

Smart Pollution Sensors

Driving Towards Cleaner Air



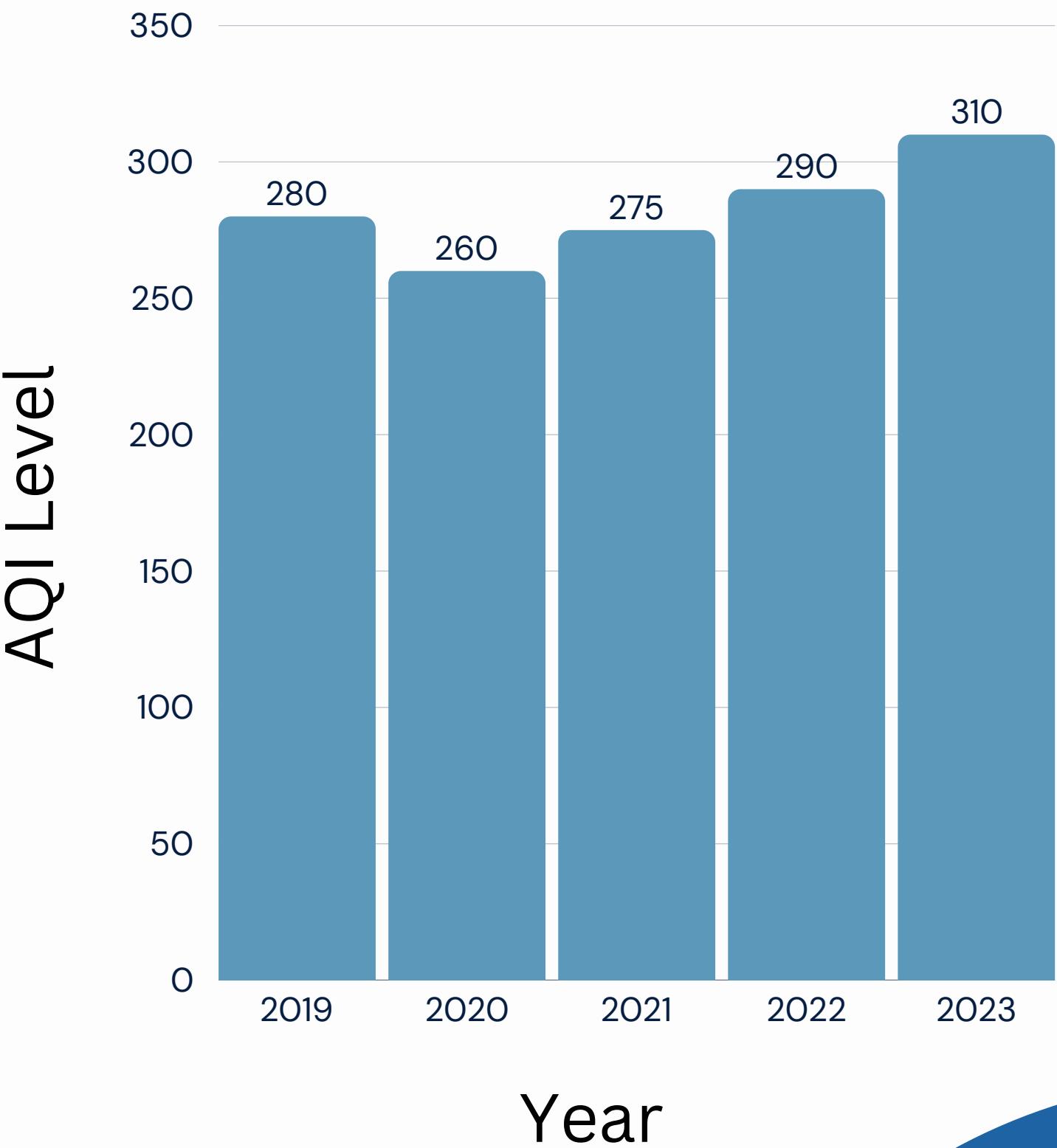
THE PROBLEM STATEMENT

What innovative, technology-driven solutions can be developed and integrated with existing policies to effectively mitigate air pollution in Delhi, ensuring both scalability and long-term sustainability

