Documentation: Interactive Limit Order Book Web Application

Developed by IIT Kanpur: Team FAC

Abstract

This project is a Django-based interactive trading simulator that replicates real-time order book book dynamics, supporting **market, limit, iceberg (configurable visible quantities), immediate-or-cancel (IOC) and stop-loss orders.** It uses **AJAX** for asynchronous requests and Django channels (WebSockets) for push-based updates, ensuring a highly responsive experience. Designed for scalability, the system accommodates more than 100 concurrent users and features an intuitive interface to **place, modify, and cancel orders.** Instructors have complete control through an **administrative dashboard** for user management and live activity monitoring. All transactions are recorded in a **PostgreSQL** database and can be exported to **C.S.V.** for offline analysis. The front end, built with **HTML, CSS, and JavaScript**, combines technical accuracy with real-time interactivity to provide a hands-on educational tool for exploring market microstructure.

System Requirements

Minimum of 4 GB RAM	Network connectivity for WebSockets
Python 3.8+	Django 3.x or 4.x
Django Channels 3.x	PostgreSQL 14+
Redis (for channel layer)	Node.js and npm

How to Setup the Repository

Clone the Repository:

```
git clone <repo-link>                # Clone the Repo
cd STOCK_LIMIT_ORDER_BOOK            # Move to the folder
```

Create and activate virtual environment:

```
python -m venv venv
source env/bin/activate              # Linux & macOS
```

Install Python dependencies:

```
pip install -r requirements.txt     # Install Dependencies
```

Install Redis:

```
#For Linux
sudo apt update
sudo apt install redis-server
sudo systemctl start redis
redis-cli ping

#For macOS
brew install redis
brew services start redis
redis-cli ping                      # This returns 'PONG'
```

Database Setup

Database Settings (trading_system/settings.py)

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'trading_platform',
        'USER': 'trading_user',
        # Enter your password here
        'PASSWORD': 'your_password_here',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

For macOs

```
# Install via Homebrew
brew install postgresql

brew services start postgresql      # Start PostgreSQL service
psql postgres

# Execute SQL commands (same for all platforms)
CREATE DATABASE trading_platform;
CREATE USER trading_user WITH PASSWORD 'your_password';
GRANT ALL PRIVILEGES ON DATABASE trading_platform TO
   trading_user;
\q                                  # Exit psql

cd trading_system
python3 manage.py makemigrations
python3 manage.py migrate           #Migrations
```

For Linux

```
# Update package list (Debian/Ubuntu)
sudo apt update

# Install PostgreSQL and development libraries
sudo apt install postgresql postgresql-contrib libpq-dev -y

sudo systemctl start postgresql
sudo systemctl enable postgresql

sudo -iu postgres                 # Postgres user

psql                              # Launch psql

# Execute these SQL commands:
CREATE DATABASE trading_platform;
CREATE USER trading_user WITH PASSWORD
   'your_secure_password_here';
GRANT ALL PRIVILEGES ON DATABASE trading_platform TO
   trading_user;
\q                                # Exit psql
exit                              # Exit postgres user

cd ~/trading_system
python manage.py makemigrations
python manage.py migrate          # Migrations
```

Bulk User Generation

For bulk user generation, you must change `EMAIL_HOST_USER` and `EMAIL_HOST_PASSWORD` to their own credentials in `trading_system/settings.py` Ensure proper security measures when storing email credentials.

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'         # Replace with your SMTP
EMAIL_PORT = 587                      # Common port for TLS
EMAIL_USE_TLS = True
EMAIL_HOST_USER = 'mail@uni.ac.in'    # Replace with your email
EMAIL_HOST_PASSWORD = 'YYYYYY'        # Replace with your password
DEFAULT_FROM_EMAIL = EMAIL_HOST_USER
```

Locally Running the server

Run the command to create a superuser:

```
python manage.py createsuperuser
```

Provide username, email, and password when prompted. The Superuser can access the order book panel and has full control. **Note:** You will need to create the superuser only **"once"**. After that, you can skip the above step.

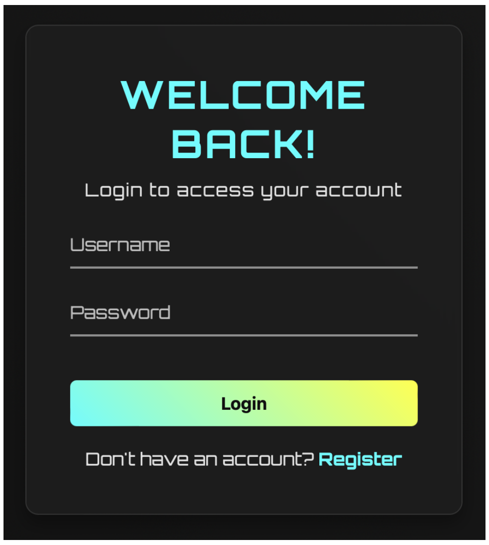
To start the server, run the following command:

```
python manage.py runserver
```

Application Workflow

Login Page

The login page is the secure entry point to the trading simulation platform. It features a clean, user-friendly interface where participants enter their credentials to gain access. Upon successful authentication, users are redirected to their respective dashboards. Failed login attempts prompt clear error messages, ensuring transparency and usability. The system distinguishes between participants (students) and administrators (faculty), with server-side redirection based on user roles. Built on Django's authentication framework, the login mechanism ensures robust session management and access control across the platform.



WELCOME BACK!

Login to access your account

Username

Password

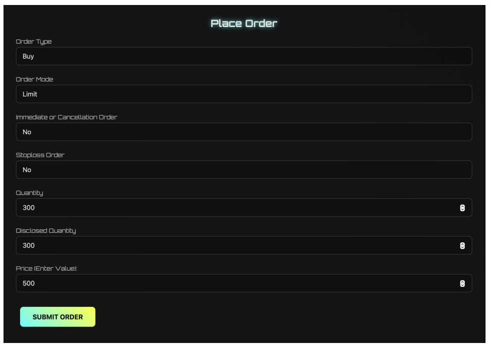
Login

Don't have an account? Register

Dashboard

Order Placement Interface

The upper section of the dashboard allows users to submit various order types—Market, Limit, Iceberg, Immediate or Cancel (IOC), and Stop Loss. The form dynamically adjusts based on the selected type, revealing relevant fields such as price, disclosed quantity (for iceberg orders), and trigger price (for stop-loss orders). Designed for clarity and speed, it supports efficient, error-free submissions during live simulations.



Place Order

Order Type: Buy

Order Mode: Limit

Immediate or Cancellation Order: No

Stoploss Order: No

Quantity: 300

Disclosed Quantity: 300

Price (Enter Value): 500

SUBMIT ORDER

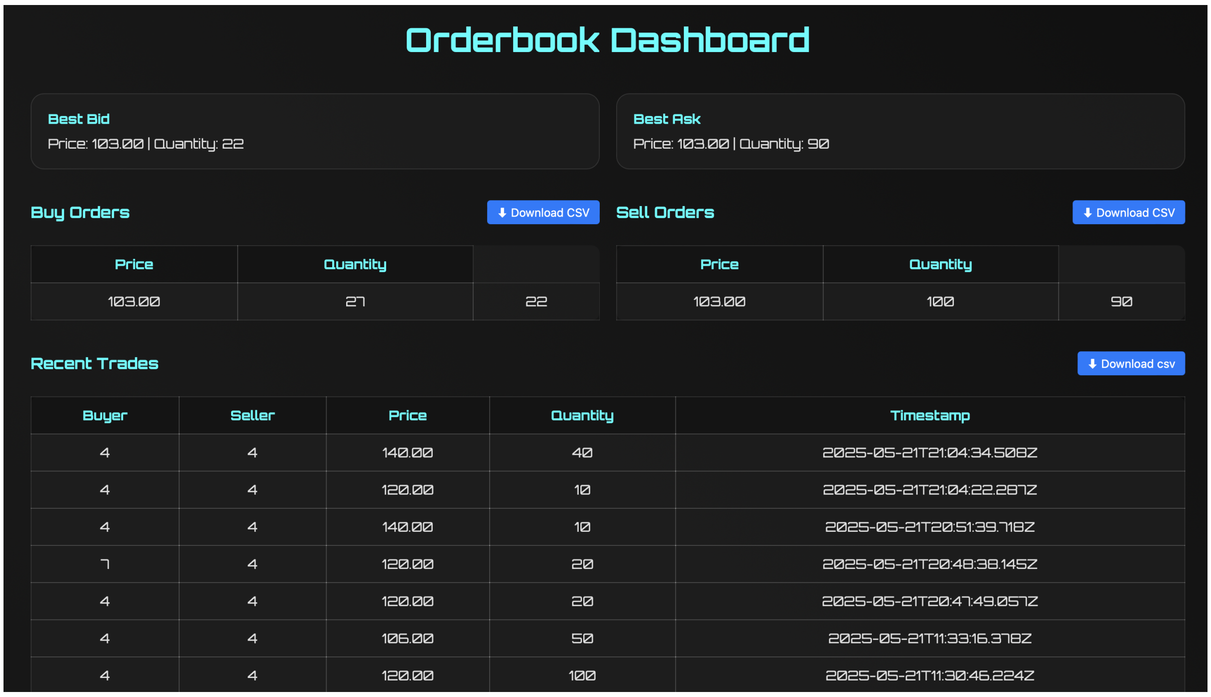
Order Book Visualization

Located below the order placement panel, the dashboard has three live-updating sections:

- **Completed Trades** – Shows the list of successfully matched and executed trades.
- **Active Orders** – Displays all open orders that are placed but not executed.
- **Stop Loss Orders** – Lists all stop-loss orders waiting for their trigger conditions.

These views are updated in real-time using WebSockets, ensuring participants receive immediate feedback and a continuously synchronized view of market activity.

Order Book Panel



Orderbook Dashboard

Best Bid
Price: 103.00 | Quantity: 22

Best Ask
Price: 103.00 | Quantity: 90

Buy Orders

Price	Quantity	
103.00	27	22

Sell Orders

Price	Quantity	
103.00	100	90

Recent Trades

Buyer	Seller	Price	Quantity	Timestamp
4	4	140.00	40	2025-05-21T21:04:34.508Z
4	4	120.00	10	2025-05-21T21:04:22.287Z
4	4	140.00	10	2025-05-21T20:51:39.718Z
7	4	120.00	20	2025-05-21T20:48:38.145Z
4	4	120.00	20	2025-05-21T20:47:49.057Z
4	4	106.00	50	2025-05-21T11:33:16.378Z
4	4	120.00	100	2025-05-21T11:30:46.224Z

This dedicated panel, accessible only to administrators, provides comprehensive real-time monitoring and control of the order book environment:

- **Best Bid and Ask:** Displays the bid and ask prices and their respective quantities, updated continuously.
