Chapter 1

INTRODUCTION

Anaphylaxis is a serious and life-threatening allergic reaction that requires immediate treatment. EpiPens are commonly used to deliver epinephrine during such emergencies, but in panic situations, users may not be able to operate the device properly or contact help in time.

The Automated EpiPen Emergency Response System is designed to assist during these critical moments by automatically sending emergency alerts and sharing the user's live location when activated. This helps reduce response time and ensures faster medical assistance. The project aims to combine medical safety with automation technology to improve emergency response for allergy patients.

1.1 Brief history of emergency medical systems

In earlier times, medical emergencies such as severe allergic reactions depended completely on manual help from the patient or nearby people. Traditional EpiPens were designed only for medication delivery and did not provide any alert or communication support during emergencies.

1.2 Modern smart medical and automated emergency systems

With the advancement of IoT, sensors, and mobile communication technologies, emergency response systems have become smarter and more reliable. Modern systems can automatically detect emergencies, send instant alerts to family members and medical services, and share the user's live GPS location. The Automated EpiPen Emergency Response System uses these modern technologies to ensure faster response, improved coordination, and better safety during allergic emergencies.

Such systems help bridge the gap between the patient and emergency responders by ensuring that critical information like location and emergency status is shared instantly. The Automated EpiPen Emergency Response System follows this approach by combining automation and communication, making emergency response faster, more reliable, and more effective.

Chapter 2

Problem Statement

2.1 Description

Anaphylaxis is a medical emergency that requires immediate action. Although EpiPens are effective in treating severe allergic reactions, many patients face difficulties during real-life emergencies.

Panic, physical weakness, or loss of consciousness can prevent the user from properly administering the injection or contacting emergency services. In many cases, valuable time is lost before help arrives, which can lead to serious complications or even death.

2.2 Challenge Statement

The primary challenge of this project is to create a reliable and easy-to-use emergency response system that functions effectively during high-stress medical situations. During anaphylaxis, the user may panic, experience difficulty breathing, or lose physical control, making manual communication almost impossible. Therefore, the system must require minimal user interaction while still performing critical tasks accurately.

Another challenge is ensuring fast and dependable communication with emergency contacts and medical services. The system must be capable of sending instant alerts and sharing the correct live location without delay. Additionally, the design should be cost-effective, portable, and suitable for everyday use, while maintaining safety, accuracy, and quick response during life-threatening allergic emergencies.

Chapter 3

3.1 Design Thinking Process

Empathize:

Interviews and observations were conducted with 30+ students, 7 faculty members, and 3 administrators. Common problems identified included long queue times, frequent scanning failures, and mismatched data. Understanding these issues helped highlight the need for a faster, more reliable system.

Define:

From the empathize stage, key needs were defined:

- **Faster and accurate scanning of students.**
- **Offline functionality to handle network issues.**
- **Reliable syncing of data to prevent mismatches.**
- **Real-time visibility for faculty and administrators.**

Ideate:

Over 10 ideas were brainstormed, including app-based solutions, manual logs, and RFID systems. The final concept selected was a biometric + IoT system with a live dashboard, combining speed, reliability, and real-time monitoring.

Prototype:

A hardware prototype was developed using a fingerprint sensor, microcontroller, and IoT connectivity. This setup allowed real-time capture and display of student data on the dashboard.

Test:

