**Requirement Gathering and Analysis Phase**

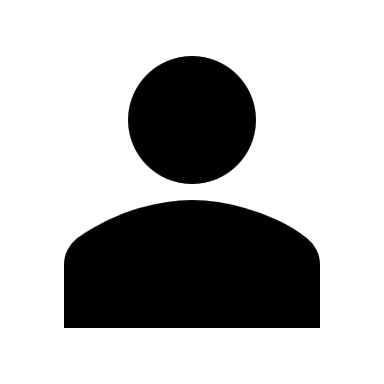
**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 6-07-2024 |
| Team ID | SWTID1719933594 |
| Project Name | ShopEZ: E-commerce Application |

**Technical Architecture:**

USER App Environment Database

User login



Customer Service

Admin

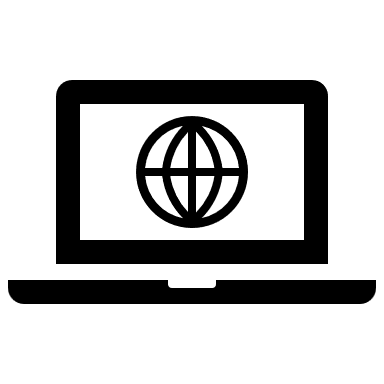
Customer

Order Service

Catalog Service

Application Gateway 

Customer



Web browser

**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | How user interacts with application | HTML, CSS, JavaScript / React.Js etc. |
|  | Frontend Framework | |  | | --- | | Framework for building interactive  UIs |  |  | | --- | |  | | JavaScript |
|  | Backend Server | |  | | --- | | Handles business logic and data  integration |  |  | | --- | |  | | |  | | --- | | Node.js, Express |  |  | | --- | |  | |
|  | Database | Data Type, Configurations etc. | MongoDB |
|  | File Storage | Storage for user-uploaded files and media | Local Filesystem |
|  | Payment Gateway | |  | | --- | | Integration for handling payments |  |  | | --- | |  | | Braintree |
|  | Order Processing | Logic for processing orders | Node.js,Express |
|  | Search & Filtering | Enables product search and filtering | Elasticsearch. |
|  | Machine Learning Model | Provides personalized product recommendations | Machine Learning Model |
|  | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud | Local Server System |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | The application leverages several open-source frameworks to ensure flexibility, ease of maintenance, and cost-efficiency. | Bootstrap,npm,reactj.s,MongoDB |
|  | Security Implementations | Secure user data and payment transactions | |  | | --- | |  |  |  | | --- | |  |   JWT-JSON Web Token |
|  | Scalable Architecture | Designed for horizontal scaling as traffic grows | Microservices architecture |
|  | Availability | Ensures minimal downtime during peak traffic | Load balancers, auto-scaling |
|  | Performance | Optimized for fast loading times and responsiveness | Caching strategies, Indexing and query optimization in MongoDB |