Internet of Things

Vaibhav Gupta	IEC2019023
Pravallika Kodi	IIT2019234
Sangam Barnwal	IIT2019094
Subhash Balla	IIT2019207

TOPIC

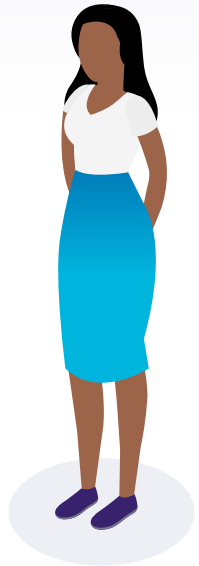
FIRE ALARMING SYSTEM

PROBLEM STATEMENT

A **fire alarm system** warns people when smoke, fire, carbon monoxide or other fire-related emergencies are detected. Whenever we detect fire, the admin and accessible people can see the dashboard and get notified as the indication of fire. Additionally we can also mail fire departments and police station about the same. The newly registered people who can’t access the dashboard will send the request. Whenever the request is granted by the admin, he/she can now view the dashboard.

PROBLEMS WE DEAL WITH

- How are we going to detect fire?
- How do we monitor them?
- How to connect nodeMCU with firebase server and then sending and fetching data from it
- How do we send alert messages when we detect fire
- How IoT helps us in building this system



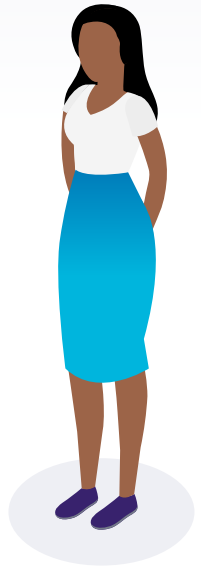
WHY HAVE WE CHOSEN THIS PROJECT

In general, we find many fire accidents due to gas leakages or due to faulty wiring or humans carelessness. As the fire spreads rapidly and causes a lot of damage, there will be a huge population loss at that area. Most of the fire accidents are due to no prior knowledge or indications. So we in this project try to build a model which can detect fire and let the subscriber know(notified) in case of fire detection or gas leakage.Due to this prior indication to the people, we can decrease these types of fire accidents and save people lives.

Application/ Uses Cases

We will targeting the fire prone areas where are more vulnerable to catch fire such as

- Chemical Factory (Fireworks and Flammables)
- Textile Factory
- Furniture Factories
- Electrical Manufacturing
- Homes and Others



HARDWARE REQUIREMENTS

- Infrared sensor PT333b - detects ignition of light - **X2**
- MQ2 - Gas sensor - **X2**
- NodeMCU X 2 - **X2**
- LED

SOFTWARE REQUIREMENTS

- Arduino IDE
- Firebase as Web Server
- Mobile application and website for viewing data
- Heroku to host our web page
- Django framework for building website
- Google spreadsheets to store data

TECHNOLOGY / LANGUAGE REQUIREMENT

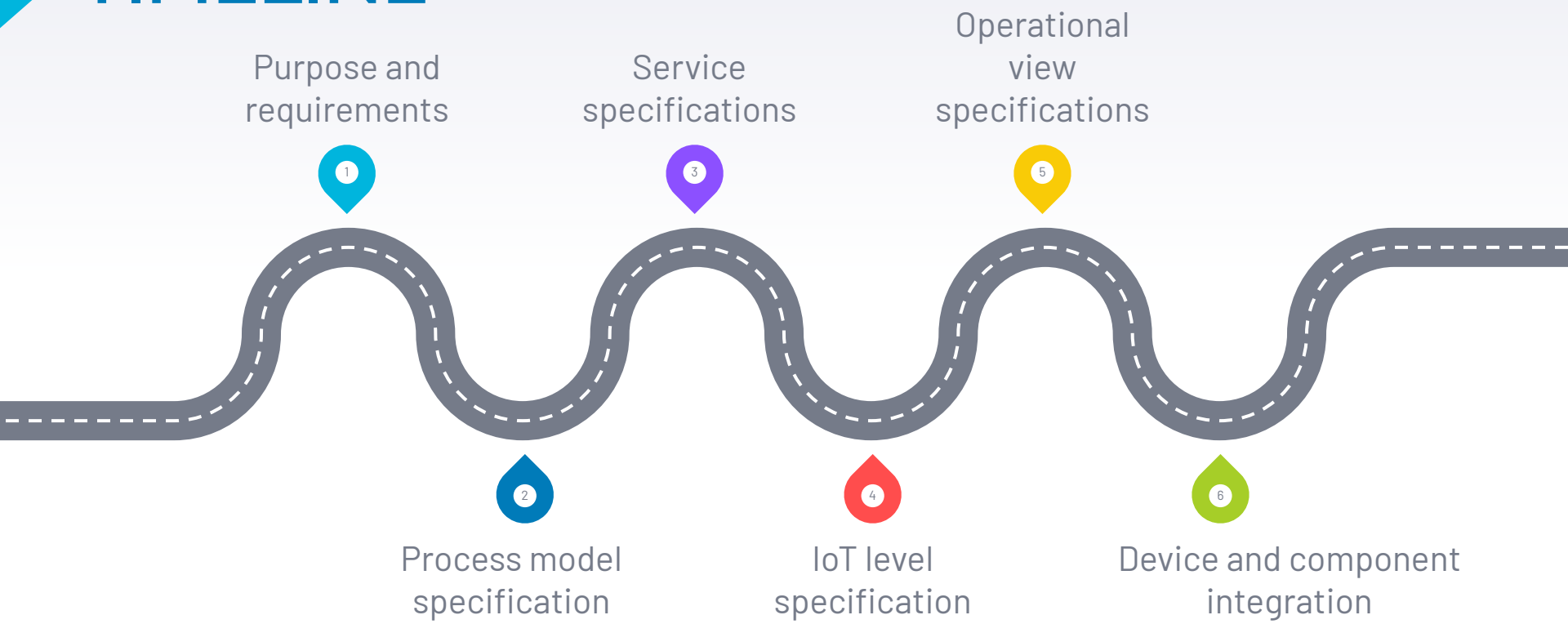
- Code in C language(Arduino IDE) - for sending data of sensors to firebase
- For website, we used Django framework which uses python
- Connecting firebase and website with spreadsheets
- Interfacing with WIFI module
- SMTP for mailing

