[Nagarro]- [CTT Online Shopping]

DAR Document

Nagarro Software Pvt. Ltd.

Sandeep Singh

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| Version | Date | Author/Contributor | Comments |
| 1.0 | 15-Jan-25 | Sandeep Singh | Initial Phase |
| 1.1 | 25-Feb-25 | Sandeep | Added new tools |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1 Introduction 5](#_Toc193269811)

[1.1 Objective and scope of document 5](#_Toc193269812)

[2 Requirements at a Glance 6](#_Toc193269813)

[3 Available tools 7](#_Toc193269814)

[3.1 Elastic Search 7](#_Toc193269815)

[3.1.1 Features 7](#_Toc193269816)

[3.1.2 Pricing 7](#_Toc193269817)

[3.2 Azure Cognitive Search 7](#_Toc193269818)

[3.2.1 Features 7](#_Toc193269819)

[3.2.2 Pricing 8](#_Toc193269820)

[3.3 MS SQL 8](#_Toc193269821)

[3.3.1 Features 8](#_Toc193269822)

[3.3.2 Pricing 8](#_Toc193269823)

[3.4 MySQL 8](#_Toc193269824)

[3.4.1 Features 8](#_Toc193269825)

[3.4.2 Pricing 8](#_Toc193269826)

[3.5 MongoDB 9](#_Toc193269827)

[3.5.1 Features 9](#_Toc193269828)

[3.5.2 Pricing 9](#_Toc193269829)

[3.6 Azure Cosmos DB 9](#_Toc193269830)

[3.6.1 Features 9](#_Toc193269831)

[3.6.2 Pricing 9](#_Toc193269832)

[4 Comparison Analysis 9](#_Toc193269833)

[4.1 Weightage Matrix 10](#_Toc193269834)

[4.2 Elastic Search vs Azure AI Search 10](#_Toc193269835)

[4.3 MS SQL Server vs MySQL 10](#_Toc193269836)

[4.4 MongoDB vs. Cosmos DB 11](#_Toc193269837)

[5 Recommendation 11](#_Toc193269838)

[5.1 Elasticsearch vs. Azure AI Search 11](#_Toc193269839)

[5.2 MS SQL vs. MySQL 11](#_Toc193269840)

[5.3 Mongo Db vs. Cosmos Db 11](#_Toc193269841)

[6 Assumptions 12](#_Toc193269842)

[7 Risks 12](#_Toc193269843)

[8 Appendix 13](#_Toc193269844)

[8.1 References 13](#_Toc193269845)

# Introduction

This Decision Analysis and Resolution (DAR) document for the "CTT Online Shopping Website" project provides a comprehensive analysis of the technical options available to meet the project's core requirements. This document serves as a structured approach for evaluating and selecting appropriate tools, frameworks, and infrastructure components based on established criteria that align with the project’s goals.

## Objective and scope of document

The objective of this DAR document is to support informed decision-making by comparing potential solutions for key architectural components of the CTT Online Shopping Website. These decisions are pivotal for creating a secure, scalable, and high-performing online shopping platform. The document will cover the analysis of backend frameworks, frontend technologies, database options, payment integration tools, hosting infrastructure, and CI/CD tools, considering aspects such as cost, scalability, performance, security, and maintainability. This analysis will conclude with a set of recommendations that aim to meet the requirements and support long-term adaptability and growth of the application.

# Requirements at a Glance

The CTT Online Shopping Website requires a secure, scalable, and high-performance platform for online retail. Below is a summary of the key requirements that guide the selection and evaluation of tools, frameworks, and services in this document:

1. **Frontend Requirements**
   * **Responsive User Interface**: Intuitive and adaptive design for seamless user experiences across devices (e.g., desktop, mobile).
   * **Product Catalog Navigation**: Efficient browsing and filtering options within the product Catalog.
   * **User Interaction Features**: Includes product zoom, featured products, quick view, comparison, and Wishlist functionality.
2. **Backend Requirements**
   * **Authentication and Authorization**: Secure user login with third-party options (e.g., Facebook, Twitter).
   * **Product Management**: Support for CRUD operations on products, stock, and pricing managed by administrators.
   * **Order Processing**: A structured purchase flow from cart management to checkout and order confirmation.
   * **Inventory and Stock Tracking**: Real-time updates on product availability and stock levels.
3. **Database Requirements**
   * **Data Storage**: Storing customer data, product Catalog, inventory, order history, and transactional data.
   * **Scalability**: Ability to handle many users, especially during peak shopping periods.
   * **Data Consistency**: ACID compliance for critical financial and order data to ensure accurate order processing and inventory management.
4. **Payment Integration**
   * **Secure Payment Gateways**: Integration with third-party payment gateways (e.g., PayPal) for secure transactions.
   * **Multiple Payment Options**: Support for direct bank transfers, check, and credit card processing.
5. **Non-Functional Requirements**
   * **Security**: SSL for secure data transmission, data encryption, and role-based access for administrators.
   * **Scalability and Availability**: Support for load balancing and fault tolerance to ensure 24/7 availability.
   * **Maintainability**: Modular, component-based architecture to support ease of updates and feature additions.
   * **Portability**: The system should be platform-independent, accessible across different operating systems and device types.
6. **DevOps and CI/CD**
   * **Automated Deployment**: A CI/CD pipeline for code integration, testing, and deployment.
   * **Version Control Integration**: Source code versioning with automated triggers for builds and deployments.
   * **Monitoring and Alerts**: Real-time monitoring, logging, and alerting for service reliability and performance tracking.

