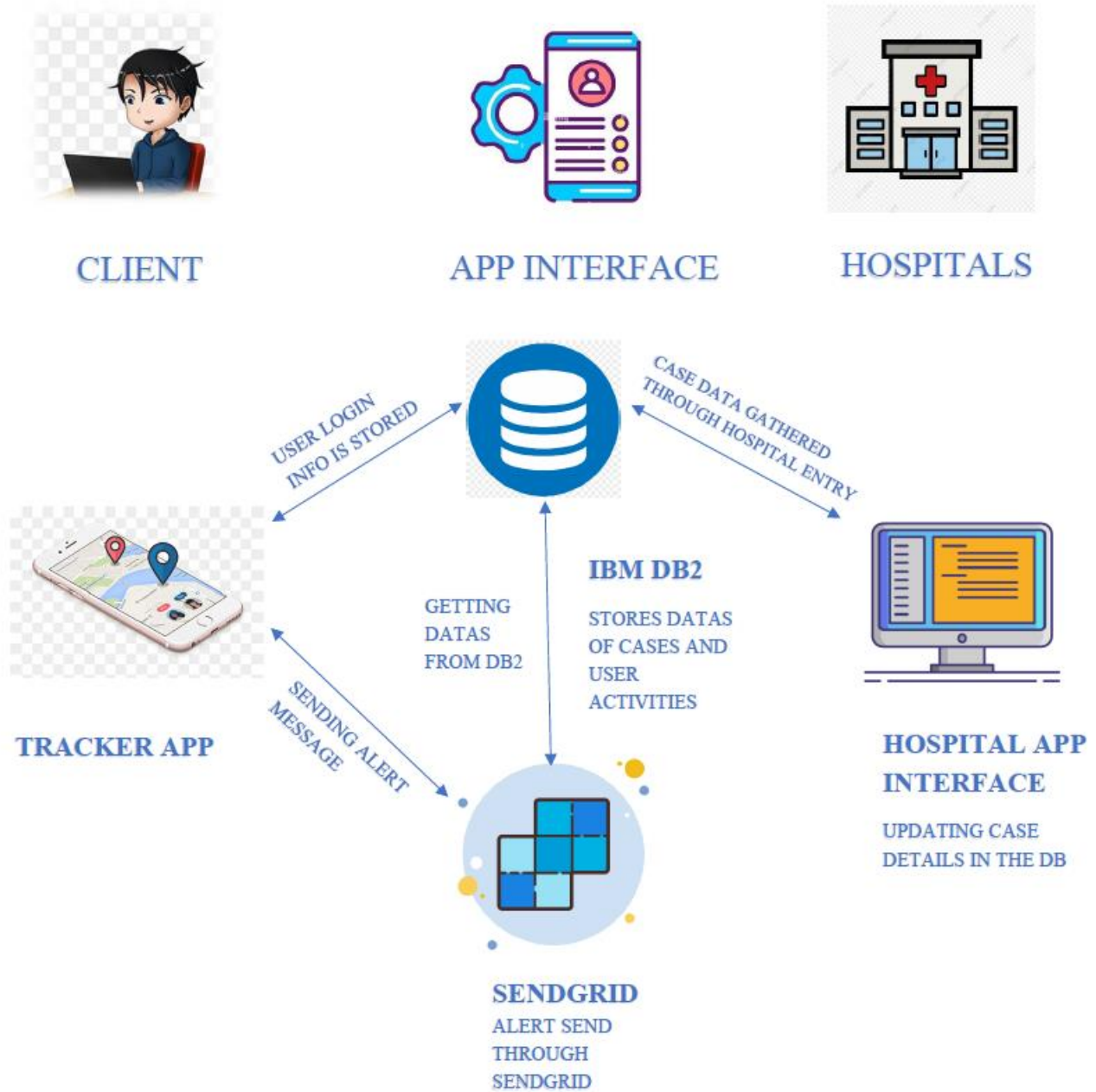
It shows how data enters and leaves the system, what changes the information, and where data is stored.

### 4.2 Data flow diagram:



## 5.2.SOLUTION ARCHITECURE:





# SOLUTION ARCHITECURE



**TECHNICAL ARCHITECTURE:**

Table-1: Components & Technologies: S. No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, Java, XML, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python-Flask
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Cloud Object Storage
8.	External API-1	Purpose of External API used in the application	IBM CLOUD API, Google Maps API
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration:	Local, Cloud Foundry, Kubernetes,

**User Stories:**

Journey Steps Which step of the experience are you describing?	Discovery Why do they even start the journey?	Registration How does the customer register in this app?	Onboarding and First Use How can they feel successful?	Sharing Why would they invite others?
<b>Actions</b> What does the customer do? What information do they look for? What is their context?	for providing information about containment zones	sign up using google account app should have user registration and login customer have to connect their location for app after login	search bar is able to search containment zones able to get help by using HELP icon learn more about feedback button is able to use get notifications and alerts about the containment zones	ask and invite user friendly application easy to access and providing appropriate information
<b>Needs and Pains</b> What does the customer want to achieve or avoid? Tip: Reduce ambiguity, e.g. by using the first person narrator.	monitoring containment zones and alerting them about their safety requirements consumers can see how many are affected in those areas	avoid providing wrong information customer should have the updated app able to get info about other containment zones	gets an alert notification through mail process of finding info about the affected areas successfully monitoring the public movements while entering the containment zones	Real time tracking of outbreak real time tracking of consumer's navigation helps in isolation information
<b>Touchpoint</b> What part of the service do they interact with?	can search the containment zones	online searching containment zones websites free trial page Apps in android, ios, ...	see first confirmed to visit to unknown locations update of alert in the containment zones	finding the containment zones is simple making your travel safe and comfort ease the work of doctors and reporters
<b>Customer Feeling</b> What is the customer feeling? Tip: Use the <b>emoji app</b> to express more emotions				
<b>Backstage</b>				
<b>Opportunities</b> What could we improve or introduce?	Increase/decrease a leading metric by improving X or introducing Y.	Increase/decrease a leading metric by improving X or introducing Y.	Increase/decrease a leading metric by improving X or introducing Y.	Increase/decrease a leading metric by improving X or introducing Y.

