Assignment 1

Cloud Development

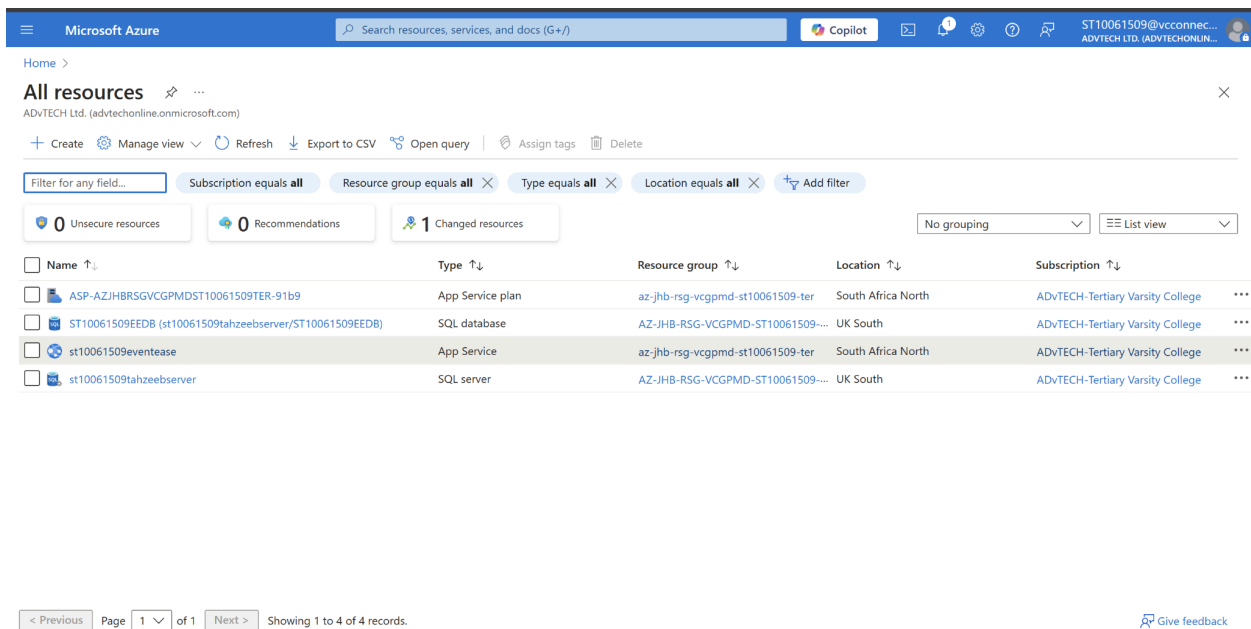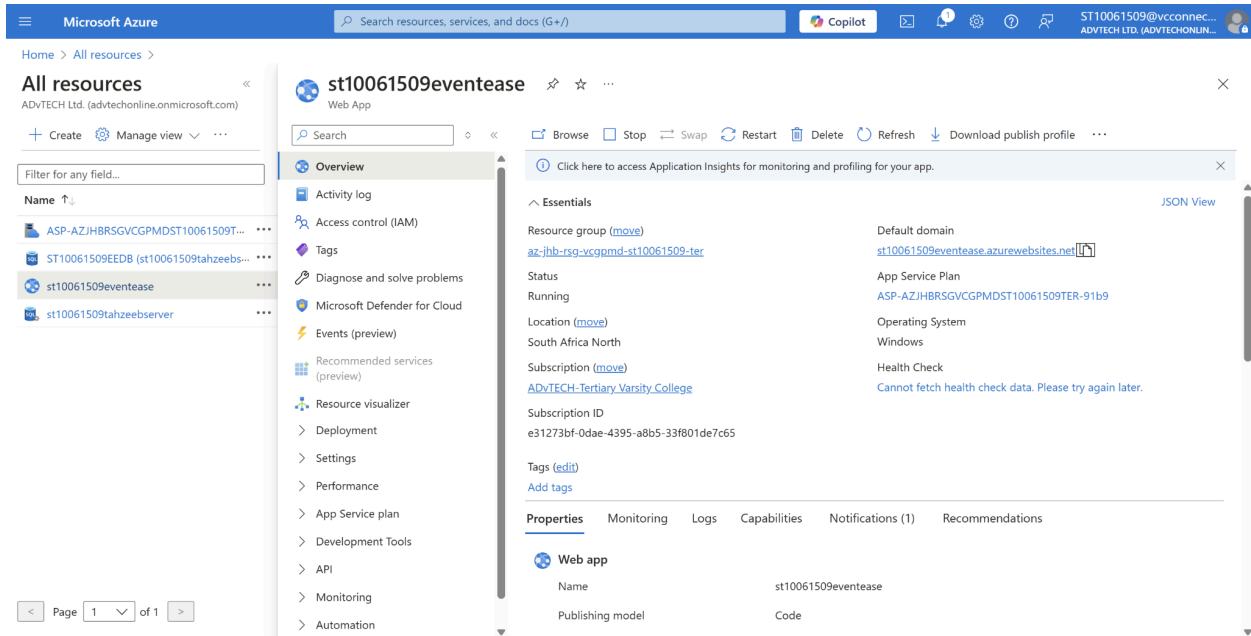CLDV6211

