CLDV6212 POE Part 3

By Zaara Salie_st10455456

Table of Contents

GitHub and YouTube Links

GitHub- https://github.com/ZSalie/CLDV6212_ABCRetail_Part3.git

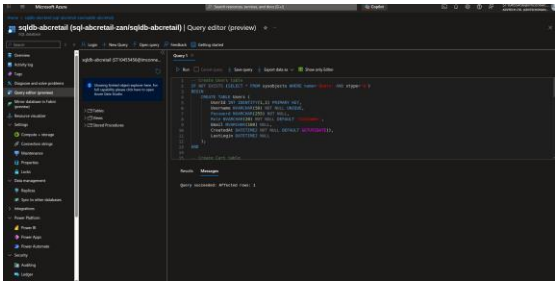
YouTube- https://youtu.be/RyFMWYBsqhM

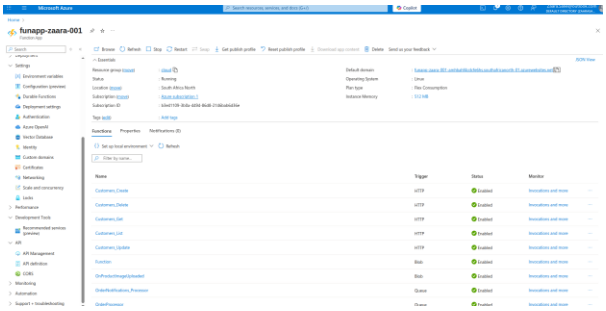
Section A

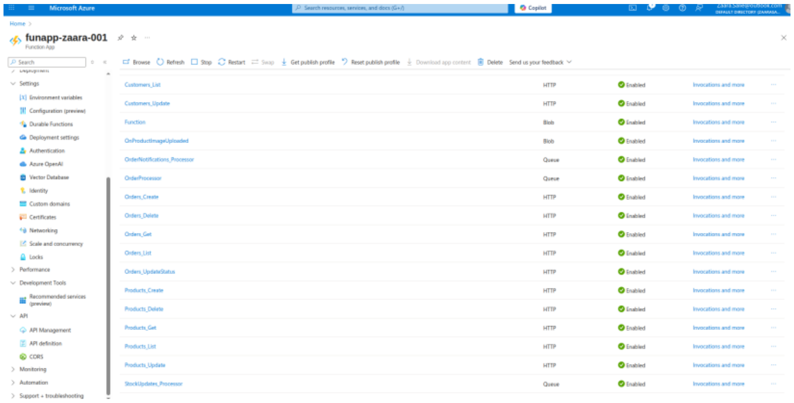
The first step in implementing the login functionality for ABC Retailers was the creation of an Azure SQL Database. This database serves as the central repository for authentication and user management. Within the Azure Portal, a new SQL Server was provisioned under the resource group ABC-Retailers-RG. The database was named abcretailerauthdb and hosted on a server called abcretailersqlserver. SQL authentication was enabled, with a secure administrator login and password configured. Networking settings were adjusted to allow Azure services and resources to access the server, ensuring that the web application could connect seamlessly (Microsoft, 2024a).

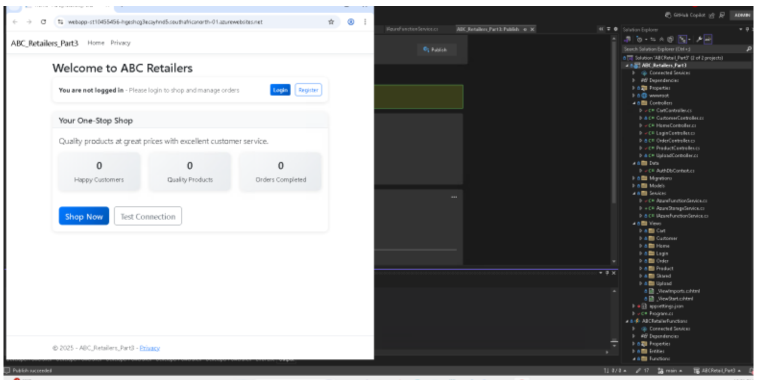
Once the database was created, tables were defined to support authentication and shopping cart functionality. A Users table was designed to store login credentials and roles, including fields such as UserID, Username, PasswordHash, and Role. This allowed the system to distinguish between customers and administrators. A Cart table was created to manage shopping cart items, linking each cart entry to the correct customer and product. An Orders table was also introduced to track purchases, including details such as OrderID, CustomerID, ProductID, OrderDate, and Status.