**6.PROJECT PLANNING & SCHEDULING:****Product Backlog, Sprint Schedule, and Estimation (4 Marks)**

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint -1	Registration	USN-1	User: I can register for the application by entering my email, password and verifying password.	3	High	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
		USN-2	User: I will receive a confirmation email once I have registered for the application.	2	High	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
		USN-3	User: I can register for the application through Gmail.	5	Medium	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
		USN-4	Management: I need to register my hospitals on the site.	2	High	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
		USN-5	User: I can log into the application by entering my email & password	3	High	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
	Login	USN-6	Management: I need to login into my dashboard with my given hospital id and password.	5	Medium	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
	Dashboard	USN-7	User: I need to give permission to access my Contacts, Location, and Storage	5	High	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		USN-8	User: I get access to the dashboard which shows a map with containment zones	5	High	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
		USN-9	Management: I need to enter the case information of the	5	High	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay

			patient that visits our hospital.			Vijay
	Services	USN-10	Admin: I need to provide valid information about the pandemic out there.	5	High	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
Sprint-3	Dashboard	USN-11	Management: I need to store all the patient information on the cloud.	5	High	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
	Services	USN-12	Admin: I need to provide medical advice through a chatbot.	5	Medium	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
		USN-13	Admin: I need to provide medical recommendations by collaborating with top hospitals.	5	Low	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
		USN-14	Admin: I need to provide preventive measures when they travel through it.	5	High	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
	Registration	USN-15	User: I can register for the application through Facebook.	2	Low	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay

Sprint-4		USN-16	User: I can register for the application through Twitter.	2	Low	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
	Services	USN-17	Admin: I need to alert the user when they enter pandemic zones.	3	Medium	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
		USN-18	Admin: I need to provide special services for premium users by giving services like monitoring health by their smart bands.	3	Low	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay
	Data Collection	USN-19	Admin: I need to store all the user information on the cloud	5	Medium	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
		USN-20	Admin: I need to collect the recent list of diseases in the world.	5	Low	Siddhardha, Dheeraj, Sai Reddy, Somu Sampath, Vijay



**6.2. Sprint Delivery Schedule**

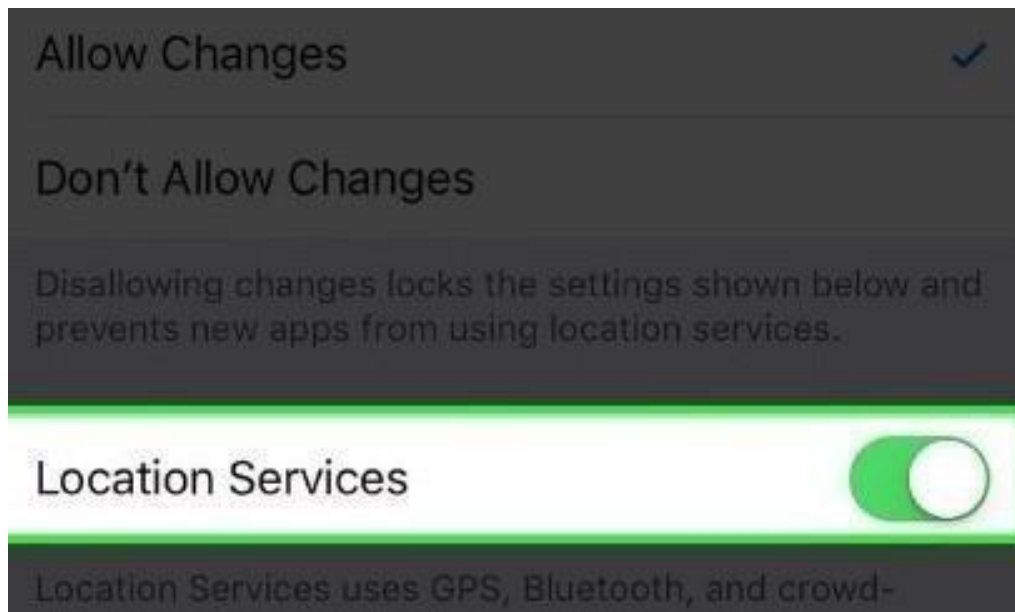
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

**Velocity:**

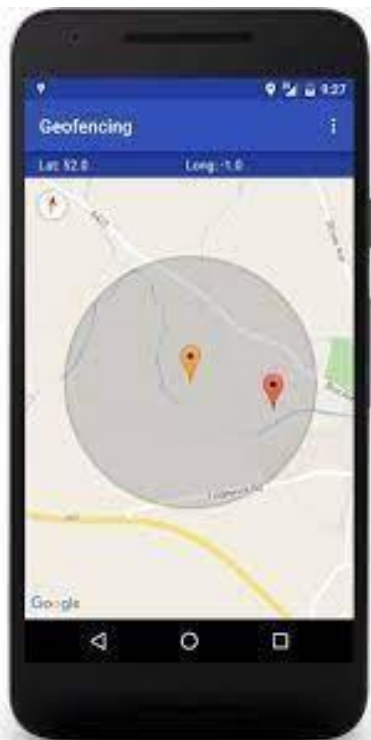
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

