MADRAS INSTITUTE OF TECHNOLOGY,
ANNA UNIVERSITY, CHENNAI - 600 044

DEPARTMENT OF INFORMATION TECHNOLOGY

5/8 B.TECH - ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

NAAN MUDHALVAN PROJECT

TITLE : FRESHBASKET: SCALABLE E-COMMERCE PLATFORM DEPLOYMENT WITH FLASK ON AWS EC2 AND RDS

Submitted by:

RAKSHITHA M – 2022510016

ARUNKUMAR S – 2022510039

MANJUVARSHENI T – 2022510055

DIVYA BHARATHI S – 2022510063

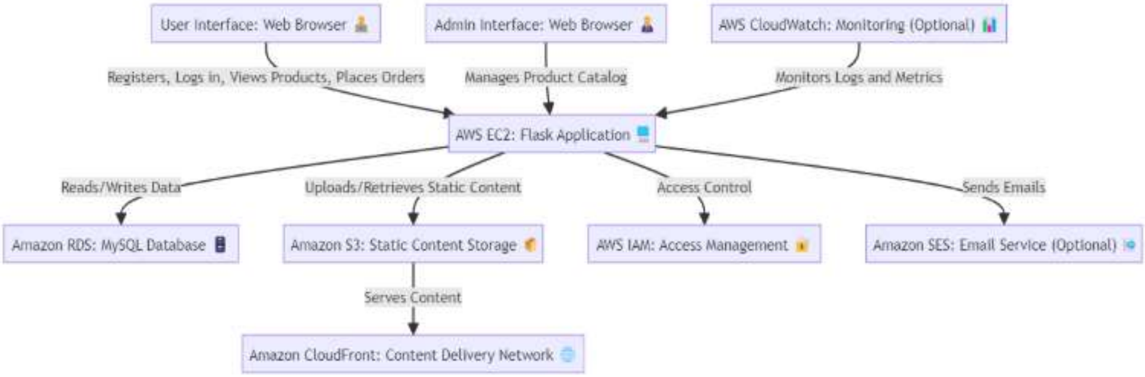
ABARNA B - 2022510303

FRESHBASKET: SCALABLE E-COMMERCE PLATFORM DEPLOYMENT WITH FLASK ON AWS EC2 AND RDS

OBJECTIVE:

The "FreshBasket" project involves developing and deploying a scalable e-commerce platform for selling vegetables and fruits. The platform uses Flask for backend development, AWS EC2 for hosting, and Amazon RDS for database management. This cloud-native solution ensures high availability, scalability, and efficient management of the platform's operations. The project demonstrates how leveraging AWS services can create a robust infrastructure for managing user interactions, product catalogs, and order processing in a seamless manner.

ARCHITECTURE:


User Interface: Web Browser
Admin Interface: Web Browser
AWS CloudWatch: Monitoring (Optional)
Registers, Logs in, Views Products, Places Orders
Manages Product Catalog
Monitors Logs and Metrics
AWS EC2: Flask Application
Reads/Writes Data
Uploads/Retrieves Static Content
Access Control
Sends Emails
Amazon RDS: MySQL Database
Amazon S3: Static Content Storage
AWS IAM: Access Management
Amazon SES: Email Service (Optional)
Serves Content
Amazon CloudFront: Content Delivery Network

PROJECT WORKFLOW:

1. Project Initialization:
    - Define objectives, scope, and KPIs for deploying the FreshBasket e-commerce platform.
    - Set up the AWS environment, including EC2 instance configuration and RDS setup.
    - Outline the use of Flask for backend development and integration of AWS services.
2. EC2 Instance Creation:
    - Launch an EC2 instance to host the FreshBasket application.
    - Select the appropriate instance type based on expected traffic and resource requirements.
3. RDS Configuration:
    - Set up Amazon RDS for database management with MySQL.
    - Configure database instances, including security settings and access controls.
4. Flask Application Deployment:
    - Develop and deploy the FreshBasket application using Flask.
    - Transfer application files to the EC2 instance and configure the environment.
5. Web Server Setup:
    - Install and configure a web server (e.g., Apache, Nginx) on the EC2 instance.
    - Configure the server to properly handle Flask application requests.

6. Testing and Optimization:
  - Test the application for functionality, performance, and security.
  - Optimize server settings, database configurations, and application performance.

7. Monitoring and Maintenance:
  - Implement monitoring tools (e.g., AWS CloudWatch) to track application performance and uptime.
  - Regularly update and maintain the application and server to ensure ongoing reliability and scalability.

KEY FEATURES :

