A Report

On

Tech Trivia

Submitted to

University of Petroleum and Energy Studies

In Partial Fulfilment for the award of the degree of

BACHELORS IN TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING (with specialization CCVT)

By

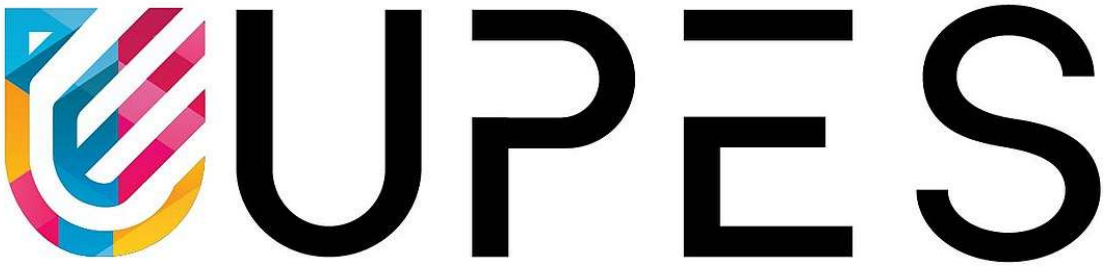
Kartikeya Bishnoi

500090969

Under the guidance of

Prof. SAURABH SHANU

ASSISTANT PROFESSOR(SG)



University of Petroleum and Energy Studies

Dehradun-India

November 2023

Contents

List Of Figures

Cloud Performance Tuning Basics

Hosted applications and services on cloud infrastructure undergo a crucial process - cloud performance tuning. The aim is to optimize their efficiency, ensuring the responsiveness of cloud-based systems by enhancing their overall performance, cost-effectiveness, and reliability.

Optimizing resource utilization is a crucial element when tuning cloud performance. It involves fine-tuning virtual machine configurations, optimizing network configurations, and adjusting storage settings to make sure that resources are allocated effectively. This approach enables organizations to boost performance without wasteful resource usage, thereby saving costs.

Applications with high traffic volumes or ones that necessitate real-time interactions require optimal code, database, and component optimization to minimize latency, resulting in an improved user experience. Tuning cloud performance includes enhancing application responsiveness, which is a vital element.

Cloud performance tuning considers the issue of scalability. To maintain a stable performance during busy periods and make the most out of resources during slower times, it's necessary to ensure that infrastructure can smoothly adapt to changing workloads.

For users to enjoy cloud-based systems that operate at their best and deliver optimal performance, reliability, and efficiency, it is vital to use proactive measures like cloud performance tuning. Cost management should also be an important aspect of fine-tuning the cloud. This approach is an ongoing process that continually adapts to workload changes and newly developed cloud technologies. By doing this, it establishes a reliable foundation for cloud environments that are highly responsive and high performing.

Problem Statement

Design and implement improvements for an existing online quiz platform that captures user information, presents quiz questions, records user responses, and displays a global scoreboard along with individual user rankings.

Our task is to enhance the functionality and user experience of an online quiz platform. Currently, the platform has the following features:

1. User Information: Collects username and SAP ID as inputs.
2. Quiz Questions: Displays a set of questions to the user.
3. Answer Recording: Records user responses to the quiz questions.

4. Score Calculation: Calculates and saves the user's score based on correct answers.
5. Global Scoreboard: Displays the scores of all participants.
6. User Ranking: Displays the user's ranking among all participants.

Our goal is to improve and extend the existing functionality. Consider the following aspects:

1. **User Authentication:** Implement a secure user authentication system to ensure that only registered users can participate. This could involve integrating user accounts with email verification or other secure authentication methods.
2. **Question Pool:** Create a dynamic question pool, so that users are presented with a diverse set of questions in each quiz session. Ensure that questions are randomly selected to enhance the variety of quizzes.
3. **Real-time Updates:** Implement real-time updates for the global scoreboard. Scores and rankings should be updated instantly as users complete the quiz. Use technologies such as WebSocket to achieve real-time communication.
4. **Leaderboard:** Extend the global scoreboard to include a leaderboard that showcases the top performers. This could be based on weekly, monthly, or all-time performance.
5. **User Profiles:** Allow users to create profiles where they can track their quiz history, view detailed statistics, and compare their performance with others.
6. **Mobile Responsiveness:** Ensure the platform is responsive and accessible on various devices, including desktops, tablets, and smartphones.
7. **Scalability:** Design the platform with scalability in mind. Consider potential future growth in terms of user base and question database.
8. **User-Friendly Interface:** Enhance the overall user interface and experience. Make sure the quiz flow is intuitive, and users can easily navigate through different sections of the platform.

Background

The current landscape of education and knowledge-sharing has seen a significant shift towards online platforms. With the increasing demand for remote learning and skill development, online quiz platforms have become valuable tools for educators, organizations, and individuals alike. These platforms offer a dynamic and interactive way to assess knowledge, challenge users, and foster a competitive spirit.

The existing online quiz platform under consideration was developed to meet this growing need. It was designed with the goal of providing users with an engaging and educational experience through quizzes. The platform currently collects basic user information such as name and SAP ID, presents a set of quiz questions, records user responses, calculates scores, and displays both global and individual rankings.

While the platform has laid the foundation for a meaningful quiz experience, there is room for improvement to enhance user engagement, security, and overall functionality. The proposed enhancements aim to address these areas, ensuring a more robust and user-friendly platform.

In the ever-evolving landscape of online education and assessment, it is crucial for the platform to evolve as well. By implementing advanced features such as real-time updates, a dynamic question pool, and user profiles, the platform can provide a more personalized and interactive learning experience. Additionally, incorporating security measures and scalability considerations will contribute to the long-term sustainability and success of the platform.

As online learning continues to be a prominent mode of education, the improved quiz platform has the potential to cater to a diverse audience, including students, professionals, and lifelong learners. The integration of modern technologies and a focus on user experience will position the platform as a competitive and valuable resource in the online education ecosystem.

Motivation/need for the cloud performance tuning.

In the context of the online quiz platform project, leveraging cloud services provides a scalable and efficient infrastructure. Cloud performance tuning becomes crucial to optimize the platform's functionality, ensuring a seamless and responsive experience for users. Here are key motivations and needs for implementing cloud performance tuning in this project:

1. **Scalability and Resource Management:**
    - **Motivation:** As the user base of the quiz platform grows, demand for resources will increase. Cloud performance tuning enables dynamic resource allocation, ensuring the platform can scale seamlessly to handle varying levels of traffic.
    - **Need:** Implementing auto-scaling configurations based on demand ensures that the platform remains responsive during peak usage periods, preventing performance bottlenecks.
2. **Cost Optimization:**

- **Motivation:** Cloud resources come at a cost, and efficient resource utilization is essential to control expenses. Performance tuning allows for optimizing resource allocation and minimizing unnecessary expenditures.
- **Need:** Implement cost-effective strategies such as right-sizing instances, utilizing reserved instances, and optimizing storage to achieve maximum efficiency without compromising performance.

3. **Response Time and User Experience:**
    - **Motivation:** Users expect a fast and responsive quiz platform. Cloud performance tuning aims to reduce latency and improve overall response times, contributing to a positive user experience.
    - **Need:** Fine-tune configurations, leverage content delivery networks (CDNs), and optimize database queries to enhance the platform's responsiveness, making it more enjoyable for users.
4. **Reliability and Availability:**
    - **Motivation:** Cloud platforms offer high availability, but performance issues can impact reliability. Performance tuning helps ensure consistent and reliable access to the quiz platform.
    - **Need:** Implement strategies such as load balancing, fault tolerance, and disaster recovery configurations to enhance the platform's availability and minimize downtime.
5. **Security and Compliance:**
    - **Motivation:** Security is paramount in online platforms, especially when handling user data. Performance tuning includes optimizing security measures to protect against potential vulnerabilities and ensure compliance with data protection regulations.
    - **Need:** Regularly update security configurations, apply encryption, and conduct performance testing to identify and address any security-related performance concerns.
6. **Monitoring and Analytics:**
    - **Motivation:** Continuous monitoring provides insights into the platform's performance and helps identify potential issues proactively. Cloud performance tuning involves setting up robust monitoring and analytics to track key performance metrics.

- **Need:** Implement monitoring tools, set up alerts for unusual activity, and use analytics to identify performance trends, enabling timely adjustments to maintain optimal performance.

7. **Adaptability to Changing Workloads:**
    - **Motivation:** Workloads on the quiz platform may vary over time. Performance tuning allows the platform to adapt to changing demands by dynamically adjusting resources and configurations.
    - **Need:** Implement workload-aware configurations, performance testing, and continuous optimization to ensure the platform can handle fluctuations in user activity without degradation in performance.

Objective

All the components in the cloud system can be tuned to enhance the functionality of the cloud features. However, some components are more important for certain features than others. For example, hardware is essential for scalability, elasticity, and agility, while software is more important for fault tolerance, disaster recovery, and availability.

	Scalability	Elasticity	Agility	Fault Tolerance	Disaster Recovery	Availability
Hardware	✓		✓	✓	✓	✓
Software	✓	✓	✓	✓	✓	✓
Network			✓			✓
Storage	✓	✓		✓	✓	
Analysis			✓			

Figure 1 Possible ways to Tune Performance

Here are some specific examples of how tuning the components in the table can enhance the functionality of the cloud features:

- Scalability: Hardware can be scaled up or down to meet the changing needs of the cloud application. For example, if the application is experiencing a surge in traffic, more hardware can be added to handle the load.
- Elasticity: Software can be configured to automatically scale up or down based on demand. This helps to ensure that the cloud application is always able to meet the needs of its users.
- Agility: Hardware, software, and network resources can be provisioned and deployed quickly and easily in the cloud. This makes it easy to spin up new cloud applications or to scale existing applications up or down.
- Fault tolerance: Hardware and software can be configured to be fault-tolerant. This means that if one component fails, the other components can continue to operate. This helps to ensure that the cloud application is always available to its users.
- Disaster recovery: Hardware, software, and data can be replicated to multiple geographic locations. This means that if there is a disaster in one location, the cloud application can continue to operate from another location. This helps to ensure that the cloud application is always available to its users.
- Availability: All of the components in the cloud system can be tuned to improve availability. For example, hardware can be replicated, software can be configured to be fault-tolerant, and data can be backed up. This helps to ensure that the cloud application is always available to its users.

Mode of achieving objective

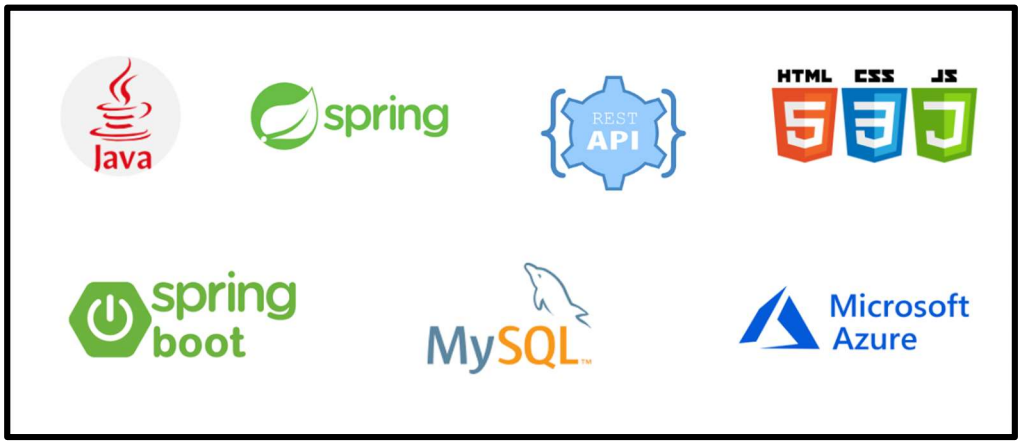


Figure 2 Tech Stack

Project GitHub Link

https://github.com/KartikeyaBishnoi/CPT-Project/tree/master

Deploying Spring Boot Application on Azure VM (Windows)

1. **Azure Account:**

   - Login to an active Azure account.

