# Full Stack Development with MERN

# Database Design and Development Report

| | |
|---|---|
| Date | 5 July 2024 |
| Team ID | SWTID1720091047 |
| Project Name | Project: E-commerce application |
| Maximum Marks | |

**Project Title**: -

**Date**: 5 July 2024
**Prepared by**: Magesh kumar S

**Objective**

The objective of this report is to outline the database design and implementation details for the ShopEZ project, including schema design and database management system (DBMS) integration.

**Technologies Used**

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

**Design the Database Schema**

The database schema is designed to accommodate the following entities and relationships:

**1. Users**

- Attributes: [list attributes like_id,username,password,email,usertype,approval,createdAt,updatedAt]

**2. Admin**

- Attributes: [list attributes like _id, Banner,categories]

**3.Product**

-Attributes:[ list attributes like_id, Title, description ,mainImg,carousel,sizes,category,gender,price ,discount]

**4.Orders**

**-** Attributes: [list attributes like _id,userId,name,email,mobile,address,pincode,title,description,mainImg,size,quantity,price,discount,paymentMethod,orderDate,orderStatus]

**5.Cart**

- Attributes: [list attributes like_id,userId,title,description,mainImg,size,quantity,price,discount]

**Implement the Database using MongoDB**

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

Database Name: Test

1. Collection: users

- Schema:

```
{
username: {type: String},
   password: {type: String},
   email: {type: String},
   usertype: {type: String}
}   ```
```

2. Collection: admin

- Schema:

```
{
banner: {type: String},
   categories: {type: Array}
```

3. Collection: Product

- Schema:

```
   ```

{
title: {type: String},
   description: {type: String},
   mainImg: {type: String},
   carousel: {type: Array},
   sizes: {type: Array},
   category: {type: String},
   gender: {type: String},
   price: {type: Number},
   discount: {type: Number}}
```
```

4. Collection: order

- Schema:

```

{

userId: {type: String},
name: {type: String},
email: {type: String},
mobile: {type: String},
address: {type: String},
pincode: {type: String},
title: {type: String},
description: {type: String},
mainImg: {type: String},
size: {type: String},
quantity: {type: Number},
price: {type: Number},
discount: {type: Number},
paymentMethod: {type: String},
orderDate: {type: String},
deliveryDate: {type: String},
orderStatus: {type: String, default: 'order placed'}
}

```

5. Collection: cart

- Schema:

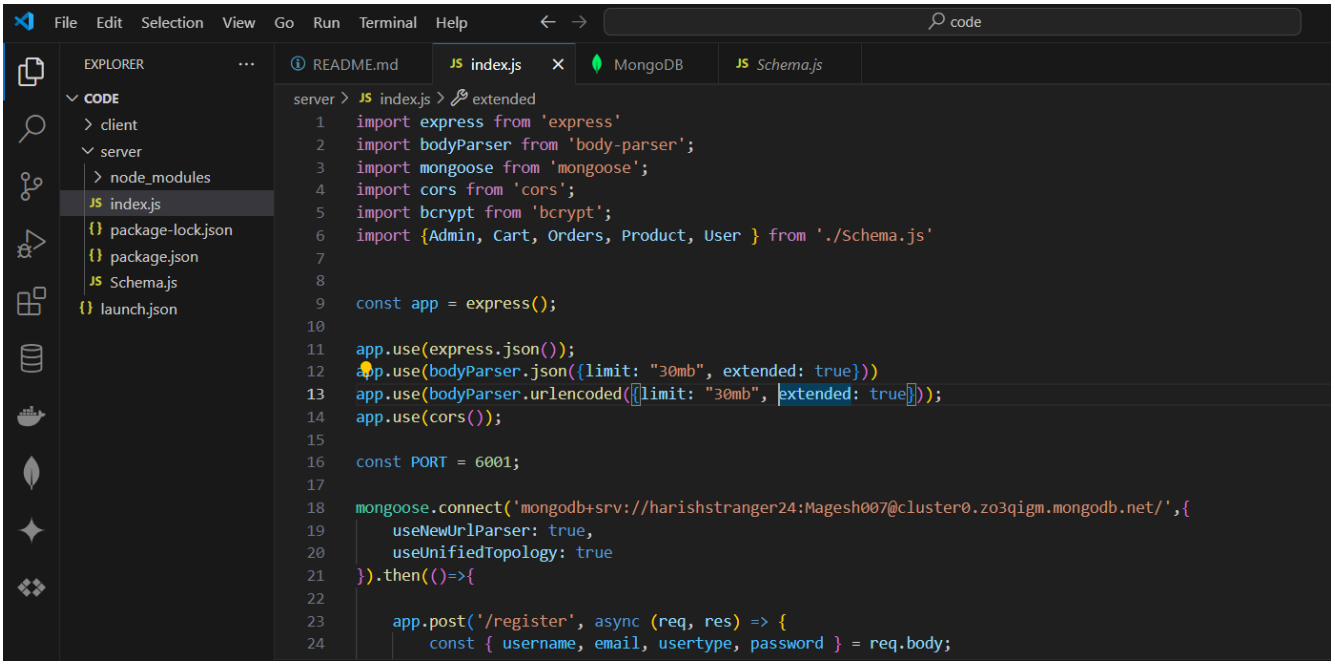
```

{
userId: {type: String},
title: {type: String},
description: {type: String},
mainImg: {type: String},
size: {type: String},
quantity: {type: String},
price: {type: Number},
discount: {type: Number}
}

```

**Integration with Backend**

- Database connection: Screenshot of Database connection done using Mongoose




- The backend APIs interact with MongoDB using Mongoose ODM Key interactions include:
  - User Management: CRUD operations for users.
  - Product Management: CRUD operations for Product.
  - Order Management: CRUD operations for orders.
  - Cart Management: CRUD operations for cart items.
  - Admin Management: Operations for managingcategories.