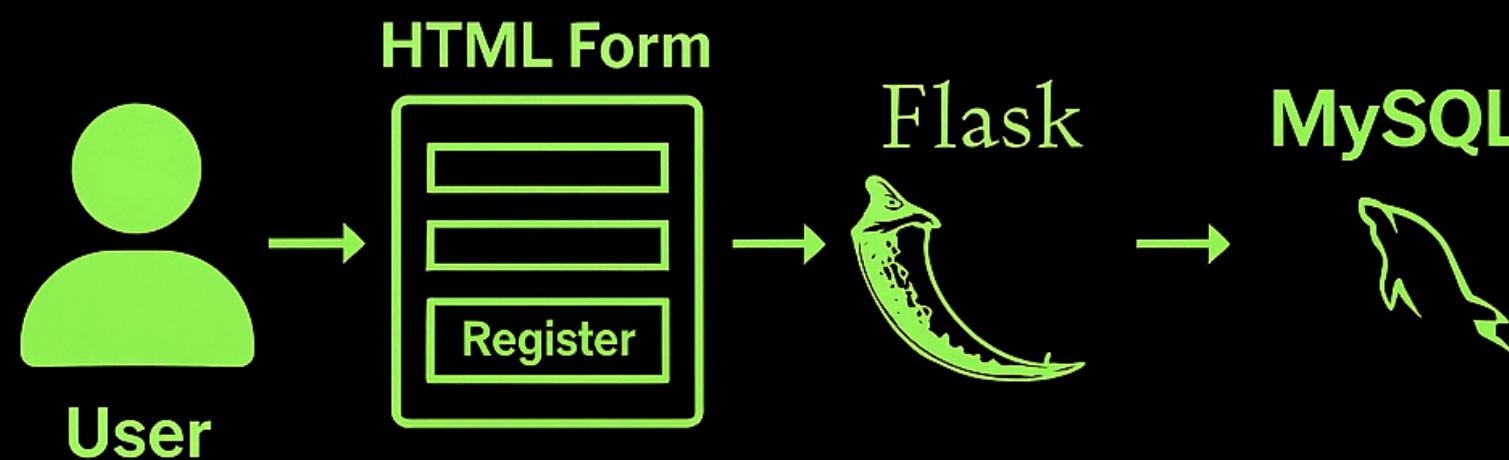


CLOUD-BASED USER MANAGEMENT SYSTEM

USING AWS
EC2 + RDS + PYTHON
FLASK

PROJECT GOAL



OBJECTIVE :

- Develop a user registration and management system
- Host the application on AWS EC2
- Store user data in AWS RDS utilizing MySQL
- Facilitate user registration through an HTML form
- Enable administrators to search for users via a web interface