# Available tools

This section provides an overview of the tools being compared, including their features and pricing.

## Elastic Search

Elasticsearch is a distributed, open-source search and analytics engine designed for horizontal scalability, reliability, and real-time search. It is commonly used for full-text search, log analytics, and real-time analytics.

### Features

* **Distributed and Scalable**: Handles large volumes of data across multiple nodes.
* **Real-Time Search**: Provides near real-time search capabilities.
* **Full-Text Search**: Supports advanced search queries and text analysis.
* **RESTful API**: Easy to integrate with other applications.
* **Customizable**: Highly flexible and customizable for specific use cases.
* **Community Support**: Strong open-source community and extensive documentation.

### Pricing

* **Self-Hosted**:Free (open-source), but requires infrastructure and maintenance costs.
* **Elastic Cloud:** Managed service starting at $95/month for basic plans**.**

## Azure Cognitive Search

Azure Cognitive Search is a fully managed search-as-a-service solution provided by Microsoft Azure. It offers AI-powered search capabilities, including natural language processing and integration with other Azure services.

### Features

* **Fully Managed:** No infrastructure or maintenance required.
* **AI-Powered Search:** Supports natural language processing, sentiment analysis, and key phrase extraction.
* **Integration with Azure:** Seamlessly integrates with Azure services like Cosmos DB, Blob Storage, and Azure Functions.
* **Security:** Built-in Azure security features, including role-based access control (RBAC) and encryption.
* **Scalability:** Automatically scales to handle high traffic and large datasets.

### Pricing

* **Free Tier:** Limited to 50 MB of storage and 3 indexes.
* **Standard Tier:** Starts at $73.73/month for 1 search unit (up to 25 GB of storage).

## MS SQL

Microsoft SQL Server (MS SQL) is a relational database management system (RDBMS) developed by Microsoft. It is widely used for enterprise applications and offers advanced features like high availability, security, and integration with Azure.

### Features

* **High Availability**: Supports Always on Availability Groups and failover clustering.
* **Security**: Advanced security features like Transparent Data Encryption (TDE) and row-level security.
* **Integration with Azure**: Seamlessly integrates with Azure services like Azure SQL Database and Azure Synapse Analytics.
* **Performance**: Optimized for high-performance workloads with in-memory OLTP and column store indexes.
* **Enterprise-Grade**: Suitable for large-scale, mission-critical applications.

### Pricing

* **Licensing**: Requires a paid license. Standard Edition starts at **$931/year** per core.
* **Azure SQL Database**: Managed service starting at **$4.99/month** for basic plans.

## MySQL

**Tool Description**:  
MySQL is an open-source relational database management system (RDBMS) known for its ease of use, reliability, and performance. It is widely used for web applications and small to medium-sized databases.

### Features

* **Open-Source**: Free to use with a large community and extensive documentation.
* **Ease of Use**: Simple to set up and manage.
* **Scalability**: Supports replication and clustering for high availability.
* **Performance**: Optimized for read-heavy workloads.
* **Community Support**: Strong open-source community and active development.

### Pricing

* **Self-Hosted**: Free (open-source), but requires infrastructure and maintenance costs.
* **Azure Database for MySQL**: Managed service starting at **0.017/hour∗∗(approx.∗∗0.017/*hour*∗∗(*approx*.∗∗12.50/month**) for basic plans.

## MongoDB

**Tool Description**:  
MongoDB is a popular NoSQL database known for its flexibility, scalability, and ease of use. It stores data in JSON-like documents and is widely used for modern applications.

### Features

* **Document-Oriented**: Stores data in flexible JSON-like documents.
* **Scalability**: Supports horizontal scaling through sharding.
* **High Performance**: Optimized for read/write operations.
* **Flexible Schema**: No fixed schema, allowing for easy changes.
* **Community Support**: Strong open-source community and extensive documentation.
* **Cloud Offering**: MongoDB Atlas provides a fully managed cloud service.

### Pricing

* **Self-Hosted**: Free (open-source), but requires infrastructure and maintenance costs.
* **MongoDB Atlas**: Managed service starting at **$57/month** for basic plans.

