



**Smart
Internz**



Cloud-Hosted Banking Data Analysis And Reporting System On AWS

TEAM PROJECT ID : NM2024TMID13709

Members name with NM ID:

- | | |
|--|------------------------|
| 1) 5867CE5A83C685D85D1434B0F271FE36 | SARAVANAKUMAR K |
| 2) BC733C2D075062C88574135355A95BF4 | PRASANA |
| VENKATESAN S | |
| 3) DFCF6D973C6689DB44B4DFEB974208B6 | SANJAI RAM K |
| 4) C861677FE325B99BEC94EA48EE3664F4 | |
| MUGESH PANDI N | |

Scenarios

Scenario

1:

- Real-time Transaction Monitoring: Sarah, a bank's fraud detection analyst, logs into CloudBank Analytics during her morning routine. The dashboard immediately alerts her to unusual transaction patterns detected overnight. Using the real-time analytics feature, Sarah quickly investigates the flagged transactions, confirms a potential fraud attempt, and takes immediate action to protect the affected accounts.

Scenario 2:

Custom Report Generation

John, a financial manager, needs to prepare a comprehensive quarterly report for the board meeting. He logs into CloudBank Analytics and uses the custom report generation feature. John selects various metrics such as loan performance, deposit growth, and customer acquisition rates. The system, leveraging AWS Lambda, quickly processes the vast amount of data stored in Amazon RDS and generates a detailed report, which is then stored in Amazon S3 for easy access and sharing.

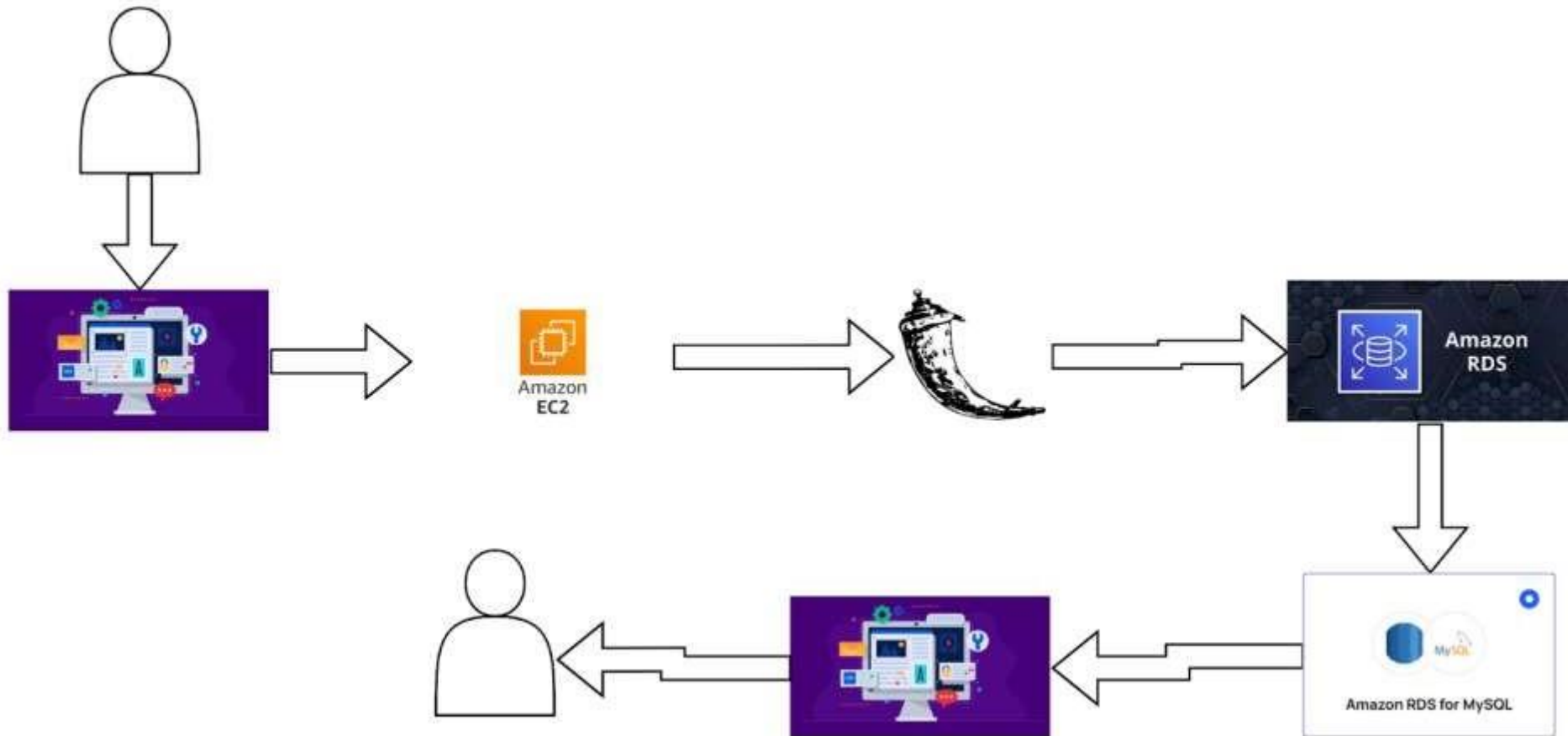
Scenario 3:

