

AWS-hosted Virtual Classroom and Learning Platform

Project Title:

AWS-hosted Virtual Classroom and Learning Platform

Category:

Cloud Deployment | Web Application | AWS Cloud Practitioner

Skills Utilized:

- Core Python
 - Flask Framework
 - AWS EC2, S3, RDS
 - MySQL
 - HTML, CSS, JavaScript
 - Git & GitHub
-

1. Project Overview:

In the modern era of digital learning, building a secure, scalable, and flexible virtual classroom is crucial. This project demonstrates the integration of Flask and multiple AWS services to develop a cloud-native educational platform.

Deployed on AWS EC2, the application leverages Amazon S3 for content storage and Amazon RDS (MySQL) for user and application data. Users can register, log in, and access course materials hosted on the cloud — making the platform both effective and expandable for future needs.

2. Key Features:

- Scalable Infrastructure using EC2
 - Secure File Storage using S3
 - User Management via RDS (MySQL)
 - Responsive Web UI for students and instructors
 - Seamless Cloud Integration with GitHub
-

3. Architecture:

```
project/
├─ app.py
├─ templates/
│   ├─ home.html
│   ├─ register.html
│   ├─ login.html
│   └─ content.html
└─ static/ # If needed for CSS, JS, or images
```

4. Final Project Flow:

4.1 Create an AWS Account:

- Sign up and verify your account.
- Explore the AWS Management Console.

4.2 Create an S3 Bucket and Upload Data:

- Create a bucket (e.g., aws-classroom-content).
- Upload files (PDFs, videos).
- Set proper permissions (public or signed URLs).

4.3 Create an RDS Instance (MySQL):

- Launch RDS with MySQL engine.
- Configure DB instance and create a database.
- Connect using MySQL Workbench to create tables.

4.4 Launch and Configure EC2 Instance:

- Launch instance with Amazon Linux 2 or Ubuntu.
- Set security groups and SSH key pair.
- Install Python, Flask, MySQL client.

4.5 Develop Flask App:

- Build routes for register, login, content.
- Create templates: `home.html`, `register.html`, `login.html`, `content.html`.
- Use Bootstrap for styling.
- Connect app to AWS S3 (using `boto3`) and RDS.

4.6 Deploy Flask App on EC2:

- SSH into EC2.
- Clone GitHub repository.
- Install dependencies: `pip install -r requirements.txt`.
- Run app using Gunicorn + Nginx (optional).

4.7 Upload Code to GitHub:

- Create repository.
- Push project files with commits and documentation.

4.8 Test Scenarios:

- **Scenario 1: Student Registration and Login**
 - **Scenario 2: Admin Upload of Course Materials**
 - **Scenario 3: Downloading Course Content**
-

5. User Scenarios:

Scenario 1: Student Registration and Course Access

- **User:** Alice Johnson
- **Process:** Registers via form, logs in, and accesses course content from S3.

Scenario 2: Admin Uploads Course Materials

- **User:** System Admin
- **Process:** Uploads PDFs; content is stored in S3 and metadata is updated in RDS.

Scenario 3: Student Downloading Course Content

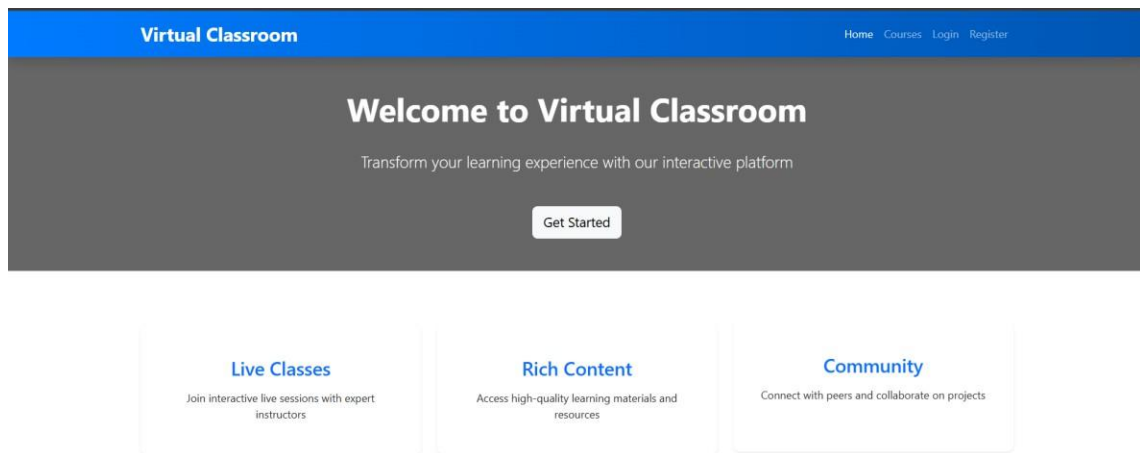
- **User:** Bob Patel
 - **Process:** Selects a file, clicks a link, and downloads directly from S3.
-