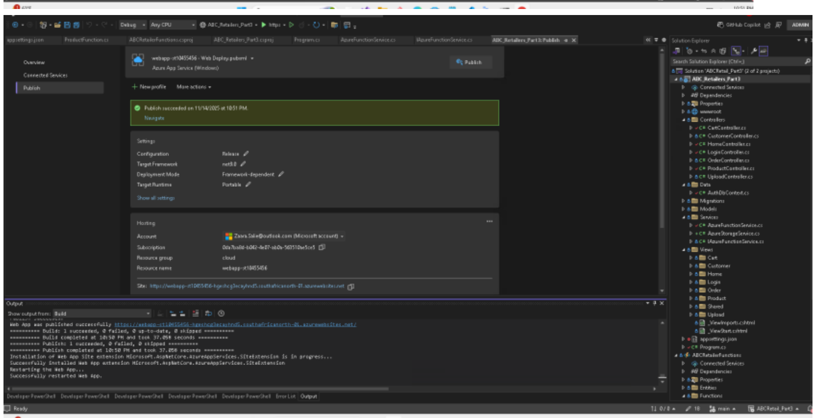
Sample data was inserted into the Users table to demonstrate functionality. At least one customer account and one administrator account were added, ensuring that both roles could be tested during the login process. For example, a record was created for a customer with the username customer1 and another for an administrator with the username admin1. Passwords were stored securely using hashing techniques, and roles were clearly defined to enforce access control (Microsoft, 2024a).

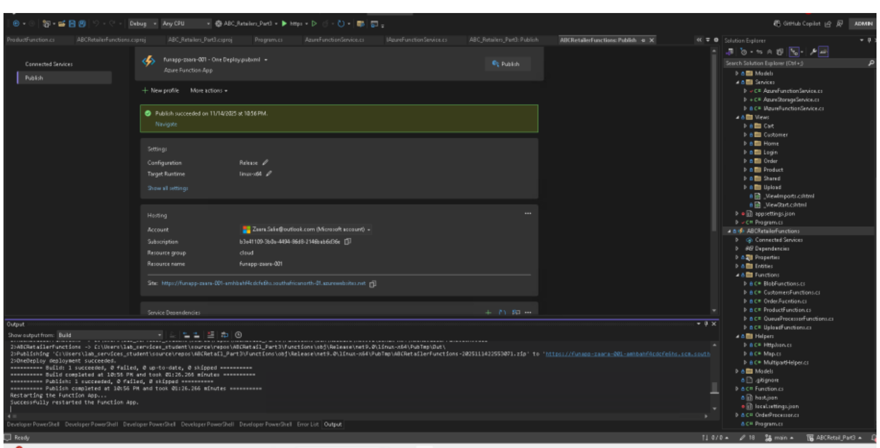





Microsoft Azure
funapp-zaara-001
Function App
Customers_List
Customers_Update
Function
OnProductImageUploaded
OrderNotifications_Processor
OrderProcessor
Orders_Create
Orders_Delete
Orders_Get
Orders_List
Orders_UpdateStatus
Products_Create
Products_Delete
Products_Get
Products_List
Products_Update
StockUpdates_Processor


ABC_Retailers_Part3
Welcome to ABC Retailers
Your One-Stop Shop
Quality products at great prices with excellent customer service.
Happy Customers
Quality Products
Orders Completed
Shop Now
Test Connection





Section B

Component	Technology Choice	Hosting Model	Justification
Web Application Hosting	Azure App Service	PaaS	GitHub, and a high-availability SLA, allowing developers to focus exclusively on application code (Microsoft, 2024c).
Relational Database	Azure SQL Database	PaaS	Serves as the managed, secure, and highly available central data store. It automates critical tasks like backups, patching, and performance tuning (including automatic indexing), which reduces administrative overhead and ensures data integrity (Microsoft, 2024a).
Unstructured Data Storage	Azure Blob Storage	PaaS	Offers a highly scalable and cost-effective service for storing unstructured data such as product images and user-uploaded documents. It provides high durability and integrates seamlessly with a Content Delivery Network (CDN) for fast global content delivery (Microsoft, 2024d).
Serverless Compute	Azure Functions	PaaS (FaaS)	Enables execution of event-driven code without server management. Ideal for background tasks like processing images upon upload or sending confirmation emails. The consumption-based pricing model ensures cost-efficiency for intermittent workloads (Microsoft, 2024e).
Application Messaging	Azure Service Bus	PaaS	Decouples application components through reliable message queuing. This asynchronous pattern enhances application resilience and scalability, ensuring that processes like order fulfillment can handle traffic spikes without becoming a bottleneck (Microsoft, 2024f).
Identity and Access Management	Microsoft Entra ID	SaaS	Acts as the centralized identity provider, managing user authentication and authorization. It enables enterprise-grade security features like Single Sign-On (SSO) and Multi-Factor Authentication (MFA) without the need for custom, complex identity code (Microsoft, 2024g).

