PANDORA

INFRA MONITERING TOOL

PURANJOT SINGH
TANISH KACKRIA
PRANAV MALHOTRA
DEVANSH AGGARWAL
SIDDHARTH CHAUHAN

PROBLEM DESCRIPTION

Team Pandora successfully developed an innovative cloud observability platform as part of the STGI Hackathon.The project focused on building a robust solution to monitor applications deployed in a Kubernetes cluster, specifically targeting two databases—PostgreSQL and MySQL.

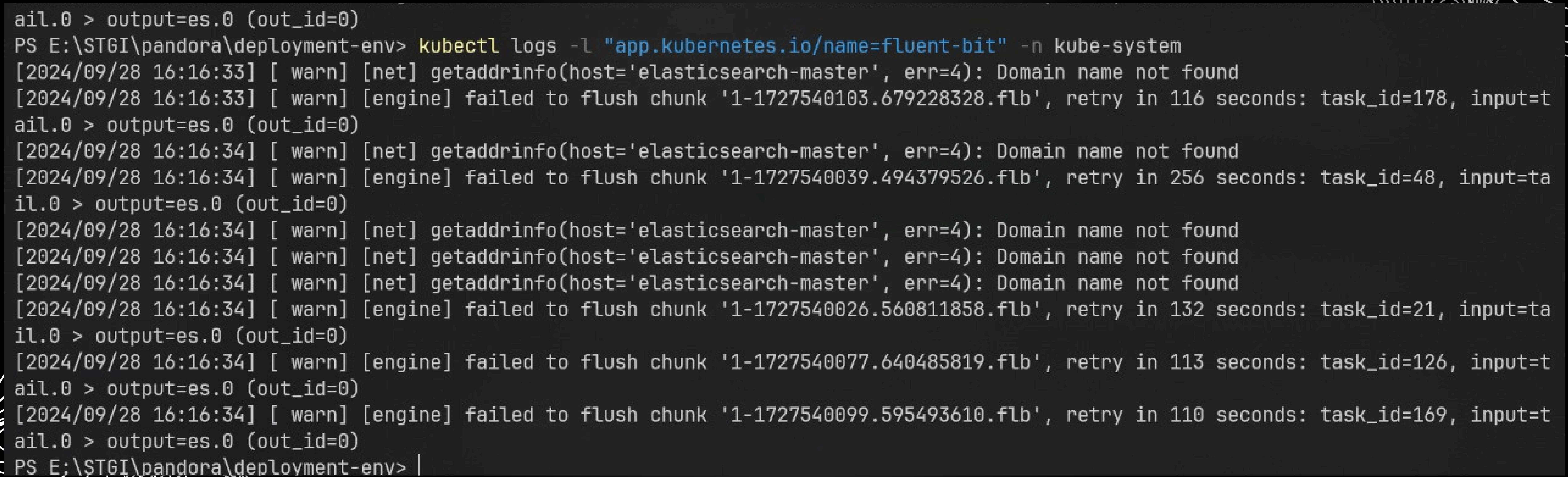
ROADMAP

- IDEA GENERATION
- GENERATION OF LOGS FROM MINIKUBE USING FLUENTBIT

- UI/UX USING NEXTJS
- FRONTEND

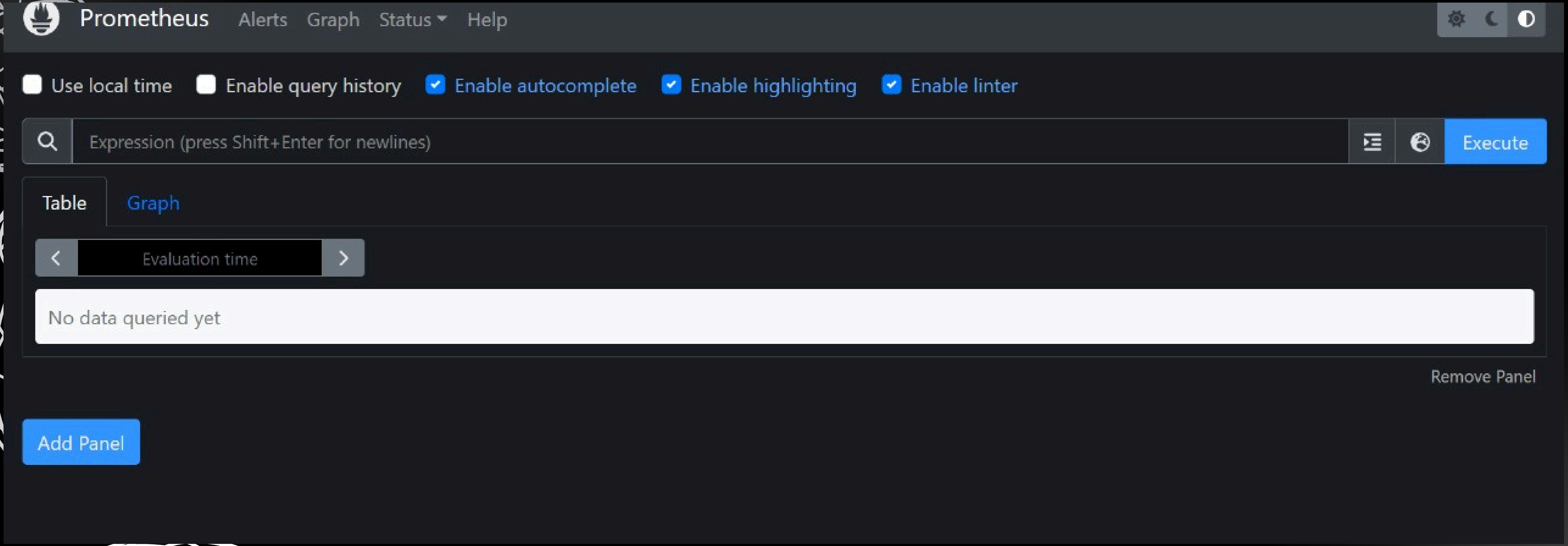
- LOGS STORAGE IN MONGODB
- ANALYTICAL DASHBOARD USING CHARTJS

LOGS FROM FLUENTBIT



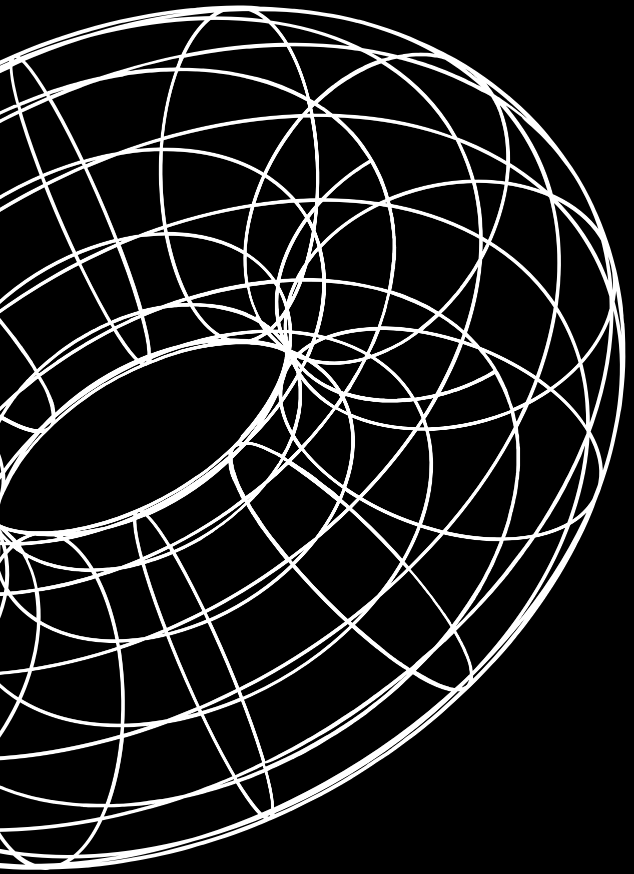
```
ail.0 > output=es.0 (out_id=0)
PS E:\STGI\pandora\deployment-env> kubectl logs -l "app.kubernetes.io/name=fluent-bit" -n kube-system
[2024/09/28 16:16:33] [ warn] [net] getaddrinfo(host='elasticsearch-master', err=4): Domain name not found
[2024/09/28 16:16:33] [ warn] [engine] failed to flush chunk '1-1727540103.679228328.flb', retry in 116 seconds: task_id=178, input=t
ail.0 > output=es.0 (out_id=0)
