Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	20 February 2026
Team ID	LTVIP2026TMIDS41763
Project Name	ShopEZ:One-Stop Shop for Online Purchases
Maximum Marks	4 Marks

Technical Architecture:

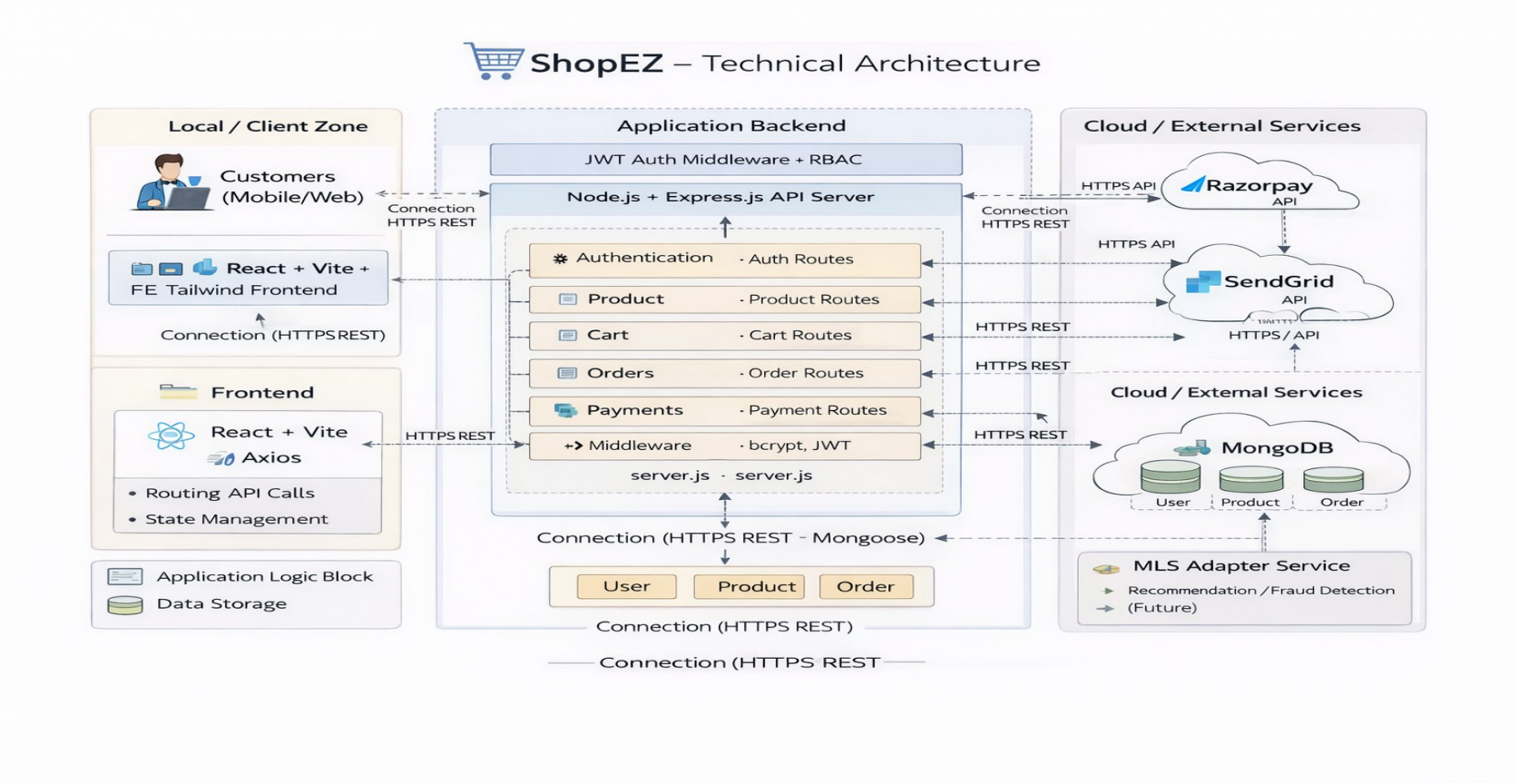

Table-1 : Components & Technologies:

S.No	Component	Description	Technology Used
1	User Interface	Web interface for customers and admin to browse products, manage cart, and orders	HTML, CSS, JavaScript, React.js, Vite, Tailwind CSS
2	Application Logic-1	Backend business logic for authentication, product, cart, and order management	Node.js, Express.js
3	Application Logic-2	Authentication & security handling	JWT (JSON Web Token), bcrypt
4	Application Logic-3	Payment processing logic	Razorpay API Integration
5	Database	Stores users, products, carts, orders, payments	MongoDB (NoSQL)
6	Cloud Database	Cloud-hosted database service	MongoDB Atlas
7	File Storage	Storage of product images	Local File System / Cloud Storage (if deployed)
8	External API-1	Online payment processing	Razorpay Payment Gateway API
9	External API-2	Email notifications (if implemented)	NodeMailer / SMTP (if used)
10	Machine Learning Model	(Future Scope – Recommendation System)	Not implemented / Future AI module
11	Infrastructure (Server / Cloud)	Application deployment environment	Local Server (Node.js), MongoDB Atlas Cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology Used
1	Open-Source Frameworks	The application is built using open-source frontend and backend frameworks for flexibility and cost efficiency.	React.js, Vite, Tailwind CSS, Node.js, Express.js, MongoDB
2	Security Implementations	Secure authentication and protected APIs using token-based authentication and password hashing.	JWT (JSON Web Token), bcrypt, HTTPS, Role-based access control (RBAC)
3	Scalable Architecture	Follows a 3-tier architecture (Frontend – Backend – Database) which allows scaling backend and database independently.	MERN Stack, MongoDB Atlas (Cloud DB)
4	Availability	Application is accessible via web and can be deployed on cloud platforms for high availability.	Node.js Server, MongoDB Atlas Cloud
5	Performance	Optimized API calls, lightweight frontend using Vite, efficient database queries using MongoDB indexing.	Axios (API calls), MongoDB Indexing, Vite build optimization

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d