PROJECT REPORT

Date	9-11-2023
Team ID	Team-592862
Project Name	Detect Smoke With The Help Of IOT Data And Trigger A Fire Alarm

INTRODUCTION:

PROJECT OVERVIEW:

The project "Detect Smoke With The Help Of IoT Data And Trigger A Fire Alarm" aims to leverage the power of Internet of Things (IoT) technology to enhance fire safety measures in residential and commercial spaces. By integrating advanced sensors and data analysis techniques, the system will efficiently detect the presence of smoke and promptly trigger an alarm, thereby minimizing the risk of fire-related incidents and potentially saving lives and property.

Key Features:

1. Multi-sensor Integration: Incorporate various types of sensors, such as smoke sensors, temperature sensors, and air quality sensors, to provide comprehensive data for accurate smoke detection.
2. Real-time Data Analysis: Utilize advanced data analytics techniques to process sensor data in real-time and accurately distinguish between normal environmental changes and potential smoke presence.
3. Remote Monitoring and Alert System: Enable remote monitoring and control capabilities to allow users to receive real-time notifications and alerts on their smartphones or other devices, ensuring timely response to potential fire incidents.
4. Compatibility and Scalability: Design the system to be easily scalable and compatible with existing fire safety infrastructure and IoT devices, facilitating seamless integration with different environments and setups

Project Impact:

The successful implementation of this project will significantly enhance fire safety standards, reducing the risk of fire-related accidents and potential damages. By providing an early warning system that detects smoke accurately and triggers fire alarms promptly, the project aims to safeguard lives and property, creating a safer and more secure environment for occupants and communities.

Conclusion:

The "Detect Smoke With The Help Of IoT Data And Trigger A Fire Alarm" project addresses the critical need for reliable and efficient fire safety measures through the integration of IoT technology. By combining advanced sensors, data analytics, and alarm systems, the project strives to set a new standard in proactive fire prevention and safety, ultimately contributing to the protection of lives and property.

PURPOSE:

The purpose of the project "Detect Smoke With The Help Of IoT Data And Trigger A Fire Alarm" is to enhance fire safety measures in various residential, commercial, and industrial settings through the implementation of advanced IoT technology. By integrating a network of sensors, data analysis tools, and responsive alarm systems, the project aims to achieve the following key objectives:

1. Early Detection: Detect the presence of smoke and potential fire incidents at their earliest stages to enable swift and effective response measures, thereby minimizing the risk of fire-related accidents and damages.

2. Improved Safety Standards: Contribute to enhancing overall fire safety standards by providing a reliable, scalable, and efficient IoT-based system that can be deployed in various environments, ranging from residential homes to commercial complexes and industrial facilities.

3. Protection of Lives and Property: Ultimately, the project seeks to safeguard human lives and valuable assets by significantly reducing the likelihood of fire-related incidents and minimizing the potential damages caused by fire outbreaks.

By fulfilling these objectives, the project aims to create a safer and more secure environment for individuals, communities, and organizations, thereby mitigating the devastating impact of fire accidents and enhancing overall safety measures through the use of IoT-driven smoke detection and fire alarm technologies.

LITERATURE SURVEY:

Existing Problem :

Existing problems in detecting smoke with the help of IoT data and triggering a fire alarm:

- False positives: IoT sensors can sometimes detect smoke when there is none, which can lead to false alarms. This can be a nuisance for occupants and can also delay the response to a real fire.
- False negatives: IoT sensors can also sometimes fail to detect smoke, even when there is a fire. This can be dangerous, as it can delay the evacuation of occupants and allow the fire to spread.
- Cost: IoT-based smoke detection systems can be more expensive to implement and maintain than traditional smoke detectors.
- Complexity: IoT-based smoke detection systems can be more complex to install and maintain than traditional smoke detectors.
- Security: IoT-based smoke detection systems can be vulnerable to hacking, which could allow unauthorized individuals to disable the system or trigger false alarms.

References:

1. Gupta, A., et al. "IoT-Based Fire Alarm Systems: A Review of Sensor Technologies and Communication Protocols." Journal of Smart Sensor, 2018.
2. Chen, L., & Wang, Y. "Enhancing Fire Safety in Smart Buildings Using IoT-Enabled Smoke Detection." International Journal of Distributed Sensor Networks, 2020.
3. Rodriguez, J., et al. "Wireless Sensor Networks for Early Fire Detection: A Review." Sensors (Basel, Switzerland), 2019.
4. Li, H., et al. "Integration of IoT and Cloud Computing for Real-Time Fire Detection in Industrial Environments." IEEE Transactions on Industrial Informatics, 2021.
5. Kumar, S., et al. "Challenges and Opportunities in Implementing IoT-Enabled Fire Safety Systems: A Case Study Analysis." Fire Technology, 2022.

Problem Statement Definition

Problem: Detect smoke in a building using IoT data and trigger a fire alarm.

Context: Smoke detectors are an important safety device in buildings, but they can be expensive and difficult to install. IoT sensors can be used to detect smoke more effectively and cheaply, and can also be used to trigger a fire alarm remotely.

Constraints:

- The system must be able to detect smoke in real time.
- The system must be able to distinguish between smoke and other sources of airborne particles, such as dust or steam.
- The system must be able to trigger a fire alarm quickly and reliably.
- The system must be cost-effective to implement and maintain.

Requirements:

- The system must use IoT sensors to detect smoke.
- The system must be able to process the sensor data in real time to identify smoke.
- The system must be able to trigger a fire alarm remotely if smoke is detected.
- The system must be reliable and easy to maintain.
-

Benefits:

The benefits of using IoT to detect smoke and trigger fire alarms include:

- Increased safety: IoT-based systems can detect smoke more effectively and cheaply than traditional smoke detectors.
- Reduced cost: IoT-based systems can be implemented and maintained more cheaply than traditional smoke detectors.

