**CLDV6212**

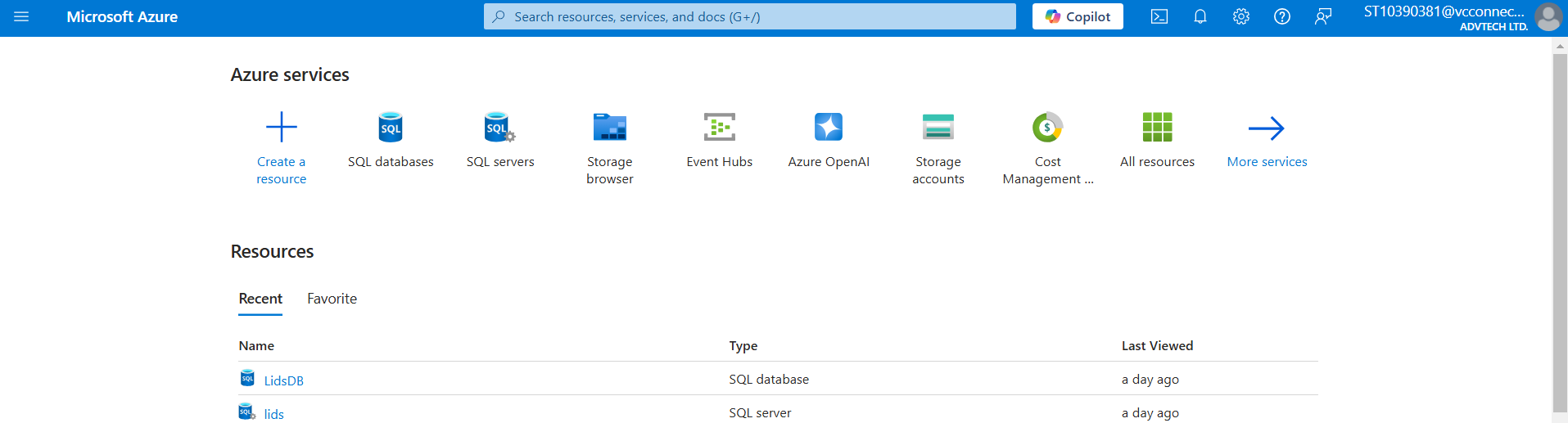
**ST10390381**

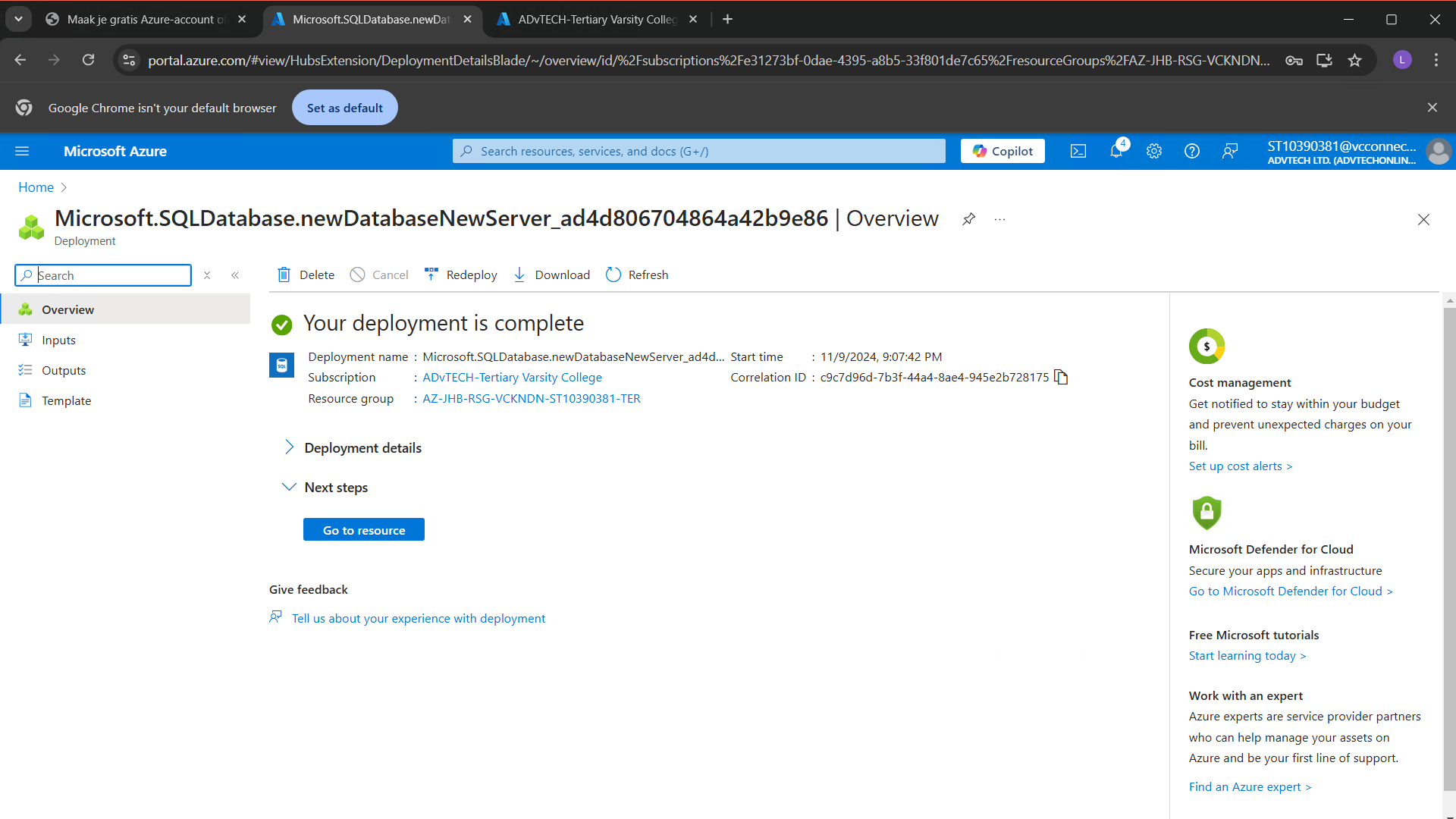
**POE Final 3**

Live URL: <https://retailabc-ekeugwcqhuc7cadf.canadacentral-01.azurewebsites.net>

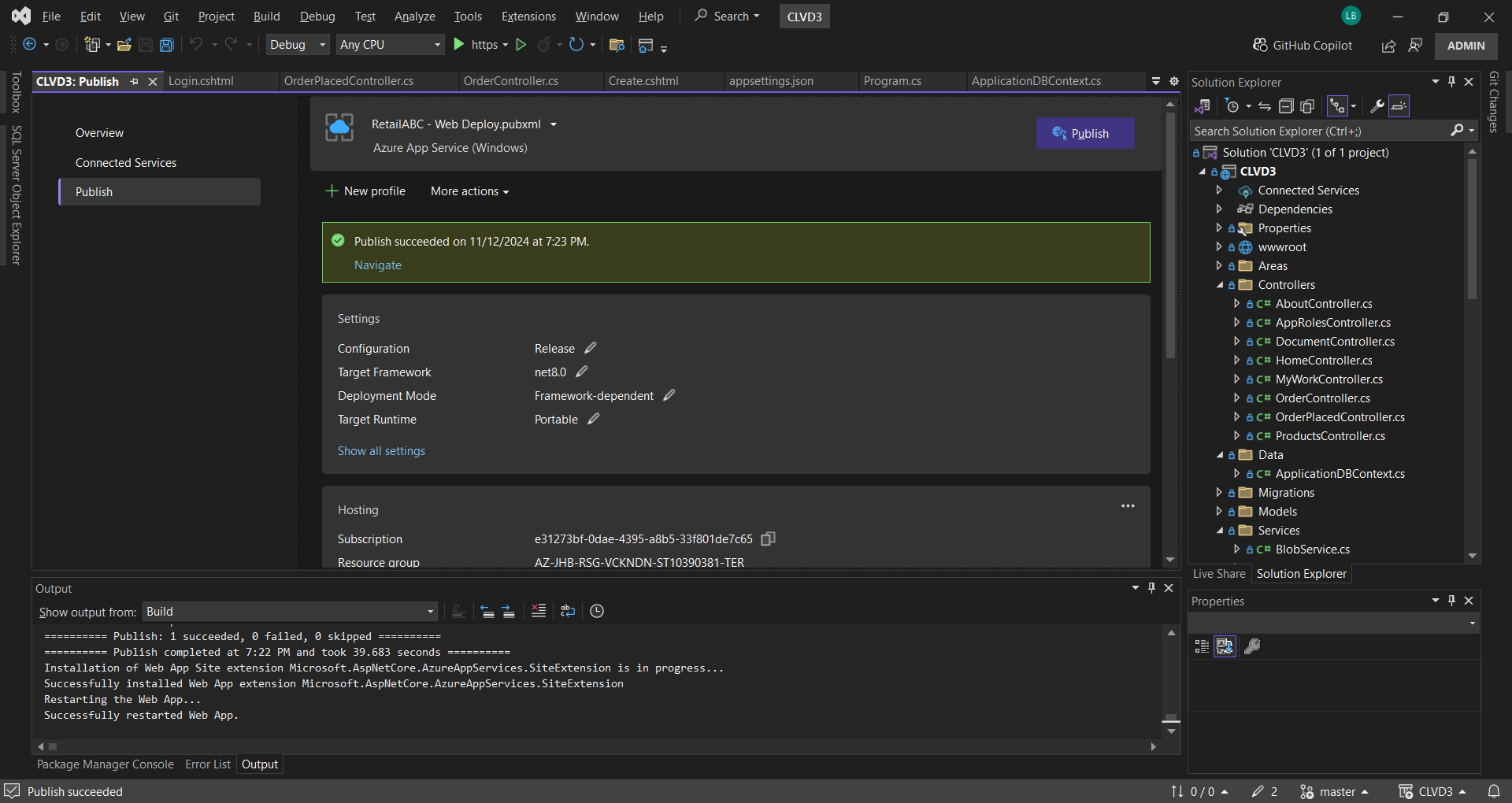
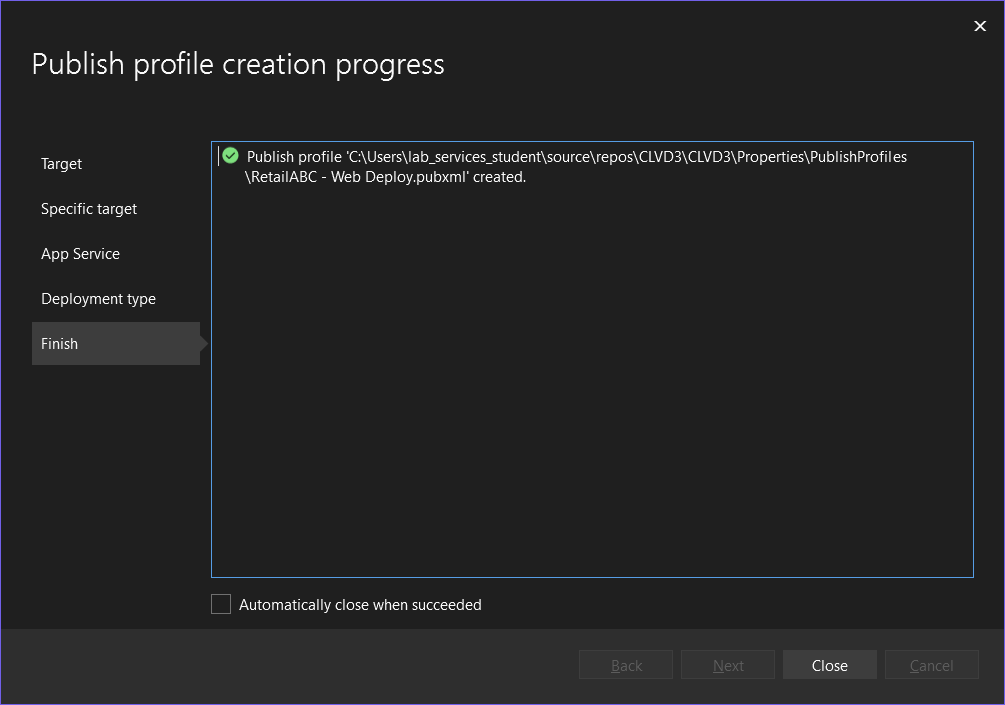
GitHub Link: <https://github.com/st10390381/CLOUDPOE3.git>

