

AWS-HOSTED VIRTUAL CLASSROOM AND LEARNING PLATFORM

A PROJECT REPORT

Submitted by

SHAMEETHA RAVIKUMAR

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE CHENNAI CITY CAMPUS

ANNA UNIVERSITY : CHENNAI 600 025

NOVEMBER 2024

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**Aws-Hosted Virtual Classroom and Learning Platform**” is the bonafide work of **SHAMEETHA RAVIKUMAR** who carried out the project work under my supervision.

SIGNATURE
PROJECT COORDINATOR

SIGNATURE
NM SPOC

INTERNAL SIGNATURE

EXTERNAL SIGNATURE

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGENO.
	ABSTRACT	4
	LIST OF TABLE	
	LIST OF FIGURES	
	LIST OF SYMBOLS	
1.	INTRODUCTION	6
	1.1 BACKGROUND	6
	1.2 OBJECTIVE	6
	1.4 SCOPE	7
2.	LITERATURE REVIEW	8-9
	2.1 EVOLUTION	
	2.2 CLOUD COMPUTING IN EDUCATION	
	2.3 ROLE OF AWS IN EDUCATION	
	2.4 SECURITY AND PRIVACY	
	2.5 COMPARATIVE ANALYSIS OF FRAMEWORKS	
	2.6 TRENDS AND CHALLENGES	
	2.7 SUMMARY	
3.	SYSTEM ARCHITECTURE AND TECHNOLOGIES USED	10-12
4.	PROJECT FLOW AND FUNCTIONALITIES	13-14
5.	IMPLEMENTATION DETAILS	15-19
6.	TESTING AND OPTIMIZATION	20-21
7.	CONCLUSION	24-25
8.	REFERENCES	26

ABSTRACT

Aws-Hosted Virtual Classroom and Learning Platform

The AWS-hosted Virtual Classroom and Learning Platform project aims to harness the power of cloud technologies to provide a comprehensive, scalable, and secure virtual learning environment. In today's rapidly evolving digital education landscape, creating a reliable and accessible platform is essential to meet the growing demand for remote learning. This project is designed to address this need by building a cloud-native educational platform that integrates several core AWS services to deliver a seamless and interactive learning experience. The platform's backend is developed using Flask, a lightweight and flexible web framework known for its simplicity and adaptability. Hosting this backend on an AWS EC2 instance ensures a robust, customizable, and scalable deployment that can accommodate varying workloads, allowing the platform to scale in response to user demand. Course materials, media files, and other educational content are stored in Amazon S3, leveraging its high availability and durability to provide secure and efficient access to resources. For data management, Amazon RDS (MySQL) serves as the primary database, storing critical information such as user registration and login data, which are essential for user account management and platform functionality. The RDS database is configured to optimize performance and ensure data integrity, enhancing the reliability of the user experience. Together, these components create a cohesive virtual learning environment that is both resilient and adaptable. By integrating Amazon's cloud infrastructure, the project not only facilitates the efficient management of resources but also ensures a high level of security, data accessibility, and ease of scalability. This virtual classroom setup exemplifies the potential of cloud-based solutions in supporting modern educational initiatives and represents a significant step forward in the delivery of online education.

I) List of Tables

- System Architecture Components
- Functional Module Descriptions
- AWS EC2 Configuration Settings: Security Group Setup
- AWS EC2 Configuration Settings: Elastic IP Assignment
- AWS RDS Configuration Details
- Summary of Testing Scenarios and Outcomes

II) List of Figures

- System Architecture Diagram
- AWS EC2 Configuration
- AWS RDS Configuration
- AWS S3 Configuration
- MYSQL Database Design
- Output Screen

III) List of Symbols, Abbreviations, and Nomenclature

- **AWS:** Amazon Web Services
- **EC2:** Elastic Compute Cloud
- **RDS:** Relational Database Service
- **HTTP:** Hypertext Transfer Protocol
- **HTTPS:** Secure Hypertext Transfer Protocol
- **S3:** Amazon Simple Storage Service

Chapter 1: Introduction

1.1 Background

The rapid evolution of technology has transformed various aspects of human life, with education being one of the most impacted domains. Traditional learning methods have increasingly been supplemented or replaced by digital solutions, enabling accessibility and efficiency in the dissemination of knowledge. This transition has been further accelerated by the adoption of cloud computing technologies, which offer robust infrastructure and scalable resources for the creation of virtual learning environments.

In response to the growing demand for flexible and secure online education systems, the **AWS-Hosted Virtual Classroom and Learning Platform** was developed. By leveraging Amazon Web Services (AWS) technologies, this platform integrates cloud-hosted functionalities with a user-centric interface to offer a seamless learning experience.