Section C

To ensure scalability, resilience, and performance, several Azure services were integrated into the architecture. The application was deployed on Azure App Service, which provided automatic scaling, built-in load balancing, and a 99.95 percent uptime SLA. This ensured that the web application remained responsive even under varying user loads (Microsoft, 2024c).

The Azure SQL Database was configured with geo-replication to support business continuity and disaster recovery. In the event of a regional outage, geo-replica could be promoted to the primary database, reducing downtime and data loss. This also enabled global read scalability, as read-only queries could be directed to the replica (Microsoft, 2024b).

Azure Blob Storage was used to store product images and other static content. By integrating Blob Storage with Azure CDN, images could be delivered quickly to users worldwide, improving the customer experience (Microsoft, 2024d). Azure Functions were introduced to handle background tasks such as sending order confirmation emails and generating image thumbnails. This serverless approach reduced infrastructure overhead and increased development agility (Microsoft, 2024e).

Finally, Azure Service Bus was implemented to decouple the web front-end from the backend order-processing logic. By introducing asynchronous messaging, the application

became more resilient to traffic spikes and component failures, ensuring that orders were processed reliably without being lost (Microsoft, 2024f).

Section D

Several alternative services were evaluated during the design process. Azure Kubernetes Service (AKS) was considered an alternative to App Service, offering granular control over containerised microservices. However, the operational complexity was significantly higher (Microsoft, 2024h). Azure Container Apps provided a middle ground, simplifying orchestration while retaining container benefits (Microsoft, 2024i).

For the database layer, Azure Database for PostgreSQL was considered an alternative to Azure SQL Database. This option would be suitable if the development team preferred PostgreSQL's extensions and ecosystem (Microsoft, 2024j). SQL Server on Azure Virtual Machines was also evaluated, offering full control over the SQL Server instance but requiring additional administrative overhead to manage the operating system and database software (Microsoft, 2024k).

In terms of messaging, Azure Storage Queues were considered a simpler and more cost-effective alternative to the Service Bus. While suitable for high-volume, basic messaging scenarios, they lacked advanced features such as topics, subscriptions, and message sessions, which were deemed necessary for the project (Microsoft, 2024l).

References

Microsoft (2024a) Azure SQL Database. Available at: https://azure.microsoft.com/en-us/products/azure-sql/database/ (Accessed: 14 October 2024).

Microsoft (2024b) Overview of active geo-replication. Available at: https://learn.microsoft.com/en-us/azure/azure-sql/database/active-geo-replication-overview?view=azuresql (Accessed: 14 October 2024).

Microsoft (2024c) Azure App Service. Available at: https://azure.microsoft.com/en-us/products/app-service (Accessed: 14 October 2024).

Microsoft (2024d) Azure Blob Storage. Available at: https://azure.microsoft.com/en-us/products/storage/blobs (Accessed: 14 October 2024).

Microsoft (2024e) Azure Functions. Available at: https://azure.microsoft.com/en-us/products/functions (Accessed: 14 October 2024).

Microsoft (2024f) Azure Service Bus. Available at: https://azure.microsoft.com/en-us/products/service-bus (Accessed: 14 October 2024).

Microsoft (2024h) Azure Kubernetes Service (AKS). Available at: https://azure.microsoft.com/en-us/products/kubernetes-service (Accessed: 14 October 2024).

Microsoft (2024i) Azure Container Apps. Available at: https://azure.microsoft.com/en-us/products/container-apps (Accessed: 14 October 2024).

Microsoft (2024j) Azure Database for PostgreSQL. Available at: https://azure.microsoft.com/en-us/products/postgresql (Accessed: 14 October 2024).

Microsoft (2024k) SQL Server on Azure Virtual Machines. Available at: https://azure.microsoft.com/en-us/products/virtual-machines/sql-server (Accessed: 14 October 2024).

Microsoft (2024l) Azure Storage Queues. Available at: https://azure.microsoft.com/en-us/products/storage/queues (Accessed: 14 October 2024).