



NAAN MUDHALVAN



Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting

Project Created by:

Yogeshwar T - 412622104025

venkateswaran T - 412622104023

Thirupathi M -412622104022

Srijayaraman T -412622104020

Project Created Date: 21/Nov/2024

College Code: 4126

College Name: Sri Venkateswaraa College Of Technology

Team Name: NM2024TMID12721

BONAFIDE CERTIFICATE

Certified that this Naan Mudhalvan project report “**Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting**” is the bonafide work of _____ who carried out the project work under my supervision.

SIGNATURE
Project Coordinator
Naan Mudhalvan

SIGNATURE
SPoC
Naan Mudhalvan

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting

The Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting project aims to harness the power of cloud technologies to provide a comprehensive, scalable, and secure Harnessing environment. In today's rapidly evolving digital financial 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 financial 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 financial 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 Harnessing 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 Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting setup

exemplifies the potential of cloud-based solutions in supporting modern financial initiatives and represents a significant step forward in the delivery of online financial.

An e-commerce company requires a high-availability solution to host its website, which experiences significant traffic fluctuations. The project uses multiple EC2 instances spread across different availability zones within a VPC to ensure fault tolerance and load balancing. MobaXterm facilitates the quick deployment of updates and management of these instances, ensuring minimal downtime. This setup allows the e-commerce platform to handle peak traffic periods efficiently while maintaining consistent performance and security.

Project Description

Scenario 1: Secure Internal Application Hosting for a Financial Institution

A financial institution needs to host a secure internal application that handles sensitive client data. The project utilizes EC2 instances within a VPC to create isolated environments for hosting the application. MobaXterm is employed for secure remote access and management of these instances, ensuring that only authorized personnel can deploy updates or perform maintenance. This setup provides the institution with a highly secure, scalable, and easily manageable hosting environment, compliant with financial regulations.

Scenario 2: High-Availability Web Hosting for an E-Commerce Platform

An e-commerce company requires a high-availability solution to host its website, which experiences significant traffic fluctuations. The project uses multiple EC2 instances spread across different availability zones within a VPC to ensure fault tolerance and load balancing. MobaXterm facilitates the quick deployment of updates and management of these instances, ensuring minimal downtime. This setup allows the e-commerce platform

to handle peak traffic periods efficiently while maintaining consistent performance and security.

Scenario 3: Custom Web Application Deployment for a Start-Up

A tech start-up needs to deploy a custom web application that will serve as its primary business platform. The project sets up an EC2 instance within a VPC to host the application, ensuring it is secure and isolated from other internet traffic. MobaXterm is used to streamline the deployment process and manage the application server. This scenario provides the start-up with a flexible, scalable, and cost-effective solution, allowing them to focus on application development and business growth while maintaining control over their hosting environment.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	3
1	INTRODUCTION	7
2	LITERATURE REVIEW	9
3	TECHNOLOGIES USED	11
4	PROJECT FLOW	13
5	IMPLEMENTATION DETAILS	14
6	TESTING AND OPTIMIZATION	18
7	CONCLUSION	20
	REFERENCES	22

CHAPTER 1

INTRODUCTION

1.1 Background

A financial institution needs to host a secure internal application that handles sensitive client data. The project utilizes EC2 instances within a VPC to create isolated environments for hosting the application. MobaXterm is employed for secure remote access and management of these instances, ensuring that only authorized personnel can deploy updates or perform maintenance. This setup provides the institution with a highly secure, scalable, and easily manageable hosting environment, compliant with financial regulations.

The rapid evolution of technology has transformed various aspects of human life, with financial 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 Harnessing environments.

In response to the growing demand for flexible and secure online financial systems, the **Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting** 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

A tech start-up needs to deploy a custom web application that will serve as its primary business platform. The project sets up an EC2 instance within a VPC to host the application, ensuring it is secure and isolated from other internet traffic. MobaXterm is used to streamline the deployment process and manage the application server. This scenario provides the start-up with a flexible, scalable, and cost-effective solution, allowing them to focus on application development and business growth while maintaining control over their hosting environment.