1.2 Objective

The primary objective of the project is to design and implement a cloud-based Virtual Classroom that delivers a centralized and scalable platform for students, educators, and administrators. The system is built to:

- Provide a secure environment for user registration and login.
- Enable efficient storage and retrieval of course materials using AWS S3.
- Support seamless application hosting and real-time interactions with Amazon EC2.
- Manage user and course-related data securely with AWS RDS (MySQL).

This platform aims to empower learners with self-paced educational resources, improve educator efficiency, and streamline administrative processes through automation and cloud integration.

1.3 Scope

The scope of the AWS-Hosted Virtual Classroom and Learning Platform encompasses the following key features and functionalities:

1. **User Management:** Facilitates user registration, login authentication, and profile management.
2. **Course Content Management:** Stores and delivers multimedia course materials, including documents and videos, leveraging AWS S3 for scalability.
3. **Interactive Web Application:** Utilizes Flask as the backend framework to provide dynamic features, including dashboards and user-specific functionalities.
4. **Scalable Cloud Infrastructure:** Deploys applications on AWS EC2, ensuring high availability and adaptability to varying user demands.
5. **Data Security and Management:** Implements secure data storage and access control mechanisms using AWS RDS, focusing on user privacy and data integrity.
6. **Future Expandability:** Designed to accommodate future enhancements such as real-time communication tools, analytics dashboards, and integration with external learning systems.

Chapter 2: Literature Review

2.1 Evolution of Virtual Learning Platforms

The journey from traditional classroom learning to virtual education has been marked by significant technological advancements. Early virtual learning platforms relied on simple file-sharing systems and asynchronous communication tools. With the rise of high-speed internet and cloud computing, modern platforms now support real-time collaboration, multimedia content delivery, and adaptive learning methodologies. This evolution underscores the importance of integrating scalable and secure technologies, such as AWS, to address the growing demand for virtual education.

2.2 Cloud Computing in Education

Cloud computing has revolutionized the education sector by providing on-demand access to computing resources. Educational institutions benefit from reduced IT infrastructure costs, enhanced data security, and scalable solutions tailored to meet diverse learning needs. Studies indicate that cloud-based solutions, like AWS, offer the flexibility required for managing dynamic workloads in virtual classrooms, such as high-volume user access and multimedia content delivery.

2.3 Role of AWS in Building Educational Platforms

Amazon Web Services (AWS) has become a preferred choice for developing educational platforms due to its comprehensive range of cloud services. Key features such as Amazon EC2, Amazon S3, and Amazon RDS make it possible to build robust, scalable, and secure applications. Literature highlights successful implementations of AWS in supporting real-time learning management systems, demonstrating its capability to handle large user bases and deliver seamless performance.

2.4 Security and Privacy in Cloud-Based Education Systems

Security is a critical aspect of virtual learning platforms. Issues such as unauthorized access, data breaches, and identity theft pose risks to students and educators. AWS addresses these challenges with features like identity and access management (IAM), encryption, and automated compliance checks. Research has shown that integrating these security measures enhances trust and reliability in cloud-based educational environments.

2.5 Comparative Analysis of Frameworks for Web Applications

Flask, Django, and Node.js are popular frameworks used for developing web applications. Studies comparing these frameworks emphasize Flask's lightweight architecture and ease of integration with cloud services like AWS. This makes it a suitable choice for projects requiring flexibility and quick deployment, such as the AWS-Hosted Virtual Classroom.

2.6 Trends and Challenges in Virtual Education

Recent trends in virtual education highlight the growing adoption of artificial intelligence, personalized learning experiences, and immersive technologies such as AR/VR. However, challenges like limited internet access in rural areas, data security concerns, and resistance to adopting new technologies remain prevalent. Addressing these challenges through innovative cloud-based solutions, such as the one proposed in this project, can help bridge these gaps.

2.7 Summary of Existing Research

The existing literature underscores the significance of cloud computing in transforming virtual education. While AWS offers a robust infrastructure, combining it with modern frameworks like Flask ensures an optimized learning experience. This project builds upon these findings, focusing on delivering a practical implementation that aligns with contemporary educational needs.

Chapter 3: System Architecture and Technologies Used

3.1 Overview of System Architecture

This section provides a high-level overview of the architecture used for the AWS-Hosted Virtual Classroom and Learning Platform. It outlines the interaction between key components, including the web application, database, cloud storage, and end-user devices. The architecture is designed to ensure scalability, reliability, and security for seamless user experiences.