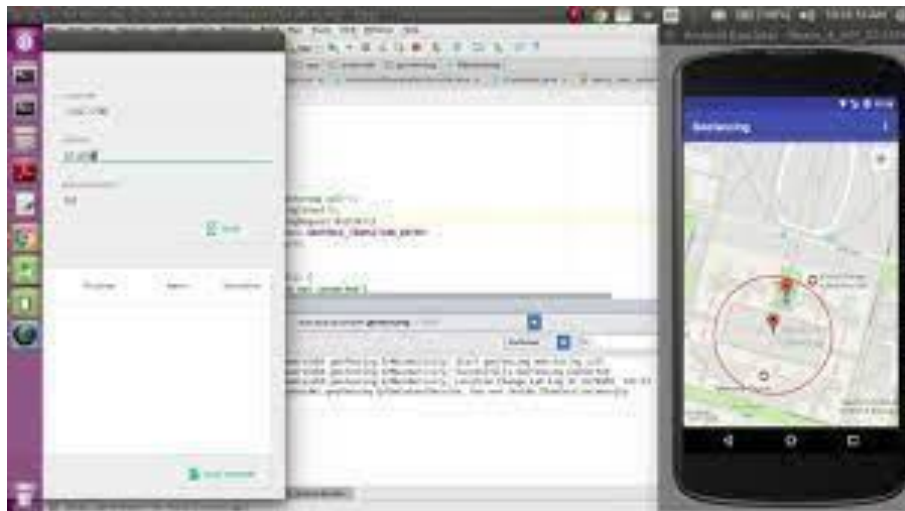
$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

**7.CODING & SOLUTIONING**



#### GEOFENCE IN ANDROID APP :







8 TESTING  
8.1 Test Cases

2

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Yes/No)	BUG ID	Executed By
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the login/signup popup when user clicked on My Account button	1 Enter URL and click go 2 Scroll down 3 Verify login/signup popup displayed on screen	http://192.168.51.204:215/50106/	Login/signup popup should display	Working as expected	PASS	Successful			ATICHAYA JENITHA KOWSALYA
LoginPage_TC_002	UI	Home Page	Verify the UI elements in login/signup popup	1 Enter URL and click go 2 Click on Sign button for User 3 Verify login/signup popup with below UI elements a id text box b password text box c Login button d New customer? Create account link e Lost password? Recovery password link	http://192.168.51.204:215/50106/	Application should show below UI elements a actual text box b password text box c Login button with proper color d New customer? Create account link e Lost password? Recovery password link	Working as expected	PASS	Successful			PRIVADHARSHINI KOWSALYA PRIVADHARSHINI
LoginPage_TC_003	Functional	Home page	Verify user is able to login application with valid credentials	1 Enter URL (http://shopowner.com) and click go 2 Click on My Account dropdown button 3 Enter invalid ID in ID text box 4 Enter valid password in password text box 5 Click on login button	ID: 5142 password: Testing123	User should navigate to user account home page	Working as expected	PASS	Successful			ATICHAYA PRIVADHARSHINI

Test Case (SPRINT 01)

3

Test Case (SPRINT 01)												
3												
LoginPage_TC_001	Functional	Login page	Verify user is able to log application with valid credentials	1 Enter URL (http://192.168.51.204:215/50106/) and click go 2 Click on My Account dropdown button 3 Enter invalid ID in ID text box 4 Enter valid password in password text box 5 Click on login button	ID: 5142 password: Testing123	Application should show Success email or password validation message	Working as expected	PASS	Successful			JENITHA
LoginPage_TC_002	Functional	Login page	Verify user is able to log application with valid credentials	1 Enter URL (http://192.168.51.204:215/50106/) and click go 2 Click on My Account dropdown button 3 Enter valid ID in ID text box 4 Enter invalid password in password text box 5 Click on login button	ID: 5142 password: Testing123	Application should show Success email or password validation message	Working as expected	PASS	Successful			PRIVADHARSHINI
LoginPage_TC_003	Functional	Login page	Verify user is able to log application with valid credentials	1 Enter URL (http://192.168.51.204:215/50106/) and click go 2 Click on My Account dropdown button 3 Enter invalid ID in ID text box 4 Enter valid password in password text box 5 Click on login button	ID: 5142 password: Testing123	Application should show Success email or password validation message	Working as expected	PASS	Successful			ATICHAYA

Test Case (SPRINT 01)

4

LoginPage_TC_007	Functional	Login page	Verify User is able to log into application with Valid Credentials	1 Press URL: http://199.51.204.21:53000/ and click go 2 Click on My Account dropdown button 3 Enter Invalid ID in ID test box 4 Enter invalid password in password test box 5 Click on login button	ID: 5131 password: Testing123	Application should show correct email or password validation message	Working as expected	PASS	Successful					ATULIYA PRIVADHARSHINI
LoginPage_TC_008	Functional	Login page for ADMIN	Verify User is able to log into application with Valid Credentials	1 Enter URL: http://199.51.204.21:53000/ and click go 2 Click on My Account dropdown button 3 Enter Valid ID in ID test box 4 Enter valid password in password test box 5 Click on login button	ID: 1111 password: 5678	Application should show correct email or password validation message	Working as expected	PASS	Successful					PRIVADHARSHINI KOTVALYA
LoginPage_TC_009	UI	ADMIN PAGE	Verify all the Customer database is visible	1 Press URL: http://199.51.204.21:53000/ and click go 2 Click on My Account dropdown button 3 Enter Invalid ID in ID test box 4 Enter Invalid password in password test box 5 Click on login button	http://199.51.204.21:53000/	Customer database is visible	Working as expected	PASS	Successful					JENITHA

Test Case (SPRINT 01)