Regulatory Compliance Monitoring

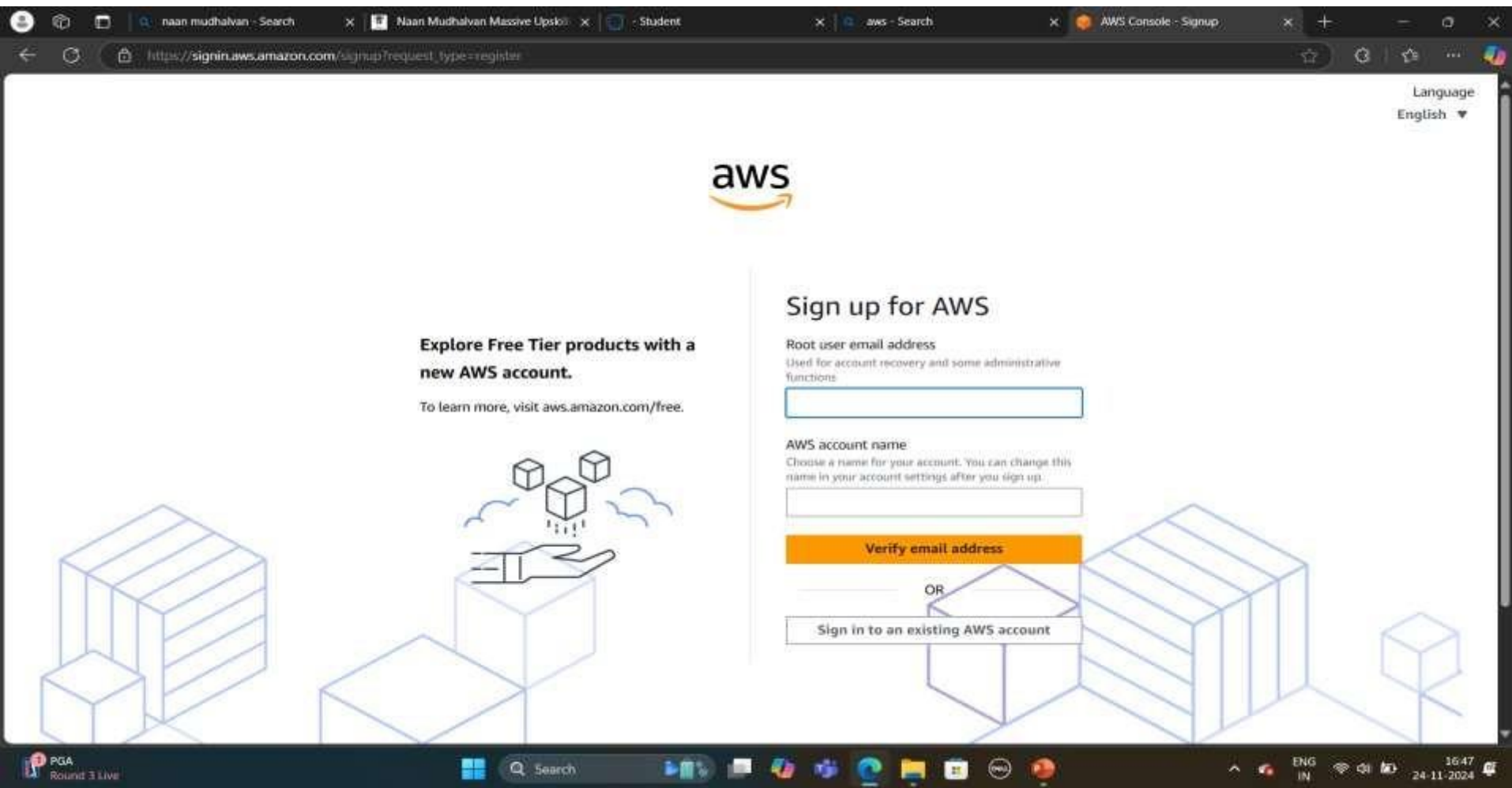
Lisa, a compliance officer, uses CloudBank Analytics to ensure the bank meets all regulatory requirements. She accesses a specialized dashboard that tracks key compliance metrics in real-time.

When she notices that a particular metric is approaching a regulatory threshold, she uses the system to drill down into the underlying data, identify the root cause, and initiate corrective actions before any compliance issues arise.

Architecture :



AWS Account Creation :



The screenshot shows the AWS sign-up page in a web browser. The browser's address bar displays the URL `https://signin.aws.amazon.com/signup?request_type=register`. The page features the AWS logo at the top center. On the left, there is a promotional message about the Free Tier. On the right, the 'Sign up for AWS' section contains two input fields for email and account name, followed by a 'Verify email address' button. Below this is an 'OR' separator and a link to sign in to an existing account. The background of the page is decorated with a blue line-art illustration of server racks and a hand placing a cube on top of them.

Language
English ▼

aws

Sign up for AWS

Root user email address
Used for account recovery and some administrative functions

AWS account name
Choose a name for your account. You can change this name in your account settings after you sign up.

Verify email address

OR

[Sign in to an existing AWS account](#)

Explore Free Tier products with a new AWS account.
To learn more, visit aws.amazon.com/free.

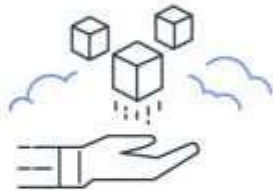
PGA Round 3 Live

Search

ENG IN 16:47 24-11-2024

Explore Free Tier products with a new AWS account.

To learn more, visit aws.amazon.com/free.



Sign up for AWS

Root user email address

Used for account recovery and some administrative functions.

hkesavan2004@gmail.com

AWS account name

Choose a name for your account. You can change this name in your account settings after you sign up.

keshav

Verify email address

OR

Sign in to an existing AWS account




Security check

Please click verify to start your security challenge.

Verify



Secure verification

 We will not charge you for usage below AWS Free Tier limits. We may temporarily hold up to \$1 USD (or an equivalent amount in local currency) as a pending transaction for 3-5 days to verify your identity.



Sign up for AWS

Billing Information

Billing country

Your billing country determines the payment methods available to you to pay for AWS services.

India

Credit or Debit card number



AWS accepts most major credit and debit cards. To learn more about payment options, review our [FAQ](#)

Expiration date

Month

Year

Security code

CVV/CVC

Cardholder's name



Set Up IAM Users and Permissions :

IAM Dashboard

Security recommendations 0



Root user has MFA

Having multi-factor authentication (MFA) for the root user improves security for this account.



Root user has no active access keys

Using access keys attached to an IAM user instead of the root user improves security.