DELIVERABLES

- **Website** - login, signup, Dashboard, request page(request sent to admin), sending emails,history, pie charts representing sensor values,line chart of data values, grant access, remove access of users

- Push notification on Email and mobile

TIMELINE


Purpose and requirements
1
Process model specification
2
Service specifications
3
IoT level specification
4
Operational view specifications
5
Device and component integration
6

TIMELINE

- **C1 -** Gone through research papers and articles related to the project. Designed the project implementation
- **C2 -** Integrated all the components and connected nodemcu with firebase via spreadsheets, and fetched the same from firebase on timely basis. Added history and line graph feature
- **C3 -** Added 2 nodes to the existing circuit, updated the website accordingly, added features, detected fire effectively

Purpose and requirements

- purpose- safeguard fire in infrastructure
- behaviours-auto mode
- system manager- monitor + control
- Data analysis - locally like comparison
- deploy-local deploy stationary remote access
- security-authentication
- Process model specification

CONNECTING TO FIREBASE VIA GOOGLE SPREADSHEETS

- We have built our firebase server to store and read the sensors data
- We connected nodeMCU to firebase via google spreadsheets
- We have been updating the sensor values on the spreadsheets from nodeMCU
- We send the spreadsheets data to firebase as an object, and fetch this data and show on the website on timely basis

FIREBASE REALTIME DATABASE

SMTP PROTOCOL FOR EMAIL NOTIFICATION

- We actually send the email notifications to the clients and nearby police stations, hospitals and fire departments if needed
- For sending the emails, we used SMTP (Simple Mail Transfer Protocol)
- These emails are sent if the sensor data is not in threshold values
- In the email, we warn the people and additionally we shown the sensor values for detailed information

EMAIL SCREENSHOTS

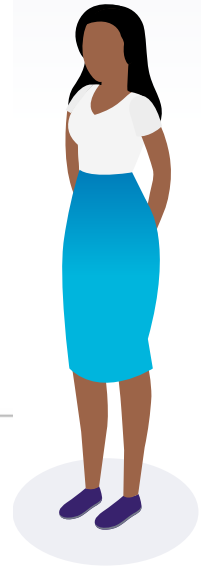
FIRE ALERT 🔥🔥🔥 Inbox ×

P **iit2019234@iiita.ac.in** <iit2019234@iiita.ac.in> Fri, M

to me

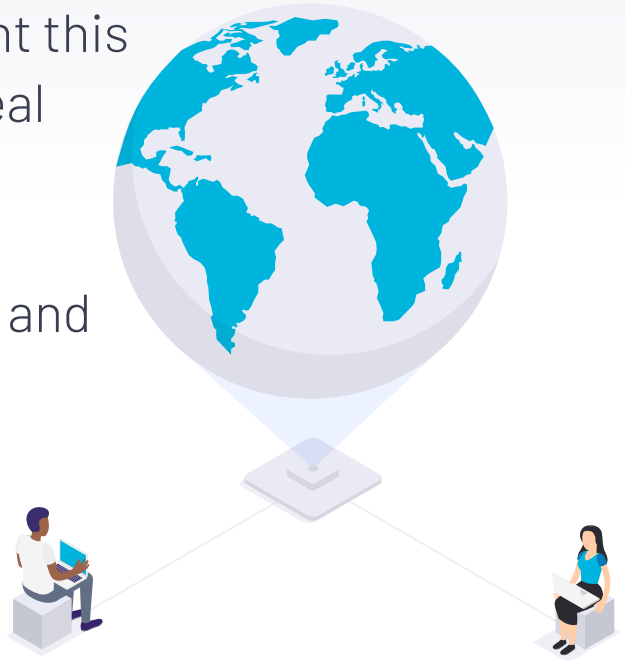
Warning: This is fire alert. Fire is detected!!!
Be cautious and take necessary actions...
flame_value1: 0
smoke_value1: 279
flame_value2: 0
smoke_value2: 283
flame_value3: 0
smoke_value3: 250

Avast This email has been checked for viruses by Avast antivirus software.
www.avast.com