3.2 Components of System Architecture

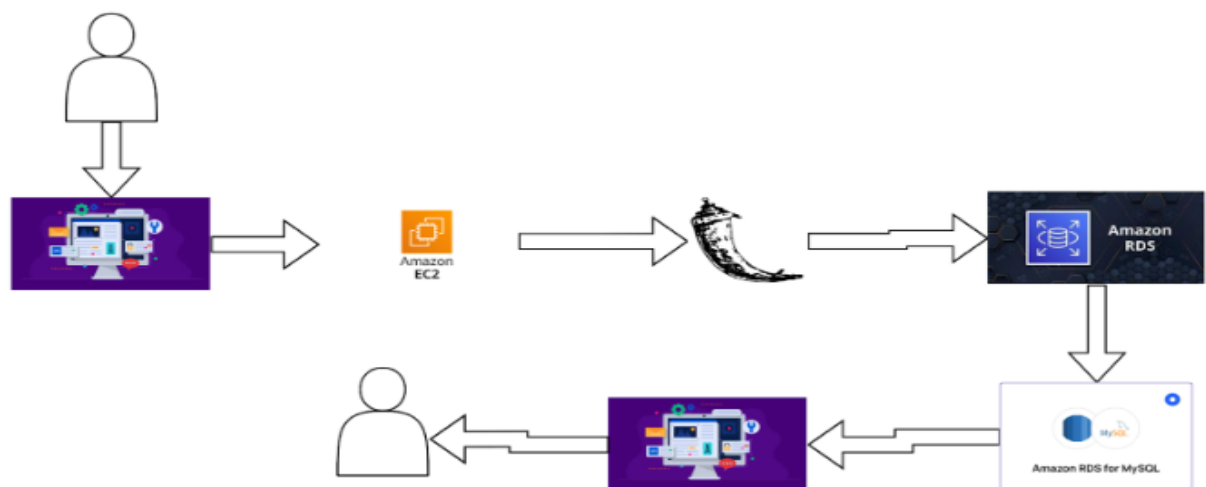
- **Frontend:** The web application interface is designed using HTML, CSS, and JavaScript, ensuring a user-friendly and responsive design.
- **Backend:** The backend is powered by Flask, a lightweight Python framework, which handles routing, API integration, and user authentication.
- **Database:** Amazon RDS with MySQL is used to manage user data, course content metadata, and other structured information.
- **Storage:** Amazon S3 provides scalable and secure storage for course materials, including videos, documents, and other multimedia content.

Component	Function
User Interface (UI)	Provides user registration, login, and access to course materials.
Web Server (Flask)	Processes user requests, connects to the database, serves dynamic content, and manages sessions.
Database (RDS)	Stores and retrieves user information (e.g., credentials), course details, and metadata.
Cloud Storage (S3)	Stores course materials securely, providing easy access for students to download and view course content.

Authentication	Verifies login credentials, manages sessions, and ensures user security.
EC2 Instance	Hosts the Flask web server and handles requests from users, providing the backend infrastructure for the platform.

3.3 Role of AWS Services

- Amazon EC2: Used for hosting the web application, providing flexible compute capacity.
- Amazon S3: Handles static content storage with high durability and accessibility.
- Amazon RDS: Manages the relational database for user authentication, course registrations, and progress tracking.



3.4 Flask Framework for Backend Development

Flask was chosen for its simplicity and modular design, allowing efficient integration with cloud services. It supports rapid prototyping and offers robust libraries for handling RESTful APIs, which facilitate smooth interaction with AWS resources.

3.5 Database Design and Management

Amazon RDS with MySQL is used to design and manage the database schema, ensuring the efficient storage and retrieval of data. The database is structured to accommodate user profiles, course details, enrollment records, and progress tracking.

Chapter 4: Project Flow and Functionalities

4.1 Workflow Description

- **User Registration and Authentication:**

Students register on the platform by providing their details, which are securely stored in Amazon RDS after password hashing. Upon successful registration, users can log in and access their dashboard.

- **Course Material Management:**

Administrators upload course materials, such as lecture notes and PDFs, through an admin panel. These files are stored in Amazon S3, and their metadata is updated in RDS for real-time access by students.

- **Content Access and Download:**

Students browse available course materials on their dashboard. The backend fetches file URLs from Amazon S3 and displays them. Students can download materials using secure pre-signed URLs generated by Flask.

4.2 System Integration

The platform leverages the following components:

- **Flask:** Manages backend operations and user interactions.
- **Amazon RDS:** Stores user and course metadata securely.
- **Amazon S3:** Provides scalable storage for course content.

4.3 Summary

The workflow ensures a seamless integration of frontend and backend operations, leveraging AWS for scalability, security, and efficiency. Students and administrators benefit from a user-friendly interface and reliable access to course materials.

4.4 Functional Module Descriptions

Module	Description
User Registration	Allows students to register on the platform by providing personal details.
User Login	Allows registered users to log in using their credentials.
Course Content Access	Students can access and download course materials from the platform's dashboard.
File Upload & Download	Users can upload & download course content securely from the platform.