5

LoginPage_TC_010	Functional	USER REGISTUR	Verify it sent to customer email address	1 Enter URL: http://199.51.204.21:53000/ and click go 2 Register the account by using credentials 3 Click on button Submit	http://199.51.204.21:53000/	Email sent successfully	Working as expected	PASS	Successful					ATULIYA
LoginPage_TC_011	Functional	AGENT REGISTUR	Verify AGENT is able to log into application with Valid Credentials	1 Enter URL: http://199.51.204.21:53000/ and click go 2 Click on My Account dropdown button 3 Enter Invalid ID in ID test box 4 Enter Invalid password in password test box 5 Click on login button	ID: 2345 password: Testing123	ID sent successfully	Application should show correct email or password validation message	PASS	Successful					PRIVADHARSHINI
LoginPage_TC_012	Functional	Login page for ADMIN	Verify User is able to log into application with Invalid Credentials	1 Enter URL: http://199.51.204.21:53000/ and click go 2 Click on My Account dropdown button 3 Enter Invalid ID in ID test box 4 Enter Invalid password in password test box 5 Click on login button	ID: 1111 password: 5678	Application should show correct email or password validation message	Working as expected	PASS	Successful					JENITHA
LoginPage_TC_013	UI	Home page for Agent	Verify user is able to see the agent home page when user login on submitting Credentials	1 Enter URL: http://199.51.204.21:53000/ and click go 2 To the Agent Login page and submit Your Credentials	ID: 1111 password: 5678	AGENT Home Page should display	Working as expected	PASS	Successful					PRIVADHARSHINI

LoginPage_TC_011	UI	Home page for USER	Verify user is able to use the User Home page when user login on submitted Credentials	1. Enter URL: http://10.51.204.215:8080/ and click go 2. To the User Login page and submit Your Credentials	http://10.51.204.215:8080/	USER Home Page group should display	Working as expected	PASS	Successful			ATCHAYA
LoginPage_TC_013	UI	Home page for ADMIN	Verify user is able to use the ADMIN Home page when user login on submitted Credentials	1. Enter URL: http://10.51.204.215:8080/ and click go 2. To the User Login page and submit Your Credentials	http://10.51.204.215:8080/	ADMIN Home Page group should display	Working as expected	PASS	Successful			PRITYADHARSHINI
LoginPage_TC_016	Functional	AGENT PAGE	On delete button the user Credentials will be deleted	1. Enter URL: http://10.51.204.215:8080/ and click go 2. To the Admin Page and click on User Credentials	http://10.51.204.215:8080/	ADMIN Home Page group should display	Working as expected	PASS	Successful			PRITYADHARSHINI KORUNALYA

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [CONTAINMENT ZONE ALERTING] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	3	1	2	17
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	40
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2

External	2	3	0	1	6
Fixed	11	2	4	20	40
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	13	12	25	78

### 3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

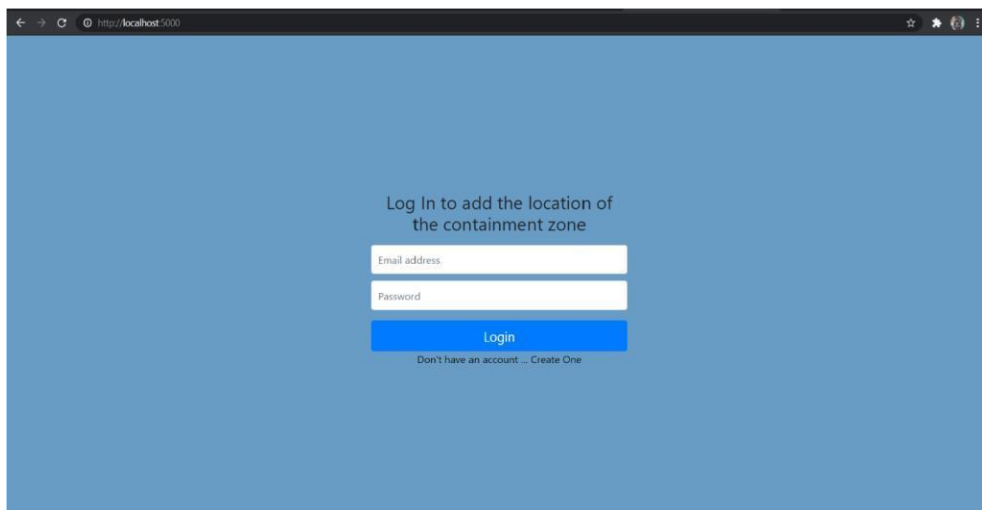
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	50	0	0	50
Security	2	0	0	2

## 9. RESULTS:

### 9.1 Performance Testing:

**Admin App:**

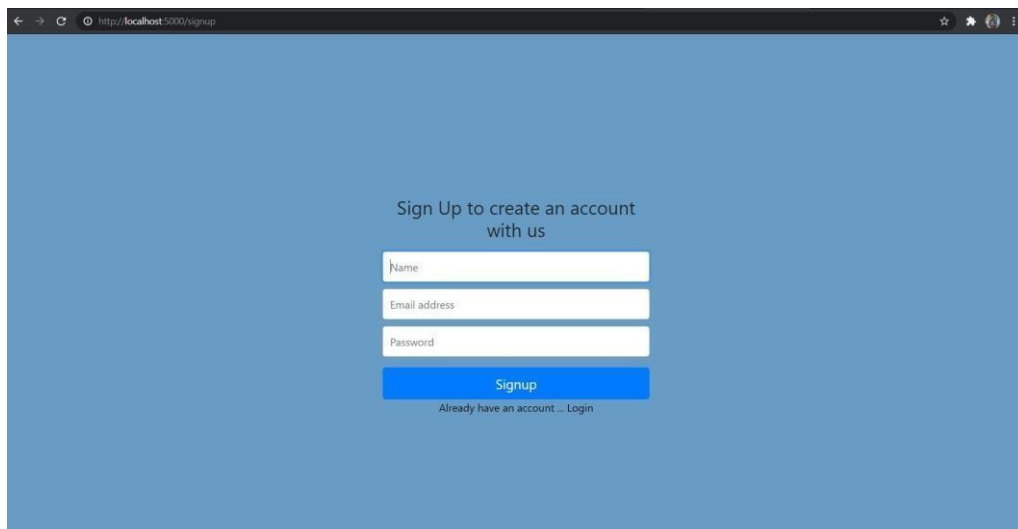
**Login Page:**



The screenshot shows a web browser window with the URL <http://localhost:5000>. The page has a solid blue background. In the center, there is a white login form. The form contains the text "Log In to add the location of the containment zone" above two input fields: "Email address" and "Password". Below these fields is a blue "Login" button. At the bottom of the form, there is a link that says "Don't have an account ... Create One".