Key Features of the E-Commerce Platform

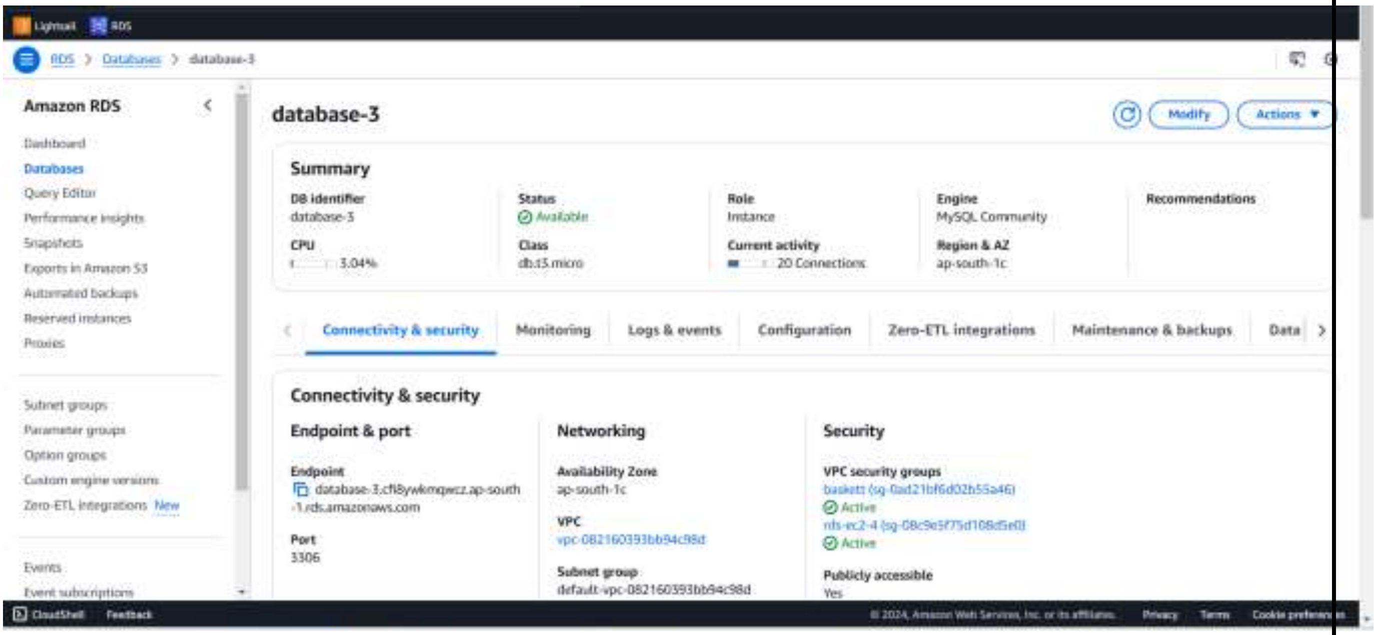
1. **User Management:**
    - **Registration and Login**: Secure user registration and login functionality with password encryption.
    - **Session Management**: Ensures users remain logged in during their session for seamless navigation.
2. **Product Catalog:**
    - **Dynamic Product Listings**: Displays a variety of products with details like name, price, and description.
    - **Categorization**: Allows easy filtering and browsing of products by category.
3. **Cart Management:**
    - **Add to Cart**: Enables users to select and add products to their shopping cart.
    - **Cart Review**: Displays the list of items in the cart with options to update quantities or remove items.
4. **Order Placement:**
    - **Checkout Process**: Guides users through placing orders with a simplified and intuitive interface.
    - **Order Summary**: Provides a detailed summary before confirming the order.
5. **Database Integration:**
    - **Relational Database (MySQL)**: Manages data for users, products, orders, and order details efficiently.
    - **CRUD Operations**: Implements Create, Read, Update, and Delete functionalities for seamless data management.
6. **Responsive Design:**
    - **User-Friendly Interface**: Optimized for various devices, ensuring a consistent experience on desktops, tablets, and mobile phones.
7. **Backend Features:**
    - **Flask Framework**: Powers the backend, providing a robust and scalable application server.
    - **RESTful APIs**: Used for data exchange between the frontend and backend.
8. **Security Measures:**
    - **Input Validation**: Prevents SQL injection and other common vulnerabilities.
    - **Session Security**: Implements secure cookie-based sessions.

9. **Scalability:**
   - **Modular Architecture**: Designed for easy feature expansion in the future.
   - **Cloud Deployment**: Hosted on an AWS EC2 instance for global accessibility.
10. **Deployment and Maintenance:**
    - **AWS Hosting**: Utilizes a t2.micro instance with appropriate security configurations.
    - **Debugging and Logs**: Logs errors and user activity for troubleshooting and analysis.

STEP BY STEP IMPLEMENTATION:

Corrected Step-by-Step Process

1. **Create RDS MySQL Database**:
   - Open AWS RDS Console.
   - Create a new MySQL instance.
   - Configure public accessibility and security group to allow traffic from your IP and EC2 instance (port 3306).
   - Note the RDS endpoint, username, and password.



database-3

Summary

DB identifier: database-3
CPU: 3.04%
Status: Available
Class: db.t3.micro
Role: Instance
Current activity: 20 Connections
Engine: MySQL Community
Region & AZ: ap-south-1c
Recommendations

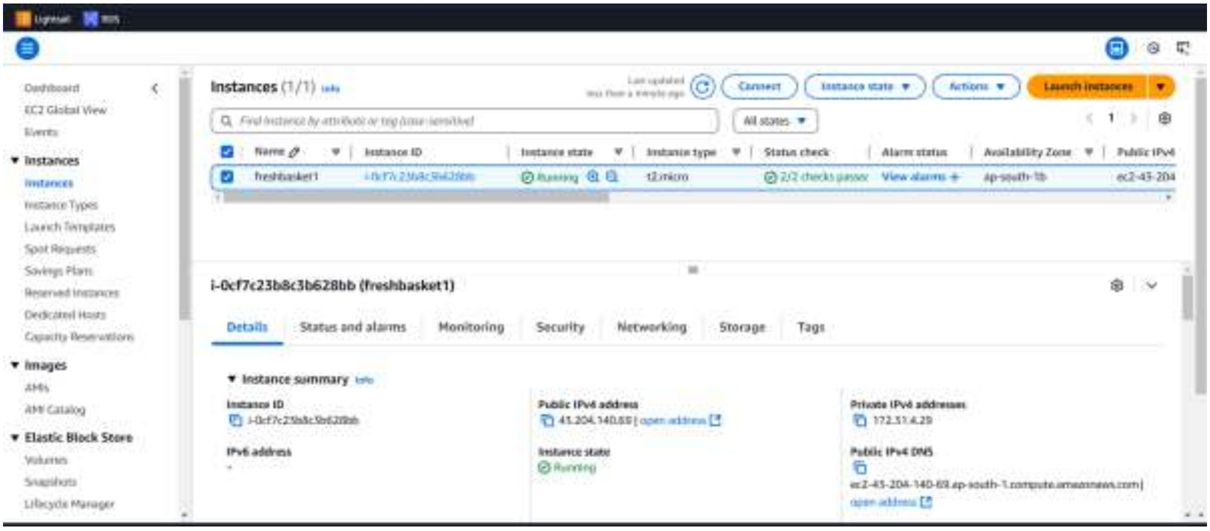
Connectivity & security

Endpoint & port
Endpoint: database-3.cfl8ywkmqwcz.ap-south-1.rds.amazonaws.com
Port: 3306