1. **SQL Database Created**

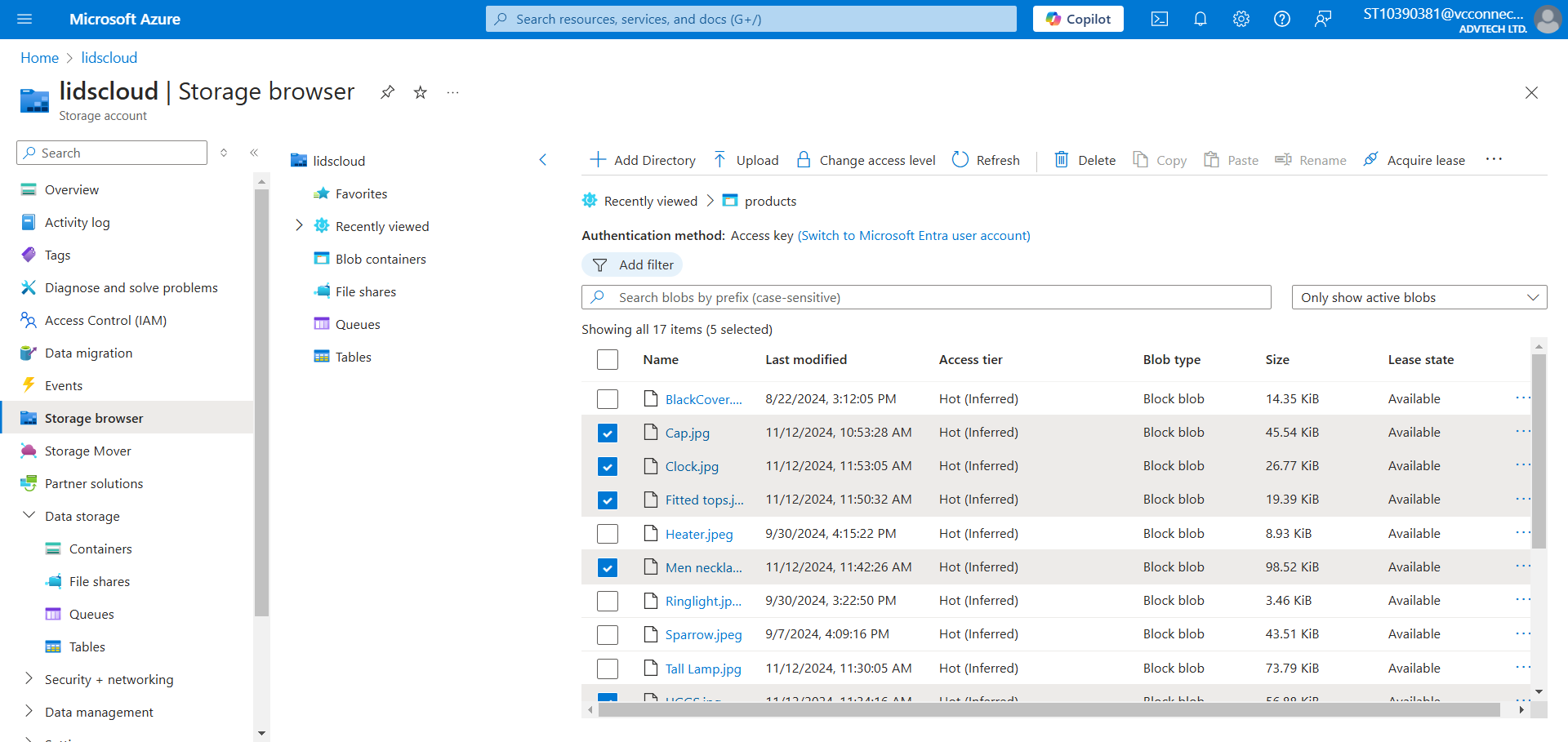


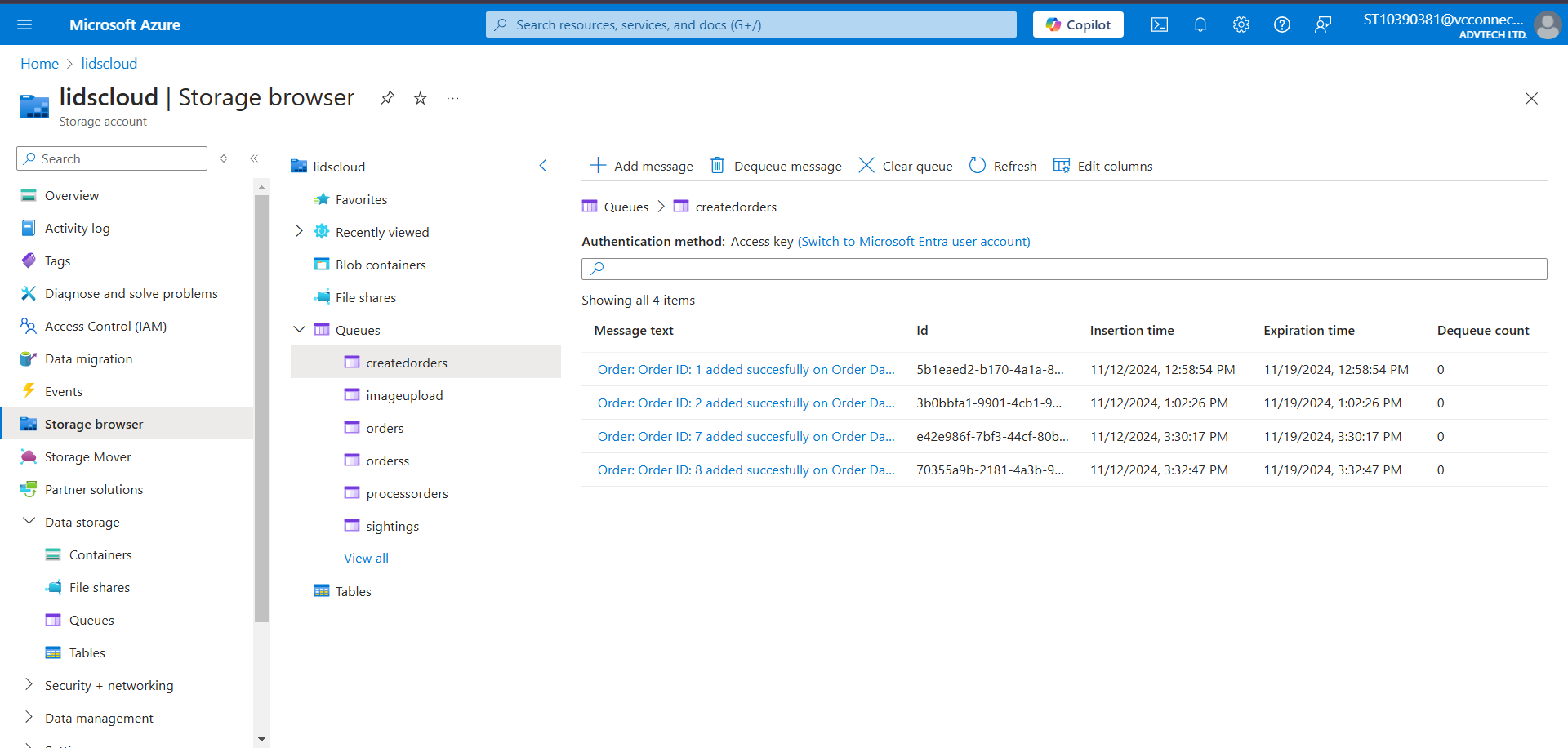


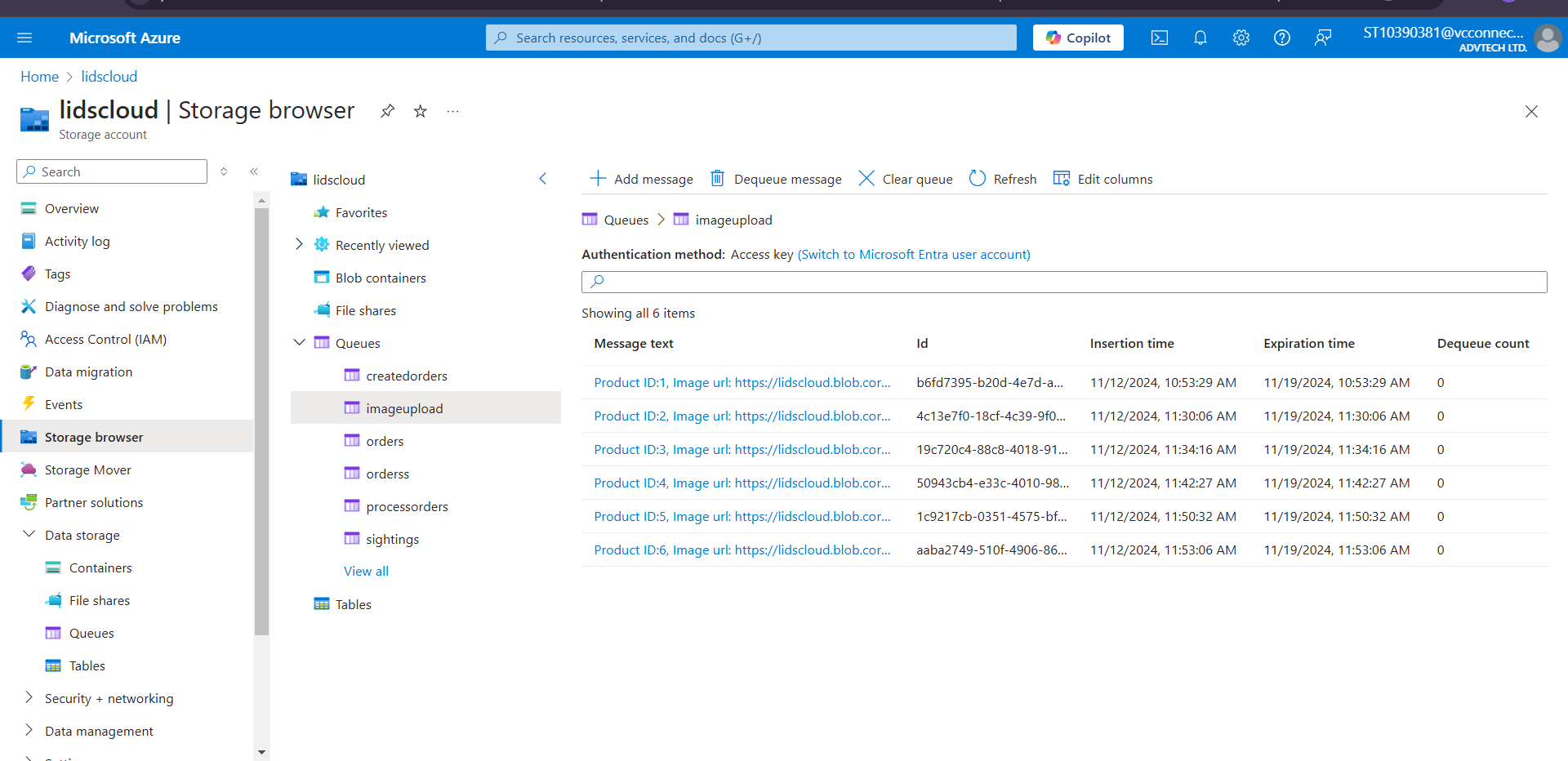
**Code Published**



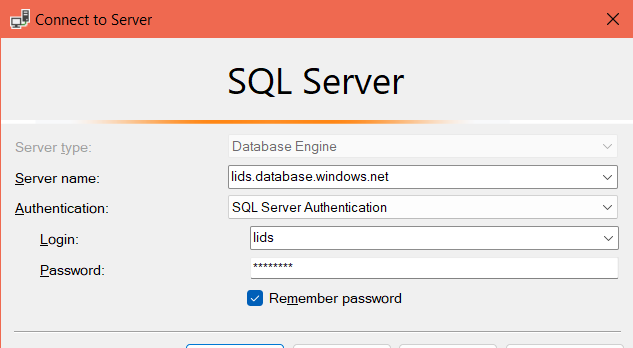
**Azure Storage Account**

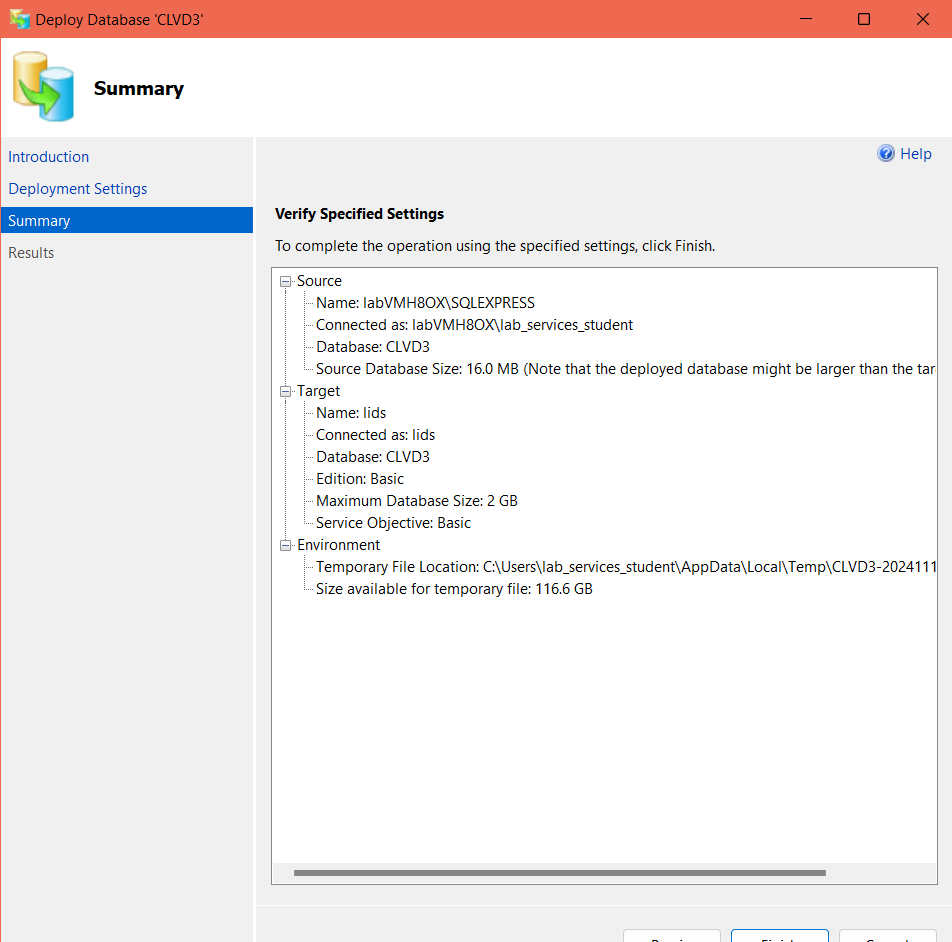
Update Blob Storage

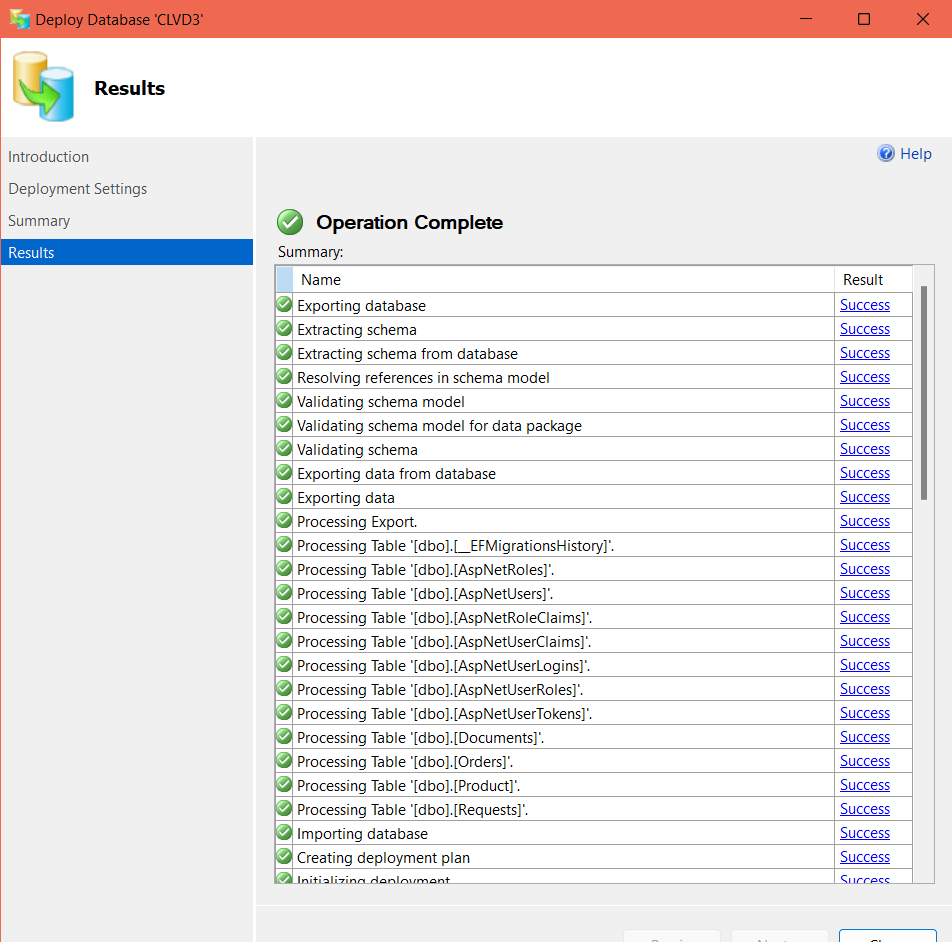
Order Stored (Queue) 