IAM resources



Resources in this AWS Account

User groups

1

Users

3

Roles

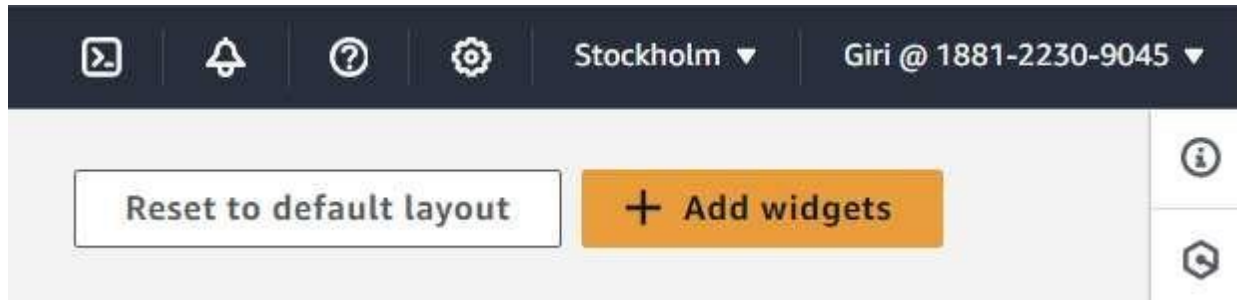
8

Policies

0

Identity providers

1



1. Access the IAM console from the AWS Management Console.
2. Create a new IAM user for yourself with administrative access.
3. Set up multi-factor authentication (MFA) for added security.
4. Create a group for developers and assign necessary permissions.
5. Generate access keys for programmatic access if needed.

Local Development Environment Setup :

1. Install Python and pip on your local machine.
2. Install Flask and other necessary Python packages (e.g., `flask-sqlalchemy`, `mysql-connector-python`).
3. Set up a virtual environment for your project.

```
Command Prompt
Microsoft Windows [Version 10.0.19045.4957]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>python --version
Python 3.12.7

C:\Users\HP>pip --version
pip 24.2 from C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.12_3.12.2032.0_x64__qbz5n2kfra8p0\Lib\site-packages\pip (python 3.12)

C:\Users\HP>pip install numpy
Defaulting to user installation because normal site-packages is not writeable
Collecting numpy
  Downloading numpy-2.1.3-cp312-cp312-win_amd64.whl.metadata (60 kB)
  Downloading numpy-2.1.3-cp312-cp312-win_amd64.whl (12.6 MB)
    ----- 12.6/12.6 MB 2.5 MB/s eta 0:00:00
Installing collected packages: numpy
  WARNING: The scripts f2py.exe and numpy-config.exe are installed in 'C:\Users\HP\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\
local-packages\Python312\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-2.1.3

[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: C:\Users\HP\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip

C:\Users\HP>python -m pip install -U pip
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pip in c:\program files\windowsapps\pythonsoftwarefoundation.python.3.12_3.12.2032.0_x64__qbz5n2kfra8p0\lib\site-packages (24.2)
Collecting pip
  Downloading pip-24.3.1-py3-none-any.whl.metadata (3.7 kB)
  Downloading pip-24.3.1-py3-none-any.whl (1.8 MB)
    ----- 1.8/1.8 MB 2.9 MB/s eta 0:00:00
Installing collected packages: pip
  WARNING: The scripts pip.exe, pip3.12.exe and pip3.exe are installed in 'C:\Users\HP\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCa
che\local-packages\Python312\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-24.3.1

C:\Users\HP>pip install numpy
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\users\hp\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-p
ackages (2.1.3)
```

```

C:\Users\HP>pip install flask-sqlalchemy
Defaulting to user installation because normal site-packages is not writeable
Collecting flask-sqlalchemy
  Downloading flask_sqlalchemy-3.1.1-py3-none-any.whl.metadata (3.4 kB)
Requirement already satisfied: flask>=2.2.5 in c:\users\hp\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from flask-sqlalchemy) (3.0.3)
Collecting sqlalchemy>=2.0.16 (from flask-sqlalchemy)
  Downloading SQLAlchemy-2.0.36-cp312-cp312-win_amd64.whl.metadata (9.9 kB)
Requirement already satisfied: Werkzeug>=3.0.0 in c:\users\hp\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from flask>=2.2.5->flask-sqlalchemy) (3.1.1)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\hp\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from flask>=2.2.5->flask-sqlalchemy) (3.1.4)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\hp\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from flask>=2.2.5->flask-sqlalchemy) (2.2.0)
Requirement already satisfied: click>=8.1.3 in c:\users\hp\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from flask>=2.2.5->flask-sqlalchemy) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in c:\users\hp\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from flask>=2.2.5->flask-sqlalchemy) (1.8.2)
Collecting typing-extensions>=4.6.0 (from sqlalchemy>=2.0.16->flask-sqlalchemy)
  Downloading typing_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)
Collecting greenlet!=0.4.17 (from sqlalchemy>=2.0.16->flask-sqlalchemy)
  Downloading greenlet-3.1.1-cp312-cp312-win_amd64.whl.metadata (3.9 kB)
Requirement already satisfied: colorama in c:\users\hp\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from click>=8.1.3->flask>=2.2.5->flask-sqlalchemy) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\hp\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from Jinja2>=3.1.2->flask>=2.2.5->flask-sqlalchemy) (3.0.2)
Downloading flask_sqlalchemy-3.1.1-py3-none-any.whl (25 kB)
Downloading SQLAlchemy-2.0.36-cp312-cp312-win_amd64.whl (2.1 MB)
----- 2.1/2.1 MB 4.5 MB/s eta 0:00:00
Downloading greenlet-3.1.1-cp312-cp312-win_amd64.whl (299 kB)
Downloading typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: typing-extensions, greenlet, sqlalchemy, flask-sqlalchemy
Successfully installed flask-sqlalchemy-3.1.1 greenlet-3.1.1 sqlalchemy-2.0.36 typing-extensions-4.12.2

C:\Users\HP>pip install mysql-connector-python
Defaulting to user installation because normal site-packages is not writeable
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.1.0-cp312-cp312-win_amd64.whl.metadata (6.2 kB)
Downloading mysql_connector_python-9.1.0-cp312-cp312-win_amd64.whl (16.1 MB)
----- 16.1/16.1 MB 5.2 MB/s eta 0:00:00
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.1.0