BUILDING WEBSITE

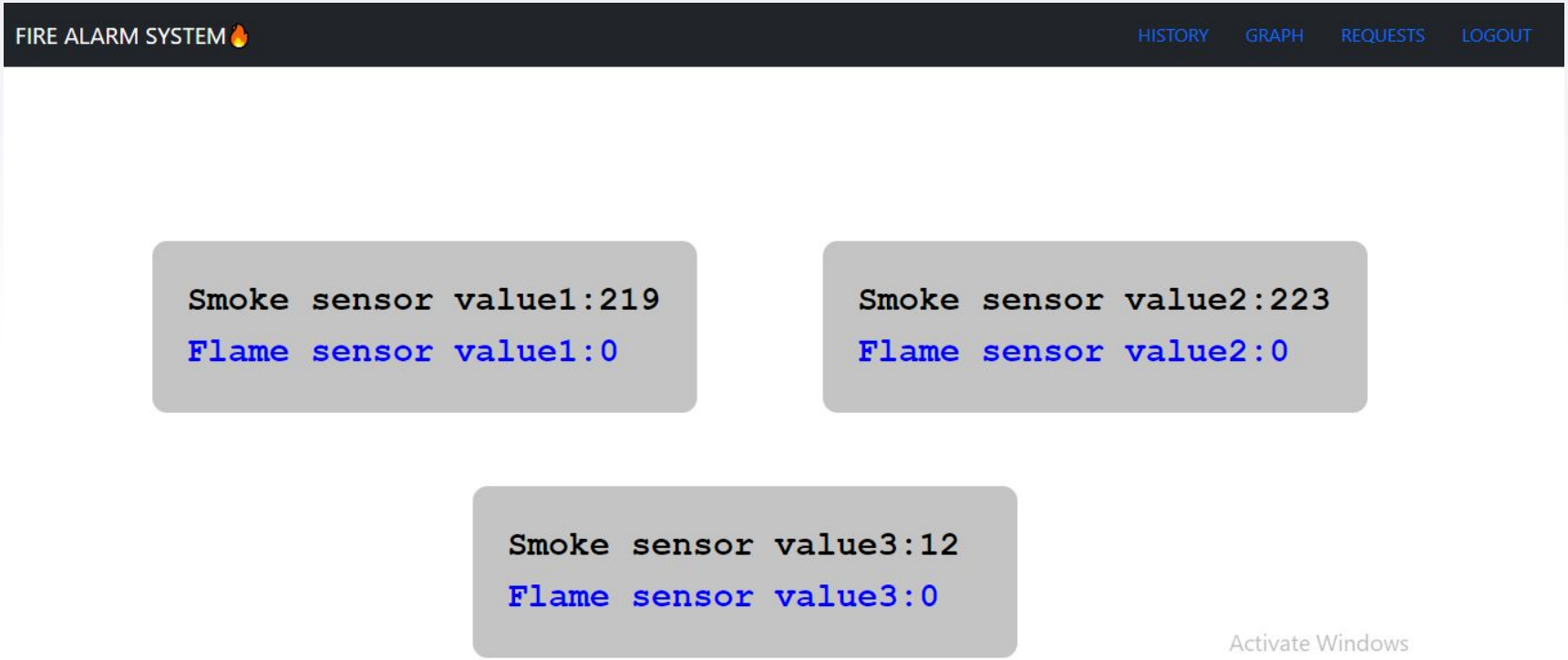
- We built a simple website which is integrated with firebase server
- We have used django framework to implement this
- As we retrieve(fetch) data from firebase on real time, we try to show this on our dashboard continuous basis
- Our dashboard changes the data dynamically and lets the client know the information
- Also shows pop-up messages for warning purposes



BUILDING WEBSITE

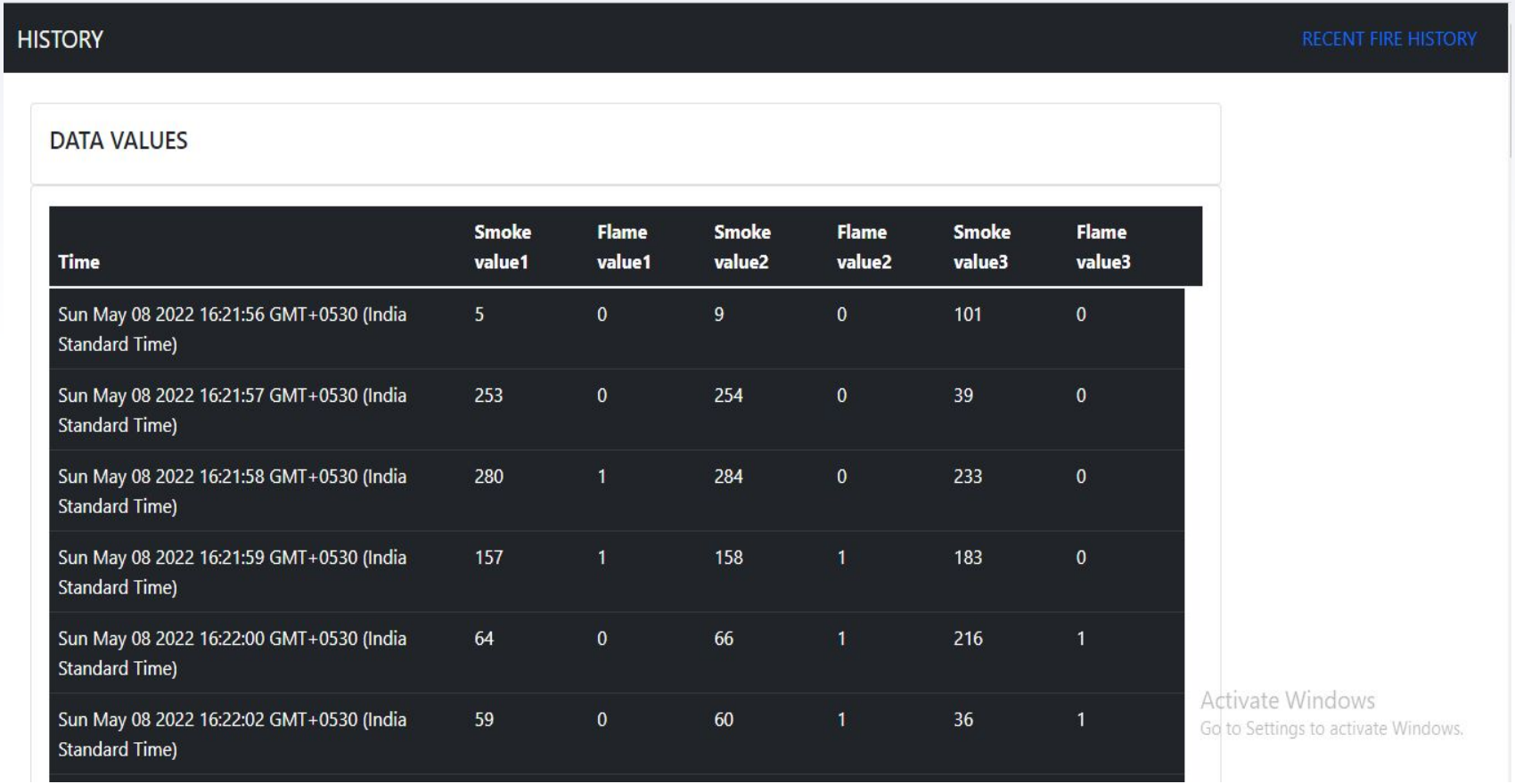
- We have added history feature, which makes user know about the previous data values of the sensors
- We fetched all the data in the firebase server and sent it to the html page and displayed it
- We hosted this website on heroku