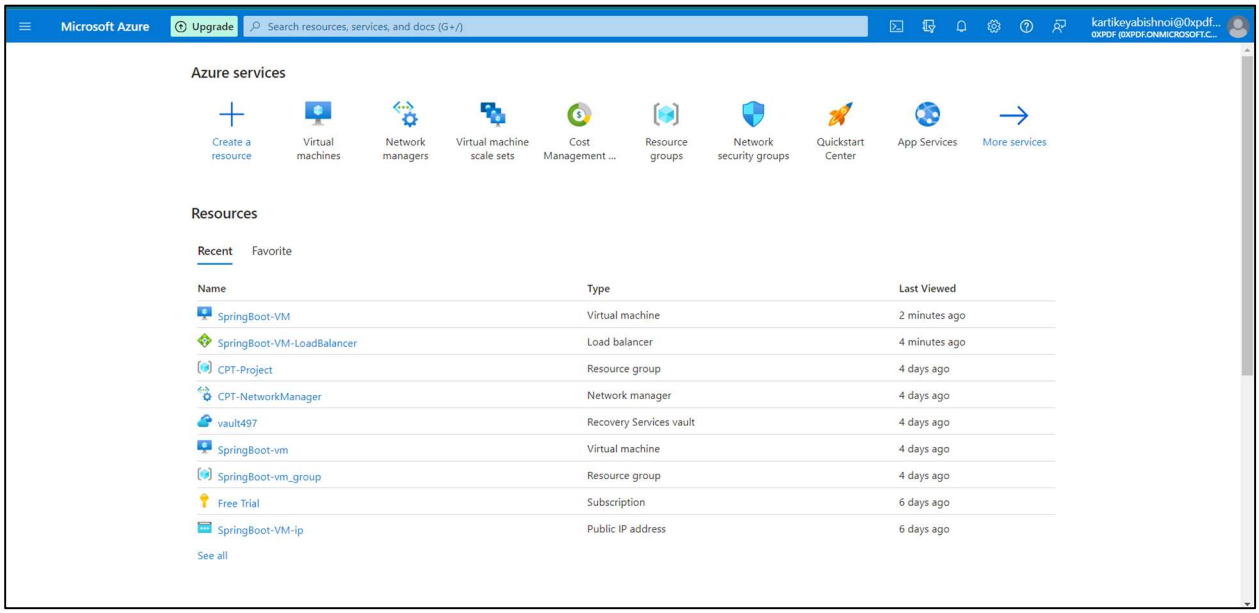


Figure 3 Azure Console

2. **Azure Resource Group**

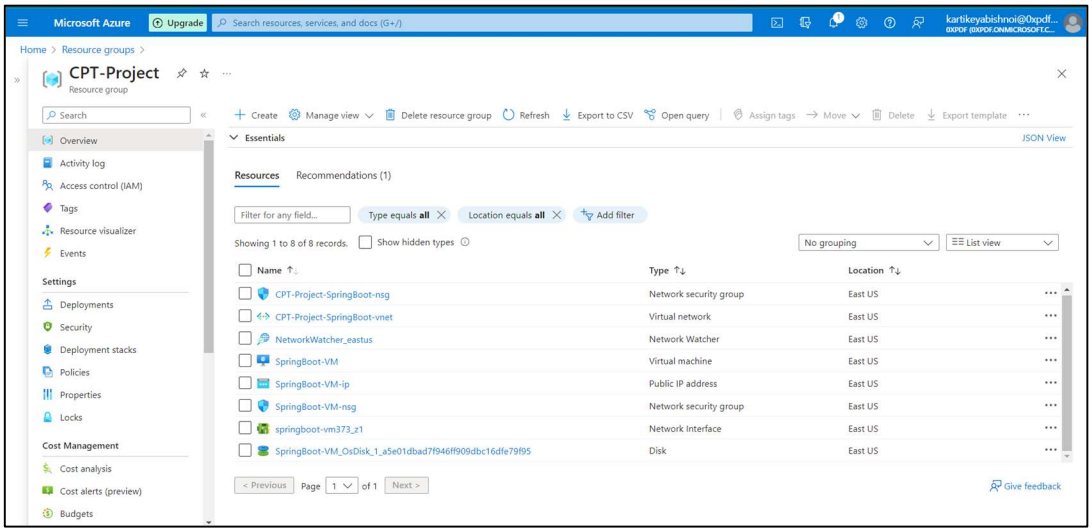


Figure 4 Azure Resource Group creation

3. **Azure VM:**

- Create a Windows VM on Azure.

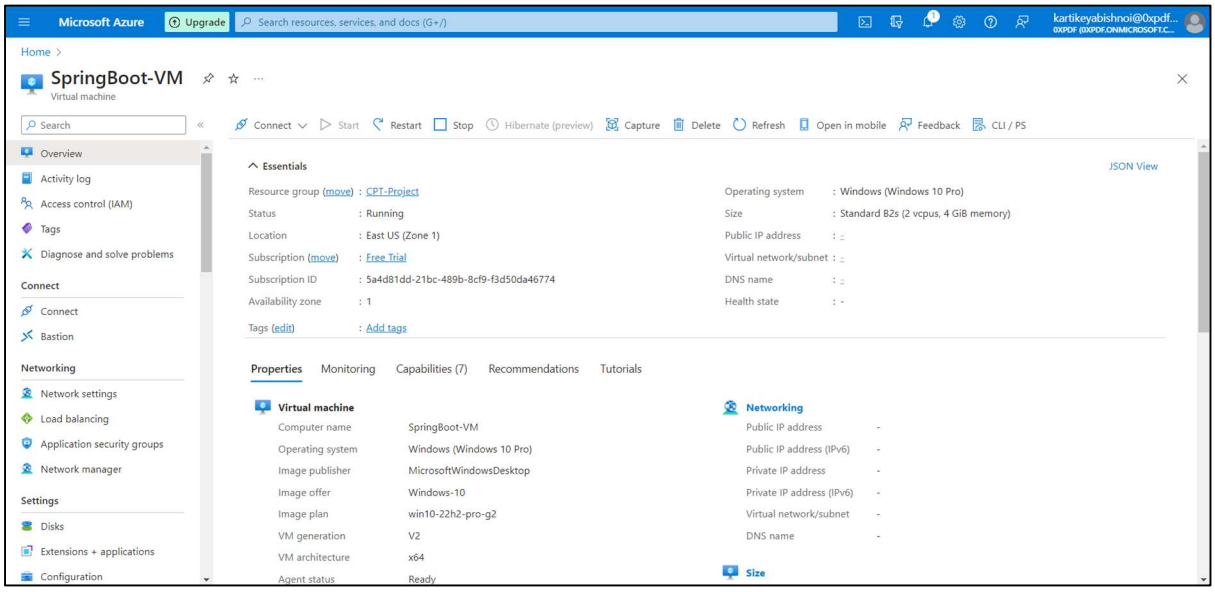


Figure 5 Azure VM creation

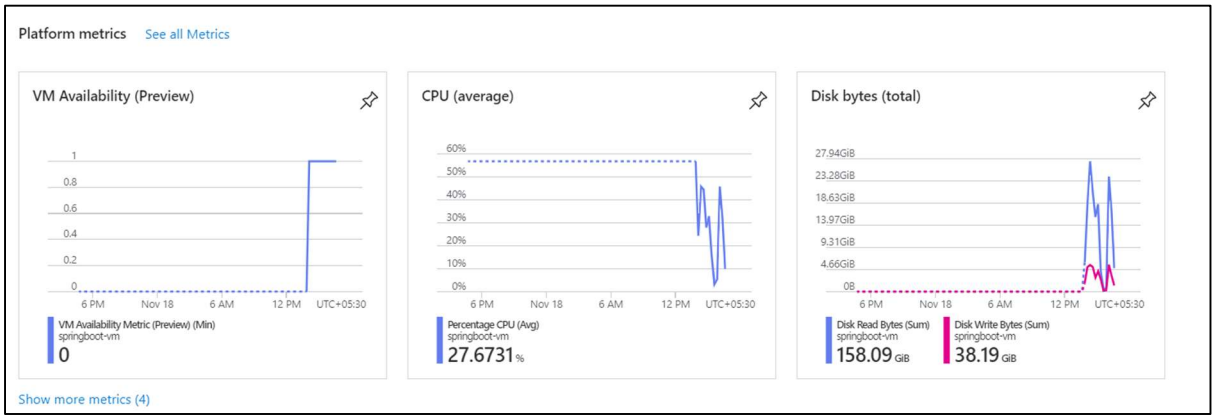


Figure 6 VM Metrics just after Spin-Up

4. **Tools Installation:**

- Install Java on the VM.
- Install IntelliJ IDEA.

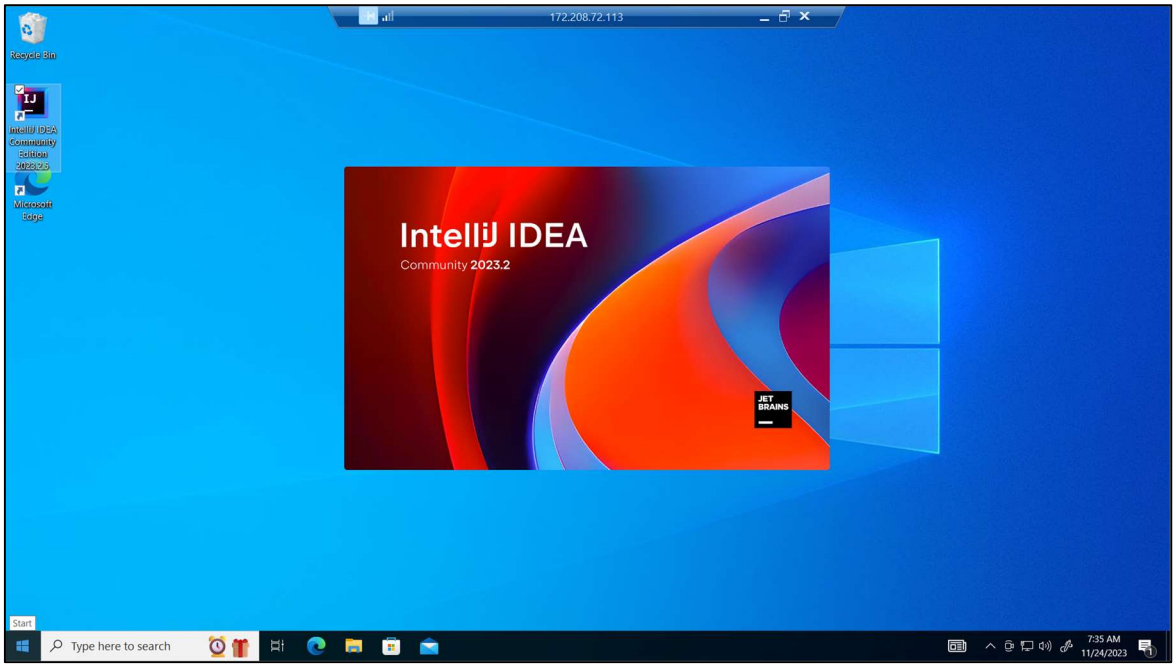


Figure 7 IntelliJ IDEA installation on Azure VM

- Install MySQL and MySQL Workbench.

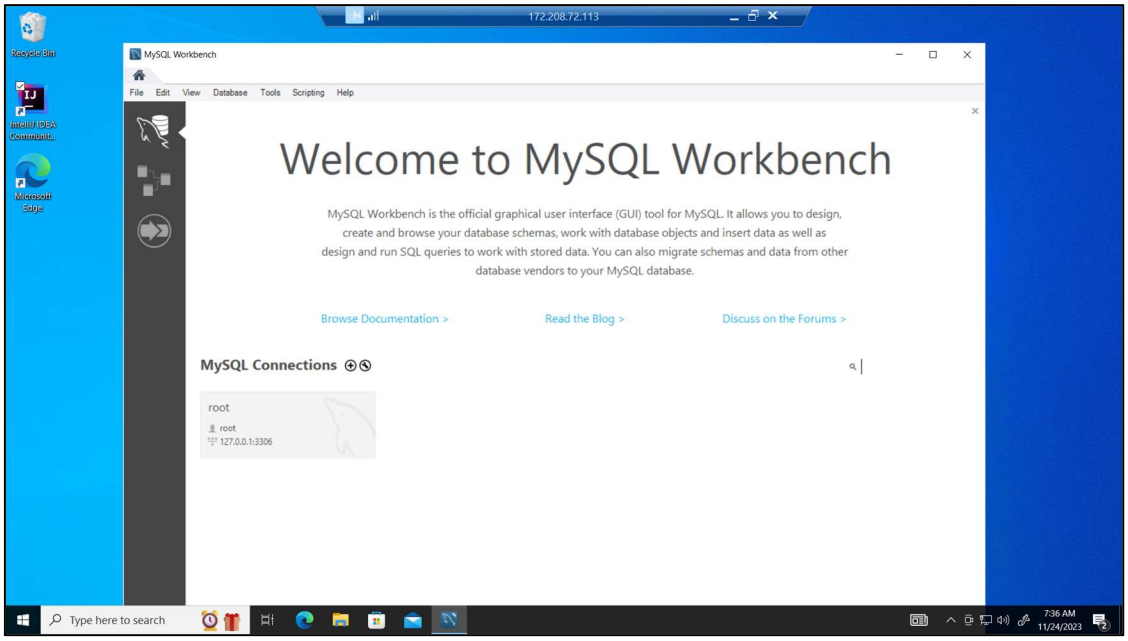


Figure 8 MySQL Workbench Setup on Azure VM

- Install Git for version control.

Step 1: Connect to Azure VM

- Use Remote Desktop Protocol (RDP) to connect to your Azure VM.

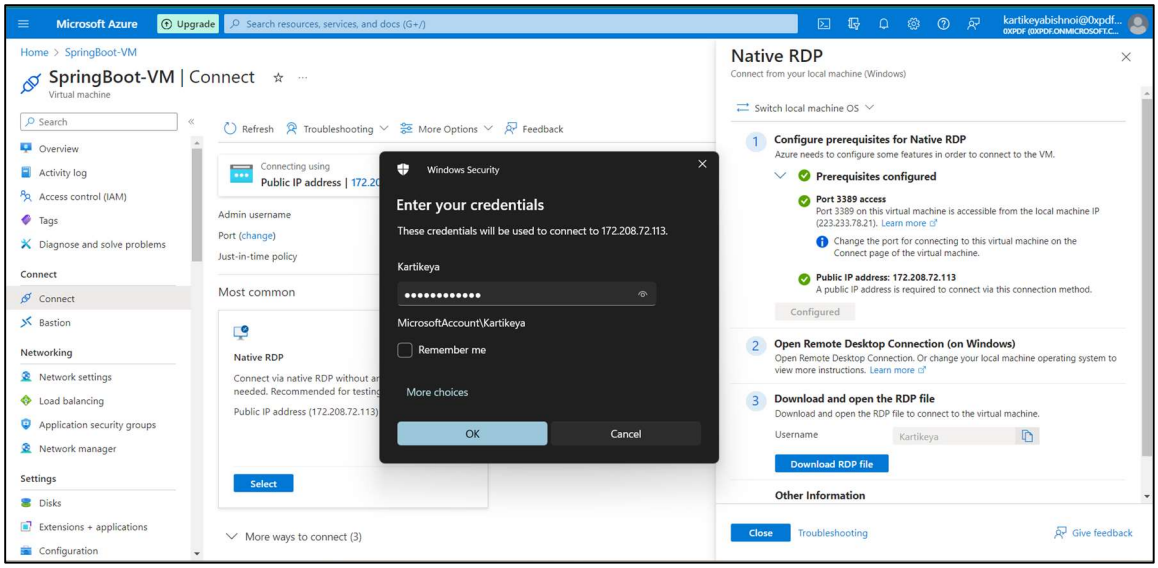


Figure 9 Use RDP to connect to VM

Step 2: Clone Spring Boot Project

- Use **GitHub** to clone your Spring Boot project from GitHub.