```

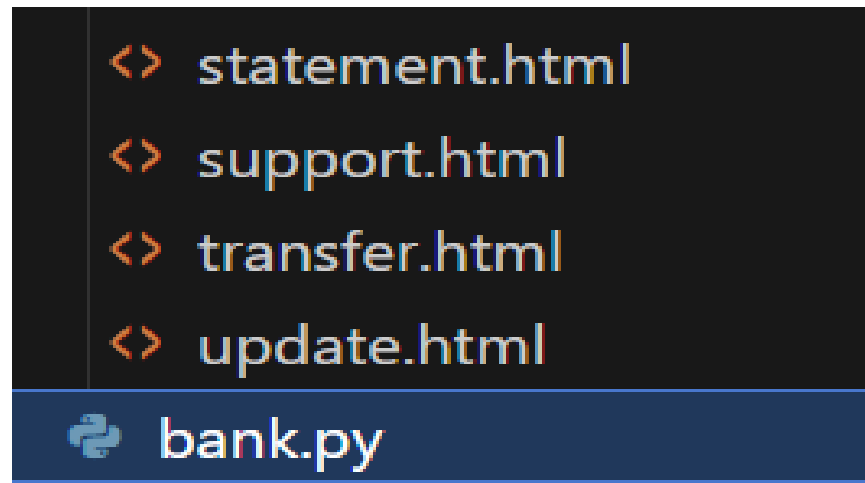
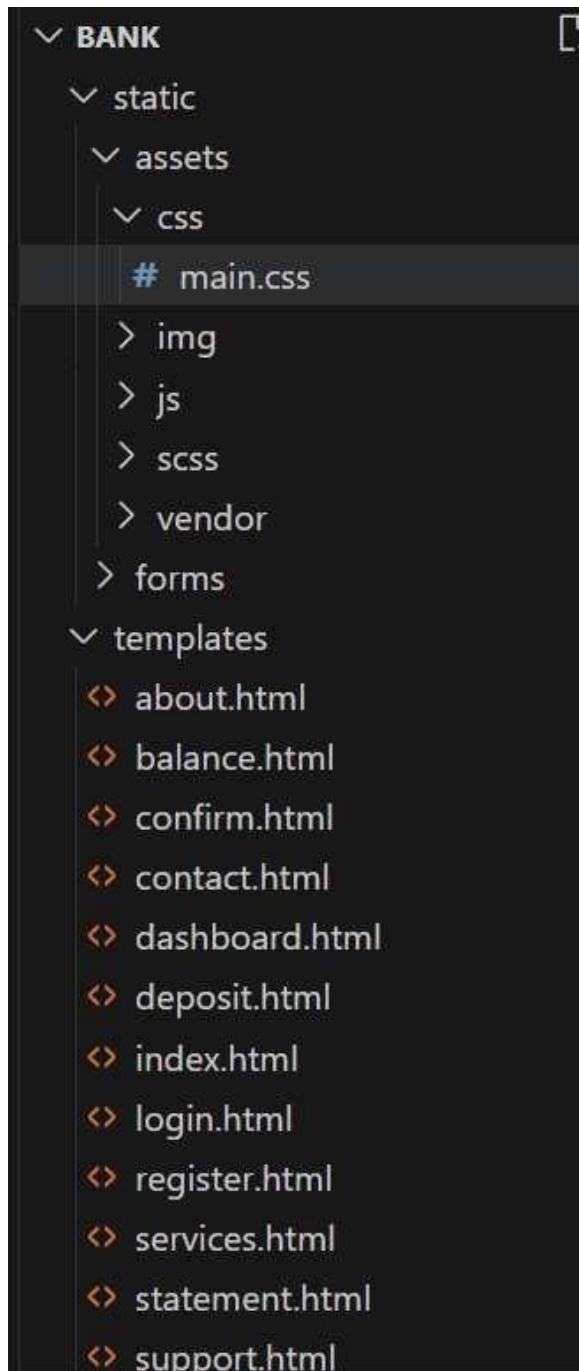
Command Prompt

Command Prompt

```
C:\Users\HP>pip install flask
Defaulting to user installation because normal site-packages is not writeable
Collecting flask
  Downloading flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Collecting Werkzeug>=3.0.0 (from flask)
  Downloading werkzeug-3.1.1-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from flask)
  Downloading jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.1.2 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.6.2 (from flask)
  Downloading blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)
Collecting colorama (from click>=8.1.3->flask)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)
  Downloading MarkupSafe-3.0.2-cp312-cp312-win_amd64.whl.metadata (4.1 kB)
Downloading flask-3.0.3-py3-none-any.whl (101 kB)
Downloading blinker-1.8.2-py3-none-any.whl (9.5 kB)
Downloading click-8.1.7-py3-none-any.whl (97 kB)
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
Downloading werkzeug-3.1.1-py3-none-any.whl (224 kB)
Downloading MarkupSafe-3.0.2-cp312-cp312-win_amd64.whl (15 kB)
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, flask
WARNING: The script flask.exe is installed in 'C:\Users\HP\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-packages\Python312\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed Jinja2-3.1.4 MarkupSafe-3.0.2 Werkzeug-3.1.1 blinker-1.8.2 click-8.1.7 colorama-0.4.6 flask-3.0.3 itsdangerous-2.2.0
```

Create Flask Application Structure :

1. Create a new directory for your Easybank project.
2. Set up a basic Flask application structure:



AWS RDS Setup and MySQL Integration :

Create Web Pages

1. Design and create HTML templates for your banking application, including:

- Home page
- User registration page
- Login page
- Account dashboard
- Transaction page
- Account creation page
- Check Balance page
- Deposit page

- Services page
- Contact Page
- confirm page
- Account Login page
- Statement page

2. Place these templates in the `templates/` directory.

3. Create CSS files in the `static/css/` directory for styling.

4. Add any necessary JavaScript files in the `static/js/` directory.

```
File Edit Selection View Go Run Terminal Help Search [Administrator]
banking.html x
C:\Users\Mugesh> banking.html > html
2 <html lang="en">
3 <body>
4 <center>
5 <form action="/register" method="POST">
6 <body>
72 <button type="submit">Deposit</button>
73 <h2>Our Services</h2>
74 <li>Online Banking</li>
75 <li>Loans</li>
76 <li>Investments</li>
77 <li>Credit Cards</li>
78 <h2>Contact Us</h2>
79 <form action="/contact" method="POST">
80 <label for="name">Name:</label>
81 <input type="text" id="name" name="name" required><br><br>
82 <label for="email">Email:</label>
83 <input type="email" id="email" name="email" required><br><br>
84 <label for="message">Message:</label><br>
85 <textarea id="message" name="message" required></textarea><br><br>
86 <button type="submit">Submit</button>
87 <h2>Transaction Confirmation</h2>
88 <p>Your transaction was successful</p>
89 <a href="/dashboard.html">Go to Dashboard</a>
90 <h2>Account Login</h2>
91 <form action="/account_login" method="POST">
92 <label for="account_number">Account Number:</label>
93 <input type="text" id="account_number" name="account_number" required><br><br>
94 <label for="password">Password:</label>
95 <input type="password" id="password" name="password" required><br><br>
96 <button type="submit">Login</button>
97 <h2>Account Statement</h2>
98 </center>
99 </body>
100 </html>
```

