Dosa Restaurant Order Management API

This project is a RESTful API for managing customer orders, items, and customer information for a dosa restaurant. Built with **FastAPI** and **SQLite**, it provides a simple and efficient way to perform CRUD operations on customers, items, and orders.

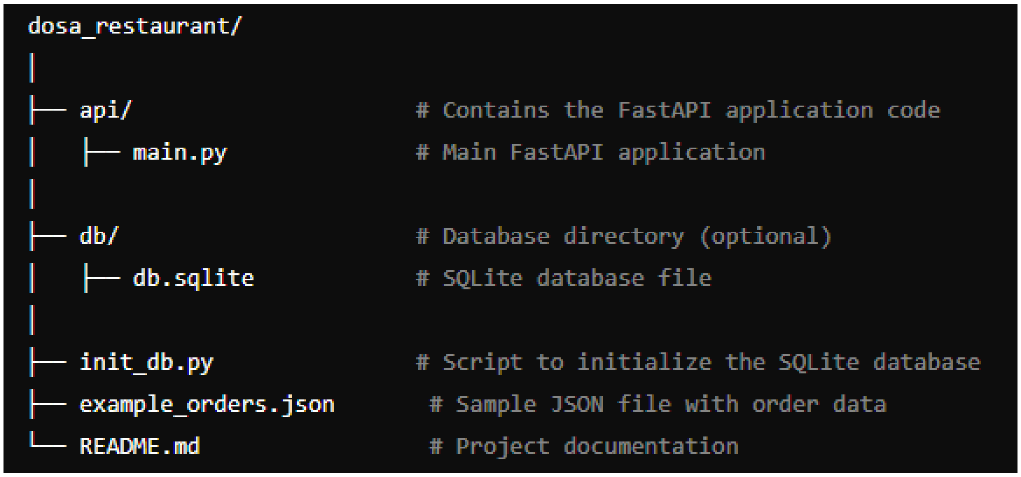
Table of Contents

- Features
- Technologies
- Installation
- Usage
- API Endpoints
- Database Structure
- Running the Application
- License

Features

- Create, retrieve, update, and delete customers, items, and orders.
- Uses relational constraints (primary and foreign keys) to maintain data integrity.
- Simple JSON-based input and output for easy integration with front-end applications.

Project Structure



```
dosa_restaurant/
│
├── api/                     # Contains the FastAPI application code
│   ├── main.py              # Main FastAPI application
│
├── db/                      # Database directory (optional)
│   ├── db.sqlite            # SQLite database file
│
├── init_db.py               # Script to initialize the SQLite database
├── example_orders.json       # Sample JSON file with order data
└── README.md                 # Project documentation
```

Technologies

- **FastAPI**: A modern web framework for building APIs with Python.
- **SQLite**: A lightweight database for storing data.
- **SQLAlchemy**: ORM for interacting with the database.
- **Pydantic**: Data validation and serialization.

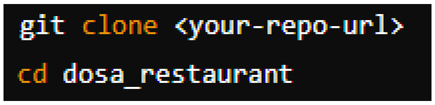
Setup and Installation

Requirements

- Python 3.7+
- FastAPI
- SQLite
- Uvicorn

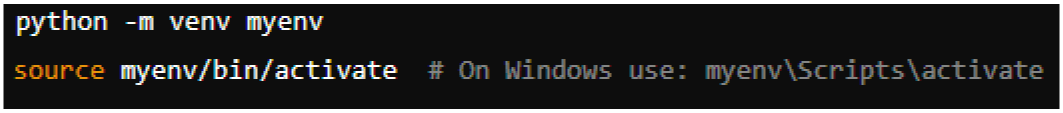
Installation Steps

1. **Clone the repository:**



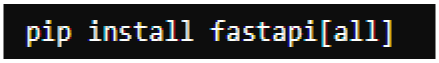
```
git clone <your-repo-url>
cd dosa_restaurant
```

2. **Create a virtual environment:**



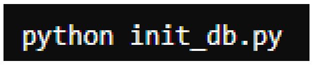
```
python -m venv myenv
source myenv/bin/activate  # On Windows use: myenv\Scripts\activate
```