ST10061509

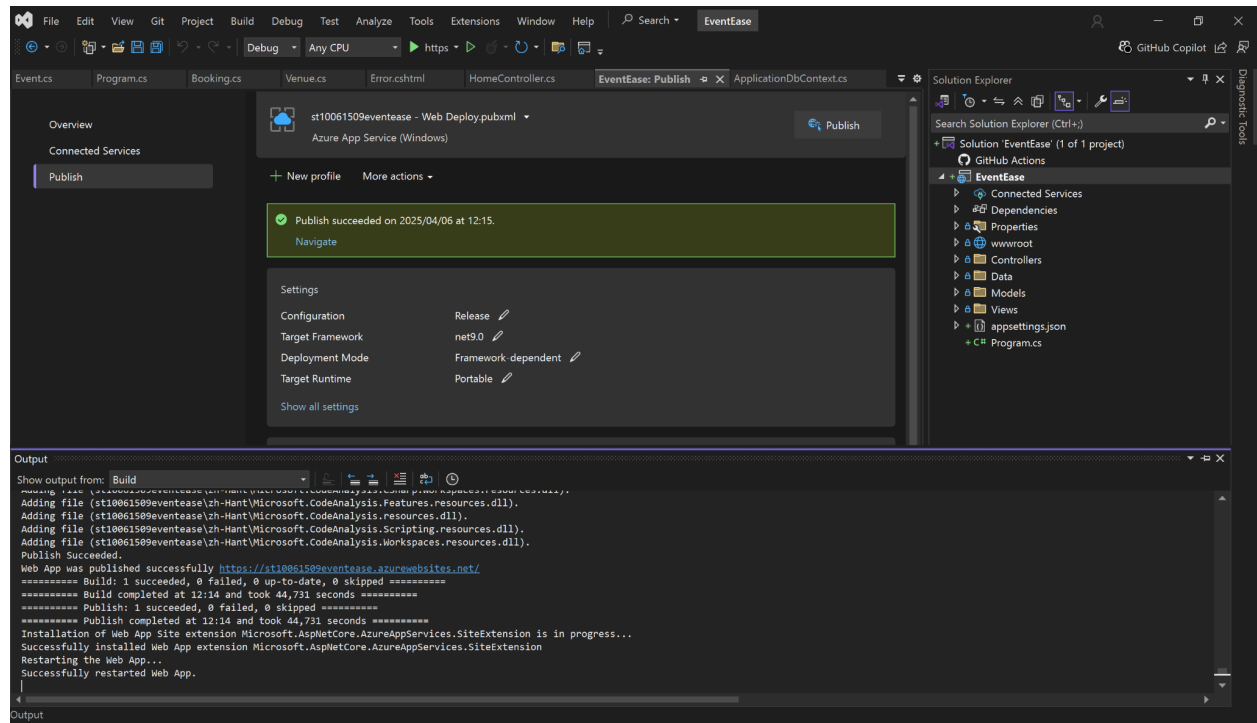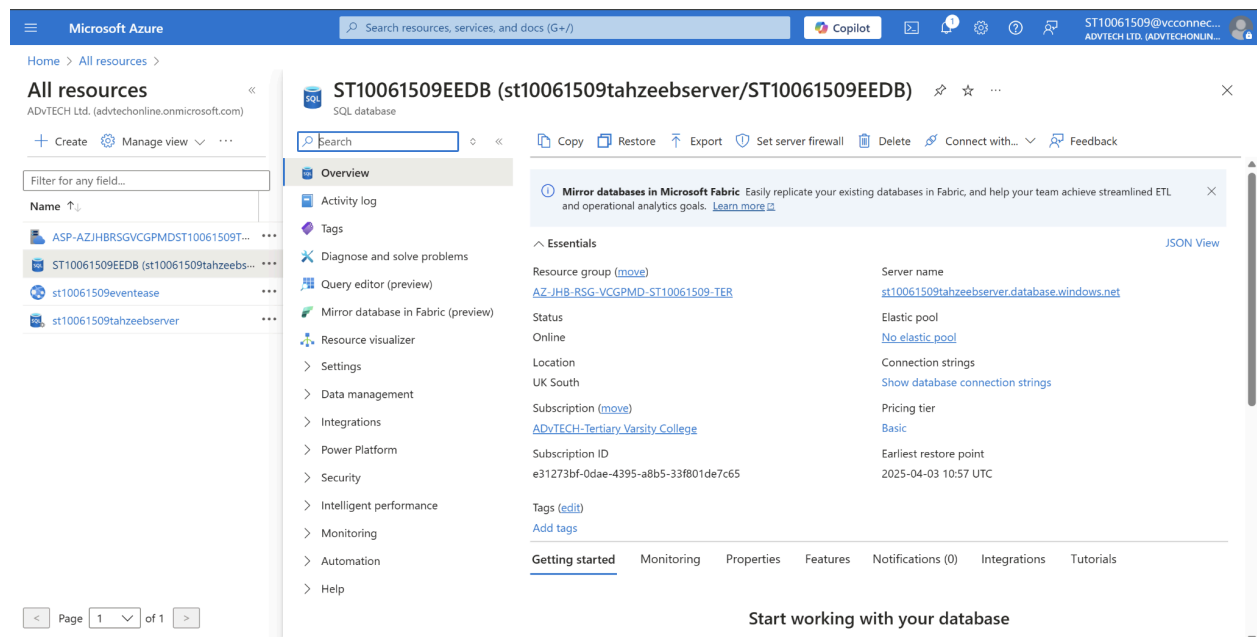Mohammed Moosa

