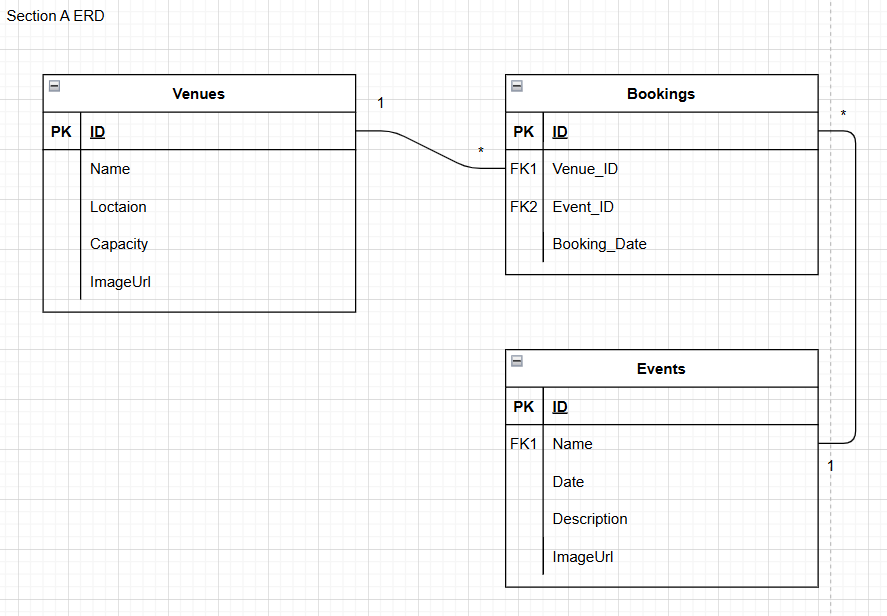
**CLDV6211 POE Part1**

**Tapiwa Sango ST10276025**

1. **Database Design**



ERD Explaination  
  
1. Venues (1) ↔ (0..\*) Bookings

One venue can be used for many bookings.

Each booking must be linked to one specific venue.

Represented via VenueId in the Bookings table.

2. Events (1) ↔ (0..\*) Bookings

One event can have multiple bookings, possibly in different venues or at different times.

Each booking is for one specific event.

Represented via EventId in the Bookings table.

Database script

-- Drop existing EventEaseDB if it exists

USE master;

IF EXISTS (SELECT \* FROM sys.databases WHERE name = 'EventEaseDB')

DROP DATABASE EventEaseDB;

GO

-- Create a fresh database

CREATE DATABASE EventEaseDB;

GO

USE EventEaseDB;

GO

-- Create Venues table

CREATE TABLE Venues (

Id INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

Name NVARCHAR(100) NOT NULL,

Location NVARCHAR(255) NOT NULL,

Capacity INT NOT NULL,

ImageUrl NVARCHAR(255) NULL

);

GO

-- Create Events table

CREATE TABLE Events (

Id INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

Name NVARCHAR(100) NOT NULL,

Date DATE NOT NULL,

Description NVARCHAR(500) NULL,

ImageUrl NVARCHAR(255) NULL

);

GO

-- Create Bookings table with foreign key relationships

CREATE TABLE Bookings (

Id INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

EventId INT NOT NULL,

VenueId INT NOT NULL,

BookingDate DATE NOT NULL,

FOREIGN KEY (EventId) REFERENCES Events(Id),

FOREIGN KEY (VenueId) REFERENCES Venues(Id)

);

GO

-- Insert sample data into Venues

INSERT INTO Venues (Name, Location, Capacity, ImageUrl)

VALUES

('Conference Hall A', 'Cape Town Convention Center', 300, 'https://example.com/hallA.jpg'),

('Open Air Grounds', 'Joburg Park', 800, 'https://example.com/park.jpg');

GO

-- Insert sample data into Events

INSERT INTO Events (Name, Date, Description)

VALUES

('Tech Summit 2025', '2025-09-15', 'A premier gathering of tech leaders.'),

('Startup Pitch Night', '2025-10-05', 'An evening of startup innovation and networking.');

GO

-- Insert sample data into Bookings

INSERT INTO Bookings (EventId, VenueId, BookingDate)

VALUES

(1, 1, '2025-08-01'),

(2, 2, '2025-09-20');

GO

-- View the tables

SELECT \* FROM Venues;