Product Image Stored (Queue) 

**SSMS Screenshots**

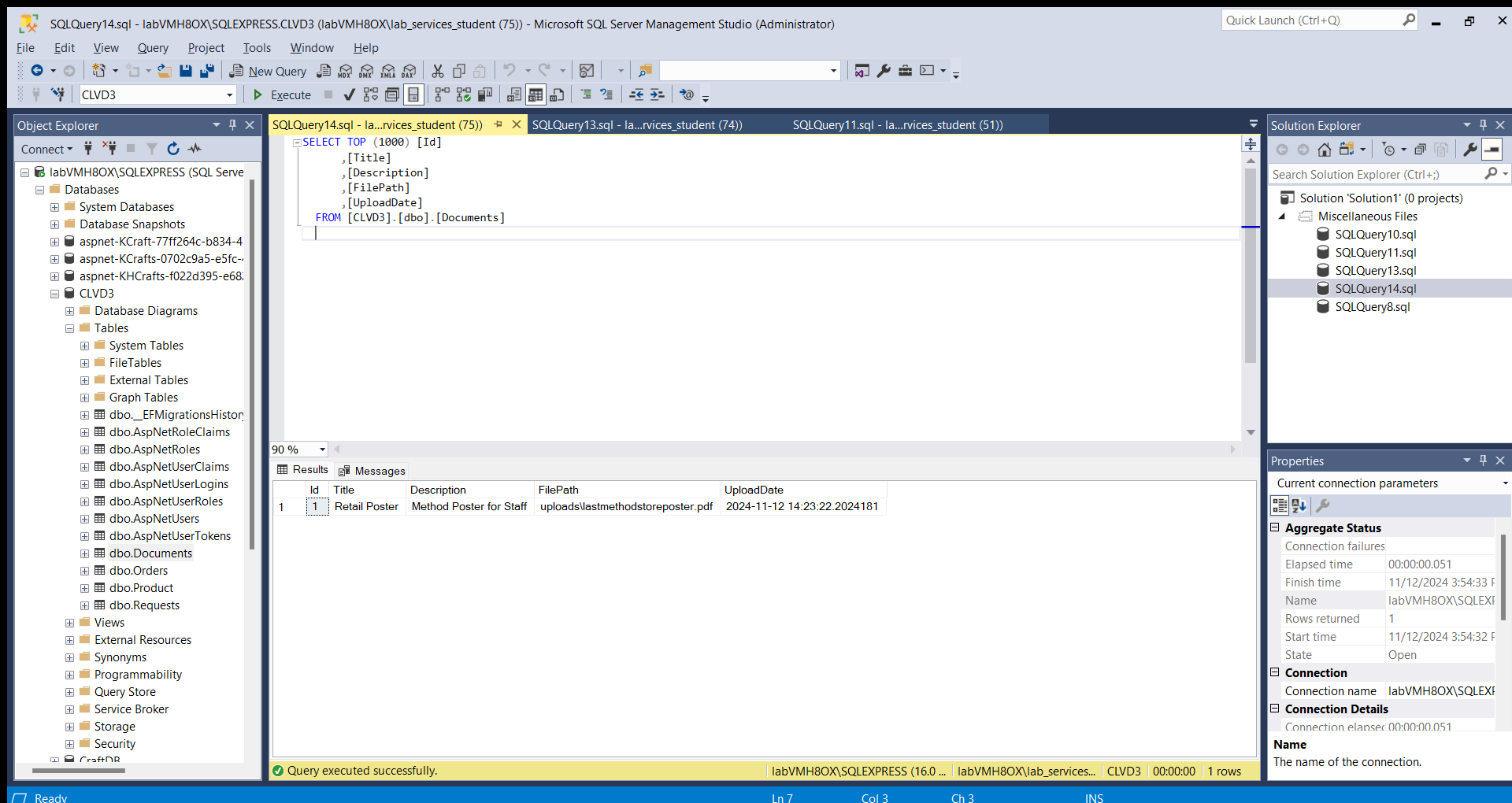
**Servers connected** 

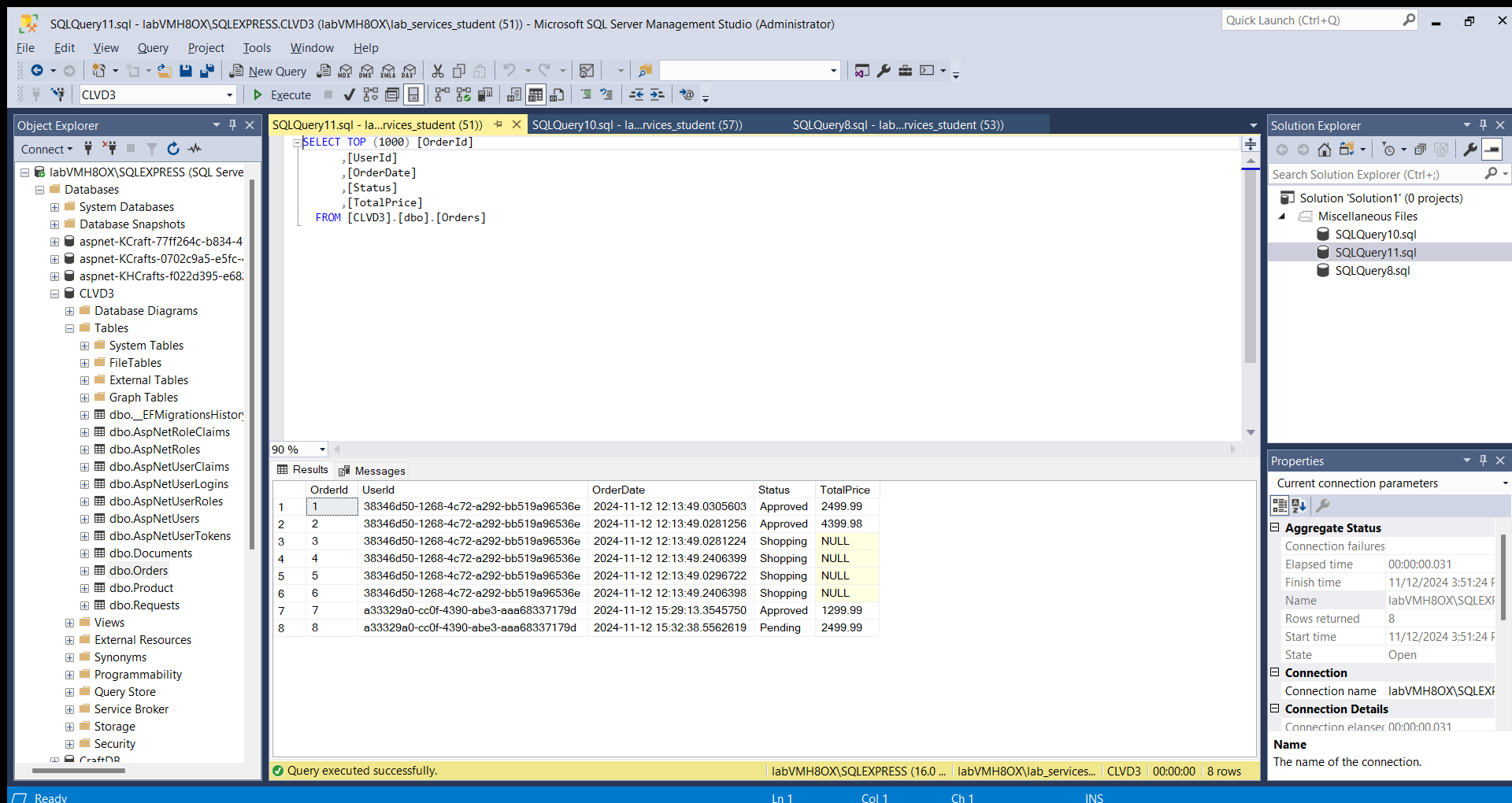


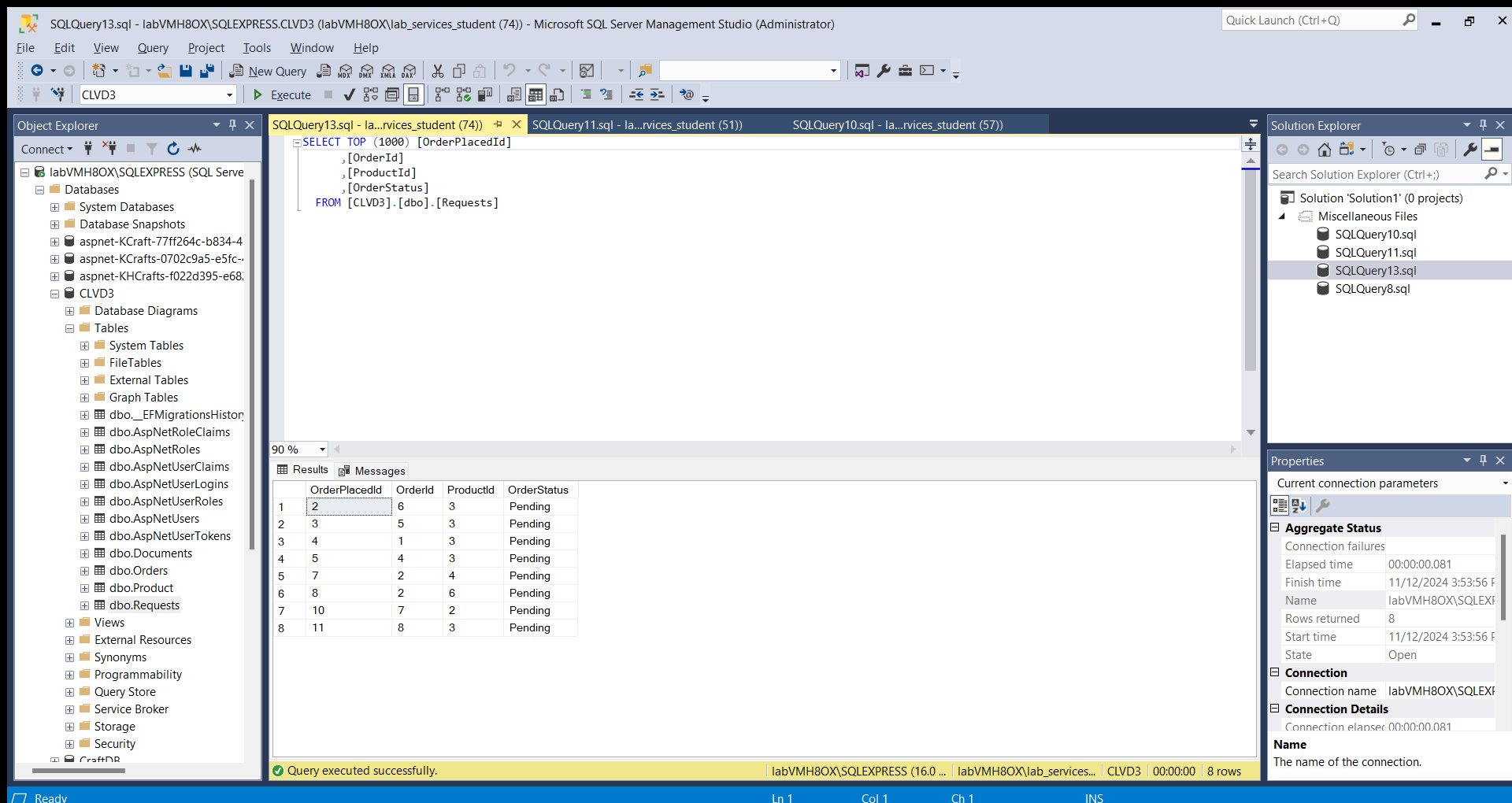


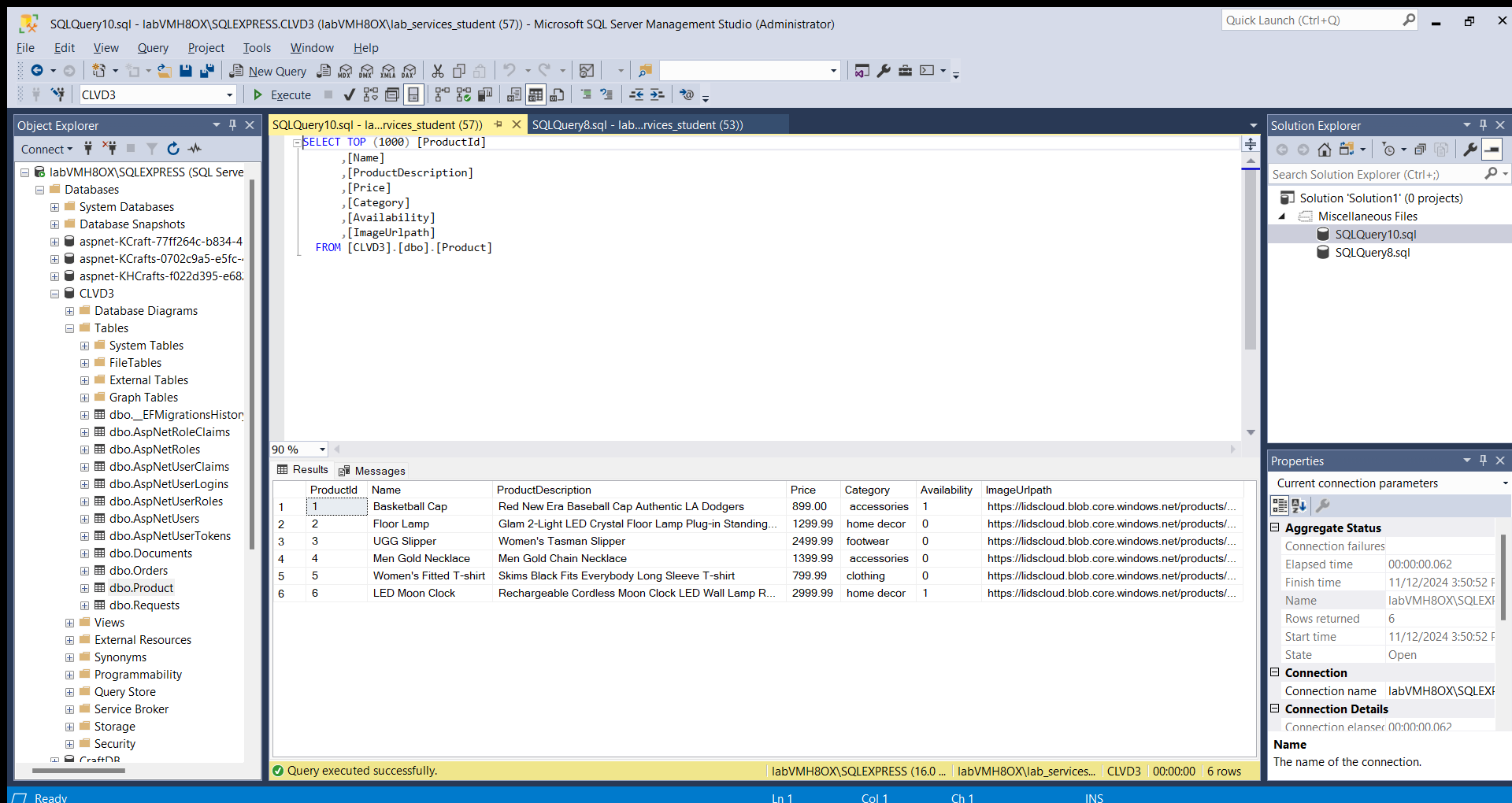
**SSMS Tables**

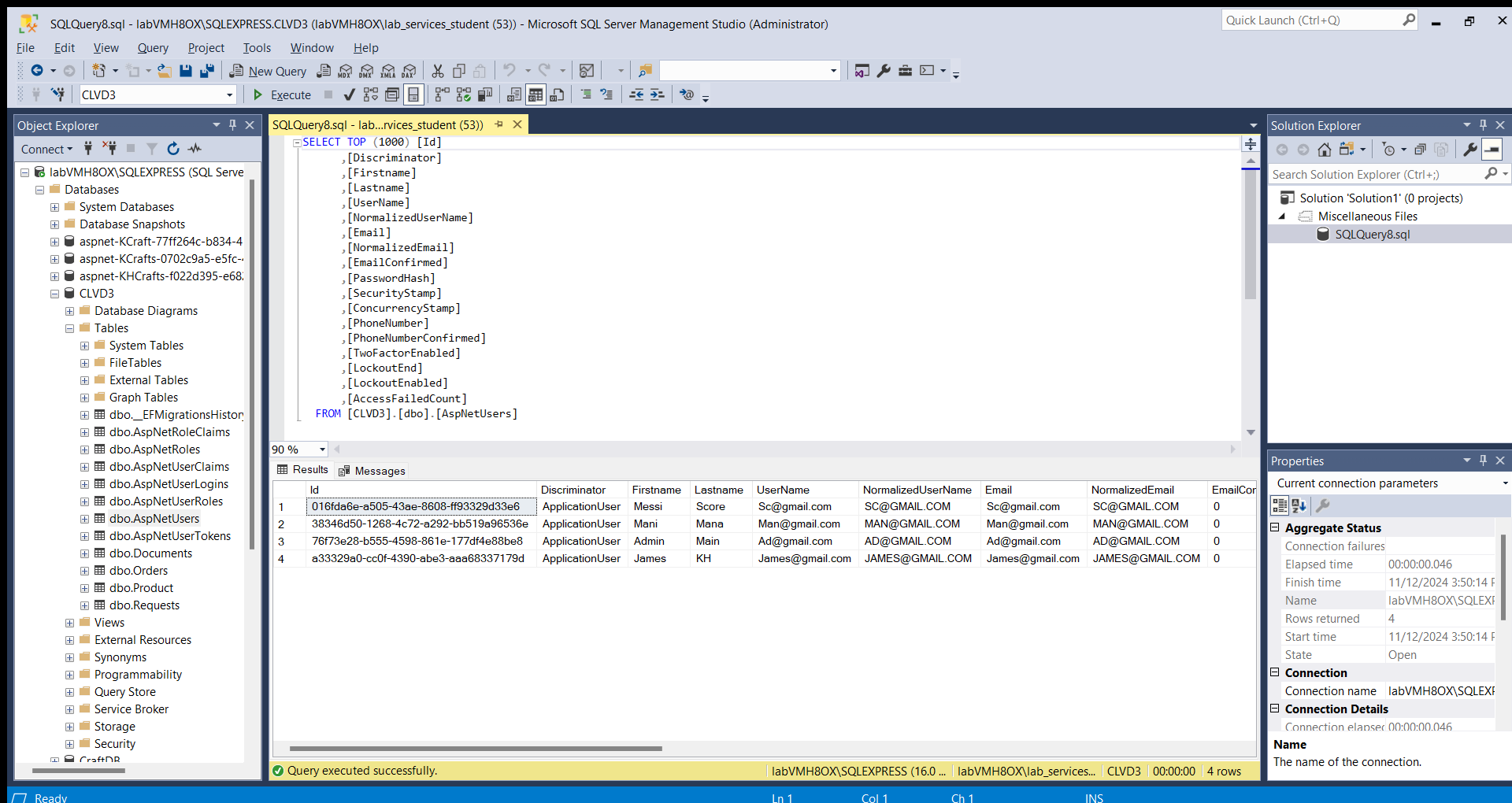
Document SSMS

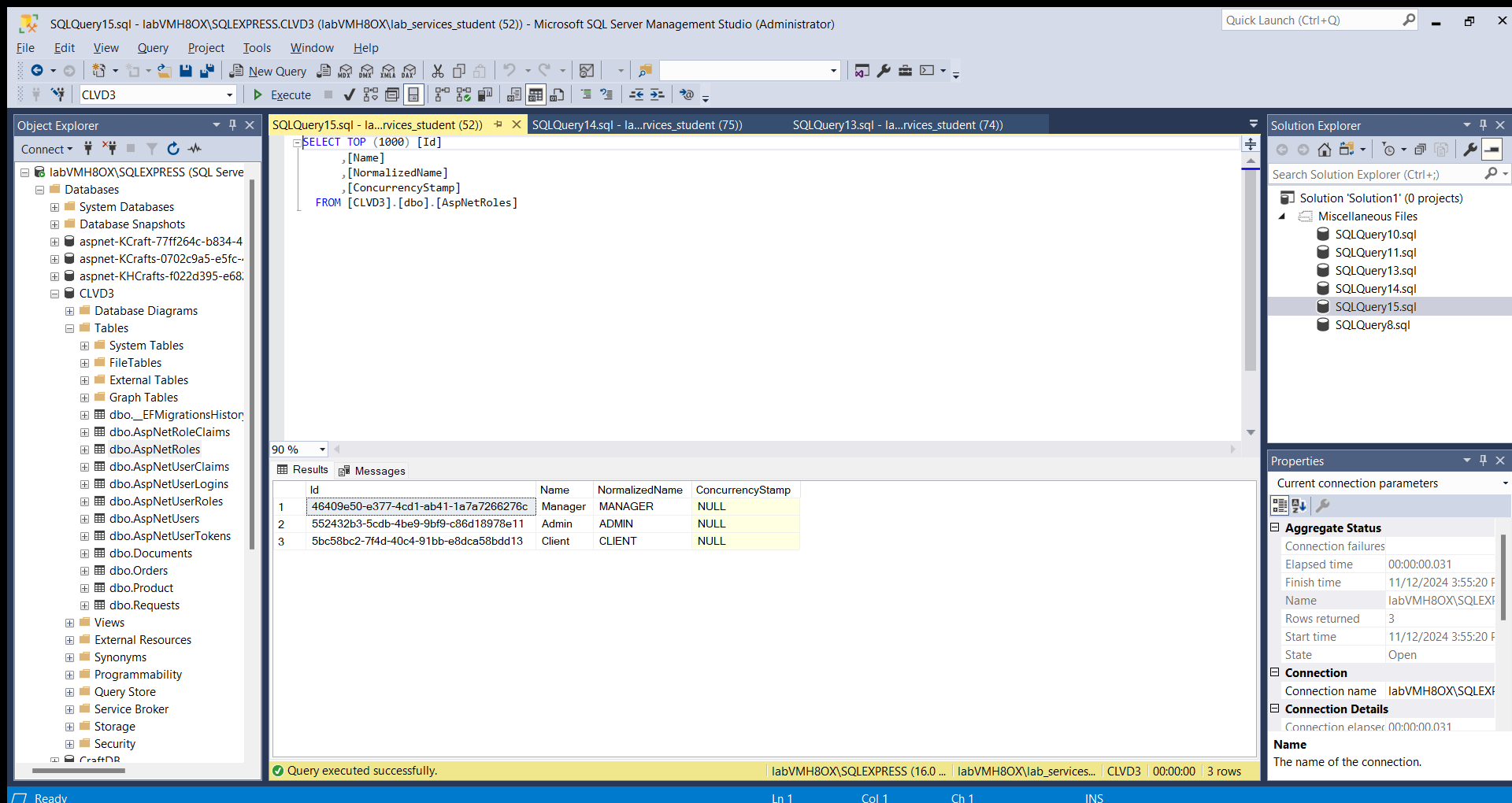


Order SSMS

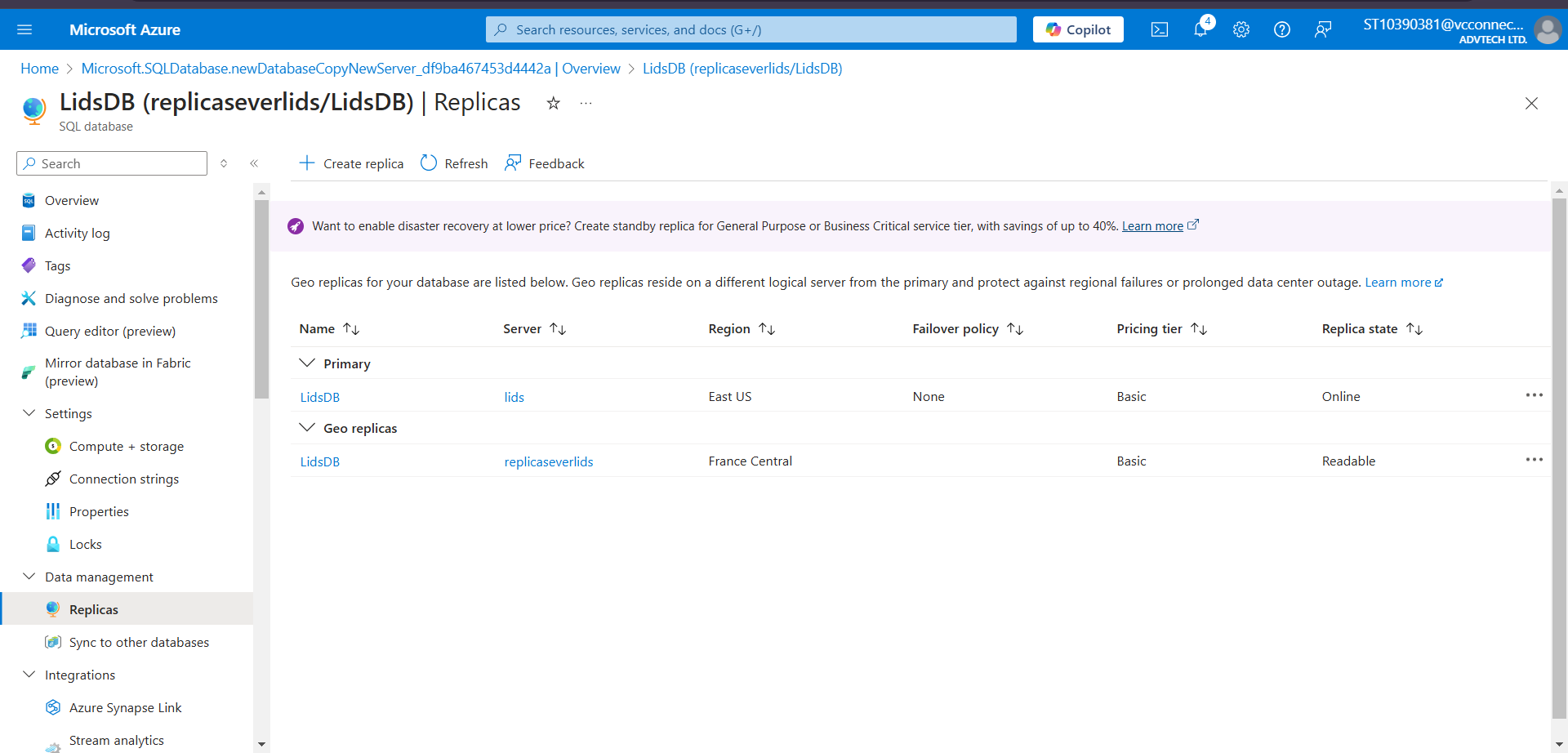
RequestsSSMS