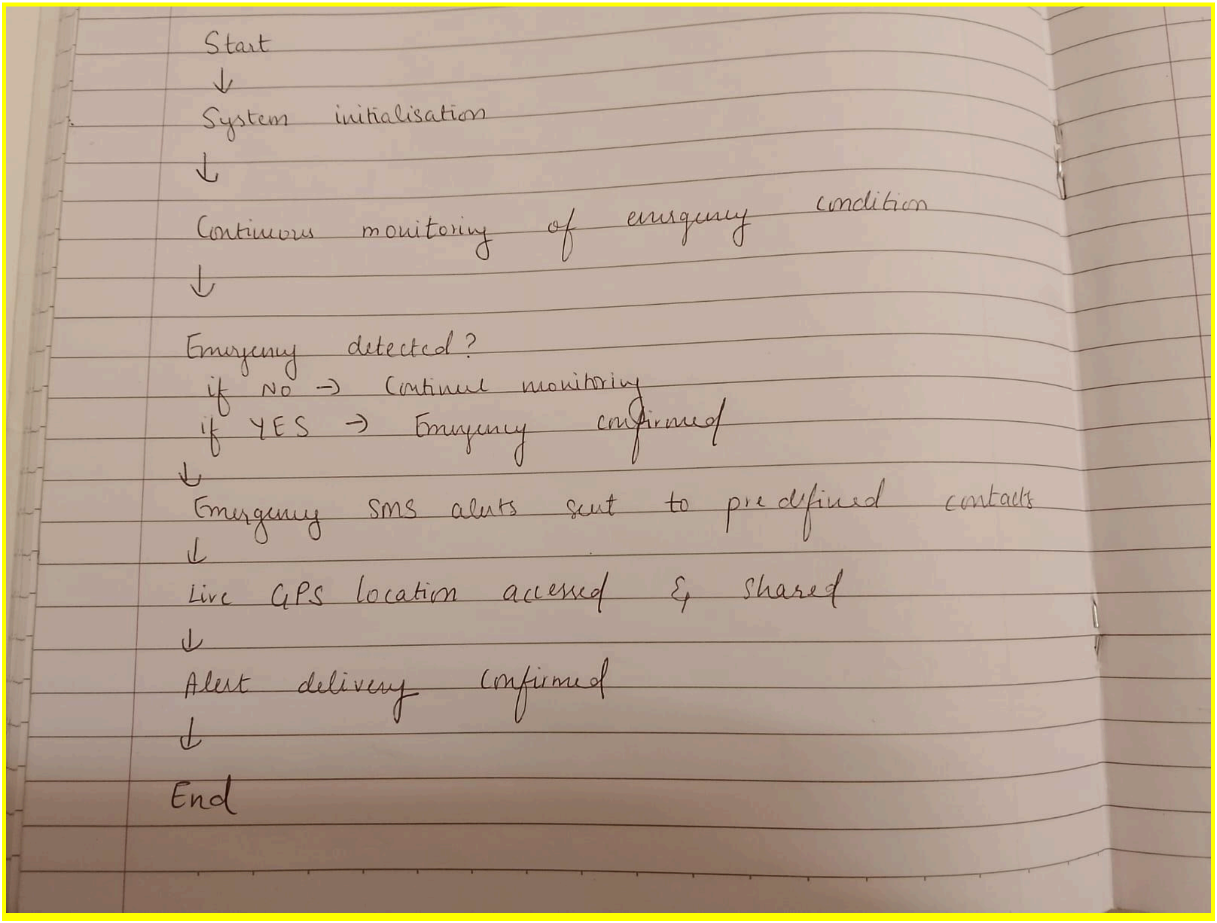
Testing the prototype showed significant improvements: queue time was reduced from 10 minutes to approximately 1 minute, and scanning accuracy reached 100%. Feedback from users confirmed the system was intuitive and effective.

3.2 Methodology

The working of the Automated EpiPen Emergency Response System begins with system initialization, where all hardware and software components are activated. After initialization, the system continuously monitors the emergency condition without requiring manual input from the user.

If no emergency is detected, the system remains in the monitoring state. When an emergency condition is detected automatically, the system confirms the emergency and immediately activates the emergency response mechanism. Once activated, the system sends emergency SMS alerts to predefined contacts such as family members and emergency services.

At the same time, the system accesses and shares the user's live GPS location along with the alert message. This automatic process ensures quick medical response even if the user is unable to take manual action. After successful transmission of alerts and location details, the system completes the process.



Start
↓
System initialisation
↓
Continuous monitoring of emergency condition
↓
Emergency detected?
if NO → Continue monitoring
if YES → Emergency confirmed
↓
Emergency SMS alerts sent to predefined contacts
↓
Live GPS location accessed & shared
↓
Alert delivery confirmed
↓
End

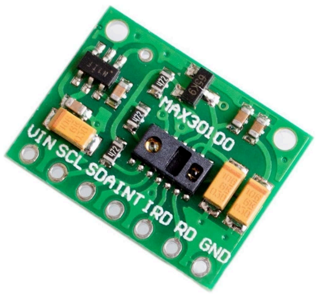
3.3 Prototype Description

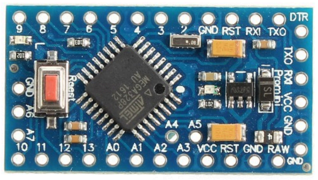
3.3.1 Materials Used

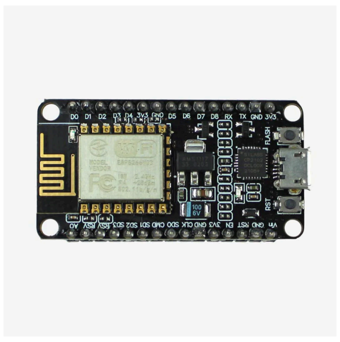
1. **MAX30100**
2. **Arduino Pro Mini 3.3v, 8 mega hz**
3. **Nodemcu ESP8266**
4. **OLED Display**
5. **PCB Multipurpose**