WEBSITE SCREENSHOT

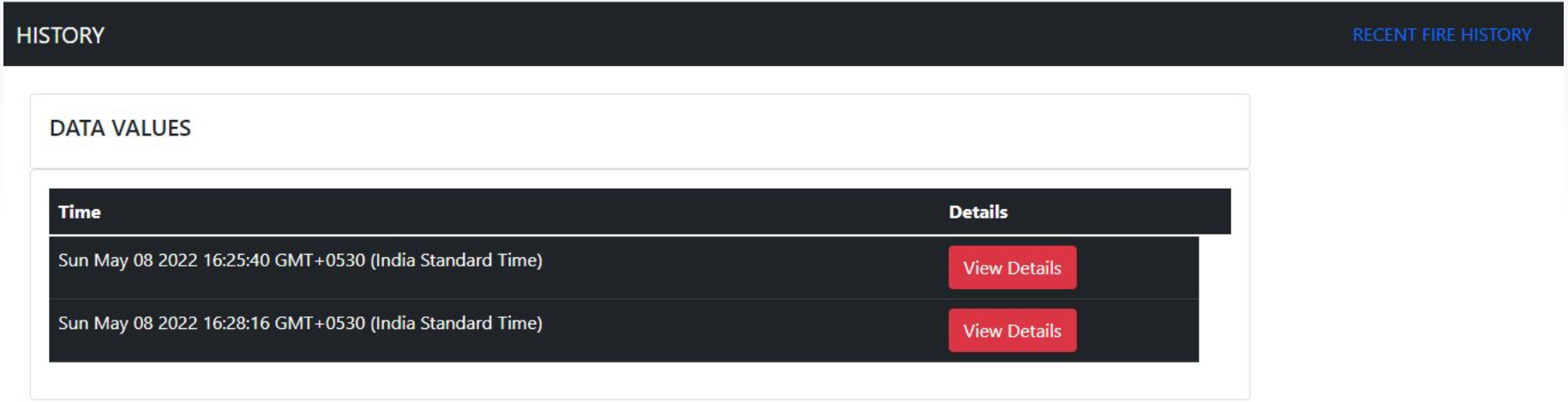


Activate Windows
Go to Settings to activate Windows.