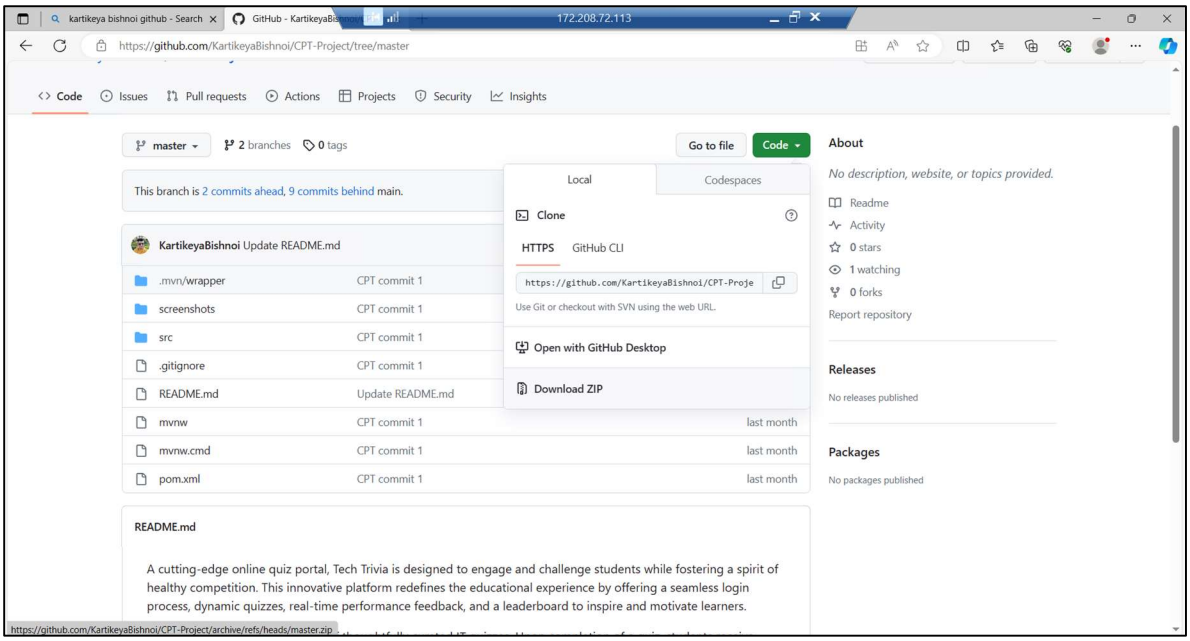


Figure 10 Download zip file from GitHub

Step 3: Set up MySQL Database

- Open MySQL Workbench and connect to the MySQL server.
- Create a database for your Spring Boot application.

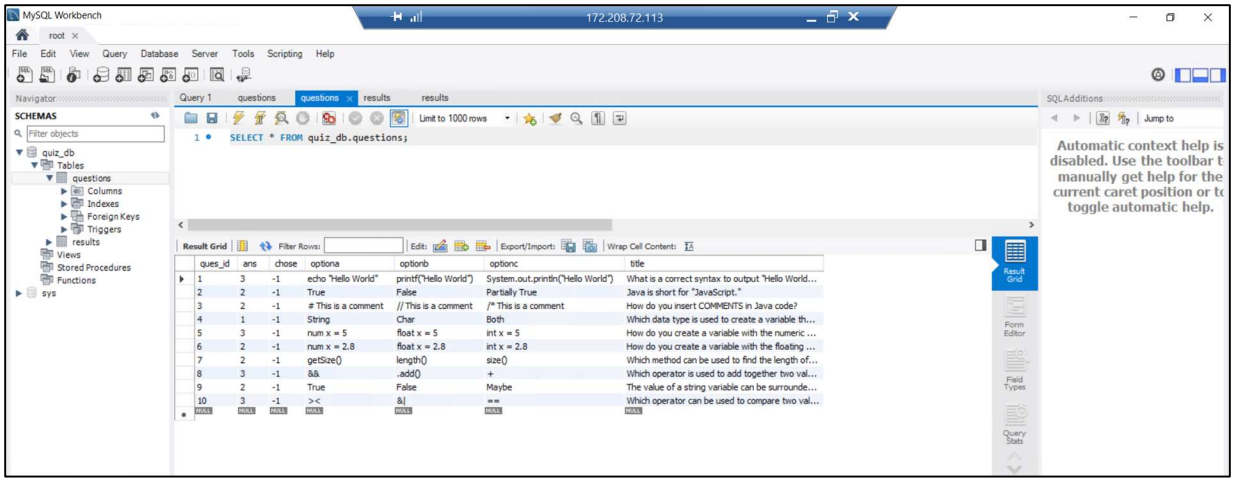


Figure 11 Setup database tables in Workbench

Step 4: Configure Spring Boot Application

- Open your Spring Boot project in IntelliJ IDEA.

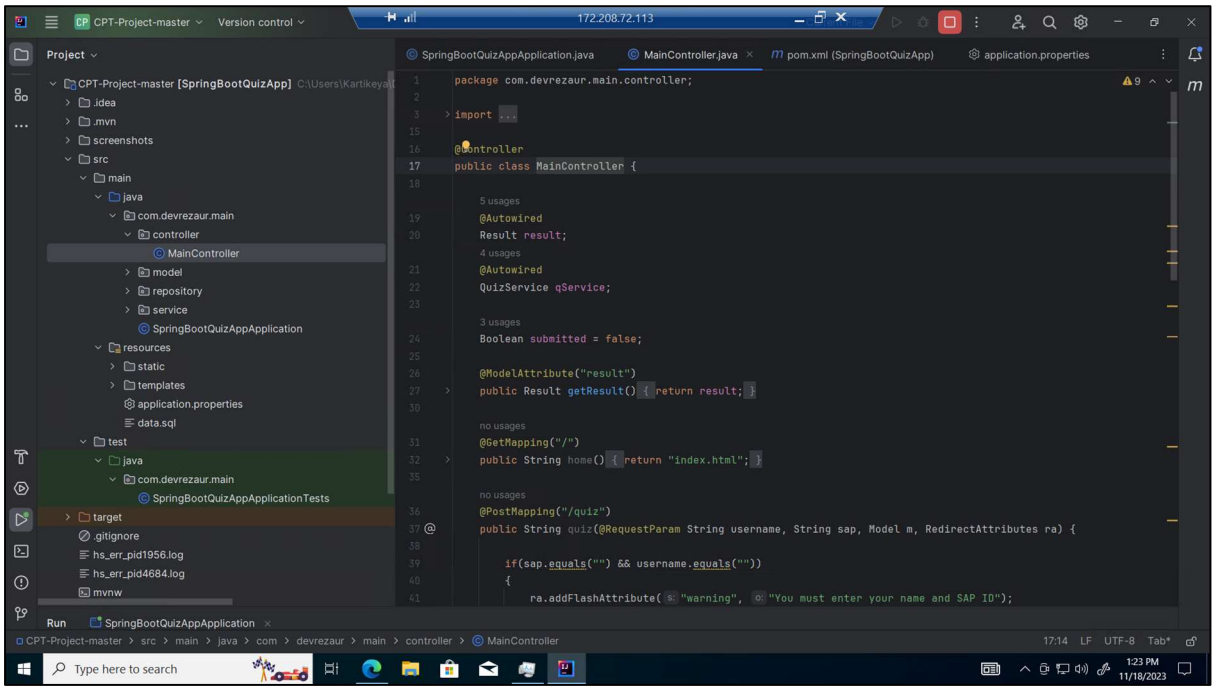


Figure 12 Open project in IntelliJ IDEA

Step 5: Build the Spring Boot Application

- Build your Spring Boot application using IntelliJ or Maven.

Step 6: Transfer JAR File to VM

- Use SCP or any preferred method to transfer the JAR file to the Azure VM.

Step 7: Run Spring Boot Application

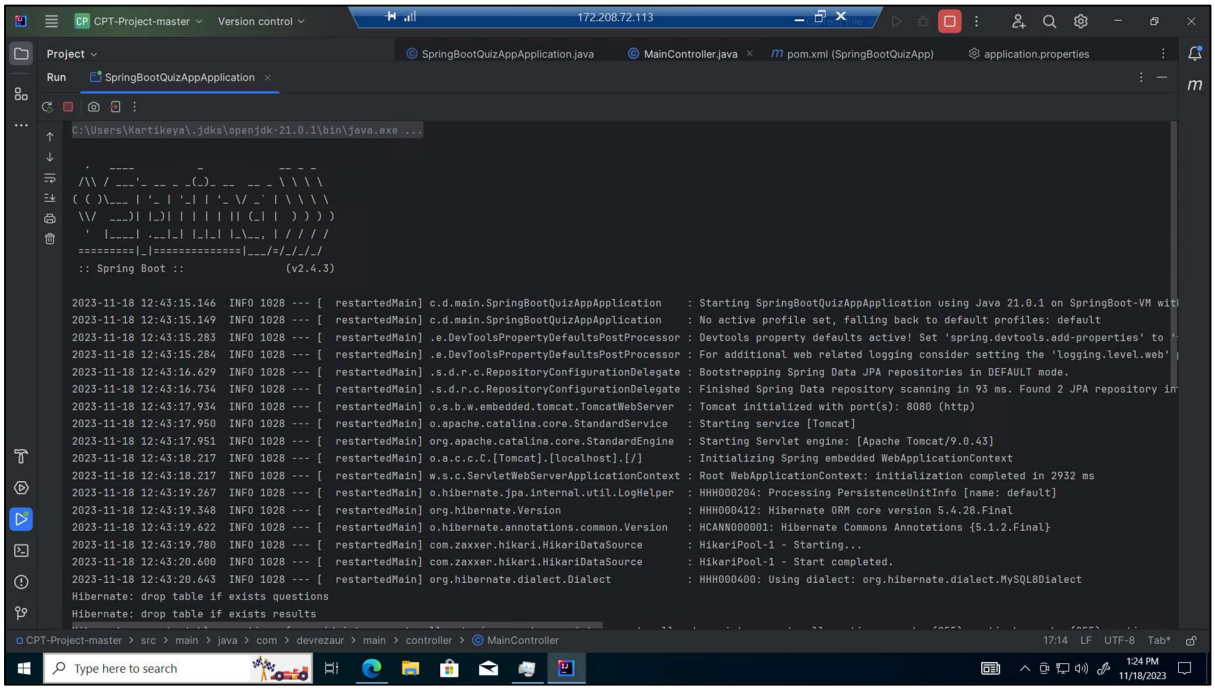


Figure 13 Run the application

Step 8: Access the Application

- Open a web browser and navigate to **http://172.208.72.113:8080/** to access the Spring Boot application.

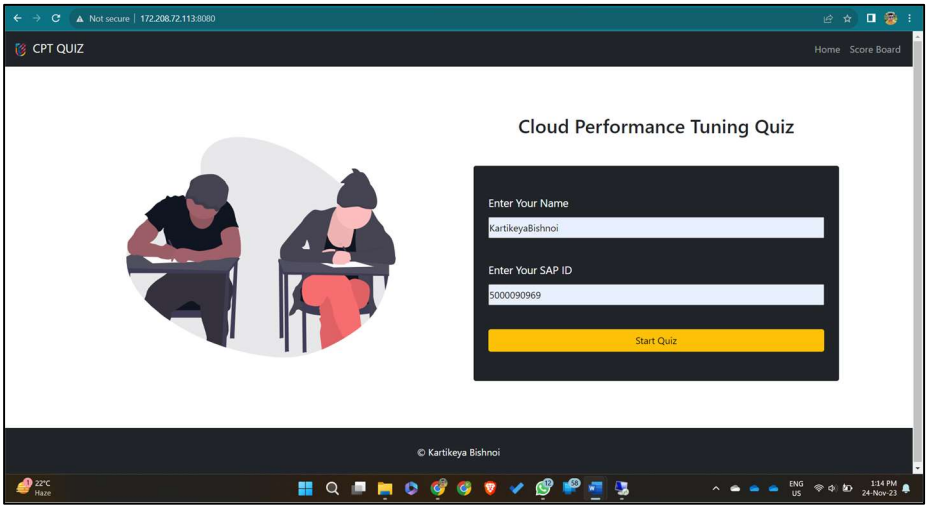
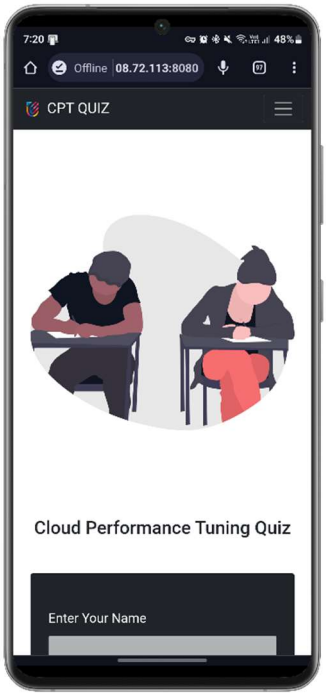
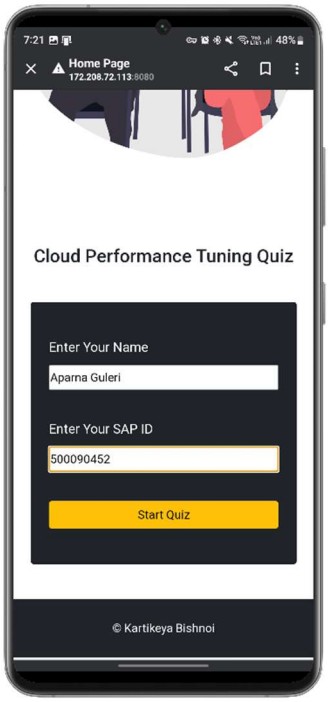
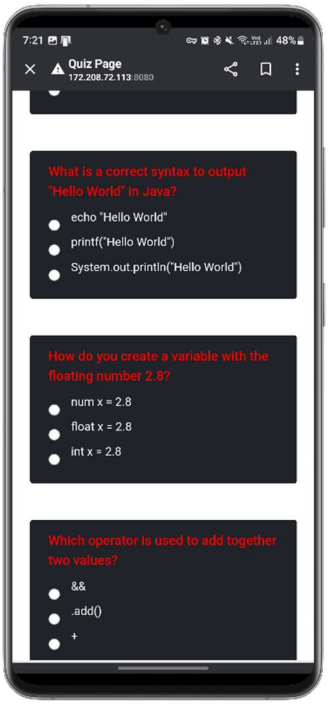
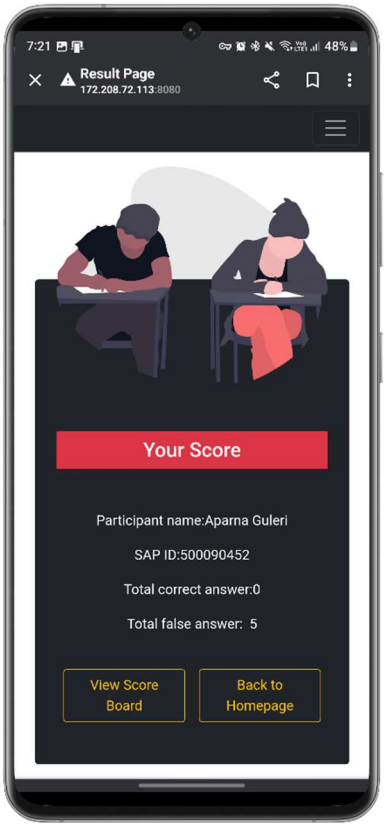