Networking
Availability Zone: ap-south-1c
VPC: vpc-082160393bb94c98d
Subnet group: default-vpc-082160393bb94c98d

Security
VPC security groups: baskett (sg-0ad21bf6d02b55a46) Active
rds-ec2-4 (sg-08c9e5f75d108d5e0) Active
Publicly accessible: Yes

2. **Launch EC2 Instance**:
   - Open AWS EC2 Console.
   - Launch a new Amazon Linux 2 instance.
   - Configure security group to allow traffic on ports 22 (SSH), 5000 (Flask app), and 3306 (RDS).
   - Download the key pair for later use.



Instances (1/1) Info

Find instance by attribute or tag (case-sensitive)

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☑	freshbasket1	i-0cf7c23b8c3b628bb	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-43-204

i-0cf7c23b8c3b628bb (freshbasket1)

Instance summary Info

Instance ID
i-0cf7c23b8c3b628bb

IPv6 address
-

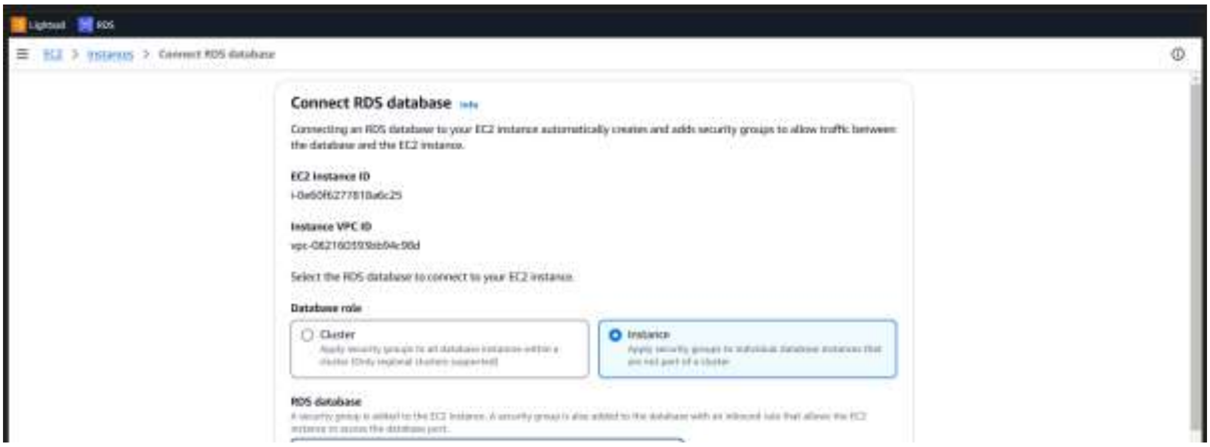
Public IPv4 address
43.204.140.69 | open address

Instance state
Running

Private IPv4 addresses
172.31.4.29

Public IPv4 DNS
ec2-43-204-140-69.ap-south-1.compute.amazonaws.com | open address

3. **Connect EC2 to RDS**:
   - Use the RDS endpoint, username, and password to test the connection between EC2 and RDS using mysql or another client from EC2.



Connect RDS database Info

Connecting an RDS database to your EC2 instance automatically creates and adds security groups to allow traffic between the database and the EC2 instance.

EC2 instance ID
i-0e50f6277810a6c25

Instance VPC ID
vpc-062160593bb04c98d

Select the RDS database to connect to your EC2 instance.