- Delhi's AQI level often exceeds over 300
- Vehicular emissions contribute 30% of PM2.5 pollution in urban areas.
- Air Pollution has a severe impact on environment, health etc.
- 1.67 million deaths annually in India due to air pollution



Delhi's AQI trends over 5 years



Solution



IoT Pollution Sensors

- Install IoT sensors in vehicles to monitor emissions in real-time
- Track vehicle pollution levels continuously

Mobile App Notifications

- Send alerts to users when maintenance is required based on pollution levels.
- Remind users about PUC (Pollution Under Control) certificate renewal deadlines

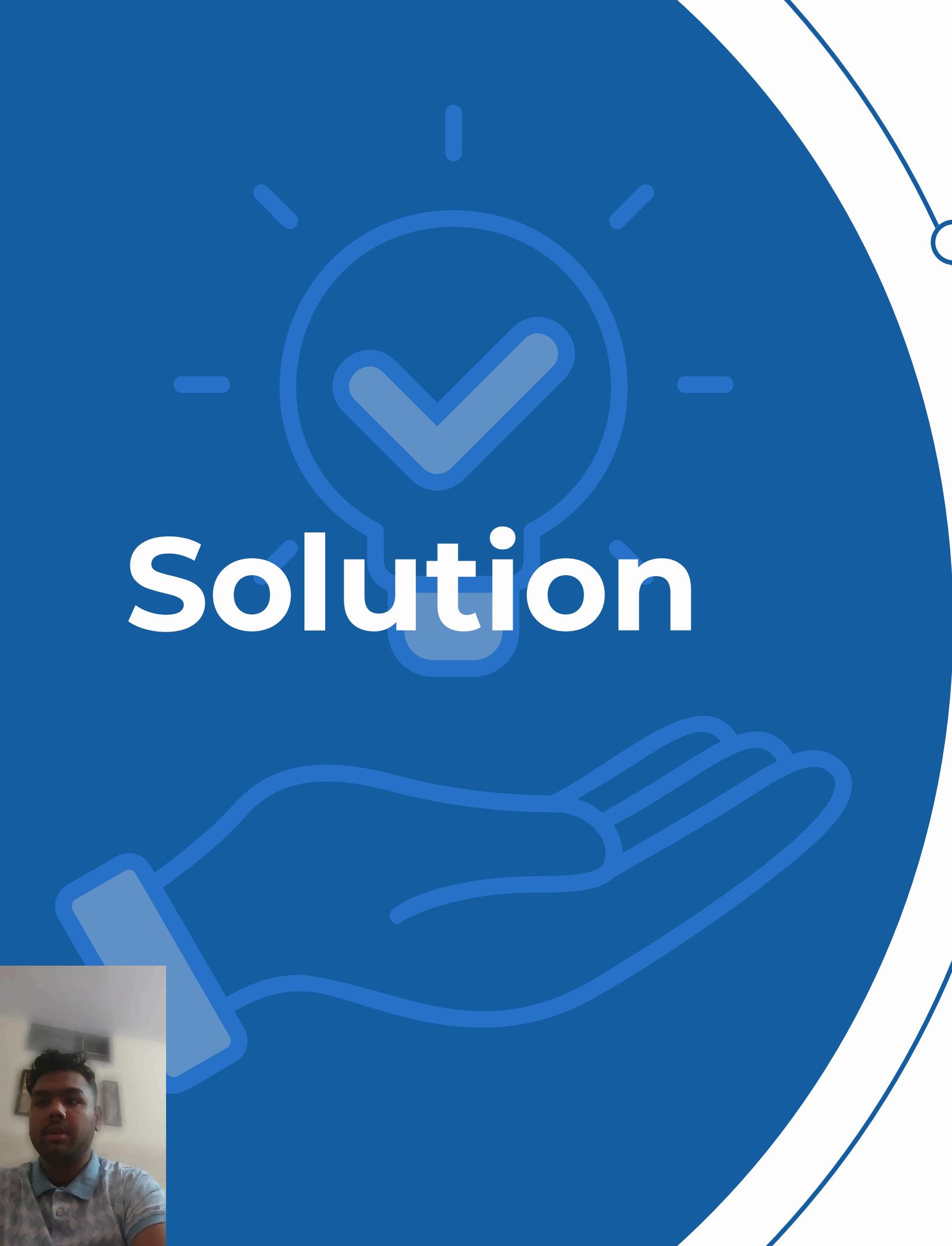
Enforced Corrective Actions

- Enforce corrective actions if maintenance or PUC renewal is ignored.
- Ensure compliance with emission standards.