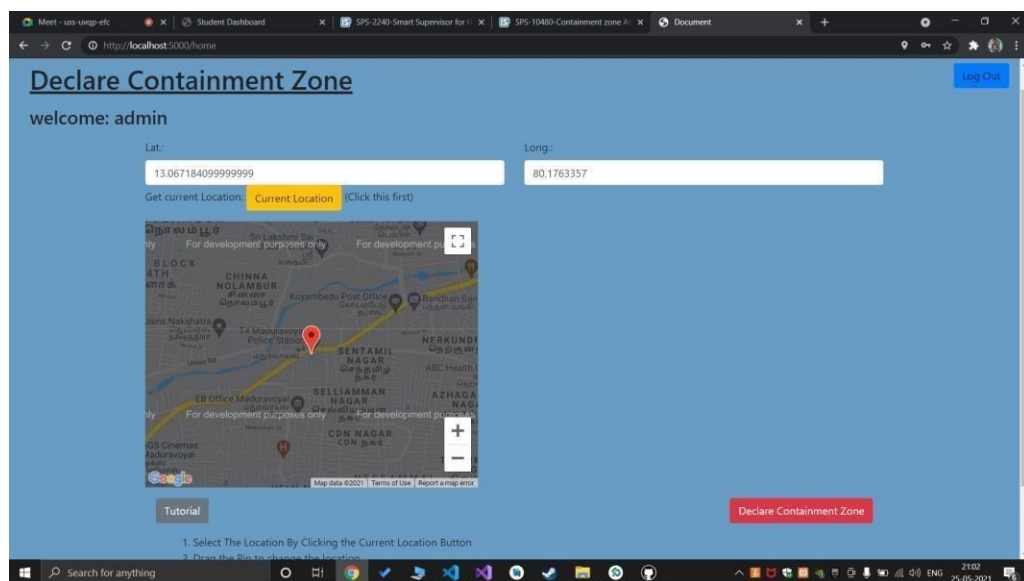


## Register page:



A screenshot of a web browser showing a registration page. The browser's address bar displays 'http://localhost:5000/signup'. The page has a solid blue background. In the center, the text 'Sign Up to create an account with us' is displayed. Below this text are three white input fields with labels: 'Name', 'Email address', and 'Password'. A blue button labeled 'Signup' is positioned below the input fields. Underneath the button, there is a link that says 'Already have an account ... Login'.

## Home page:



A screenshot of a web browser showing a home page for a 'Containment Zone' application. The browser's address bar displays 'http://localhost:5000/home'. The page has a blue header with the title 'Declare Containment Zone' and a 'Log Out' button in the top right corner. Below the header, the text 'welcome: admin' is displayed. There are two input fields for 'Lat.' and 'Long.'. The 'Lat.' field contains the value '13.067184099999999' and the 'Long.' field contains '80.1763357'. Below these fields are buttons for 'Get current Location', 'Current Location' (highlighted in yellow), and '(Click this first)'. A map of a region in India is shown, with a red pin indicating the current location. Below the map is a 'Tutorial' button. At the bottom of the page, there is a red button labeled 'Declare Containment Zone'. The browser's taskbar at the bottom shows various application icons and the system clock indicating 21:02 on 25-05-2021.

PNT2022TMID14431

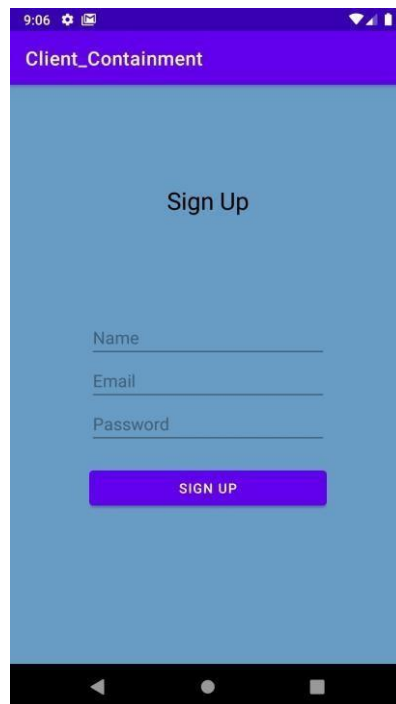
Location data page:



The screenshot shows a web browser window with the address bar displaying 'http://localhost:5000/data'. The page title is 'Location data and Visited People'. It contains a table with 4 columns: S.No, Latitude, Longitude, and No. Visited. There are 7 rows of data. A red button labeled 'Go to location update Page' is located at the bottom right of the table area.