Screenshots.

Screenshot of service created in portal:

Screenshot of successful deployment message in Visual Studio:

Screenshot of database in Azure portal:

Screenshots of tables inside the query editor:

Query 1 ✕   dbo.Bookings ✕   dbo.Events ✕

›☰ Create New Row   💾 Save   ↻ Refresh   ↺ Discard   🗑 Delete Row

🔍 Search to filter items...

EventId	Name	Description	StartDate	EndDate
1	Tech Summit 2025	Annual tech confer...	2025-06-15 00:00:0...	2025-06-16 00:00:0...
2	Business Networking	Professional networ...	2025-07-01 00:00:0...	2025-07-01 00:00:0...
3	Music Festival	Live performances ...	2025-08-10 00:00:0...	2025-08-12 00:00:0...

◀ ━━━━━━━━━━━━━━━━ ▶

Query 1 ✕   dbo.Bookings ✕   dbo.Events ✕   dbo.Venues ✕

›☰ Create New Row   💾 Save   ↻ Refresh   ↺ Discard   🗑 Delete Row

🔍 Search to filter items...

Id	Name	Location	Capacity	ImageUrl
1	Grand Hall	Johannesburg	510	https://via.placehol...
2	Conference Room A	Cape Town	200	https://via.placehol...
3	Outdoor Garden	Durban	1000	https://via.placehol...

◀ ━━━━━━━━━━━━━━━━ ▶

Website URL.

https://st10061509eventease.azurewebsites.net/

GitHub URL.

https://github.com/IIEWFL/cldv6211-part-1-ST10061509-Mohammed-Moosa

Theory Questions.

1. What are the factors that make it different to deploy an application on cloud versus on premise as far as security is concerned and the speed of deployment and the resource management?

The security, deployment speed, as well as the management of resources are all different when you are deploying to the cloud compared to traditional on premises deployment. In order to work in the on-premises environment, an organization is initially responsible for purchasing/obtaining hardware to run the control plane, the application plane, and any other needed services; managing security by themselves; assigning all the resources themselves; etc. Frequently, this approach is expensive, slow, and subject to human error. Let's say EventEase were to configure their venue booking system to be hosted on-premises: They would need to have physical servers that are in place with the data backed up by them, a secure network throughout and regular monitoring to ensure uptime – all of which will require an actual dedicated IT staff and infrastructure.

Where cloud deployment comes in handy is in terms of its better version of security with features like encryption, access control, automatic updates and real time threat detection. Since cloud providers like Microsoft Azure or Amazon Web Services (AWS) invest so heavily in security, any feature in cloud security is an advantage in them and it provides features such as role based access control (RBAC), secure storage, and compliances on international standards. It allows a company such as EventEase to protect customer and booking data without having to build complex security systems from scratch.