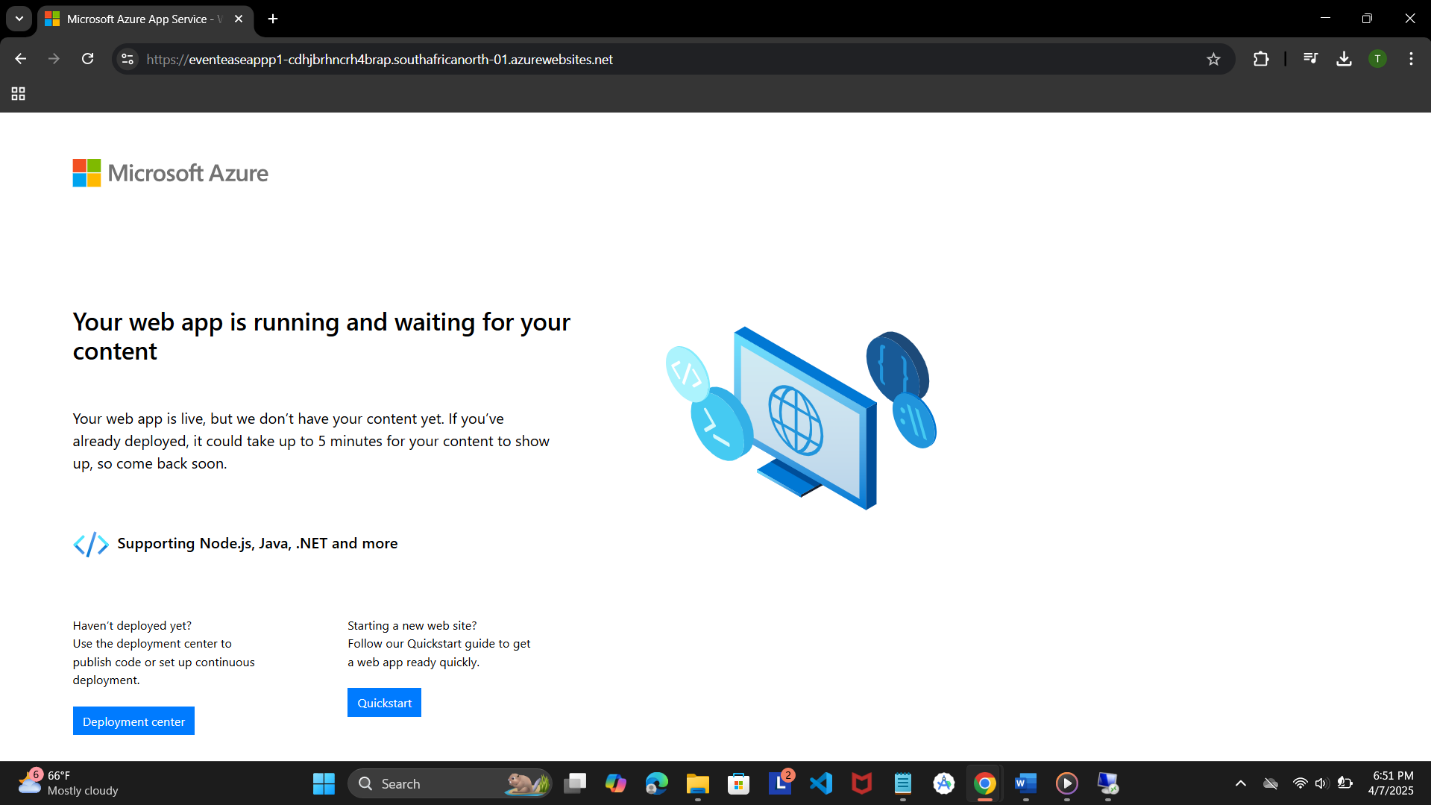
SELECT \* FROM Events;

SELECT \* FROM Bookings;