6. Challenges Faced:

- Learning AWS services and IAM policies
 - Managing AWS credentials securely
 - Flask and AWS integration (using `boto3`)
 - RDS connection issues and MySQL Workbench setup
 - Debugging EC2 deployment issues
-

7. Output Pages:

- `home.html` → Landing page with navigation



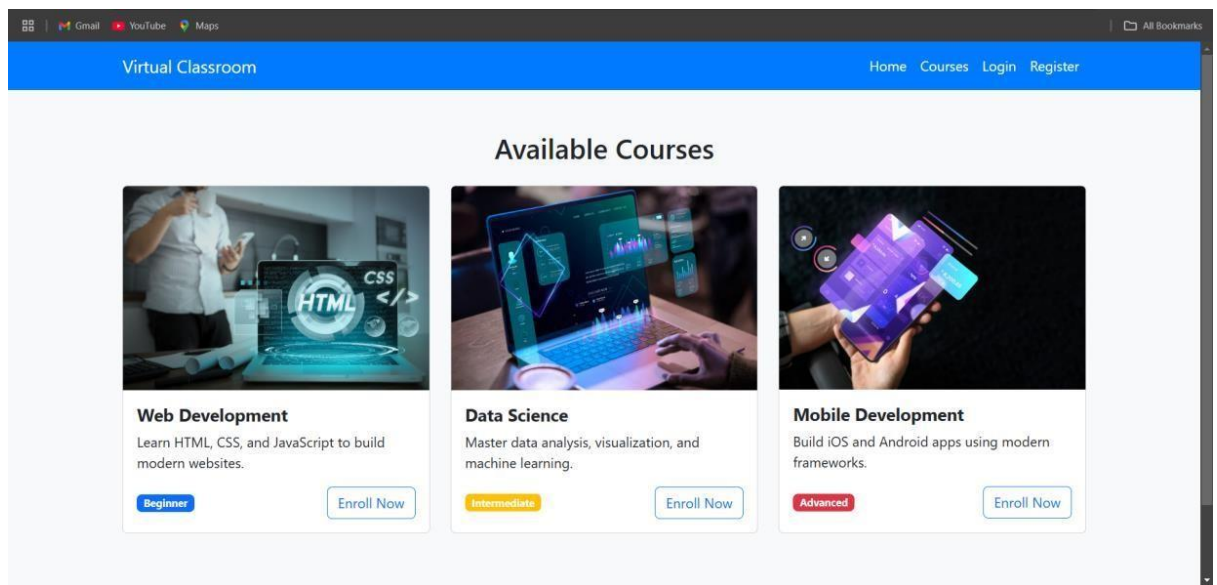
- `register.html` → Registration form

The screenshot shows the registration form titled 'Create Your Account'. It includes input fields for 'First Name', 'Last Name', 'Email address', 'Password', and 'Confirm Password'. There is a checkbox for 'I agree to the Terms and Conditions' and a 'Register' button. A link for 'Already have an account? Login here' is located at the bottom of the form.

- `login.html` → Login page

The screenshot shows the login page titled 'Login to Your Account'. It includes input fields for 'Email address' and 'Password', a 'Remember me' checkbox, and a 'Login' button. A link for 'Don't have an account? Register here' is located at the bottom of the form. The footer of the page contains the copyright notice: '© 2024 Virtual Classroom. All rights reserved.'

- `content.html` → Course materials page with download links from S3



8. Conclusion:

This project highlights the integration of cloud computing with web development to build a fully operational virtual classroom. Leveraging AWS's scalability and Flask's simplicity, the platform achieves reliable user access, secure data handling, and an overall smooth educational experience.

9. References :

1. AWS Account Setup: https://youtu.be/CjKhQoYeR4Q?si=ui8Bvk_M4FfVM-Dh
2. Web Application Stack: [FLask](#) || [MySQL Connector using flask](#) || [HTML/JS/CSS](#)
3. AWS EC2 Instance: https://www.youtube.com/results?search_query=aws+ec2+oneshot
4. RDS Database: https://www.youtube.com/results?search_query=rds+oneshot
5. MySQL: https://www.youtube.com/results?search_query=mysql+tutorial
6. RDS connects MySQL:
https://www.youtube.com/results?search_query=mysql+connector+for+rds
7. Clone Git repo: https://www.youtube.com/results?search_query=clone+github+repository
8. AWS Cost Management: <https://youtu.be/OKYJCHHSWb4?si=aY3DQl1v26CfZxXA>

10. Source code and Video:

1. Video Link: [<https://drive.google.com/file/d/1GTTz1Vi2LnGkOIUC-6TDKz5gSlizMF1n/view?usp=sharing>]