

AWS-hosted Virtual Classroom and Learning Platform

Project Created By :

Suraj Vaijnath Milake



AWS-hosted Virtual Classroom and Learning Platform

Category: AWS Cloud Practitioner

Skills Required: Core Python, AWS, Python -flask

Project Description:

In the evolving digital education landscape, creating an efficient, scalable, and secure virtual learning environment is essential. The "AWS-hosted Virtual Classroom and Learning Platform" project addresses this need by developing a cloud-native platform that integrates key AWS services to deliver a seamless educational experience. The platform leverages Flask for backend development and is hosted on AWS EC2, ensuring customizable and scalable deployment. Amazon S3 is used to store course content and data, while Amazon RDS (MySQL) is employed to manage user registration, login data, and other relevant information.

Key Features;

- **Scalable Infrastructure:** AWS automatically handles increasing users.
- **Secure Data Management:** S3 and RDS ensure user data and course content are secure.
- **User-Friendly Design:** The platform provides an easy-to-use interface for students and educators.

Challenges;

- **AWS Service Configuration:** Setting up S3, RDS, and EC2 for the first time was complex and required thorough research and understanding of AWS's documentation.
- **Flask-AWS Integration:** Ensuring seamless integration between the Flask application and AWS services, particularly managing secure data storage and transfer.
- **Security Management:** Protecting sensitive user data while implementing secure login and data encryption protocols to prevent breaches.
- **Debugging and Testing:** Troubleshooting errors during development and deployment required meticulous testing and analyzing AWS logs for issue resolution.

Project Flow;

1. Create an AWS Account
2. Create an S3 Bucket and Upload Data
3. Create an RDS Instance
4. Create an EC2 Instance
5. Develop the Flask Application
6. Deploy Flask App on EC2
7. upload to Github

Architecture

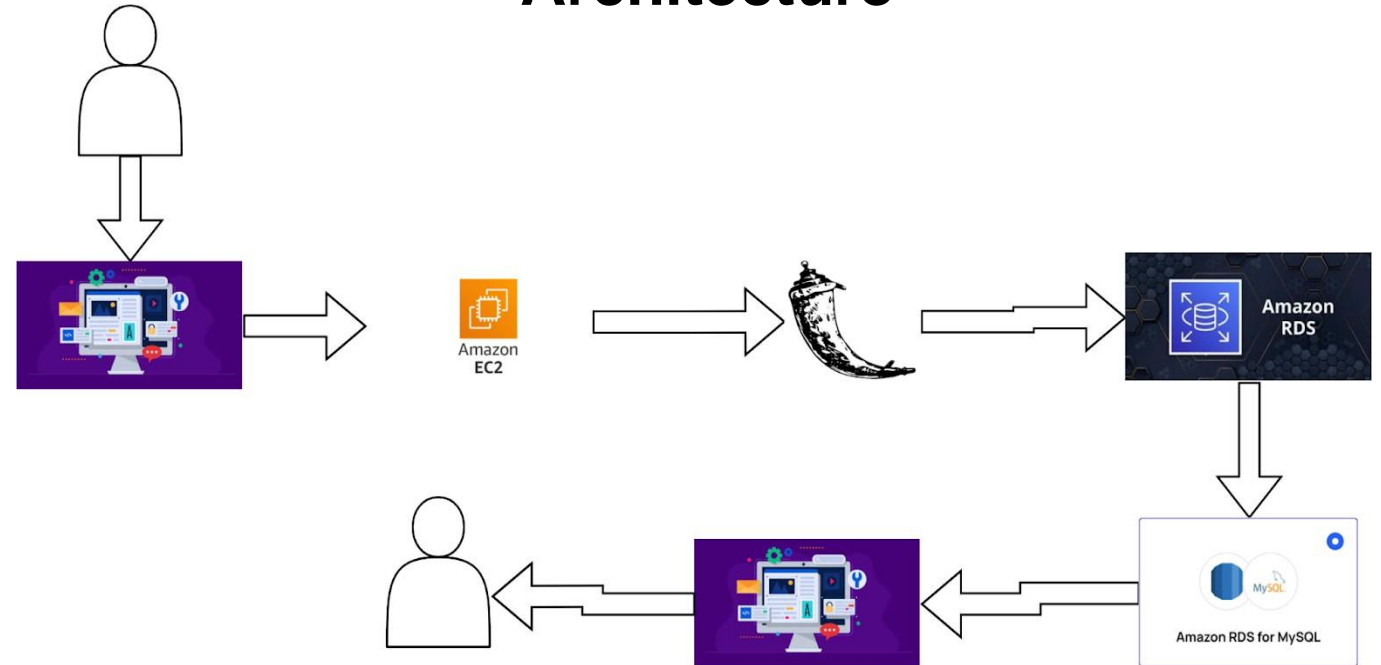
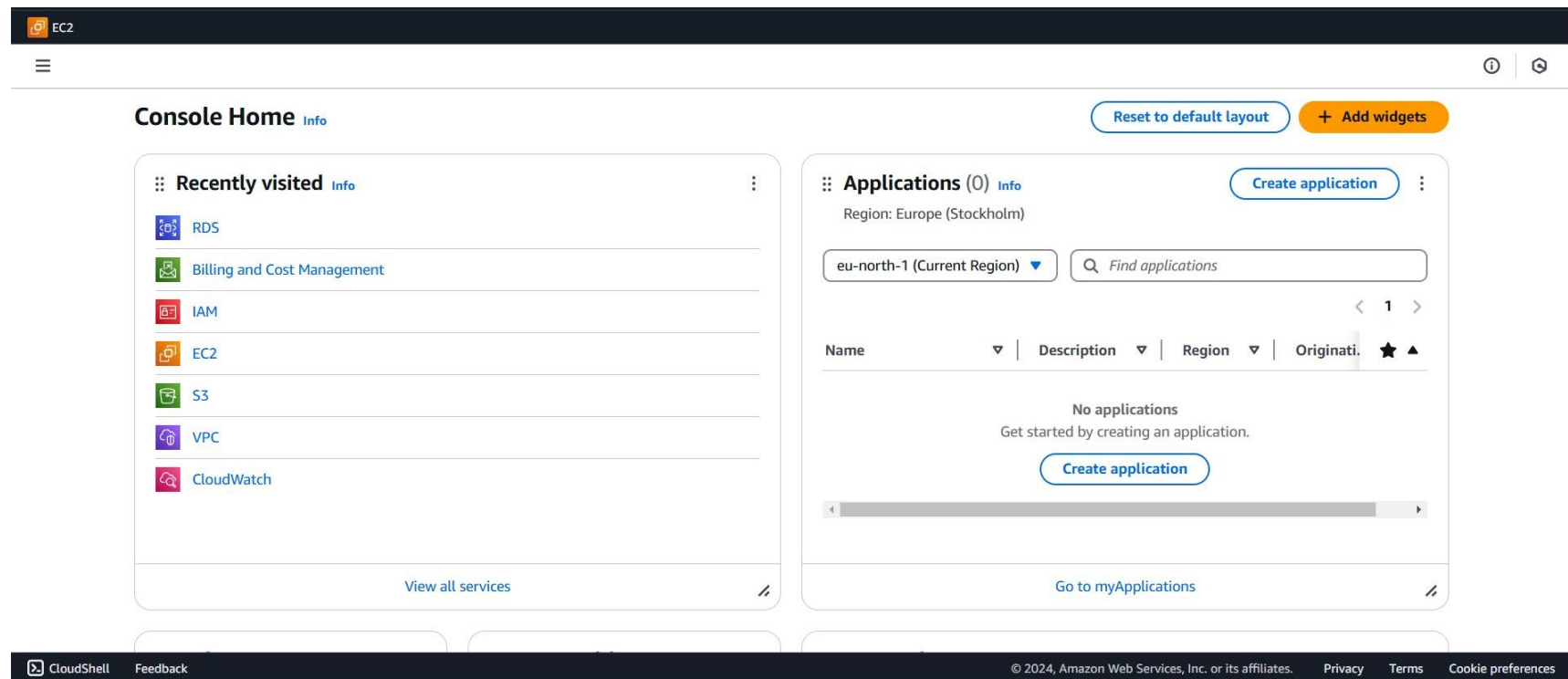


Fig 1: Architecture

Create an AWS Account

- Begin by creating an AWS account, and providing your personal and payment information.
- Complete the verification process to ensure the security of your AWS account.
- Navigate the intuitive AWS Management Console to familiarize yourself with the available services and features.

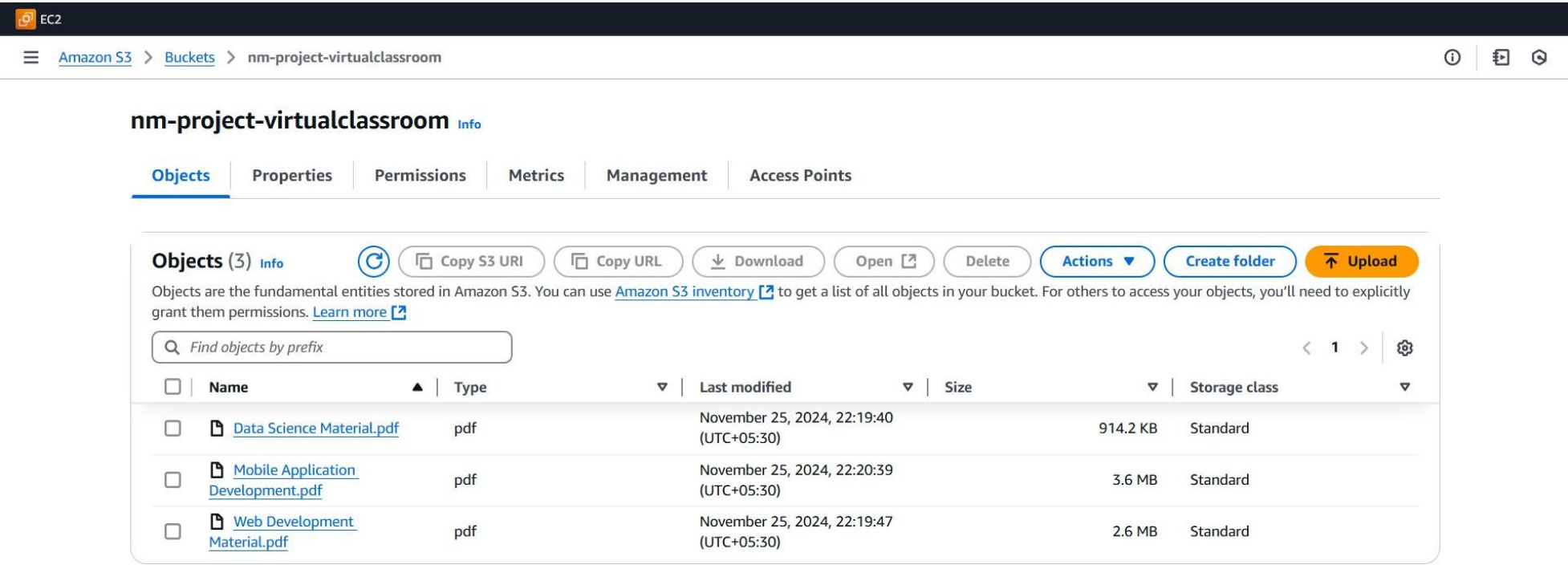
Fig 2: Create an AWS Account



Create an S3 Bucket and Upload Data

- Provision a secure and scalable S3 bucket to store your application data.
- Seamlessly transfer your data to the S3 bucket, ensuring it's readily available for your application.
- Configure appropriate permissions and policies to control your S3 bucket and data access.

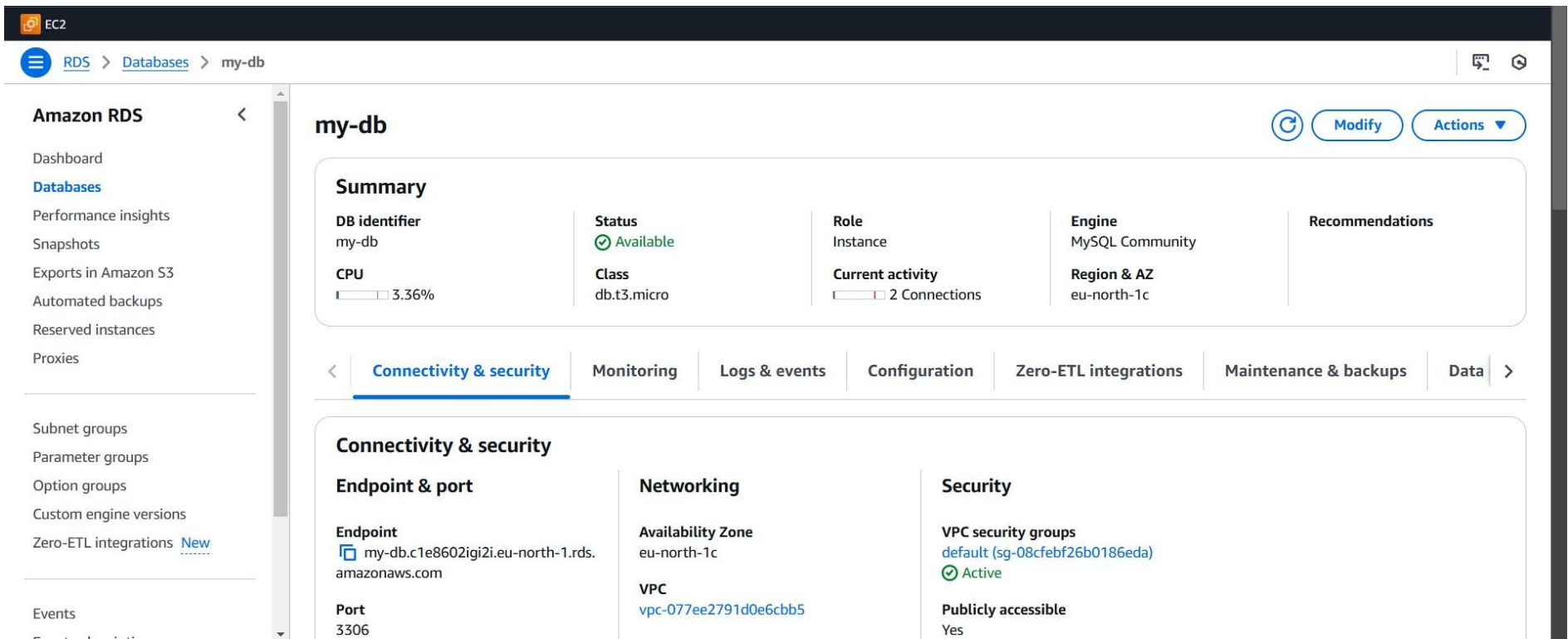
Fig 3: Create an S3 Bucket and Upload Data



Create an RDS Instance

- Select the appropriate database engine, MySQL, based on your application's requirements.
- Customize the RDS instance size, storage, and other settings to ensure optimal performance and scalability.
- Implement robust security measures, including network access control and encryption, to protect your data.
- Set up automated backups and disaster recovery mechanisms to safeguard your data.

Fig 4: Create an RDS Instance



Develop the Flask Application

Action:

1. Create Flask App:

- Develop app.py with routes for registration, login, and content.
- Create supporting HTML files (home.html, login.html, register.html, content.html).
- Add CSS styling using Bootstrap and custom styles.

2. Test Locally:

- Run the Flask application locally to ensure functionality.

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

Fig 5: Create a Project directory

Flask Application



Fig 6: home.html

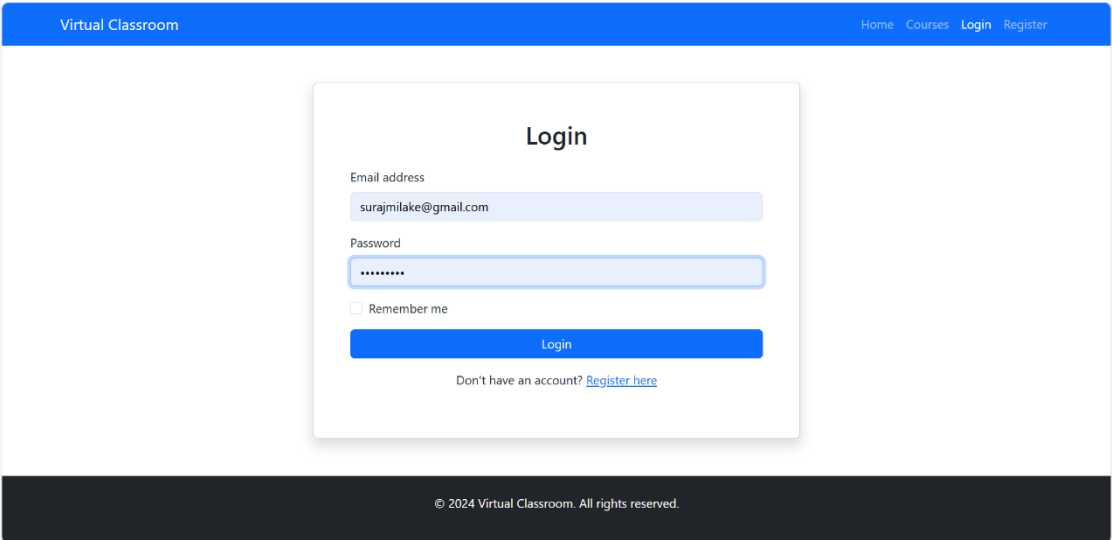


Fig 7: Login.html

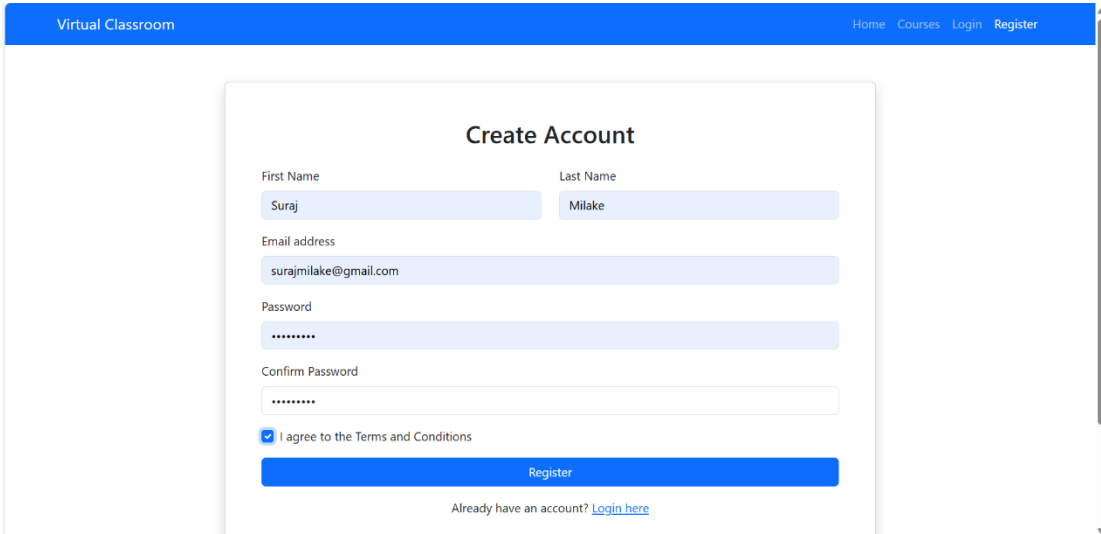


Fig 8: register.html

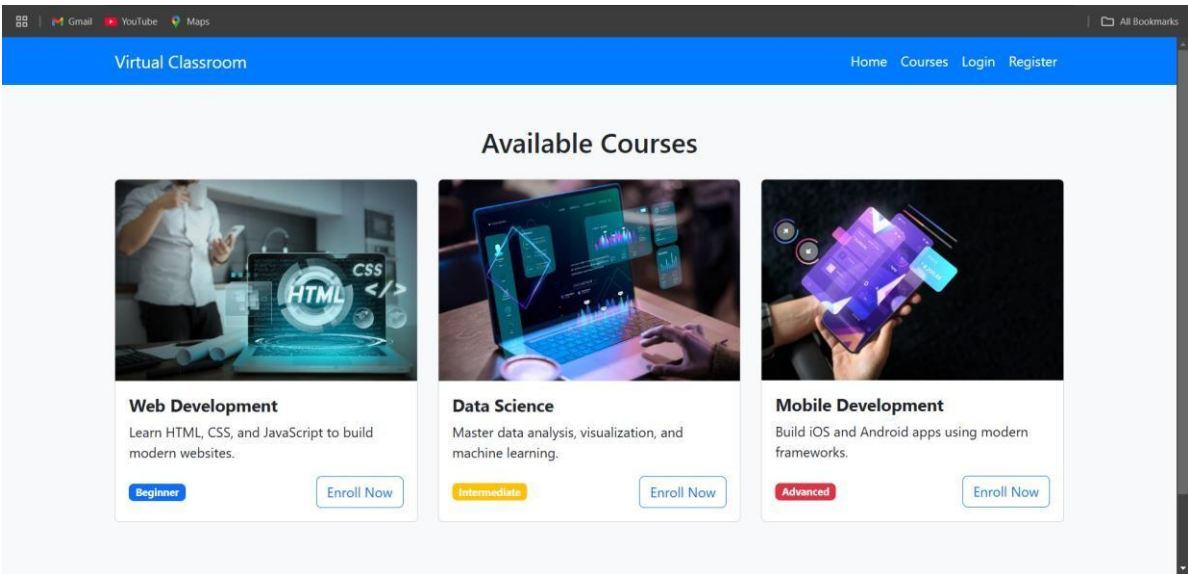


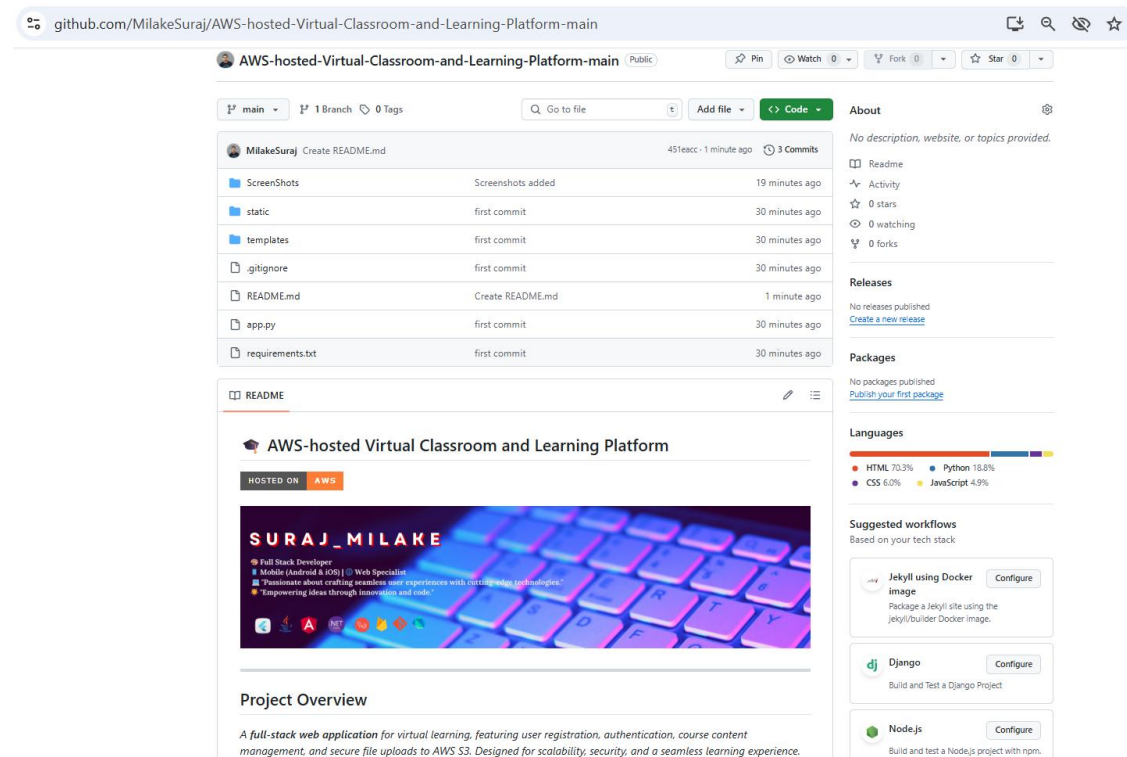
Fig 9: content.html

Upload to GitHub

Action:

- Create a new repository on GitHub.
- Commit your project files and push them to the repository.

Fig 10: Upload to GitHub



Conclusion

Summary: The project successfully integrates Flask with AWS services to create a scalable and secure virtual classroom platform. Users can register, log in, and access course materials hosted on S3, with data managed in RDS and the application deployed on EC2. The solution leverages AWS's robust infrastructure to deliver a seamless educational experience.

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>

THANK YOU