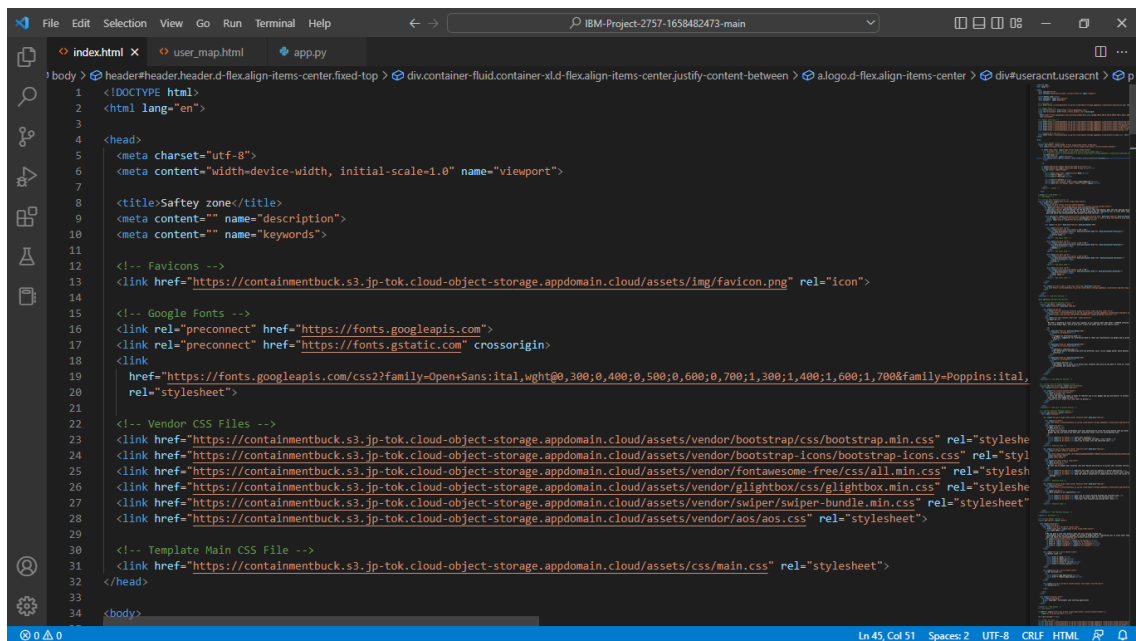


Sprint 3

Get the coordinates from database and plat the area in google map

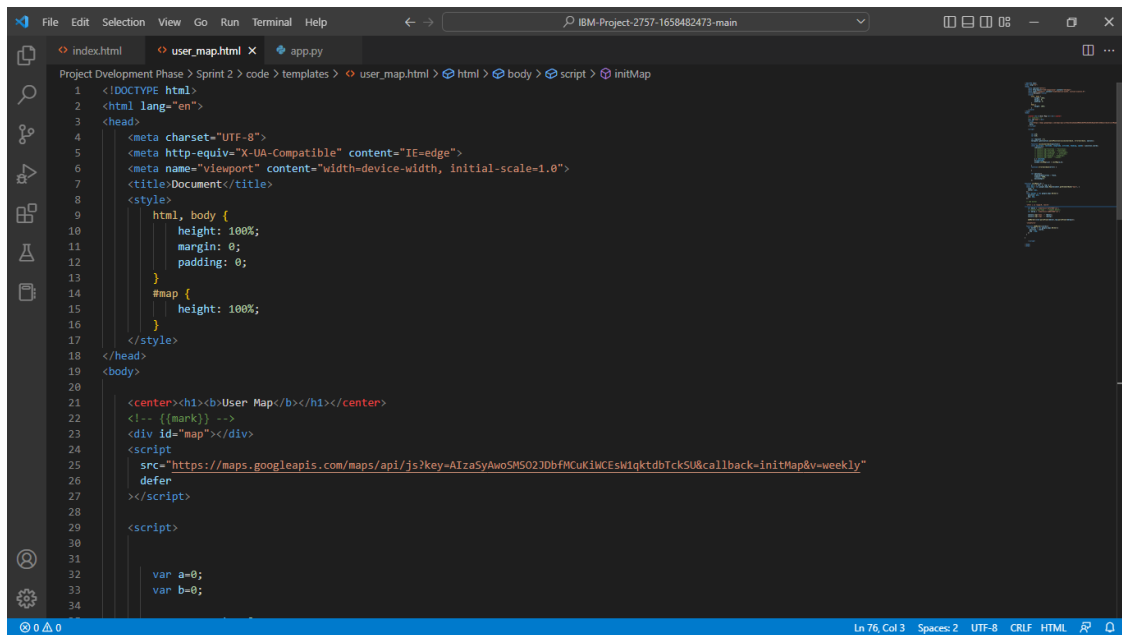
Date	15 NOV 2022
Team ID	PNT2022TMID13289
Project Name	CONTAINMENT ZONE ALERTING APPLICATION
Team Members:	Kishorekumar P, Rohith V, Ramesh S, Mohan P