ABOUT TECHNOLOGY



Flask



aws

Python

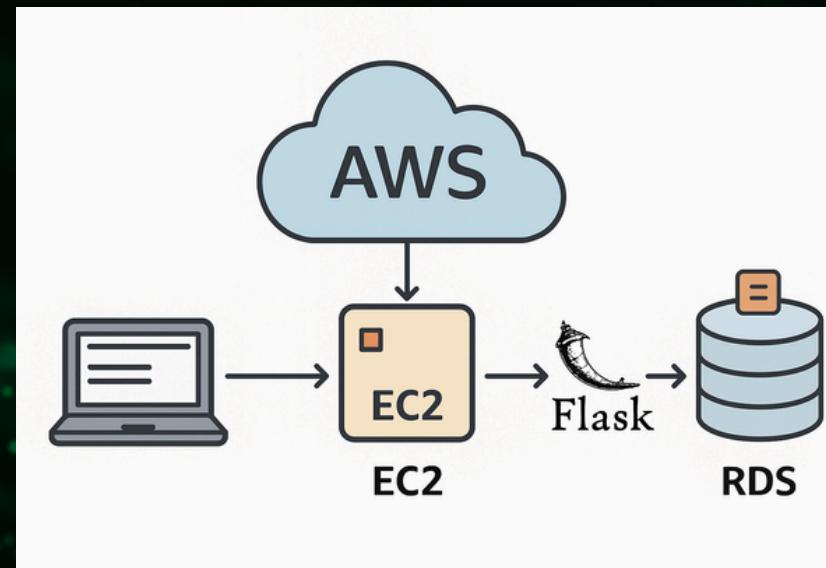
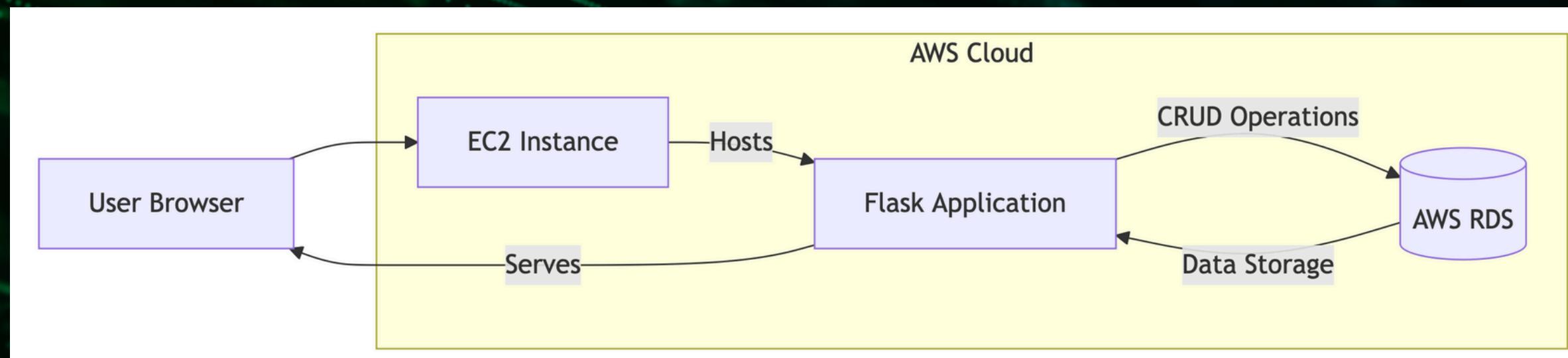


POSTMAN

Tech Stack

- **Frontend:** HTML, CSS
- **Backend:** Python + Flask
- **Database:** AWS RDS (MySQL)
- **Cloud Services:** AWS EC2, RDS
- **Tools:** Postman

ARCHITECTURE OVERVIEW



- User submits form via browser (HTTPS)
- Request routed to Flask app on EC2
- Flask processes logic/validations
- CRUD operations executed on RDS
- RDS returns data to Flask
- Flask renders response to user
- HTML content displayed in browser

PROJECT SETUP STEPS

Project Setup Steps



EC2 Instances (running)

Name	Instance ID	Instance type	Instance state
ip-172-31-90118	i-08fa46488df88a7	t2.micro	running

RDS Databases

DB Identifier	Engine	Region and AZ	DB instance class
userdb	MySQL	us-east-1a	db.t3.micro
userdb	MySQL	us-east-1a	db.t3.micro

Last login: Fri Jul 18 10:11:35 on ttys000
jeetpatel@Jeets-MacBook-Air ~ % chmod 400 ~/Desktop/flask_key.pem
ssh -i ~/Desktop/flask_key.pem ubuntu@3.18.225.240

The authenticity of host '3.18.225.240 (3.18.225.240)' can't be established.
ED25519 key fingerprint is SHA256:Qd20YBdy5KdXX2cs00FOa7Riv5E+wGQdoufi7ZGBUUI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.18.225.240' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Fri Jul 18 17:06:04 UTC 2025

System load: 0.0 Temperature: -273.1 C
Usage of /: 25.3% of 6.71GB Processes: 106
Memory usage: 24% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 172.31.46.231

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-46-231:~\$

Admin Panel - Search Users

```

<h2>Admin Panel - Search Users</h2>
<form method="GET">
    <input name="keyword" placeholder="Search by name, email, or username">
    <input type="submit" value="Search">
</form>

<ul>
    {% for user in users %}
        <li>{{ user[1] }} | {{ user[2] }} | {{ user[3] }}</li>
    {% endfor %}
</ul>

```

User Registration Form

```

<h2>User Registration</h2>
<form method="POST">
    Name: <input name="name"><br>
    Email: <input name="email"><br>
    Username: <input name="username"><br>
    Password: <input name="password" type="password"><br>
    <input type="submit" value="Register">
</form>