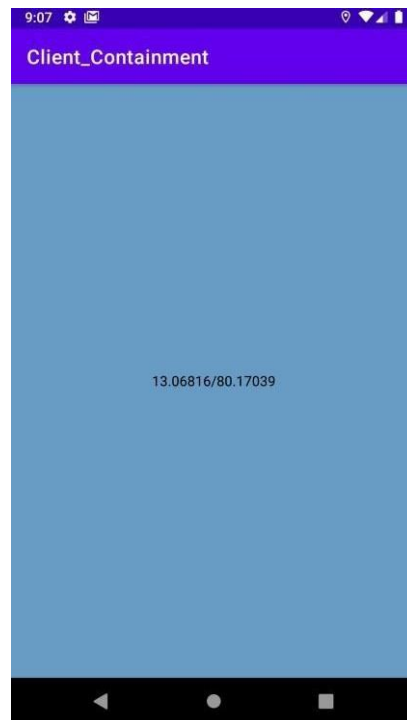
S.No	Latitude	Longitude	No. Visited
1	13.069148883848849	80.17551259999999	0
2	13.068498821079215	80.1704513893799	0
3	12.979174795975714	77.59973092596437	0
4	14.469858338289407	75.91959519903565	0
5	13.062359612480321	77.5638966135254	0
6	15.840542738858232	76.64209647695924	0
7	15.3172775	75.7138884	0

Client Application:  
Register screen:



The screenshot shows a mobile application interface with a purple header bar labeled 'Client\_Containment'. The main screen is light blue and features the text 'Sign Up' in the center. Below this, there are three input fields labeled 'Name', 'Email', and 'Password'. At the bottom, there is a purple button labeled 'SIGN UP'.

Current Location:



An Email will be sent to the registered mail id if the location is within 100 meters of the locations present in the admin app.



## 10. ADVANTAGES & DISADVANTAGES

### ADVANTAGES:

- People can be alerted before entering containment zone.
- Further spread of virus can be reduced considerably.

### DISADVANTAGES:

- Accuracy of application depends on the number of data given to the application.
- Application's accuracy is directly proportional to the number of data given to the application
- about the infected patients.

## 11. CONCLUSION

This application is intended to provide information about containment zones in a particular region by alerting people, through continuous monitoring of an individuals location. Key benefits of

the application are monitoring peoples activity and alerting them to their safety movements.

## 12. FUTURE SCOPE

Although we tried to cover almost all of the aspects during our developmental phase, however we were forced to leave some aspects because of lack of time as well as monetary and other reasons. Just like in the field of software development where there are always some shortcomings and room for improvement our application can be enhanced further:-

- 1) The application can include various government organization to help act faster.
- 2) The dataset obtained from the application can be used for predictive analysis to determine prone areas and include special method for tackling the problem in those areas.
- 3) Emergency signal in case of network failure and internet connection loss.
- 4) Tackling victim's movements.
- 5) Improved Google positioning system's precision.
- 6) The client part of application can be integrated in a single intelligent device.

For analysis purpose, we could use machine learning (ML) algorithms as well as data mining applications. There is a sub branch of machine learning known as time series analysis (TSA), which could be used to predict and analyze the data obtained through this application. Time series analysis is used to predict crop production as well as sales in different quarter.

## 13 APPENDIX

### Source Code

```
# Project : CONTAINMENT ZONE ALERTING APPLICATION  
# Team ID : PNT2022TMID14431
```

#### **APP.PY**

```
from logging import error from flask import *  
from jinja2.utils import select_autoescape import bcrypt  
from flask_mysqldb import MySQL
```

```

import json

from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

# initialization
app = Flask(_name_)

# config
app.secret_key = "\x19Ts\xbe\xe7\x8c_\r\x12Q\x14\x13>q\x7'WTH0\x9f\xe4\xec\xb1"
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ""
app.config['MYSQL_DB'] = 'zone2'
mysql = MySQL(app)

# functions
def send_mail(email):
    print(email)
    message = Mail(from_email='varundutia.h@gmail.com',
to_emails=email,
subject='caution',
plain_text_content='Please Stay Safe',
html_content='<h2>You are entering into a containment Zone</h2>')
    try:
        sg = SendGridAPIClient(
'SG.7BJDtQDIS8unH0r5_TufVQ.Ykpcz19QcqqcNwYZC3a0mNRPhGksG117YURqOTa
2HL')
        response = sg.send(message)
    print(response.status.code)
    print(response.body)
    print(response.headers)
except Exception as e:
    print(e)

def create_bcrypt_hash(password): # convert the string to bytes
    password_bytes = password.encode()
    # generate a salt

```

```

salt = bcrypt.gensalt(14)    # calculate a hash as bytes
password_hash_bytes = bcrypt.hashpw(password_bytes, salt)
# decode bytes to a string
password_hash_str = password_hash_bytes.decode()    return password_hash_str

```

```

def verify_password(password, hash_from_database):
    password_bytes = password.encode()    hash_bytes =
hash_from_database.encode()

    # this will automatically retrieve the salt from the hash,
    # then combine it with the password (parameter 1) # and then hash that, and
    compare it to the user's hash does_match = bcrypt.checkpw(password_bytes,
hash_bytes)

    return does_match

# Api's

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

    # get the data from the form    password =
    request.form['password']    email
    = request.form['email']

```

```

# initialize the cursor
signup_cursor = mysql.connection.cursor()

# check whether user already exists      user_result = signup_cursor.execute(
    "SELECT * FROM USERS WHERE user_email=%s", [email]
)

if(user_result > 0):
    data = signup_cursor.fetchone()      data_password =
data[3]      if(verify_password(password, data_password)):
        signup_cursor.close()      session['id'] =
data[0]      session['name'] = data[1]      session['email'] =
data[2]      return redirect(url_for("home"))      else:
        return render_template('login.html', error=1)      else:
        return      render_template('login.html',      error=2)      return
render_template('login.html', error=3)

@app.route("/signup", methods=["POST", "GET"])