File Edit Selection View Go Run Terminal Help Search [Administrator]

Walkthrough: Setup VS Code banking.html

C:\Users\> Users\> Mugesh\> banking.html > html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Banking Application - Home</title>
7 </head>
8 <body>
9   <center>
10    <h1>Welcome to Our Bank</h1>
11    <nav>
12      <a href="login.html">Login</a>
13      <a href="register.html">Register</a>
14      <a href="services.html">Services</a>
15      <a href="contact.html">Contact Us</a>
16    </nav>
17    <h2>Register</h2>
18    <form action="/register" method="POST">
19      <label for="name">Name:</label>
20      <input type="text" id="name" name="name" required><br><br>
21
22      <label for="email">Email:</label>
23      <input type="email" id="email" name="email" required><br><br>
24
25      <label for="password">Password:</label>
26      <input type="password" id="password" name="password" required><br><br>
27
28      <button type="submit">Register</button>
29    </form>
30    <h2>Login</h2>
31    <form action="/login" method="POST">
32      <label for="email">Email:</label>
33      <input type="email" id="email" name="email" required><br><br>
34
35      <label for="password">Password:</label>
36      <input type="password" id="password" name="password" required><br><br>
37
38      <button type="submit">Login</button>
39    </form>
40  </center>
41</body>
42</html>
```

Ln 105, Col 11 Spaces: 4 UTF-8 CRLF HTML

Implement Flask Routes and Views

1. In `app.py`, create Flask routes for each of your web pages.
2. Implement view functions to render the appropriate templates.
3. Add form handling for user input (registration, login, transactions, etc.).

[illegible]

1.from flask import Flask, render_template, request, redirect, url_for:

- Imports the necessary modules from the Flask web framework.
- Flask is the main application class.
- render_template is used to render HTML templates.
- request is used to access the current request object.
- redirect is used to redirect the user to a different URL.
- url_for is used to generate URLs for routes.

2.import mysql.connector:

- Imports the MySQL Connector/Python module, which is used to connect to a MySQL database.

3.from flask import flash, session:

- Imports the flash and session modules from Flask.
- flash is used to display flash messages to the user.
- session is used to store and retrieve data in the user's session.

4.from datetime import datetime:

- Imports the datetime module, which is used to work with dates and times.

```

21 # Function to establish a database connection
22 def get_db_connection():
23     try:
24         return cnxpool.get_connection()
25     except mysql.connector.Error as err:
26         print(f"Error: {err}")
27         return None
28
29 @app.route("/test-db-connection")
30 def test_db_connection():
31     try:
32         conn = get_db_connection()
33         cursor = conn.cursor()
34         cursor.execute("SELECT DATABASE();") # Test query to check connection
35         db_name = cursor.fetchone()
36         cursor.close()
37         conn.close()
38         return f"Connected to the database: {db_name[0]}"
39     except mysql.connector.Error as err:
40         return f"Error: {err}"
41

```

The provided code is used to:

1. Establish a connection to a MySQL database using a connection pool.
2. Test the connection by executing a query.
3. Handle any errors that occur during the connection process.

AWS RDS Setup And MySQL Integration :

Create Amazon RDS Instance:

Access RDS Console from the AWS Management Console.

Create a new RDS instance:

- Choose MySQL as the engine type.
Select an appropriate instance size (e.g., db.t3.micro for testing).
- Configure storage, network settings, and security groups.
Set up the master username and password.
- Make sure to allow connections from your local IP for development and testing.

Services

Q

N. Virg

voclabs/user3564878-MOUNIKA_N @ 5374-

Amazon RDS

Dashboard

Databases

Query Editor

Performance insights

Snapshots

Exports in Amazon S3

Automated backups

Reserved instances

Proxies

Subnet groups

Parameter groups

Option groups

Custom engine versions

Zero-ETL integrations

New

Creating database database-1

Your database might take a few minutes to launch. You can use settings from database-1 to simplify configuration of suggested database add-ons while we finish creating your DB for you.

View credential details

Introducing Global Database writer endpoint

Each global cluster now has a writer endpoint that you can use to send your application's requests to the writer instance in the primary cluster of your Global Database. Aurora automatically updates the endpoint upon a cross-region failover or switchover operation, ensuring that requests are routed to the writer instance in the new primary cluster without the need for changes to your application code or configuration. [Learn more](#)

Notifications

0 0 0 0 2 0

RDS > Databases

Consider creating a Blue/Green Deployment to minimize downtime during upgrades

CloudShell

Feedback

Privacy

Terms

Cookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates.

Deployment provides a staging environment for changes to production databases. [RDS User Guide](#) [Aurora User Guide](#)

Databases (1)

Group resources

Modify

Actions

Restore from S3

Create database

Filter by databases

< 1 >

DB identifier

database-1

4

Connect To RDS Using MySQL Client:

- Install MySQL client on your local machine if not already done.
- Use the following command to connect to your RDS instance:

Copy