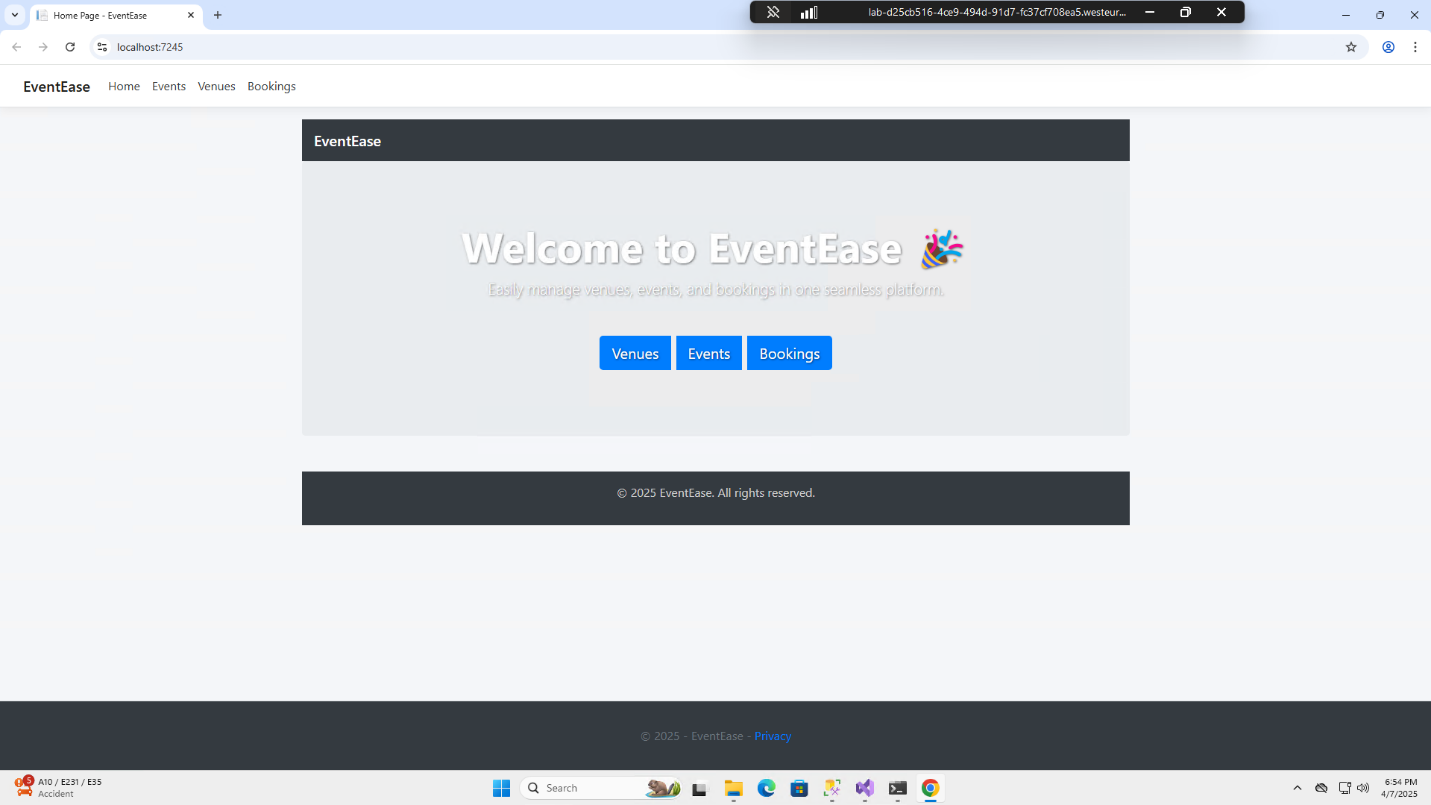
1. **Developed Web Application**