```

```

from flask import Flask, render_template, request
import mysql.connector

app = Flask(__name__)

# MySQL RDS connection configuration
db = mysql.connector.connect(
    host="userdb.ch62kqu8cwq7.us-east-2.rds.amazonaws.com",
    user="admin",
    password="Patel2025",
    database="userdb"
)

cursor = db.cursor()

@app.route('/')
def home():
    return "Welcome to the User Management System"

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        # Get form data
        username = request.form['username']
        password = request.form['password']
        first_name = request.form['first_name']
        last_name = request.form['last_name']
        email = request.form['email']

        # Insert into database
        query = "INSERT INTO users (username, password, first_name, last_name, email) VALUES (%s, %s, %s, %s, %s)"
        values = (username, password, first_name, last_name, email)

        cursor.execute(query, values)
        db.commit()

        return 'User registered successfully!'

    return render_template('register.html')

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

```

User_Database_Schema

```

CREATE DATABASE user_db;
USE user_db;

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    username VARCHAR(50),
    password VARCHAR(100)
);

```

Admin
Administrator Access

Admin Credentials:
Username: admin
Password: admin123

Admin Username

Admin Password
 

Admin Login

[Back to Home](#)

Regular user? [User login](#)

Admin Panel - Search Users

```
<h2>Admin Panel - Search Users</h2>
<form method="GET">
  <input name="keyword" placeholder="Search by name, email, or username">
  <input type="submit" value="Search">
</form>
<ul>
  {% for user in users %}
    <li>{{ user[1] }} | {{ user[2] }} | {{ user[3] }}</li>
  {% endfor %}
</ul>
```

ADMIN PANEL

Admin Panel [Dashboard](#) [Users](#) [Home](#) [Logout](#)

[← Back to Dashboard](#)

User Management

Search Users

Search By:

All Users

Showing {{ users|length if users else 0 }} users

[Export Users](#)

User	Email	Username	Role	Status	Actions
{{ user.first_name }} {{ user.last_name }} {{ user.email }}	{{ user.email }}	{{ user.username }}	{{% if user.is_admin %}} Admin {{% else %}} User {{% endif %}}	{{% if user.is_active %}} Active {{% else %}} Inactive {{% endif %}}	Edit Delete

No users found in the database.

USER FUNCTIONALITY

Register
Create your account

Username

Choose a unique username (3-20 characters)

Email

First Name

Last Name

Password

Minimum 8 characters with uppercase, lowercase, and number

Confirm Password

Already have an account? [Login here](#)

User Registration Form ▾

```
<h2>User Registration</h2>
<form method="POST">
  Name: <input name="name"><br>
  Email: <input name="email"><br>
  Username: <input name="username"><br>
  Password: <input name="password" type="password"><br>
  <input type="submit" value="Register">
</form>
```

Route: /register

Fields:

Username, Password,
First Name, Last Name,
Email

Features:
Validate input
Insert into users table
Retrieve/Search/Delete users

BACKEND CODE (FLASK APP.PY)

```
from flask import Flask, render_template, request, jsonify, redirect, url_for
import mysql.connector

app = Flask(__name__)

# MySQL RDS connection config
db_config = {
    'host': 'userdb.ch62kqu8cwq7.us-east-2.rds.amazonaws.com',
    'user': 'admin',
    'password': 'Patel2025',
    'database': 'userdb'
}

def get_db_connection():
    return mysql.connector.connect(**db_config)

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

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

@app.route('/register', methods=['GET'])
def register_page():
    return render_template('simple_register.html')

@app.route('/register', methods=['POST'])
def register_user():
    data = request.form
    conn = get_db_connection()
    cursor = conn.cursor()
    try:
        query = """
            INSERT INTO users (username, password, first_name, last_name, email)
            VALUES (%s, %s, %s, %s, %s)
        """
        values = (
            data['username'],
            data['password'], # Note: hash passwords in production
            data['first_name'],
            data['last_name'],
            data['email']
        )
        cursor.execute(query, values)
        conn.commit()
        return "✅ User registered successfully!"
    except mysql.connector.Error as err:
        return f"❌ Error: {err}"
    finally:
        cursor.close()
        conn.close()
```

The screenshot shows the Postman application interface with three requests:

- POST http://localhost:5000/register**:
 - Body tab selected.
 - Form-data type.
 - Fields:
 - username: john_doe
 - password: password123
 - first_name: John
 - last_name: Doe
 - email: john.doo@example.com
- GET http://localhost:5000/users?<query>**:
 - Body tab selected.
 - Text area contains a JSON object:

```
1 {  
2   "id": 2,  
3   "username": "jane_smith",  
4   "password": "securepass",  
5   "first_name": "Jame",  
6   "last_name": "Smith",  
7   "email": "jane.smith@example.com"  
8 }
```
- DELETE http://localhost:5000/delete/1**:
 - Body tab selected.
 - JSON type.
 - Text area contains:

```
1 {  
2   "message": "User deleted successfully"  
3 }
```