HISTORY SCREENSHOT

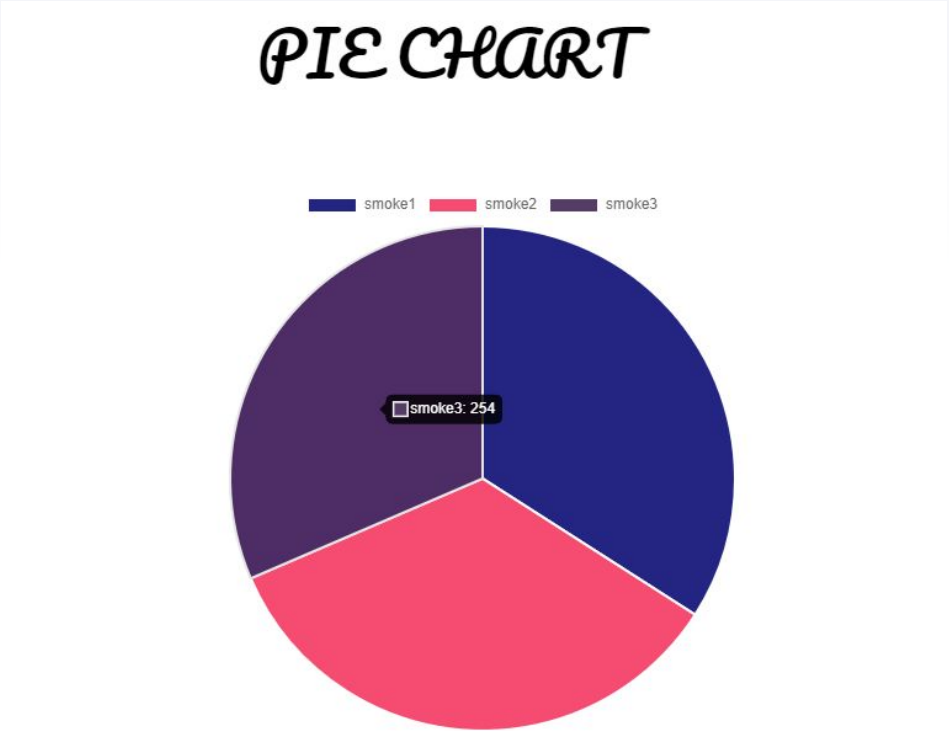


RECENT FIRE HISTORY



It shows the history about the information of data when fire is detected

DETAILED INFORMATIONS ABOUT FIRE VALUES

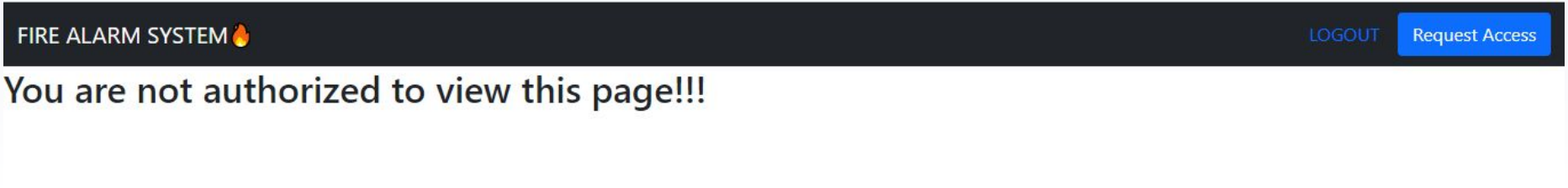


This pie chart gives us information about sensor values when fire is detected

LINE CHARTS TO REPRESENT VALUES

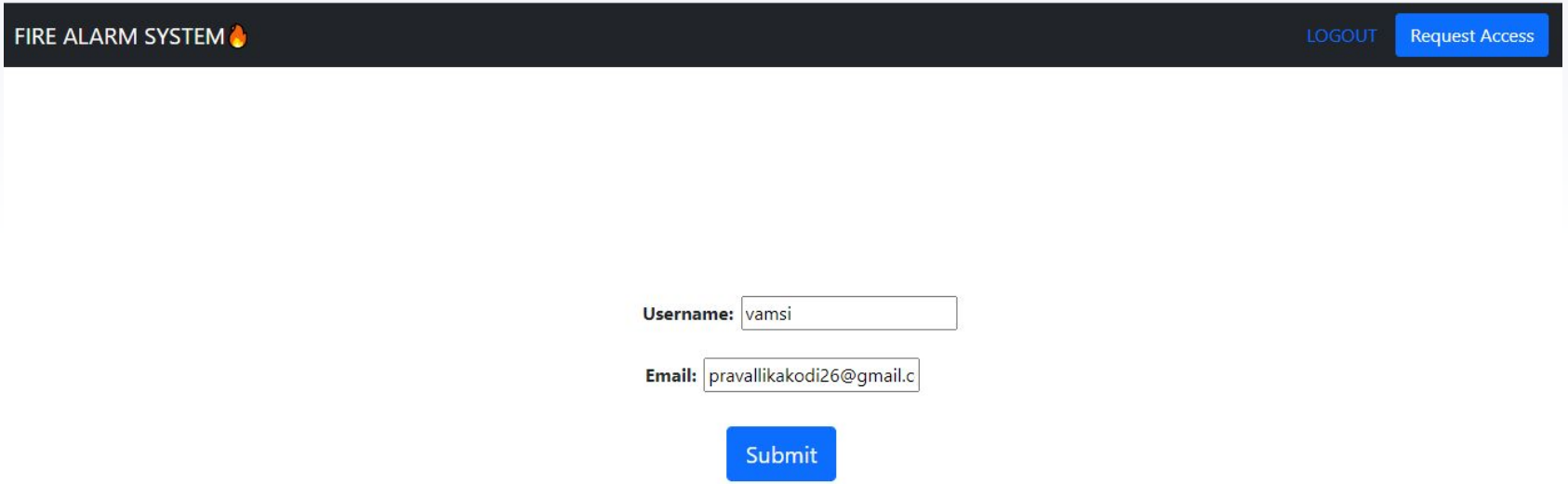
Line chart gives the data points at each time interval. Three charts refers to 3 nodes.

DASHBOARD IF WE ARE NOT GIVEN ACCESS



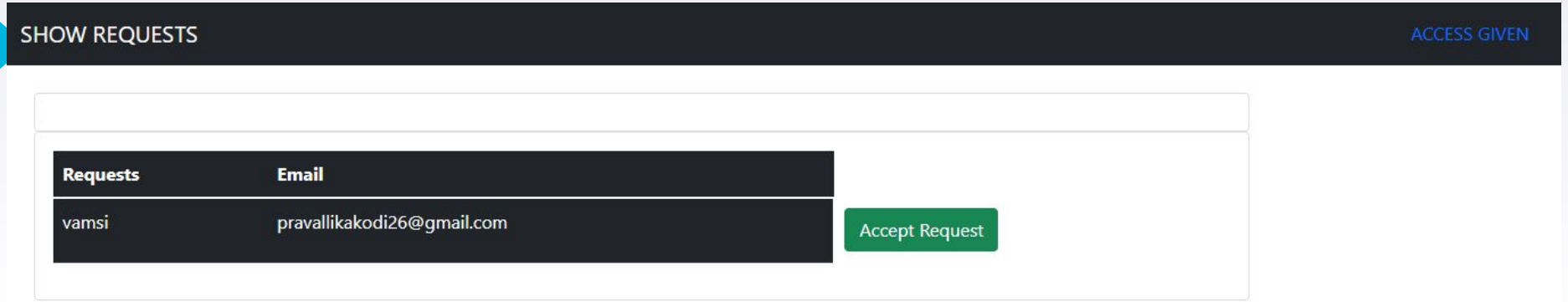
If a person newly enters the website, he won't be able to view the data until admin gives him the access

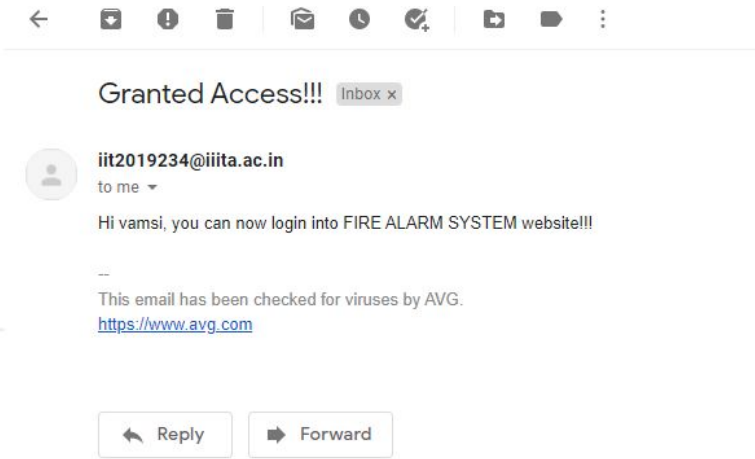
REQUESTING ACCESS



Form will be autofilled, the user just need to press submit button, to request the access to admin

