

Order Tracing Bot

DOCUMENTATION

Nahom Teguade

KEDAME GEBEYA | ADDIS ABABA, ETHIOPIA

Table of Contents

1. Introduction
2. Features
3. Installation
4. Usage
5. Commands
6. Database Structure
7. Error Handling
8. Security Considerations
9. Future Enhancements
10. Conclusion

Introduction

The Order Bot is a Telegram bot designed to streamline the process of registering orders for shopkeepers and managing their information. It provides a user-friendly interface for interacting with the bot and utilizes a database to store and retrieve order and shopkeeper data. With the Order Bot, shopkeepers can easily record new orders and maintain accurate records of their shopkeeper information.

Telegram Bot for Kedame Gebeya

Name: Kedame Gebeya Order Tracing

Username: kg_orderTracing_bot

Token: 6048352789:AAG5hIYOZpiSEf3YwUMUFFX3ILwQ6azNd88

Features

The Order Bot offers the following features:

- i. **Order Registration:** Shopkeepers can use the `/addOrder`` command to initiate the process of registering a new order. The bot will guide them through a series of prompts to gather information about the order, including the product name, type, price, weight, quantity, and vehicle license.
- ii. **Shopkeeper Information:** The `/addInfo`` command enables shopkeepers to register their information. They will be prompted to provide their full name, branch of the shop, and phone number. This information is stored securely for future reference.
- iii. **Help and Start Commands:** The `/help`` command provides users with information about the available commands and their usage. The `/start`` command displays a welcome message and presents users with a custom keyboard containing buttons for each command, facilitating easy navigation.
- iv. **Intuitive Conversation Handling:** The bot employs a conversation command to handle messages from users that do not match any predefined commands. It responds with a friendly message, indicating that it didn't understand and requesting the user to rephrase their message.
- v. **Persistent Storage:** The bot utilizes an SQLite database to store registered orders and shopkeeper information. This ensures data persistence, allowing shopkeepers to retrieve and manage their records effectively.

Installation

To use the Order Bot, follow these installation steps:

1. Clone the repository or download the source code to your local machine.
2. Install the required dependencies by running the following command:

```
...
```

```
pip install pyTelegramBotAPI
```

```
...
```

3. Set up your Telegram bot by following the instructions provided by the BotFather. Obtain the bot token required for authentication.
4. Update the ``bot_token`` variable in the code with your Telegram bot token.
5. Ensure that you have SQLite installed or install it using appropriate package managers.
6. Run the Python script to start the bot:

```
...
```

```
python order_bot.py
```

```
...
```

7. The Order Bot is now active and ready to receive commands and messages.

Usage

Once the bot is running, users can interact with it through the Telegram messaging platform. Start a conversation with the bot and utilize the following commands:

- ``/start``: Initiates the conversation with the bot and displays a welcome message. It also provides a custom keyboard with buttons for each available command, simplifying interaction.
- ``/help``: Provides detailed information about the available commands and their usage.
- ``/addOrder``: Starts the process of registering a new order. The bot will prompt the user to enter specific details about the order, one step at a time.
- ``/addInfo``: Facilitates the registration of shopkeeper information. The bot guides the user through a series of prompts to gather necessary details such as the full name, branch of the shop, and phone number.

Additionally, users can send messages to the bot using natural language. The bot will respond with a friendly message, acknowledging that it didn't understand and requesting the user to rephrase their message.

Commands

The Order Bot supports the following commands:

- `/start``: Displays a welcome message and presents a custom keyboard with buttons for each command.
- `/help``: Provides information about the available commands and their usage.
- `/addOrder``: Initiates the process of registering a new order.
- `/addInfo``: Facilitates the registration of shopkeeper information.

Database Structure

The Order Bot utilizes a SQLite database to store registered orders and shopkeeper information. The database structure consists of two tables:

1. ``orders``: Stores information about the registered orders. The table has the following columns:

- ``order_id`` (Primary Key): Unique identifier for the order (auto-incremented).
- ``name``: Name of the product.
- ``type``: Type of the product.
- ``price``: Price of the product.
- ``weight``: Weight of the product.
- ``quantity``: Quantity of the product being ordered.
- ``vehicle_license``: Vehicle license associated with the order.

2. ``shopkeepers``: Stores information about the registered shopkeepers. The table has the following columns:

- ``shopkeeper_id`` (Primary Key): Unique identifier for the shopkeeper (auto-incremented).
- ``full_name``: Full name of the shopkeeper.
- ``branch``: Branch of the shop.
- ``phone_number``: Phone number of the shopkeeper.

Error Handling

The Order Bot incorporates error handling mechanisms to ensure a smooth user experience. It includes the following error scenarios and corresponding error messages:

- i. Invalid Command: If a user enters an invalid command, the bot responds with an appropriate error message, suggesting the use of the ``/help`` command to learn about available commands.

- ii. **Missing or Invalid Input:** During the order registration process, if a user fails to provide valid input for any required field, the bot prompts them to retry with valid information.
- iii. **Database Errors:** In case of any database-related errors, such as connection issues or SQL query failures, the bot notifies the user with a friendly error message, suggesting them to try again later.

Security Considerations

The Order Bot takes the following security considerations into account:

- i. **Authentication:** The bot utilizes the Telegram Bot API token for authentication, ensuring that only authorized users can interact with the bot.
- ii. **Secure Data Storage:** The registered orders and shopkeeper information are stored securely in a SQLite database. The database file is kept in a safe location, and appropriate access controls are implemented to protect the data.
- iii. **Input Validation:** The bot performs input validation to ensure that the entered information adheres to the expected format and prevents any potential security vulnerabilities, such as SQL injection attacks.
- iv. **Privacy:** The bot does not share or disclose any user information or order details to third parties. All data is kept confidential and used solely for the purpose of order registration and management.

Future Enhancements

The Order Bot has the potential for further enhancements and improvements. Some possible future enhancements include:

- **Order Management:** Implement additional functionality to allow shopkeepers to view, update, and delete registered orders.
- **Analytics and Reporting:** Integrate reporting capabilities to generate insights and analytics based on order data, such as sales trends, popular products, and revenue analysis.
- **User Authentication:** Enhance security by introducing user authentication mechanisms, ensuring that only authorized users can access and manage their order and shopkeeper information.
- **Localization:** Provide multi-language support to cater to a wider range of users.
- **Integration with Payment Gateways:** Enable seamless integration with payment gateways to facilitate online payment for orders.
- **Natural Language Processing (NLP):** Enhance the Chabot functionality by integrating NLP capabilities to improve the understanding of user messages and generate more accurate and contextually relevant responses.

Conclusion

The Order Bot is a powerful and user-friendly tool designed to simplify the order registration process for shopkeepers. With its intuitive interface, secure data storage, and conversational capabilities, the bot streamlines the interaction between shopkeepers and the registration system. By automating the process and maintaining accurate records, the Order Bot empowers shopkeepers to manage their orders more efficiently, leading to improved productivity and customer satisfaction.