Chapter 5: Implementation Details

5.1 AWS EC2 Configuration

1. **EC2 Instance Launch:** Created through the AWS console with the following specifications:

- **AMI (Amazon Machine Image):** Amazon Linux 2, which is optimized for cloud applications.
- **Instance Type:** t2.micro (sufficient for small-scale applications, as part of the AWS Free Tier).
- **Storage:** 8 GB of General Purpose SSD (gp2) to handle basic application files and logs. an Amazon Linux 2 AMI, which supports the application's dependencies.

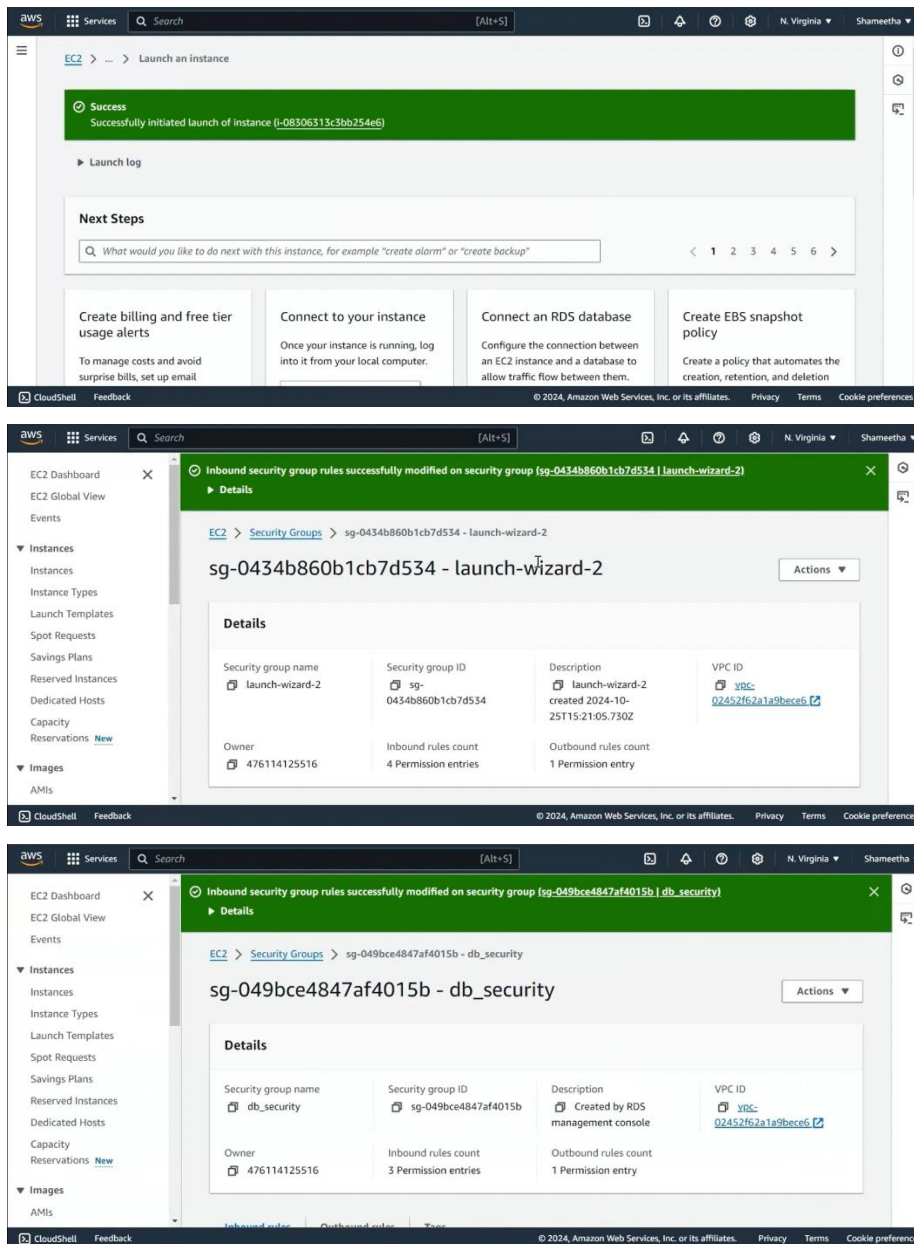
2. **Security Group Setup:** Configured to permit HTTP/HTTPS access for web traffic and SSH for administrative access. The MySQL port is also opened but limited to secure interactions between EC2 and RDS.

Type	Protocol	Port Range	Source
SSH	TCP	22	Your IP (e.g., 192.168.x.x/32)
MYSQL	TCP	3306	Your IP (e.g., 192.168.x.x/32)
HTTP	TCP	80	Anywhere (0.0.0.0/0)
HTTPS	TCP	443	Anywhere (0.0.0.0/0)

3. **Elastic IP Assignment:** Ensures a stable IP address for public access.