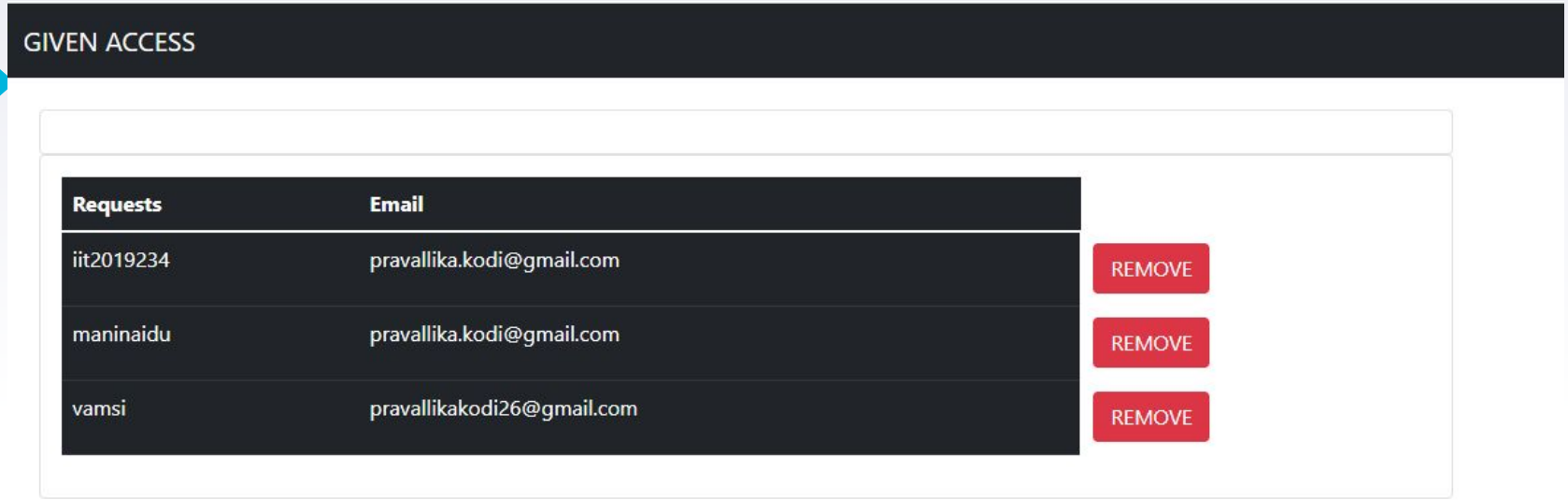
REQUESTS RECEIVED





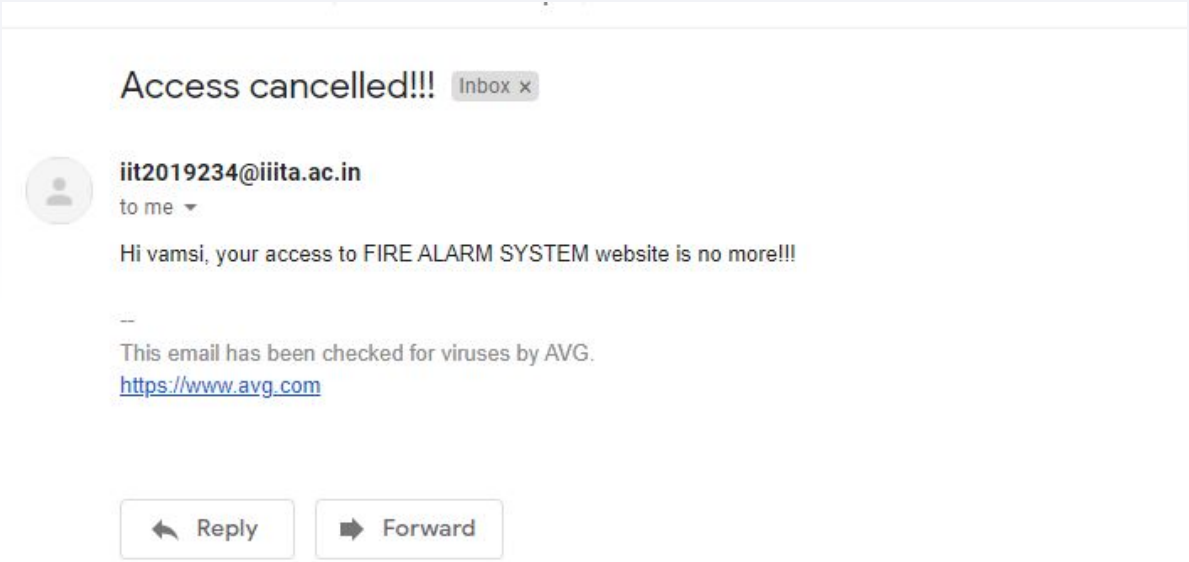
Show request page shows the requests of the users, when the admin accepts them, user will be notified and now he can access the website and see the details

GIVEN ACCESS



Given access page shows the users who are given access to the website to see the data values

REMOVE ACCESS



When the user's access is cancelled, he will receive this mail and no longer can access the website

CONFIGURING AND CONNECTING TO FIREBASE

```
config = {
    'apiKey': "AIzaSyBV3jP5cl1AVP3HgO6_1J_wrjyffqQUfko",
    'authDomain': "iotproject-2800f.firebaseapp.com",
    'databaseURL': "https://iotproject-2800f-default-rtdb.firebaseio.com",
    'projectId': "iotproject-2800f",
    'storageBucket': "iotproject-2800f.appspot.com",
    'messagingSenderId': "863247827424",
    'appId': "1:863247827424:web:ba640e5eabf0b412062b85",
    'measurementId': "G-727CF3R175"
  }
firebase=pyrebase.initialize_app(config)
# Create your views here.

db=firebase.database()
data=db.child("15-itebVVyWrgPlr5qjMDYk3LNG55fhygj66YDEG9pdw/Sheet1").get().val()
#print(data)
```

HISTORY

```
def history(request):
    db=firebase.database()
    data=db.child("15-itebVVyWrgPlr5qjMDYk3LNG55fhygj66YDEG9pdw/Sheet1").get().val()
    return render(request,'history.html',{'sensors_data':data})
```

In the above code, we are connecting to firebase and fetching the data

Statistics

```
def graph(request):
    db=firebase.database()
    data=db.child("15-itebVVyWrgPlr5qjMDYk3LNG55fhygj66YDEG9pdw/Sheet1").get().val()
    flame_val=[]
    smoke_val=[]
    time=[]
    for key,val in data.items():
            #print(data[val]['flame_sensor'])
            print("key: ",key)
            print("val: ",val)
            flame_val.append(val['Flame'])
            smoke_val.append(val['Smoke'])
            #print(type(data[val]['Date']))
            date_split=key.split(" ")
            date_part=date_split[4]
            time_split=date_part.split(":")
            tym=''.join(time_split)
            print("modified_time: ",time)
            time.append(int(tym))
    print(flame_val)
    print(smoke_val)
    return render(request,'graph.html',{'flame_val': flame_val,'smoke_val':smoke_val,'time':time})
    #return render(request,'graph.html',{'flame_val': flame_val,'smoke_val':smoke_val})
```

For the purpose of plotting the graph, we are fetching the data from the firebase. We are sending this data in the form of arrays into html templates.