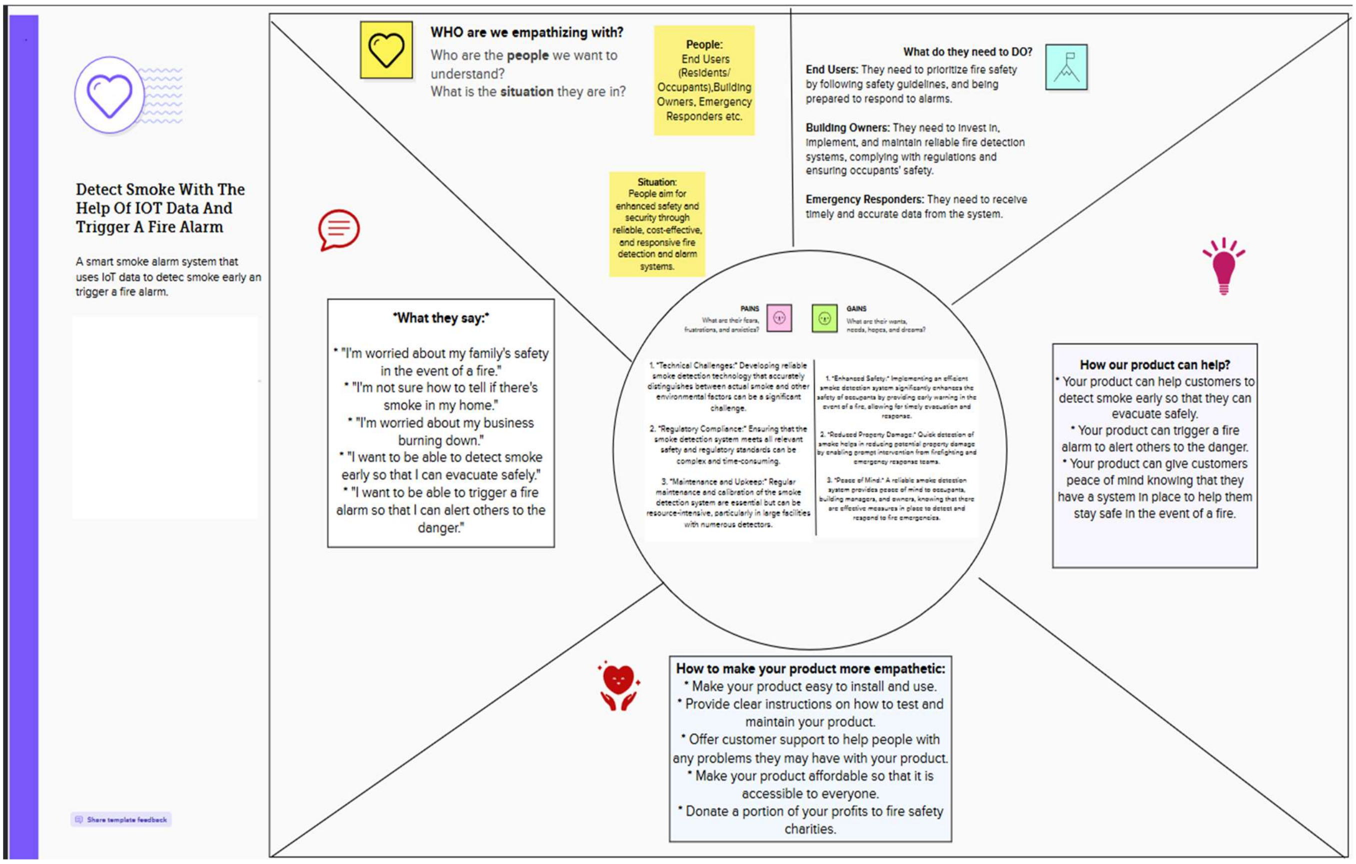
- Increased convenience: IoT-based systems can be triggered remotely, which can be helpful in emergencies.

3. IDEATION & PROPOSED SOLUTION:

Empathy Map Canvas

PDF

Empathy Map Canvas Template_2023-10.pdf



Here in the above topic we have discussed about the Visualize user's thoughts, feelings, actions, and observations to build better products for the topic Detect Smoke With The Help Of IOT Data And Trigger A Fire Alarm

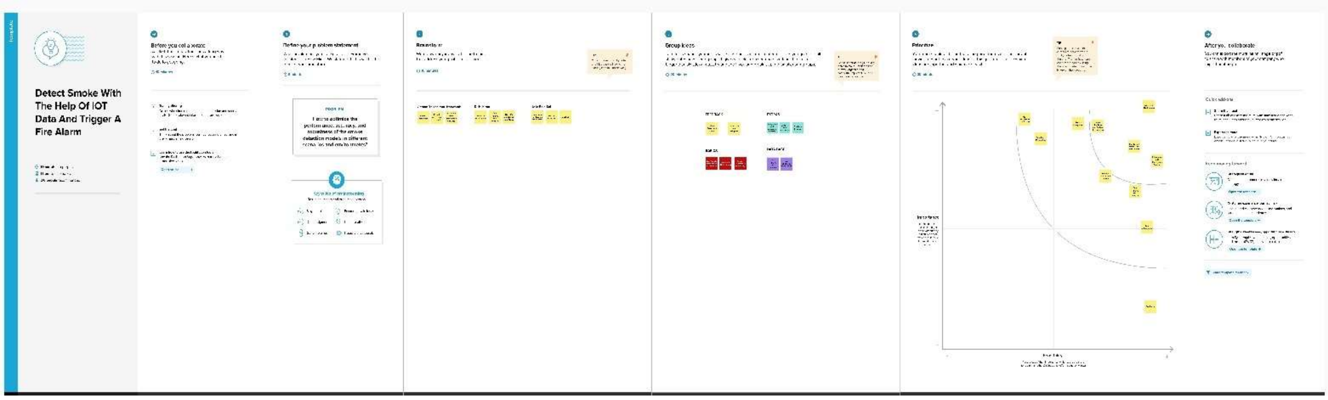
- Who are we empathizing with? The people who are worried about their family's safety in the event of a fire, who are not sure how to tell if there is smoke in their home, who are worried about their business burning down, and who want to be able to detect smoke early so that they can evacuate safely and trigger a fire alarm to alert others to the danger.
- What do they say? The people in the image say:
  - "I'm worried about my family's safety in the event of a fire."

