**1. Introduction**

* **Project Title and Type**
* Order management system
* Web-based project
* **Student's Name**
* Om Nakum (235370694027)
* Viraj Pala (235370694030)
* Hiten Chopda (235370694007)
* **Brief Introduction of the Project**

The Order Management System is web-based project developed using MERN Stack. It is mainly a restaurant based order taking system not like an food delivery app or site. It will computerize the order taking System from customers by our staff using digital platform, where the customer can see a digital menu by scanning the QR code and place the order to our staff. This helps the owner to know the customer preference well and can decide which of the food items are popular among customers and also improves the order tracking and invoice management.

* **Description of the problem that the project aims to solve**

The current system addresses to the problem in traditional order taking system, in which the order taking is a slow process which is replaced by a digital platform, this system reduces errors, speeds up food ordering service for the customer. The project will solve the problems like Order management, Food menu management, staff management using computerized way.

**2. Scope of the Project**

* **Project Scope (What it will cover):**
  + **Order Management:** It will offer a platform for staff to record and manage customer orders, ensuring accurate and efficient processing.
  + **Food Menu Display:** The system will provide a digital menu for guests to view available food items category wise, popular food items, today’s special, etc.
* **Out of Scope (What it will not cover):**
  + **Direct Order:** Guests will not be able to place orders directly through the system. Orders will be taken by staff only.

**3. Modules Description**

* **Detailed description of each module, sub modules and reports of the system.**
  + **User Module:** It will cover Authentication, user\_roles and profile.
  + **Menu Module:** Displays all available food items through landing page.
  + **Order Module:** It will cover order entry by staff wise and table wise and invoice management.
  + **Report Module:** It will cover Reports like track orders, most sell food items, staff in/out, etc.

**4. Database Design**

* **Food schema:**

{

name: { type: String, required: [true, "Name is required"] },

price: { type: Number, default: 0, required: [true, "price is required"] },

description: { type: String, required: [true, "description is required"] },

image: { type: String, required: [true, "image is required"] },

category\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'category', required: [true, "category id is required"] },

},

{

timestamps: true,

}

* **Category schema:**

{

name: { type: String, unique: true, required: [true, "Name is required"] },

status: {

type: String,

default: "active",

required: [true, "Status is required"],

},

{

timestamps: true,

}

* **Table Schema:**

{

table\_no: { type: Number, unique: true, required: [true, "Table\_no is require"] },

status: { type: String, default:"unoccupied", required: [true, "Status is require"] },

size: { type: Number,default:4, required: [true, "Size is require"] },

type: { type: String, required: [true, "Type is require"] },

},

{

timestamps: true,

}

* **User Schema:**

{

name: { type: String, required: [true, "Name is required"] },

email: { type: String, required: [true, "Email is required"] },

password: { type: String, required: [true, "password is required"] },

role\_as: { type: String, required: [true, "role\_as is required"] },

},

{

timestamps: true,

}

* **User Profile Schema:**

{

address: { type: String, required: [true, "Address is required"] },

mobile: { type: String, required: [true, "Mobile is required"] },

image: { type: String, required: [true, "Image is required"] },

gender: { type: String, required: [true, "Gender is required"] },

user\_id: { type: String, required: [true, "profile\_id is required"] },

},

{

timestamps: true,

}

* **Order Schema**

{

order\_no: {type: String, unique: true, required: [true, "order no is required"],},

customer\_mob: {type: String, required: [true, "Customer number is required"],},

status: {type: String, default: "open", required: [true, "status is required"],},

grand\_total: {type: Number, default: 0, required: [true, "total is required"],},

table\_id: {type: mongoose.Schema.Types.ObjectId, ref: "table",required: [true, "table id is required"],},

user\_id: {type: mongoose.Schema.Types.ObjectId, ref: "users",required: [true, "user id is required"],},

},

{

timestamps: true,

}

* **Order Items Schema**

{

quantity: {type: Number, default:1, required: [true, "quantity is required"],},

food\_id: {type: mongoose.Schema.Types.ObjectId, ref: "food", required: [true, "food id is required"],

},

order\_id: {type: mongoose.Schema.Types.ObjectId, ref: "orders", required: [true, "order id is required"],

},

},

{

timestamps: true,

}

**5. Implementation Plan**

**6. Project Plan**

**7. Limitations and Future Scope**

* **Limitations :**

1. The main limitation in project is that user can’t place direct order from digital menu.
2. The second is no live order status display to user.

* **Future Enhancements :**

1. We will add more feature like kitchen module.
2. We will remove errors and bugs to make the site efficient.

**8. References**

MongoDb : <https://www.mongodb.com/docs/>

Express Js : <https://expressjs.com/en/5x/api.html>

React Js: <https://react.dev/learn>

Node Js: <https://nodejs.org/docs/latest/api/>

Twilio : <https://www.twilio.com/docs>

Bootstrap : <https://getbootstrap.com/docs/5.3/getting-started/introduction/>