```

```

def verify_password(password, hash_from_database):
    password_bytes = password.encode()
    hash_bytes = hash_from_database.encode()
    # this will automatically retrieve the salt from the hash,
    # then combine it with the password (parameter 1)
    # and then hash that, and compare it to the user's hash
    does_match = bcrypt.checkpw(password_bytes, hash_bytes)
    return does_match
# Api's
@app.route("/", methods=["GET", "POST"])
def login():
    if(request.method == "POST"):
        # get the data from the form
        password = request.form['password']
        email = request.form['email']
        # initialize the cursor
        signup_cursor = mysql.connection.cursor()
        # check whether user already exists
        user_result = signup_cursor.execute(
            "SELECT * FROM USERS WHERE user_email=%s", [email]
        )
        if(user_result > 0):
            data = signup_cursor.fetchone()
            data_password = data[3]
            if(verify_password(password, data_password)):
                signup_cursor.close()
                session['id'] = data[0]
                session['name'] = data[1]
                session['email'] = data[2]
                return redirect(url_for("home"))
            else:
                return render_template('login.html', error=1)
        else:
            return render_template('login.html', error=2)
        return render_template('login.html', error=3)
    @app.route("/signup", methods=["POST", "GET"])

def create_bcrypt_hash(password):
    # convert the string to bytes
    password_bytes = password.encode()
    # generate a salt
    salt = bcrypt.gensalt(14)
    # calculate a hash as bytes
    password_hash_bytes = bcrypt.hashpw(password_bytes, salt)
    # decode bytes to a string
    password_hash_str = password_hash_bytes.decode()
    return password_hash_str
PNT2022TMID14431
def verify_password(password, hash_from_database):
    password_bytes = password.encode()
    hash_bytes = hash_from_database.encode()
    # this will automatically retrieve the salt from the hash,

```



```

# then combine it with the password (parameter 1)
# and then hash that, and compare it to the user's hash
does_match = bcrypt.checkpw(password_bytes, hash_bytes)
return does_match
# Api's
@app.route("/", methods=["GET", "POST"])
def login():
    if(request.method == "POST"):
        # get the data from the form
        password = request.form['password']
        email = request.form['email']
        # initialize the cursor
        signup_cursor = mysql.connection.cursor()
        # check whether user already exists
        user_result = signup_cursor.execute(
            "SELECT * FROM USERS WHERE user_email=%s", [email]
        )
        if(user_result > 0):
            data = signup_cursor.fetchone()
            data_password = data[3]
            if(verify_password(password, data_password)):
                signup_cursor.close()
                session['id'] = data[0]
                session['name'] = data[1]
                session['email'] = data[2]
                return redirect(url_for("home"))
            else:
                return render_template('login.html', error=1)
        else:
            return render_template('login.html', error=2)
        return render_template('login.html', error=3)
    @app.route("/signup", methods=["POST", "GET"])
    def signup():
        if(request.method == "POST"):

            # get the data from the form
            name = request.form['name']
            email = request.form['email']
            password = request.form['password']

            # hash the password
            pw_hash = create_bcrypt_hash(password)

            # initialize the cursor
            signup_cursor = mysql.connection.cursor()

            # check whether user already exists
            user_result = signup_cursor.execute(
                "SELECT * FROM USERS WHERE user_email=%s", [email]
            )
            if(user_result > 0):
                signup_cursor.close()

```

```

        return render_template('signup.html', error=True)
    else:
        # execute the query
        signup_cursor.execute(
            'INSERT INTO USERS(user_name,user_email,user_password,user_type)
VALUES(%s,%s,%s,%s)', (
            name, email, str(pw_hash), "2"
        )
        )

        mysql.connection.commit()
        signup_cursor.close()
        return redirect(url_for('login'))

    return render_template('signup.html', error=False)

@app.route("/home", methods=["POST", "GET"])
def home():
    if(session['id'] == None):
        return redirect(url_for('login'))
    def upload():
        if(request.method == "POST"):
            # get the data from the form
            name = request.json['name']
            email = request.json['email']
            password = request.json['password']
            # hash the password
            pw_hash = create_bcrypt_hash(password)
            # initialize the cursor
            signup_cursor = mysql.connection.cursor()
            # check whether user already exists
            user_result = signup_cursor.execute(
                "SELECT * FROM USERS WHERE user_email=%s", [email]
            )
            if(user_result > 0):
                signup_cursor.close()
                return {'status': 'failure'}
            else:
                # execute the query
                signup_cursor.execute(
                    'INSERT INTO USERS(user_name,user_email,user_password,user_type)
VALUES(%s,%s,%s,%s)', (
                    name, email, str(pw_hash), "1"
                )
                )
                mysql.connection.commit()
                id_result = signup_cursor.execute(
                    'SELECT user_id FROM USERS WHERE user_email = %s', [email]
                )
                if(id_result > 0):
                    id = signup_cursor.fetchone()
                    return {"id": id[0]}

```

```

signup_cursor.close()
return {"status": "failure"}
@app.route("/get_all_users")
def getusers():
    signup_cursor = mysql.connection.cursor()
    # check whether user already exists
    user_result = signup_cursor.execute(
        "SELECT * FROM USERS"

        if(request.method == "POST"):
            # get data
            lat = request.form["lat"]
            lon = request.form["lon"]
    vis = 0
    if(lat == "" or lon == ""):
    return render_template('home.html', name=session['name'], email=session['email'],
id=session['id'], success=0)
    # create a location cursor
    location_cursor = mysql.connection.cursor()
    # Execute the query
    location_cursor.execute(
        'INSERT INTO LOCATION(location_lat,location_long,location_visited) VALUES(%s,%s,%s)', (
        lat, lon, vis
        )
    )
    mysql.connection.commit()
    location_cursor.close()
    return render_template('home.html', name=session['name'], email=session['email'],
id=session['id'], success=True)
    return render_template('home.html', name=session['name'], email=session['email'],
id=session['id'])

@app.route("/logout")
def logout():
    # remove the username from the session if it is there
    session['id'] = None
    session['name'] = None
    session['email'] = None
    return redirect(url_for('login'))
@app.route("/data")
def data():
    if(session['id'] == None):
    return redirect(url_for('login'))
    location_cursor = mysql.connection.cursor()
    # check whether user already exists
    user_result = location_cursor.execute(
        "SELECT * FROM LOCATION"
    )
    if(user_result == 0):
    return render_template("data.html", responses=0)
    else:
    res = location_cursor.fetchall()