Screenshot:



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta content="width=device-width, initial-scale=1.0" name="viewport">
7
8   <title>Saftey zone</title>
9   <meta content="" name="description">
10  <meta content="" name="keywords">
11
12  <!-- Favicons -->
13  <link href="https://containmentbuck.s3.jp-tok.cloud-object-storage.appdomain.cloud/assets/img/favicon.png" rel="icon">
14
15  <!-- Google Fonts -->
16  <link rel="preconnect" href="https://fonts.googleapis.com">
17  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
18  <link
19    href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300;0,400;0,500;0,600;0,700;1,300;1,400;1,600;1,700&family=Poppins:ital,
20    rel="stylesheet">
21
22  <!-- Vendor CSS Files -->
23  <link href="https://containmentbuck.s3.jp-tok.cloud-object-storage.appdomain.cloud/assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
24  <link href="https://containmentbuck.s3.jp-tok.cloud-object-storage.appdomain.cloud/assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
25  <link href="https://containmentbuck.s3.jp-tok.cloud-object-storage.appdomain.cloud/assets/vendor/fontawesome-free/css/all.min.css" rel="stylesheet">
26  <link href="https://containmentbuck.s3.jp-tok.cloud-object-storage.appdomain.cloud/assets/vendor/lightbox/css/lightbox.min.css" rel="stylesheet">
27  <link href="https://containmentbuck.s3.jp-tok.cloud-object-storage.appdomain.cloud/assets/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">
28  <link href="https://containmentbuck.s3.jp-tok.cloud-object-storage.appdomain.cloud/assets/vendor/aos/aos.css" rel="stylesheet">
29
30  <!-- Template Main CSS File -->
31  <link href="https://containmentbuck.s3.jp-tok.cloud-object-storage.appdomain.cloud/assets/css/main.css" rel="stylesheet">
32 </head>
33
34 <body>
```

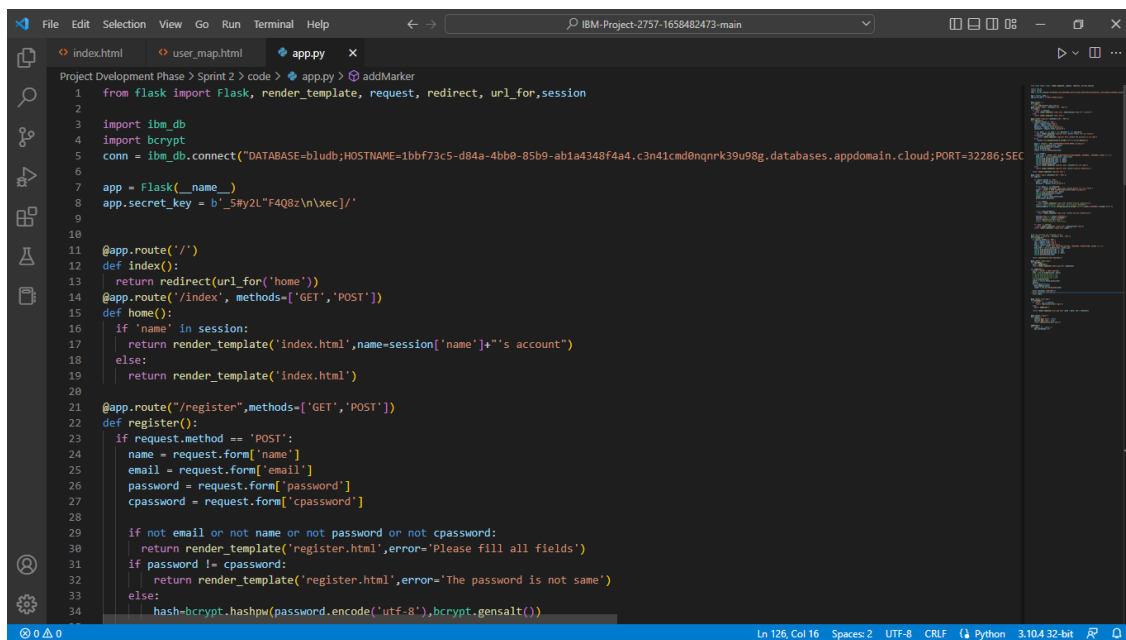
Fig 1. index.html



The screenshot shows a code editor with the file 'user_map.html' open. The code is an HTML document with a dark theme. It includes a head section with meta tags for charset, http-equiv, and viewport, and a title 'Document'. The body contains a CSS style block for a map container, a script tag for Google Maps API, and some initial JavaScript variables.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <style>
9     html, body {
10       height: 100%;
11       margin: 0;
12       padding: 0;
13     }
14     #map {
15       height: 100%;
16     }
17   </style>
18 </head>
19 <body>
20
21   <center><h1><b>User Map</b></h1></center>
22   <!-- {{mark}} -->
23   <div id="map"></div>
24   <script
25     src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAwoSM5O2J0bFmCuKiWCEsWtqktDbTckSU8callback=initMap&v=weekly"
26     defer
27   ></script>
28
29   <script>
30
31     var a=0;
32     var b=0;
```

