

Full Stack Development with MERN

Database Design and Development Report

Date	10 th July 2024
Team ID	SWTID1720126010
Project Name	SBFoods – Food Delivery
Maximum Marks	5 Marks

Project Title: SBFoods

Date: 10th July 2024

Prepared by: Yash Tekwani, Sarthak Ray, Kripa Verma, Pavan Gunupati

Objective

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

Technologies Used

- **Database Management System (DBMS):** MongoDB • **Object-Document Mapper (ODM):** Mongoose
- **Express JS :** Framework for Node.js web apps
- **JWT:** JSON web token for authentication and authorization
- **Bcrypt:** Hashing function for securely storing password

Design the Database Schema

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

1. Users

- Attributes: [_id, email, password, isAdmin, JWT Tokens, createdAt, updatedAt]

2. Food

- Attributes: [_id, image, title, price, createdAt, updatedAt]

3. Order

- Attributes: [_id, userId, OrderItems[title, image, price, _id], address]

Implement the Database using MongoDB

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

Database Name: [your_database_name]

1. Collection: users

- Schema:

```
    email: { type: String,
required: true,    unique: true,
validate(value){
if(!validator.isEmail(value)){    throw
new Error("Email is invalid");
    }
    }
},
password: {
type: String,
required: true,
},
tokens: [{
token: {
type: String,
    }
}],
isAdmin: {
type: Boolean,
required: true,
default: false,
    }
}
```

2. Collection: Food

- Schema:

```
    ...
```

```

    image:{
type: String,
required: true

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

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

    }  ``

```

3. Collection: Order

- Schema:

```

``

userId: {

    type: mongoose.Schema.Types.ObjectId,

    ref: 'User',

required: true

},

orderItems: [orderItemSchema],

address: {    type: String,

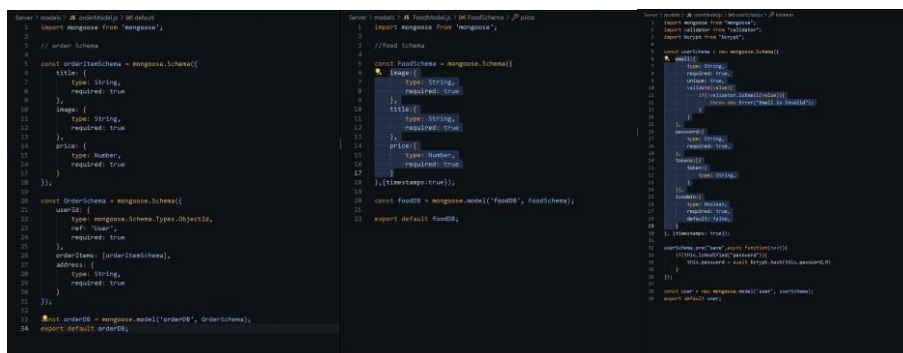
required: true

}  ``

```

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.
 - Food Management: CRUD operations for Managing Food List, with Admin authentication.
 - Order Management: CRUD operations for Managing Orders via Admin Authentication.