[2024/09/28 16:16:34] [ warn] [net] getaddrinfo(host='elasticsearch-master', err=4): Domain name not found
[2024/09/28 16:16:34] [ warn] [engine] failed to flush chunk '1-1727540039.494379526.flb', retry in 256 seconds: task_id=48, input=ta
il.0 > output=es.0 (out_id=0)
[2024/09/28 16:16:34] [ warn] [net] getaddrinfo(host='elasticsearch-master', err=4): Domain name not found
[2024/09/28 16:16:34] [ warn] [net] getaddrinfo(host='elasticsearch-master', err=4): Domain name not found
[2024/09/28 16:16:34] [ warn] [net] getaddrinfo(host='elasticsearch-master', err=4): Domain name not found
[2024/09/28 16:16:34] [ warn] [engine] failed to flush chunk '1-1727540026.560811858.flb', retry in 132 seconds: task_id=21, input=ta
il.0 > output=es.0 (out_id=0)
[2024/09/28 16:16:34] [ warn] [engine] failed to flush chunk '1-1727540077.640485819.flb', retry in 113 seconds: task_id=126, input=t
ail.0 > output=es.0 (out_id=0)
[2024/09/28 16:16:34] [ warn] [engine] failed to flush chunk '1-1727540099.595493610.flb', retry in 110 seconds: task_id=169, input=t
ail.0 > output=es.0 (out_id=0)
PS E:\STGI\pandora\deployment-env> |
```