Fig 2. User Map.html



The screenshot shows a code editor with the file 'app.py' open. The code is a Python Flask application with a dark theme. It imports Flask, render_template, request, redirect, url_for, session, and bcrypt. It defines a Flask app, sets a secret key, and defines routes for index, register, and login. The index route redirects to the home page. The register route handles user registration and password hashing.

```
1 from flask import Flask, render_template, request, redirect, url_for, session
2
3 import ibm_db
4 import bcrypt
5 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnk39u98g.databases.appdomain.cloud;PORT=32286;SEC
6
7 app = Flask(__name__)
8 app.secret_key = b'_5#y2L"F4Q8z\n\xec)/'
9
10
11 @app.route('/')
12 def index():
13     return redirect(url_for('home'))
14 @app.route('/index', methods=['GET', 'POST'])
15 def home():
16     if 'name' in session:
17         return render_template("index.html", name=session['name']+"s account")
18     else:
19         return render_template("index.html")
20
21 @app.route("/register", methods=['GET', 'POST'])
22 def register():
23     if request.method == 'POST':
24         name = request.form['name']
25         email = request.form['email']
26         password = request.form['password']
27         cpassword = request.form['cpassword']
28
29     if not email or not name or not password or not cpassword:
30         return render_template("register.html", error='Please fill all fields')
31     if password != cpassword:
32         return render_template("register.html", error='The password is not same')
33     else:
34         hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())
```

