**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 28 June 2025 |
| Team ID | LTVIP2025TMID54367 |
| Project Name | ShopEZ |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

**User Interface:**

* **Responsive Web UI**
* **Built using React components**
* **Allows product browsing, cart management, and checkout**

**Application Logic:**

* **Frontend: React JS for single-page application and user interactions**
* **Backend: Node.js with Express.js for APIs**
* **Product Search & Filtering Logic using JavaScript functions**
* **Cart & Checkout Workflow integrated with REST API**

**Database & Storage:**

* **Database: MongoDB for storing user data, product catalog, orders, and cart**
* **Authentication: Firebase Auth (optional) or custom JWT-based system**
* **Storage: Cloudinary or Firebase Storage for storing product images**

**Payment Integration:**

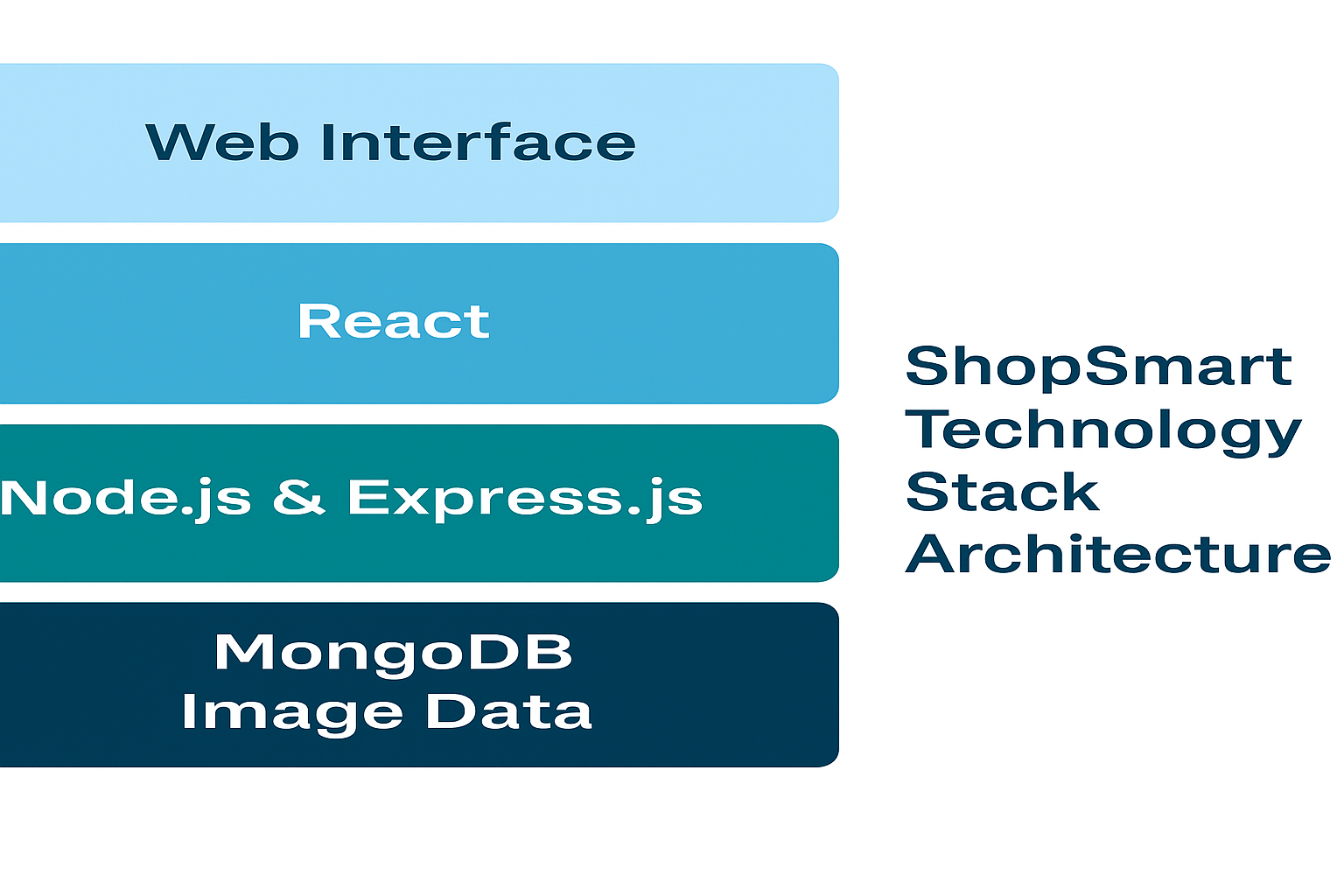
* **Razorpay / Stripe for secure online transactions**
* **Webhooks for payment confirmation**

**Infrastructure:**

* **Hosted on Render / Vercel / Heroku / Netlify**
* **CI/CD via GitHub**
* **Scalable using Docker (optional for future enhancement)**

**External Interfaces :**

* **Payment gateway API (e.g., Razorpay/Stripe)**
* **Firebase / Google Auth for optional login**
* **Email service (e.g., Nodemailer + Gmail SMTP) for order confirmations**



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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web UI for shopping | React JS, HTML, CSS, Bootstrap |
|  | Application Logic-1 | Product search, filter, cart logic | JavaScript, React Hooks |
|  | Application Logic-2 | API logic and backend | Node.js, Express.js |
|  | Database | Store users, products, orders | MongoDB |
|  | File Storage | Product image uploads | Firebase Storage / Cloudinary |
|  | Payment Gateway | Secure payment process | Razorpay / Stripe |
|  | Infrastructure | Deployment & scaling | Vercel / Heroku / Netlify / Docker |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Frameworks and libraries used | React JS, Node.js, Express, MongoDB |
|  | Security Implementations | Data safety and access control | JWT Auth, HTTPS, Password Hashing (bcrypt) |
|  | Scalable Architecture | Designed to support more users/products | RESTful API, modular components, Docker |
|  | Availability | Accessible across platforms and regions | Cloud hosting with 99.9% uptime |
|  | Performance | |  | | --- | |  |  |  | | --- | | Optimized browsing and purchase workflow | | Lazy loading, caching, optimized DB queries |