**Flipkart Clone - Project Documentation**

**1. Introduction**

The Flipkart Clone project is a full-stack e-commerce web application that replicates key functionalities of Flipkart. This system allows users to browse products, add them to a cart, manage orders, and proceed with payments.

**2. Objectives**

* Develop a scalable and responsive e-commerce platform.
* Implement user authentication and authorization.
* Enable product management with CRUD operations.
* Integrate shopping cart and checkout features.
* Provide search, filters, and pagination functionalities.

**3. Technology Stack**

**Frontend:** React.js with Vite, Material-UI for styling.  
**Backend:** Node.js, Express.js.  
**Database:** MySQL.  
**Other Tools:** Cloudinary (for image uploads), JWT (for authentication), Stripe/Razorpay (for payments).

**4. System Architecture**

The project follows a **client-server architecture** where:

* The frontend communicates with the backend via RESTful APIs.
* The backend interacts with the MySQL database for data storage.
* Authentication and authorization are managed via JWT.

**5. Features**

**User Module:**

* User Registration & Login (JWT authentication).
* Profile management (update name, email, and password).

**Product Module:**

* Admin can add, update, and delete products.
* Users can browse and view product details.
* Product search, category-wise filtering, and pagination.

**Cart Module:**

* Users can add/remove products to/from the cart.
* Quantity updates in real time.

**Order & Payment Module:**

* Users can place an order from the cart.
* Payment gateway integration.
* Order history and status tracking.

**Admin Panel:**

* Dashboard with order and user management.
* Product inventory management.

**6. Database Schema**

**User Table:**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| id | INT (PK) | Unique user ID |
| name | VARCHAR | User full name |
| email | VARCHAR | User email (unique) |
| password | VARCHAR | Hashed password |
| role | ENUM | 'user' or 'admin' |

**Product Table:**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| id | INT (PK) | Unique product ID |
| name | VARCHAR | Product name |
| price | DECIMAL | Product price |
| image | VARCHAR | Image URL |
| category | VARCHAR | Product category |
| stock | INT | Available stock |

**Order Table:**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| id | INT (PK) | Unique order ID |
| user\_id | INT (FK) | User placing order |
| total\_price | DECIMAL | Total order price |
| status | ENUM | 'Pending', 'Shipped', 'Delivered' |

**7. API Endpoints**

**Authentication:**

* POST /api/auth/register - User registration.
* POST /api/auth/login - User login.

**Products:**

* GET /api/products - Fetch all products.
* GET /api/products/:id - Fetch product details.
* POST /api/products - Add a new product (Admin only).
* PUT /api/products/:id - Update product details (Admin only).
* DELETE /api/products/:id - Delete a product (Admin only).

**Cart:**

* POST /api/cart - Add product to cart.
* GET /api/cart - Fetch user cart.
* DELETE /api/cart/:id - Remove item from cart.

**Orders:**

* POST /api/orders - Place an order.
* GET /api/orders - Fetch user orders.
* PUT /api/orders/:id - Update order status (Admin only).

**8. Setup Instructions**

1. **Clone the repository:**
2. git clone https://github.com/your-repo/flipkart-clone.git
3. cd flipkart-clone
4. **Backend Setup:**
5. cd backend
6. npm install
7. npm start
8. **Frontend Setup:**
9. cd frontend
10. npm install
11. npm run dev
12. **Database Setup:**
    * Create a MySQL database and import the schema.
    * Configure .env files with database credentials.

**9. Future Enhancements**

* Implement a recommendation system for users.
* Add live chat support for customer queries.
* Develop a mobile application for a better user experience.

**10. Conclusion**

This project successfully replicates the core functionalities of Flipkart, providing a robust and scalable e-commerce solution. Further enhancements can make the platform more user-friendly and competitive.