GRAPH CONTINUATION

```
async function data_to_graph(){

console.log({{time}})

var ctx = document.getElementById("myChart").getContext("2d");
var myChart = new Chart(ctx, {
type: "line",
data: {
    labels: {{time}},
    //labels: [1,2,3,4,5,6,7,8,9,10,11,12,13],
    datasets: [
    {
        label: "Flame sensor",
        data: {{flame_val}},
        backgroundColor: "rgba(75, 192, 192,0.6)",
    },
    {
        label: "Smoke sensor",
        data: {{smoke_val}},
        backgroundColor: "rgba(255,20,147,0.6)",
    },
    ],
},
});
```

Fetching Data (on real time basis)

```
var search_time=new Date();
console.log(time)
var str2="15-itebVVyWrgPlr5qjMDYk3LNG55fhygj66YDEG9pdw/Sheet1/"+search_time;
firebase.database().ref(str2).on("value", (sanpshot) => {
    console.log("neede--",sanpshot.val());
    if(sanpshot.val()!=undefined){
        console.log(sanpshot.val().Smoke,sanpshot.val().Flame)
        sendToDashboard(sanpshot.val().Smoke,sanpshot.val().Flame);
        if(sanpshot.val().Smoke>600 && sanpshot.val().Flame==0){

            sendEmail(sanpshot.val().Smoke,sanpshot.val().Flame);
            sendalert();
            // alert("Warning: FIRE DETECTED🔥🔥🔥")
        }
        else{
            remove_warning();
        }
```

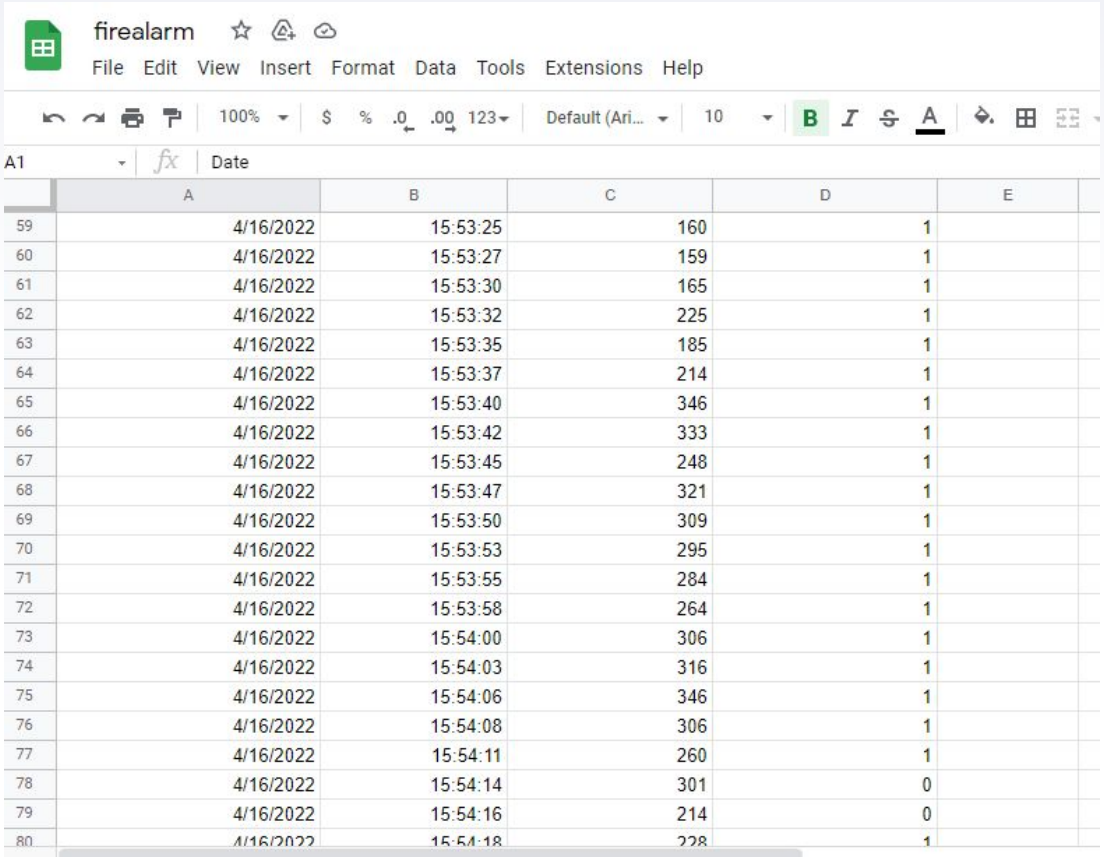
We are extracting the data from the firebase w.r.t current time and sending it to dashboard.
We will check the threshold values.if they cross the cutoff,we send alerts to user via mail.

MAIL NOTIFICATIONS

```
function sendEmail(gas_value,flame_value){
    Email.send({
        Host : "smtp.gmail.com",
        Username : "iit2019234@iiita.ac.in",
        Password : "Pravallika@26",
        To : ["{{email}}"],
        From : "iit2019234@iiita.ac.in",
        Subject : "FIRE ALERT🔥🔥🔥",
        Body : "Warning: This is fire alert. Fire is detected!!!<br/> Be cautious and take necessary actions...<br/> "+
        "flame_value: "+flame_value+"<br/>gas_value: "+gas_value
    }).then(
        //message => alert(message)
        console.log("Mail sent successfully!!!")
    );
}
```

We send mail to user via SMTP protocol.Additionally, mail contains sensor values of smoke and flame.