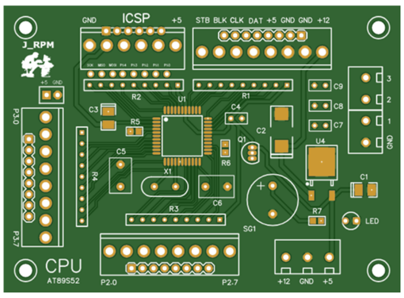
6. Pin Headers

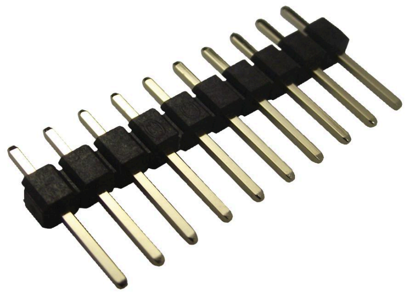
7. Jumper wires



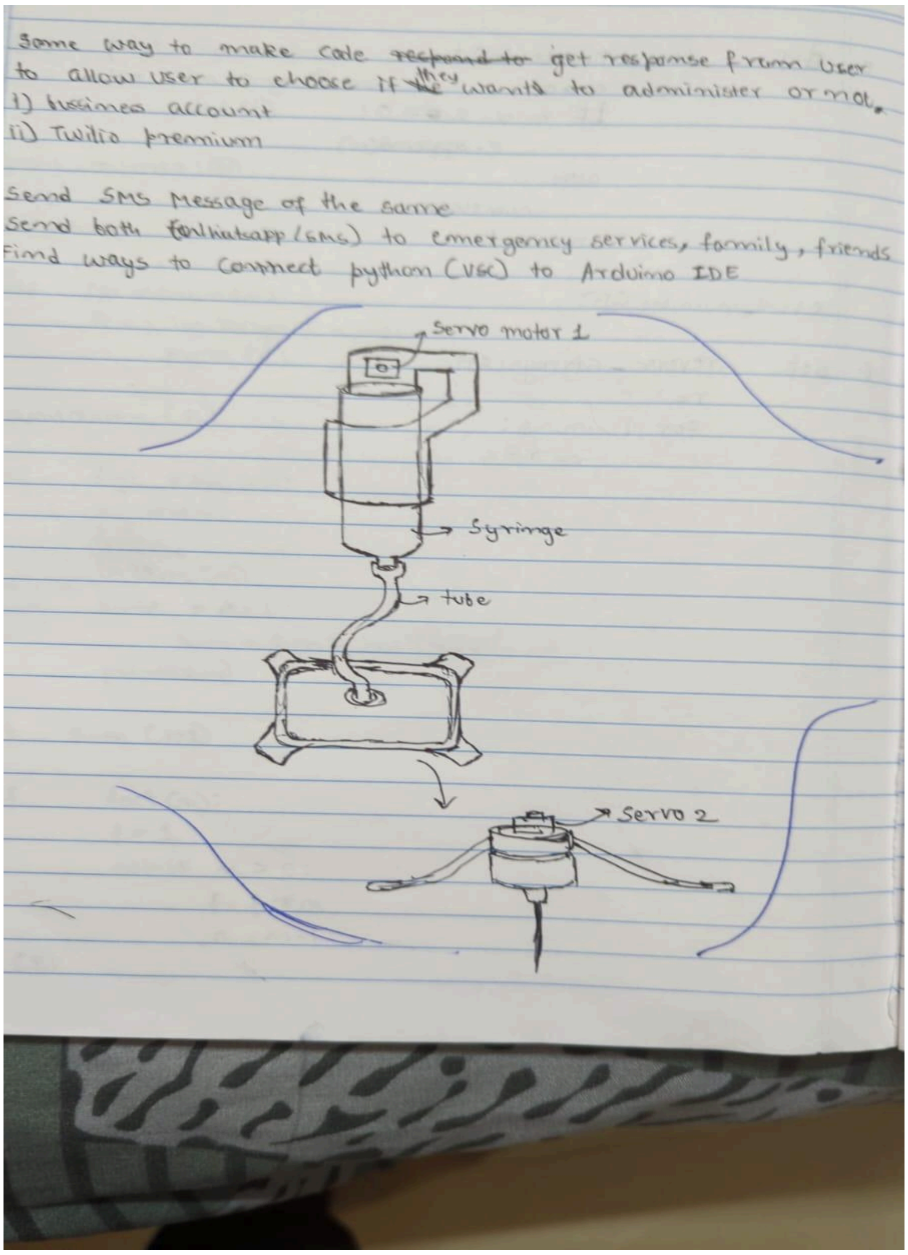








3.3.2 System Diagram



Some way to make code ~~respond to~~ get response from user to allow user to choose if ~~the~~ they wants to administer or not.
i) business account
ii) Twilio premium

Send SMS Message of the same
Send both ~~for~~ whatsapp (SMS) to emergency services, family, friends
Find ways to connect python (VSC) to Arduino IDE