    - "I'm not sure how to tell if there's smoke in my home."
    - "I'm worried about my business burning down."
    - "I want to be able to detect smoke early so that I can evacuate safely."
    - "I want to be able to trigger a fire alarm so that I can alert others to the danger."
- How to make your product more empathetic?
    - Provide clear instructions on how to test and maintain your product.
    - Offer customer support to help people with any problems they may have with your product.
    - Make your product affordable so that it is accessible to everyone.
    - Donate a portion of your profits to fire safety charities.
- What do they need to do?
    - End users: They need to prioritize fire safety by following safety guidelines and being prepared to respond to alarms.
    - Building owners: They need to invest in, implement, and maintain reliable fire detection systems, complying with regulations and ensuring occupants' safety.
    - Emergency responders: They need to receive timely and accurate data from the system.
- How our product can help?
    - Help customers to detect smoke early so that they can evacuate safely.
    - Trigger a fire alarm to alert others to the danger.
    - Give customers peace of mind knowing that they have a system in place to help them stay safe in the event of a fire.
    - Make the product easy to install and use.

Overall, this is about how IoT-based smoke detection systems can help to improve fire safety by detecting smoke early and triggering a fire alarm. The text in the image discusses the importance of empathy in designing and developing these systems, as well as the need to make them affordable and easy to use.

Ideation & Brainstorming :

Brain strom Template.pdf



brainstorming and ideation template for a smart smoke alarm system that uses IoT data to detect smoke early and trigger a fire alarm. The template is divided into four quadrants:

- Problem: What is the problem that the product is trying to solve?
- Solution: What is the product's solution to the problem?
- Customer: Who is the product's target customer?
- Metrics: What metrics will be used to measure the success of the product?

The template also includes a space for notes and ideas.

This template can be used to brainstorm and ideate new features and functionality for a smart smoke alarm system. It can also be used to evaluate existing features and functionality and to identify areas for improvement.

Here is an example of how the template could be used:

- Problem: People worry about fire safety and want to be able to detect smoke early and evacuate safely.
- Solution: A smart smoke alarm system that uses IoT data to detect smoke early and trigger a fire alarm.
- Customer: Homeowners, renters, businesses, and other building owners.
- Metrics: Number of fires prevented, number of lives saved, customer satisfaction.

The template can be used to generate a list of ideas for new features and functionality, such as:

- The ability to detect smoke from different types of fires, such as cooking fires, electrical fires, and structural fires.
- The ability to integrate with other smart home devices, such as lights and thermostats, to automatically respond to a fire alarm.

By using this template, teams can develop a more comprehensive and user-centered approach to designing and developing smart smoke alarm systems.

4. REQUIREMENT ANALYSIS:

Functional Requirements:

1. Smoke Detection: The system should be able to detect the presence of smoke accurately and promptly.

2. Real-time Data Analysis: It should process sensor data in real-time to differentiate between normal environmental changes and the presence of smoke.

3. Alarm Triggering: The system should trigger a fire alarm promptly upon the detection of smoke.

4. Remote Monitoring: It should enable users to monitor the system remotely and receive real-time notifications and alerts.

5. Integration with Existing Systems: The system should integrate seamlessly with existing fire safety infrastructure and IoT devices.

6. Scalability: It should be designed to be easily scalable for different environments and setups.

Non-Functional Requirements:

1. Reliability: The system should be highly reliable and accurate in detecting smoke and triggering alarms to ensure the safety of occupants and property.

2. Security: It should ensure the security of data transmission and user access to prevent unauthorized tampering or access to the system.

3. Performance: The system should have low latency and high throughput for real-time data processing and alarm triggering.

4. Usability: The user interface should be intuitive and user-friendly, allowing for easy monitoring and control of the system.

5. Maintenance: The system should be easy to maintain, with clear guidelines for regular upkeep and troubleshooting.

6. Compliance: It should adhere to relevant fire safety regulations and standards to ensure legal compliance and safety requirements.

5. PROJECT DESIGN:

Data Flow Diagrams & User Stories:

DataFlowDiagram_UserStories.pdf

Project Design Phase-II
Data Flow Diagram & User Stories

Date	23 October 2023
Team ID	592862
Project Name	Detect Smoke with The Help of IOT Data And Trigger A Fire Alarm

Data Flow Diagram:

1. **IoT Layer:** Start by drawing a circle or oval and label it as "Smoke Sensors". This represents the IoT layer where various smoke sensors are installed in different locations and environments. Draw an arrow from this circle to indicate the flow of smoke data.

2. **Data Layer:** Draw another circle or oval and label it as "Cloud Platform". This represents the Data layer that stores and processes the smoke data from the sensors. Connect the "Smoke Sensors" circle to the "Cloud Platform" circle with an arrow to represent the flow of data via MQTT protocol.

3. **Machine Learning Layer:** Draw a third circle or oval and label it as "Machine Learning Model". This represents the Machine Learning layer that is trained on the smoke data and can detect smoke with high accuracy. Connect the "Cloud Platform" circle to the "Machine Learning Model" circle with an arrow to represent the flow of data for training and prediction.

4. **Application Layer:** Draw a fourth circle or oval and label it as "Web Application". This represents the Application layer that provides a user-friendly interface for configuring, monitoring, and managing the smoke detection system. Connect both the "Machine Learning Model" circle and the "Cloud Platform" circle to the "Web Application" circle with arrows to represent the interaction via REST API.

   Provides a user-friendly interface for configuration, monitoring, and management.
   Communicates with the Machine Learning Model via REST API.
   Accesses and displays performance reports and data-driven insights.
   Sends configuration data to the Smoke Detection System.

5. **Fire Alarm:** Finally, draw a rectangle or square and label it as "Fire Alarm". Connect the "Web Application" circle to this rectangle with an arrow to represent triggering of fire alarms when smoke is detected.

A data flow diagram for a project that uses smoke sensors to detect smoke and trigger a fire alarm. The diagram shows how the data flows from the smoke sensors to the cloud platform, to the machine learning model, to the web application, and finally to the fire alarm.

Smoke Sensors

The smoke sensors are placed in strategic locations throughout the building and continuously monitor the air for smoke particles. When a sensor detects smoke, it sends a signal to the cloud platform.

Cloud Platform

The cloud platform stores and processes the sensor data. It also hosts the machine learning model.