## Azure Cosmos DB

**Tool Description**:  
Azure Cosmos DB is a globally distributed, multi-model database service provided by Microsoft Azure. It offers low-latency, high-availability, and seamless integration with Azure services.

### Features

* **Globally Distributed**: Data is replicated across multiple regions for low latency and high availability.
* **Multi-Model Support**: Supports document, key-value, graph, and column-family data models.
* **Automatic Scaling**: Scales throughput and storage automatically based on demand.
* **SLAs**: Guarantees high availability, low latency, and consistency.
* **Integration with Azure**: Seamlessly integrates with Azure services like Functions, Logic Apps, and Power BI.
* **Security**: Built-in encryption, role-based access control (RBAC), and compliance certifications.

### Pricing

* **Pay-as-you-go**: Starts at **$24/month** for 400 RU/s and 5 GB storage.
* **Auto scale**: Automatically scales based on usage, with a minimum cost of **$48/month**.

# Comparison Analysis

This section compares the tools based on predefined features and assigns scores using a weightage matrix.

## Weightage Matrix

The weightage matrix defines the importance of each feature on a scale of 1 to 5, where 5 is the most important.

|  |  |
| --- | --- |
| Feature | Points |
| Performance | 5 |
| Scalability | 5 |
| Ease of Use | 4 |
| Cost | 4 |
| Integration with Azure | 5 |
| Community Support | 3 |
| Security | 5 |
| Learning Curve | 3 |
| Managed Service | 5 |
| Customization | 4 |

## Elastic Search vs Azure AI Search

|  |  |  |
| --- | --- | --- |
| Feature | Elasticsearch | Azure AI Search |
| Performance | 5 | 3.5 |
| Scalability | 5 | 4 |
| Ease of Use | 4 | 5 |
| Cost | 4 | 3 |
| Integration with Azure | 3 | 5 |
| Community Support | 5 | 3 |
| Security | 5 | 5 |
| Learning Curve | 3.5 | 4 |
| Managed Service | 3 | 5 |
| Customization | 5 | 3 |
| Total | 182.5 | 177.5 |

## MS SQL Server vs MySQL

|  |  |  |
| --- | --- | --- |
| Feature | MS SQL Server | MySQL |
| Performance | 5 | 4 |
| Scalability | 5 | 4 |
| Ease of Use | 5 | 4 |
| Cost | 3 | 5 |
| Integration with .NET | 5 | 3 |
| Community Support | 4 | 5 |
| Security | 5 | 4 |
| Learning Curve | 4 | 3.5 |
| Managed Service | 5 | 4 |
| Customization | 4 | 5 |
| Performance | 5 | 4 |
| Scalability | 5 | 4 |
| Total | 197 | 176.5 |

## MongoDB vs. Cosmos DB

|  |  |  |
| --- | --- | --- |
| Feature | Mongo Db | Cosmos Db |
| Performance | 5 | 5 |
| Scalability | 5 | 5 |
| Ease of Use | 5 | 4 |
| Cost | 4 | 3 |
| Integration with .NET | 3 | 5 |
| Community Support | 5 | 3 |
| Security | 4 | 5 |
| Learning Curve | 4 | 3 |
| Managed Service | 4 | 5 |
| Customization | 5 | 4 |
| Performance | 5 | 5 |
| Scalability | 5 | 5 |
| Total | 168 | 167 |

# Recommendation

Based on the weighted scores and analysis:

## Elasticsearch vs. Azure AI Search

**Elasticsearch** is the preferred choice for applications requiring high customization, scalability, and extensive community support. It provides greater flexibility in fine-tuning search relevance and managing large-scale data but requires more effort in setup and maintenance. **Azure Cognitive Search**(Azure AI Search), on the other hand, offers a fully managed service with seamless Azure integration, making it a suitable option for cloud-native applications. Given its higher performance and scalability, **Elasticsearch** stands out as the better option when customization and control are key priorities.

## MS SQL vs. MySQL

**MS SQL Server** is the optimal choice for enterprise-grade applications, offering better security, transaction handling, and deep integration with Azure services. It excels in complex queries, analytics, and large-scale data management. **MySQL**, being an open-source solution, provides cost-effectiveness and strong community support, making it suitable for smaller projects and applications with budget constraints. However, with superior support for enterprise workloads and .NET applications, **MS SQL Server** is the recommended option for robust and scalable database solutions.

## Mongo Db vs. Cosmos Db

**Cosmos DB** is better suited for scenarios requiring **seamless integration with Azure services, global distribution**, and **enterprise-grade security**. Its fully managed nature and automatic scaling capabilities make it ideal for applications that demand high availability, low latency, and compliance with industry standards.

**MongoDB**, on the other hand, is a strong choice for applications that prioritize **flexibility, open-source customization, and strong community support**. It is particularly well-suited for projects where cost-effectiveness and self-managed infrastructure are key considerations.