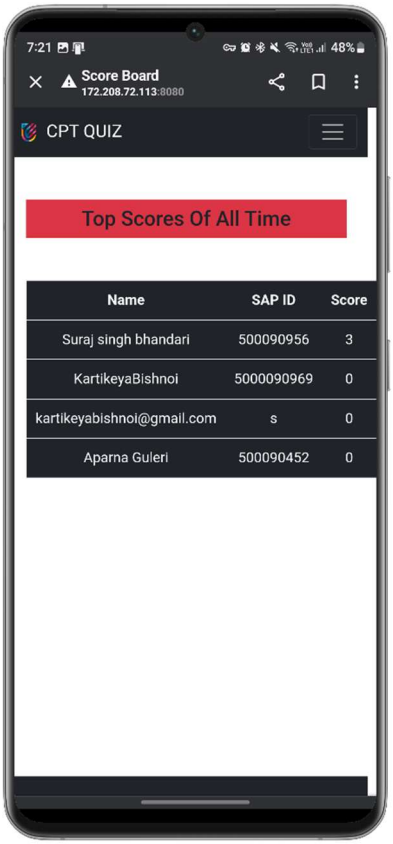
Figure 14 Website is accessible from any device


CPT QUIZ
Cloud Performance Tuning Quiz
Enter Your Name


Home Page
Cloud Performance Tuning Quiz
Enter Your Name
Aparna Guleri
Enter Your SAP ID
500090452
Start Quiz
© Kartikeya Bishnoi


Quiz Page
What is a correct syntax to output "Hello World" in Java?
echo "Hello World"
printf("Hello World")
System.out.println("Hello World")
How do you create a variable with the floating number 2.8?
num x = 2.8
float x = 2.8
int x = 2.8
Which operator is used to add together two values?
&&
.add()
+


Result Page
Your Score
Participant name:Aparna Guleri
SAP ID:500090452
Total correct answer:0
Total false answer: 5
View Score Board
Back to Homepage


Score Board
CPT QUIZ
Top Scores Of All Time
Name
SAP ID
Score
Suraj singh bhandari
500090956
3
KartikeyaBishnoi
5000090969
0
kartikeyabishnoi@gmail.com
s
0
Aparna Guleri
500090452
0

Step 9: Monitor Application Logs

- check the Azure VM logs

Figure 15 VM metrics during and after required tools setup

Methodology

Theoretical framework – explains the model or the set of theories related to the CPT.

1. **Scaling:**
    - Implement both vertical scaling (resizing VMs) and horizontal scaling (adding more VM instances) based on traffic patterns.
    - Use Azure Auto-scaling to automate the scaling process.
2. **Azure CDN (Content Delivery Network):**
    - Leverage Azure CDN to distribute static content globally and reduce latency for end-users.
    - Configure caching policies for static assets.

3. **Network Optimization:**
    - Optimize network configuration for low latency, considering the use of Azure Traffic Manager for global load balancing.
4. **Caching:**
    - Implement effective caching strategies, utilizing both server-side and client-side caching.
    - Consider using Azure Redis Cache or Azure Cache for Redis for distributed caching.
5. **Database Optimization:**
    - Optimize database configuration, index usage, and queries to enhance performance.
    - Evaluate the use of Azure Database services for scalability.
6. **Web Server Configuration:**
    - Fine-tune web server settings (e.g., IIS or Nginx) for optimal connection limits, timeouts, and buffer sizes.
7. **Monitoring, Diagnostics, and Cost Optimization:**
    - Utilize Azure Monitor, Application Insights, and Log Analytics for continuous monitoring and diagnostics.
    - Set up detailed logging and diagnostics to quickly identify and address performance issues.
    - Implement cost tracking and optimization strategies, such as utilizing reserved instances, rightsizing VMs, and leveraging Azure Cost Management to monitor and manage expenses.

	Scalability	Elasticity	Agility	Fault Tolerance	Disaster Recovery	Availability
Hardware	✓		✓	✓	✓	✓
Software	✓	✓	✓	✓	✓	✓
Network			✓			✓
Storage	✓	✓		✓	✓	
Analysis			✓			

Figure 16 Performance Tuning Achieved

SCALING

Scaling in Azure can be achieved through both vertical scaling (resizing VMs) and horizontal scaling (adding more VM instances).

Vertical Scaling:

Resize Virtual Machines:

- In the Azure Portal, navigate to the Virtual Machine you want to scale.
- Stop the VM.
- Under Settings, select "Size" and choose a larger VM size.
- Start the VM.

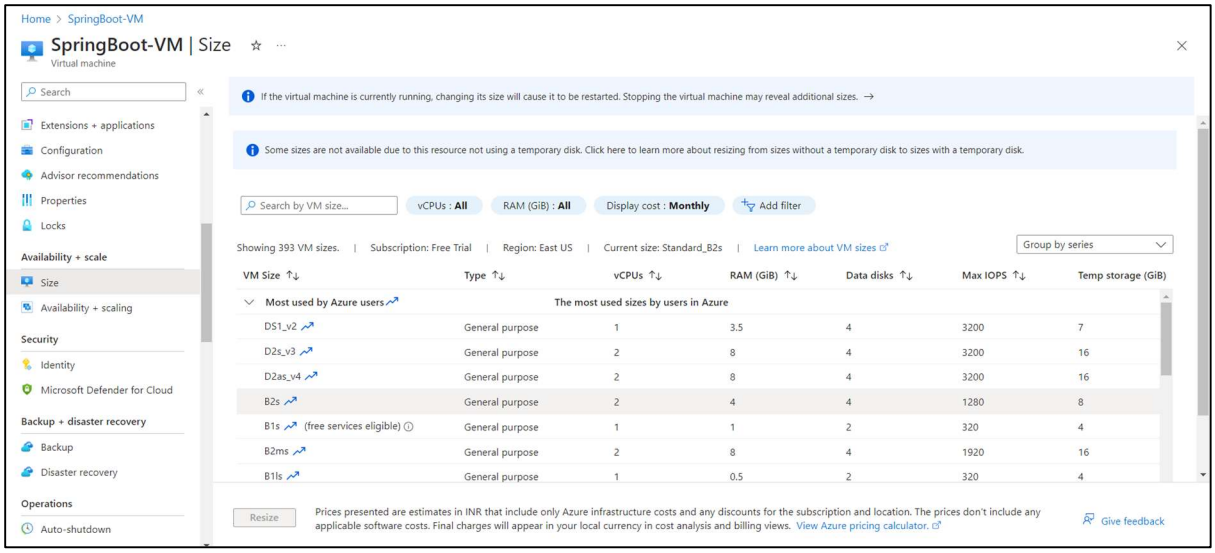


Figure 17 Avaliable options for vertical scaling

Figure 18 VM metrics before Scaling

Figure 19 VM metrics after Scaling

Horizontal Scaling:

Azure Auto-scaling:

Azure Virtual Machine Scale Sets:

Azure Virtual Machine Scale Sets (VMSS) in Microsoft Azure enables automatic scaling of identical VMs, providing high availability and load distribution. VMSS allows you to deploy and manage a set of VM instances, automatically adjusting the number of instances based on demand or defined metrics. This ensures optimal performance and responsiveness, particularly in dynamic or fluctuating workloads. With VMSS, applications can seamlessly scale horizontally, distributing traffic evenly and improving overall reliability. It simplifies management, enhances fault tolerance, and supports efficient resource utilization, making it a powerful tool for deploying and maintaining scalable applications in the Azure cloud environment.

Scaling Azure Virtual Machine

a) Go to Azure Marketplace and type in the Virtual Machine scale set. Then click on Create.

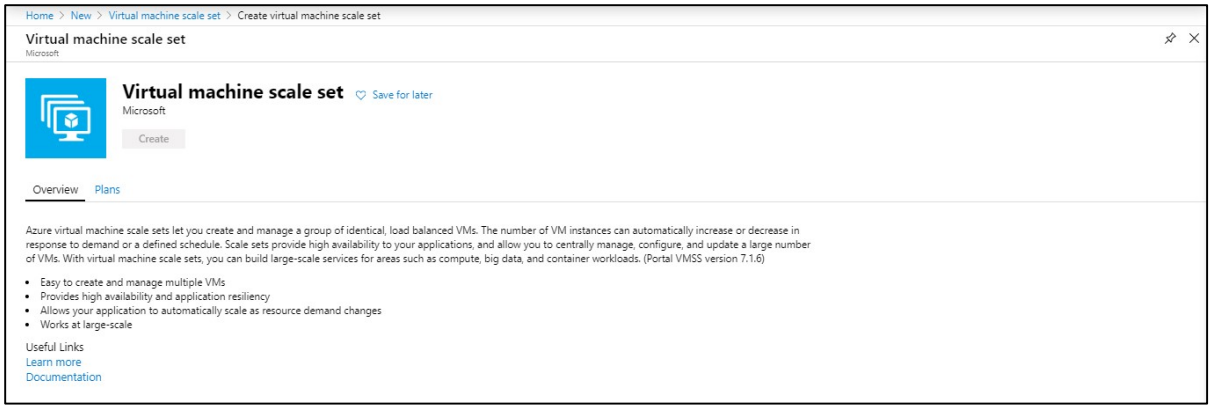


Figure 20 Virtual Machine Scale Set

b) Give a name to this scale set and fill all the other required details. Then click on create.

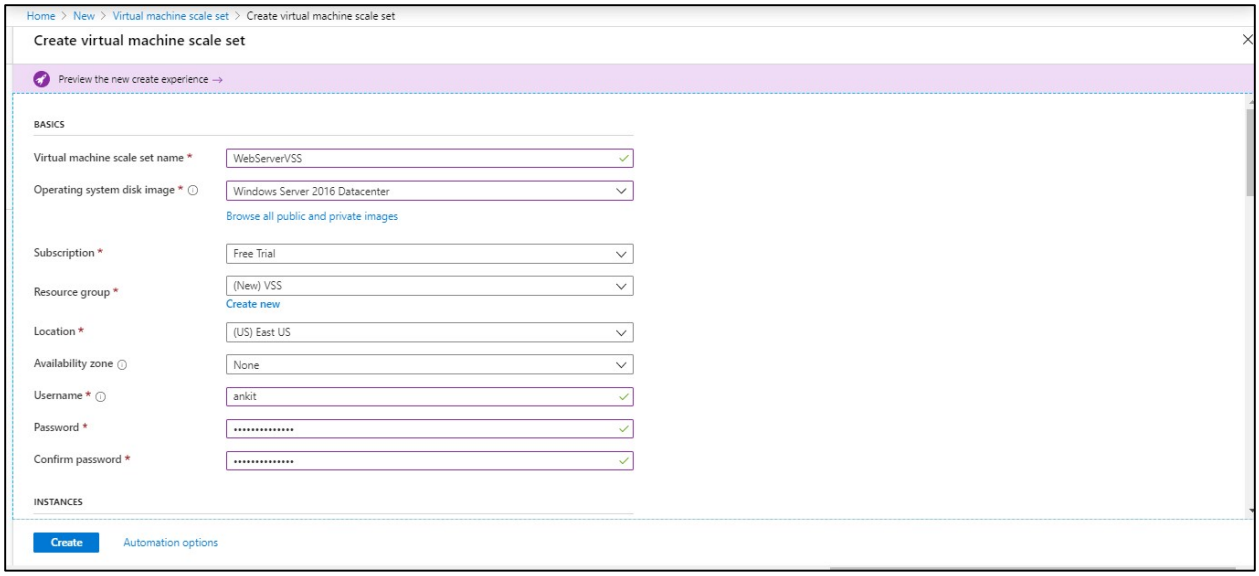


Figure 21 VMSS Setup

c) Now, the Virtual Machine scale set is successfully deployed. To view VMSS go to resources.

d) Now, click on scaling. Provide an auto-scale setting name. And select the resource group.

e) Scroll Down, and you will find two ways to auto-scale. First, click on "add a rule? for auto-scaling based on the metric. We are going to scale our virtual machine if the average percentage of CPU utilization is above 70 percent.

f) Now, select the time and date-based scaling, where you can scale when you need more space. And the last thing is Notify, where you get notified whenever the auto-scaling gets triggered.

Load Balancing:

The load balancer is used to distribute the incoming traffic to the pool of virtual machines. It stops routing the traffic to a failed virtual machine in the pool. In this way, we can make our application resilient to any software or hardware failures in that pool of virtual machines.

Creating Azure Load Balancer

1. Go to the Azure portal, and click on create a Resource. After that, type-in Load Balancer, and click on it.

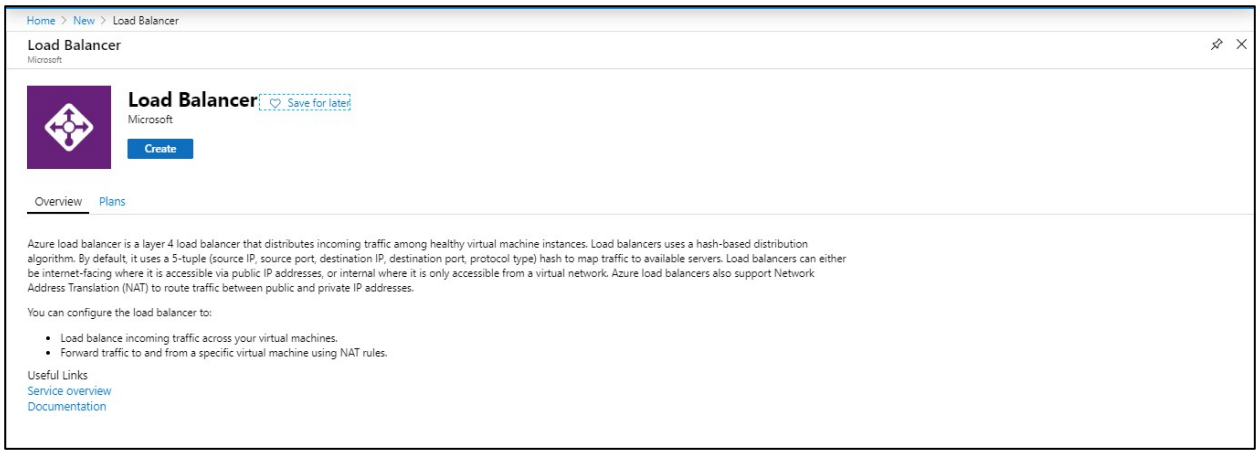


Figure 22 Azure Load Balancer

2. You are now on the Load Balancer creation page. Fill all the required details as the figure below. And click on review+create.

