# Project Design Phase-II

## Technology Stack (Architecture & Stack)

Team Details:

|  |  |
| --- | --- |
| Team ID | LTVIP2025TMID20429 |
| Project Name | ShopEZ: One Stop Shop For Online Purchases |
| Project Name | 19 July 2025 |
| Date | 4 Marks |
| Maximum Marks |  |

Date: 19 July 2025

Maximum Marks: 4 Marks

## Technical Architecture:

ShopEZ is a full-stack e-commerce web application built using the MERN stack. It facilitates seamless product browsing, cart management, ordering, and seller dashboards. The application is divided into frontend, backend, and database components, hosted locally or on cloud infrastructure.  
  
Architecture Diagram Summary:  
- Frontend: User interacts via Web UI built with ReactJS.  
- Backend: Node.js + Express handles API requests.  
- Database: MongoDB stores user, product, order, and cart data.  
- Infrastructure: Runs on local server or can be deployed on cloud platforms.  
- External APIs: None currently, but architecture supports future integration.

## Table-1: Components & Technologies:

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Component | Description | Technology |
| 1 | User Interface | Web interface for user/admin to interact | ReactJS, HTML, CSS, JavaScript |
| 2 | Application Logic-1 | Handles routing, authentication, and request processing | Node.js, Express.js |
| 3 | Application Logic-2 | Handles shopping logic (cart, checkout, order processing) | Node.js, Express.js |
| 4 | Application Logic-3 | Admin functionalities (add products, view users/orders) | Node.js, Express.js |
| 5 | Database | Stores users, products, orders, carts | MongoDB, Mongoose |
| 6 | Cloud Database | Optional hosting on cloud-based MongoDB (MongoDB Atlas) | MongoDB Atlas |
| 7 | File Storage | Stores media/images locally or on cloud | Local File System or AWS S3 |
| 8 | External API-1 | (Future scope) Product recommendation / payment integration | Stripe API (optional) |
| 9 | External API-2 | (Future scope) Authentication or location API | Google Maps API / Firebase Auth |
| 10 | Machine Learning Model | (Future scope) Personalized recommendations | TensorFlow.js (optional) |
| 11 | Infrastructure | Local development or cloud deployment | Localhost / AWS / Render / Vercel |

## Table-2: Application Characteristics:

|  |  |  |
| --- | --- | --- |
| S.No | Characteristics | Technology |
| 1 | Open-Source Frameworks: MERN stack: MongoDB, Express.js, React.js, Node.js | Open-source JavaScript stack |
| 2 | Security Implementations: Password hashing, secure checkout, role-based access | bcrypt, JWT, HTTPS |
| 3 | Scalable Architecture: 3-tier architecture; supports microservices structure | React frontend + RESTful APIs |
| 4 | Availability: Supports cloud deployment; can use load balancing | Nginx, PM2, AWS Elastic Load Balancer |
| 5 | Performance: Uses async processing, MongoDB indexing, API caching (optional) | Node.js, MongoDB, Redis (optional) |

## References:

https://c4model.com/

https://www.mongodb.com/docs/

https://reactjs.org/

https://developer.mozilla.org/

https://aws.amazon.com/architecture/