01

02

03



Solution

IoT Pollution Sensors

- Install IoT sensors in vehicles to monitor emissions in real-time
- Track vehicle pollution levels continuously

Mobile App Notifications

- Send alerts to users when maintenance is required based on pollution levels.
- Remind users about PUC (Pollution Under Control) certificate renewal deadlines

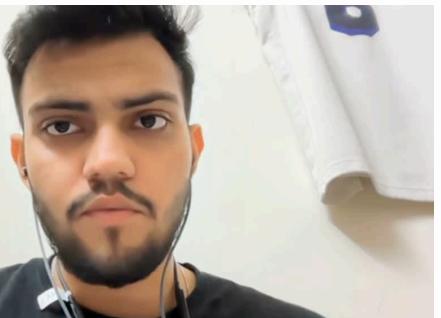
Enforced Corrective Actions

- Enforce corrective actions if maintenance or PUC renewal is ignored.
- Ensure compliance with emission standards.

CODING

```
1  from flask import Flask, request, jsonify
2  from werkzeug.utils import secure_filename
3  import os
4  import uuid
5
6  app = Flask(__name__)
7  app.config['UPLOAD_FOLDER'] = 'uploads'
8  os.makedirs(app.config['UPLOAD_FOLDER'], exist_ok=True)
9
10 # Temporary in-memory storage
11 users = {}
12 vehicles = {}
13 certificates = {}
14
15 @app.route('/register', methods=['POST'])
16 def register_user():
17     data = request.form
18     name = data.get('name')
19     age = data.get('age')
20     email = data.get('email')
21     gender = "Male" # Fixed for this initiative
22
23     if not all([name, age, email]):
24         return jsonify({'error': 'Missing required fields'}), 400
25
26     user_id = str(uuid.uuid4())
27     users[user_id] = {
```

```
27     user_id: {
28         'name': name,
29         'age': age,
30         'email': email,
31         'gender': gender
32     }
33     return jsonify({'message': 'User registered', 'user_id': user_id}), 201
34
35     @app.route('/add_vehicle/', methods=['POST'])
36     def add_vehicle(user_id):
37         if user_id not in users:
38             return jsonify({'error': 'User not found'}), 404
39
40         data = request.form
41         vehicle_number = data.get('vehicle_number')
42         vehicle_type = data.get('vehicle_type')
43         pollution_level = data.get('pollution_level')
44
45         if not all([vehicle_number, vehicle_type, pollution_level]):
46             return jsonify({'error': 'Missing vehicle details'}), 400
47
48         vehicles[user_id] = {
49             'vehicle_number': vehicle_number,
50             'vehicle_type': vehicle_type,
51             'pollution_level': pollution_level
52         }
53     return jsonify({'message': 'Vehicle details added'}), 200
54
```