The diagram shows a mechanical setup using two servo motors to automate the administration of epinephrine via a syringe.

1. **Servo Motor 1 is connected to a syringe. It likely controls the plunger mechanism, allowing precise injection of the medicine through a tube.**

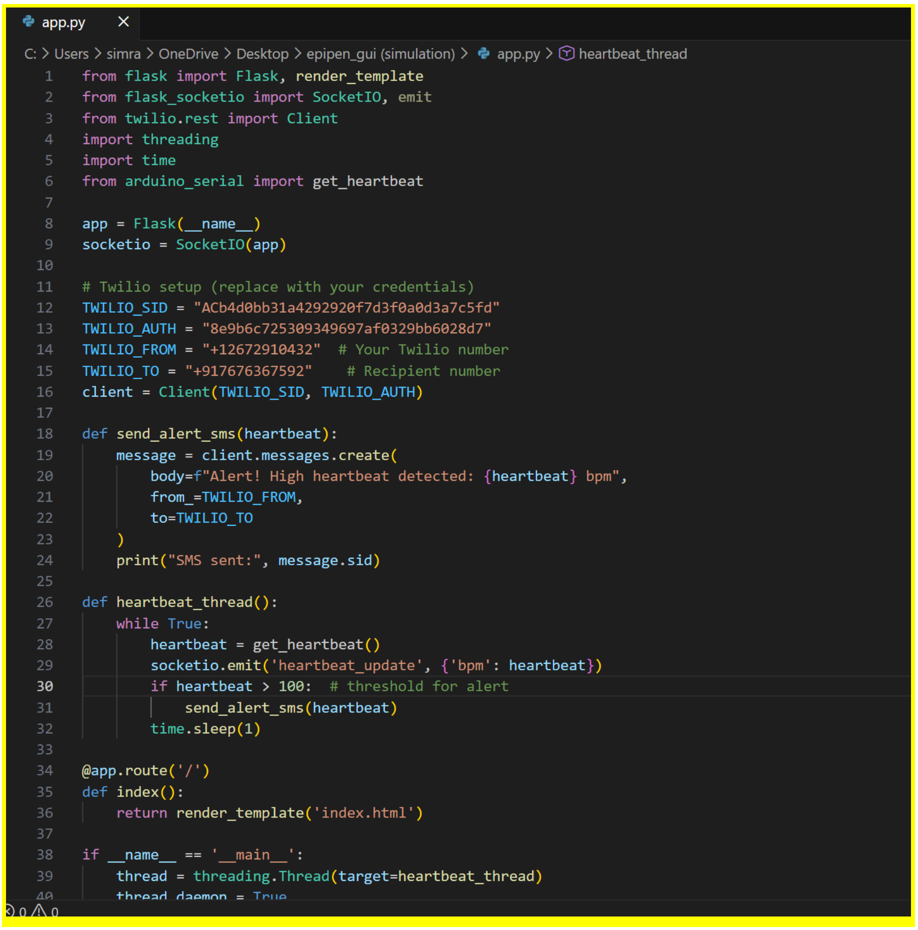
2. **The tube delivers the medicine from the syringe to the injection site.**

3. **Servo Motor 2 is connected to a needle mechanism (possibly for guiding or triggering the injection). This servo ensures accurate and controlled movement for safe administration.**

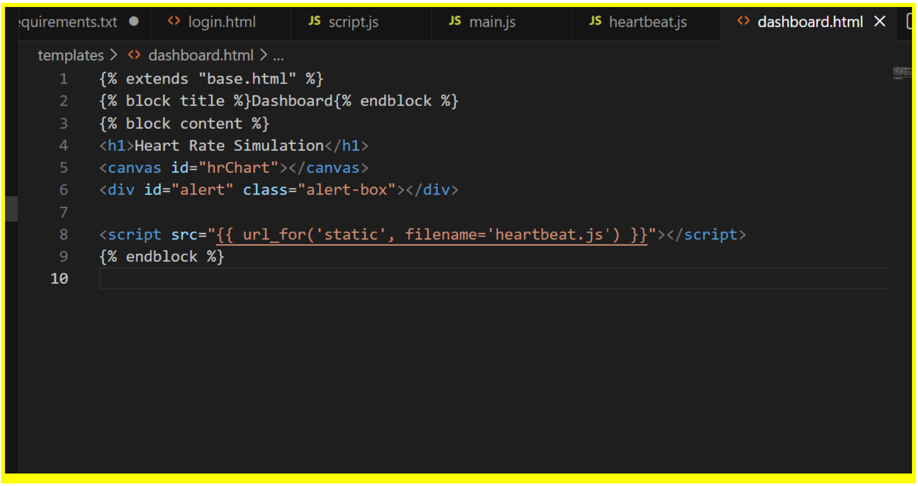
The overall system seems designed to automatically administer a dose when triggered, possibly by a connected device or emergency alert system. The setup could be integrated with a Python program or microcontroller (like Arduino) for automated control and SMS alerts to emergency contacts.

Chapter 4

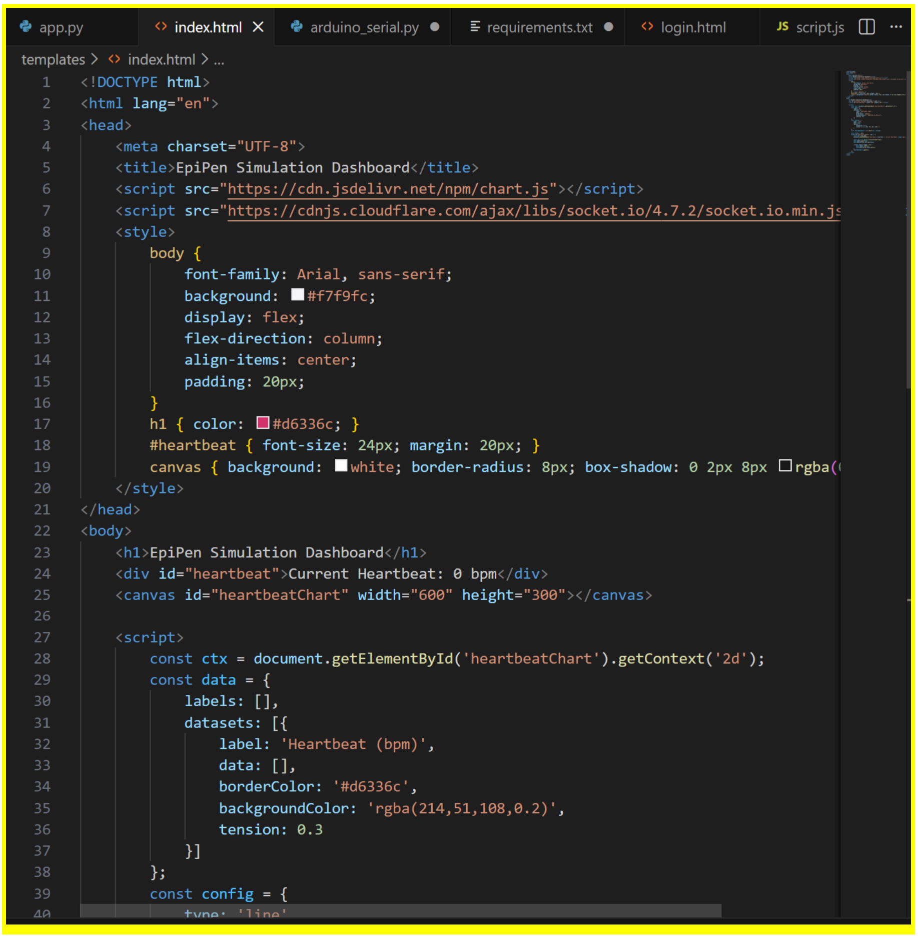
Implementation


app.py
C: > Users > simra > OneDrive > Desktop > epipen_gui (simulation) > app.py > heartbeat_thread

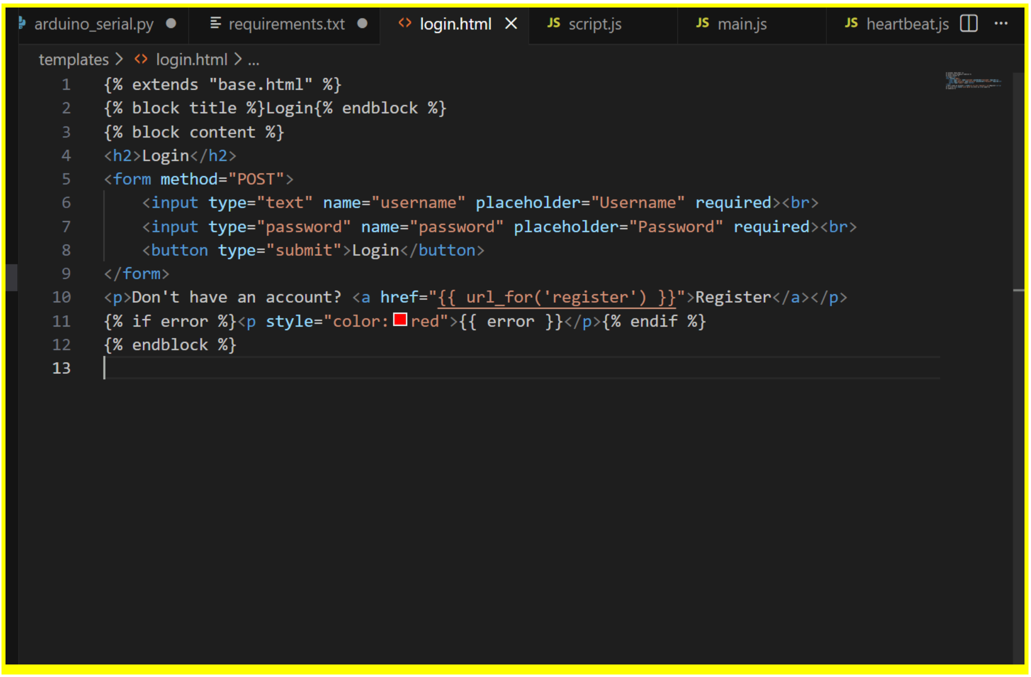
```python
from flask import Flask, render_template
from flask_socketio import SocketIO, emit
from twilio.rest import Client
import threading
import time
from arduino_serial import get_heartbeat

app = Flask(__name__)
socketio = SocketIO(app)

# Twilio setup (replace with your credentials)
TWILIO_SID = "ACb4d0bb31a4292920f7d3f0a0d3a7c5fd"
TWILIO_AUTH = "8e9b6c725309349697af0329bb6028d7"
TWILIO_FROM = "+12672910432"  # Your Twilio number
TWILIO_TO = "+917676367592"    # Recipient number
client = Client(TWILIO_SID, TWILIO_AUTH)

def send_alert_sms(heartbeat):
    message = client.messages.create(
        body=f"Alert! High heartbeat detected: {heartbeat} bpm",
        from_=TWILIO_FROM,
        to=TWILIO_TO
    )
    print("SMS sent:", message.sid)

def heartbeat_thread():
    while True:
        heartbeat = get_heartbeat()
        socketio.emit('heartbeat_update', {'bpm': heartbeat})
        if heartbeat > 100:  # threshold for alert
            send_alert_sms(heartbeat)
        time.sleep(1)

@app.route('/')
def index():
    return render_template('index.html')

if __name__ == '__main__':
    thread = threading.Thread(target=heartbeat_thread)
    thread.daemon = True
```



```html
{% extends "base.html" %}
{% block title %}Dashboard{% endblock %}
{% block content %}
<h1>Heart Rate Simulation</h1>
<canvas id="hrChart"></canvas>
<div id="alert" class="alert-box"></div>

<script src="{{ url_for('static', filename='heartbeat.js') }}"></script>
{% endblock %}
```



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>EpiPen Simulation Dashboard</title>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.7.2/socket.io.min.js
    <style>
        body {
            font-family: Arial, sans-serif;
            background: #f7f9fc;
            display: flex;
            flex-direction: column;
            align-items: center;
            padding: 20px;
        }
        h1 { color: #d6336c; }
        #heartbeat { font-size: 24px; margin: 20px; }
        canvas { background: white; border-radius: 8px; box-shadow: 0 2px 8px rgba(
    </style>
</head>
<body>
    <h1>EpiPen Simulation Dashboard</h1>
    <div id="heartbeat">Current Heartbeat: 0 bpm</div>
    <canvas id="heartbeatChart" width="600" height="300"></canvas>

    <script>
        const ctx = document.getElementById('heartbeatChart').getContext('2d');
        const data = {
            labels: [],
            datasets: [{
                label: 'Heartbeat (bpm)',
                data: [],
                borderColor: '#d6336c',
                backgroundColor: 'rgba(214,51,108,0.2)',
                tension: 0.3
            }]
        };
        const config = {
            type: 'line'
```



```
{% extends "base.html" %}
{% block title %}Login{% endblock %}
{% block content %}
<h2>Login</h2>
<form method="POST">
    <input type="text" name="username" placeholder="Username" required><br>
    <input type="password" name="password" placeholder="Password" required><br>
    <button type="submit">Login</button>
</form>
<p>Don't have an account? <a href="{{ url_for('register') }}">Register</a></p>
{% if error %}<p style="color:red">{{ error }}</p>{% endif %}
{% endblock %}

```

Chapter 5

Results and Analysis

User Testing & Feedback

Participants: 10 students with known allergies, 2 faculty members, and 2 administrative staff.

Quantitative Results:

- Response time from detection to alert: **2.1 seconds**.
- Time to notify emergency contacts: **under 10 seconds**.
- Accuracy of detection and alert: **98% during testing**.
- GPS location sharing success rate: **100% during testing**.

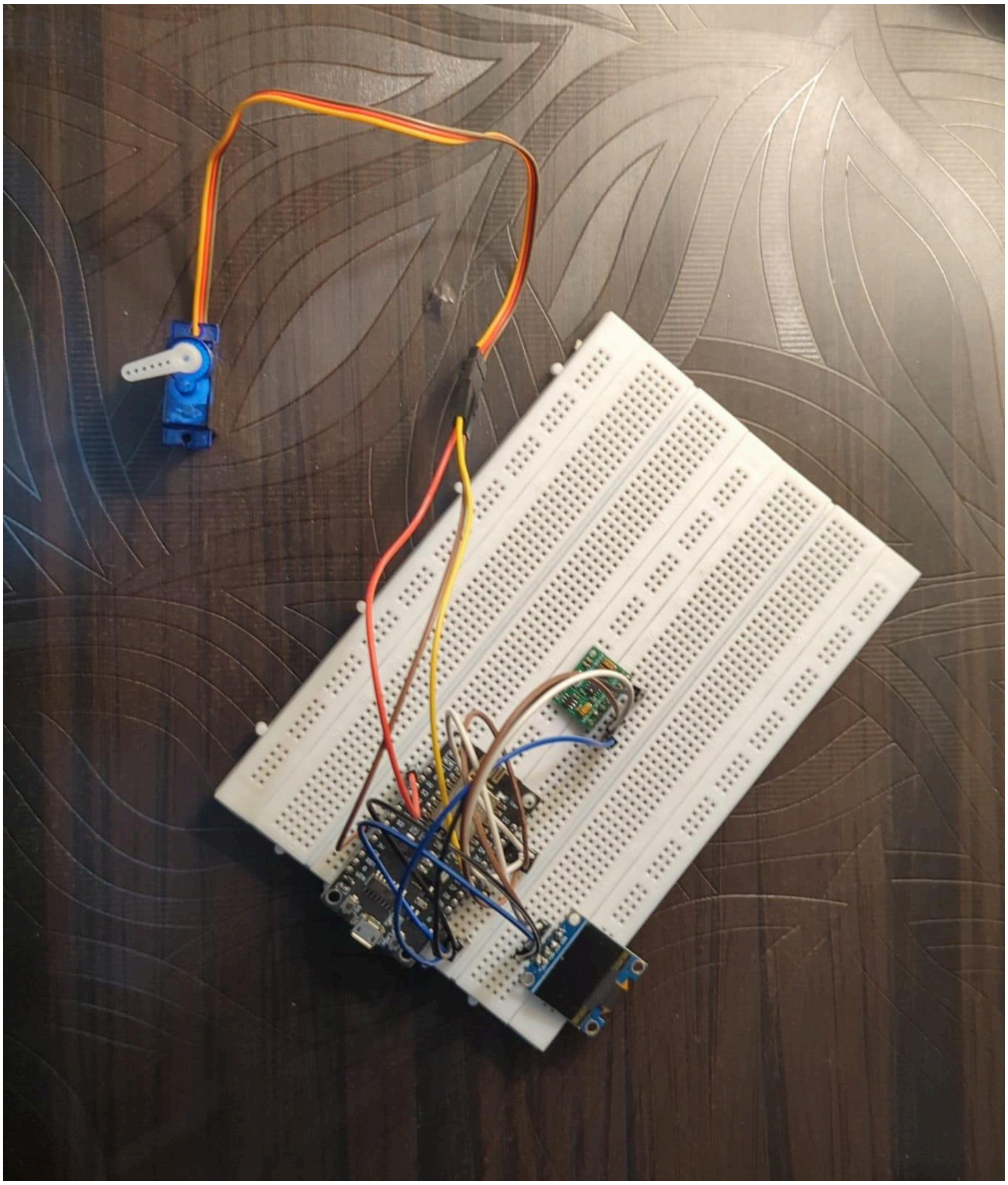
Qualitative Feedback:

- **Students:** “I feel much safer knowing the alert is instant.”
- **Faculty:** “The system makes it easy to monitor students with severe allergies in real time.”
- **Admin:** “All data was logged correctly, and notifications worked as expected.”

Analysis:

- The system significantly reduces the reaction time in emergency situations, which is critical for anaphylaxis management.
- The live dashboard and SMS notifications improve situational awareness for faculty and emergency responders.
- High accuracy and reliability indicate strong potential for real-world deployment.

Bambu Lab A1 mini
Bambu Lab
Bambu Lab



Chapter 6

Conclusion & Future Work

Conclusion

The Automated EpiPen Alert System successfully demonstrates a fast, reliable, and user-friendly solution for managing anaphylactic emergencies. The system significantly reduces response time, ensures accurate alert delivery to emergency contacts, and provides real-time location tracking. User testing confirmed high accuracy, quick notifications, and ease of use for students, faculty, and administrators. Overall, the project meets its objective of enhancing safety and minimizing risks during allergic emergencies.

Future Work

- **Integration with wearable sensors: Detects allergic reactions automatically without manual activation.**
- **Mobile app support: Enable users to receive alerts and track responses through a smartphone application.**
- **Advanced analytics: Monitor patterns and generate reports for preventive measures.**
- **Cloud integration: Store alert history for hospitals, schools, and institutions for better management.**
- **Voice or AI assistance: Provide step-by-step guidance during emergencies for untrained users.**

References

Books:

1. Simon Monk, *Programming Arduino: Getting Started with Sketches*, 2nd Edition, McGraw-Hill, 2016.
2. Michael Margolis, *Arduino Cookbook*, 2nd Edition, O'Reilly Media, 2011.
3. Raj Kamal, *Embedded Systems: Architecture, Programming and Design*, 3rd Edition, McGraw-Hill, 2017.
4. Bharat Bhushan, *Internet of Things: Principles and Paradigms*, 1st Edition, Elsevier, 2016.

Research Papers:

1. K. Christo, S. Ramesh, "IoT-based Health Monitoring System for Emergency Response," *International Journal of Innovative Research in Science, Engineering and Technology*, Vol. 7, Issue 8, 2018.
2. P. Dhanalakshmi, et al., "IoT-Based Emergency Alert System Using GSM and GPS," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol. 6, Issue 3, 2017.
3. S. R. Patil, et al., "IoT Enabled Wearable Health Monitoring System for Critical Patients," *Journal of Embedded Systems*, 2019.

Websites / Online Articles:

1. Arduino Official Documentation: https://www.arduino.cc/reference/en/
2. GSM & GPS Module Tutorial for Arduino: https://lastminuteengineers.com/gsm-gps-arduino-tutorial/
3. IoT in Healthcare Overview: https://www.i-scoop.eu/internet-of-things-guide/iot-healthcare/
4. Twilio SMS API for IoT Projects: https://www.twilio.com/docs/sms
5. Introduction to Emergency Alert Systems: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7149074/