An important advantage is deployment speed. Continuous integration and continuous deployment (CI/CD) allows developers to deploy their applications on cloud platforms in minutes. Once I started making changes, I was able to spool along in an even faster way due to this agility of just making changes to the router. Even more importantly, the cloud is much more efficient in doing resource management with the help of auto scaling and pay as you go pricing models. With EventEase, you avoid guessing capability needs in hopes of over- or under-provisioning, and can have the application scale up or down as needed based on actual usage. All in all, cloud has an ability to perform faster, securely and at a much larger scale than the traditional on premises installations.

2. So what are the fundamental differences between IaaS, PaaS, and SaaS and why could EventEase derive an advantage in using PaaS as we create our application?

Infrastructure as a service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS) are all types of cloud computing services and they each provide the level of control and the level of responsibility at different levels. The most control of all is established by IaaS, which provides Internet based virtualized computers. With IaaS, providers keep hardware and networking and users manage OS, middleware, and applications. An example would be Azure Virtual Machines, albeit EventEase would still need to install and manage the software, manage updates and perform backups.

Software as a Service (SaaS) has the lowest control but the highest convenience. Some of these apps in SaaS (SaaS) are fully managed by the provider and can be accessed through a browser or an app including Microsoft 365 or Google Workspace. While SaaS suits well with common business tools, EventEase's booking system is a customized platform requiring specially written logic and workflows, and therefore it makes no sense to host such tailor made logic and workflows on a SaaS platform.

PaaS is the right solution on the line between the two and fits perfectly for EventEase. In PaaS, the cloud provider takes care of the infrastructure, the OS and the platform software, leaving the development team's work on building and deploying the application only. Tools such as Azure App Service or Google App Engine come with built in development features, forum, cloud database integration, as well as automated backups. For EventEase's goals, PaaS is perfect since it takes care of all the overhead of managing the infrastructure, which allows us to develop a scalable, secure, and user friendly application. Since EventEase is growing, PaaS helps with seamless scaling and pulling in the added value of the other cloud services, thus gaining a long term advantage compared to the IaaS or SaaS model.

Code Attribution.

```
/*
 *
 * Project Name : EventEase
 * Description  : A web application for managing events, venues, and bookings.
 *                Built with ASP.NET Core MVC and Entity Framework Core.
 *
 * Code Attribution:
 * - ASP.NET Core and Entity Framework Core:
 *     © Microsoft Corporation. Licensed under the MIT License.
 *     Documentation and code samples available at
 https://docs.microsoft.com/aspnet/core
 *
 * - Bootstrap (if used in Views):
 *     © The Bootstrap Authors. Licensed under the MIT License.
 *     https://getbootstrap.com/
 *
 * - jQuery (if used in Views):
 *     © The jQuery Foundation. Licensed under the MIT License.
 *     https://jquery.com/
 *
 * - Any scaffolding code or templates used are generated using Visual Studio's
 *   built-in ASP.NET Core MVC scaffolding tools provided by Microsoft.
 *
 * - Code snippets and best practices may be adapted from:
 *     - Microsoft Learn (https://learn.microsoft.com/)
 *     - Stack Overflow (https://stackoverflow.com/) — individual answers credited in
 comments where used.
 *     - GitHub sample repositories (open source, MIT-licensed).
 *
 *
 */
```

Theory References.

Amazon Web Services (AWS). (2023) *What is Cloud Computing?*. Available at:
https://aws.amazon.com/what-is-cloud-computing/ (Accessed: 6 April 2025).

Microsoft Azure. (2023) *What is Azure App Service?*. Available at:
https://learn.microsoft.com/en-us/azure/app-service/overview (Accessed: 6 April 2025).

Microsoft. (2023) *What is Role-Based Access Control (RBAC)?*. Available at:
https://learn.microsoft.com/en-us/azure/role-based-access-control/overview (Accessed: 6 April 2025).

Google Cloud. (2023) *App Engine Documentation*. Available at:
https://cloud.google.com/appengine/docs (Accessed: 6 April 2025).

Armbrust, M. et al. (2010) 'A View of Cloud Computing', *Communications of the ACM*, 53(4), pp. 50–58.

Rittinghouse, J.W. and Ransome, J.F. (2017) *Cloud Computing: Implementation, Management, and Security*. 2nd edn. Boca Raton: CRC Press.