API TESTING WITH POSTMAN

Test POST /register with form data

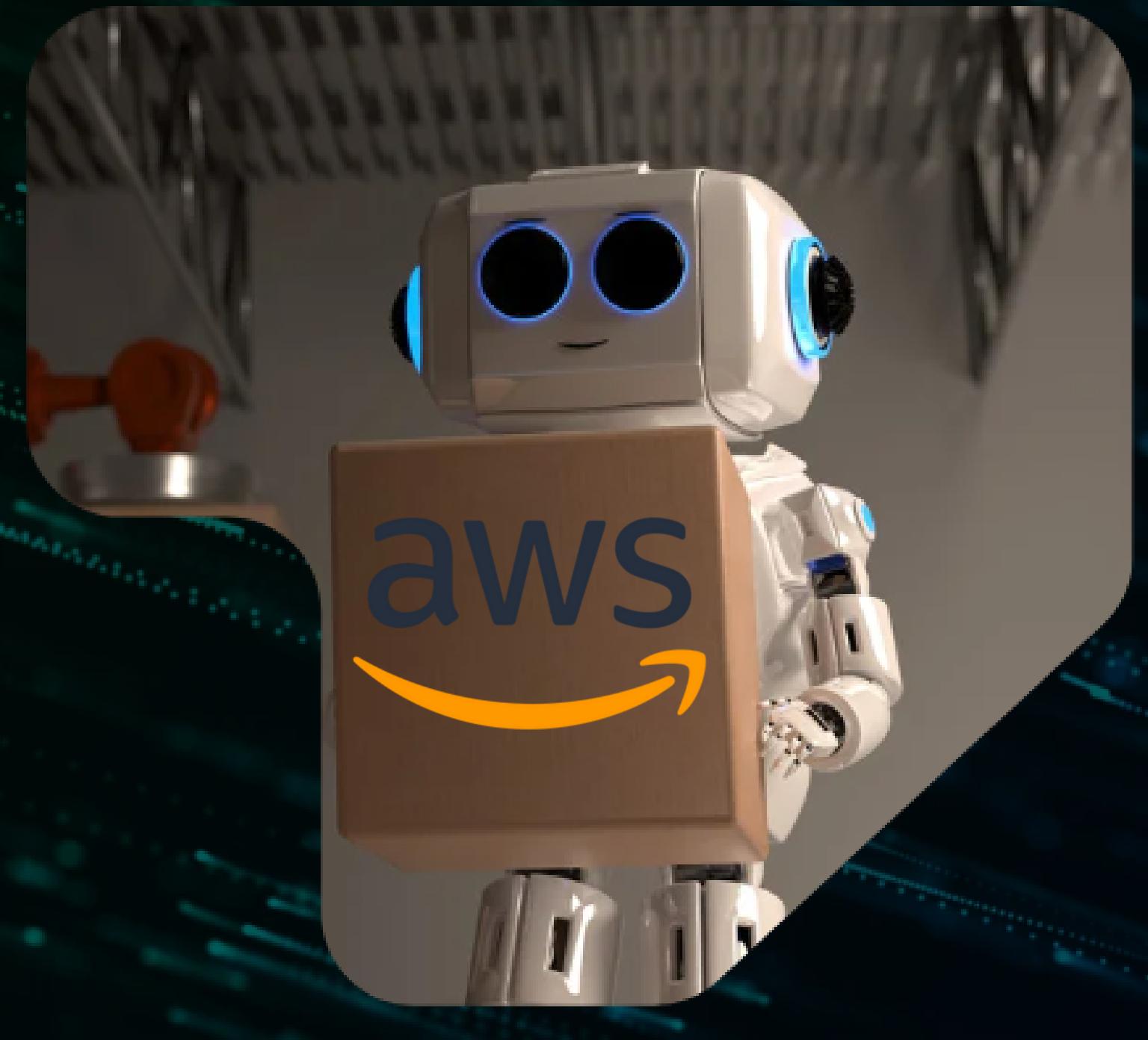
Test GET /search?q=<keyword>

Test DELETE /delete/<user_id>

Correct response codes

DELIVERABLES

- EC2-hosted Flask Web App
- Registration + Admin HTML Pages
- MySQL RDS Database



THANK YOU

...

...