Course: IT485

Project: HawkinCloud

Members: Aliyev Omar, Haider Zaheer, Michael Hernandez, Nephthro Pierre.

REPORT #2

Management Guidance:

Our "Hawkincloud" first group meeting was scheduled for 19th February for the weekend. We will customize some basic ideas and information based on the project guidelines. During the last presentation, the instructor & other group members shared some key points. Due to scheduling conflicts, we will start working on the software next week to get its results before Presentation II.

Team Member participant:

According to the concept of the project create new establishments statements and steps applying for software. By following these steps, you can set up a Flask app that interacts with CloudTrail to retrieve information about your AWS account. Flexibility and adaptability are also important qualities for a team member participant. We can adapt to changing circumstances and are open to new ideas and approaches. And willing to take on new responsibilities and help in areas outside of their comfort zone. The CloudTrail project requires a team of dedicated and knowledgeable individuals, each bringing their unique skills and characteristics to the table.

Output Deliverables:

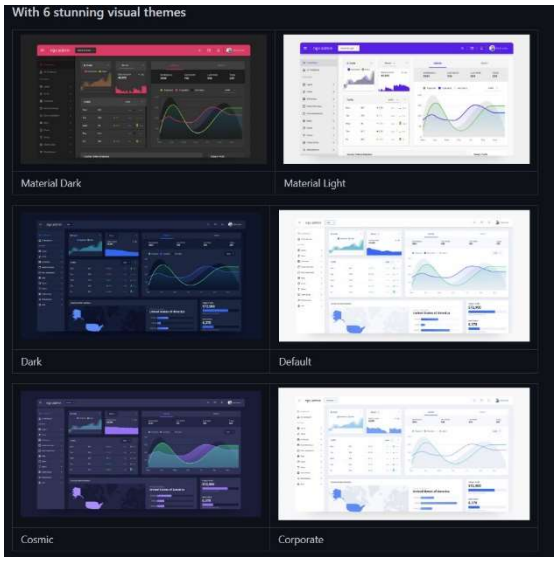
Participants in a CloudTrail project need a strong technical background, and experience with Amazon Web Services (AWS). The CloudTrail service must be thoroughly understood and navigated with ease. It requires a deep understanding of AWS's infrastructure, including CloudFormation, S3, and IAM. We must be able to understand and interpret the vast amount of data generated by CloudTrail in order to provide insights and recommendations. Developing high-quality code and improving our coding skills and knowledge should be our top priorities.

o.aliyev001@umb.edu
m.zaheer001@umb.edu
michael.hernandez003@umb.edu
nephthro.pierre001@umb.edu

HAWKINCLOUD BETA II 2.20.23

Current Status:

- Each progress of the software we are assuming to create a resource box and deploy there, who has knowledge problem about coding it will be more useful and recognize coding documentation. This fact is useful to put into the new statements and arguments start doing Front-end & Backend. We deployed py-env and placed configure requirements. The idea of the appearance researching is to stabilize the project's Front-end part. We are looking to use Angular +8, CSS grids, and Bootstrap.



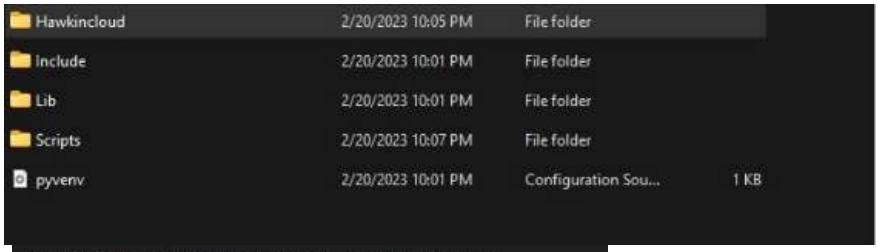
Sources:

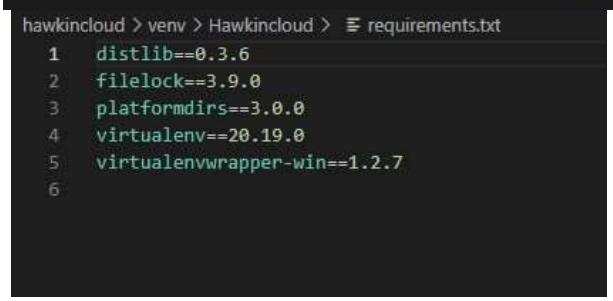
https://github.com/akveo/ngx-admin

https://www.w3.org/TR/css-grid-1/

- CloudTrail is a service offered by AWS that provides detailed information about actions taken in your AWS account. It can be used to monitor, troubleshoot, and audit your AWS resources. Flask is a popular Python web framework that can be used to build web applications. Define backend steps into project we need to be set up your AWS account and create a new CloudTrail trail. You can do this by logging in to the AWS Management Console and navigating to the CloudTrail service then Install the AWS SDK for Python (boto3) using pip, which is the recommended way to interact with AWS services from Python.

- We are currently providing the py-env installed and setup environment package. This is the initial configuration we are deploying for the login page and dashboard HTML layers. The stages should be created according to the steps we will understand in the project. Below are assets provided by us and each process is listed clearly.