GOOGLE SPREADSHEETS



	A	B	C	D	E
59	4/16/2022	15:53:25	160	1	
60	4/16/2022	15:53:27	159	1	
61	4/16/2022	15:53:30	165	1	
62	4/16/2022	15:53:32	225	1	
63	4/16/2022	15:53:35	185	1	
64	4/16/2022	15:53:37	214	1	
65	4/16/2022	15:53:40	346	1	
66	4/16/2022	15:53:42	333	1	
67	4/16/2022	15:53:45	248	1	
68	4/16/2022	15:53:47	321	1	
69	4/16/2022	15:53:50	309	1	
70	4/16/2022	15:53:53	295	1	
71	4/16/2022	15:53:55	284	1	
72	4/16/2022	15:53:58	264	1	
73	4/16/2022	15:54:00	306	1	
74	4/16/2022	15:54:03	316	1	
75	4/16/2022	15:54:06	346	1	
76	4/16/2022	15:54:08	306	1	
77	4/16/2022	15:54:11	260	1	
78	4/16/2022	15:54:14	301	0	
79	4/16/2022	15:54:16	214	0	
80	4/16/2022	15:54:18	228	1	

SEND DATA FROM NODEMCU TO SHEETS

```
var sheet_id = '15-itebVVyWrgPlr5qjMDYk3LNG55fhygj66YDEG9pdw';  // Spreadsheet ID
var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet();
var newRow = sheet.getLastRow() + 1;
var rowData = [];
var Curr_Date = new Date();
rowData[0] = Curr_Date; // Date in column A
var Curr_Time = Utilities.formatDate(Curr_Date, "Asia/Kolkata", 'HH:mm:ss');
rowData[1] = Curr_Time; // Time in column B
for (var param in e.parameter) {
  Logger.log('In for loop, param=' + param);
  var value = stripQuotes(e.parameter[param]);
  Logger.log(param + ':' + e.parameter[param]);
  switch (param) {
    case 'smoke':
      rowData[2] = value; // Smoke in column C
      result = 'Smoke Written on column C';
      break;
    case 'flame':
      rowData[3] = value; // Flame in column D
      result += ' ,Flame Written on column D';
      break;
    default:
      result = "unsupported parameter";
  }
}
Logger.log(JSON.stringify(rowData));
var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
newRange.setValues([rowData]);
```

As shown in the picture in the left side, we have linked the google sheet using the google sheet id. So we are sending nodemcu data to the google sheet along with the current time.

SENDING THE DATA FROM GOOGLE SHEET TO FIREBASE

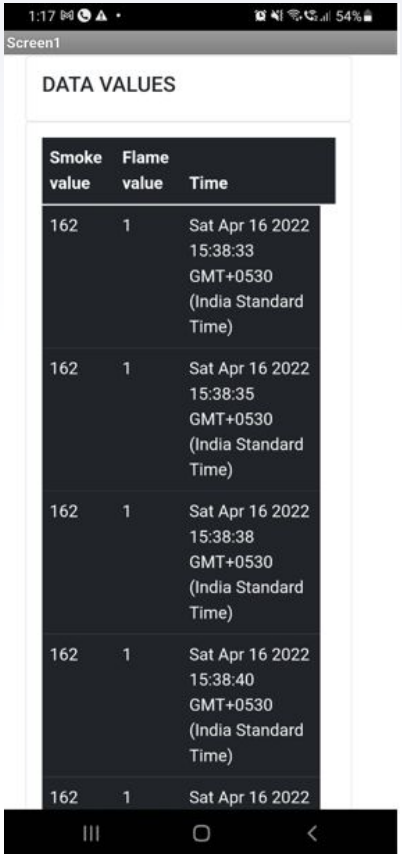
```
var dataToImport = {};
//console.log("size: ",data.length)
for (var i = 1; i < data.length; i++) {
  console.log("date: ",new Date())
  dataToImport[data[i][0]] = {};
  for (var j = 0; j < data[0].length; j++) {

    assign(dataToImport[data[i][0]], data[0][j].split("__"), data[i][j]);

  }
  var token = ScriptApp.getOAuthToken();

var firebaseUrl =
  getEnvironment().firebaseUrl + sheet.getParent().getId() + "/" + name;
var base = FirebaseApp.getDatabaseByUrl(firebaseUrl, token);
base.setData("", dataToImport);
}

var token = ScriptApp.getOAuthToken();

var firebaseUrl =
  getEnvironment().firebaseUrl + sheet.getParent().getId() + "/" + name;
var base = FirebaseApp.getDatabaseByUrl(firebaseUrl, token);
base.setData("", dataToImport);
```

The image shows the process of adding the google sheet value to the firebase. We are sending the data using the nested for loops and sending in the form of an object to the firebase server.

APP

- We built an android app, which makes people's work easy to check the status of their data
- We implemented the same as that we did for website

APP SCREENSHOTS



CHALLENGES FACED

- **CHALLENGE 1:** We were not able to connect nodeMCU to firebase directly, due to some package errors

  SOLVED: We tried to send the data from nodeMCU to firebase via google spreadsheets, which is very easy to handle as it is structured, sending and fetching data became easy

- **CHALLENGE 2:** When we send data to firebase, we are used to click the run button every time when we get new data

  SOLVED: Instead of having two different functions, we tried to send the data to firebase whenever we getting data from sensors

LIMITATION OF THE PROJECT

- False alarms can’t be accurately measured because of low accuracy sensors
- It is not insulated from external big fire accidents

FUTURE SCOPE

- We can add openCV library and thermal sensor to accurately calculate the size and temperature of the fire/smoke and easily judge about the fire situation

- We improve our project by insulating and making compact

FOR REFERENCE

CODE GITHUB LINK: https://github.com/iit2019234/fire_alarm_system_version_3.0
HEROKU WEBSITE: https://firealarmupdatedversion.herokuapp.com/