1.3 Scope

The scope of the Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting encompasses the following key features and functionalities:

- **User Management:** Facilitates user registration, login authentication, and profile management.
- **Course Content Management:** Stores and delivers multimedia course materials, including documents and videos, leveraging AWS S3 for scalability.
- **Interactive Web Application:** Utilizes Flask as the backend framework to provide dynamic features, including dashboards and user-specific functionalities.
- **Scalable Cloud Infrastructure:** Deploys applications on AWS EC2, ensuring high availability and adaptability to varying user demands.
- **Data Security and Management:** Implements secure data storage and access control mechanisms using AWS RDS, focusing on user privacy and data integrity.

CHAPTER 2

LITERATURE REVIEW

2.1 Evolution of Harnessing Platforms: The journey from traditional Entity Hosting learning to Hosting financial has been marked by significant technological advancements. Early Harnessing 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 Hosting financial.

2.2 Cloud Computing in Financial: Cloud computing has revolutionized the financial sector by providing on-demand access to computing resources. Financial 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 Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting, such as high-volume user access and multimedia content delivery.

2.3 Role of AWS in Building Financial Platforms: Amazon Web Services (AWS) has become a preferred choice for developing financial 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 Financial Systems: Security is a critical aspect of harnessing 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 financial 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 Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting.

2.6 Trends and Challenges in Hosting Financial: Recent trends in Hosting financial 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 Hosting financial. 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 financial needs.

CHAPTER 3

TECHNOLOGIES USED

3.1 Overview of System Architecture

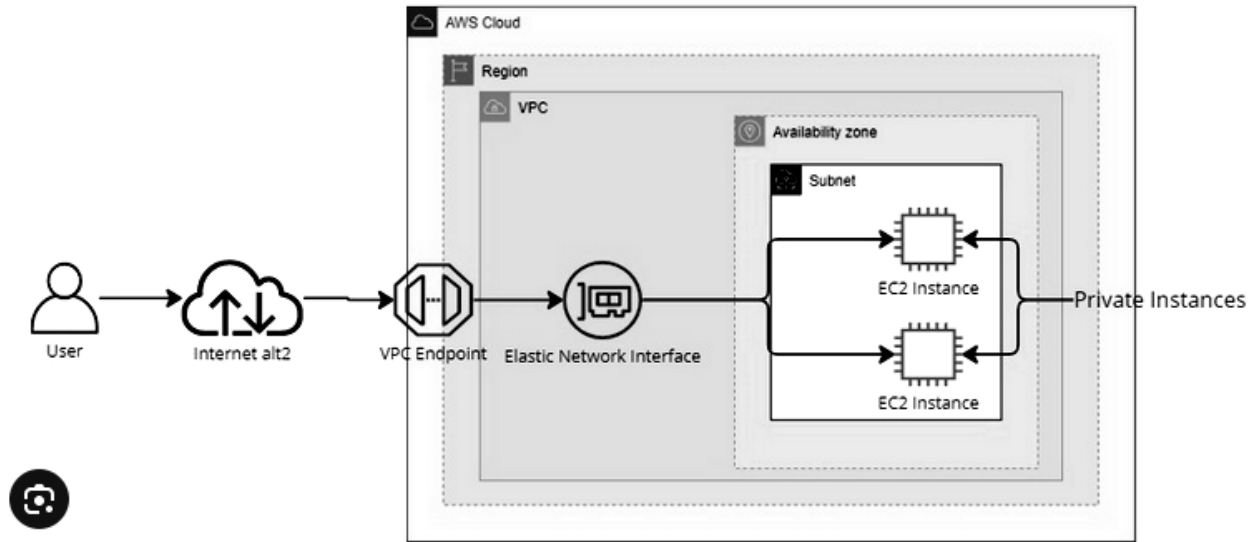
This section provides a high-level overview of the architecture used for the Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting. It outlines the interaction between key components, including the web application, database, cloud storage, and end-user devices. The architecture is designed 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.

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

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.

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.

