**Full Stack Development with MERN**

**Database Design and Development Report**

|  |  |
| --- | --- |
| Date | 18 July 2024 |
| Team ID | SWTID1720195916 |
| Project Name | Stock Trading Web App |
| Maximum Marks |  |

**Project Title**: Stock Trading Web App

**Date**: 18/07/24

**Prepared by**: Mukundan Sriram, Kirthick B Easwar, Anish, Adithya Hosur

**Objective**

The objective of this report is to outline the database design and implementation details for the Stock Trading Web App 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: username, email, usertype, balance, password

**2. Transactions**

- Attributes: user, type, paymentMode, amount, time

**3. Stocks**

- Attributes: user, symbol, name, price, count, totalPrice, stockExchange

**4. Orders**

- Attributes: user, symbol, name, price, count, totalPrice, stockType, orderType, orderStatus

**Implement the Database using MongoDB**

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

Database Name: Stock-App

userSchema: {

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

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

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

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

balance: {type: Number, default: 0}

}

transactionSchema: {

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

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

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

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

time: {type: String}

}

stocksSchema: {

user:{type: String},

symbol: {type: String},

name: {type: String},

price: {type: Number},

count: {type: Number},

totalPrice: {type: Number},

stockExchange: {type: String}

}

ordersSchema: {

user: {type: String},

symbol: {type: String},

name: {type: String},

price:{type: Number},

count: {type: Number},

totalPrice: {type: Number},

stockType: {type: String}, // intraday / delivery

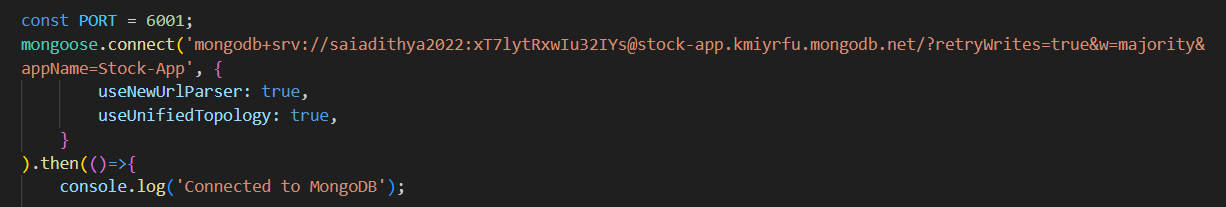
orderType: {type: String}, // buy / sell

orderStatus: {type: String}

}

**Integration with Backend**

* Database connection: Give 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.
  + Transaction Management: CRUD operations for transactions with user authentication.
  + Stock Management: CRUD operations for Stocks.
  + Order Management: CRUD operations for Orders.