Order Management System (Cross Option)

Project Overview

This project is a **Order Management System** built in C using MySQL for managing user data and trade records. The system allows users to sign in, register, and access trade records, it also allows user to simulate buying and selling of shares, while administrators can manage user data and view and edit trades. It is designed for CLI-based interaction, running in the Code::Blocks environment with MySQL database integration.

Features:

- User Registration, Viewing of Trades and creating market orders (buy and sell trades)
- Admin functionality for viewing users and trades
- MySQL database integration for persistent data storage
- Logging functionality for recording system events

Requirements

To run this project, you will need the following (for WINDOWS)

- **MySQL Database**: The project interacts with a MySQL database. Make sure MySQL is installed and running on your system.
- **C Compiler**: A C compiler (such as GCC) is required to compile the project code.
- **MySQL C API**: The project uses the MySQL C library (`libmysqlclient`), which must be installed.

Installation

Step 1: Install XAMPP (MySQL)

Download and install XAMPP on your system.

1. Download and install XAMPP on your system
2. Start MySQL on XAMPP Control Panel
3. To setup your database. You can manually create database or use the sqldump file that can be found in this package (db_oms_xo.sql)

```
CREATE DATABASE  db_oms_xo`
USE `db_oms_xo`;

/*Table structure for table `tbl_trades` */

DROP TABLE IF EXISTS `tbl_trades`;

CREATE TABLE `tbl_trades` (
  `id` int(12) unsigned NOT NULL AUTO_INCREMENT,
  `num_of_shares` int(12) DEFAULT NULL,
  `symbol` varchar(12) DEFAULT NULL,
  `market_order` varchar(12) DEFAULT NULL,
  `user_id` varchar(12) DEFAULT NULL,
  `total_amount` float DEFAULT NULL,
  `date_executed` datetime DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=31 DEFAULT
CHARSET=utf8mb4;


DROP TABLE IF EXISTS `tbl_user`;

CREATE TABLE `tbl_user` (
  `id` int(12) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(64) DEFAULT NULL,
  `age` int(4) unsigned DEFAULT NULL,
  `user_id` varchar(12) DEFAULT NULL,
  `date_created` datetime DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT
CHARSET=utf8mb4;
```

4. Edit the databse config file that can be found in this package(db_config.cdg).

```
host=localhost
user=root
password=
database=db_oms_xo
port=3306
```

Step 2: Install MySQL C API

1. Install the MySQL development libraries.

   Make sure to download and link the correct version of the MySQL C API.

Step 3: Setup Code::Blocks

1. Download and install Code::Blocks from the official website: Code::Blocks Downloads.
2. Configure the project settings to link the MySQL C API library:
   - Go to **Settings > Compiler**.
   - Under **Linker settings**, add `libmysql` to the linked libraries.
   - Add the MySQL include directory to **Search directories** under **Compiler** and **Linker**.

Step 4: Build and Run the Project

1. Clone the project repository or download the source files.
2. Open the project in Code::Blocks. (oms)
3. Build the project by selecting **Build > Build** or pressing `F9`.
4. Run the executable, and follow the on-screen prompts.

Usage

1. **Sign In / Register**: Upon running the system, you will be prompted to sign in or register.
   - If you are an admin, additional options will be available for user and trade management.
2. **Admin Functionality**: Admin users can view and manage user and trade records, logging each action for tracking purposes.
3. **Log file** can be found in oms folder “log+datetoday” ex. log09092024

REMINDER:; To create an ADMIN ROLE, you must register your USER ID as ADMIN

License

Contributions

Contributions are welcome! If you have suggestions or find any bugs, feel free to create an issue or submit a pull request.