Machine Learning Model

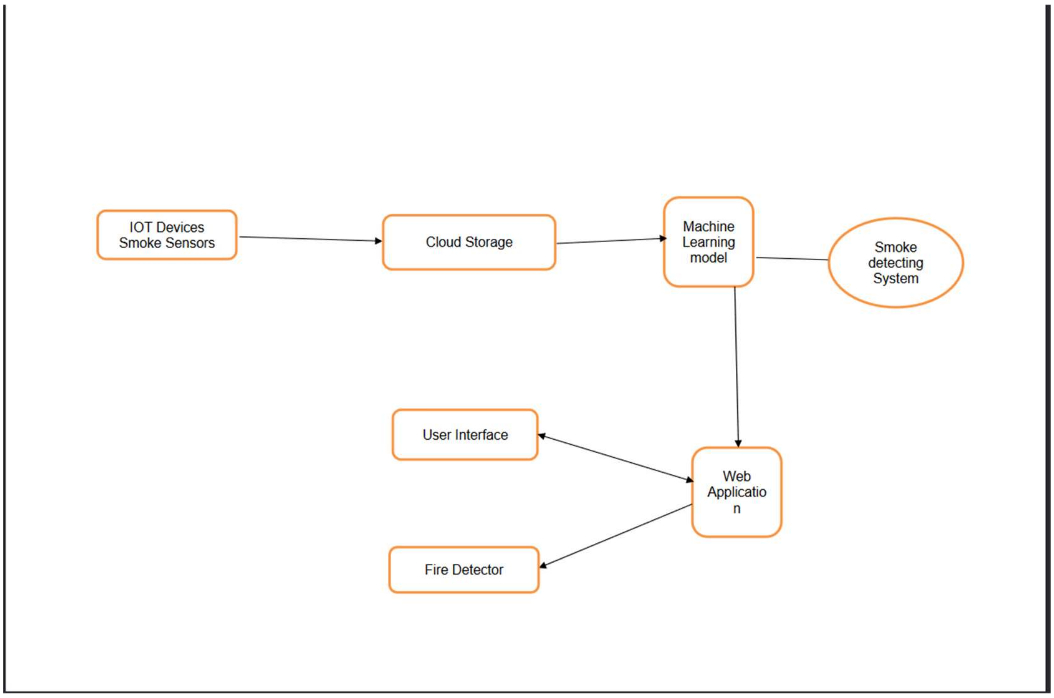
The machine learning model is trained to identify smoke particles in the sensor data. When the model detects smoke, it sends a signal to the web application.

Web Application

The web application is a user-friendly interface that allows users to configure, monitor, and manage the smoke detection system. When the web application receives a signal from the machine learning model, it triggers the fire alarm.

Fire Alarm

The fire alarm sounds to alert occupants of the building to the fire. This data flow diagram shows how IoT sensors, cloud computing, and machine learning can be used to create a smart smoke detection system that can detect smoke early and trigger a fire alarm quickly and reliably. This system has the potential to save lives by giving occupants more time to evacuate safely in the event of a fire.



User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
	Monitoring smoke detection	USN-6	Monitor the smoke detection system continuously	The system should continuously monitor data from IoT smoke sensors and provide information to user	High	Sprint-1
		USN-7	I get an immediate notification	When smoke is detected by any sensor, I should receive an immediate notification via email or SMS, specifying the location of the sensor triggering the alert.	High	Sprint-1
		USN-8	I get the notification with time stamp	The notification should include a timestamp indicating when the smoke was detected.	High	Sprint-1
		USN-9	I can have access to all events once I login to UI	I should have access to a dashboard or interface where I can view a history of smoke detection events, including timestamps and locations.	High	Sprint-1

The diagram of a smart smoke alarm system that uses IoT data to detect smoke early and trigger a fire alarm. The system consists of the following components:

- Smoke sensors: Smoke sensors are placed in strategic locations throughout the building and continuously monitor the air for smoke particles.
- IoT gateway: The IoT gateway is a device that collects data from the smoke sensors and sends it to the cloud platform.
- Cloud platform: The cloud platform stores and processes the sensor data. It also hosts the machine learning model.
- Machine learning model: The machine learning model is trained to identify smoke particles in the sensor data.
- Web application: The web application is a user-friendly interface that allows users to configure, monitor, and manage the smoke detection system.
- Fire alarm: The fire alarm sounds to alert occupants of the building to the fire.

Overall, smart smoke alarm systems that use IoT data can help to improve fire safety in buildings by detecting smoke earlier and providing more accurate information about the location of the fire. This can give occupants more time to evacuate safely and can help to reduce the damage caused by fires.

Solution Architecture :

Solution_Architecture.
pdf

Project Design Phase-I
Solution Architecture

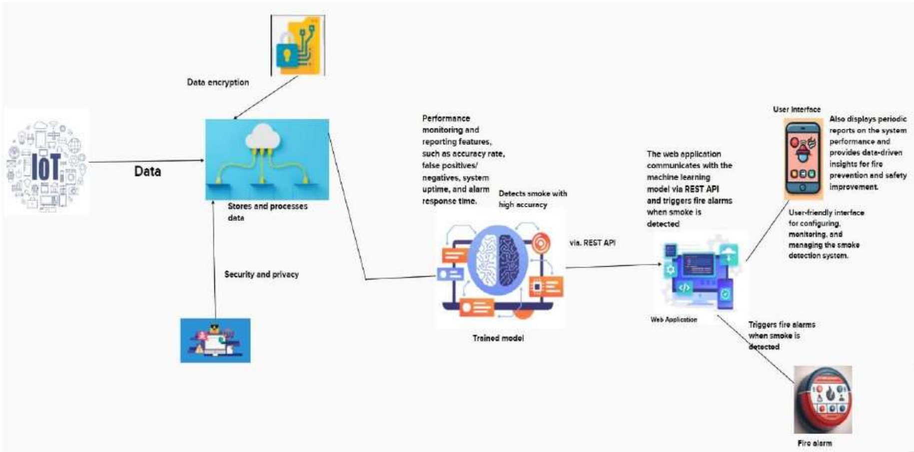
Date	**23 October 2023**
Team ID	**592862**
Project Name	**Detect Smoke with The Help of IOT Data nd Trigger A Fire Alarm**

Solution Architecture:

The smoke detection system consists of four layers: IoT layer, Data layer, Machine Learning layer, and Application layer.

