

# FASTRADE - An Online Trading Portal

FASTRADE is an online trading portal created for the end users and the employees of FASTRADE.

## Overview

- Lightweight & speed Online Trading Portal
- Python Asyncio & aiohttp
- MVT Pattern
- CRUD Operation
- The Web page is categorized for two types of users
  - i) Customers
  - ii) Employees.

## Installation

Run the application program in either command line mode or by clicking the run button using any of the IDE's.

### Using Command line

- Open the command terminal and type the below.
- Install the Python library(if any) using pip.

```
$ pip install aiohttp
$ python models.py
```

## Run Application

Run the application using Command Prompt or any of the IDE.

### 1.1 Using Command Prompt.

```
$ cd trading_app_master
$ python models.py
```

Copy `http://0.0.0.0:8080` & paste it your browser. You can now see that the application is launced.

### 1.2 Using IDE

- I have used PyCharm as IDE.
- If you use Pycharm, import the project to Pycharm using `File -> Open -> Your Project folder`.
- Create a configuration available in the top right of the Pycharm Project.
- Provide the Python interpreter. I have given the Project default which is Python 3.7 and click Apply.
- Click the Run button and launch the application.
- If you have troubles running the application using Pycharm go to [run configurations](#) to know more about creating a configuration.

## Steps to use Application

There are two types of users

1. The customer (normal/VIP user) who can order.
2. The Employee who receives instant notifications of these orders.

## For Users

### 1) Sign up

- New users signup with the necessary information. I have provided the signup validation.
- Username is considered as a unique field. So please provide a unique username.
- I have enabled the VIP user to track VIP users.
- After you have provided the necessary information a sign-up successful notification is received.

### 2) Sign in

- Please click Existing member? [Log in](#) for the next step
- Provide the same username and password you have provided during signup.

### 3) Place an Order

- You will be given access to a variety of products available in-store currently and a facility to order it online or not.
- Click [Place an Order](#) button once you have decided the orders.
- A new page is now received notifying that your order is received confirming with your unique order number.

### 4) Check order status

- Click [check your order status](#) link to update or receive notifications regarding the order.
- You can now update Order / Return Order/ Cancel order.

#### 4.1) Update Order

- Update order will now update your order with order quantity. Give yourself a try.
- An update order received confirmation page is received. Click [Check](#) button to see the updated order.
- Status is now changed to “Updated”

#### 4.2) Return Order

- You can return your specific product with order quantity. Give yourself a try.
- A notification page is received with the order return on the progress page. Click the [Check](#) button to see your order is now updated with the returned quantity.
- Status is now changed to “Returned” unless all the orders are returned. Or else you get the “Updated” status.

#### 4.3) Cancel Order

- A cancel order option is to cancel the order. Give yourself a try.
- A cancel order in the progress notification page is received. Click the [Check](#) button to see the order is now cancelled.
- Status is now changed to “Canceled”

### 5) Order history

- Order history contains the history of all your orders that the customer has ordered.
- You can track any order using the Track your order facility.

#### 5.1) Track your Order

- Copy the order number and click the **Track Order** button.
- You will be navigated to the order status page with all the necessary information of the order and give access to update/return/cancel the order if you want.

## 6) Update Profile

- You can update your profile updating your password, email address and Delivery Address.
- Click **Update Profile** and your profile can now be updated.

## For Employees

### 1)Sign in

- Go to **http://0.0.0.0:8080**.
- Click the login button available in Employee of FASTRADE? **Log in**
- Select the type of Team you belong to. User validation is not available in this current version. Hence any user can log in to any respective team.
- Provide the same username and password you have provided during signup.

### 2)Notification

- **Type of Teams - Sales Team, VIP Assistance, Emergency Team, Returns Team, Large Orders Team, Products Team, Delivery Team, Refunds Team**
- Each team has it's functional design receiving notifications based on the types of the customer orders. Please check the functional document to know more about the roles & responsibilities of each teams.

## Futute enhancements

- Username validation and password encrpytion.
- Session validation enabling user login and logout.
- Use Django Rest API for backend & Reactjs for frontend.
- Use Docker containerization
- Move to AWS cloud

## Github code- **Online Trading App**

## Extra Features added

- I have added bootstrap in front end for a better look and feel.
- I have added javascript validations in pages like signup, return orders, etc.

## Granted Assumptions

- You can provide a Username. However unique username is recommended because your records from the database are taken as per your username. **While filling the signup form, please provide some unique username.**
- The username and password in the user login are not validated. You can login with any username and password, but the information that you have given during your signup is taken into consideration. So **when you login anytime, please provide the exact username that you have given while signing-up.**

- I have not provided a seperate sign-up form for employee. For employees any username is accepted. In this assignment, I have emphasized more on the design but it can be also implemented if needed.

## Out of Scope

- Passwords are not encrypted. Any password that you provide will be accepted.
- User and Employee authentication. Can be implemented in the future.

## Future enhancements

- Username validation and password encrpytion.
- Session validation enabling user login and logout.
- Use Django Rest API for backend & Reactjs for frontend.
- Use Docker containerization.
- Move to AWS cloud.