3. **Install the required packages:**



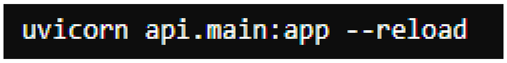
```
pip install fastapi[all]
```

4. **Initialize the database:**



```
python init_db.py
```

5. **Run the FastAPI application:**



```
uvicorn api.main:app --reload
```

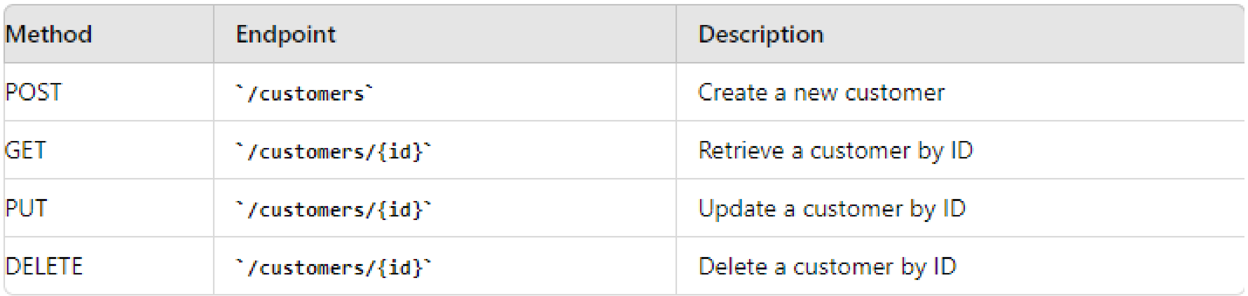
The API will be accessible at **http://127.0.0.1:8000**.

Usage

You can interact with the API using tools like Postman or directly through the interactive API documentation at `http://127.0.0.1:8000/docs`.

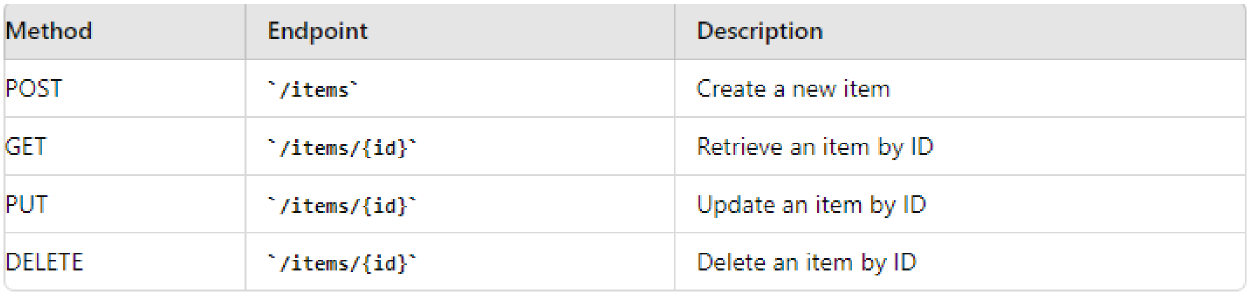
API Endpoints

Customers



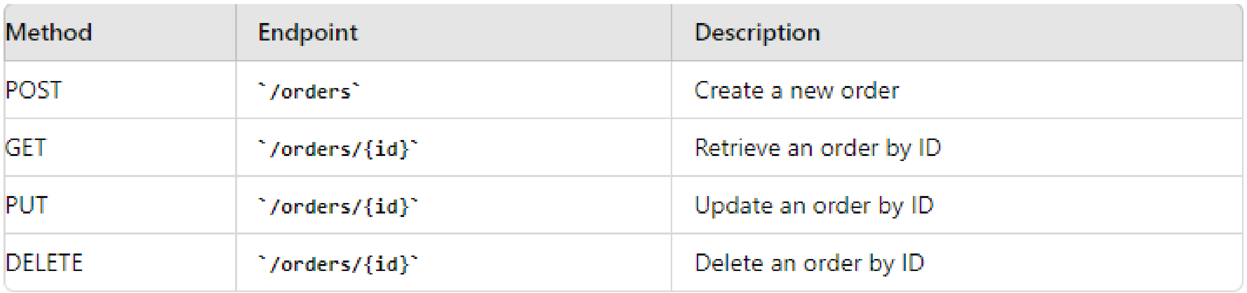
Method	Endpoint	Description
POST	`/customers`	Create a new customer
GET	`/customers/{id}`	Retrieve a customer by ID
PUT	`/customers/{id}`	Update a customer by ID
DELETE	`/customers/{id}`	Delete a customer by ID

Items



Method	Endpoint	Description
POST	`/items`	Create a new item
GET	`/items/{id}`	Retrieve an item by ID
PUT	`/items/{id}`	Update an item by ID
DELETE	`/items/{id}`	Delete an item by ID