```
mysql -h <your-rds-endpoint> -P 3306 -u <your-username> -p
```

- Enter your RDS master password when prompted.

Setup New Connection

Connection Name: Type a name for the connection

Connection Method: Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: Port: Name or IP address of the server host - and TCP/IP port.

Username: Name of the user to connect with.

Password: The user's password. Will be requested later if it's not set.

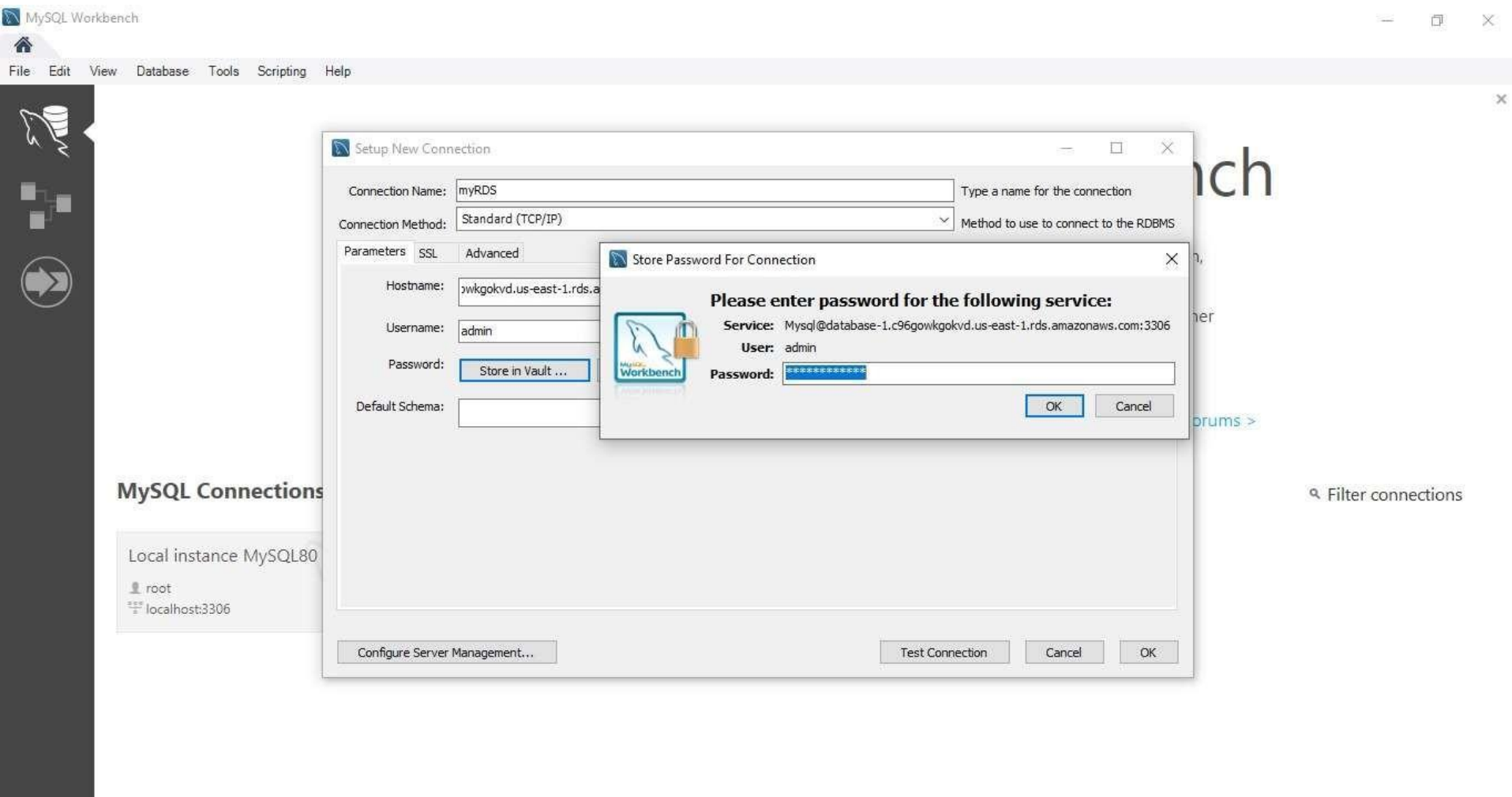
Default Schema: The schema to use as default schema. Leave blank to select it later.

MySQL Connections

Local instance MySQL80

root
localhost:3306

Filter connections



EC2 Deployment :

Launch EC2 Instance :

The screenshot displays the AWS Management Console interface. At the top, a green notification banner states: "Inbound security group rules successfully modified on security group (sg-09fe20b6aa14de176 | default)". Below this, the breadcrumb navigation shows "EC2 > Security Groups > sg-09fe20b6aa14de176 - default". The main heading is "sg-09fe20b6aa14de176 - default" with an "Actions" dropdown menu to its right. A "Details" section contains a table with the following information:

Security group name default	Security group ID sg-09fe20b6aa14de176	Description default VPC security group	VPC ID vpc-0b9b0f7bce0509e72
Owner 730335604592	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Below the table, there are tabs for "Inbound rules", "Outbound rules", "Sharing - new", "VPC associations - new", and "Tags". The "Inbound rules" tab is currently selected. The left sidebar shows the navigation menu with categories like "Instances", "Images", and "Elastic Block Store". The bottom of the console features a footer with "CloudShell", "Feedback", and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for "Privacy", "Terms", and "Cookie preferences".

←

→

↺

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:group-id=sg-09fe20b6aa14de176

☆

☰

Verify it's you

⋮

aws

Services

Search

[Alt+S]

📧

🔔

?

⚙️

N. Virginia

voclabs/user3616372=DHIVYADHARSHINI_N @ 7303-3560-4592

Dashboard

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity

Reservations New

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

✔️ Outbound security group rules successfully modified on security group (sg-09fe20b6aa14de176 | default)

▶ Details

[EC2](#) > [Security Groups](#) > sg-09fe20b6aa14de176 - default

sg-09fe20b6aa14de176 - default

Actions

Details

Security group name	Security group ID	Description	VPC ID
📄 default	📄 sg-09fe20b6aa14de176	📄 default VPC security group	📄 vpc-0b9b0f7bce0509e72
Owner	Inbound rules count	Outbound rules count	
📄 730335604592	1 Permission entry	2 Permission entries	

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

←→↻🔍

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

☆🔒 Verify it's you

aws

Services

Search

[Alt+S]

🖨⏏🔍⚙

N. Virginia

voclabs/user3616372=DHIVYADHARSHINI_N @ 7303-3560-4592

Dashboard

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity

Reservations New

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Instances (2) Info

Last updated less than a minute ago

🔄