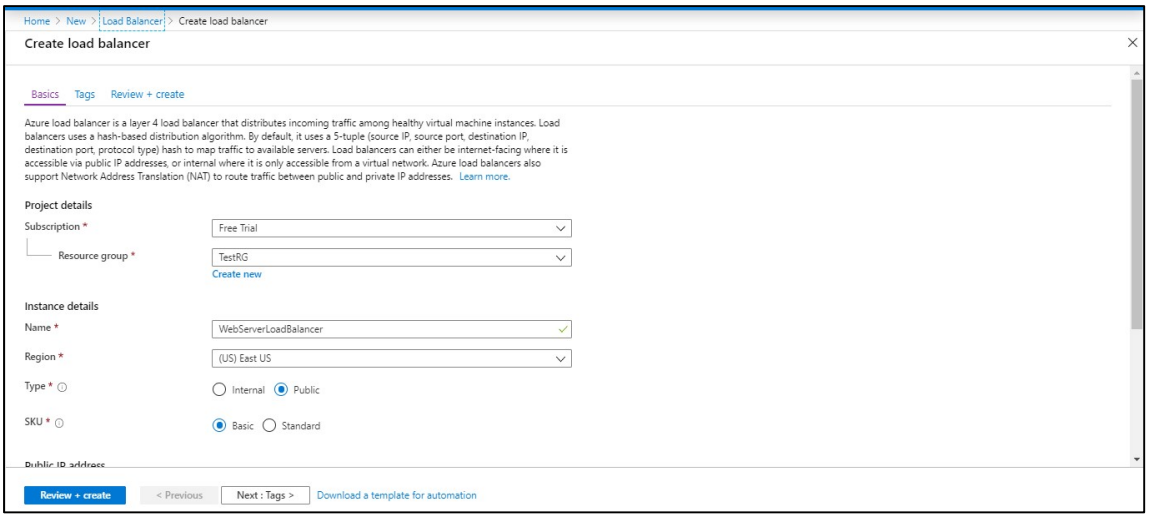


Figure 23 Azure Load Balancer setup

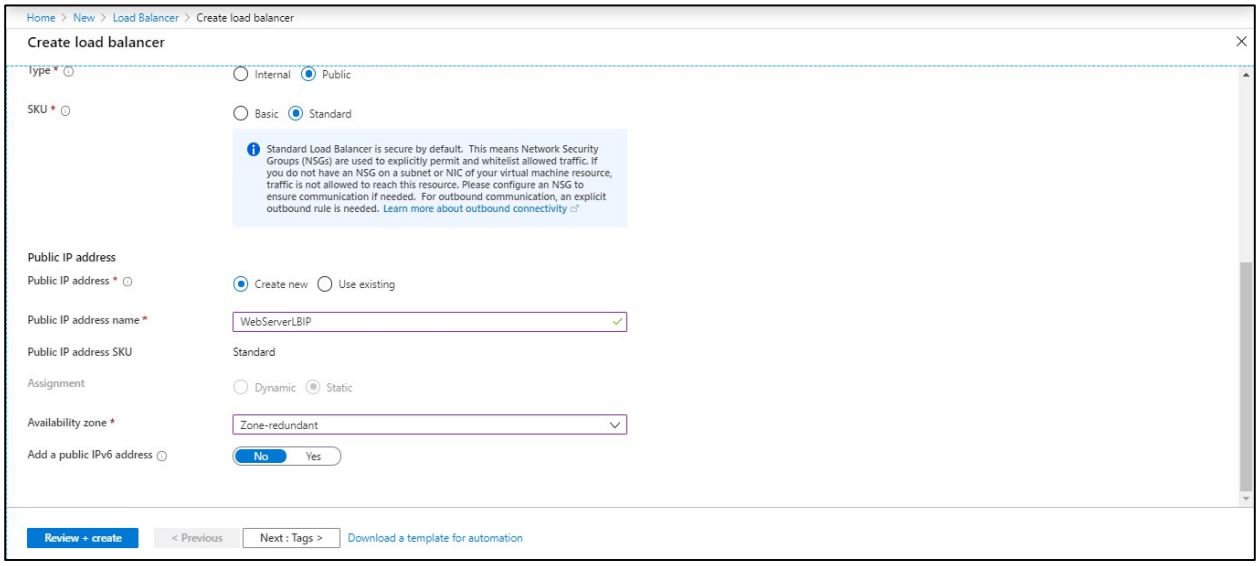


Figure 24 Load Balancer configuration

3. You will be redirected to the review page. Check all the details and click on Create.

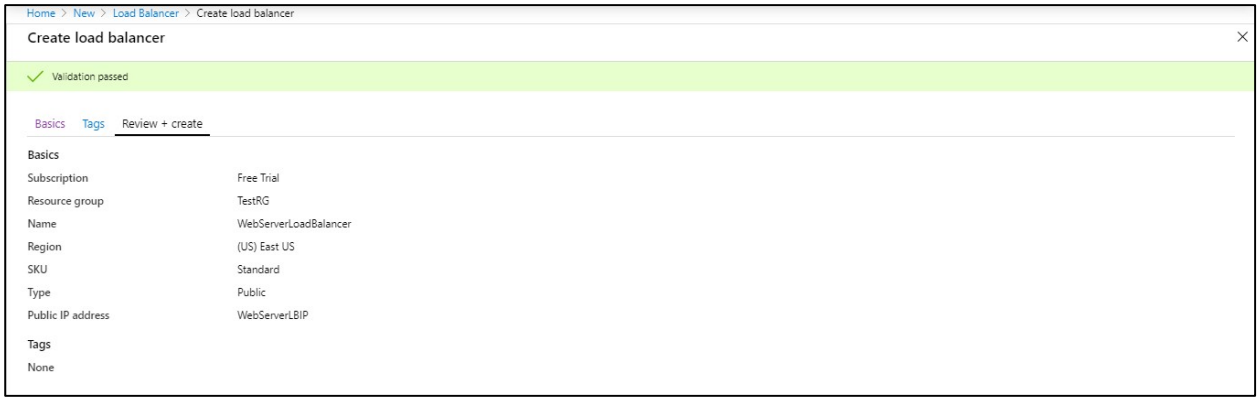


Figure 25 Load Balancer review

The Load Balancer is now created.

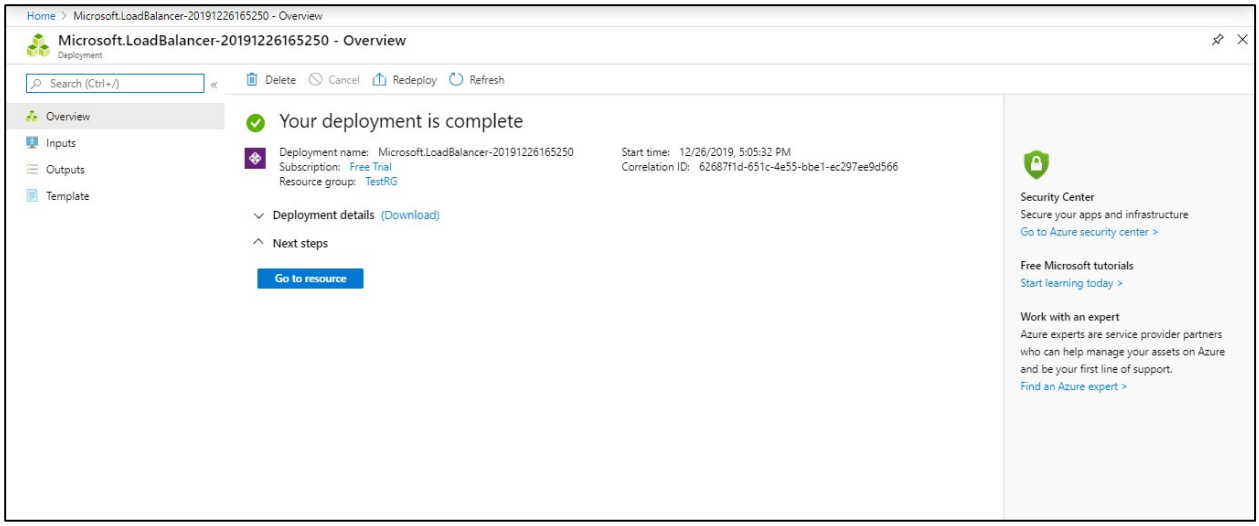


Figure 26Load Balancer setup complete

Code optimization.

1. Algorithm and Data Structure Optimization:

Algorithms and data structures are carefully reviewed and selected tailored to the unique requirements of the quiz platform. This ensures optimal time and space complexity for critical operations, enhancing overall performance.

2. Database Query Optimization:

Database queries were thoroughly examined and optimized to minimize the number of interactions with the database. Indexing strategies were implemented judiciously, and queries were fine-tuned to retrieve only the essential data, reducing database load.

4. Code Profiling:

Profiling tools were employed to identify performance bottlenecks and hotspots in the codebase. This allowed us to pinpoint areas of improvement, guiding our optimization efforts effectively.

5. Asynchronous Programming:

Asynchronous programming techniques were implemented, particularly for I/O-bound operations. This prevents the application from being blocked during resource-intensive tasks, contributing to better responsiveness.

6. Frontend Optimization:

Frontend assets, including CSS, JavaScript, and images, underwent minification and compression.

8. Error Handling:

error handling processes is optimized by streamlining try-catch blocks and handling errors gracefully. This not only improves performance but also ensures a more robust and reliable application.

9. Testing:

Thorough testing was conducted to assess the impact of each optimization on performance. This rigorous approach allowed validation for the effectiveness of the efforts made and identify areas for further refinement.

Results Of Code Optimization:

As a result of these optimizations, the Online Quiz Platform has experienced notable improvements in responsiveness, resource utilization, and overall user experience.

Software Agility

Achieving software agility in Azure involves adopting agile development methodologies and leveraging Azure services and tools to support agile practices. By adopting Azure DevOps, Continuous Integration and Continuous Deployment (CI/CD), Infrastructure as Code (IaC), Microservices Architecture, and promoting collaboration and communication, organizations can create an agile development environment. This environment facilitates rapid and reliable software delivery, fostering a culture of continuous improvement and adaptation.

1. Azure DevOps:

Azure DevOps provides a comprehensive set of development tools, encompassing version control, build automation, and release management. Leveraging Azure Boards for work tracking and project management, Azure Repos for version control, and Azure Pipelines for CI/CD, teams can streamline their development processes.

2. Continuous Integration and Continuous Deployment (CI/CD):

Setting up CI/CD pipelines in Azure Pipelines is essential for automating the building, testing, and deployment of applications. Automation extends to deploying applications to Azure services such as Azure App Service, Azure Kubernetes Service (AKS), or Azure Functions, ensuring a seamless and efficient deployment process.

3. Infrastructure as Code (IaC):

The use of Azure Resource Manager (ARM) templates or tools like Terraform allows organizations to define and manage infrastructure as code. This approach enables version

control of infrastructure, ensures predictability in making changes, and automates deployment processes, contributing to a more agile and efficient development lifecycle.

4. Microservices Architecture:

Breaking down monolithic applications into microservices is a crucial step towards achieving agility. Azure offers various services for managing microservices, including Azure Service Fabric, Azure Kubernetes Service (AKS), and Azure Functions. These services provide scalability, flexibility, and ease of management for microservices-based applications.

5. Collaboration and Communication:

Effective collaboration and communication are pivotal for Agile development. Tools like Microsoft Teams and Azure DevOps facilitate seamless communication and collaboration within development teams. Encouraging regular feedback and communication fosters a culture of continuous improvement and responsiveness.

Software Disaster Management

Steps to Back Up an Azure VM:

1. **Azure Backup Service:**
    - Ensure that the Azure subscription includes the Azure Backup service.
    - Navigate to the Azure and select the appropriate subscription and resource group.
2. **Enable Backup for VM:**
    - In the Azure portal, go to the specific virtual machine you want to back up.
    - Under the VM settings, select "Backup" and then click on "Backup" in the Backup pane.

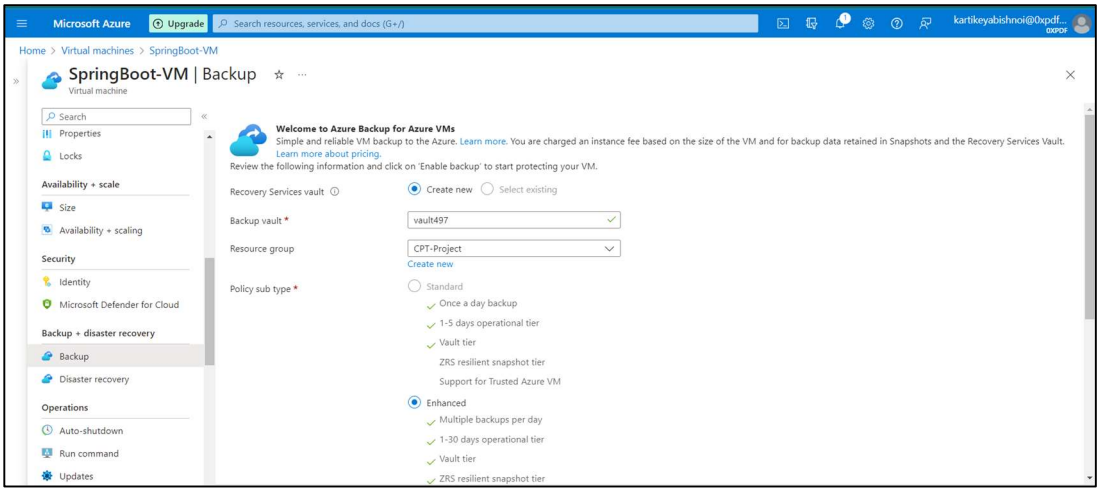


Figure 27 Azure VM Backup

3. **Backup Policy:**
    - Create or select a backup policy that defines the backup frequency, retention period, and other settings.
    - Configure the desired backup schedule based on your recovery point objectives.
4. **Backup Vault:**
    - Choose an Azure Backup vault to store your backup data. If you don't have one, create a new vault in the same region as your VM.
5. **Review Configuration:**
    - Review the backup configuration settings to ensure they align with your backup strategy and compliance requirements.
6. **Initiate Backup:**
    - Once the configuration is reviewed and finalized, initiate the backup process. This may involve creating an initial backup and scheduling subsequent backups based on your policy.