- **Order Listings:** Separate lists for buy and sell orders, showing price, quantity, and disclosed quantity to give clear visibility into market depth. Also, there are convinient buttons for exporting the list of orders into a c.s.v. file.
- **Recent Trades:** A live feed of recently executed trades to track market activity and price movements along with the option to export the trade log as a c.s.v. file for offline analysis and record keeping.

Bulk User Upload

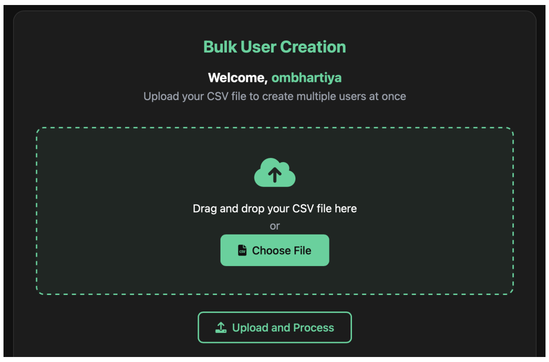

Bulk User Creation
Welcome, ombhartiya
Upload your CSV file to create multiple users at once
Drag and drop your CSV file here
or
Choose File
Upload and Process

Figure 1: Bulk User Upload Interface

Admin Panel – Uploading Users in Bulk

This section of the admin panel enables administrators to efficiently add multiple users at once by uploading a c.s.v. file.

The system reads the file, parses each row, and creates user accounts automatically. This is especially useful at the start of a term when many students need to be onboarded quickly.

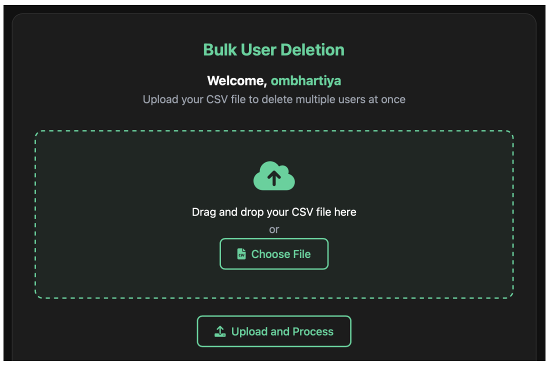
Users are assigned default roles (e.g., participant), and optional email confirmations can be configured post-upload.

C.S.V. Format Guidelines

The uploaded c.s.v. file must contain the following three columns in this exact order:

- **Name** – Full name of the user (e.g., `Om Bhartiya`)
- **Email** – Valid email address (e.g., `om@university.ac.in`)
- **Password** – Plaintext initial password (e.g., `ombhartiya`)

Bulk User Deletion


Bulk User Deletion
Welcome, ombhartiya
Upload your CSV file to delete multiple users at once
Drag and drop your CSV file here
or
Choose File
Upload and Process

Bulk Deletion Panel Overview

This interface enables administrators to delete multiple users in a single step by uploading a c.s.v. file. It's particularly helpful for periodic cleanup tasks, such as post-event user removal. The system validates the provided usernames and confirms successful deletions while skipping any invalid entries.

Guidelines for Deletion:

- The uploaded file must be in a .csv format and should contain a single column with the exact header: `Name`.
- Each row should contain a valid username that corresponds to a registered user.

Reset Password Window

The Reset Password window allows users to securely update their account credentials. The form is designed for simplicity and clarity, ensuring a smooth user experience while maintaining platform security.

Form Fields:

- **Current Password** – The user must enter their existing password to verify identity.
- **New Password** – A field to input the desired new password.
- **Confirm New Password** – Ensures the new password is typed correctly by matching both entries.

At the bottom of the page, a quick navigation link is provided to return directly to the Dashboard panel, enhancing usability for users who may have navigated from within an active session.

Contact

For queries, kindly contact the Finance and Analytics Club team of IIT Kanpur by mail at `fac_snt@iitk.ac.in`