- IoT layer: This layer consists of various smoke sensors that are installed in different locations and environments. The sensors collect smoke data and send it to the cloud via MQTT protocol.
- Data layer: This layer consists of a cloud platform that stores and processes the smoke data from the sensors. The cloud platform also provides data encryption, security, and privacy features to protect the data from unauthorized access or breaches.
- Machine Learning layer: This layer consists of a machine learning model that is trained on the smoke data and can detect smoke with high accuracy. The model is deployed on the cloud platform and can be accessed via REST API. The model also provides performance monitoring and reporting features, such as accuracy rate, false positives/negatives, system uptime, and alarm response time.
- Application layer: This layer consists of a web application that provides a user-friendly interface for configuring, monitoring, and managing the smoke detection system. The web application communicates with the machine learning model via REST API and triggers fire alarms when smoke is detected. The web application also displays periodic reports on the system performance and provides data-driven insights for fire prevention and safety improvement.

Solution Architecture Diagram:



The diagram of a smart smoke alarm system that uses IoT data to detect smoke early and trigger a fire alarm. The diagram is divided into four main sections:

- **Smoke sensors:** These sensors are placed in strategic locations throughout the building and continuously monitor the air for smoke particles.
- **Machine learning model:** This model is trained to identify smoke particles in the sensor data.
- **Web application:** This application is used to configure, monitor, and manage the smoke detection system. It also triggers the fire alarm when smoke is detected.
- **Fire alarm:** This alarm sounds to alert occupants of the building to the fire.

The diagram shows how the data flows from the smoke sensors to the machine learning model to the web application to the fire alarm.

The smoke sensors send data to the cloud platform, where it is processed by the machine learning model. The machine learning model then determines if there is a fire. If there is a fire, the machine learning model sends a signal to the web application, which triggers the fire alarm.

The web application also provides users with information about the smoke detection system, such as the status of the smoke sensors and the location of the fire.

This diagram shows how IoT data can be used to create a smart smoke alarm system that can detect smoke early and trigger a fire alarm quickly and reliably. This system has the potential to save lives by giving occupants more time to evacuate safely in the event of a fire.

Here are some of the benefits of using IoT data to detect smoke and trigger a fire alarm:

- **Earlier detection:** IoT sensors can detect smoke earlier than traditional smoke detectors. This gives occupants more time to evacuate safely.
- **More accurate location information:** IoT sensors can provide more accurate information about the location of the fire. This can help firefighters to respond more quickly and effectively.
- **Remote monitoring and management:** IoT sensors can be remotely monitored and managed, which can help to ensure that the smoke detection system is always in good working order.
- **Integration with other smart home devices:** IoT smoke detectors can be integrated with other smart home devices, such as lights and thermostats, to automatically respond to a fire alarm.

Overall, IoT-based smoke detection systems have the potential to revolutionize fire safety. By detecting smoke early and providing more accurate information about the location of the fire, these systems can help to save lives and reduce property damage.

6. PROJECT PLANNING & SCHEDULING:

Technical Architecture:

The technical architecture for the project "Detect Smoke With The Help Of IoT Data And Trigger A Fire Alarm" can involve various components and layers working together to ensure the seamless functioning of the system. Here's a high-level outline of the technical architecture:

1.Sensors Layer:

• Smoke sensors: Deploy high-quality and sensitive smoke sensors strategically throughout the building or designated area.

• Temperature sensors: Utilize temperature sensors to monitor the ambient temperature and detect any unusual heat patterns that could indicate a potential fire.

2.Data Collection and Transmission Layer:

• Microcontrollers: Use microcontrollers to collect data from the sensors and process it for further analysis.

• Communication modules: Employ reliable communication modules, such as Wi-Fi, Bluetooth, or Zigbee, to transmit the collected data to the central processing unit.

3.Data Processing and Analysis Layer:

• Central Processing Unit (CPU): Process the data using a central processing unit, employing algorithms and data analytics techniques to differentiate between normal environmental changes and the presence of smoke.

• Machine Learning Models: Implement machine learning models to continuously improve the system's ability to detect smoke accurately and reduce false alarms.

4.Decision and Action Layer:

• Fire Alarm System: Connect the system to a robust and responsive fire alarm system that triggers alarms and alerts occupants and relevant authorities in the event of smoke detection.

• User Interface: Develop a user-friendly interface for users to monitor the system, receive alerts, and manage system settings and configurations.

5.Security Layer:

• Data Encryption: Implement robust data encryption techniques to secure data transmission and storage, ensuring the privacy and integrity of sensitive information.

• Access Control: Employ strict access control measures to regulate user access to the system and prevent unauthorized manipulation or interference.

6.Integration and Scalability:

• API Integration: Create APIs for seamless integration with existing fire safety infrastructure and IoT devices, allowing for interoperability and easy deployment in different environments.

• Scalable Architecture: Design the system with scalability in mind, enabling the addition of more sensors and devices as the building or environment expands.

The technical architecture should prioritize reliability, security, scalability, and real-time responsiveness to ensure the efficient and effective functioning of the system in detecting smoke and triggering fire alarms.

Sprint Planning & Estimation and Sprint Delivery Schedule:

Sprint 1

Task: Research and select appropriate smoke sensors and IoT devices.

Task: Set up the basic IoT framework and establish communication between sensors and the central processing unit.

Task: Develop a prototype for data collection and transmission from the sensors to the central processing unit.

Sprint 2

Task: Implement data processing algorithms for real-time smoke detection and analysis.

Task: Integrate the fire alarm system with the smoke detection mechanism.

Task: Design a simple user interface for monitoring and managing the system.

Sprint 3

Task: Conduct rigorous testing and debugging of the integrated system.

Task: Develop a remote monitoring and alert system for users.

Task: Ensure the security of data transmission and storage.

Sprint 4

Task: Finalize the user interface design and usability testing.

Task: Conduct scalability and compatibility testing with various environments.

Task: Prepare comprehensive documentation for the system.