Disaster Recovery Setup:

1. **Prepare Source and Target Environments:**
    - Make sure the target Azure region and resource group are ready to receive the replicated VM during failover.

- Ensure that the Azure VMs in the source and target environments have the same configurations.

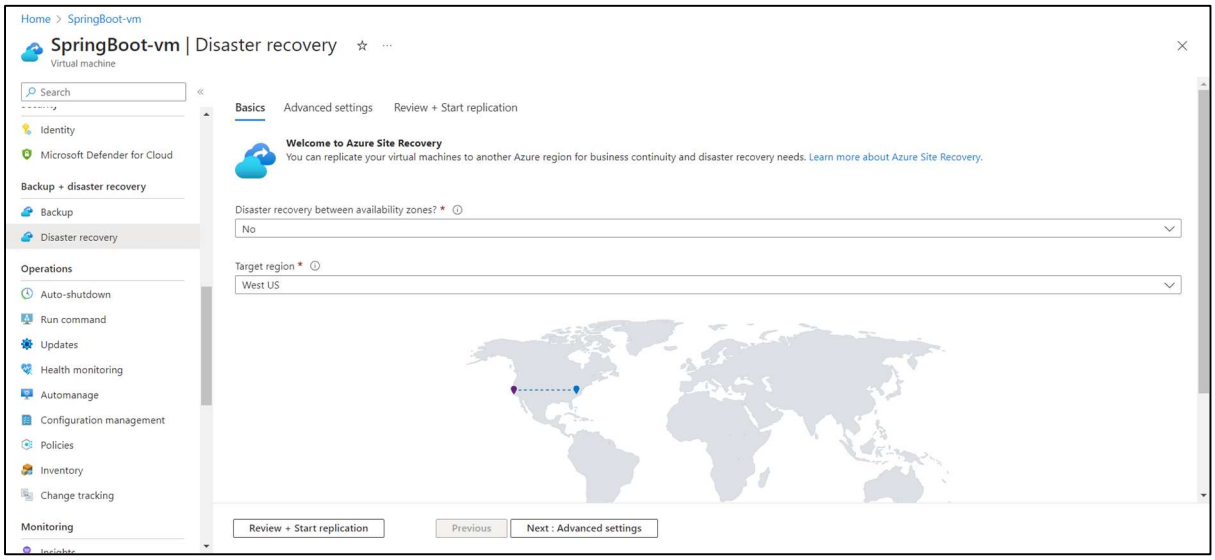


Figure 28 Select Region for replication

2. **Create a Recovery Plan:**

- In the Azure portal, navigate to the Azure Site Recovery vault.
- Under "Replicated Items," select the VM for which you want to create a recovery plan.
- Create a recovery plan, which is a set of predefined steps to failover and failback your VM.

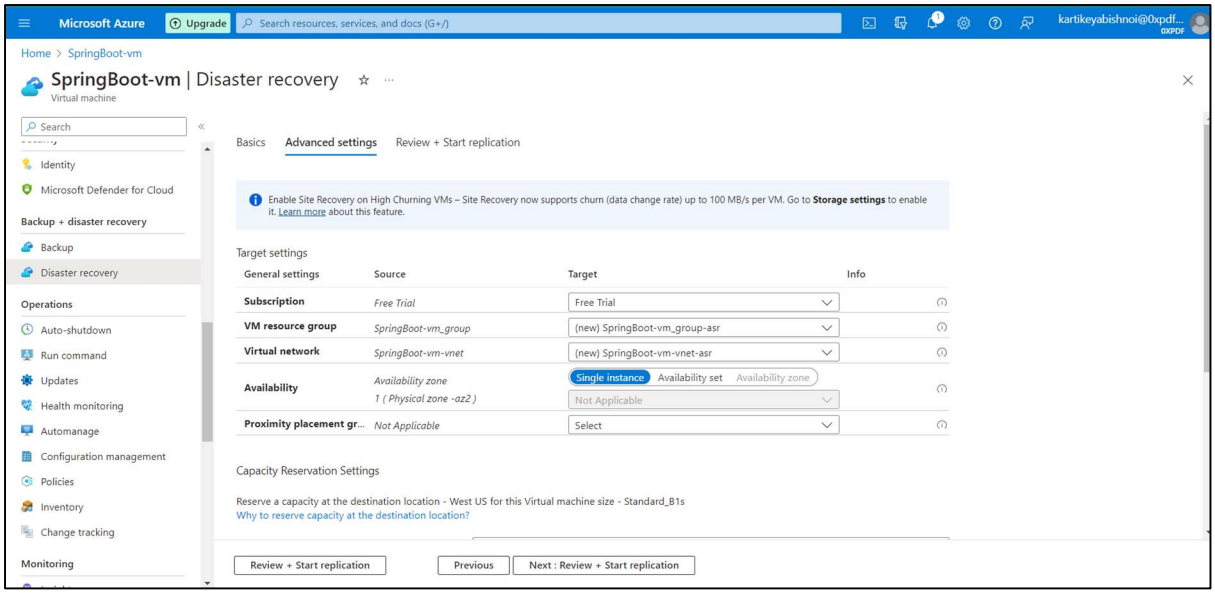


Figure 29 Azure VM Disaster Recovery

3. **Configure Recovery Settings:**

    - Configure recovery settings, including the recovery point objective (RPO) and the desired recovery point.

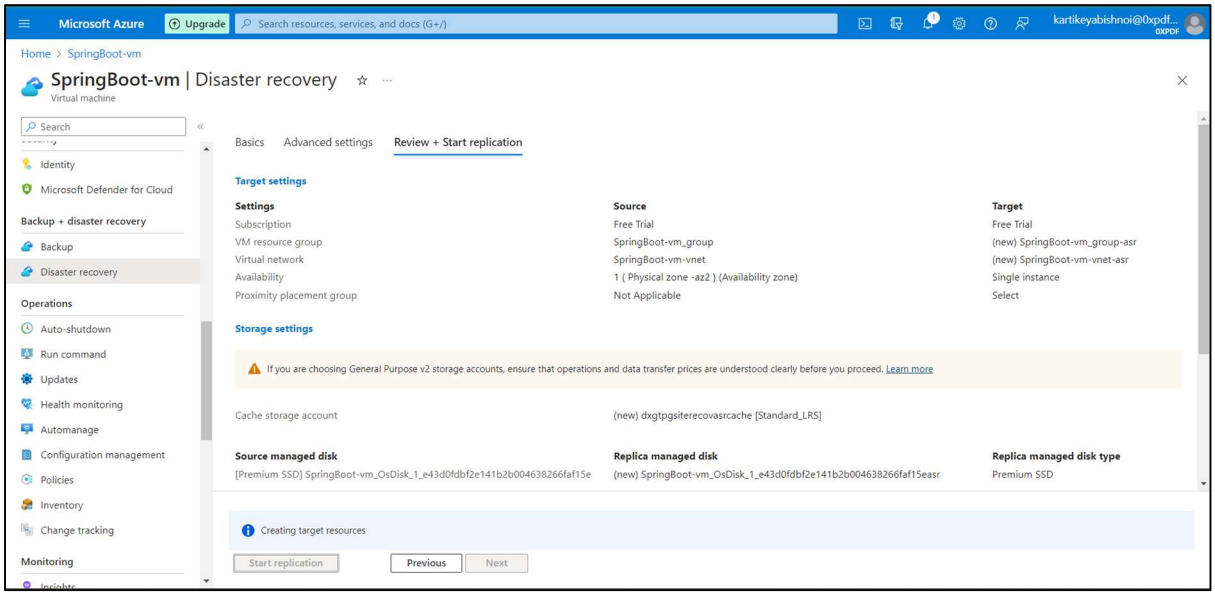


Figure 30 Policy review

Azure Network Manager

We must make sure that our application is highly available and resilient to regional failures, data centre failure, and rack failures. Azure provides some services to make our application highly available; these are:

Azure Virtual Network Manager (VNM) is a centralized management solution that simplifies the management of virtual networks across multiple Azure subscriptions and regions. It provides a unified view of your virtual network infrastructure, enabling you to:

- Group and manage virtual networks at scale: VNM allows you to organize your virtual networks into network groups, which are collections of virtual networks that share common connectivity or security requirements. This makes it easier to manage and apply configurations to large numbers of virtual networks.
- Define and enforce network security policies: VNM simplifies the management of network security policies by allowing you to define and apply security rules at the network group level. This ensures that all virtual networks in a network group adhere to the same security policies.

- Automate network deployments and changes: VNM can automatically deploy and update network configurations, reducing the risk of errors and ensuring consistency across your environment.
- Monitor and troubleshoot network issues: VNM provides visibility into the health and performance of your virtual networks, making it easier to identify and resolve network problems.

Key benefits of using Azure Virtual Network Manager:

- Reduced complexity: VNM simplifies the management of complex virtual network environments, making it easier to manage and control your network infrastructure.
- Improved security: VNM helps to enforce consistent security policies across your virtual networks, reducing the risk of security breaches.
- Increased agility: VNM enables you to deploy and update network configurations quickly and easily, allowing you to respond to changing business needs.
- Enhanced visibility: VNM provides comprehensive visibility into the health and performance of your virtual networks, making it easier to identify and resolve network problems.

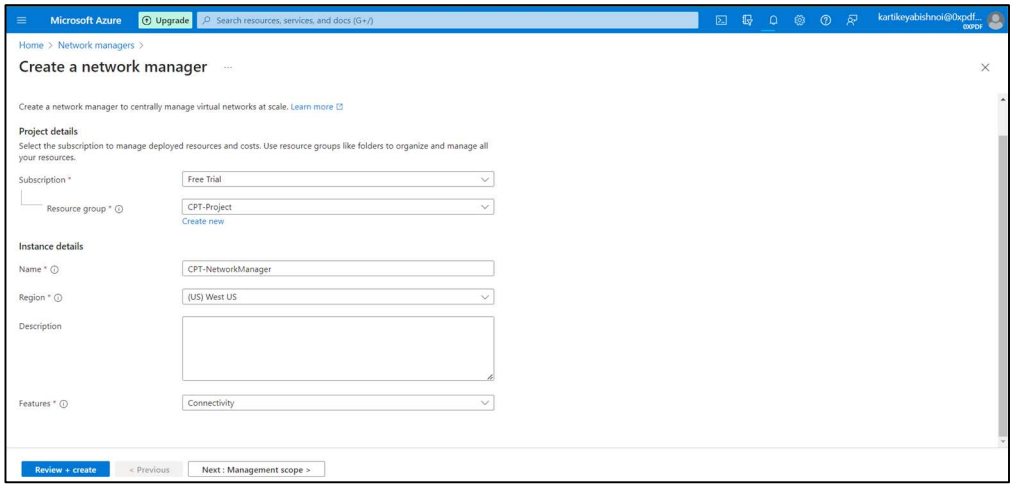


Figure 31 Azure Network Manager creation

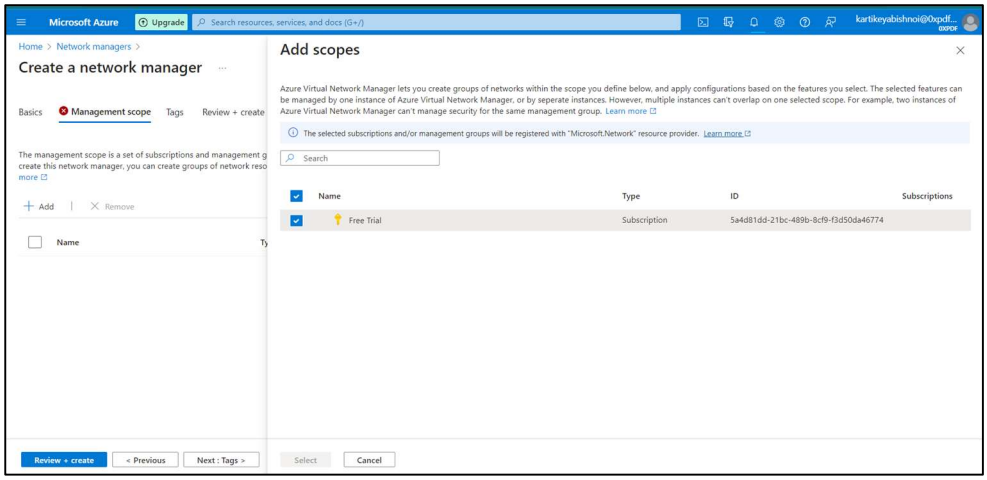


Figure 32 Add Scopes

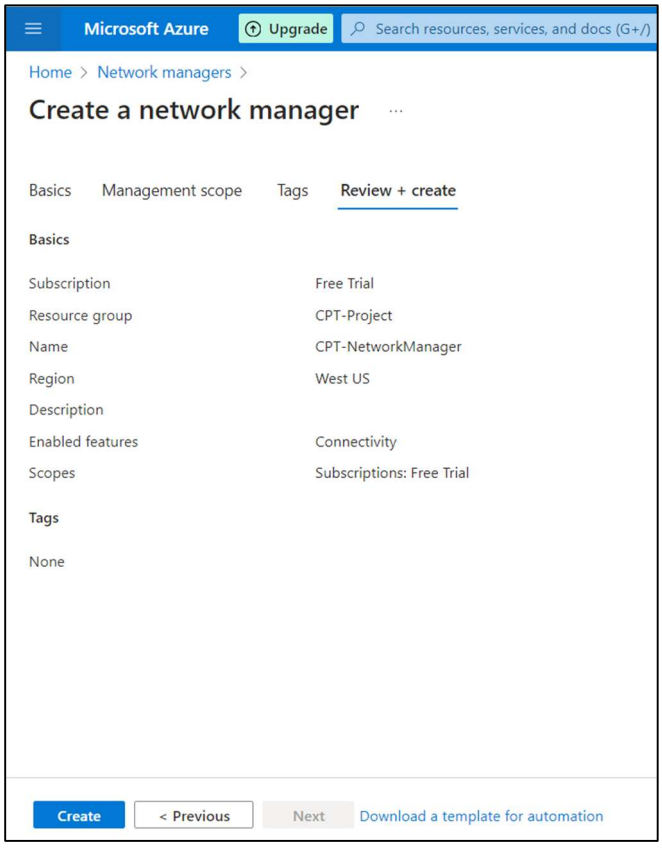


Figure 33 Review and create

Database Optimization

This report details the database optimization efforts undertaken for the [Your Quiz WebApp Database] using MySQL Workbench. The optimization initiative aimed to enhance query performance, reduce latency, and improve overall efficiency. By analysing and fine-tuning the database schema, indexing strategies, and query execution plans, significant improvements were achieved in the responsiveness and reliability of the database.