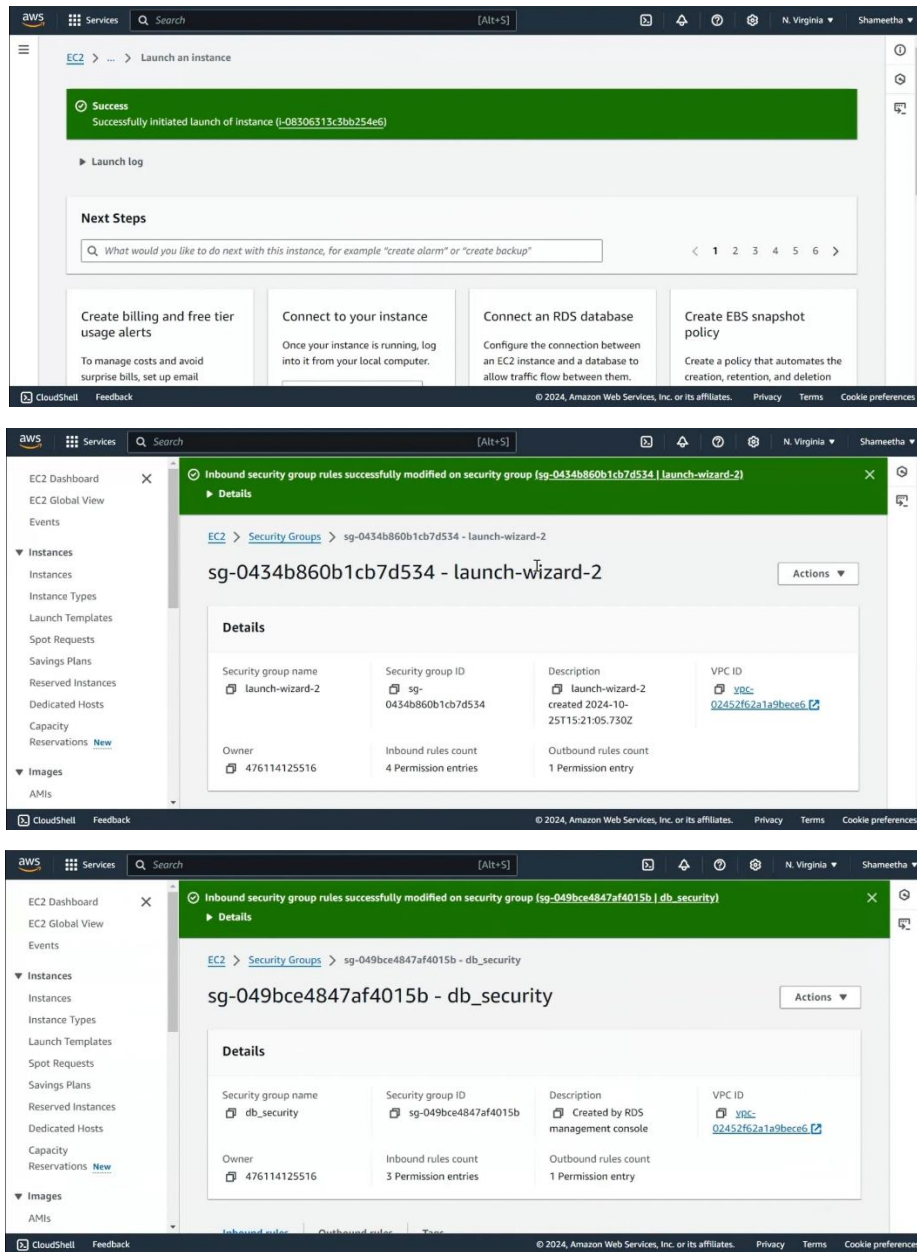
2. **Security Group Setup:** Configured to permit HTTP/HTTPS access for web traffic and SSH for administrative access. The MySQL port is also opened but limited to secure interactions between EC2 and RDS.

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:



