

Project Overview: Restaurant Menu Manager

Description: The Restaurant Menu Manager will be a web application where restaurant owners or staff can manage the menu items of their restaurant. Users can add new dishes, view the current menu, update dish details, and remove items no longer served.

Features

1. **Create:** Users can add new menu items to the restaurant's menu.
2. **Read:** Users can view all menu items, search for specific dishes, and see detailed information about each item.
3. **Update:** Users can edit the details of existing menu items.
4. **Delete:** Users can remove items from the menu.

Technology Stack

- **Frontend:** HTML, CSS, JavaScript (with a framework like React, Vue.js, or Angular)
- **Backend:** Node.js with Express
- **Database:** MongoDB (to store menu item information)

Database Schema

You can use a simple schema for the MongoDB collection like this:

json

Copy code

```
{
  "name": "Dish Name",
  "description": "Brief description of the dish",
  "price": 12.99,
  "category": "Appetizer / Main Course / Dessert",
  "imageUrl": "URL to dish image",
  "availability": true
}
```

Endpoints

1. **Create a Menu Item**
 - **POST** /api/menu-items
 - Body: { "name": "Dish Name", "description": "Delicious...", "price": 12.99, ... }

2. **Get All Menu Items**
 - `GET /api/menu-items`
 - Response: List of all menu items
3. **Get a Single Menu Item**
 - `GET /api/menu-items/:id`
 - Response: Details of the menu item with the specified ID
4. **Update a Menu Item**
 - `PUT /api/menu-items/:id`
 - Body: `{ "name": "Updated Dish Name", "description": "Updated description", "price": 14.99, ... }`
5. **Delete a Menu Item**
 - `DELETE /api/menu-items/:id`

Frontend Components

1. **MenuList:** Displays all menu items with options to edit or delete each item.
2. **MenuItemForm:** A form for creating or updating a menu item, including fields for all necessary attributes.
3. **MenuItemDetail:** A detailed view of a specific menu item when clicked.

Example Workflows

1. **Adding a Menu Item:**
 - User navigates to the MenuItemForm.
 - Fills out the form with dish details and submits it.
 - The dish is added to the database, and the MenuList updates to reflect the new item.
2. **Viewing Menu Items:**
 - User accesses the MenuList to see all dishes.
 - Each dish shows basic info (name, price) and has buttons for editing or deleting.
3. **Editing a Menu Item:**
 - User clicks the edit button on a dish, which pre-fills the MenuItemForm with that dish's details.
 - User updates the details and submits.
 - The MenuList reflects the updated information.
4. **Deleting a Menu Item:**
 - User clicks the delete button next to a dish.
 - A confirmation prompt appears.
 - If confirmed, the dish is removed from the database, and the list updates.

Additional Features (Optional)

- **User Authentication:** Allow restaurant owners or staff to sign up and log in to manage the menu.
- **Search Functionality:** Implement a search bar to find specific dishes by name or category.
- **Category Filtering:** Provide options to filter dishes by category (e.g., appetizers, main courses, desserts).
- **Image Upload:** Allow users to upload images for menu items.