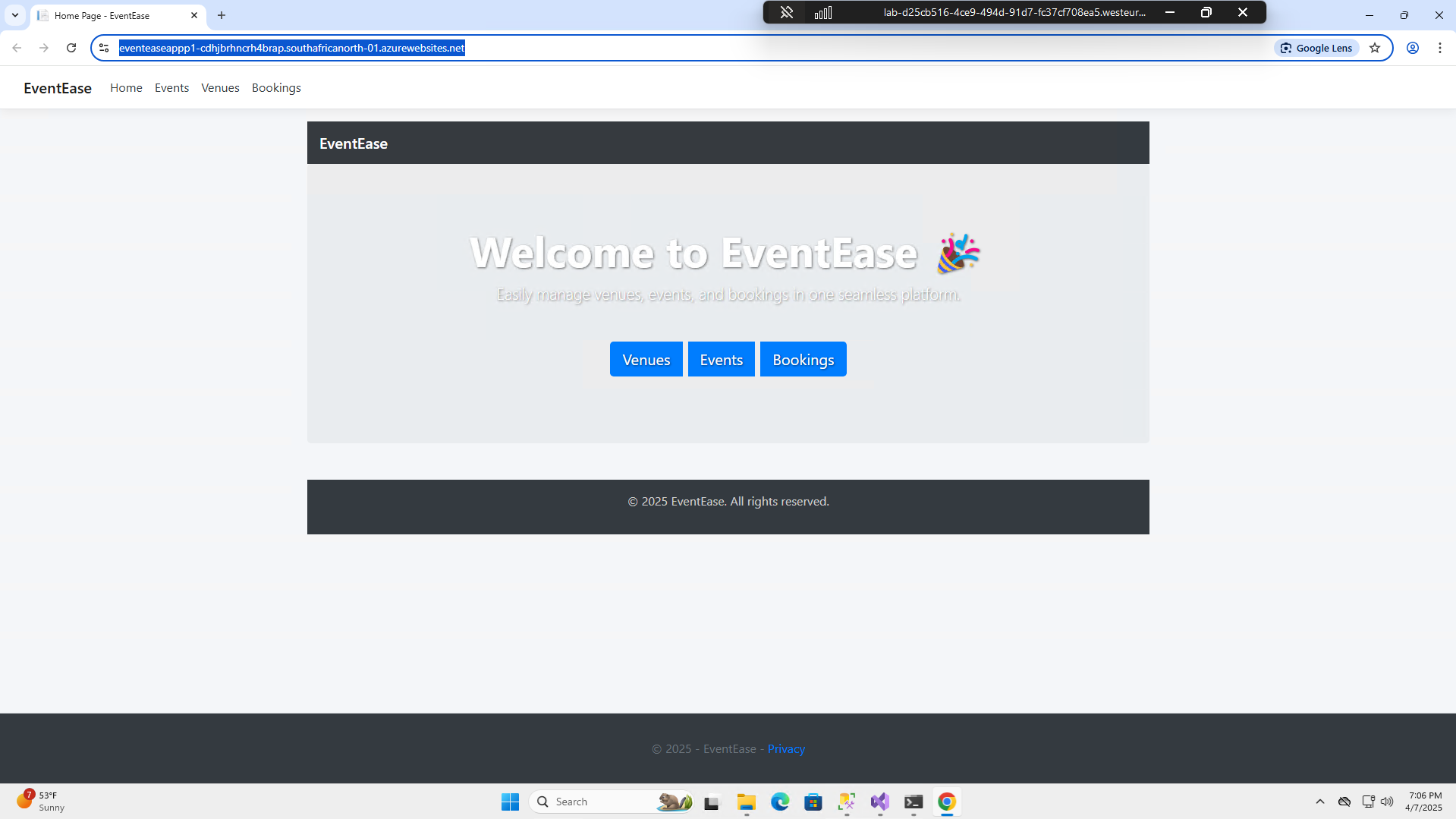
<https://github.com/IIEMSA/part-1-poe-Sango-Tapiwa>

