The journey of developing the Event Ease Application has been a thorough and deep dive into cloud-based solutions which include combining Azure services such as Azure web app, Azure blob storage, Azure SQL Database and Azure Storage services. This report will look back at the experience, the report will also document the application’s features, the application’s architectural components. The report will also outline how Azure services were leveraged, how the different technologies adopted, and lastly the different lessons learned throughout the development of the application.

The application has several features that make its operation very smooth and comprehensive. These features include:

Event Creation and management: The application has functionality for creating, updating, and deleting events with associated data such as the event’s name, event date and the event’s location.

Data upload and Storage: The application has the ability to upload event-related information such as the event’s date and images for the venue reference

View and add Bookings: The application has functionality to enable to the user to add and view their bookings. Additionally, venues linked to any bookings cannot be deleted and any events booked to any bookings cannot be deleted.

Search and filter Functionality: This feature enables the user to search for any events, venues and filter through these results by bookings, venue, date range, and availability. ( Whether the venue is available or not) .

CRUD Operations for Events and Venues: The applications users are able to create, read, update and delete certain events and venues, this also includes uploading and managing any images associated with those venues and events.

There were certain components and technologies used is the development of the EventEaseApplication, these components include the using ASP.NET CORE MVC(which is known for its Model-View-Controller architecture) for the front-end and back-end of the application. The MVC architecture allows for a clear separation between the application’s interface elements and the logic and the data( which is the models).Additionally, the MVC framework provides a framework that utilized for server-side operations. This MVC architecture allows for simple scalability while also ensuring a robust API. Furthermore, the technologies used in application include the languages used in the development of the application. These languages include C# Programming Language, the reason this language is used is because it is an ideal language used for enterprise level applications. HTML was also used in the development of the EventEaseApplication, this markup language allows for better user interface development for web applications, giving the user a visually appealing experience.

For the database layer the application uses Azure SQL Database. It provides very security, scalability and it also widely available. The reason it used is because it has simplified any scalability requirements to accommodate growing data requirements. Plus, it also has automated backups and built-in redundancy. But there are some alternatives to Azure SQL Database, these include, PostgreSQL on Azure, these can be used for any complicated queries or the MongoDB Atlas can be used for NoSQL data queries and structures

For storage in application, Azure Blob storage was used. It is useful for managing unstructured data such as documents and images. The reason it used in the EventEaseApplication is because that is very cost-effective when it comes to storing images and documents. Plus, it is also very scalable as it also has CDN integration for optimized delivery. But there are also some alternatives to Azure Blob storage, such as Amazon’s S3, is has similar functionalities as Azure’s blob storage but less coherent with Azure’s ecosystem

Azure Web application was used for the EventEaseApplication because it is able to host the application’s back-end and front-end in a scalable, secure environment. Azure’s web application has integrated CI/CD pipelines which allow for simple deployments. Other alternatives that could have been used to Azure’s Web Application are AWS Elastic Beanstalk which has similar functionalities to Azure’s Web Application but it doesn’t have seamless integration to other Azure services.

Reflection on the project

The project started with a thorough analysis of the application’s requirements and careful planning of architectural requirements. One of the challenges faced with this project was mastering the HTML side of things and integrating Azure services with ASP.NET CORE MVC.

The lessons learned while developing the application include:

* **Collaboration in development**: The importance of team communication tools, such as Github, was reinforced. The need of learning the functions of Github was very important.
* **Documentation**: Maintaining comprehensive documentation for each component proved invaluable in streamlining debugging and future development.
* **Cloud-Native requirements**: Understanding the ins and outs of cloud services and their configurations gave insights into optimizing costs in producing the application and improving performance.

https://github.com/MokoenaMolemo/CLOUDDEVPART3.git