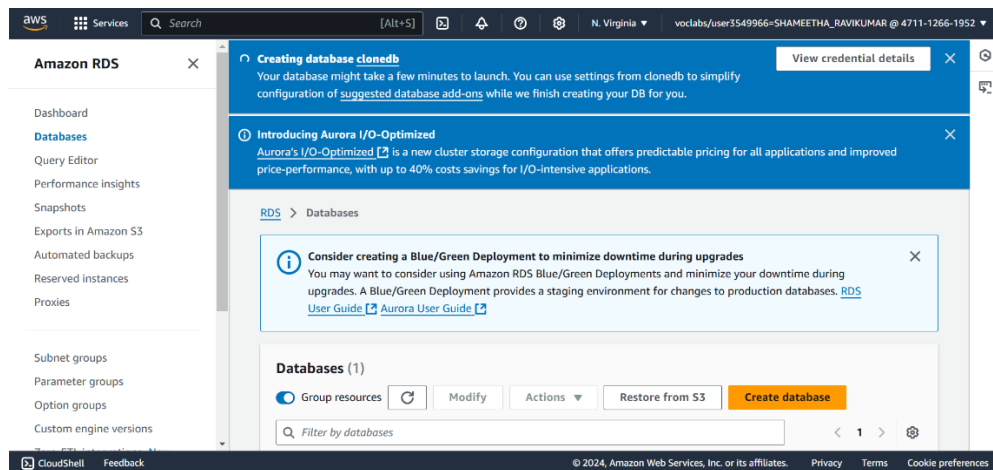
AWS EC2 Configuration

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 Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting 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 Hosting 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.

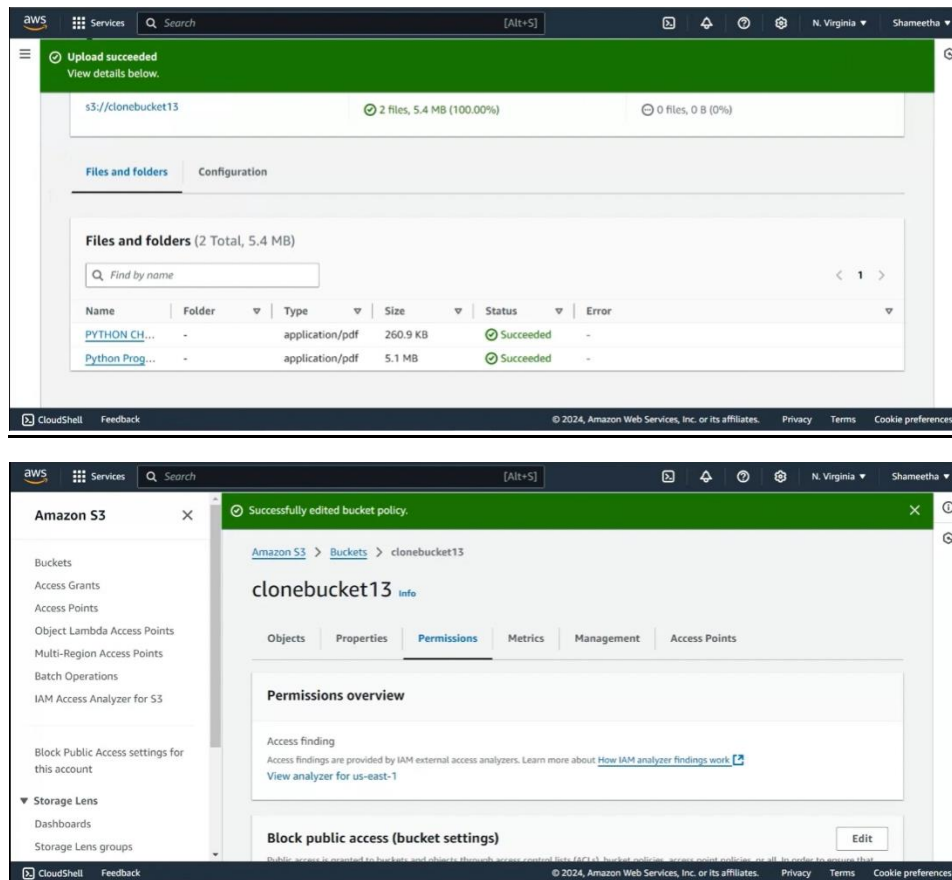
RDS Configuration	Details
Engine	MySQL
Instance Type	db.t2.micro



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 Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting 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.

CHAPTER 6

TESTING AND OPTIMIZATION

6.1 Functional Testing

Each module, from user registration to inventory updates, underwent rigorous testing to ensure accurate functionality:

- **Registration and Login:** Tested for valid and invalid input cases.
- **Requests:** Verified data submission, retrieval, and update accuracy.

6.2 Database Optimization

Optimization techniques, such as indexing frequently queried fields (e.g., user IDs, blood types), improved database response times. Additionally, reducing redundant queries minimized server load.