7.Coding and solutioning

```
pip install flask-ngrok
```

```
Requirement already satisfied: flask-ngrok in /usr/local/lib/python3.10/dist-packages (0.0.25)
Requirement already satisfied: Flask>=0.8 in /usr/local/lib/python3.10/dist-packages (from flask-ngrok) (2.2.5)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from flask-ngrok) (2.31.0)
Requirement already satisfied: Werkzeug>=2.2.2 in /usr/local/lib/python3.10/dist-packages (from Flask>=0.8->flask-ngrok) (3
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.10/dist-packages (from Flask>=0.8->flask-ngrok) (3.1.2
Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.10/dist-packages (from Flask>=0.8->flask-ngrok)
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.10/dist-packages (from Flask>=0.8->flask-ngrok) (8.1.7)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->flask-ng
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->flask-ngrok) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->flask-ngrok) (
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->flask-ngrok) (
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=3.0->Flask>=0.8->fl
```

```
[ ] #!pip install pyngrok

!curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee /etc/apt/trusted.gpg.d/n

deb https://ngrok-agent.s3.amazonaws.com buster main
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:6 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  InRelease
Hit:7 https://ngrok-agent.s3.amazonaws.com buster InRelease
Hit:8 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy InRelease
Hit:9 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Hit:10 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy InRelease
Hit:11 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
19 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ngrok is already the newest version (3.4.0).
0 upgraded, 0 newly installed, 0 to remove and 19 not upgraded.
```

```
[ ] #!ls

SmokeDetection_AI_.ipynb  smoke_detection_iot.csv  smoke.pkl  template

[ ] #!ngrok config add-authtoken 2XwiNZpdpfhh1T3kInInwRQVaj5_6WYQLW6miW8VN3duGYNE7

Authtoken saved to configuration file: /root/.ngrok2/ngrok.yml

[ ] '''from flask import Flask
    from pyngrok import ngrok'''

'from flask import Flask\nfrom pyngrok import ngrok'

[ ] from google.colab import drive
    drive.mount('/drive')

Drive already mounted at /drive; to attempt to forcibly remount, call drive.mount("/drive", force_remount=True).

[ ] cd /drive/MyDrive/SmokeDetection_fire_alarm

/drive/MyDrive/SmokeDetection_fire_alarm

[ ] from google.colab.output import eval_js
    print(eval_js("google.colab.kernel.proxyPort(5000)"))

https://mzsqult2cco-496ff2e9c6d22116-5000-colab.googleusercontent.com/
```

```python
from flask_ngrok import run_with_ngrok
from flask import Flask, request, render_template
import pickle
import numpy as np

# Load the model
with open('smoke.pkl', 'rb') as file:
    model = pickle.load(file)

app = Flask(__name__,template_folder='/drive/MyDrive/SmokeDetection_fire_alarm/template')
run_with_ngrok(app)  # Start ngrok when app is run

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

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

@app.route('/submit', methods=['POST'])
def submit():
    temperature = float(request.form['Temperature[C]'])
    humidity = float(request.form['Humidity[%]'])
    tvoc = float(request.form['TVOC[ppb]'])
    raw_h2 = float(request.form['Raw H2'])
    raw_ethanol = float(request.form['Raw Ethanol'])
    pressure = float(request.form['Pressure[hPa]'])
    nc0_5 = float(request.form['NC0.5'])
    cnt = float(request.form['CNT'])
```

```python
    final_features = np.array([[temperature, humidity, tvoc, raw_h2, raw_ethanol, pressure, nc0_5, cnt]])

    # Make the prediction
    prediction = model.predict(final_features)[0]

    # Set the prediction text based on the model prediction
    if prediction == 0:
        prediction_text = 'The input does not indicate smoke detection.'
    else:
        prediction_text = 'The input indicates smoke detection.'

    # Render the result template with the prediction text
    return render_template('submit.html', prediction_text=prediction_text)

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

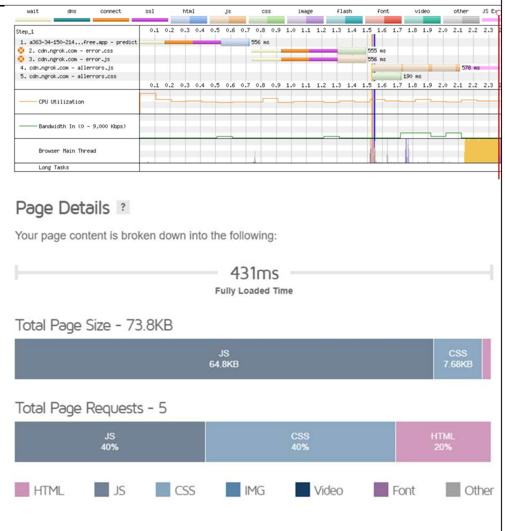
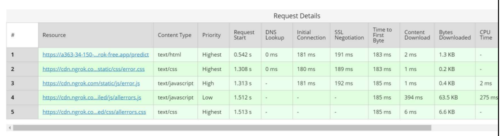
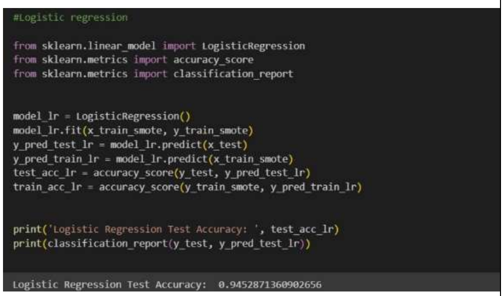
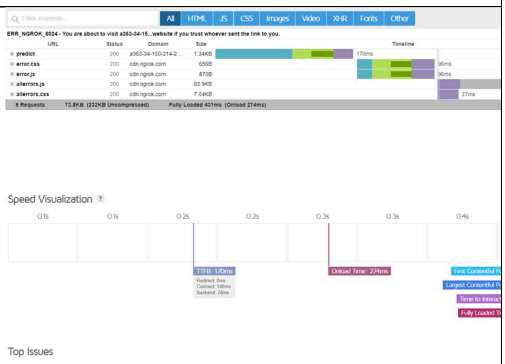
```python
#Gradient boosting
from sklearn.ensemble import GradientBoostingClassifier
model_gb = GradientBoostingClassifier()
model_gb.fit(x_train_smote, y_train_smote)
y_pred_test_gb = model_gb.predict(x_test)
y_pred_train_gb = model_gb.predict(x_train_smote)
test_acc_gb = accuracy_score(y_test, y_pred_test_gb)
train_acc_gb = accuracy_score(y_train_smote, y_pred_train_gb)

print('Gradient Boosting Test Accuracy: ', test_acc_gb)
print(classification_report(y_test, y_pred_test_gb))
```