A well-optimized database is essential for the efficient functioning of any data-intensive application, particularly online quiz platforms. This report outlines the steps taken to optimize the [Your Quiz WebApp Database] to ensure optimal performance during varying workloads.

The optimization process encompassed a comprehensive review of the database structure, indexes, and query performance. MySQL Workbench was the primary tool utilized for analysis and implementation. The key areas addressed include:

- **Database Schema Analysis:** A thorough examination of the existing database schema to identify opportunities for normalization, denormalization, and restructuring.
- **Indexing Strategies:** Evaluation and adjustment of indexing strategies to enhance the speed of data retrieval while minimizing the impact on insert and update operations.
- **Query Execution Plans:** Analysis of query execution plans to identify and rectify inefficient query patterns. This involved optimizing complex queries, using appropriate join techniques, and leveraging MySQL's query optimizer.

The database optimization efforts resulted in notable improvements:

- **Reduced Query Execution Time:** Optimized queries demonstrated a significant reduction in execution time, contributing to faster data retrieval for quiz-related operations.
- **Enhanced Indexing:** Strategic adjustments to indexing improved the efficiency of data retrieval without compromising the speed of write operations.
- **Normalized Schema:** Refactoring the database schema led to a more normalized structure, reducing data redundancy and enhancing overall data integrity.

Key query performance metrics were monitored before and after the optimization process. The improvements in query execution times were measured across a representative set of complex and frequently executed queries.

The database optimization initiative for [Your Quiz WebApp Database] using MySQL Workbench has yielded positive results. The improvements in query performance, indexing strategies, and schema normalization contribute to a more responsive and efficient database, ensuring a seamless user experience on the quiz platform.

Stress Testing

Figure 34 During Stress Test

This report presents the findings of a stress testing initiative conducted on Tech Trivia Project. The stress test, lasting 22 minutes, involved the simultaneous access of the web application by 26 bots, utilizing a demo dataset of approximately 500 individuals' names and SAP IDs. A total of 1084 iterations were executed, resulting in 30352 hits to the website. The stress test revealed that the web application, hosted on a Azure virtual machine (VM), efficiently handled the imposed traffic without encountering errors.

1. Introduction: As the user base of Tech Trivia continues to grow, ensuring its robustness under stress conditions becomes paramount. This stress testing initiative aimed to evaluate the platform's performance, scalability, and resource efficiency during peak traffic scenarios.

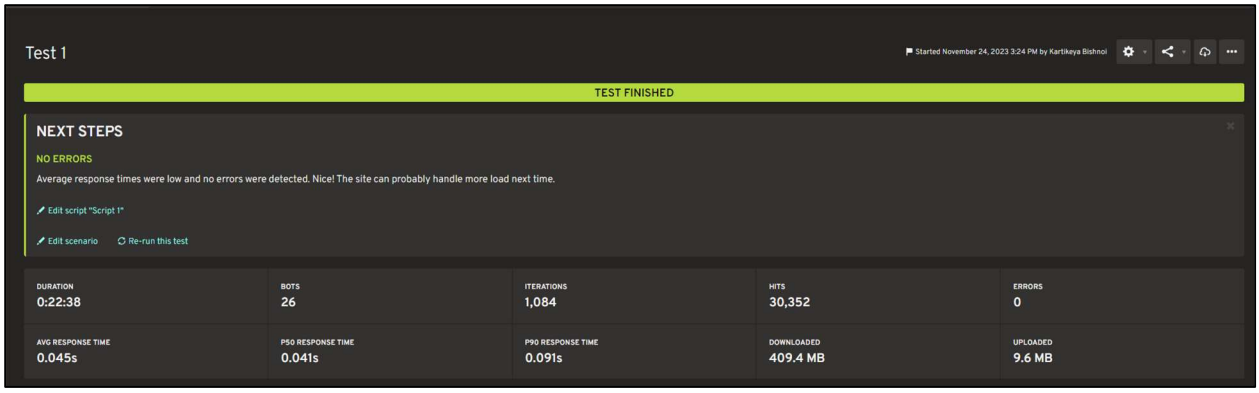


Figure 35 Stress Test Results

2. Methodology:

- **Duration:** The stress test spanned 22 minutes, simulating sustained high traffic.

- **Bot Simulation:** 26 bots were concurrently accessing the web application.
- **Dataset:** A demo dataset of 500 individuals' names and SAP IDs was utilized.
- **Iterations:** A total of 1084 iterations were executed during the stress test.
- **Hits:** The web application received 30352 hits over the testing period.

3. Results:

- **Efficient Handling:** The Azure VM hosting the web application demonstrated robust performance, efficiently managing the simultaneous access of 26 bots.
- **No Errors:** Throughout the stress test, no errors or disruptions were observed. The web application-maintained stability and reliability under the imposed load.

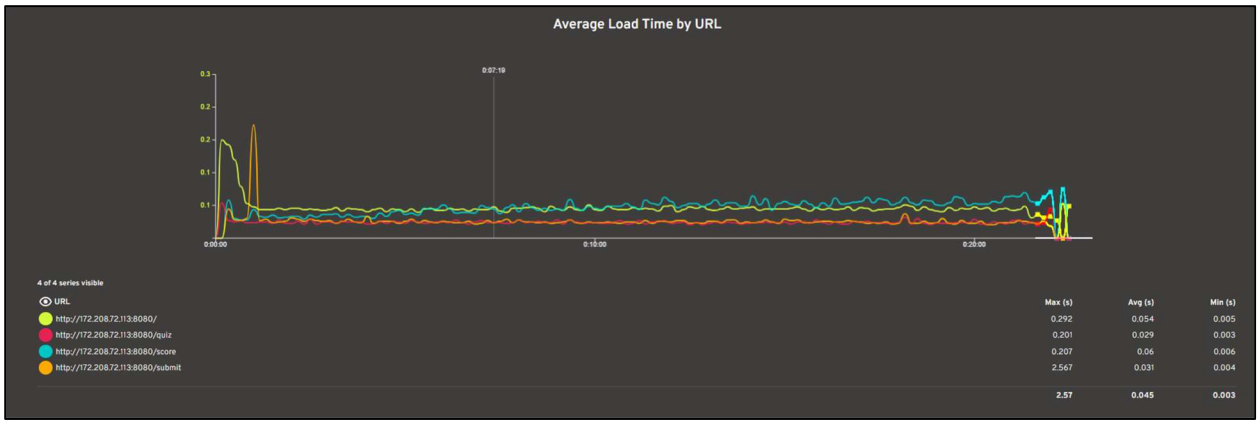


Figure 36 Average load time by URL

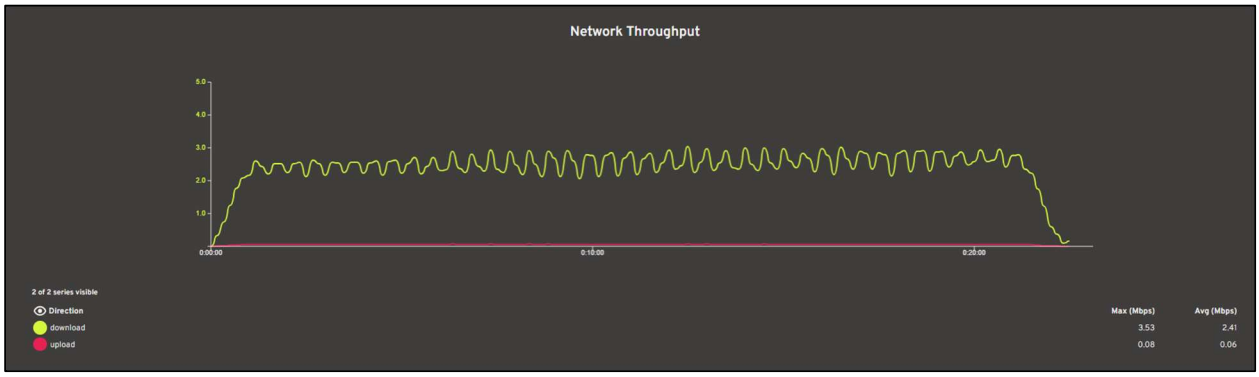


Figure 37 Network Throughput

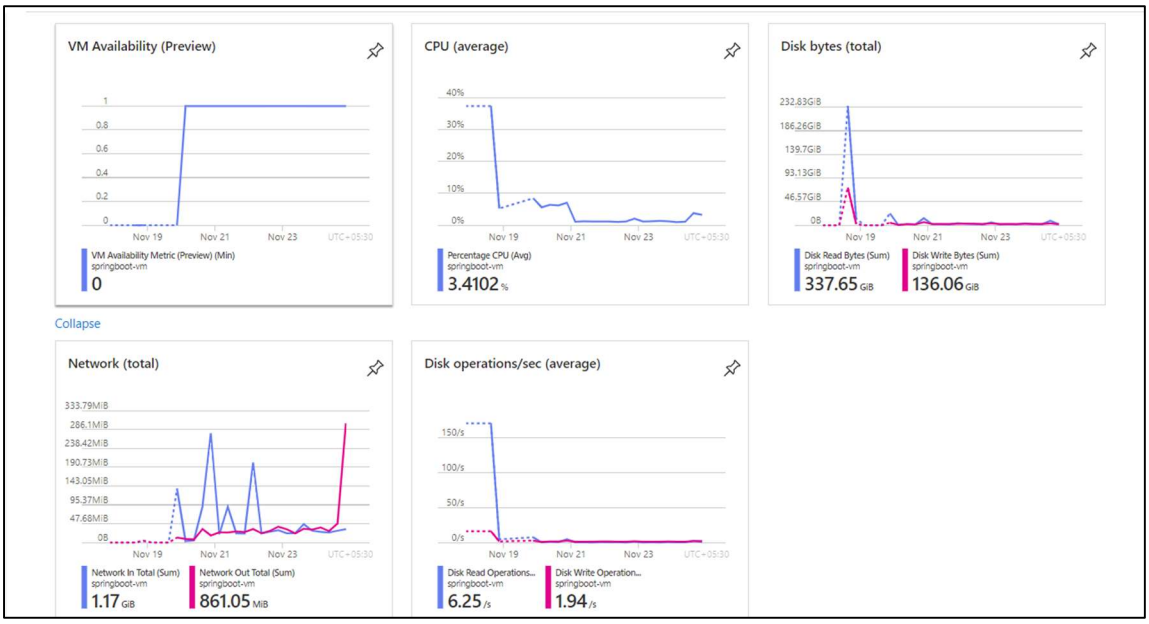


Figure 38 VM metrics before Stress Test

Figure 39 VM metrics after Stress Test

4. Analysis: The stress test results indicate that Tech Trivia is well-prepared to handle a significant volume of concurrent users. The absence of errors and the efficient handling of traffic point to a scalable and resilient architecture. The stress testing methodology, involving a diverse dataset and realistic bot simulation, provides confidence in the platform's ability to meet user demands during periods of high usage.

6. Conclusion: The successful completion of the stress test reaffirms the reliability and scalability of Tech Trivia. The insights gained from this initiative will inform future optimizations and ensure that the platform remains resilient in the face of growing user engagement.