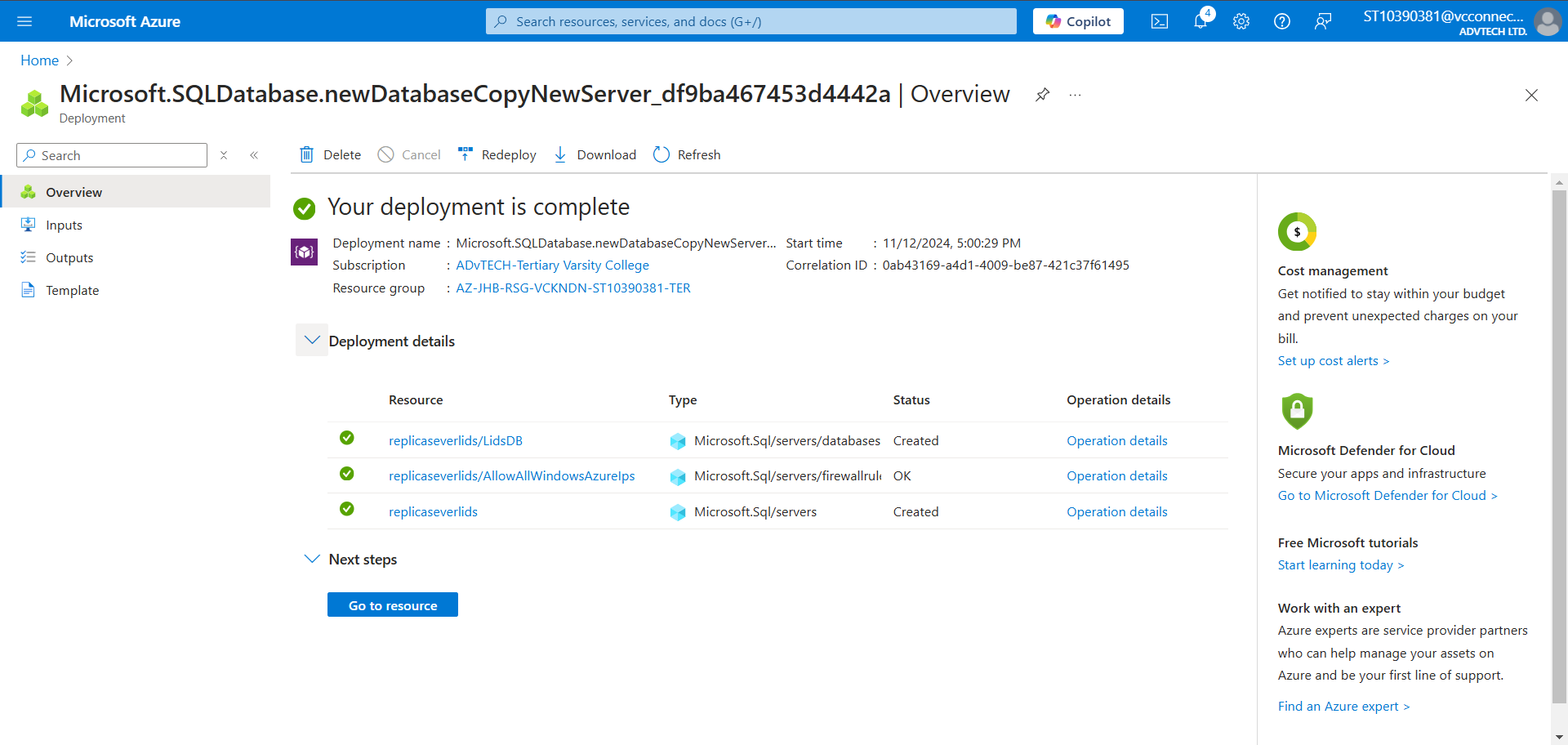
Products SSMS

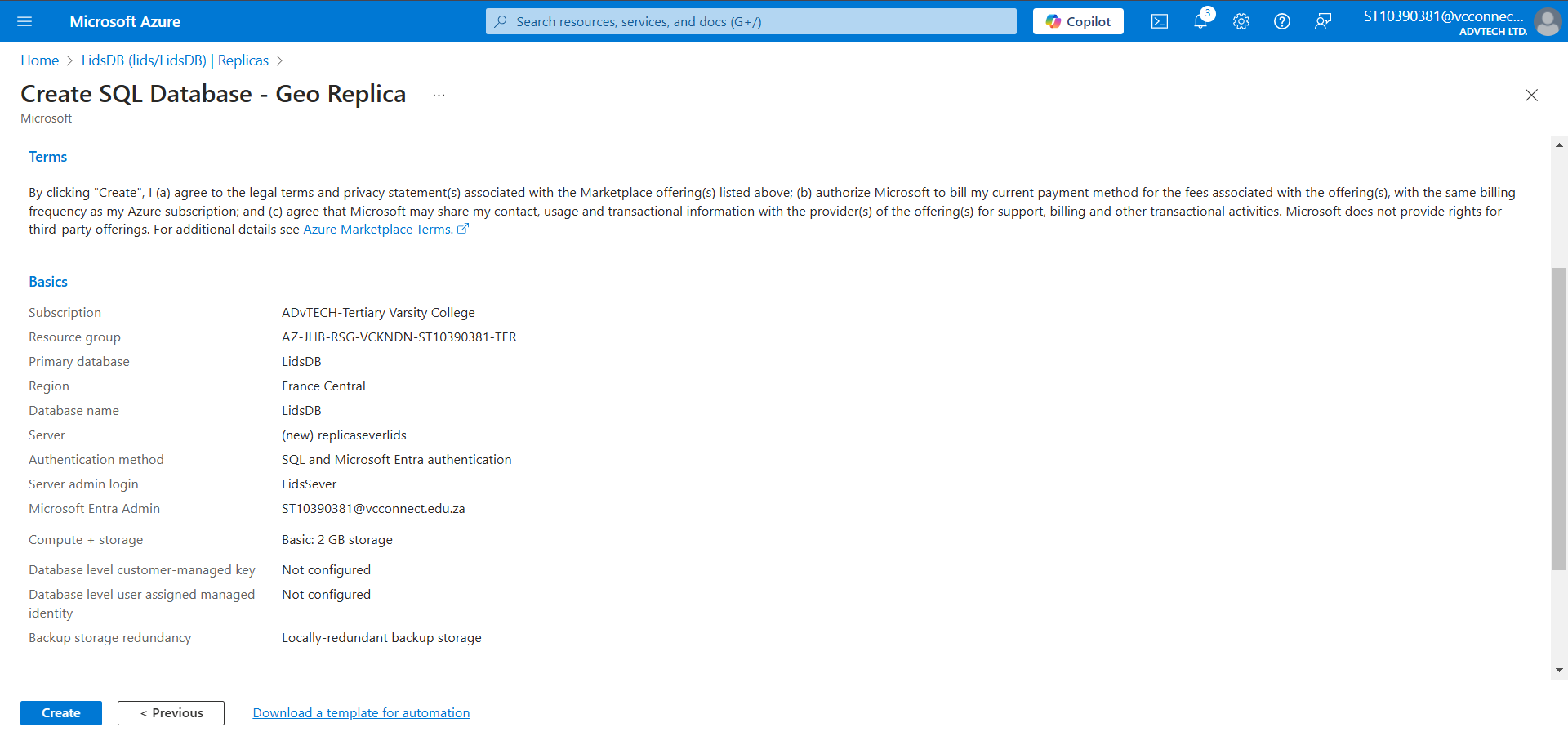
Users SSMS 

Roles SSMS

**Replicas Azure**



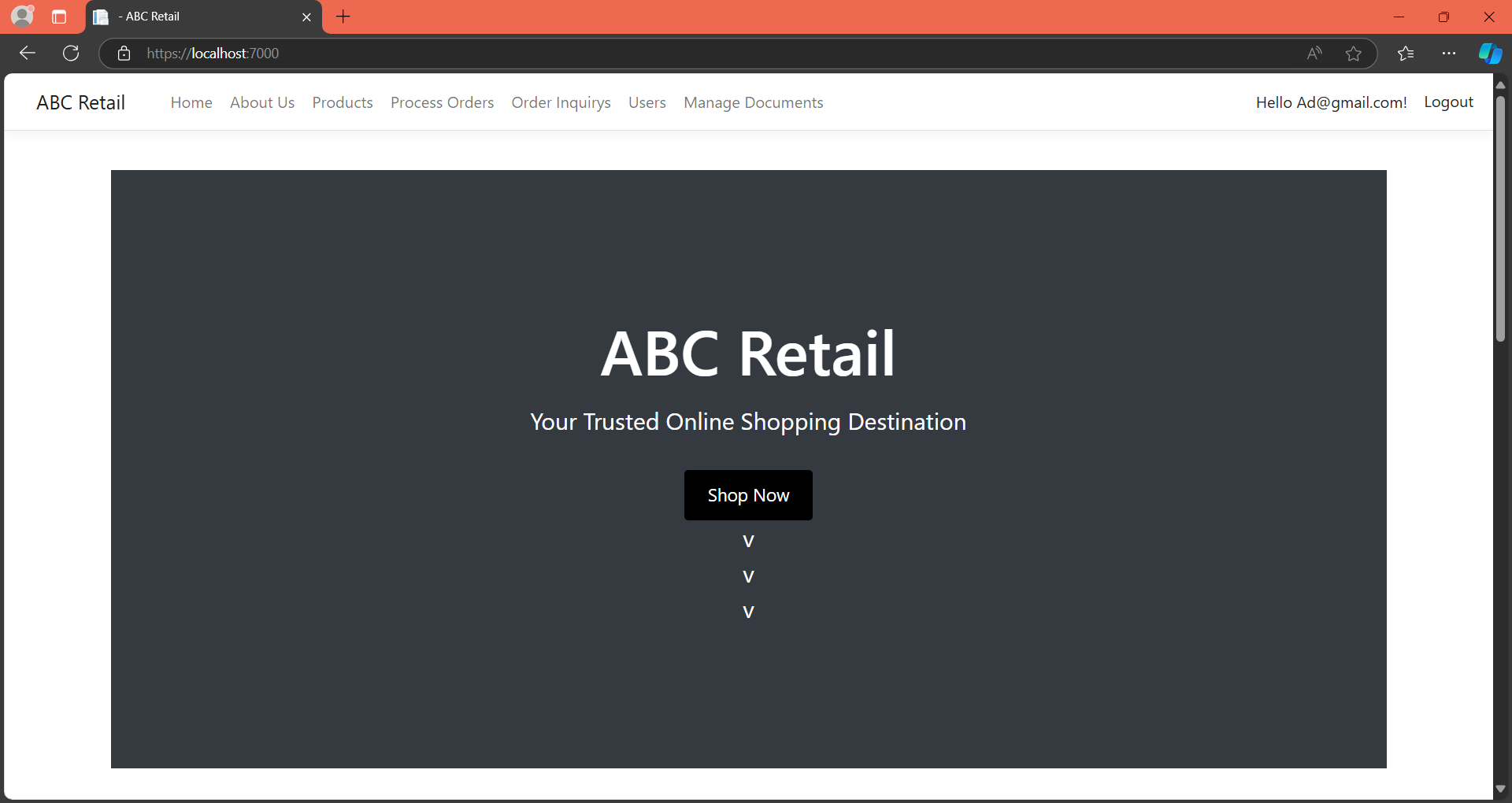


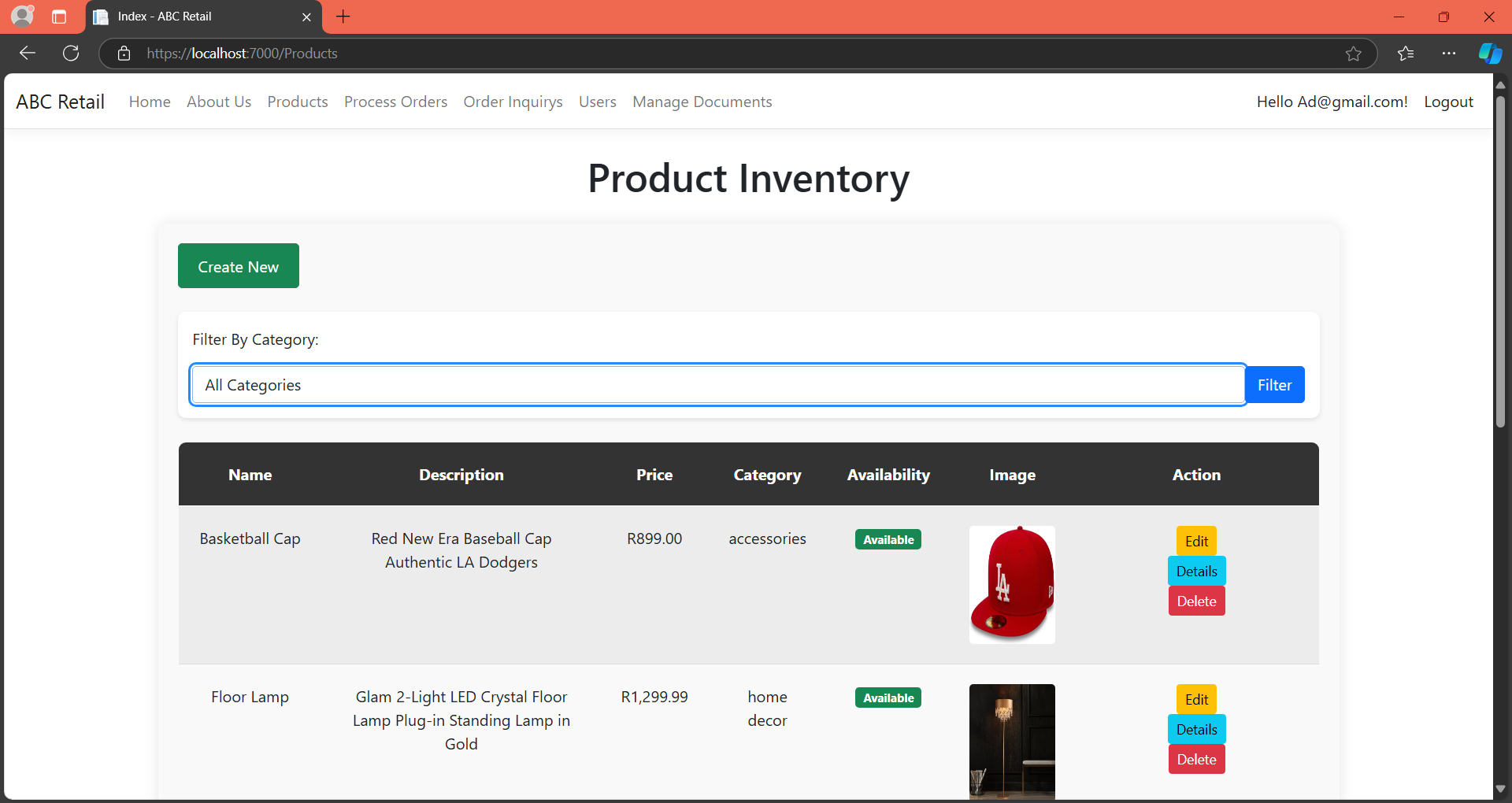


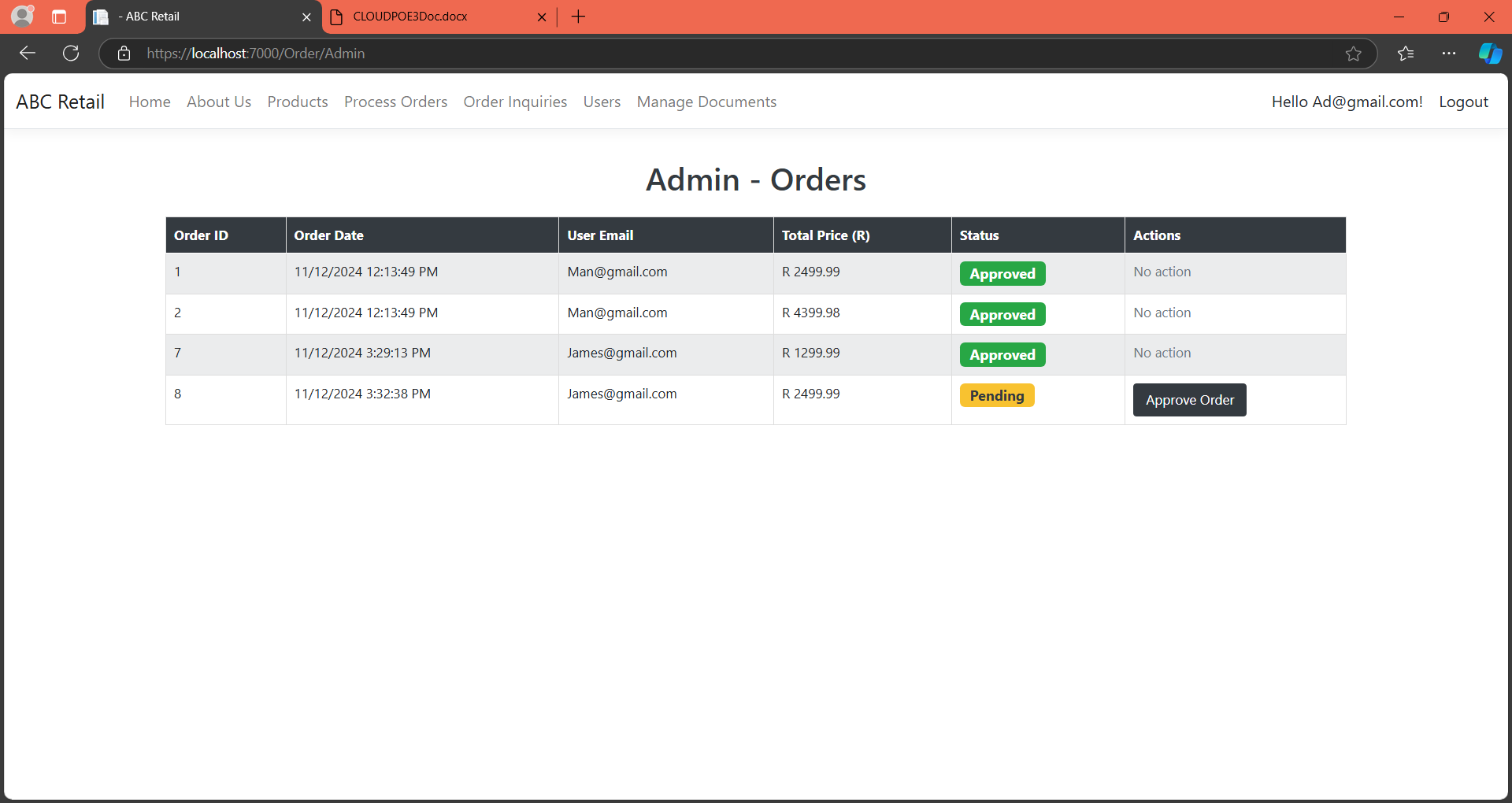
**Motivation**

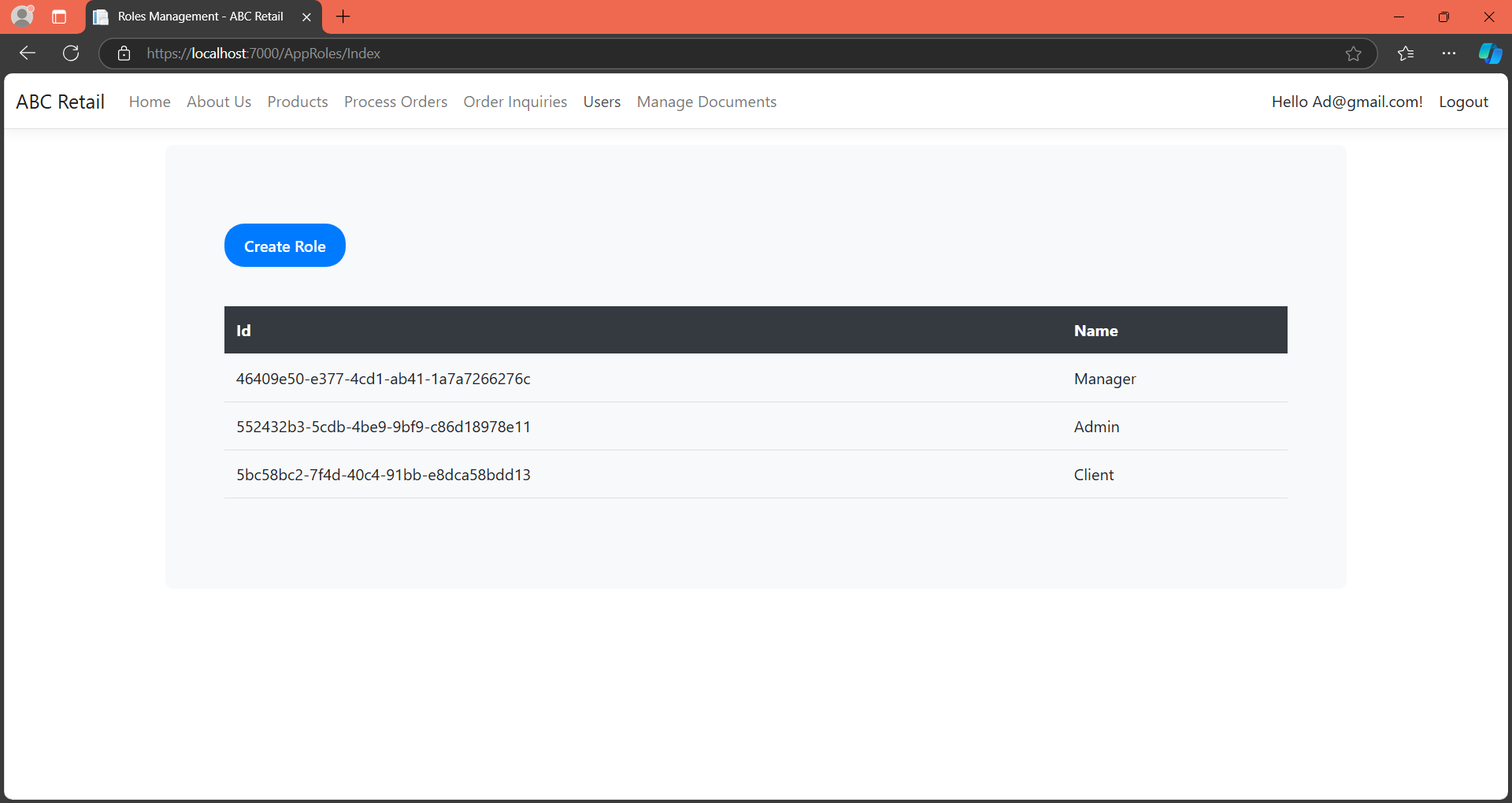
In my ABC Retail application, having an Azure replica would be essential to ensure high availability and fault tolerance for the critical data, such as product details, customer information, and order transactions. With the unpredictable nature of customer demand and the need for real-time access to data, a replica would provide a backup solution in case of system failures or outages, minimizing downtime. By replicating my Azure SQL Database across multiple regions, I can ensure that if one region experiences issues, the application can seamlessly switch to a healthy replica, keeping the system running without interruptions. This would not only enhance the resilience of the application but also improve the user experience, as customers and employees would always have access to the data they need, even during high traffic periods or maintenance windows. Additionally, replicas can help with read scalability, allowing for faster query responses and reducing the load on the primary database.