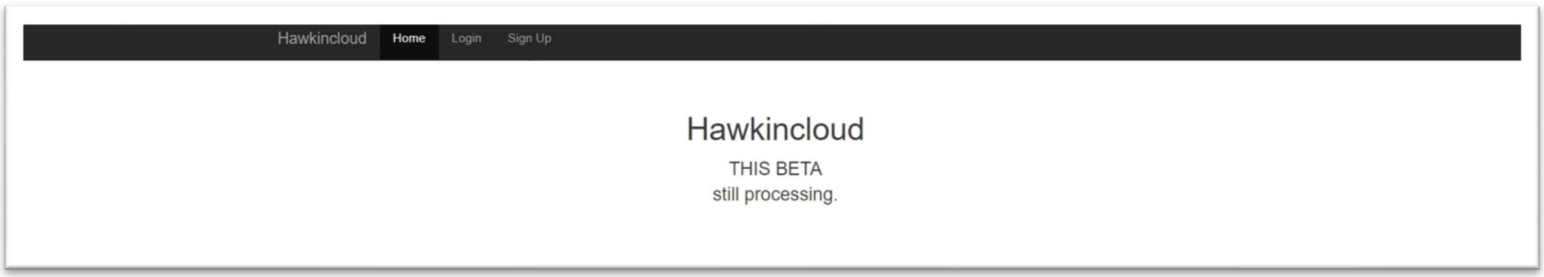


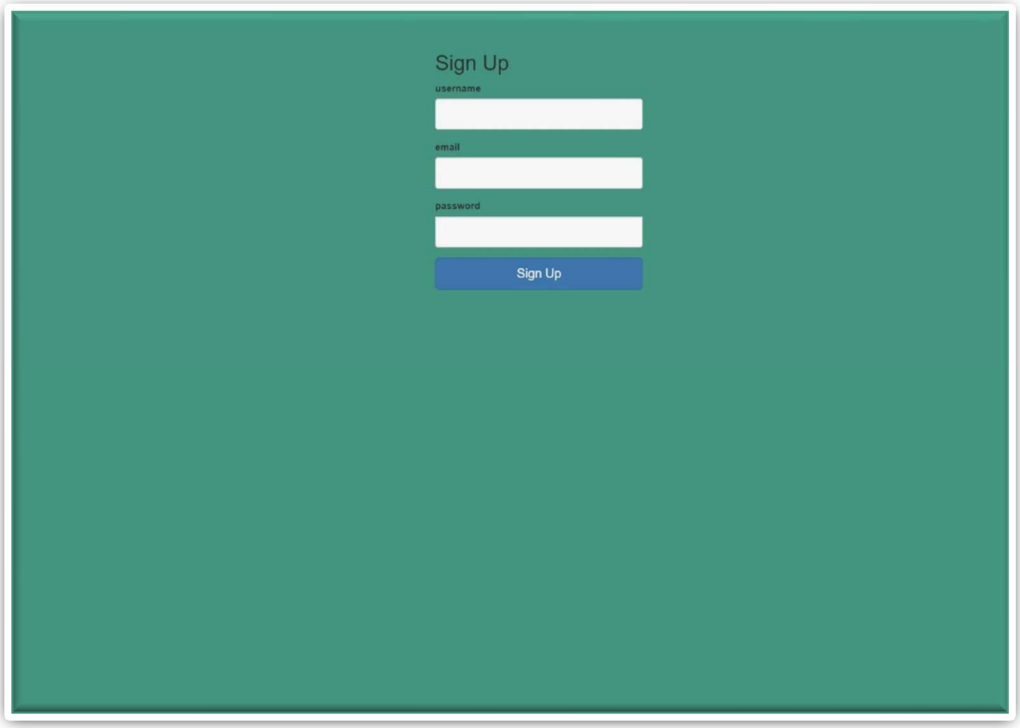



```python
from flask import Flask, render_template
from flask_bootstrap import Bootstrap

app = Flask(__name__)
Bootstrap(app)

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

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

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

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

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


o.aliyev001@umb.edu
m.zaheer001@umb.edu
michael.hernandez003@umb.edu
nephthro.pierre001@umb.edu

HAWKINCLOUD BETA II 2.20.23





- As well as we are working with charting libraries such as D3.js, Chart.js, Flask-Charts, angular-google-charts, or Highcharts. Creating real-time visualization will be an effective tool for embedding our project. The flask-charts documentation lists coding help for getting the refresh rate so that we can adjust. If you are pulling data from a URL, you can specify how often it should update.

```
my_chart.refresh = 5000 # 5 seconds interval
```

Sources:

https://github.com/albinmedoc/flask-charts

https://www.npmjs.com/package/angular-google-charts

o.aliyev001@umb.edu
m.zaheer001@umb.edu
michael.hernandez003@umb.edu
nephthro.pierre001@umb.edu

o.aliyev001@umb.edu
m.zaheer001@umb.edu
michael.hernandez003@umb.edu
nephthro.pierre001@umb.edu