```

```

print(res)
return render_template("data.html", responses=res)
@app.route("/android_sign_up", methods=["POST"])
def upload():
    if(request.method == "POST"):
        # get the data from the form
        name = request.json['name']
        email = request.json['email']
        password = request.json['password']
        # hash the password
        pw_hash = create_bcrypt_hash(password)
        # initialize the cursor
        signup_cursor = mysql.connection.cursor()
        # check whether user already exists
        user_result = signup_cursor.execute(
            "SELECT * FROM USERS WHERE user_email=%s", [email]
        )
        if(user_result > 0):
            signup_cursor.close()
            return {'status': 'failure'}
        else:
            # execute the query
            signup_cursor.execute(
                'INSERT INTO USERS(user_name,user_email,user_password,user_type)
                VALUES(%s,%s,%s,%s)', (
                    name, email, str(pw_hash), "1"
                )
            )
            mysql.connection.commit()
            id_result = signup_cursor.execute(
                'SELECT user_id FROM USERS WHERE user_email = %s', [email]
            )
            if(id_result > 0):
                id = signup_cursor.fetchone()
                return {"id": id[0]}
            signup_cursor.close()
            return {"status": "failure"}
        @app.route("/get_all_users")
        def getusers():
            signup_cursor = mysql.connection.cursor()
            # check whether user already exists
            user_result = signup_cursor.execute(
                "SELECT * FROM USERS"
            )
            PNT2022TMID14431
            if(user_result > 0):
                rv = signup_cursor.fetchall()
                row_headers = [x[0] for x in signup_cursor.description]
                json_data = []
                for result in rv:
                    json_data.append(dict(zip(row_headers, result)))
            return json.dumps(json_data)

```

```

@app.route("/post_user_location_data", methods=["POST"])
def post_user_location():
    if(request.method == "POST"):
        # get the data from the form
        lat = request.json['lat']
        lon = request.json['long']
        id = request.json['id']
        ts = request.json['timestamp']
        # initialize the cursor
        user_location_cursor = mysql.connection.cursor()
        # execute the query
        user_location_cursor.execute(
            'INSERT INTO USER_LOCATION(location_lat,location_long,user_id,timestamp)
            VALUES(%s,%s,%s,%s)', (
            lat, lon, id, ts
            )
        )
        mysql.connection.commit()
        return {"response": "success"}
@app.route("/location_data")
def location_data():
    location_cursor = mysql.connection.cursor()
    # check whether user already exists
    user_result = location_cursor.execute(
        "SELECT * FROM LOCATION"
    )
    if(user_result != 0):
        res = location_cursor.fetchall()
        print(res)
        row_headers = [x[0] for x in location_cursor.description]
        json_data = []
        PNT2022TMID14431
        for result in res:
            json_data.append(dict(zip(row_headers, result)))
        return json.dumps(json_data)
    else:
        return {"response": "failure"}
@app.route("/send_trigger", methods=["POST"])
def send_trigger():
    if(request.method == "POST"):
        # get the data from the form
        email = request.json['email']
        location_id = request.json['id']
        location_cursor = mysql.connection.cursor()
        # check whether user already exists
        user_result = location_cursor.execute(
            "SELECT location_visited FROM LOCATION WHERE location_id=%s", [
            location_id
            ]
        )
        if(user_result == 0):
            return {"response": "failure"}
        else:

```

```

res = location_cursor.fetchone()
print(res[0])
visited = res[0]
visited = visited+1
location_cursor.execute(
"UPDATE LOCATION SET location_visited = %s WHERE location_id=%s",
(visited, location_id)
)
mysql.connection.commit()
send_mail(email)
return {"response": "success"}
# main
if __name__ == "__main__":
app.run(host='0.0.0.0', port=5000)

```

## DATA.HTML

```

<!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+Hhvxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous" />
<style>
body {
padding-top: 30px;
padding-bottom: 30px;
background-color: #699cc5;
}
a {
color: black;
}
</style>
</head>
<body>
<div class="m-4 container">
<h1><u>Location data and Visited People</u></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 responses %}
<tr>
<th scope="row">{{loop.index}}</th>
<td>{{row[1]}}</td>
<td>{{row[2]}}</td>
<td>{{row[3]}}</td>
</tr>
{% 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.HTML

```

<!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>Document</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-
Vkoo8x4CGsO3+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous" />
  <style>    body {
padding-top: 30px;
padding-bottom: 30px;
background-color: #699cc5;
}

    a {
color: black;
}
  </style> </head>

<body>

```

```

{% if success == True %}
<script>
    alert("Location Uploaded Successfully");
</script>
{% elif success == 0 %}
<script>
    alert("Enter Proper Location data");
</script>
{% endif %}
<div class="m-3 float-right">
    <button type="button" class="btn btn-primary"><a href={{url_for("logout")}}>Log
Out</a></button>
</div>
<div class="container m-3">
    <h1><u>Declare Containment Zone</u></h1>
</div>
<div class="container m-3">
    <h3>welcome: {{name}}</h3>
</div>
<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>
</form>

```



```

        </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-
+YQ4JLhgyBLPDQt//l+STsc9iw4uQqACwlvpslubQzn4u2UU2UFM80nGisd026JF"        cros
sorigin="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>
</body>
</html>
```

**GitHub Link:**

<https://github.com/IBM-EPBL/IBM-Project-16429-1659614398>

**Video Demo Link:**

[https://youtu.be/YlllqJ\\_1F84](https://youtu.be/YlllqJ_1F84)