Full Stack Development with MERN

Database Design and Development Report

Date	20-July-2024
Team ID	PNT2022TMID1720168902
Project Name	FOOD MINE – FOOD ORDERING
Maximum Marks	5Marks

Project Title: Rent Ease – House Hunt

Date: 16-July-2024

Prepared by: Raghu ram.ch & k.yoshitha

Objective

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

Technologies Used

- **Database Management System (DBMS):** MongoDB

Design the Database Schema

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

1. **Users Attributes:** name, email , password , cartData.
2. **Order Attributes:** userId ,items , amount , address , Status ,date , payment.
3. **Food Attributes:** name , description , price , image ,category.

Implement the Database using MongoDB

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

Database Name:FOODapp

1. Collection: users

- Schema:

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

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

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

cartData:{type:Object,default:{}}

2. Collection: Order

- Schema:

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

items: { type: Array, required:true},

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

address:{type:Object,required:true},

status: {type:String,default:"Food Processing"},

date: {type:Date,default:Date.now()},

payment:{type:Boolean,default:false}

3. Collection:food

- Schema:

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

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

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

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

category:{ type:String, required:true}

Integration with Backend

- Database connection: Screenshot of Database connection done

```
import mongoose from "mongoose";

export const  connectDB = async () =>{

    await mongoose.connect('mongodb://127.0.0.1:27017/FOODapp').then(()=>console.log("DB Connected"));

}
```

- Backend APIs interact with MongoDB using the Mongoose library, which provides a schema-based solution to model data.

✧ Connecting to MongoDB

The code provided sets up a connection to a MongoDB database using Mongoose. Here's a Step by-step breakdown:

✧ Importing Mongoose:

import mongoose from "mongoose";

This line imports the Mongoose library, which is used to connect to and interact with MongoDB.

✧ Connecting to the Database:

export const connectDB = async () => {
await mongoose.connect('mongodb://127.0.0.1:27017/FOODapp').then(()
=> console.log("DB Connected"));}

This function connectDB establishes a connection to a MongoDB instance running on the local machine (localhost) on the default port (27017) and connects to a database named FOODapp.

✧ mongoose.connect: This method is used to connect to the MongoDB server. It returns a promise, allowing the use of then to handle successful connections and catch to handle errors.

✧ await: Ensures that the connection process is completed before proceeding with any further operations.

✧ console.log("DB Connected"): Logs a message to the console once the connection is successfully established.