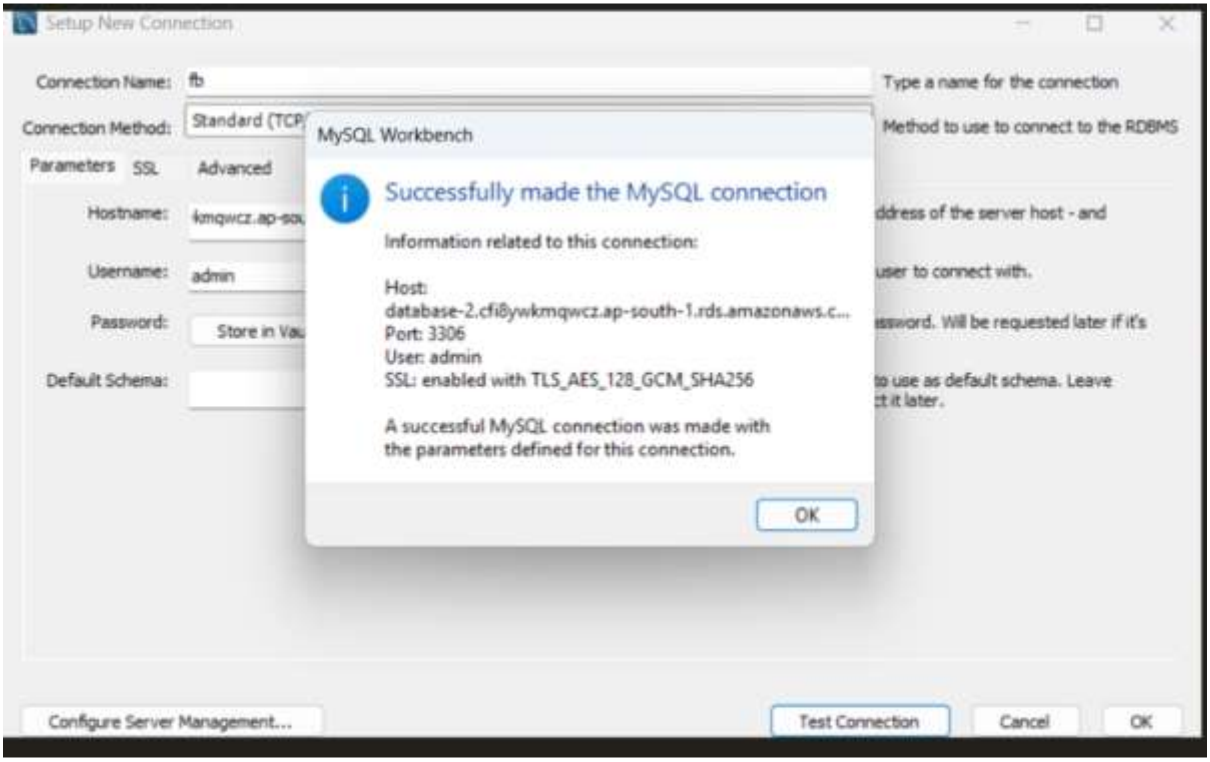
Database role

○ Cluster
Apply security groups to all database instances within a cluster (Only regional clusters supported)

● Instance
Apply security groups to individual database instances that are not part of a cluster

RDS database
A security group is added to the EC2 instance. A security group is also added to the database with an inbound rule that allows the EC2 instance to access the database port.

4. **Set Up MySQL Workbench**:
   - Use MySQL Workbench to connect to the RDS endpoint.
   - Create and configure the required database schema and tables.



Setup New Connection

Connection Name: fb

Type a name for the connection

Connection Method: Standard (TCP/

Method to use to connect to the RDBMS

Hostname: kmqwcz.ap-sou

ddress of the server host - and

Username: admin

user to connect with.

Password: Store in Vau

ssword. Will be requested later if it's

Default Schema:

to use as default schema. Leave
t it later.

MySQL Workbench

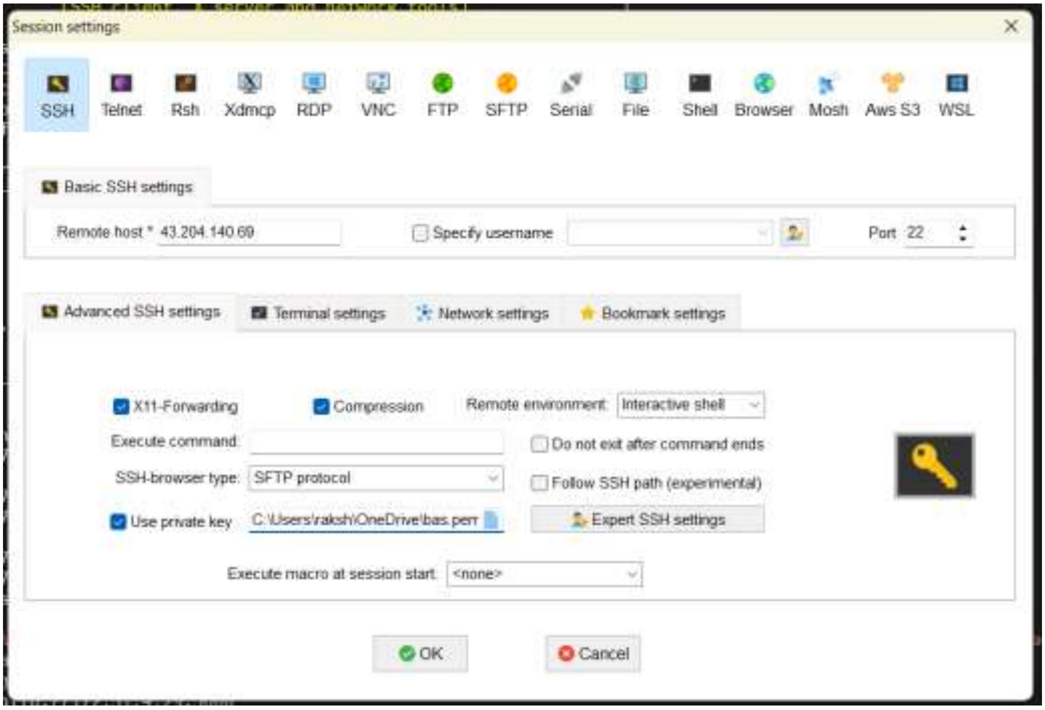
Successfully made the MySQL connection

Information related to this connection:

Host:
database-2.cfi8ywkmqwcz.ap-south-1.rds.amazonaws.c...
Port: 3306
User: admin
SSL: enabled with TLS_AES_128_GCM_SHA256

A successful MySQL connection was made with
the parameters defined for this connection.

5. **Connect to EC2 Using MobaXterm**:
   - SSH into your EC2 instance using the key pair.
6. **Install Required Packages on EC2**:
   - Update packages: sudo yum update -y.
   - Install Python, pip, and Flask:
   - sudo yum install python3 -y
   - pip3 install flask pymysql



Session settings

Basic SSH settings

Remote host * 43.204.140.69 ☐ Specify username Port 22

☑ X11-Forwarding ☑ Compression Remote environment: Interactive shell

Execute command:

☐ Do not exit after command ends

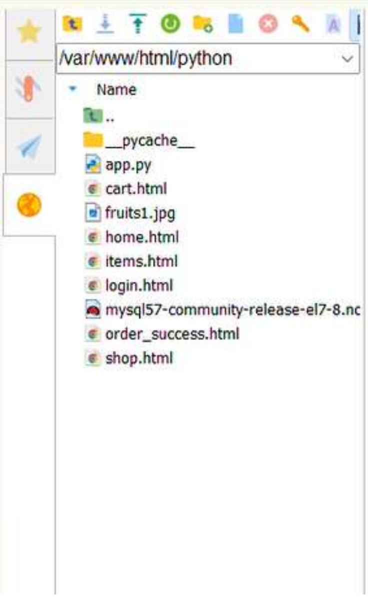
SSH-browser type: SFTP protocol

☐ Follow SSH path (experimental)