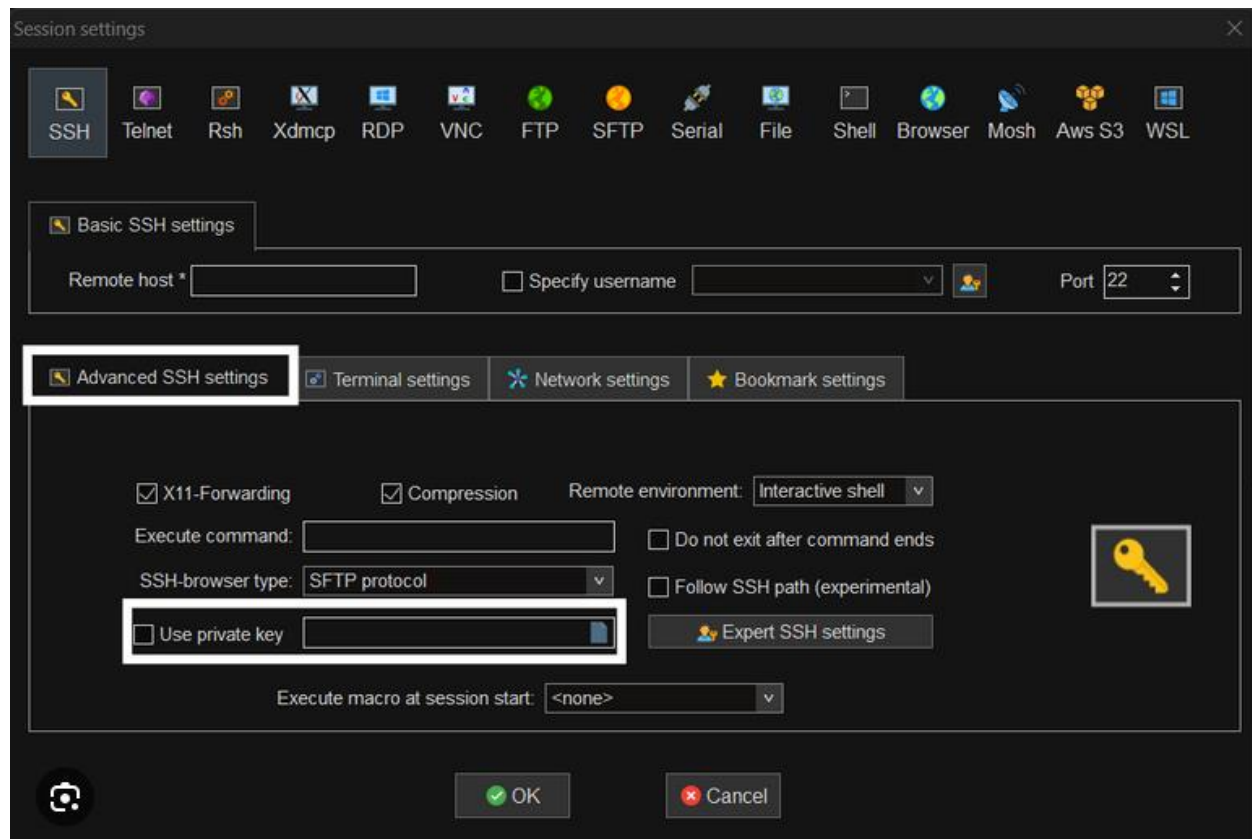
6.3 Load Testing

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.

SAMPLE USER REGISTRATION SCREEN



CHAPTER 7

CONCLUSION

7.1 Summary

The Harnessing EC2, VPC & MobaXterm for Cutting-Edge Entity Hosting 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 Hosting financial.

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

- **AWS Account Setup:** https://youtu.be/CjKhQoYeR4Q?si=ui8Bvk_M4FfVM-Dh
- **Web Application Stack : FLask || MySQL Connector using flask || HTML/JS/CSS**
- **AWS EC2 Instance:** https://www.youtube.com/results?search_query=aws+ec2+oneshot
- **RDS Database:** https://www.youtube.com/results?search_query=rds+oneshot
- **MySQL:** https://www.youtube.com/results?search_query=mysql+tutorial
- **AWS Cost Management:** <https://youtu.be/OKYJCHHSWb4?si=aY3DQl1v26CfZxXA>