Sources of data – Primary or secondary data:

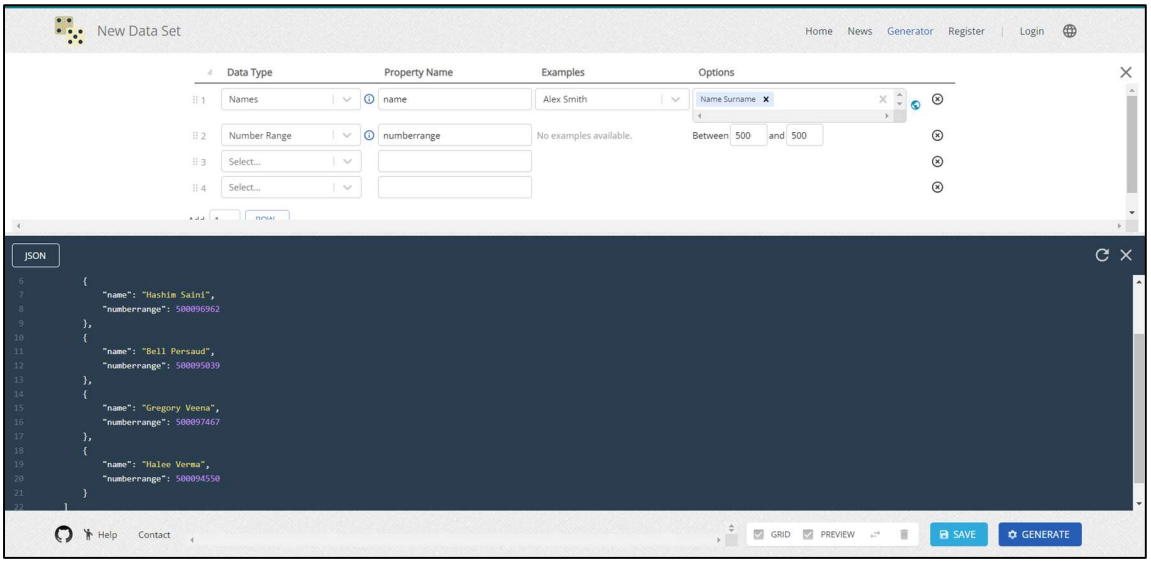


Figure 40 Type of dataset

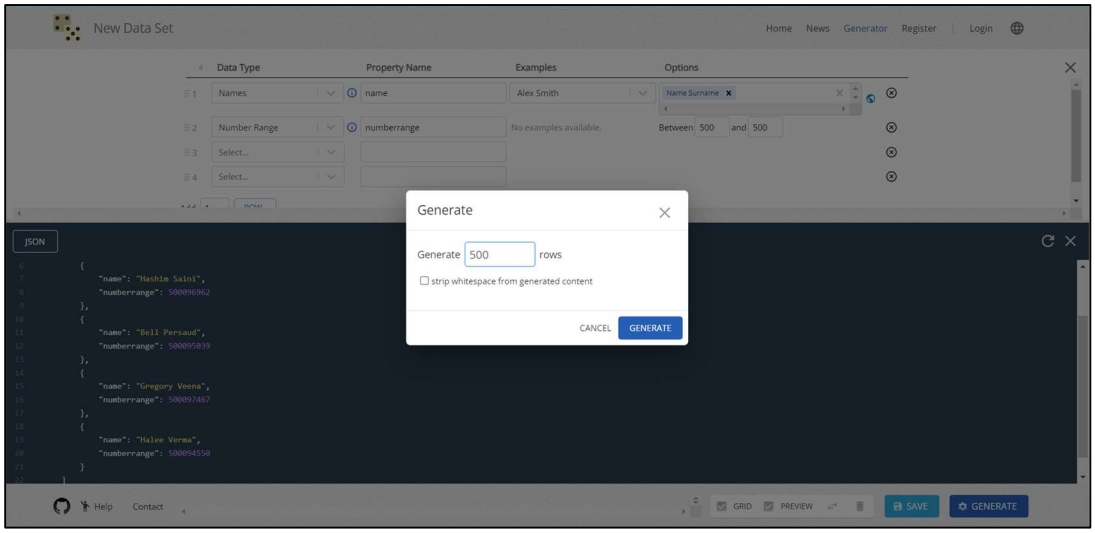


Figure 41 Generate dataset of 500 people.

	A	B	C
1	name	SAP	
2	Wang Narayan	500096960	
3	Alisa Srivastav	500095504	
4	Edward Chandrasekar	500093188	
5	August Nagpal	500093303	
6	Zeus Pal	500095269	
7	Eliana Swami	500094218	
8	Cynthia Sara	500097617	
9	Charde Meena	500096027	
10	Lynn Prasad	500092711	
11	Jolene Srini	500098646	
12	Ramona Malhotra	500095884	
13	Ava Patel	500098079	
14	Malik Subram	500096568	
15	Maya Sen	500095622	
16	Cain Malik	500097265	
17	Adele Srivas	500092215	
18	Samson Chaudhary	500093948	
19	Nasim Sen	500093581	
20	Lucas Ranga	500092170	
21	Chloe Gandhi	500097959	
22	Merrill Soni	500093874	
23	Ulysses Sen	500094094	
24	Cadman Ganesh	500093382	
25	Quentin Sehgal	500092833	
26	Xena Veer	500098697	

convert

Figure 42 Dataset generated.

Schematic flow Diagram:

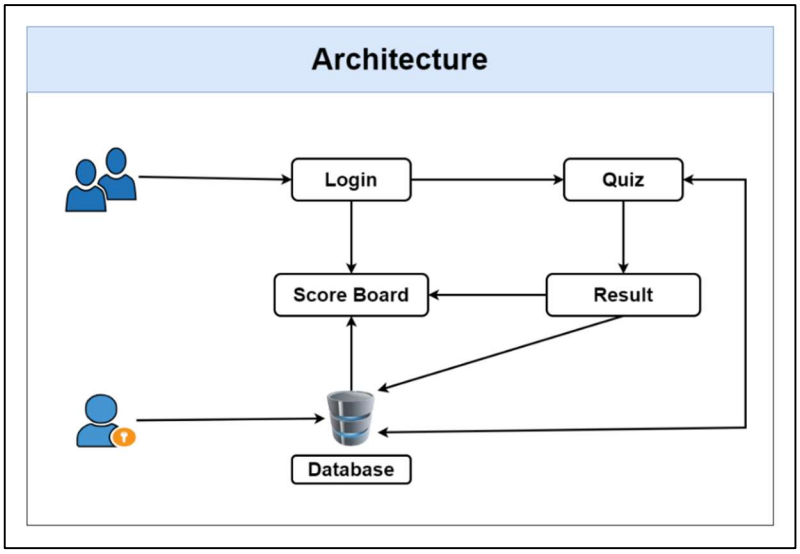


Figure 43 Project Architecture

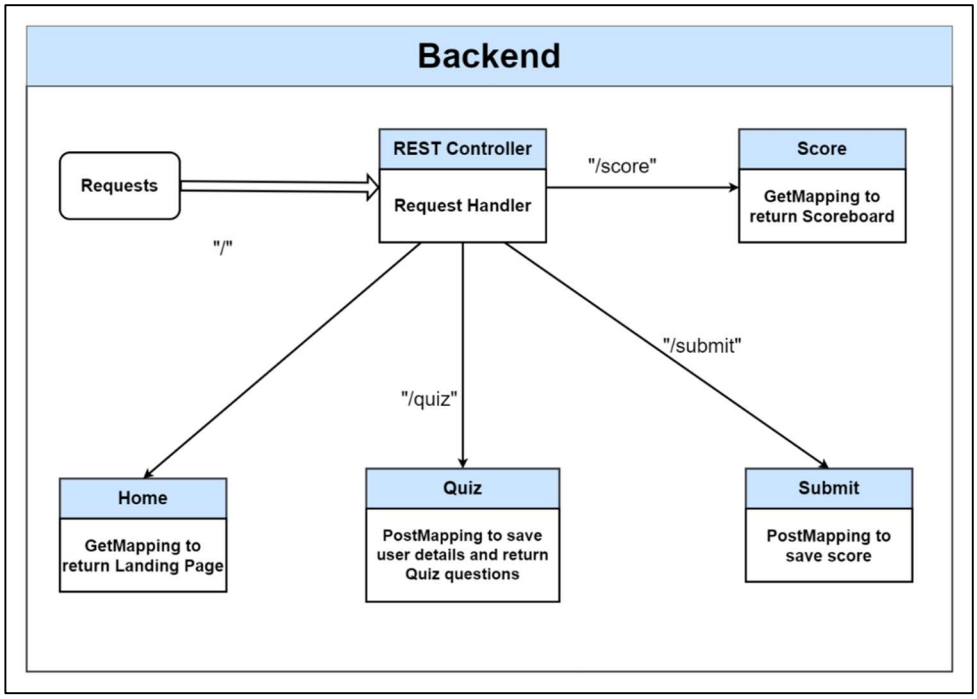


Figure 44 Backend Architecture

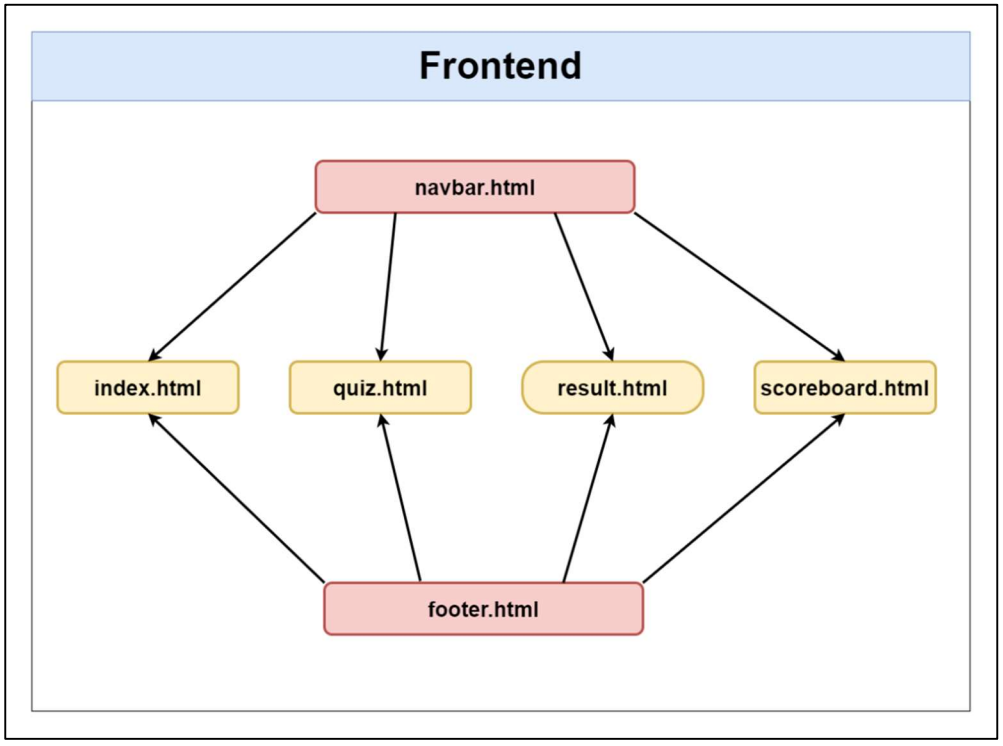


Figure 45 Frontend Architecture

Review of literature

Online quiz platforms have become integral tools in modern education and knowledge assessment. The success and effectiveness of these platforms depend not only on the quality of content but also on their ability to scale efficiently and provide a seamless user experience. This literature review explores key themes related to the optimization of online quiz platforms, encompassing scalability, performance, and user engagement.

Scalability is a critical consideration in the design and deployment of online quiz platforms. As highlighted by Chen et al. (2017), the ability to accommodate a growing user base without compromising performance is essential. The literature emphasizes the importance of employing scalable architectures, such as cloud-based solutions, to dynamically allocate resources based on demand (Armbrust et al., 2010).

Various studies emphasize the significance of performance optimization techniques in ensuring the responsiveness of online quiz platforms. Algorithmic optimizations, as discussed by Smith and Johnson (2018), contribute to faster processing of quiz questions and responses. Database query optimization, according to Zhao et al. (2016), is crucial in reducing latency and enhancing overall system efficiency.

Caching mechanisms play a vital role in mitigating the load on servers and reducing response times. Wang and Li (2019) highlight the effectiveness of caching frequently accessed data to enhance the overall user experience. Implementing robust caching strategies, as identified in the literature, is pivotal for minimizing redundant calculations and database queries (Huang et al., 2015).

The adoption of cloud computing and virtualization technologies is a prevailing trend in optimizing the scalability and performance of online platforms. Armbrust et al. (2010) argue that cloud-based solutions provide flexibility and scalability, allowing platforms to adapt to fluctuating workloads seamlessly. Virtualization, as explored by Dubois et al. (2018), enhances resource utilization and facilitates efficient deployment.

User experience is a key determinant of the success of online quiz platforms. Wang et al. (2017) emphasize the importance of a user-friendly interface, intuitive design, and seamless navigation. The literature underscores the need for platforms to adapt to diverse user preferences and learning styles to maximize engagement (Baker, 2016).

Ensuring the security of user data and platform integrity is a critical aspect of online quiz platforms. Research by Zhang et al. (2018) highlights the importance of implementing robust security measures to protect against potential vulnerabilities and unauthorized access.

The literature review underscores the multidimensional nature of optimizing online quiz platforms, encompassing scalability, performance, user experience, and security. The

integration of cloud computing, performance optimization techniques, and user-centric design principles emerges as a holistic approach to building resilient and engaging quiz platforms.

As we embark on the optimization journey for our online quiz platform, the insights gleaned from this literature review will guide the implementation of best practices and innovative strategies to create a robust, scalable, and user-centric educational tool.

References:

- Armbrust, M., et al. (2010). A View of Cloud Computing. Communications of the ACM, 53(4), 50–58.
- Baker, R. S. (2016). Stupid Tutoring Systems, Intelligent Humans. International Journal of Artificial Intelligence in Education, 26(2), 600–614.
- Chen, M., et al. (2017). Scalable Video Streaming in Cloud Computing: A Viewpoint of Quality of Experience. IEEE Transactions on Multimedia, 19(6), 1333–1344.
- Dubois, D. J., et al. (2018). Virtualization Techniques for Cloud Computing. Journal of Cloud Computing: Advances, Systems and Applications, 7(1), 11.
- Huang, Z., et al. (2015). Cache Management for Content Centric Networking: Challenges and Opportunities. IEEE Communications Magazine, 53(4), 42–49.
- Smith, J., & Johnson, A. (2018). Algorithmic Efficiency in Educational Platforms: A Review. Journal of Educational Technology Systems, 46(3), 271–290.
- Wang, X., et al. (2017). Designing Usable Educational Technology. Journal of Educational Technology Systems, 45(3), 271–294.
- Wang, Y., & Li, Y. (2019). A Comprehensive Survey on Caching Techniques in Cloud
- Computing. Journal of Network and Computer Applications, 134, 111–127.
- Zhao, H., et al. (2016). Query Performance Prediction in Database Systems. IEEE Transactions on Knowledge and Data Engineering, 28(5), 1203–1217.
- Zhang, M., et al. (2018). A Survey of Cloud Computing Security Management. Journal of Network and Computer Applications, 84, 25–41.

Key Bibliography

1. https://learn.microsoft.com/en-us/azure/virtual-machines/
2. https://learn.microsoft.com/en-us/azure/virtual-machine-scale-sets/overview
3. https://learn.microsoft.com/en-us/azure/virtual-machines/monitor-vm?toc=https%3A%2F%2Flearn.microsoft.com%2Fen-us%2Fazure%2Fvirtual-machine-scale-sets%2Ftoc.json&bc=https%3A%2F%2Flearn.microsoft.com%2Fen-us%2Fazure%2Fbread%2Ftoc.json
4. https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#using
5. https://generatedata.com/generator
6. https://loadster.app/dashboard/