Architecture Style

I would use a 3-tier architecture:

1. Presentation Layer – The front end (web app) built with ASP.NET Core MVC where users can interact with Venues, Events, and Bookings.

2. Application Layer – This will include controllers, models, and logic to handle CRUD operations.

3. Data Layer – This connects to an Azure SQL Database where all data (Venues, Events, Bookings) is stored.

This structure makes the app easier to manage and scale.

Compute

For compute, I would use Azure App Service:

- It lets me deploy the ASP.NET Core MVC web app easily to the cloud.
- It's scalable, so if traffic increases, the app can handle it without downtime.
- It's also simple to integrate with other Azure services like Azure SQL Database and Blob Storage.

Storage

I would use two types of storage:

1. Azure SQL Database for:
    - Storing structured data like Venue, Event, and Booking tables.
    - It's reliable, scalable, and supports Entity Framework easily.
2. Azure Blob Storage for:

    - Storing image files linked to venues and events using ImageUrl.
    - This keeps the database light and the media in the right place.

Messaging

While messaging wasn't directly required in Part 1, I would plan for Azure Service Bus or Azure Queue Storage in future phases for:

- Sending notifications or reminders for upcoming bookings/events.
- Handling background tasks like sending confirmation emails or SMS asynchronously.