1. **Deployment of Database and Web App onto Azure**



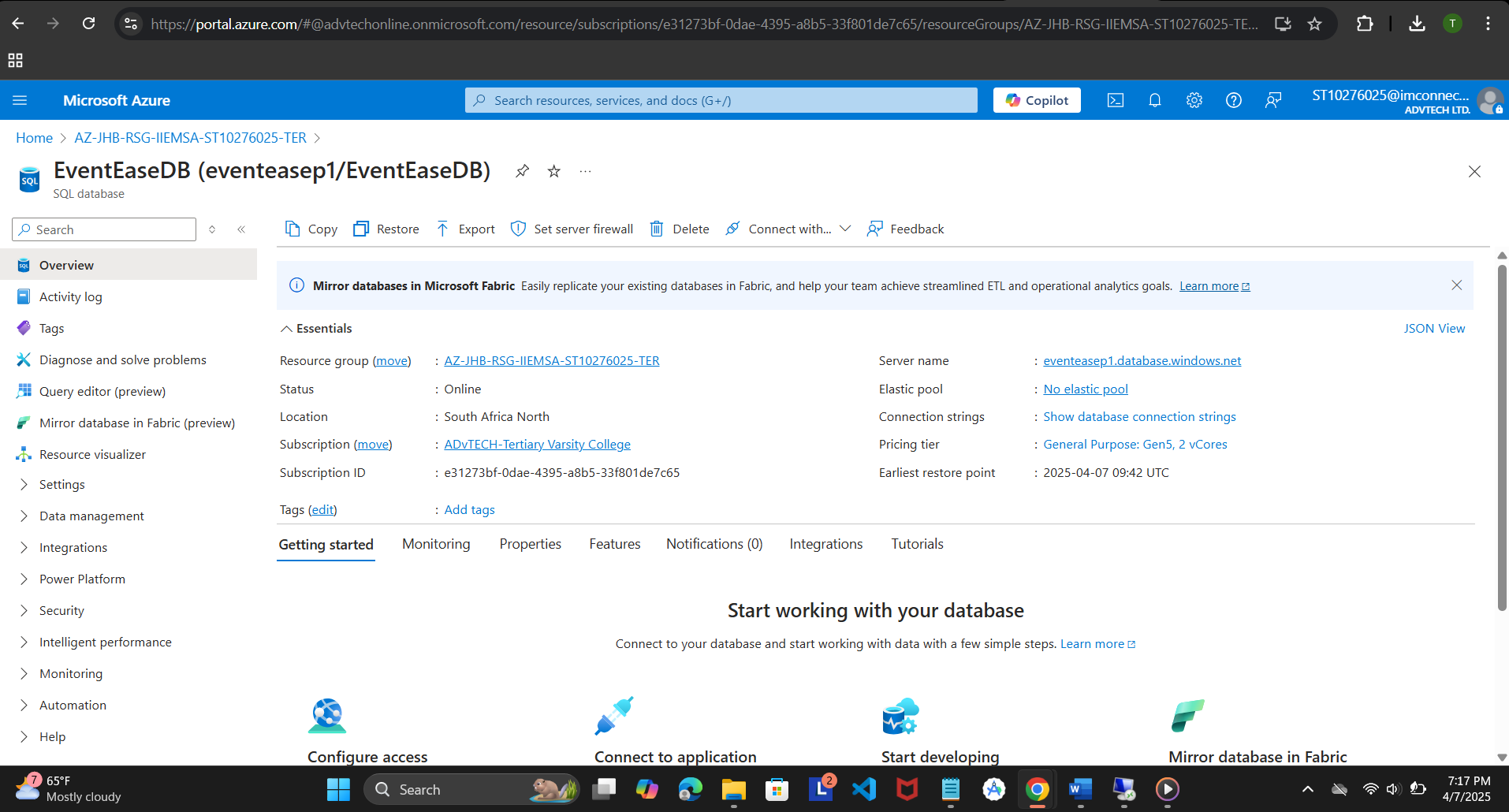
Web App Service created in portal





Successful deployment in Visual Studio

<https://eventeaseappp1-cdhjbrhncrh4brap.southafricanorth-01.azurewebsites.net/>



SQL database created in Azure Portal  
  
A computer screen with a computer screen

AI-generated content may be incorrect.

Migrated data and tables into query editor

A computer screen with a computer screen

AI-generated content may be incorrect.

A computer screen with a white screen

AI-generated content may be incorrect.

1. **Cloud Computing Basics**
2. Distinctions Between On-Premises and Cloud Deployment  
   There are notable differences between delivering an application on-premises and in the cloud, particularly with regard to resource management, deployment speed, and security:

* Security:
  + With on-premises solutions, the company must oversee the whole security stack, from application-level security to physical infrastructure. This covers user access control, patch management, and firewalls (Rittinghouse & Ransome, 2017). On the other hand, users are in charge of application and data-level security, while cloud providers such as Microsoft Azure handle infrastructure-level security. This shared responsibility paradigm lowers the possibility of physical breaches while increasing overall efficiency.
* Deployment Speed:
  + Setting up infrastructure for on-premises deployment, which includes hardware purchasing, configuration, and testing, can take a lot of time. By providing ready-to-use virtual environments that can be created in a matter of minutes, cloud platforms remove this latency (Hussain et al., 2021). For instance, by integrating GitHub CI/CD with Azure App Service, EventEase may swiftly introduce a new feature.
* Resource Management:
  + Due to fixed hardware capacity limitations, on-premises solutions scale slowly and expensively. Elastic scalability is made possible by cloud computing, in which resources are distributed automatically in response to demand (Armbrust et al., 2010). Cloud services can automatically scale EventEase to preserve performance in the event of a popular event that causes a spike in traffic.