Setting	Configuration Details
Instance Type	t2.micro
AMI	Amazon Linux 2

EC2 Setup Workflow outlines the steps for launching, configuring, and managing the EC2 instance:

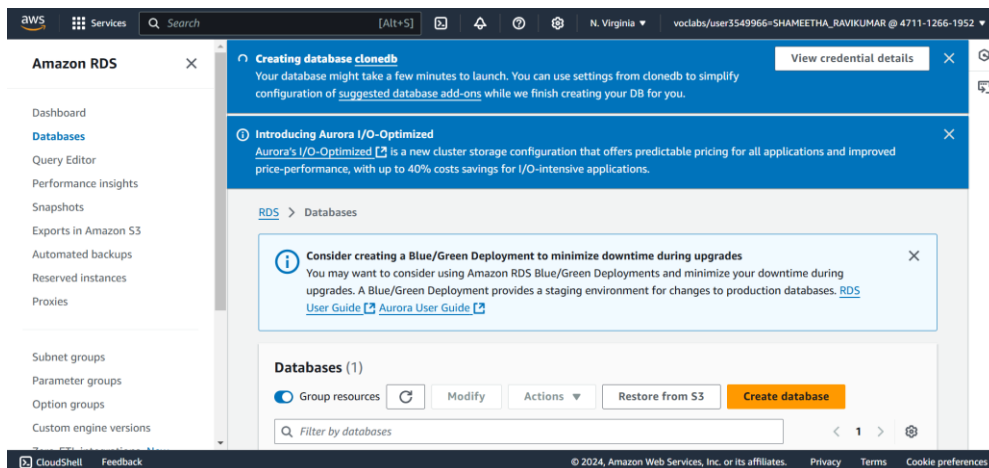


AWS EC2 Configuration

5.2 AWS RDS Configuration for MySQL Database

1. **Database Creation:** The MySQL engine is configured through AWS RDS with a designated initial database for the virtual classroom platform. The database is created with the necessary security settings, ensuring safe and reliable data storage.
2. **Security Group and VPC:** The RDS instance is placed in the same Virtual Private Cloud (VPC) as the EC2 instance hosting the Flask backend. The security settings ensure that access to the database is restricted to only the EC2 instance and trusted sources, with MySQL access limited to port 3306.

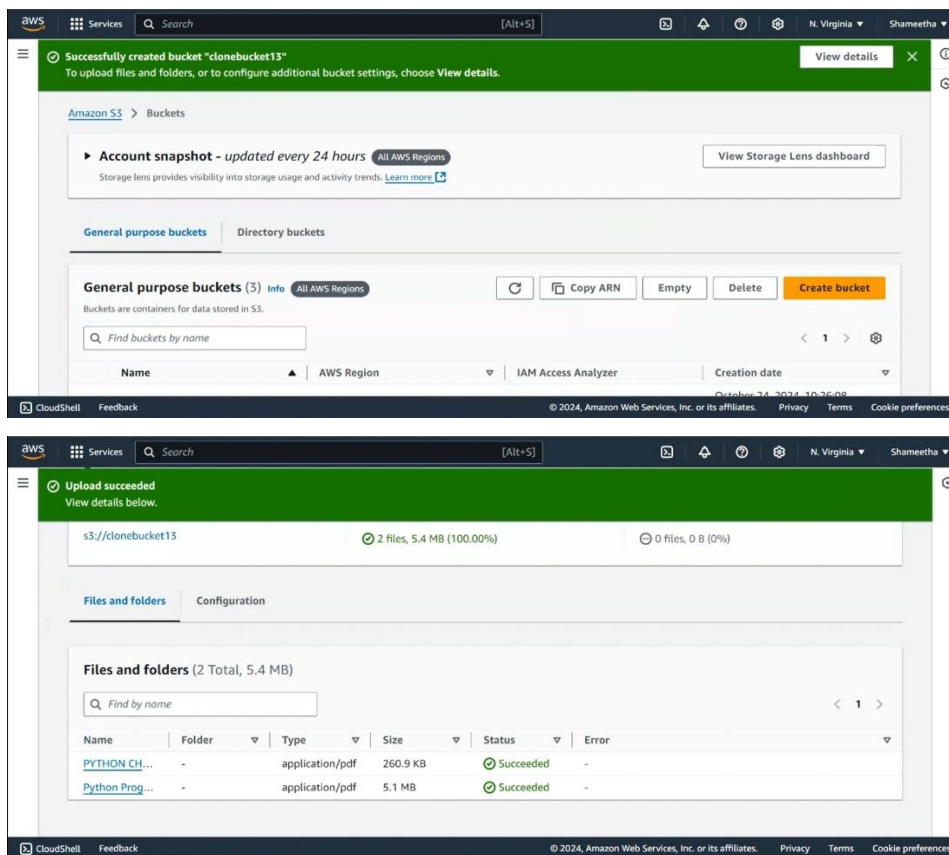
RDS Configuration	Details
Engine	MySQL
Instance Type	db.t2.micro