```
Gradient Boosting Test Accuracy:  0.9999467773697376
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5423
           1       1.00      1.00      1.00     13366

    accuracy                           1.00     18789
   macro avg       1.00      1.00      1.00     18789
weighted avg       1.00      1.00      1.00     18789
```

```python
import pandas as pd
from sklearn.metrics import accuracy_score, classification_report

# Define the list of model names
model_names = ['Logistic Regression', 'SVM', 'Gradient Boosting', 'KNN']

# Define the list of predicted test labels for each model
y_pred_tests = [y_pred_test_lr,y_pred_test_svm, y_pred_test_gb, y_pred_test_knn]

# Create an empty dataframe to store the comparison results
results_df = pd.DataFrame(columns=['Model', 'Test Accuracy', 'Precision', 'Recall', 'F1-score'])

# Loop through each model and calculate the evaluation metrics
for i, model_name in enumerate(model_names):
    model = model_names[i]
    y_pred_test = y_pred_tests[i]

    test_acc = accuracy_score(y_test, y_pred_test)
    classification = classification_report(y_test, y_pred_test, output_dict=True)
    precision = classification['macro avg']['precision']
    recall = classification['macro avg']['recall']
    f1_score = classification['macro avg']['f1-score']

    results_df = results_df.append({'Model': model_name,
                                    'Test Accuracy': test_acc,
                                    'Precision': precision,
                                    'Recall': recall,
                                    'F1-score': f1_score}, ignore_index=True)