☑ Use private key C:\Users\raksh\OneDrive\bas.perr

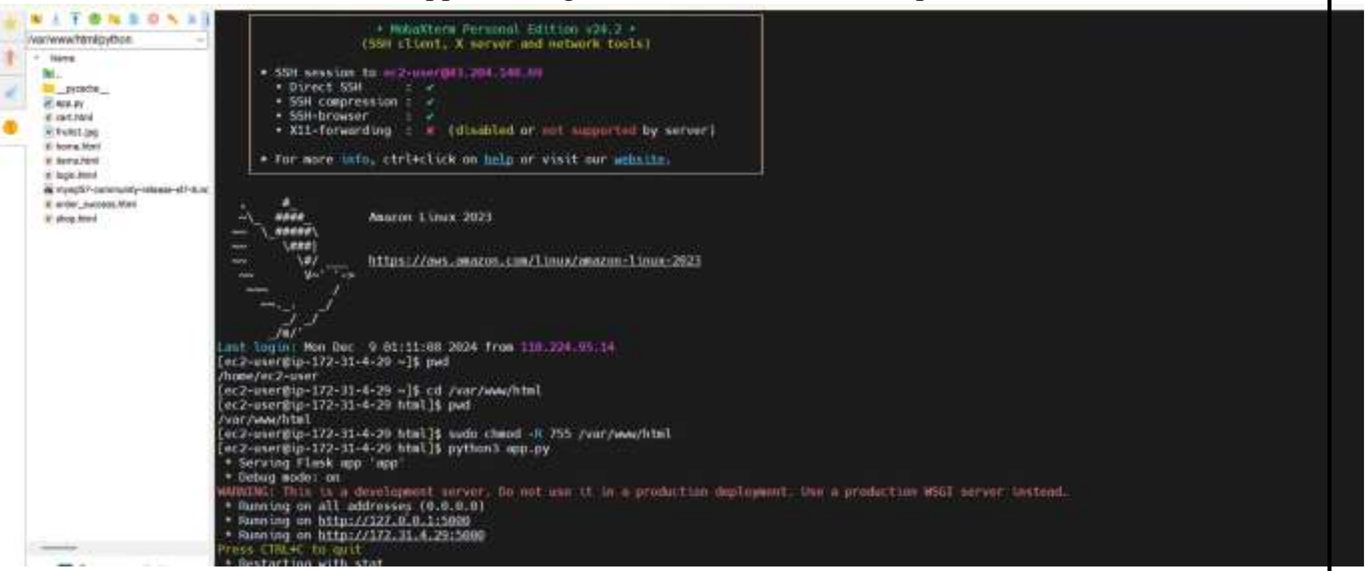
Execute macro at session start: <none>

7. **Transfer Flask Application to EC2**:
   - Use MobaXterm's SFTP feature to upload all HTML files, app.py, and other necessary files to the EC2 /var/www/html directory.



```
/var/www/html/python
Name
..
__pycache__
app.py
cart.html
fruits1.jpg
home.html
items.html
login.html
mysql57-community-release-el7-8.nc
order_success.html
shop.html
```

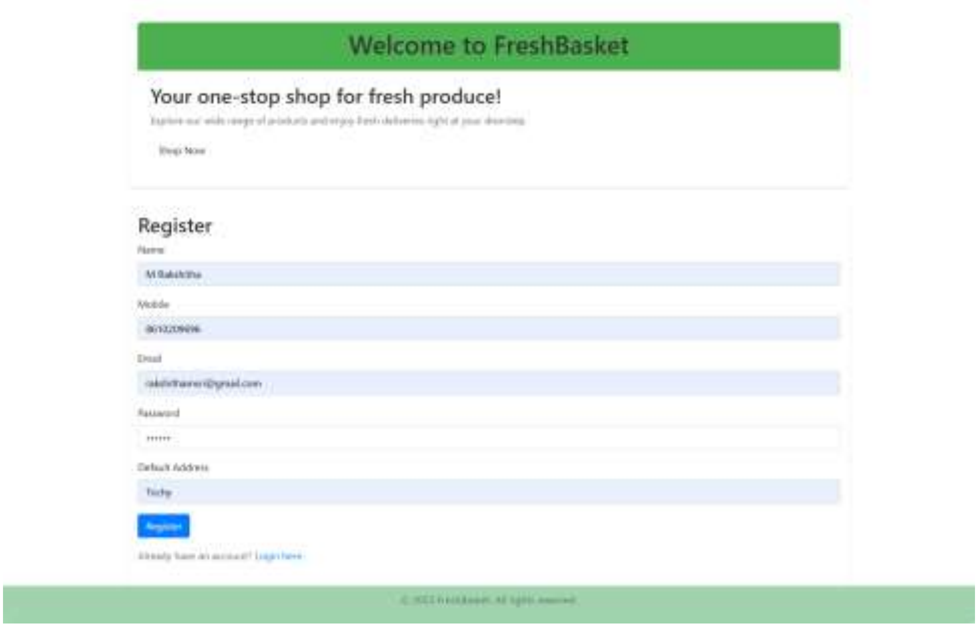
8. **Modify Flask App to Connect to RDS**:
   - Update the database connection details in app.py with the RDS endpoint, username, and password.
9. **Run Flask Application**:
   - Navigate to the application directory.
   - Start the app:
   - python3 app.py
   - Ensure the Flask app is configured to run on 0.0.0.0 and port 5000.



```
            • MobaXterm Personal Edition v24.2 •
         (SSH client, X server and network tools)

 • SSH session to ec2-user@43.204.140.49
   • Direct SSH      :  ✓
   • SSH compression :  ✓
   • SSH-browser     :  ✓
   • X11-forwarding  :  ✗  (disabled or not supported by server)

 • For more info, ctrl+click on help or visit our website.

       #
   ~\_ ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Dec  9 01:11:08 2024 from 110.224.95.14
[ec2-user@ip-172-31-4-29 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-4-29 ~]$ cd /var/www/html
[ec2-user@ip-172-31-4-29 html]$ pwd
/var/www/html
[ec2-user@ip-172-31-4-29 html]$ sudo chmod -R 755 /var/www/html
[ec2-user@ip-172-31-4-29 html]$ python3 app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.31.4.29:5000
Press CTRL+C to quit
 * Restarting with stat
```