Fig 3. App.py

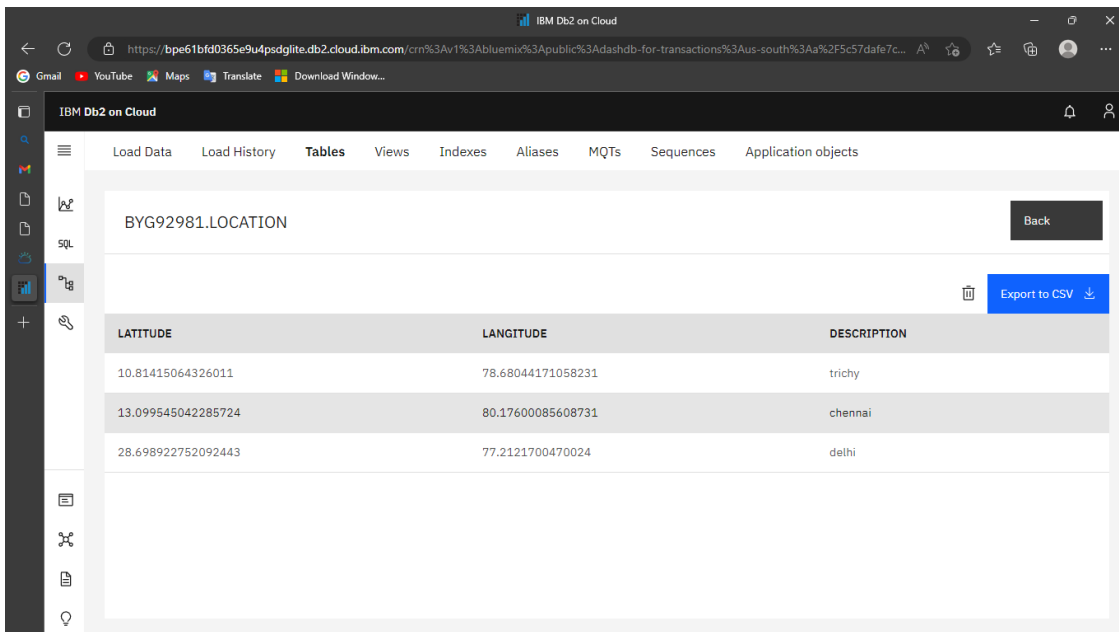
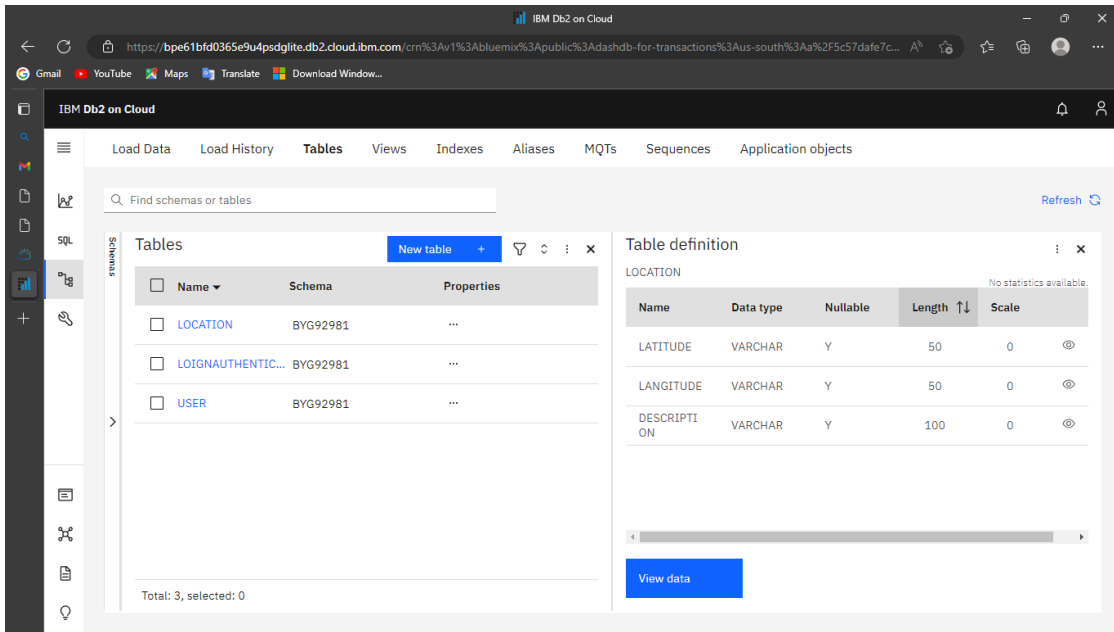