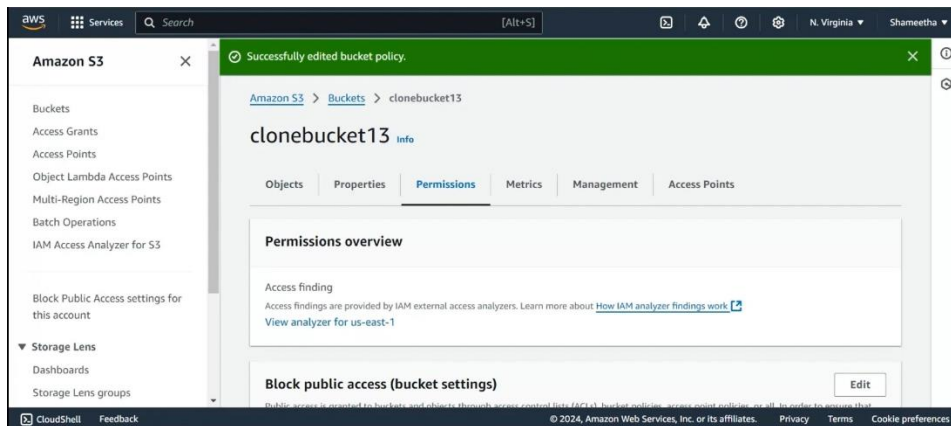


AWS RDS Configuration

5.3 AWS S3 Configuration

- **Bucket Creation:** The S3 bucket is created to store course materials such as PDFs, videos, assignments, and other resources. The bucket is configured to ensure secure and efficient access to these files.
- **Security Settings:** The bucket is configured with proper access controls and permissions to ensure that only authorized users and systems can access the files. These controls are enforced using IAM roles and policies.
- **File Storage:** All course-related files are stored in the S3 bucket, with the option to organize them into subfolders based on courses or file types.
- **Access Control:** Access to files is restricted through pre-signed URLs, ensuring that only authorized users can download materials.





AWS S3 Configuration

5.3 Front-End and Back-End Integration

The front-end of the virtual classroom platform is developed using HTML, CSS, and JavaScript, while the back-end is powered by Flask. Flask handles routing, authentication, and data processing, interacting with the MySQL database hosted on RDS to store and retrieve user information, course materials, and other platform data.

5.4 Database Design

The MySQL database schema consists of the following key table:

- User authentication system: Stores the usernames and hashed passwords of users.

```
C:\Windows\system32\cmd.exe - mysql -h clonedb.c56g4kmq4b38.us-east-1.rds.amazonaws.com -P 3306 -u admin1 -p
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>mysql -h clonedb.c56g4kmq4b38.us-east-1.rds.amazonaws.com -P 3306 -u admin1 -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 64
Server version: 8.0.39 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database clone_db
-> ;
Query OK, 1 row affected (0.30 sec)

mysql> use clone_db;
Database changed
mysql> create table users(username varchar(100) primary key,password_hash varchar (255));
Query OK, 0 rows affected (0.29 sec)
```

MYSQL Database Design

Chapter 6: Testing and Optimization

6.1 Functional Testing

Each module of the AWS-Hosted Virtual Classroom and Learning Platform was rigorously tested to ensure the system operates as intended. This included testing various user interactions, from the registration process to the course content access. The key areas of testing included:

- **Registration and Login**
 - Test Scenarios: Both valid and invalid inputs were tested, such as valid email addresses, weak passwords, and incorrect credentials.
 - Outcome: The system successfully allowed valid registrations and logins, while invalid inputs were rejected with appropriate error messages.
- **Blood Requests**
 - Test Scenarios: Users' ability to access and download course materials after logging in was tested, including tests for file access control through pre-signed URLs from S3.
 - Outcome: The system correctly allowed users to view available courses and download materials securely.

6.2 Database Optimization

To optimize the platform's performance, strategies like indexing frequently queried fields (e.g., user IDs and course names) were implemented, improving query response times. Redundant queries were minimized by optimizing Flask backend interactions with the RDS database, reducing load and enhancing performance. Additionally, query caching was used for frequently accessed data like course metadata, further reducing database load and improving response times.

6.3 Load Testing

Simulated high user volumes to evaluate EC2 and RDS scalability. Results indicated stable performance under expected load, and additional load balancers may be added as demand grows.

6.4 Security Testing

Conducted HTTPS and data encryption tests to secure sensitive information and applied access control policies, ensuring restricted access to the RDS database only from authorized sources.

Test Type	Outcome Summary
Functional Tests	Passed for all primary features
Load Testing	Stable performance under 500 concurrent users
Security Tests	Verified secure HTTPS connections and restricted access

OUTPUT

The image displays the AWS Virtual Classroom website and the terminal output of the registration process. The website has a purple header with the text "Welcome to AWS Virtual Classroom" and a "Register Now" button. Below the header is an "About Us" section. The terminal output shows the user running commands to set up the environment, including creating a virtual environment, installing dependencies, and running the application. The output also shows the user registering a new user with the username "Newuser" and password "*****".

Website Screenshot:

Register

Username:

Password:

Register

Login

Username:

Password:

Login

Terminal Output:

```
(venv) [ec2-user@ip-172-31-81-209:~/.aws-vc]$ scp -i C:\Users\lenovo\Downloads\clone.pem -r C:\shamee_aws_prjct\templates/home.html ec2-user@18.234.190.159:/home/ec2-user/
Warning: Identity file C:\Users\lenovo\Downloads\clone.pem not accessible: No such file or directory.
ec2-user@18.234.190.159: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

Connection closed
(venv) [ec2-user@ip-172-31-81-209:~/.aws-vc]$ mkdir /home/ec2-user/aws_vc/static
(venv) [ec2-user@ip-172-31-81-209:~/.aws-vc]$ python app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.31.81.209:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 820-605-310
49.47.243.171 - - [25/Oct/2024 21:11:54] "GET / HTTP/1.1" 200 -
49.47.243.171 - - [25/Oct/2024 21:11:55] "GET /static/styles.css HTTP/1.1" 200 -
49.47.243.171 - - [25/Oct/2024 21:11:55] "GET /favicon.ico HTTP/1.1" 404 -
49.47.243.171 - - [25/Oct/2024 21:11:58] "GET /register HTTP/1.1" 200 -
(venv) [ec2-user@ip-172-31-81-209:~/.aws-vc]$ python app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.31.81.209:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 820-605-310
49.47.243.171 - - [25/Oct/2024 21:13:03] "GET / HTTP/1.1" 200 -
49.47.243.171 - - [25/Oct/2024 21:13:05] "GET /static/styles.css HTTP/1.1" 304 -
49.47.243.171 - - [25/Oct/2024 21:13:12] "GET /register HTTP/1.1" 200 -

Using cached visitor-0.1.3-py3-none-any.whl
Requirement already satisfied: MarkupSafe>=2.0 in /home/ec2-user/aws_vc/venv/lib/python3.9/site-packages (from Jinja2>=3.0->Flask==2.0.1->r requirements.txt (line 1)) (3.0.2)
Installing collected packages: werkzeug, visitor, protobuf, Flask, dominate, mysql-c
connector-python, Flask-Bootstrap
Attempting uninstall: werkzeug
Found existing installation: werkzeug 3.0.6
Uninstalling werkzeug-3.0.6:
Successfully uninstalled werkzeug-3.0.6
Attempting uninstall: Flask
Found existing installation: flask 3.0.3
Uninstalling flask-3.0.3:
Successfully uninstalled flask-3.0.3
Attempting uninstall: mysql-connector-python
Found existing installation: mysql-connector-python 9.1.0
Uninstalling mysql-connector-python-9.1.0:
Successfully uninstalled mysql-connector-python-9.1.0
Successfully installed Flask-2.0.1 Flask-Bootstrap-3.3.7.1 dominate-2.9.1 mysql-conn
ector-python-8.0.26 protobuf-5.28.3 visitor-0.1.3 werkzeug-2.0.1
WARNING: You are using pip version 21.3.1; however, version 24.2 is available.
You should consider upgrading via the '/home/ec2-user/aws_vc/venv/bin/python3 -m pip
install --upgrade pip' command.
(venv) [ec2-user@ip-172-31-81-209:~/.aws-vc]$ python app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
Traceback (most recent call last):
  File "/home/ec2-user/aws_vc/app.py", line 76, in <module>
    app.run(debug=True)
  File "/home/ec2-user/aws_vc/venv/lib64/python3.9/site-packages/Flask/app.py", line 922, in run
    run_simple(t.cast(str, host), port, self, **options)
  File "/home/ec2-user/aws_vc/venv/lib64/python3.9/site-packages/werkzeug/serving.py", line 982, in run_simple
    s.bind(server_address)
OSError: [Errno 98] Address already in use
(venv) [ec2-user@ip-172-31-81-209:~/.aws-vc]$
```