1. Difference between on premises and cloud deployments

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Provision | Example | Who manages what |
| IaaS | Virtual machines, storage, and networks | Azure Virtual Machines | User manages OS, runtime, app |
| PaaS | |  | | --- | |  |  |  | | --- | | Infrastructure and platform tools for building apps | | Azure App Service, Azure SQL Database | User manage the app and data |
| SaaS | Ready-to-use applications | Microsoft 365, Google Drive | Provider manages everything |

**Why PaaS is the Best for EventEase**

One cloud computing architecture that provides a full development and deployment environment in the cloud is called Platform as a Service (PaaS). Database management systems, development tools, middleware, business intelligence services, and infrastructure (servers, storage, and networking) are all included. PaaS offers a startup like EventEase the best possible mix of control, flexibility, cost, and speed when creating a web application for venue booking.

1. Pay attention to application logic rather than infrastructure.  
PaaS isolates the infrastructure layer, in contrast to Infrastructure as a Service (IaaS), which requires developers to set up and manage virtual machines. Instead of manually handling patch updates, OS configurations, or infrastructure scaling, EventEase may now concentrate entirely on writing code, developing features, and enhancing user experience (Rittinghouse & Ransome, 2017).  
  
Azure App Service, for instance, enables developers to launch.NET Core web applications straight from GitHub or Visual Studio. The solution significantly cuts down on setup and maintenance time by handling runtime updates, security patches, and hosting (Microsoft, 2023).

2. Quicker Time to Market  
By offering integrated tools for version control, CI/CD, application monitoring, and testing environments, PaaS speeds up the development lifecycle. This enables EventEase to build, test, and launch new features more quickly, such as calendar integration, venue availability, or user authentication.  
  
Automated deployments are made possible by integrated CI/CD pipelines, like Azure DevOps or GitHub Actions. The majority of PaaS systems now enable these technologies, which decreases human error during deployment and expedites release cycles (Hussain et al., 2021).

3. Flexibility and Scalability  
Traffic and user demand will rise as EventEase expands, particularly during the busiest event seasons. Because PaaS facilitates automatic horizontal scaling, EventEase can manage increased demands without requiring human infrastructure upgrades (Armbrust et al., 2010).  
  
For example, Azure App Service can generate more instances of the application automatically in response to incoming HTTP traffic or CPU consumption. This guarantees that there is no downtime or performance deterioration and that the program stays responsive.

4. Economical for Small Groups  
Startups and student-led projects with tight resources benefit from PaaS's predictable and controllable pricing. Without having to pay for complicated maintenance procedures or physical gear, EventEase may choose a pricing tier that best suits its present requirements and scale up as the company expands.  
  
PaaS offers the ideal blend of control and simplicity when compared to SaaS, which allows less flexibility, or IaaS, which frequently necessitates maintaining VM backups, patching, and networking.

5. Effortless Service Integration  
Databases, AI, identity services, and APIs are just a few of the cloud services that PaaS systems have native connections with. EventEase may use Azure Active Directory for safe user login or connect to an Azure SQL database to store reservations. These integrations improve security and functionality, save development time, and are easy to set up (Microsoft, 2023).

**Reference List:**

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M., 2010. *A view of cloud computing*. Communications of the ACM, 53(4), pp.50–58.
2. Hussain, F.K., Hussain, O.K. and Chang, E., 2021. *Cloud Computing: Theory and Practice*. 3rd ed. Burlington: Morgan Kaufmann.
3. Microsoft, 2023. *What is Azure App Service?* [online] Available at: <https://learn.microsoft.com/en-us/azure/app-service/overview> [Accessed 5 Apr. 2025].
4. Rittinghouse, J.W. and Ransome, J.F., 2017. *Cloud Computing: Implementation, Management, and Security*. 2nd ed. Boca Raton: CRC Press.
5. Google Drive, 2025. *Cloud computing resources folder*. [online] Available at: <https://drive.google.com/drive/folders/1ewgeCIH-Zx8MFqBUDnKZKJPxYLlHNgiH?usp=sharing> [Accessed 1 Apr. 2025].
6. Google Drive, 2025. *Cloud deployment case study document*. [online] Available at: <https://drive.google.com/file/d/1XM_gLWRX3ba2IPSSBZAK4i1j5BZ5s8YU/view?usp=sharing> [Accessed 5 Apr. 2025].