Fig 4. Database

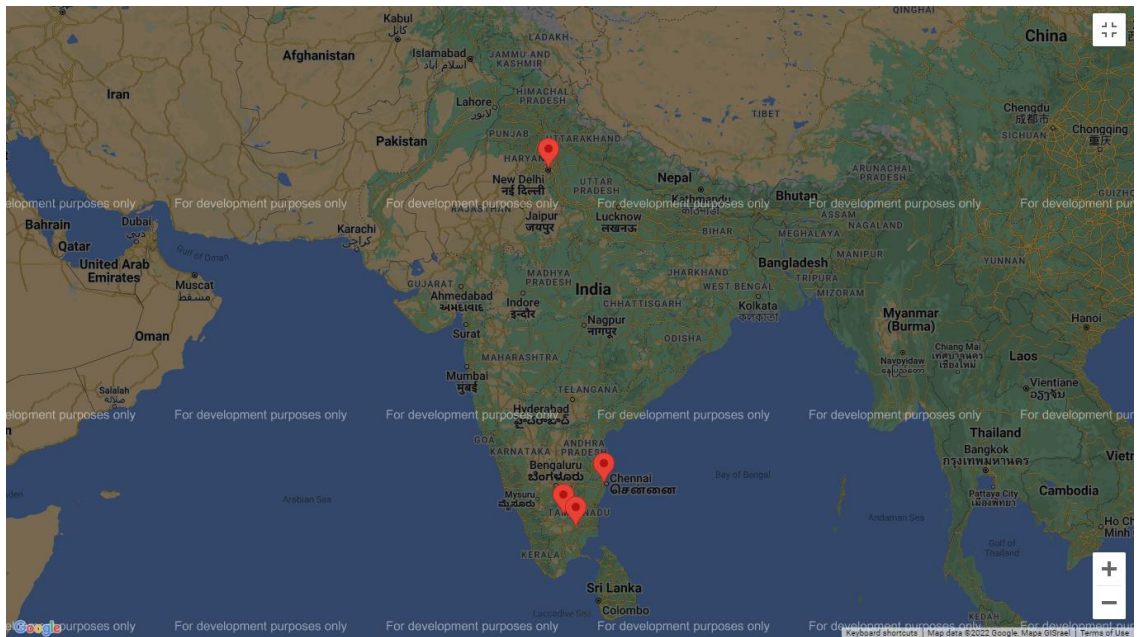


Fig 5. Mark the coordinates taken from db and plot

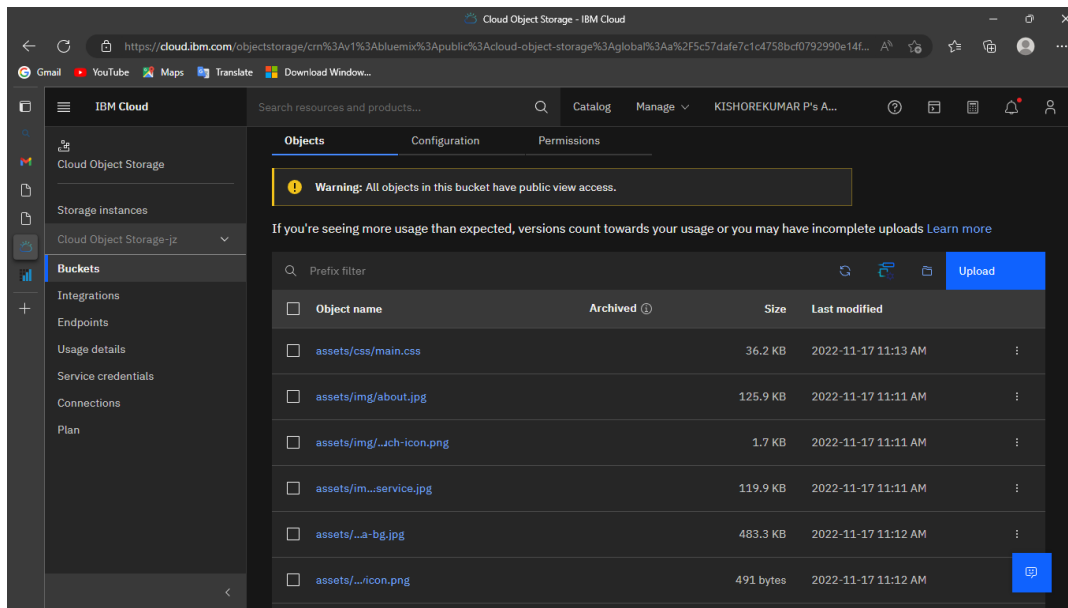


Fig 7. Using object storage