#Display the results in table
print(results_df)
```

8. Model Performance Testing:

The project team shall fill in the following information in the model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	**Code Quality: good (no issues)** **Screen record time – 4.56** **Speed Valuation – 431 ms**	
2.	Accuracy	Training Accuracy – 0.945287 Validation Accuracy – 0.8154	
3.	Confidence Score	Class Detected - 4 Confidence Score - 80	

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td><pre>
#Logistic regression

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

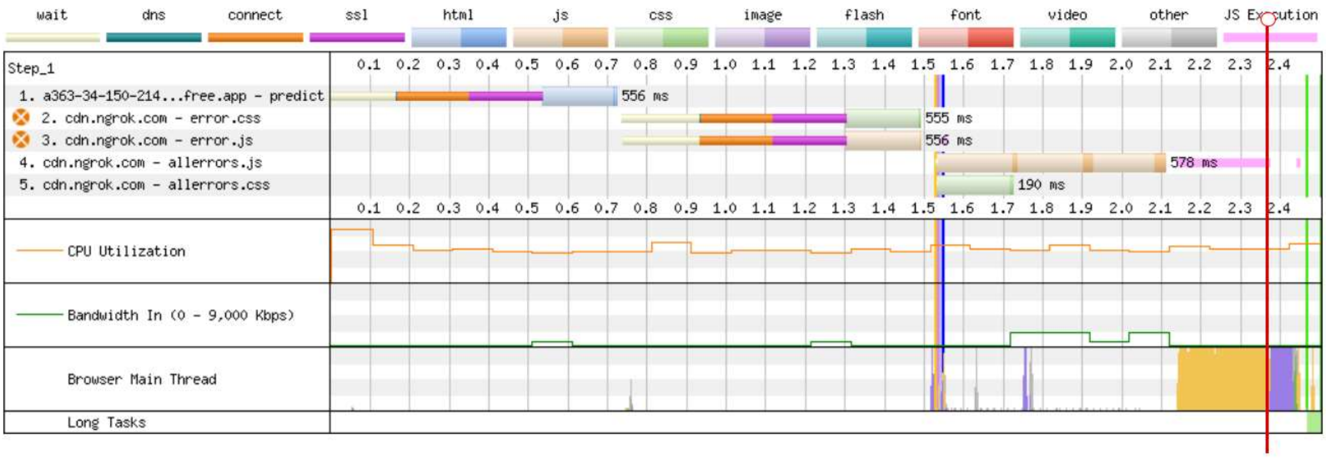
model_lr = LogisticRegression()
model_lr.fit(x_train_smote, y_train_smote)
y_pred_test_lr = model_lr.predict(x_test)
y_pred_train_lr = model_lr.predict(x_train_smote)
test_acc_lr = accuracy_score(y_test, y_pred_test_lr)
train_acc_lr = accuracy_score(y_train_smote, y_pred_train_lr)

print('Logistic Regression Test Accuracy: ', test_acc_lr)
print(classification_report(y_test, y_pred_test_lr))

Logistic Regression Test Accuracy: 0.9452871360902656
              precision   recall f1-score   support

           0   0.85   0.98   0.91   5423
           1   0.99   0.93   0.96   13366

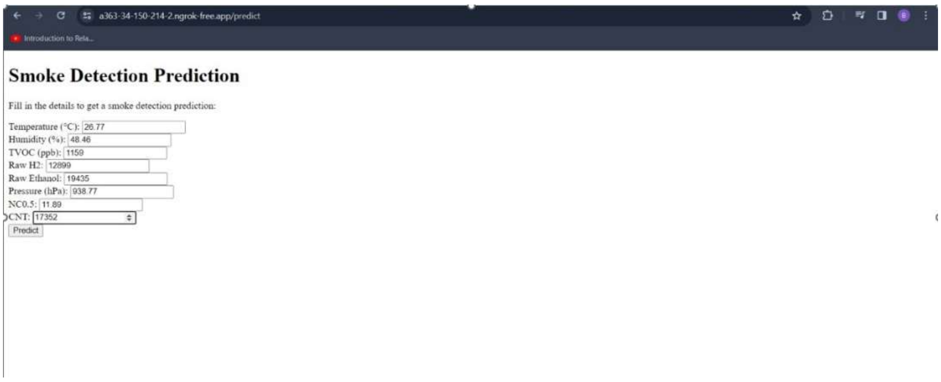
    accuracy   0.95   18789
   macro avg   0.92   0.96   0.94   18789
weighted avg   0.95   0.95   0.95   18789
</pre></td>
</tr>
</table>



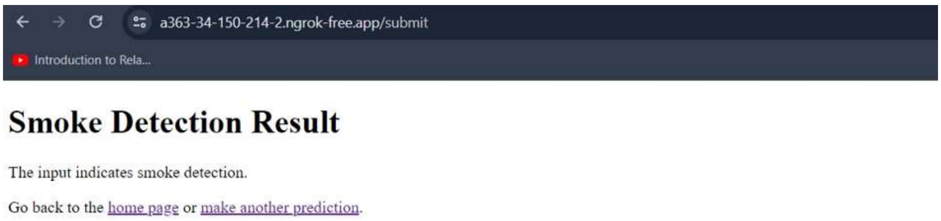
9. RESULTS:

Home page:

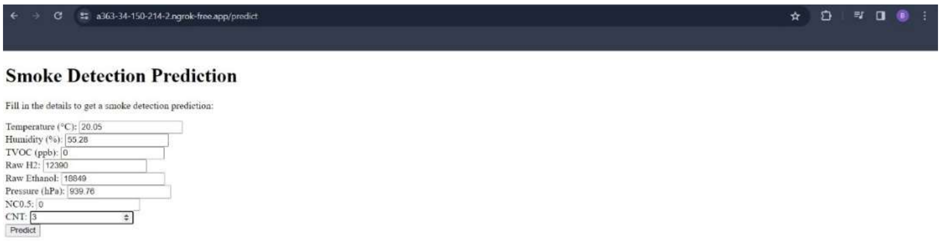
Predict page: 1



Submit Page 1



Predict page 2 :



Submit page:2

10.ADVANTAGES & DISADVANTAGES :

Here are some advantages and disadvantages of using IoT data to detect smoke and trigger a fire alarm:

Advantages:

- Earlier detection: IoT sensors can detect smoke earlier than traditional smoke detectors, giving occupants more time to evacuate safely.
- More accurate location information: IoT sensors can provide more accurate information about the location of the fire, helping firefighters to respond more quickly and effectively.

- Remote monitoring and management: IoT sensors can be remotely monitored and managed, helping to ensure that the smoke detection system is always in good working order.
- Integration with other smart home devices: IoT smoke detectors can be integrated with other smart home devices, such as lights and thermostats, to automatically respond to a fire alarm.
- Reduced costs: IoT-based smoke detection systems can be more cost-effective to install and maintain than traditional smoke detectors, especially in large buildings.

Disadvantages:

- False alarms: IoT sensors can sometimes trigger false alarms, which can be disruptive and can lead to complacency.
- Cybersecurity risks: IoT sensors can be vulnerable to hacking, which could allow attackers to disable the system or trigger false alarms.
- Privacy concerns: IoT sensors collect data about the environment, including data about people's movements. This data could be collected and used without people's consent.
- Reliability: IoT-based smoke detection systems are complex systems, and there is a risk that they could fail. This could lead to a delay in detecting a fire or a failure to trigger the fire alarm.

11.CONCLUSION :

IoT-based smoke detection systems can improve fire safety by detecting smoke earlier and more accurately than traditional smoke detectors. This can give occupants more time to evacuate safely and can help to reduce property damage. However, there are some potential risks associated with these systems, such as false alarms and cybersecurity risks. These risks can be mitigated by taking steps such as using high-quality sensors, implementing a redundancy system, using encryption, and educating users. Overall, the benefits of using IoT data to detect smoke and trigger a fire alarm outweigh the risks.

12.FUTURE SCOPE :

The future scope of the project "Detect Smoke With The Help Of IOT Data And Trigger A Fire Alarm" is very promising. As IoT technology continues to develop and become more affordable, it is likely that these systems will become increasingly common in buildings of all types.

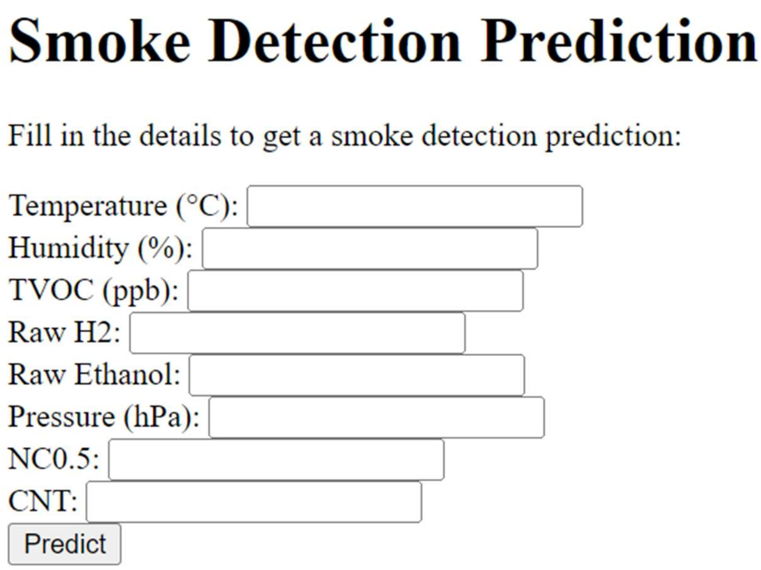
Here are some specific areas where IoT-based smoke detection systems are likely to evolve in the future:

- Reduced costs: As the cost of IoT sensors and cloud computing continues to decline, IoT-based smoke detection systems will become more affordable to install and maintain. This will make them more accessible to a wider range of businesses and homeowners.
- Integration with other smart home devices: IoT-based smoke detection systems are already being integrated with other smart home devices, such as lights and thermostats. In the future, this integration is likely to become even more sophisticated and seamless. This will allow IoT-based smoke detection systems to play a larger role in overall home automation and security.

  In addition to these general trends, there are a number of specific areas where IoT-based smoke detection systems are likely to be developed in the future. For example:
- AI-powered smoke detection: Artificial intelligence (AI) is already being used to develop new and innovative ways to detect smoke. For example, AI-powered smoke detection systems can be used to analyze video footage from security cameras to identify smoke particles. In the future, AI is likely to play an even greater role in IoT-based smoke detection systems.
- Wearable smoke detectors: Wearable smoke detectors are a new type of smoke detector that can be worn on the body. These devices are still in their early stages of development, but they have the potential to revolutionize the way that smoke is detected. For example, wearable smoke detectors could be used to protect firefighters and other first responders from smoke inhalation.

Overall, the future scope of IoT-based smoke detection systems is very promising. As IoT technology continues to develop, these systems are likely to become more accurate, reliable, affordable, and integrated with other smart home devices. This will make them more accessible and effective at protecting people and property from fire.

13.APPENDIX:



GitHub Link: