Smart Waste Management System (B2-G2)

Sahana P	1AP21CS055
Chandana H N	1AP22CS400
Tejaswini	1AP21IS040

Overview:

The “**Smart Waste Management System”** is a Raspberry Pi-based project designed to automate and monitor waste management efficiently. It uses ultrasonic sensor to measure the bin's fill level, and moisture sensor detect moisture content of the waste to determine if it is wet or dry, enabling waste segregation and control stepper motors for automatic operations. The system provides real-time data through a web interface.

Software Installation:

1. **Update your Raspberry Pi** sudo apt-get update sudo
   apt-get upgrade

2. **Install Python and pip if not already installed** sudo
   apt-get install python3 python3-pip

3. **Install required Python packages Backend**
   Libraries:

   1. **Flask**: Web framework to host the backend API.
      - **Install**: pip install flask
   2. **Adafruit_MCP3008**: Interface with MCP3008 ADC for analog sensor readings.
      - **Install**: pip install Adafruit-MCP3008
   3. **Adafruit_GPIO**: Manage SPI for ADC communication.

- **Install**: pip install Adafruit-GPIO

4. **RPi.GPIO**: GPIO library for Raspberry Pi.

    - **Install**: pip install RPi.GPIO

Frontend Libraries:

1. **JQuery**: Simplify JavaScript-based interactions.
2. **Bootstrap**: CSS for a responsive, styled UI.
3. **Roboto Font**: Improve design aesthetics.

- Included via CDNs in the HTML files.

4. Create a project directory mkdir smart_waste_management cd smart_waste_management

Project Setup:

Hardware Setup:

1. **Ultrasonic Sensor**: TRIG and ECHO pins connected to GPIO 19 and 26.
2. **Moisture Sensor**: Connected via MCP3008 ADC.
3. **Stepper Motor**: Controlled via GPIO pins 13, 4, 6, and 5.
4. **Buzzer:** GPIO pins 21 and 18, respectively, for alerts.

Software Setup:

1. Clone or place project files:
    - app.py: Backend server code.
    - index.html: Displays sensor data and system status.

2. Start the Flask server: o python app.py
3. Access the dashboard:
    - Open http://<RaspberryPi_IP>:5000 in a web browser.

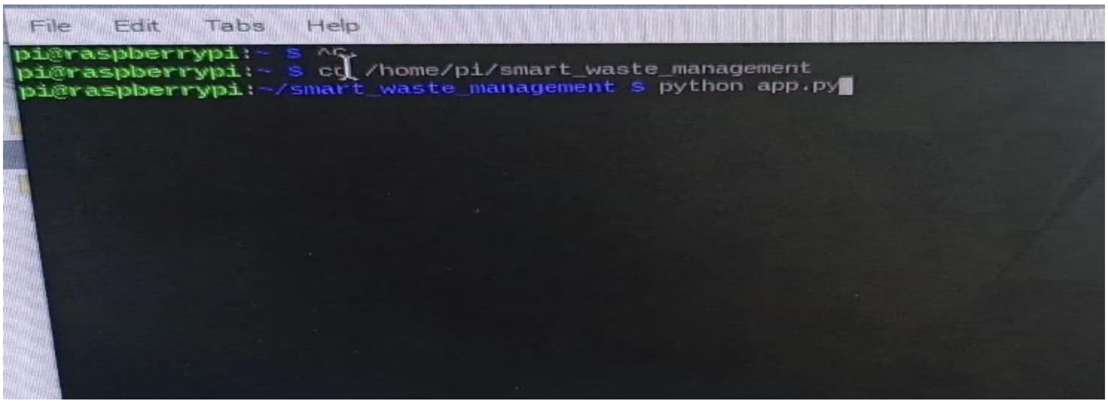
Features:

1. **Real-Time Monitoring**:
    - Displays bin fill level, soil moisture, and alerts.
2. **Alerts**:
    - **Buzzer** : Activates during critical conditions.
3. **Dashboard**:
    - Interactive web interface with live updates.
4. **Automated Control**:
    - Stepper motor adjusts based on moisture levels.

Note:

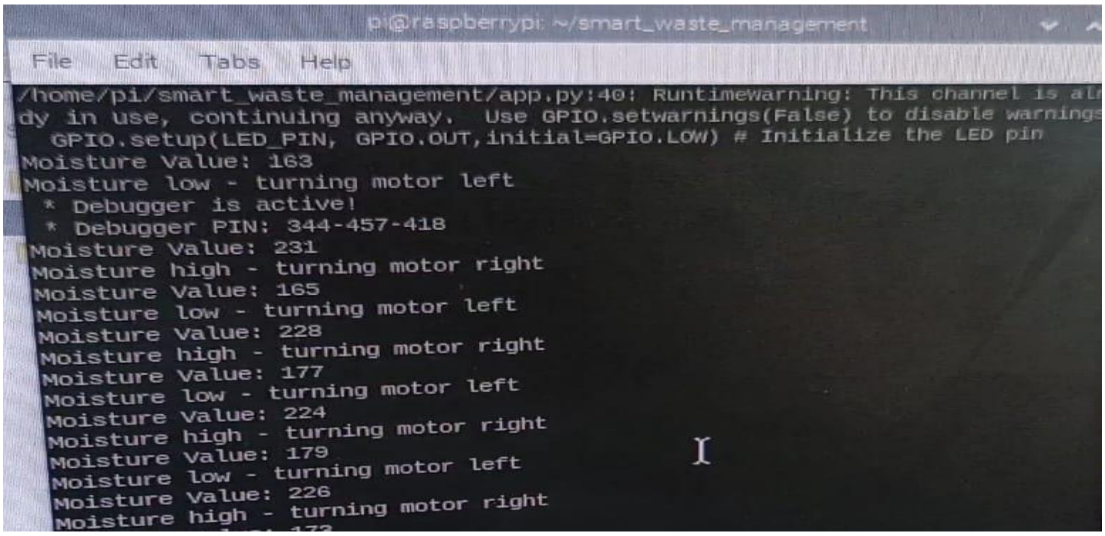
1. Ensure all hardware connections are secure before operation.
2. Replace placeholder credentials before running the system.
3. Test individual components to ensure proper functionality.
4. Keep the bin within the ultrasonic sensor's effective range (2–400 cm).

Snapshots:



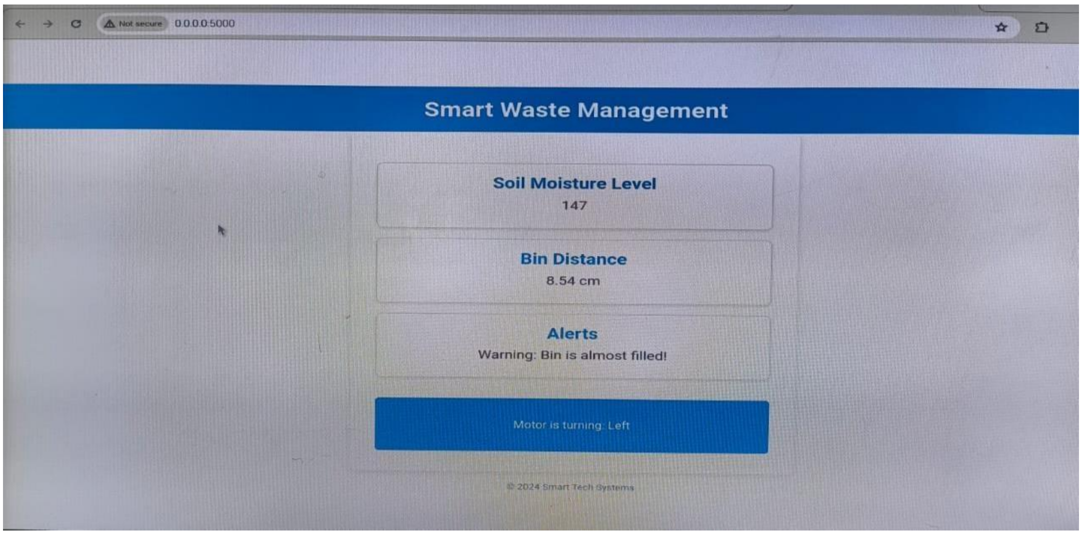
```
pi@raspberrypi:~ $ ^C
pi@raspberrypi:~ $ cd /home/pi/smart_waste_management
pi@raspberrypi:~/smart_waste_management $ python app.py
```

Fig. Project running in terminal



```
/home/pi/smart_waste_management/app.py:40: RuntimeWarning: This channel is alr
dy in use, continuing anyway.  Use GPIO.setwarnings(False) to disable warnings
  GPIO.setup(LED_PIN, GPIO.OUT,initial=GPIO.LOW) # Initialize the LED pin
Moisture Value: 163
Moisture low - turning motor left
 * Debugger is active!
 * Debugger PIN: 344-457-418
Moisture Value: 231
Moisture high - turning motor right
Moisture Value: 165
Moisture low - turning motor left
Moisture Value: 228
Moisture high - turning motor right
Moisture Value: 177
Moisture low - turning motor left
Moisture Value: 224
Moisture high - turning motor right
Moisture Value: 179
Moisture low - turning motor left
Moisture Value: 226
Moisture high - turning motor right
```

Fig. Status of motor rotation



Smart Waste Management

Soil Moisture Level
147

Bin Distance
8.54 cm

Alerts
Warning: Bin is almost filled!

Motor is turning: Left

© 2024 Smart Tech Systems

Fig. The website of the project