MyS

Naar

PTY

pyth

18.234.190.159:5000/dashboard

Course Dashboard

Download Course Material 1

Download Course Material 2

Logout

```

ec2-user@ip-172-31-81-209:~/aws_vc/aws_vc
Using cached visitor-0.1.3-py3-none-any.whl
Requirement already satisfied: MarkupSafe>=2.0 in /home/ec2-user/aws_vc/venv/lib/python3.9/site-packages (from Jinja2>=3.0->Flask==2.0.1->r requirements.txt (line 1)) (3.0.2)
Installing collected packages: werkzeug, visitor, protobuf, Flask, dominate, mysql-connector-python, Flask-Bootstrap
  Attempting uninstall: werkzeug
    Found existing installation: werkzeug 3.0.6
    Uninstalling werkzeug-3.0.6:
      Successfully uninstalled werkzeug-3.0.6
  Attempting uninstall: Flask
    Found existing installation: flask 3.0.3
    Uninstalling flask-3.0.3:
      Successfully uninstalled flask-3.0.3
  Attempting uninstall: mysql-connector-python
    Found existing installation: mysql-connector-python 9.1.0
    Uninstalling mysql-connector-python-9.1.0:
      Successfully uninstalled mysql-connector-python-9.1.0
  Successfully installed Flask-Bootstrap-3.3.7.1 dominate-2.9.1 mysql-connector-python-8.0.26 protobuf-5.28.3 visitor-0.1.3 werkzeug-2.0.1
WARNING: You are using pip version 21.3.1; however, version 24.2 is available.
You should consider upgrading via the '/home/ec2-user/aws_vc/venv/bin/python3 -m pip install --upgrade pip' command.
(ec2-user@ip-172-31-81-209 aws_vc)$ python app.py
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
Traceback (most recent call last):
  File "/home/ec2-user/aws_vc/app.py", line 76, in <module>
    app.run(debug=True)
  File "/home/ec2-user/aws_vc/venv/lib64/python3.9/site-packages/flask/app.py", line 922, in run
    run_simple(t.cast(str, host), port, self, **options)
  File "/home/ec2-user/aws_vc/venv/lib64/python3.9/site-packages/werkzeug/serving.py", line 982, in run_simple
    s.bind(server_address)
OSError: [Errno 98] Address already in use
(ec2-user@ip-172-31-81-209 aws_vc)$

```

MyS

Naar

Dash

F

pyth

aws-project-virtualclassroom.s3.ap-south-1.amaz...

PYTHON+P... 35 / 142 75% +

```

"""This function greets to
the person with the provided message"""
print("Hello",name + ' ' + msg)

wish("MRCET","Good morning!")

Output:
Hello MRCET Good morning!

26

```

PYTHON PROGRAMMING

III YEAR/II SEM

MRCET

Below is a call to this function with one and no arguments along with their respective error messages.

```

>>> wish("MRCET") # only one argument
TypeError: wish() missing 1 required positional argument: 'msg'
>>> wish() # no arguments
TypeError: wish() missing 2 required positional arguments: 'name' and 'msg'

```

```

def hello(wish,hello):

```


Chapter 7: Conclusion

7.1 Summary

The AWS-hosted Virtual Classroom and Learning Platform project successfully demonstrates the integration of essential cloud services with Flask-based web application architecture to provide a scalable, accessible, and efficient learning platform. By utilizing Amazon EC2 for hosting, S3 for content storage, and RDS for managing user data, the application provides a robust backend infrastructure that ensures data security and application performance. Git and GitHub integration have streamlined version control and collaboration, while secure file transfer and key management practices, including SSH authentication and PuTTY configuration, have facilitated a secure deployment process. Overall, this project highlights the capability of cloud-based solutions to support dynamic, real-time applications that can be expanded and adapted to meet the evolving needs of users, making it a valuable resource for virtual education.

7.2 Future Work

Potential future developments include:

- **Scalability Enhancements:** Load balancing and auto-scaling will be implemented to ensure performance during peak usage, automatically adjusting infrastructure based on demand for high availability.
- **AI-Based Personalization:** Artificial intelligence can be integrated to recommend personalized courses and materials based on user preferences and performance, improving the learning experience.
- **Mobile App Development:** A mobile application would allow users to access courses and materials on-the-go, enhancing accessibility and convenience.
- **Enhanced Security Features:** Two-factor authentication (2FA) for

logins and more granular access control can be added to improve data protection and user security.

- **Offline Access:** Offline functionality could allow users to download and access course materials without an internet connection, beneficial in low-connectivity areas.
- **Gamification and Engagement Tools:** Adding gamification features like badges and leaderboards could boost user engagement, making learning more interactive and motivating.
- **Collaborative Features:** Real-time collaboration tools like live chat and video conferencing would enhance interaction between students and instructors, creating a more dynamic learning environment.

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>