10. **E-Commerce Platform Functionalities**:
    - Ensure user registration, login, cart management, and order placement are working as intended.
    - Validate database operations are reflected in RDS.

HOMEPAGE:



Welcome to FreshBasket

Your one-stop shop for fresh produce!

Explore our wide range of products and enjoy fresh deliveries right at your doorstep.

Register

Name: M Rakshitha

Mobile: 8610209696

Email: rakshithamsri@gmail.com

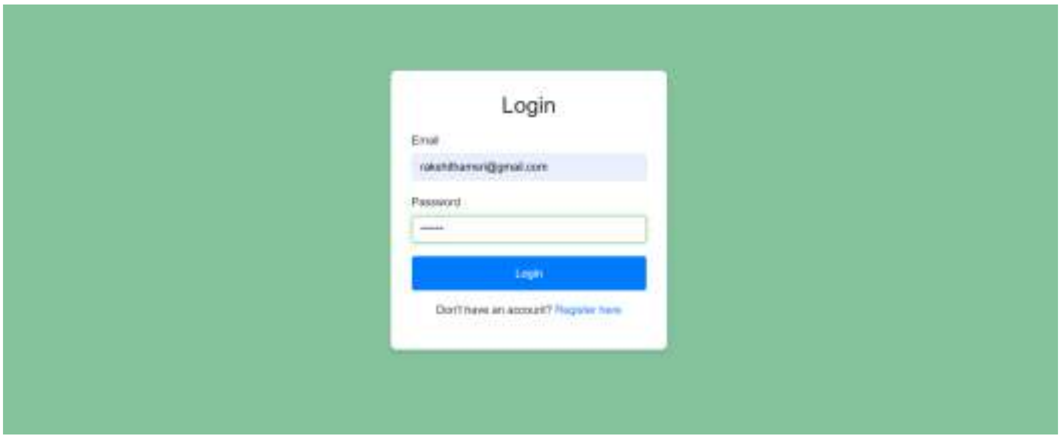
Password: ******

Default Address: Trichy

Already have an account? Login here

© 2023 FreshBasket. All rights reserved.

LOGIN:



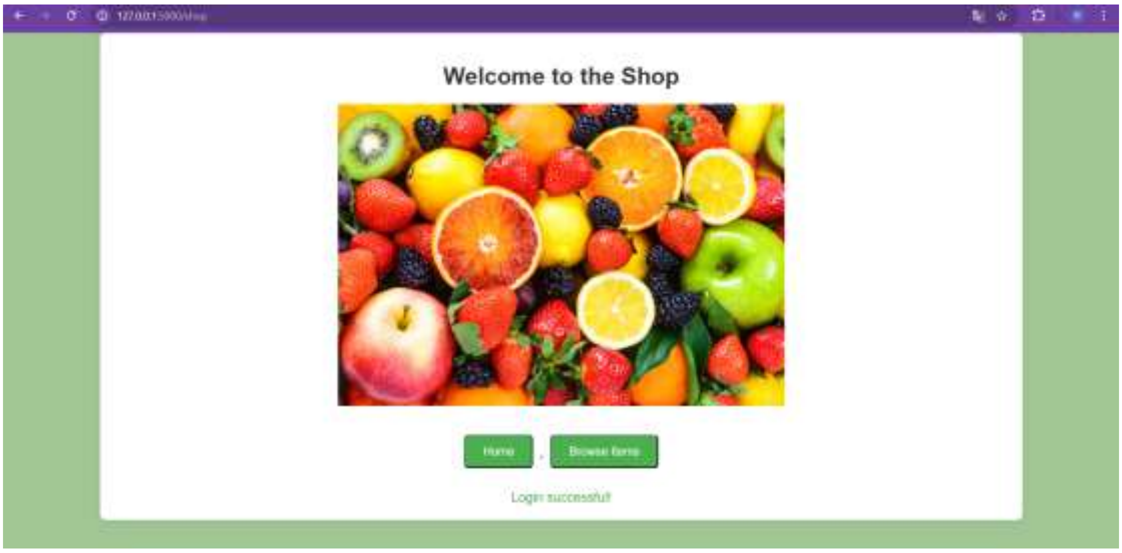
Login

Email: rakshithamsri@gmail.com

Password: ******

Don't have an account? Register here

SHOP:

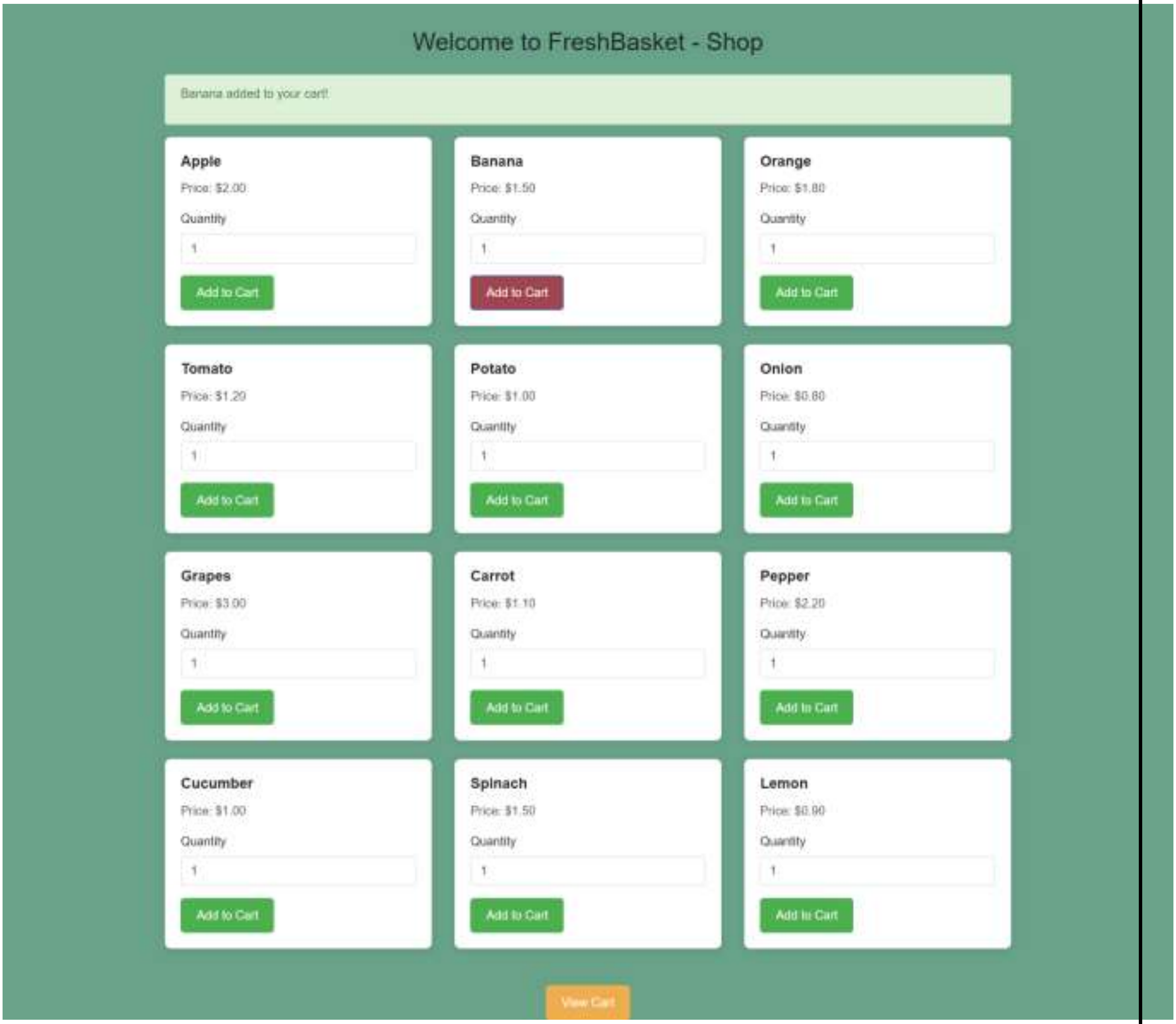


127.0.0.1:5000/shop

Welcome to the Shop

Login successful!

SELECT ITEMS:



Welcome to FreshBasket - Shop

Banana added to your cart!