WORKING PROMETHEUS



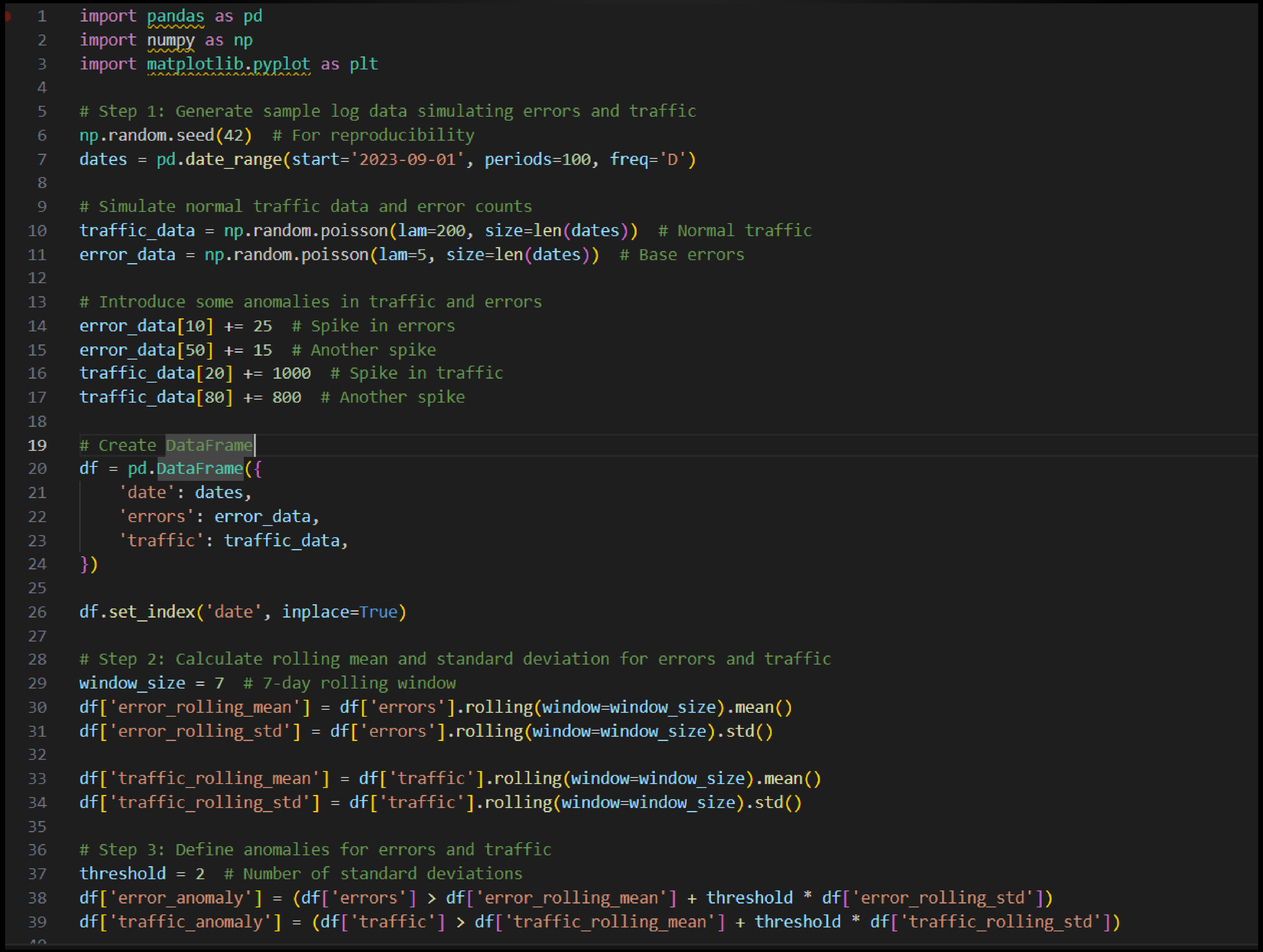
☐ Use local time ☐ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

No data queried yet

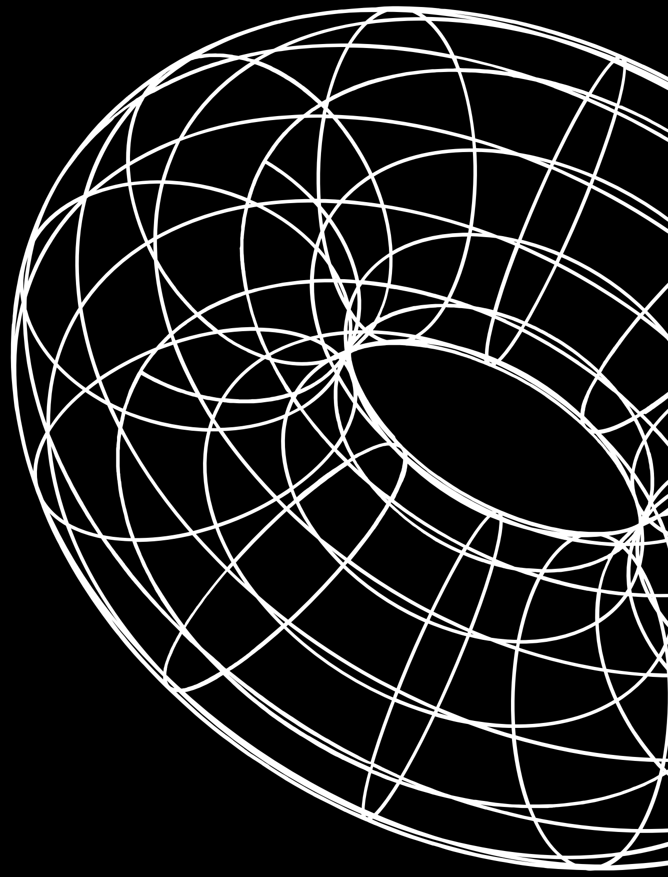


FEATURES

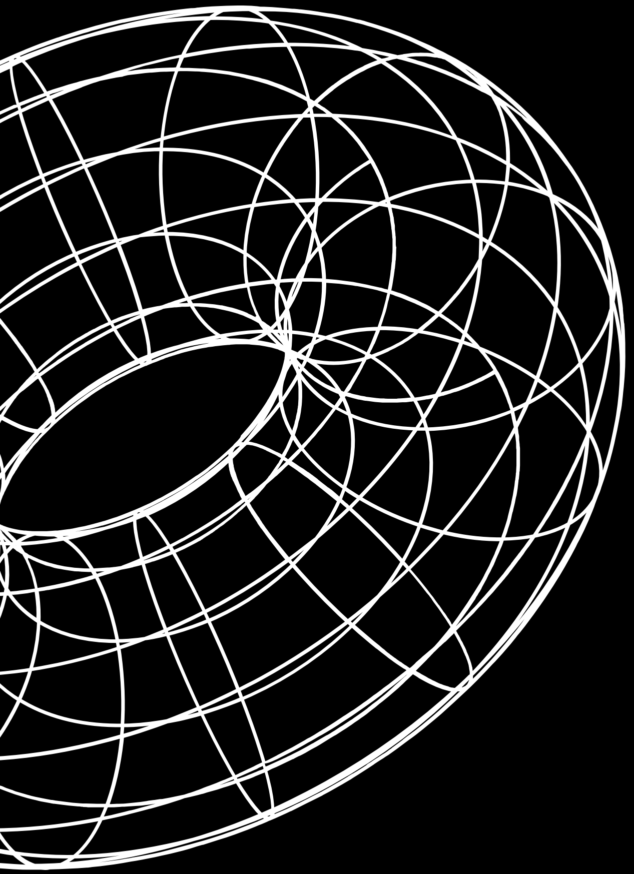
Real-Time Alerting System

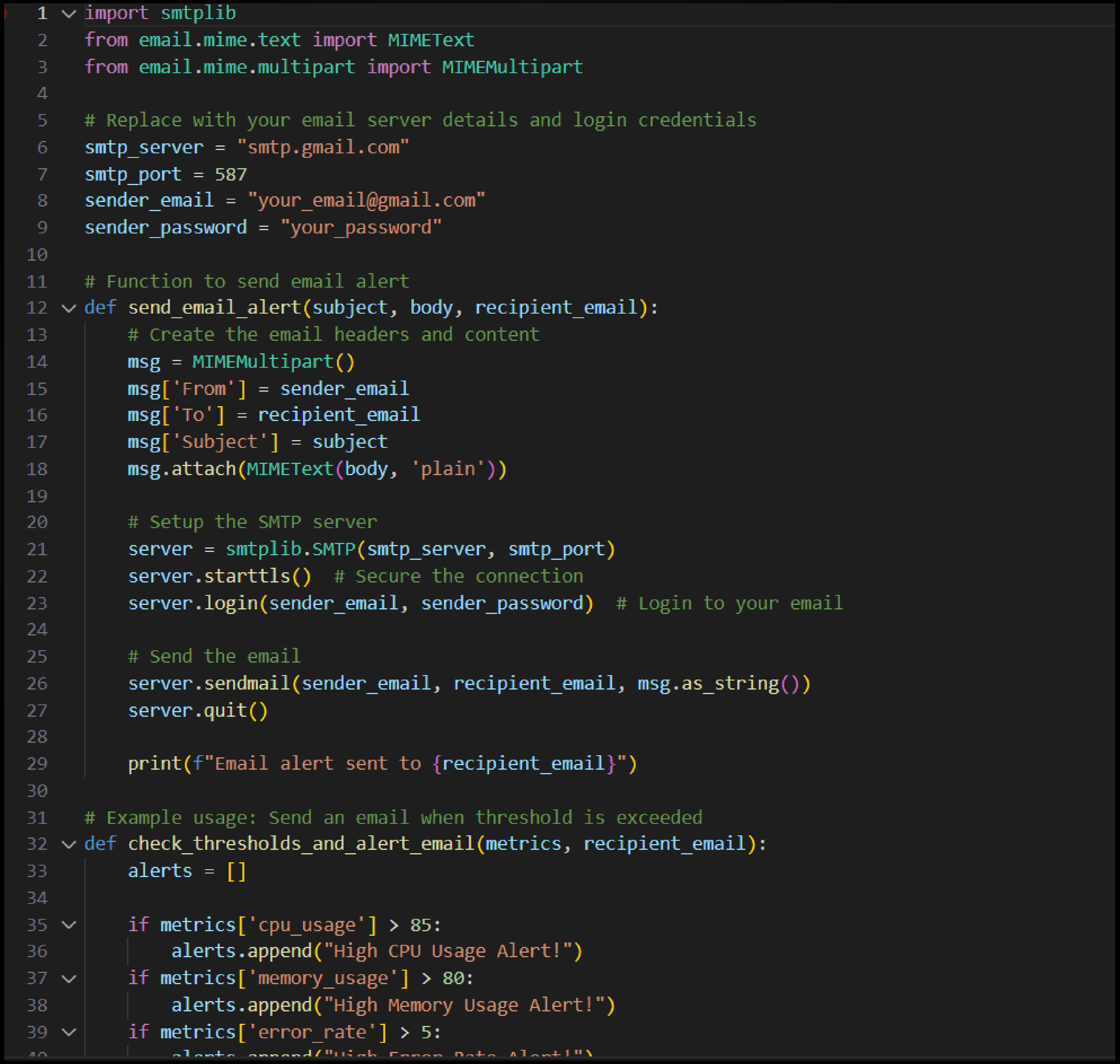


```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Step 1: Generate sample log data simulating errors and traffic
np.random.seed(42)  # For reproducibility
dates = pd.date_range(start='2023-09-01', periods=100, freq='D')

# Simulate normal traffic data and error counts
traffic_data = np.random.poisson(lam=200, size=len(dates))  # Normal traffic
error_data = np.random.poisson(lam=5, size=len(dates))  # Base errors

# Introduce some anomalies in traffic and errors
error_data[10] += 25  # Spike in errors
error_data[50] += 15  # Another spike
traffic_data[20] += 1000  # Spike in traffic
traffic_data[80] += 800  # Another spike

# Create DataFrame
df = pd.DataFrame({
    'date': dates,
    'errors': error_data,
    'traffic': traffic_data,
})

df.set_index('date', inplace=True)

# Step 2: Calculate rolling mean and standard deviation for errors and traffic
window_size = 7  # 7-day rolling window
df['error_rolling_mean'] = df['errors'].rolling(window=window_size).mean()
df['error_rolling_std'] = df['errors'].rolling(window=window_size).std()

df['traffic_rolling_mean'] = df['traffic'].rolling(window=window_size).mean()
df['traffic_rolling_std'] = df['traffic'].rolling(window=window_size).std()

# Step 3: Define anomalies for errors and traffic
threshold = 2  # Number of standard deviations
df['error_anomaly'] = (df['errors'] > df['error_rolling_mean'] + threshold * df['error_rolling_std'])
df['traffic_anomaly'] = (df['traffic'] > df['traffic_rolling_mean'] + threshold * df['traffic_rolling_std'])
```

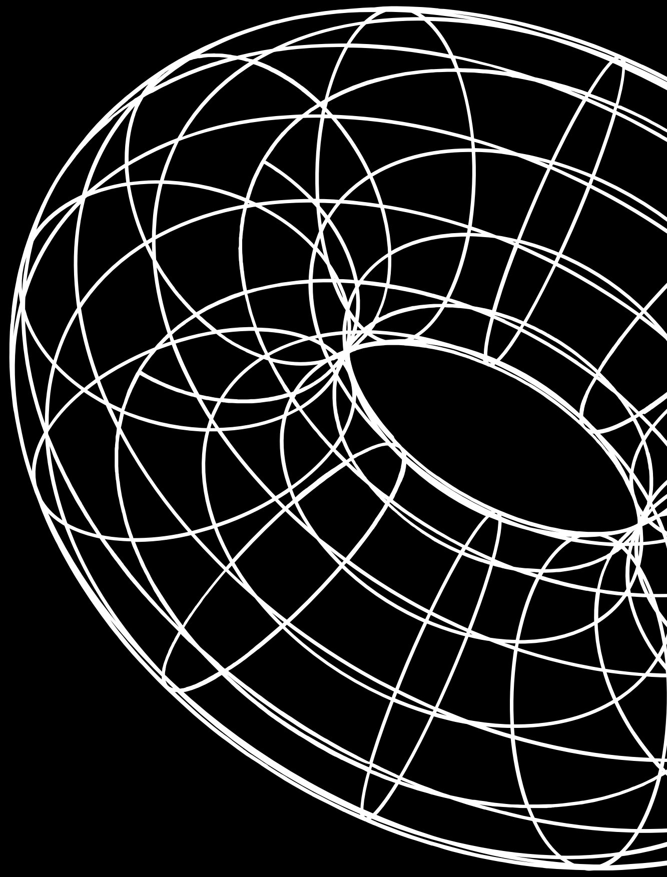


Log Anomaly Detection

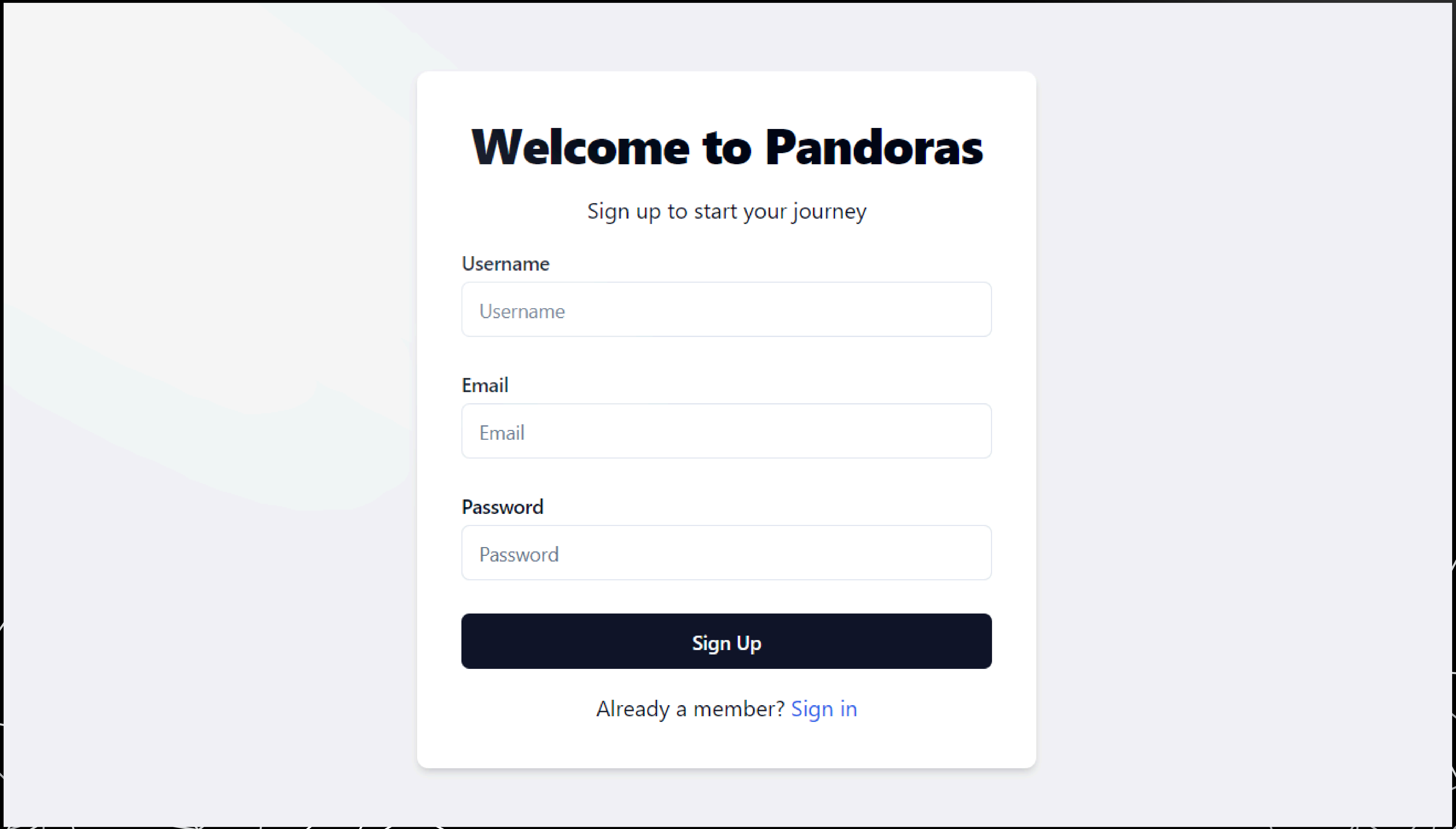




```python
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

# Replace with your email server details and login credentials
smtp_server = "smtp.gmail.com"
smtp_port = 587
sender_email = "your_email@gmail.com"
sender_password = "your_password"

# Function to send email alert
def send_email_alert(subject, body, recipient_email):
    # Create the email headers and content
    msg = MIMEMultipart()
    msg['From'] = sender_email
    msg['To'] = recipient_email
    msg['Subject'] = subject
    msg.attach(MIMEText(body, 'plain'))

    # Setup the SMTP server
    server = smtplib.SMTP(smtp_server, smtp_port)
    server.starttls()  # Secure the connection
    server.login(sender_email, sender_password)  # Login to your email

    # Send the email
    server.sendmail(sender_email, recipient_email, msg.as_string())
    server.quit()

    print(f"Email alert sent to {recipient_email}")

# Example usage: Send an email when threshold is exceeded
def check_thresholds_and_alert_email(metrics, recipient_email):
    alerts = []

    if metrics['cpu_usage'] > 85:
        alerts.append("High CPU Usage Alert!")
    if metrics['memory_usage'] > 80:
        alerts.append("High Memory Usage Alert!")
    if metrics['error_rate'] > 5:
        alerts.append("High Error Rate Alert!")
```