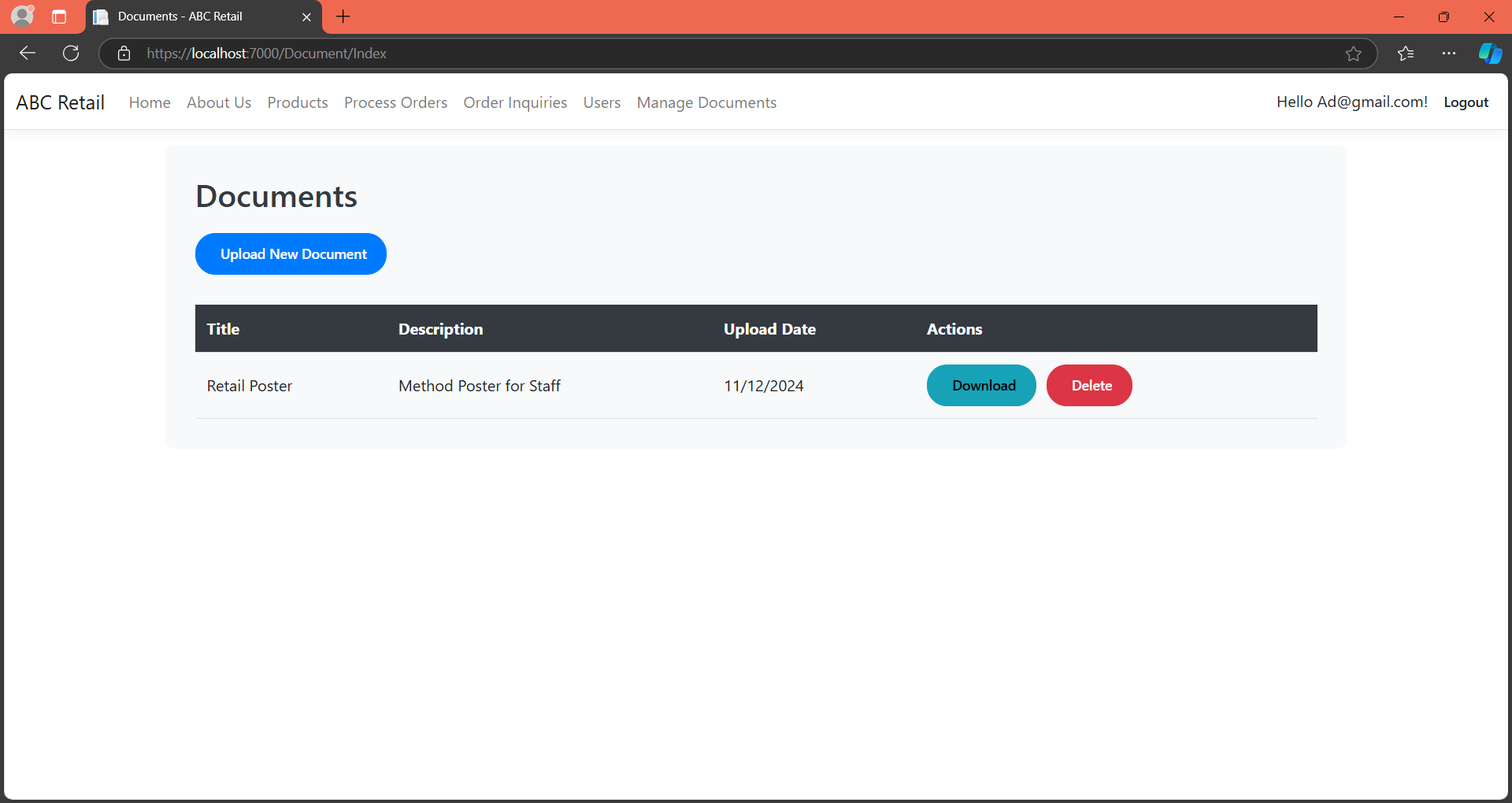
**Admin View**



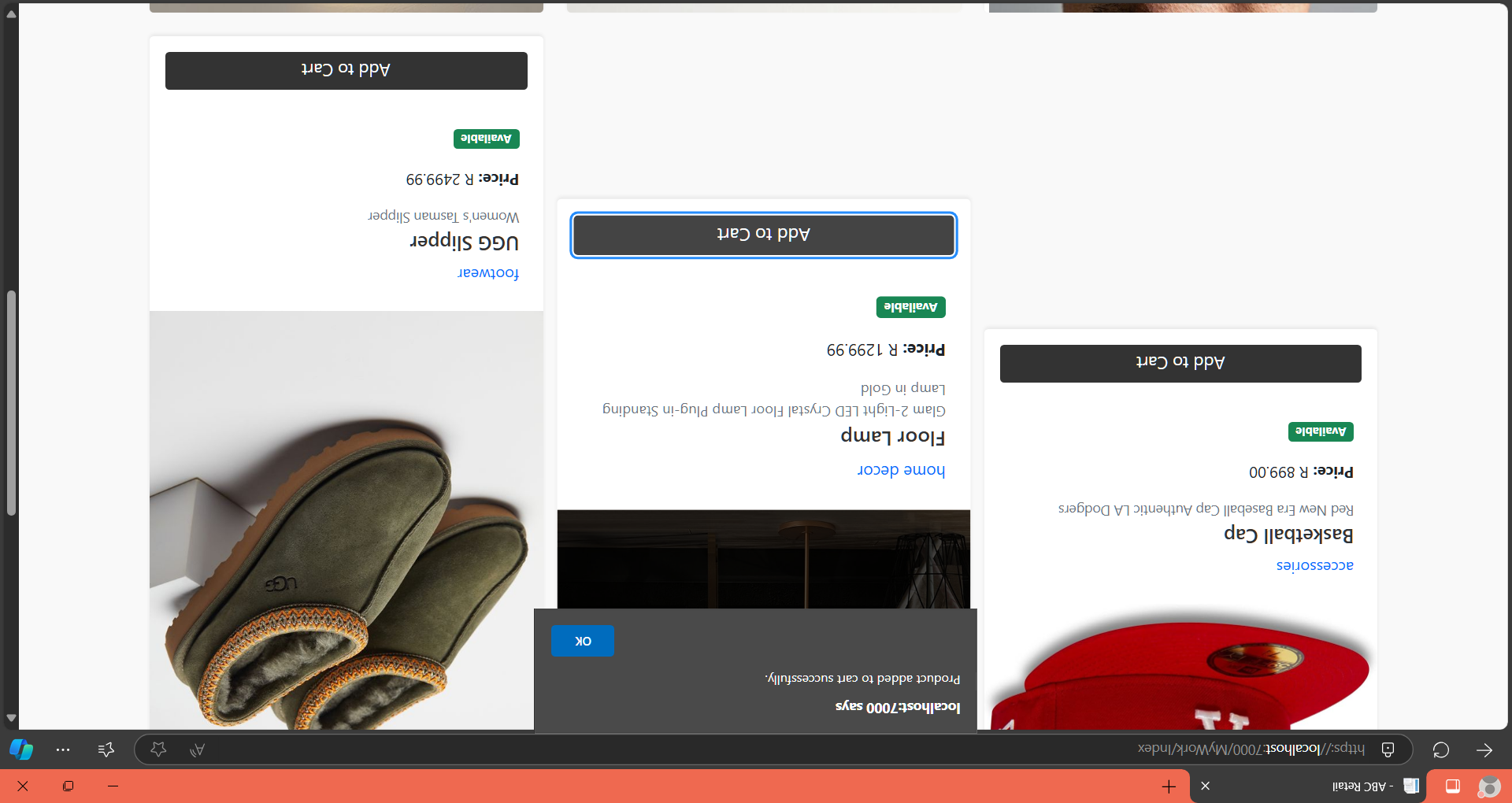


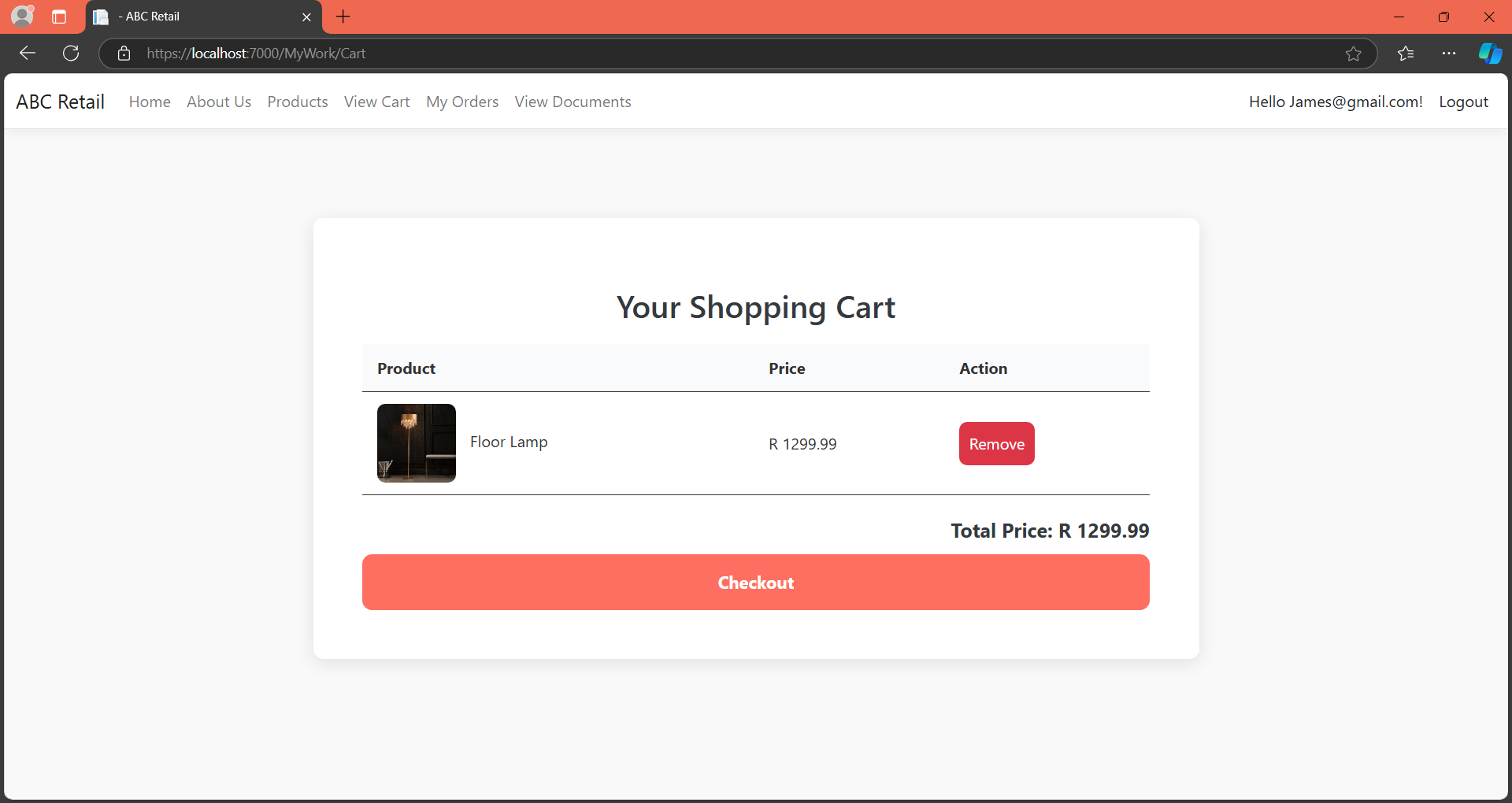


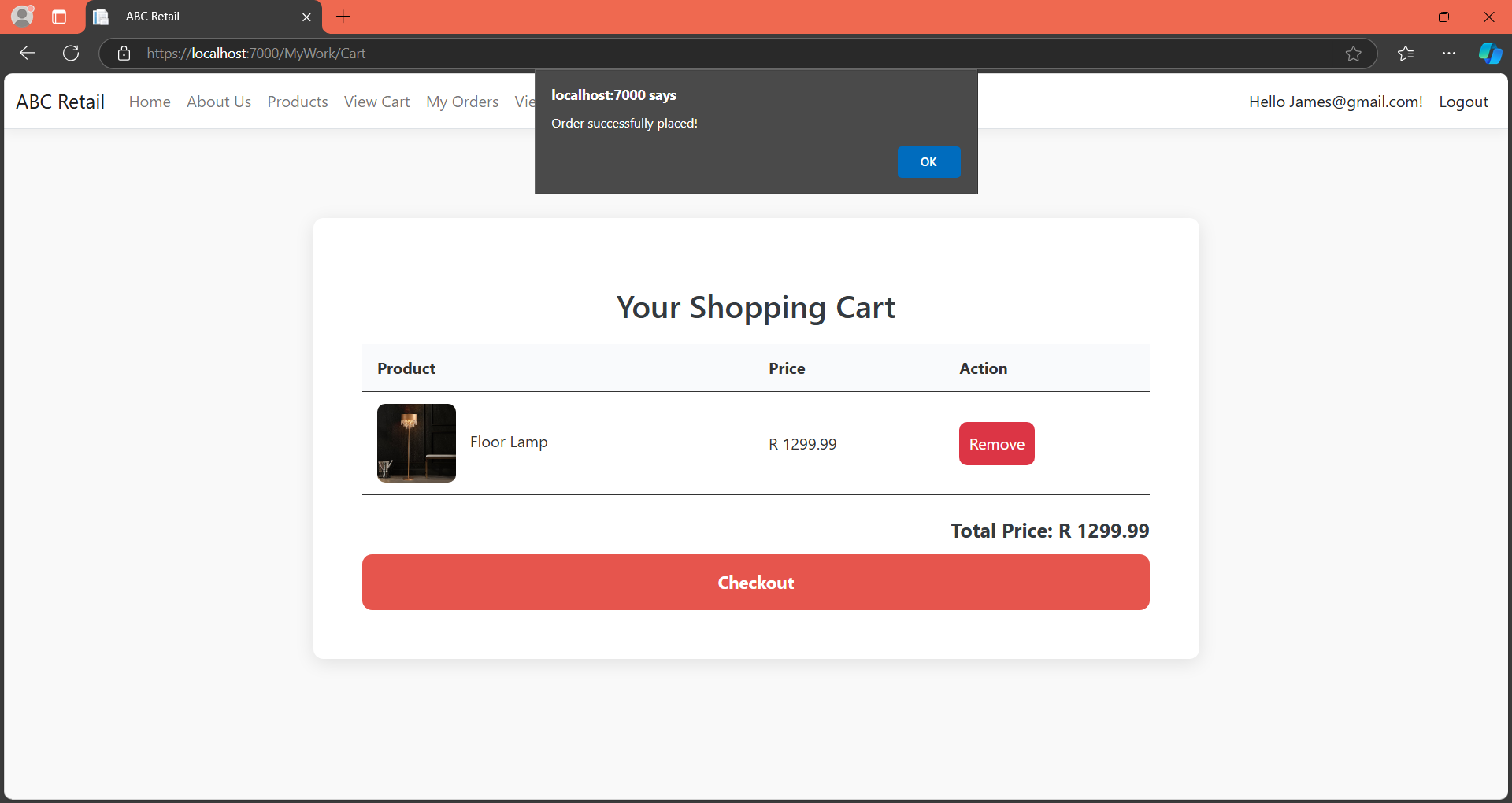


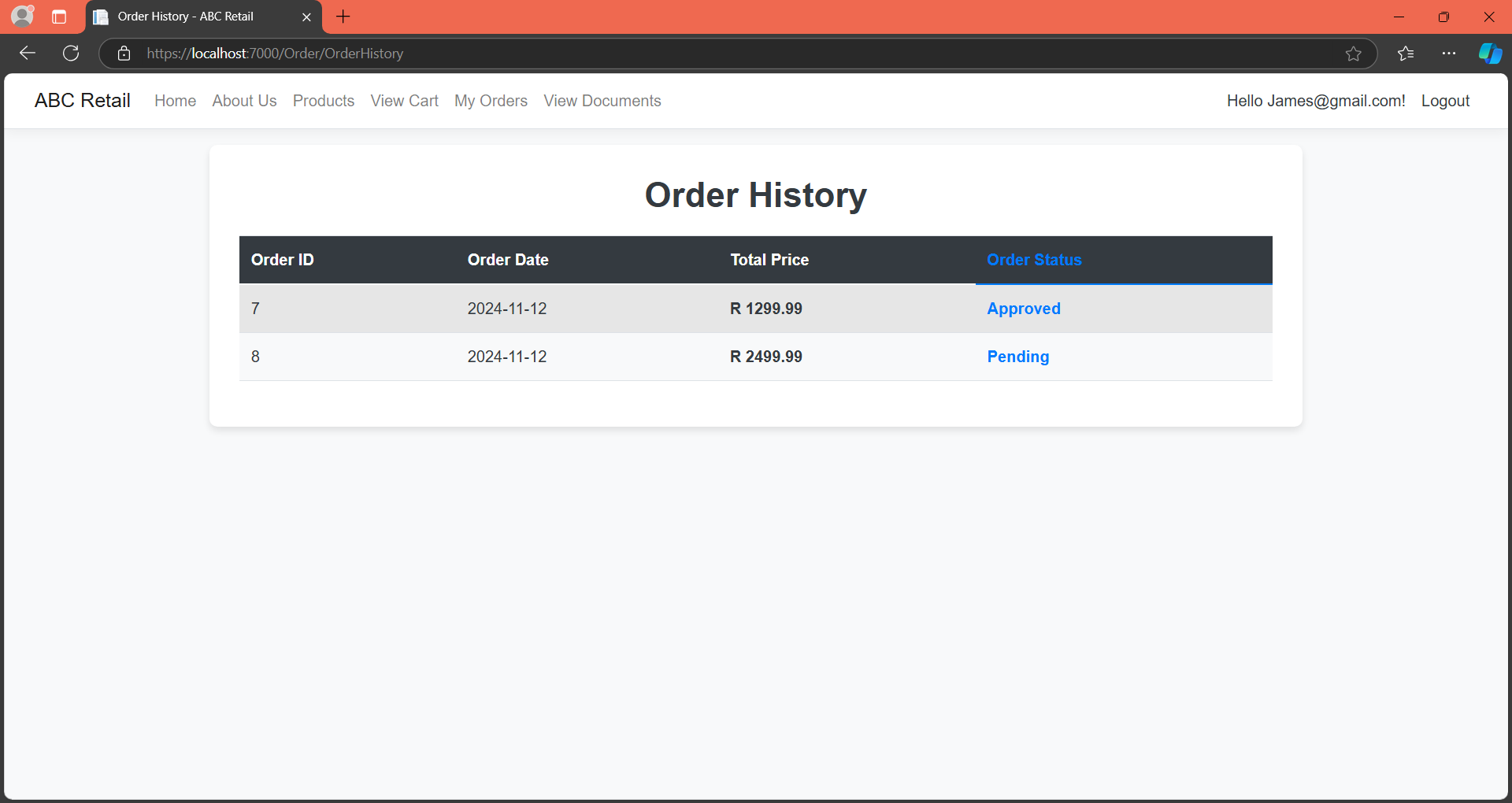


**Customer View**









**B. Document the Technology Choices for My Solution**

**Table of Technology used:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Technology Choice (Category)** | **Hosting Model** | **Purpose** |
| Azure SQL Database | Cloud-hosted relational database (Database Management) (Microsoft, 2024) | PaaS | Azure SQL Database is a fully managed cloud relational database service that ensures high availability, scalability, and security for structured data. It simplifies database management by handling tasks like patching, backups, and monitoring, while offering robust querying and transaction processing capabilities for applications. (Microsoft, 2024) |
| Azure Blob Storage | Cloud file storage for unstructured data (Data Storage) (Microsoft, 2024) | PaaS | |  | | --- | | Provides scalable cloud storage for unstructured data, storing media files such as product images, documents, and backups needed by applications. (Microsoft, 2024) | |
| Azure Server | Cloud-based server for database hosting (Cloud Infrastructure) (Microsoft, 2024) | PaaS | Provides a secure and scalable environment for hosting databases, enabling centralized access and management of data for applications and users. |
| Azure Queue Storage | Data Processing (Microsoft, 2024) | Paas | Provides a message queue service to handle asynchronous tasks and manage background processing for smooth communication between distributed application components. (Microsoft, 2024) |
| Azure Files | |  | | --- | |  |  |  | | --- | | File Storage |   (Microsoft, 2024) | Paas | Offers a fully managed file sharing solution in the cloud, allowing for shared storage and easy access across applications via SMB or REST API. (Microsoft, 2024) |
| Azure Function | Application Development (Microsoft, 2024) | Faas (Function as a service) | Executes event-driven code in response to triggers, enabling automated processing tasks without a dedicated server. (Microsoft, 2024) |

**C. Motivate the Azure Services Used for Each Application Functionality**

### **1) Azure SQL Database**

**-Requirement**: I need a fully managed relational database service to store, query, and manage structured data such as customer details, product inventories, and order records.

**-Service**: Azure SQL Database

**-Motivation**: Azure SQL Database is a fully managed relational database service that simplifies the process of maintaining a cloud-based database. It ensures high availability, automatic backups, and built-in security features, eliminating the need for manual database management tasks like patching and scaling. With its robust querying capabilities, it provides efficient and real-time access to application data. Azure SQL Database’s ability to scale on-demand, coupled with its high availability and disaster recovery features, ensures that the application can handle growing amounts of data while remaining secure and responsive. Additionally, it integrates seamlessly with other Azure services, allowing for easy reporting and analytics, which enhances overall application functionality. The automatic scaling and flexible pricing options make it cost-efficient for both small and large-scale applications, providing the flexibility needed as the application grows. *(Jain, 2019)*

### **2) Azure Blob Storage**

**-Requirement**: I need to store large amounts of unstructured data, such as product images, videos, documents, and backups, in a scalable and accessible manner.

**-Service**: Azure Blob Storage

**-Motivation**: Azure Blob Storage is an ideal solution for storing unstructured data, especially media files like product images, videos, and documents, which are essential for the application's functionality. It offers highly scalable storage, which allows the application to store and retrieve vast amounts of data efficiently. Blob Storage also ensures durability by providing redundancy across multiple data centers, ensuring that the data remains safe even in case of hardware failure. It supports multiple access tiers, allowing the application to optimize costs by choosing the appropriate storage tier based on the frequency of access. Azure Blob Storage integrates seamlessly with the application, making it easy to upload, retrieve, and manage large media files, improving the user experience while keeping storage costs manageable. Furthermore, its global distribution feature enables the application to access data from any region, ensuring low-latency access to media files. The ability to store both structured and unstructured data in the same platform provides a unified storage solution for the application. (Polkovnikov, 2024)

### **3) Azure Server**

**-Requirement**: I need a secure, scalable, and centralized environment to host the backend services, including database servers and applications, to ensure seamless connectivity and performance.

**-Service**: Azure Server

**-Motivation**: Azure Server provides a robust cloud-based infrastructure for hosting applications and databases. It offers flexible scalability, allowing the application to grow as traffic increases without needing to manually provision additional hardware. This scalable infrastructure is crucial for supporting high traffic volumes, especially during peak periods. Azure Server also ensures high security through built-in features such as encrypted data storage, access controls, and network security options. By hosting the database on Azure Server, I benefit from centralized access to data, simplified management, and seamless integration with other Azure services. It ensures that the backend is always available, secure, and capable of handling increased demands from users or the application. Additionally, Azure Server offers powerful monitoring and diagnostic tools, enabling proactive performance tuning and issue resolution. The ability to integrate seamlessly with other Azure resources ensures smooth management and troubleshooting across multiple services. *(Jain, 2019)*

### **4) Azure Queue Storage**

**-Requirement**: I need a reliable message queuing service to manage asynchronous communication between various components in my application, such as processing orders, managing notifications, or handling background tasks.

**-Service**: Azure Queue Storage

**-Motivation**: Azure Queue Storage is a highly efficient, scalable service designed to handle asynchronous tasks in distributed applications. It acts as a messaging queue, enabling different application components to communicate by sending and receiving messages, which are processed asynchronously. This decouples different parts of the system, enhancing scalability and reliability. For example, when a new order is placed, a message can be added to the queue, and background services can process it at their own pace without blocking the main user interface. Azure Queue Storage ensures reliable message delivery, even if components fail temporarily, and provides features like message retention, retry policies, and dead-letter queues for failed messages. This makes it an ideal choice for managing background tasks, transaction processing, and ensuring smooth, asynchronous communication between distributed application components. Its ability to scale to handle high volumes of messages ensures that the application can effectively process a large number of requests during peak times. Furthermore, Azure Queue Storage integrates seamlessly with other Azure services, such as Azure Functions, to trigger real-time processing and automation. *(Jain, 2019)*

### **5) Azure Files**

**-Requirement**: I need a shared file storage solution that enables my application to access and manage files across multiple users or services in a secure and scalable manner.

