REQUIREMENT GATHERING AND ANALYSIS PHASE
TECHNOLOGY STACK (ARCHITECTURE & STACK)

Date	6th July 2024
Team ID	SWTID1720001202
Project Name	Shop-EZ (E- Commerce Website)
Maximum Marks	

TECHNICAL ARCHITECTURE:

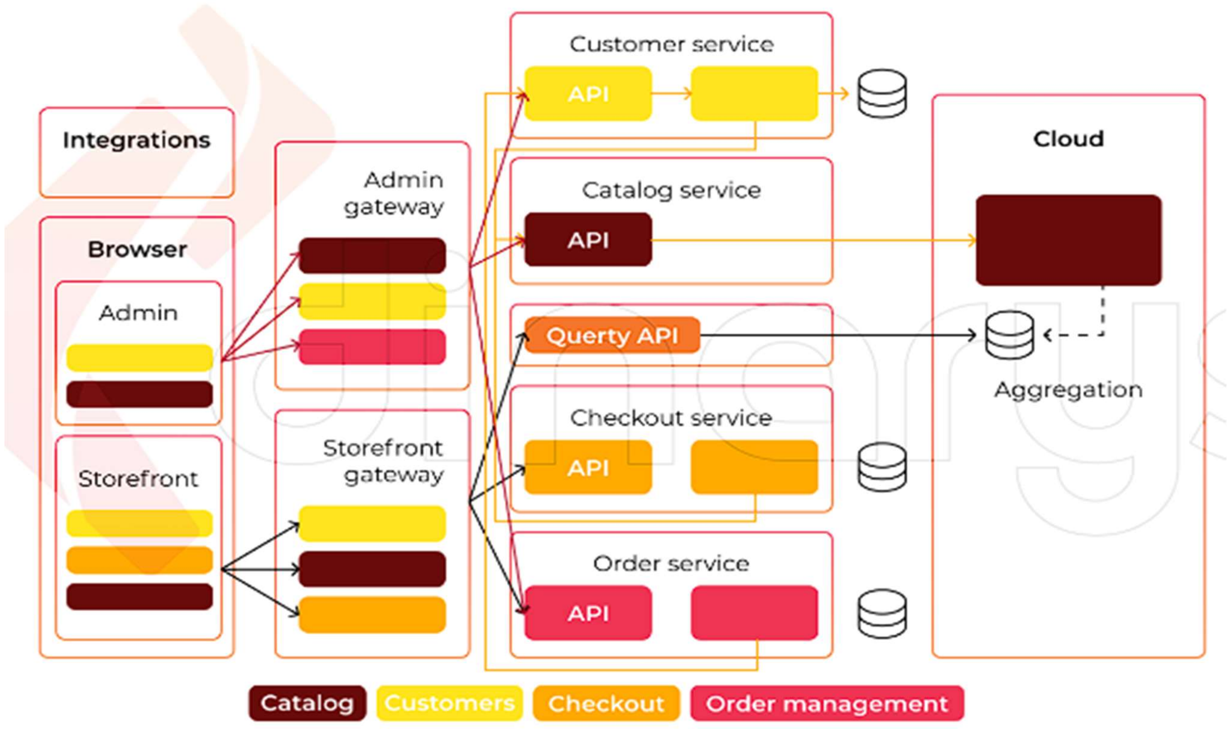

Table-1 : Components & Technologies:

S No	Component	Description	Technology
1	User Interface	Web UI	React.js
2	Application Logic-1	Display total number of items as notification on cart symbol	React.js, JavaScript
3	Application Logic-2	Total Cost of Products are displayed in Final amount of cart page	React.js, JavaScript
4	Application Logic-3	Total Quantity of items are shown in cart page with Net Quantity	React.js, JavaScript
5	Database	Numbers, Strings, Boolean, Float are datatypes used as schema	MongoDB
6	Cloud Database	Atlas is the cloud platform in which data is stored	MongoDB Atlas, Azure
7	File Storage	A folder cloned with GIT on local machine is required to run code	Git, GitHub
8	External API-1	API used to add product or remove product from DB linked with MongoDB syntax	Node.js, Express.js, MongoDB
9	External API-2	API used to show all products from DB linked with MongoDB syntax	Node.js, Express.js, MongoDB
10	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration:	Node.js, Express.js, Azure, GitHub

Table-2: Application Characteristics:

S No	Characteristics	Description	Technology
1	Open-Source Frameworks	List the open-source frameworks used	MongoDB, Express.js, React, Node.js
2	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Encryption, Hashing, Access Controls, Network Security, Azure Security Center
3	Scalable Architecture	3-Tier Architecture: Separates the presentation (React), logic (Node.js/Express), and data (MongoDB) layers. This separation allows each layer to be scaled independently. Microservices Architecture: Decomposes the application into smaller services, each handling a specific function, which can be developed, deployed, and scaled independently.	React for frontend presentation Node.js/Express for backend logic MongoDB for database storage Docker & Kubernetes for containerization and orchestration Azure Kubernetes Service (AKS) for managing Kubernetes clusters Azure Load Balancer and Application Gateway for load balancing
4	Availability	Load Balancers: Distribute incoming traffic across multiple servers so that no single server becomes a point of failure. Distributed Servers: Deploy application and data centers to provide redundancy and failover capabilities. Use replica sets to ensure high availability and disaster recovery.	Azure Load Balancer, Azure Application Gateway for load balancing Azure Kubernetes Service (AKS) for managing distributed server instances Azure Cosmos DB with MongoDB API for database redundancy and high availability

S No	Characteristics	Description	Technology
5	Performance	Request Handling: Optimize backend to handle a high number of requests per second using asynchronous, non-blocking I/O operations in Node.js.	Node.js for efficient request handling, Azure Functions for serverless computing