OUR SIGNUP PAGE



Welcome to Pandoras

Sign up to start your journey

Username

Username

Email

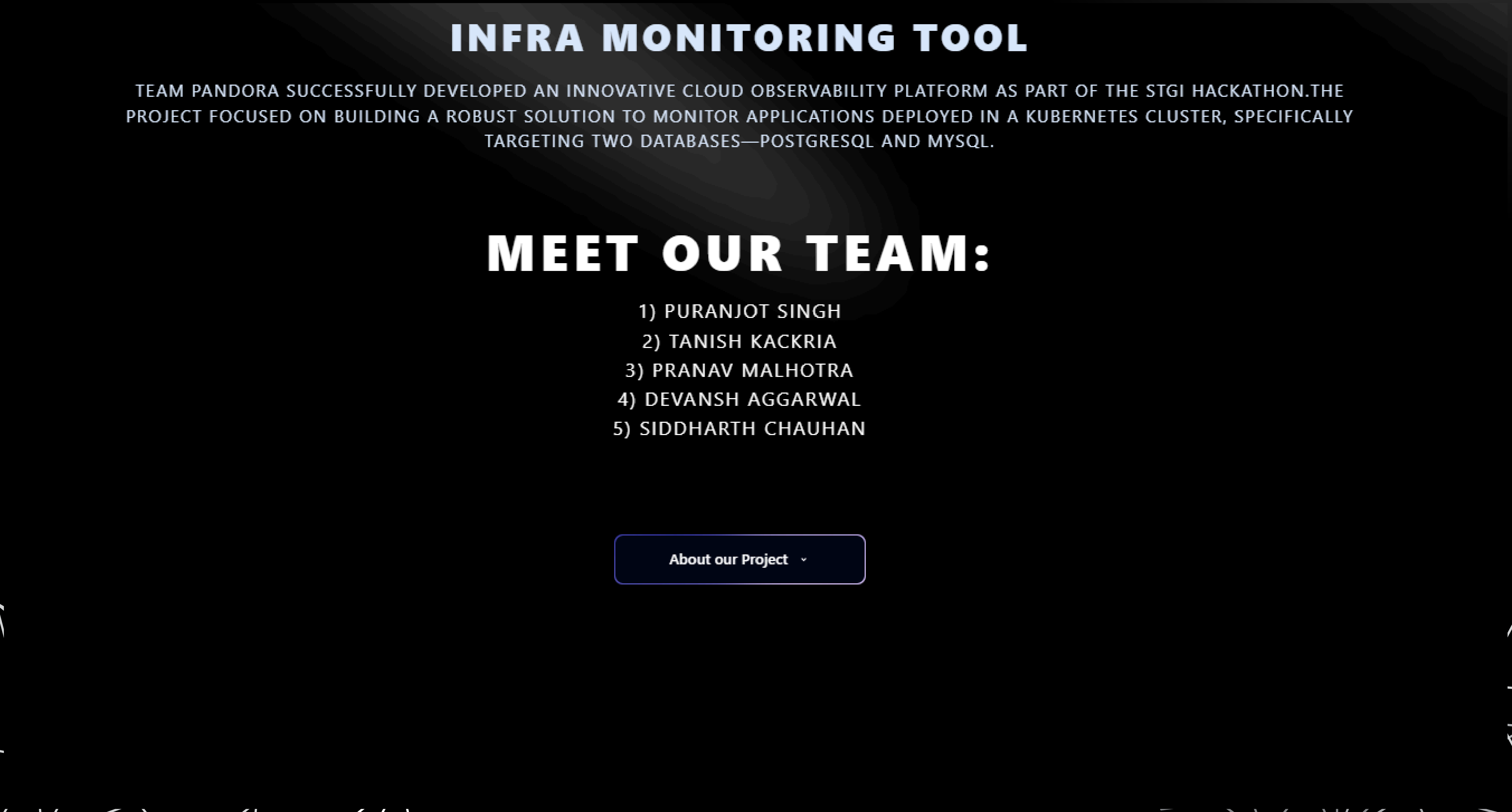
Email

Password

Password

Already a member? Sign in

OUR LANDING PAGE



INFRA MONITORING TOOL

TEAM PANDORA SUCCESSFULLY DEVELOPED AN INNOVATIVE CLOUD OBSERVABILITY PLATFORM AS PART OF THE STGI HACKATHON.THE PROJECT FOCUSED ON BUILDING A ROBUST SOLUTION TO MONITOR APPLICATIONS DEPLOYED IN A KUBERNETES CLUSTER, SPECIFICALLY TARGETING TWO DATABASES—POSTGRESQL AND MYSQL.

MEET OUR TEAM:

1) PURANJOT SINGH
2) TANISH KACKRIA
3) PRANAV MALHOTRA
4) DEVANSH AGGARWAL
5) SIDDHARTH CHAUHAN

THANK YOU