**-Service**: Azure Files

**-Motivation**: Azure Files is a fully managed file-sharing solution that allows applications to store and access files using standard SMB (Server Message Block) protocol or REST API. It provides a shared file system accessible by multiple applications, making it ideal for storing configuration files, logs, and other resources that need to be accessed by different services within the application. Azure Files offers high availability and redundancy by storing files across multiple servers, ensuring data integrity and accessibility. It integrates seamlessly with other Azure services, allowing for secure and centralized management of files. Additionally, Azure Files can be mounted directly on virtual machines, providing a consistent and secure way to share files across different application components, ensuring easy collaboration and reducing the need for complex file management. With built-in encryption and access control features, Azure Files guarantees the security of sensitive data, while its flexibility and scalability ensure it can support the growing needs of the application over time. (Polkovnikov, 2024)

### **6) Azure Functions**

**-Requirement**: I need a serverless compute solution that enables event-driven execution of code in response to specific triggers, such as file uploads or data processing tasks.

**-Service**: Azure Functions

**-Motivation**: Azure Functions offers a serverless compute model that allows the execution of event-driven code without requiring dedicated infrastructure management. This is perfect for running short, stateless tasks, such as processing images after they are uploaded to Azure Blob Storage or executing specific business logic in response to certain triggers (like an order being placed). Azure Functions automatically scales to meet the demands of incoming events, ensuring the application can handle varying workloads efficiently. This pay-as-you-go model makes it cost-effective for handling unpredictable workloads, while eliminating the overhead of provisioning and managing servers. By integrating Azure Functions with other Azure services, such as Blob Storage or Queue Storage, it enables seamless automation of tasks and workflows, reducing the need for manual intervention and improving application performance. Furthermore, its integration with monitoring tools allows easy tracking of execution performance, helping to optimize resource usage and detect any issues quickly. Azure Functions also supports multiple programming languages, giving flexibility in development while ensuring high performance and low latency for event processing. *(InterVision, 2023)*

**D. Identify alternative Azure technologies**

### **1) Azure Event Grid (Azure Service Bus)**

**Use Case:** For handling message brokering and event-driven architectures.

**Motivation:** While Azure Service Bus is designed for reliable message delivery with advanced features like message queuing and topic subscriptions, Azure Event Grid is an event routing service that efficiently handles high volumes of events in real-time. Event Grid would be a better alternative if the application requires lightweight, event-based communication with simpler integration to Azure services. Unlike Service Bus, which focuses more on guaranteed message delivery, Event Grid is optimized for quickly reacting to event-based triggers, making it ideal for modern, event-driven applications. Additionally, it integrates easily with various Azure services, such as Azure Functions, Logic Apps, and Webhooks, providing a seamless, scalable event processing solution. Moreover, Event Grid's ability to scale automatically based on demand ensures that high event loads can be handled without manual intervention, improving overall system efficiency.(*Polkovnikov, 2024)*

### **2) Azure Front Door (Azure CDN)**

**Use Case:** For content delivery and caching.

**Motivation:** Azure CDN is used for caching static content and improving web performance by distributing content globally. However, Azure Front Door provides a more advanced solution with global load balancing, application acceleration, and enhanced security, including Web Application Firewall (WAF), making it an ideal alternative for applications requiring both content delivery and traffic routing optimization. In addition to providing a content delivery network, Front Door’s global routing capabilities allow it to route traffic based on the nearest available endpoint, reducing latency and improving performance. The built-in security features such as WAF help mitigate common web application attacks, providing an extra layer of protection for applications. Furthermore, Azure Front Door is highly customizable with intelligent traffic management, allowing for a more tailored content delivery strategy that optimizes both the user experience and system resilience. *(Jain, 2019)*

### **3) Azure Disk Storage (Azure Blob Storage)**

**Use Case:** For storing persistent data.

**Motivation:** Azure Blob Storage is commonly used for unstructured data, like media files or logs. Azure Disk Storage, on the other hand, provides high-performance, durable block storage for virtual machines, making it an ideal alternative for persistent data that needs to be attached directly to VMs, particularly for databases or applications that require low-latency access to data. Unlike Blob Storage, which is best suited for unstructured data, Azure Disk Storage offers higher IOPS and throughput, making it the go-to solution for applications that need fast and reliable storage for virtual machines. It provides superior performance for workloads like databases or applications running in virtual machines that require low-latency access to storage. Additionally, it supports persistent data and allows for higher durability and reliability compared to traditional file storage, ensuring that data is always available even during VM restarts. *(Polkovnikov, 2024)*

### **4) Azure Machine Learning (Azure Cognitive Services)**

**Use Case:** For AI and machine learning tasks.

**Motivation:** Azure Cognitive Services provides ready-to-use APIs for tasks like image recognition, language understanding, and sentiment analysis, without needing in-depth knowledge of machine learning. However, for more complex or custom machine learning models, Azure Machine Learning is the better alternative. It allows for building, training, and deploying custom AI models, giving greater control over the process. Azure Machine Learning also offers an end-to-end data science and machine learning workflow, enabling users to prepare data, build models, train, and deploy them all within the same environment. Unlike Cognitive Services, which are best for simpler, pre-built models, Azure Machine Learning is designed for more sophisticated AI projects, allowing data scientists to fine-tune algorithms, experiment with different models, and manage versioned deployments. Additionally, its integration with Azure Databricks and other data services allows for collaborative machine learning development, helping teams build powerful AI models more efficiently. *(Polkovnikov, 2024)*

**Azure Load Balancer (Azure Traffic Manager)**

**Use Case:** For distributing network traffic and optimizing application performance across multiple regions.

**Motivation:** Azure Load Balancer is a highly available, low-latency solution for distributing incoming traffic among healthy virtual machines or services. However, Azure Traffic Manager offers more sophisticated traffic-routing capabilities, including DNS-based traffic distribution to different geographic regions. Traffic Manager enables multi-region deployment, which helps ensure that users are directed to the nearest available endpoint, improving latency and availability. Unlike Azure Load Balancer, which operates at the network layer, Traffic Manager uses DNS to distribute traffic across multiple Azure regions, enabling more flexible traffic routing strategies based on performance, priority, or geographic location. This makes it an ideal solution for applications that need to support global users and optimize performance by balancing load across multiple Azure regions. *(Microsoft, 2024)*

**Reference**

InterVision. “What Are the Service Categories Provided by Microsoft Azure?” *InterVision Systems*, 6 Apr. 2023, intervision.com/blog-what-are-the-service-categories-provided-by-microsoft-azure/. Accessed 7 Nov. 2024.

Jain, Neeru. “List of Top 10 Azure Services.” *Whizlabs Blog*, 27 Nov. 2019, www.whizlabs.com/blog/top-azure-services/. Accessed 6 Nov. 2024.

Microsoft. “Directory of Azure Cloud Services | Microsoft Azure.” *Azure.microsoft.com*, azure.microsoft.com/en-us/products, 2024. Accessed 6 Nov. 2024.

Polkovnikov, Alexey. “Azure Services Overview.” *Azurecharts.com*, azurecharts.com/overview, 2024 Accessed 6 Nov. 2024.

*Code Reference done on script*