**Database Design and Development Report**

|  |  |
| --- | --- |
| Date | 10 April 2025 |
| Team ID | SWTID1743315070 |
| Project Name | ShopEZ – Seamless Online Shopping Platform |
| Maximum Marks |  |

**Project Title**: ShopEZ – Seamless Online Shopping Platform

**Date**: 15/04/25

**Prepared by**: Aditya Dubey, Anirudh Srivastava, Anuj Bhushan Tiwari, Utsav Jha

**Objective**

This report outlines the design and development of the database architecture for **ShopEZ**, ensuring robust data storage and efficient interaction with backend APIs to support online shopping features like product listings, user profiles, cart handling, and order tracking.

**Technologies Used**

* **Database Management System (DBMS):** MongoDB
* **Object-Document Mapper (ODM):** Mongoose

**Database Schema Design**

**The schema accommodates key entities for ShopEZ:**

**1. Users**

* username, email, usertype (buyer/seller/admin), password, address

**2. Products**

* sellerId, name, category, price, stock, description, imageUrl

**3. Orders**

* userId, products, totalPrice, status, shippingAddress, paymentMethod, createdAt

**4. Cart**

* userId, items[] (productId, quantity)

**Implement the Database using MongoDB**

The MongoDB database is implemented with the following collections and structures:

Database Name: shop-ez-db

{

username: { type: String, required: true },

email: { type: String, required: true, unique: true },

usertype: { type: String, required: true }, // buyer/seller/admin

password: { type: String, required: true },

address: { type: String }

}

{

sellerId: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },

name: { type: String, required: true },

category: { type: String },

price: { type: Number, required: true },

stock: { type: Number, default: 0 },

description: { type: String },

imageUrl: { type: String }

}

{

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },

products: [{ productId: String, quantity: Number }],

totalPrice: { type: Number },

status: { type: String, default: "Pending" },

shippingAddress: { type: String },

paymentMethod: { type: String },

createdAt: { type: Date, default: Date.now }

}

{

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },

items: [{ productId: String, quantity: Number }]

}

**Integration with Backend**

* MongoDB connected using Mongoose via URI string in .env.
* Database interactions handled through Mongoose models.
* Used for authentication, user management, product CRUD, cart handling, and order processing.