In environments heavily reliant on Azure, **Cosmos DB** provides a more integrated and streamlined experience, making it the preferred option for modern, cloud-native applications.

# Assumptions

This section outlines the assumptions made during the analysis.

1. **Integration with Azure**:
   * The analysis assumes that the tools will be used in an Azure environment, and integration with Azure services is a key requirement.
2. **Managed Service Preference**:
   * The analysis assumes that a fully managed service is preferred over self-hosted solutions for ease of use and reduced operational overhead.
3. **Performance and Scalability**:
   * The analysis assumes that high performance and scalability are critical requirements for the application.
4. **Cost Sensitivity**:
   * The analysis assumes that cost is an important factor, but not the sole deciding factor, in tool selection.
5. **Community and Documentation**:
   * The analysis assumes that strong community support and extensive documentation are important for troubleshooting and development.

# Risks

This section identifies potential risks associated with the chosen tools.

1. **Elasticsearch**
   * **Operational Overhead**: Self-hosted Elasticsearch requires significant operational effort for setup, maintenance, and scaling.
   * **Cost**: While the software is free, infrastructure and maintenance costs can be high.
   * **Learning Curve**: Elasticsearch has a steeper learning curve compared to managed solutions.
2. **Azure Cognitive Search**
   * **Vendor Lock-In:** Heavy reliance on Azure services may lead to vendor lock-in.
   * **Cost:** Managed services can become expensive as usage scales.
   * **Customization Limitations:** Limited customization compared to self-hosted solutions like Elasticsearch.
3. **MS SQL**
   * **Cost:** Licensing costs for MS SQL can be high, especially for enterprise-grade features.
   * **Complexity:** Advanced features like high availability and security may require additional configuration and expertise.
   * **Vendor Lock-In:** Heavy reliance on Microsoft technologies may lead to vendor lock-in.
4. **MySQL**
   * **Performance:** MySQL may not perform as well as MS SQL for complex, high-performance workloads.
   * **Security:** While MySQL is secure, it may lack some advanced security features available in MS SQL.
   * **Managed Service Costs:** While the software is free, managed services like Azure Database for MySQL can incur costs.
5. **MongoDB**
   * **Operational Overhead:** Self-hosted MongoDB requires significant operational effort for setup, maintenance, and scaling.
   * **Cost:** While the software is free, infrastructure and maintenance costs can be high.
   * **Learning Curve:** MongoDB has a steeper learning curve compared to managed solutions.
6. **Cosmos DB**
   * **Vendor Lock-In**: Heavy reliance on Azure services may lead to vendor lock-in.
   * **Cost**: Managed services can become expensive as usage scales.
   * **Customization Limitations**: Limited customization compared to self-hosted solutions like MongoDB.

# Appendix

## References

1. **Elasticsearch:**
   * Official Website: <https://www.elastic.co/>
   * Pricing: <https://www.elastic.co/pricing/>
   * Documentation: <https://www.elastic.co/guide/index.html>
2. **Azure Cognitive Search**:
   * Official Website: <https://azure.microsoft.com/en-us/services/search/>
   * Pricing: <https://azure.microsoft.com/en-us/pricing/details/search/>
   * Documentation: <https://docs.microsoft.com/en-us/azure/search/>
3. **MS SQL:**
   * Official Website: <https://www.microsoft.com/en-us/sql-server/>
   * Pricing: <https://www.microsoft.com/en-us/sql-server/sql-server-2019-pricing>
   * Azure SQL Database Pricing: <https://azure.microsoft.com/en-us/pricing/details/sql-database/>
   * Documentation: <https://docs.microsoft.com/en-us/sql/sql-server/>
4. **MySQL**:
   * Official Website: <https://www.mysql.com/>
   * Pricing (Azure Database for MySQL): <https://azure.microsoft.com/en-us/pricing/details/mysql/>
   * Documentation: <https://dev.mysql.com/doc/>
5. **Azure Pricing Calculator:**
   * <https://azure.microsoft.com/en-us/pricing/calculator/>
6. **Elastic Cloud Pricing:**
   * <https://www.elastic.co/cloud/pricing>
7. **Azure Documentation:**
   * <https://docs.microsoft.com/en-us/azure/>
8. **MongoDB**
   * Official Website: <https://www.mongodb.com/>
   * Pricing: <https://www.mongodb.com/pricing>
   * Documentation: <https://docs.mongodb.com/>
9. **Azure Cosmos DB**
   * Official Website: <https://azure.microsoft.com/en-us/services/cosmos-db/>
   * Pricing: <https://azure.microsoft.com/en-us/pricing/details/cosmos-db/>
   * Documentation: <https://docs.microsoft.com/en-us/azure/cosmos-db/>