Orders

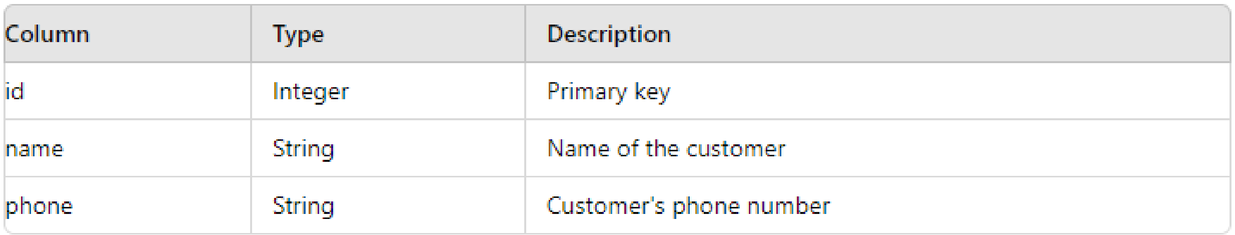


Method	Endpoint	Description
POST	`/orders`	Create a new order
GET	`/orders/{id}`	Retrieve an order by ID
PUT	`/orders/{id}`	Update an order by ID
DELETE	`/orders/{id}`	Delete an order by ID

Database Structure

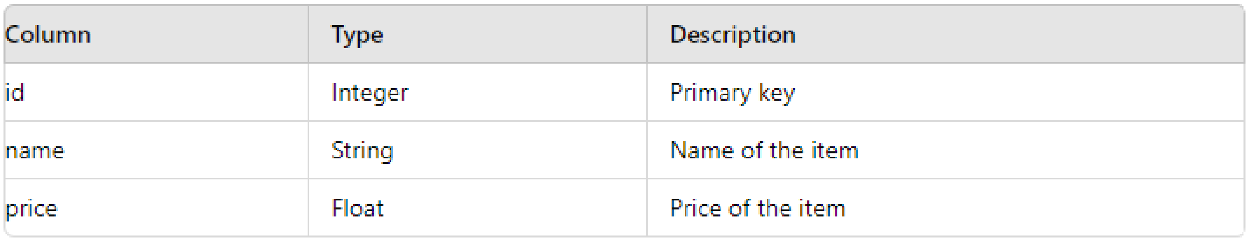
The application uses SQLite with the following tables:

Customers Table



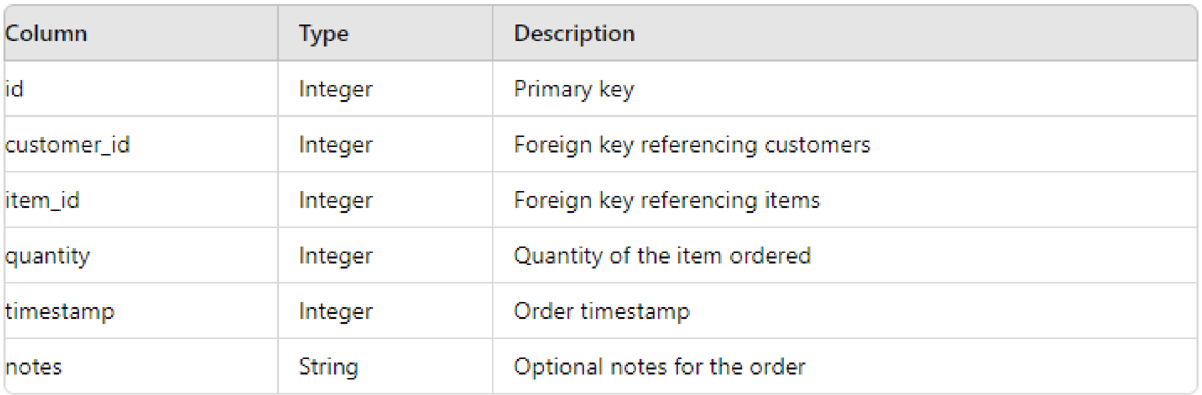
Column	Type	Description
id	Integer	Primary key
name	String	Name of the customer
phone	String	Customer's phone number

Items Table



Column	Type	Description
id	Integer	Primary key
name	String	Name of the item
price	Float	Price of the item

Orders Table



Column	Type	Description
id	Integer	Primary key
customer_id	Integer	Foreign key referencing customers
item_id	Integer	Foreign key referencing items
quantity	Integer	Quantity of the item ordered
timestamp	Integer	Order timestamp
notes	String	Optional notes for the order

License

This project is licensed under the MIT License.

Acknowledgments

- Thanks to FastAPI for providing an easy and efficient framework for building APIs.
- Special thanks to SQLite for lightweight database management.