Connect

Instance state

Actions

Launch instances

🔍 Find Instance by attribute or tag (case-sensitive)

All states

< 1 > ⚙

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Bastion Host	i-06410581c48b2aecc	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a
<input type="checkbox"/>	EduBridgeServer	i-041bb4ccc430c16ee	Running	t2.micro	2/2 checks passed	View alarms	us-east-1c

Select an instance

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Monitoring and Management

Set Up CloudWatch Monitoring

1. Access the CloudWatch console from the AWS Management Console.
2. Create a new dashboard for your Easybank application.
3. Add widgets to monitor key metrics:
 - EC2 instance CPU utilization, network traffic, and status checks
 - RDS instance CPU utilization, free storage space, and database connections
 - Application-specific metrics (e.g., number of transactions, active users)
4. Set up CloudWatch Alarms for critical thresholds (e.g., high CPU usage, low free storage).

CPUUtilization: Average



Percent



Configure CloudWatch Logs

1. Install and configure the CloudWatch Logs agent on your EC2 instance.
2. Set up log groups for your application logs, EC2 system logs, and Nginx logs (if applicable).
3. Create log filters to extract and analyze important log events.

Testing, Optimization, and Maintenance

Conduct Thorough Testing

1. Perform functionality testing of all banking features.
2. Conduct security testing, including penetration testing if possible.
3. Perform load testing to ensure the application can handle expected traffic.

Optimize Performance

1. Analyze and optimize database queries.
2. Implement caching mechanisms where appropriate (e.g., Flask-Caching).
3. Optimize front-end assets (minify CSS/JS, optimize images).

Implement Backup and Disaster Recovery

1. Configure automated RDS snapshots for database backups.
2. Set up a disaster recovery plan, including steps for failover to a different region if necessary.

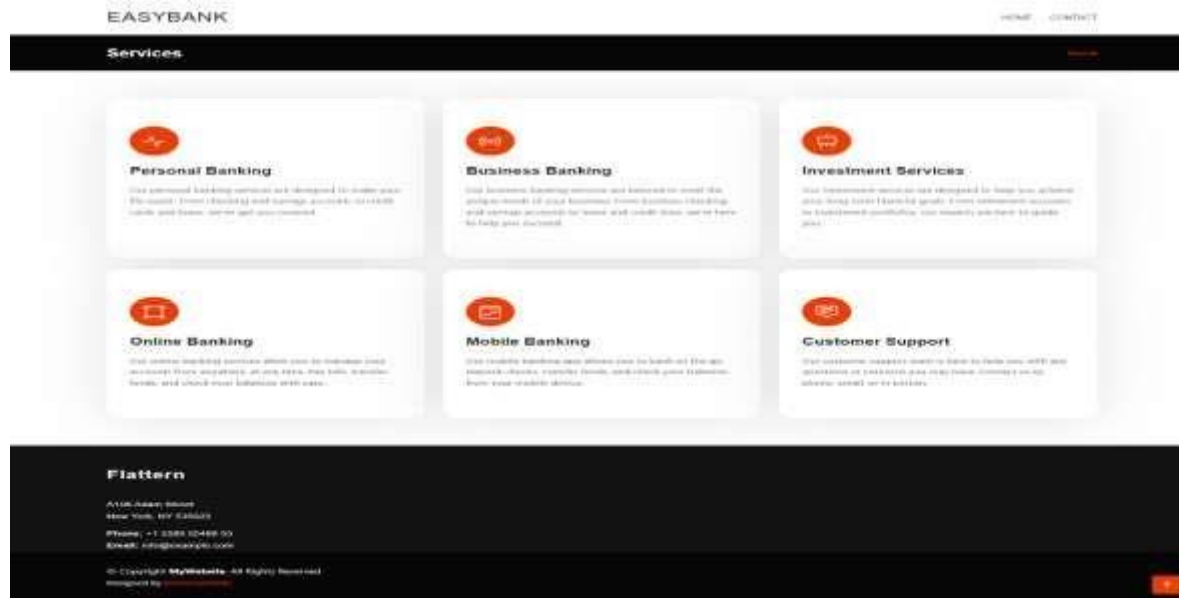
Continuous Improvement

1. Regularly update EC2 instances and RDS with the latest security patches.
2. Monitor CloudWatch metrics and logs to identify areas for improvement.
3. Gather user feedback and iterate on the application features and user experience.
4. Continuously optimize AWS resource usage for cost-effectiveness.

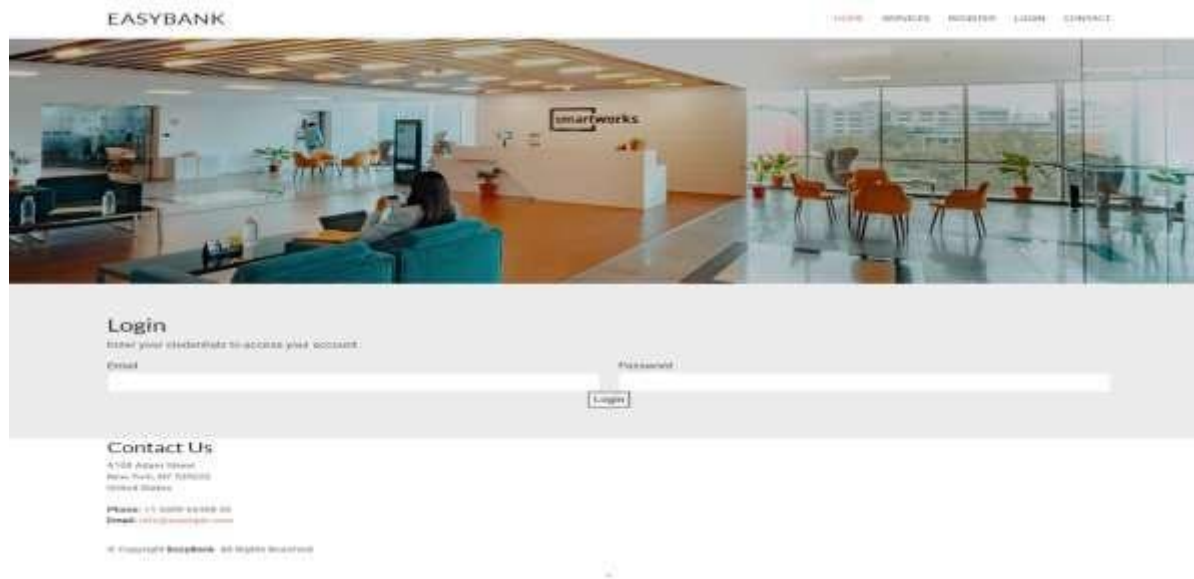
INDEX :



SERVICE :



LOGIN :



DASHBOARD :

Start Automatic Now



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim

Register Here

Financial Performance Analytics



Online Account Opening

Open a new bank account from the comfort of your home. It's quick, easy and secure.



24/7 Account Access

Log into your account anytime, anywhere to manage your finances.



Analyse your spends

Analyse your spends and give your insights on your monthly spend data.

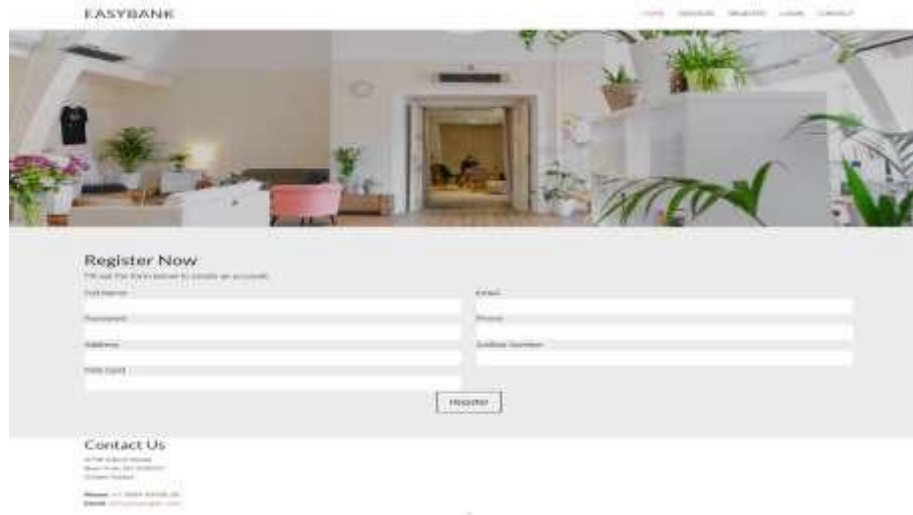


View Statements

Access your account statements online, download or print them as needed.

Read More

REGISTER :