Apple
Price: $2.00
Quantity
1

Banana
Price: $1.50
Quantity
1

Orange
Price: $1.80
Quantity
1

Tomato
Price: $1.20
Quantity
1

Potato
Price: $1.00
Quantity
1

Onion
Price: $0.80
Quantity
1

Grapes
Price: $3.00
Quantity
1

Carrot
Price: $1.10
Quantity
1

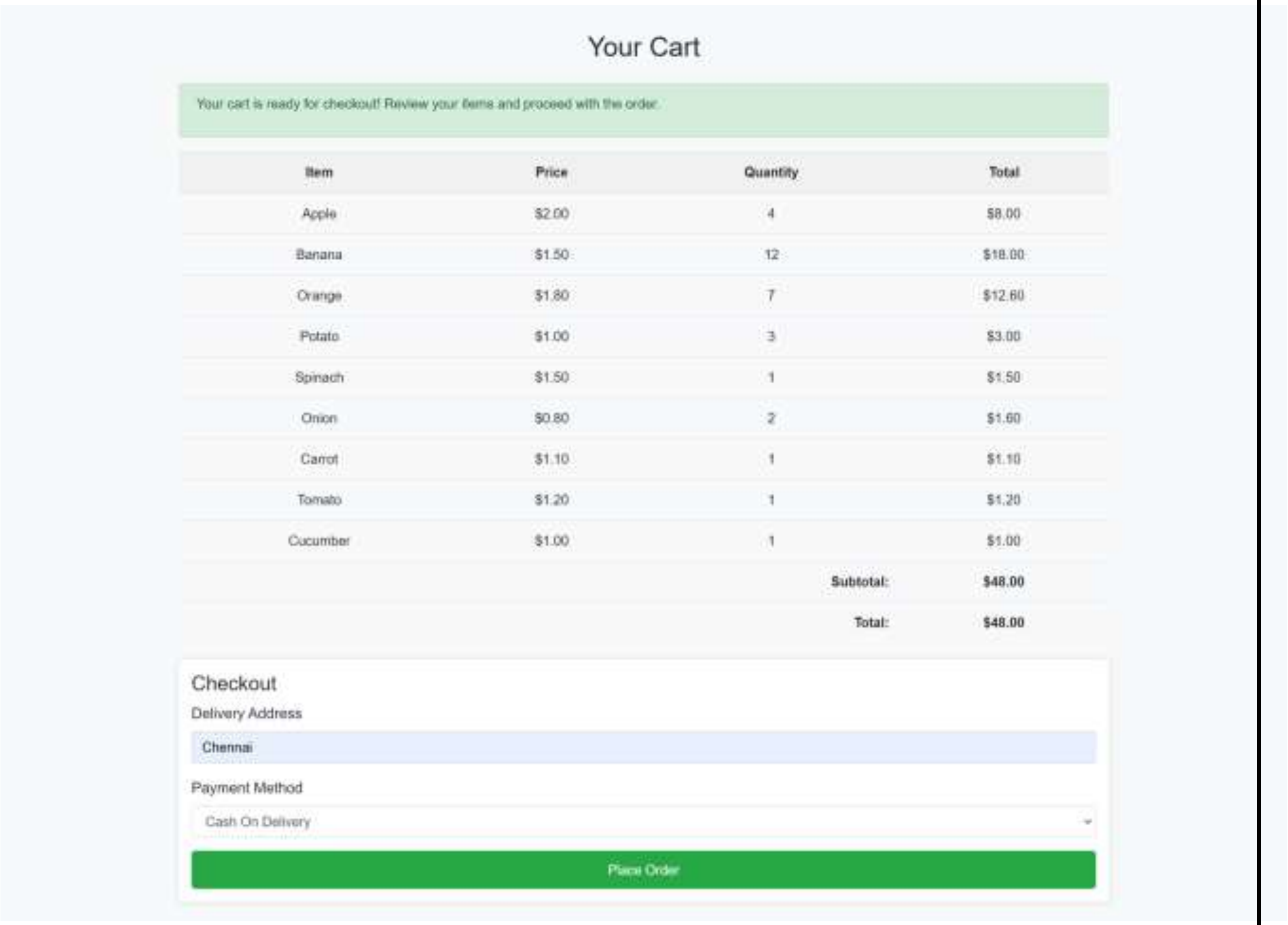
Pepper
Price: $2.20
Quantity
1

Cucumber
Price: $1.00
Quantity
1

Spinach
Price: $1.50
Quantity
1

Lemon
Price: $0.90
Quantity
1

VIEW CART:



Your Cart

Your cart is ready for checkout! Review your items and proceed with the order.

Item	Price	Quantity	Total
Apple	$2.00	4	$8.00
Banana	$1.50	12	$18.00
Orange	$1.80	7	$12.60
Potato	$1.00	3	$3.00
Spinach	$1.50	1	$1.50
Onion	$0.80	2	$1.60
Carrot	$1.10	1	$1.10
Tomato	$1.20	1	$1.20
Cucumber	$1.00	1	$1.00
		Subtotal:	**$48.00**
		Total:	**$48.00**

Checkout

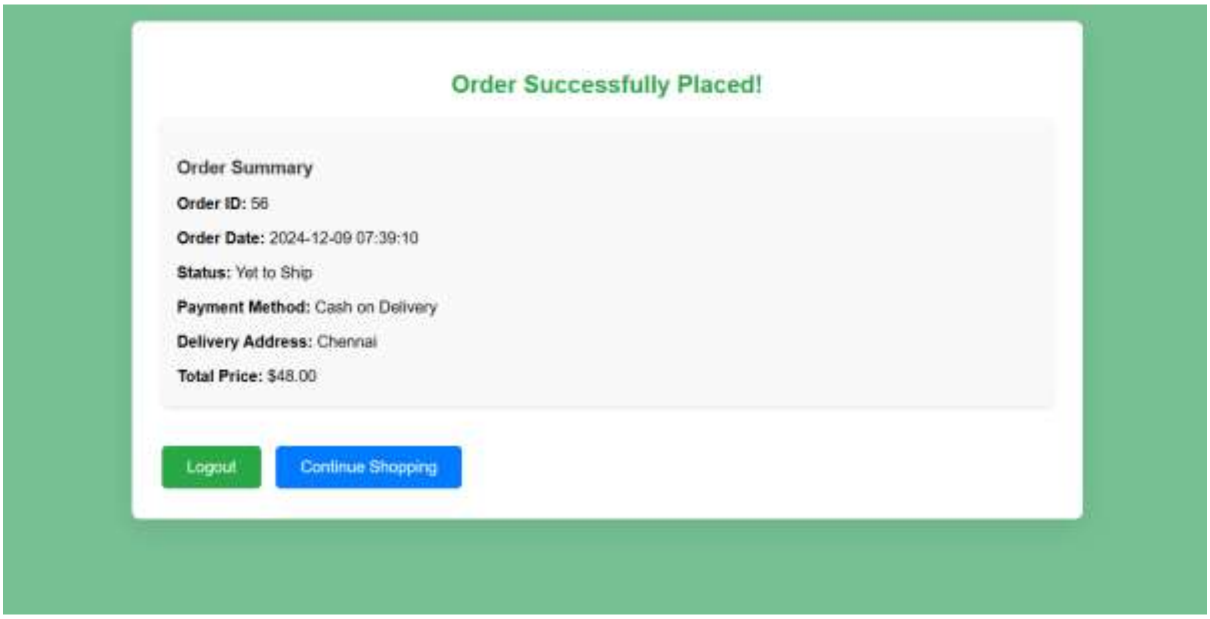
Delivery Address

Chennai

Payment Method

Cash On Delivery

ORDER PLACEMENT:



Order Successfully Placed!

Order Summary

Order ID: 56

Order Date: 2024-12-09 07:39:10

Status: Yet to Ship

Payment Method: Cash on Delivery

Delivery Address: Chennai

Total Price: $48.00

CONCLUSION:

In conclusion, this project successfully demonstrates the development and deployment of a robust e-commerce platform, integrating key functionalities like user management, shopping workflows, and admin order management. Leveraging Flask for the backend, AWS RDS for scalable database management, and EC2 for hosting, the application ensures efficient performance and reliability. Despite initial deployment challenges, the integration of modern tools and technologies highlights the platform's potential for scalability and real-world application, laying a solid foundation for future enhancements and extended functionalities.