**Cloud Development Assignment - Part 2**

# **Student: Keagile Chaane | Student No: ST10250839**

# **Part 2: Enhancing Functionality and Integrating Cloud Storage**

## **1. Error Handling and Validation**

### a. Preventing Double Bookings

To maintain data integrity and ensure a venue cannot be double-booked for the same date:  
  
Validation Rule (in Booking Controller or Service Layer): Before saving a new booking, check if a booking already exists for the selected venue on the chosen date.

bool isAlreadyBooked = \_context.Bookings  
 .Any(b => b.VenueId == booking.VenueId && b.BookingDate.Date == booking.BookingDate.Date);  
  
if (isAlreadyBooked)  
{  
 ModelState.AddModelError("", "This venue is already booked on the selected date.");  
 return View(booking);  
}

### b. Restricting Deletion of Active Venues/Events

To avoid inconsistencies and user frustration from deleted venues or events tied to active bookings:

var hasActiveBookings = \_context.Bookings  
 .Any(b => b.VenueId == venueId && b.Status != "Cancelled");  
  
if (hasActiveBookings)  
{  
 TempData["AlertMessage"] = "This venue cannot be deleted because it is associated with active bookings.";  
 return RedirectToAction("Index");  
}

## 2. Image Management with Azure Blob Storage

Purpose: Store uploaded images (e.g., venue photos or event banners) securely in Azure Blob Storage instead of using static placeholder URLs.

Steps Implemented:

1. Create a Blob Storage Account in Azure with a container named 'venueimages'.

2. Install Azure.Storage.Blobs NuGet Package:  
 dotnet add package Azure.Storage.Blobs

3. Configure appsettings.json:

{  
 "AzureBlobStorage": {  
 "ConnectionString": "Your\_Azure\_Storage\_Connection\_String",  
 "ContainerName": "venueimages"  
 }  
}

4. Image Upload Logic in Controller:

var containerClient = new BlobContainerClient(\_configuration["AzureBlobStorage:ConnectionString"], \_configuration["AzureBlobStorage:ContainerName"]);  
var blobClient = containerClient.GetBlobClient(file.FileName);  
  
using (var stream = file.OpenReadStream())  
{  
 await blobClient.UploadAsync(stream, **overwrite** true);  
}  
  
venue.ImageUrl = blobClient.Uri.ToString();

## 5.image upload logic in controller

The uploaded images URL is stored in the venue. ImageUrl or Event. ImageUrl field, allowing dynamic rendering on the frontend.

## 3. Enhanced Display and Search

To enhance clarity for admin or booking specialists, a consolidated dashboard is developed:

Features:  
- List all bookings with venue name, event name, date, and status.  
- Display bookings grouped by status: Confirmed, Pending, or Cancelled.

var bookings = \_context.Bookings  
 .Include(b => b.Event)  
 .Include(b => b.Venue)  
 .OrderByDescending(b => b.BookingDate)  
 .ToList();

Search Logic Example (Events):

public async Task<IActionResult> Index(string searchString)  
{  
 var events = from e in \_context.Events  
 select e;  
  
 if (!string.IsNullOrEmpty(searchString))  
 {  
 events = events.Where(e => e.EventName.Contains(searchString) || e.Description.Contains(searchString));  
 }  
  
 return View(await events.ToListAsync());  
}

## B. Search feature

Implemented using LINQ and search form controls on the index views for Venues, Events and bookings.

## 4. Azure Deployment Updates

The application, enhanced with image upload and error-handling features, was redeployed to Azure App Service.

CI/CD Pipeline (Optional): Set up with GitHub Actions for automatic deployment on push.

## Conclusion

The enhancements introduced in Part 2 significantly improve the reliability, user experience, and cloud readiness of the EventEase application. Through Azure Blob integration, advanced validation, and user-centric search features, the system is more scalable, secure, and usable.

## Reference List:

1. Amazon Web Services (AWS), 2024. Cloud Computing Models. [online] Available at: <https://aws.amazon.com/types-of-cloud-computing/> [Accessed 7 Apr. 2025].
2. IBM, 2023. IaaS vs. PaaS vs. SaaS: What’s the difference? [online] Available at: <https://www.ibm.com/cloud/blog/iaas-vs-paas-vs-saas> [Accessed 7 Apr. 2025].
3. Microsoft Azure, 2024. Quickstart: Upload, download, and list blobs with .NET. [online] Available at: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-dotnet> [Accessed 7 Apr. 2025].
4. Microsoft Learn, 2024. Deploy an ASP.NET Core app to Azure App Service. [online] Available at: <https://learn.microsoft.com/en-us/azure/app-service/quickstart-dotnetcore> [Accessed 7 Apr. 2025].
5. Microsoft Docs, 2024. Use Application Insights with ASP.NET Core. [online] Available at: <https://learn.microsoft.com/en-us/azure/azure-monitor/app/asp-net-core> [Accessed 7 Apr. 2025].
6. Rountree, D. and Castrillo, I., 2014. The Basics of Cloud Computing: Understanding the Fundamentals of Cloud Computing in Theory and Practice. 1st ed. Waltham, MA: Syngress.