The image shows the 'Register Now' form on the EasyBANK website. The form is set against a background image of a modern, bright interior space with large windows and indoor plants. The form fields are arranged in two columns. The first column contains fields for 'Full Name', 'Email', 'Address', and 'Phone Code'. The second column contains fields for 'Email' and 'Business Number'. A 'REGISTER' button is located at the bottom right of the form. Below the form, there is a 'Contact Us' section with a phone number and a website URL.

EASYBANK

HOME ABOUT US SERVICES CONTACT

Register Now

Fill out the form below to create an account.

Full Name:

Email:

Address:

Business Number:

Phone Code:

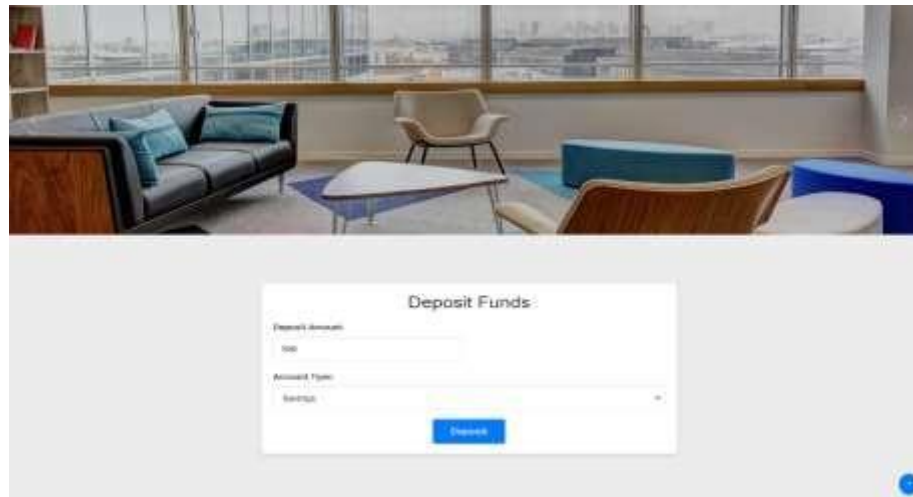
REGISTER

Contact Us

After 9:00 AM - 5:00 PM
Mon-Fri, Sat 9:00 AM - 5:00 PM
Sunday - Closed

Phone: +91 9848 8488 88
Email: info@easybank.com

DEPOSIT :



The image shows the 'Deposit Funds' form on the EasyBANK website. The form is set against a background image of a modern, bright interior space with large windows and indoor plants. The form fields are arranged in a single column. The first field is 'Deposit Amount' with a 'USD' dropdown menu. The second field is 'Amount Type' with a 'Deposit' dropdown menu. A 'Deposit' button is located at the bottom right of the form. A small blue circular icon with a white arrow is located at the bottom right of the page.

Deposit Funds

Deposit Amount:

USD

Amount Type:

Deposit

Deposit

TRANSFER :

Transfer Funds

Securely transfer funds between accounts with ease.

Funds transferred successfully!

Registration successful! Please log in.

Invalid login credentials!

Recipient's User ID:


Transfer Amount:

Transfer

© 2024 Your Bank. All rights reserved.

CONTACT :


Testimonial



POACHABLE

Content of a page

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem



READABLE

Content of a page

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem

Read More

Bank

Contact Us

200 New Thompsonville

Channahon, IL

+1-815-466-0847

contact@bank.com

Invest Money

There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour

Useful Links

There are many variations of passages of Lorem Ipsum available, but the majority have suffered

Quizzes

Enter your email

Submit

CONCLUSION :

The Easybank project represents a significant step forward in the development of cloud-based banking systems. Through the strategic use of AWS services, we have built a platform that is both scalable and secure, capable of evolving alongside the demands of contemporary banking. This project showcases not only our expertise in cloud technologies but also the complexities of building financial applications that prioritize security, efficiency, and compliance.

As the financial industry continues its digital transformation journey, projects like Easybank serve as a blueprint for creating more secure, innovative, and user-friendly banking solutions. The insights gained and the architectural approach developed here offer invaluable guidance for future fintech projects, helping to shape the future of cloud-based applications in regulated industries.

Final Output Page Link :

[BEST BANK](#)

Github Repository Link :

[https://github.com/Saro041018/BankingA
WS.git](https://github.com/Saro041018/BankingAWS.git)