CODING

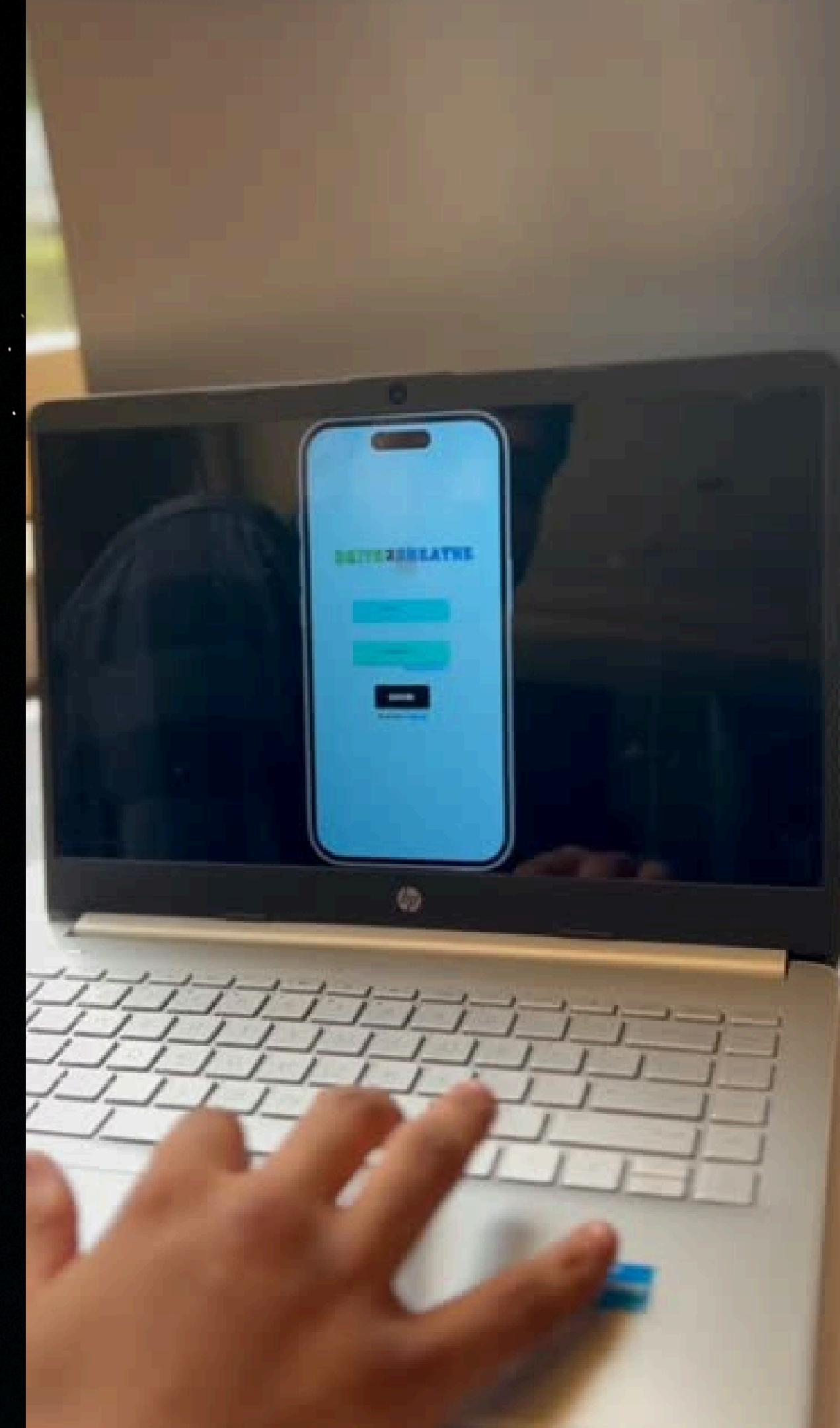
```
54
55     @app.route('/upload_certificate<user_id>', methods=['POST'])
56     def upload_certificate(user_id):
57         if user_id not in users:
58             return jsonify({'error': 'User not found'}), 404
59
60         if 'certificate' not in request.files:
61             return jsonify({'error': 'No file uploaded'}), 400
62
63         file = request.files['certificate']
64         filename = secure_filename(file.filename)
65         file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
66         file.save(file_path)
67
68         certificates[user_id] = filename
69         return jsonify({'message': 'Certificate uploaded', 'filename': filename}), 200
70
71     @app.route('/profile<user_id>', methods=['GET'])
72     def get_profile(user_id):
73         if user_id not in users:
74             return jsonify({'error': 'User not found'}), 404
75
76         profile = {
77             'user_info': users[user_id],
78             'vehicle_info': vehicles.get(user_id, 'No vehicle data'),
79             'certificate_file': certificates.get(user_id, 'No certificate uploaded')
80         }
81
82         return jsonify(profile), 200
```

```
if user_id not in users:
    return jsonify({'error': 'User not found'}), 404

profile = {
    'user_info': users[user_id],
    'vehicle_info': vehicles.get(user_id, 'No vehicle data'),
    'certificate_file': certificates.get(user_id, 'No certificate uploaded')
}
return jsonify(profile), 200

if __name__ == '__main__':
    app.run(debug=True)
```





Our Team

Vansh
Luthra
CSE(UI/UX)

1st Year

Team Leader

Vishesh
CSE(UI/UX)

1st Year

Team Member 1

Kiratpal
Singh
CSE(UI/UX)

1st Year

Team Member 2

Jivansh
Dainwal
CSE(UI/UX)

1st Year

Team Member 3

THANK YOU