Restaurant Management System

Project Overview

This project is a Django-based restaurant management system designed to handle various aspects of restaurant operations, including managing tables, reservations, users, customers, staff, menu items, orders, payments, inventory, and suppliers.

Features

- **Table Management**: Manage tables with unique numbers and capacities.
- **Reservation System**: Create and manage reservations with automatic expiration handling.
- **User Management**: Handle different user roles including customers, staff, and kitchen staff.
- **Menu and Order Management**: Create and manage menu items, and handle customer orders.
- **Payment Processing**: Process and store payment information securely.
- **Inventory Management**: Track and manage inventory items and suppliers.

Technologies Used

- **Django**: Web framework for building the application.
- **SQLite**: Default database for development.
- **Bootstrap**: Front-end framework for responsive design.

Installation

To get a local copy up and running, follow these steps:

1. **Download the ZIP file** of the project and extract it.
2. **Install Django**
3. **Run the development server**:

   ```
   python manage.py runserver
   ```

4. **Open your browser and visit**:
   `http://127.0.0.1:8000/`

Usage

Adding Data to the Database

To add data to the database, you can use the Django management command `python manage.py shell` to access the Django shell and execute the following Python scripts:

Adding a Table

```
from reservations.models import Table

table = Table(number=1, capacity=4)
table.save()
print(f"Table {table.number} (Capacity: {table.capacity}) added to the
database.")
```

Adding a Menu Item

```
from menu.models import MenuItem

menu_item = MenuItem(name="Pasta", description="Delicious pasta with tomato
sauce", price=12.99, available=True)
menu_item.save()
print(f"Menu item {menu_item.name} added to the database.")
```

Running the Application

After setting up the database and adding initial data, you can run the application using Django's development server:

```
python manage.py runserver
```

Visit `http://127.0.0.1:8000/` in your browser to access the application.