

Stock Portfolio Management System

A simple application that lets users create and manage their stock portfolios using data fetched from the Yahoo Finance API. Users can store their custom portfolios in a MySQL database for later retrieval.

Features

- Fetch stock data from Yahoo Finance.
- Create custom stock portfolios.
- Manage (add/remove stocks from) existing portfolios.
- Display portfolio data.
- Save and retrieve portfolios from a MySQL database.

Prerequisites

- Python 3.x
- MySQL Server
- Required Python libraries: `yfinance`, `mysql-connector-python`.

Setup

1. Install the required Python libraries:

```
1 | pip install yfinance mysql-connector-python
```

2. Ensure that your MySQL server is up and running.
3. Setup your database and tables. Check it in *Configure your MySQL Database* Section.

Running the Program

1. Navigate to the directory containing the code files.
2. Run the `main.py` script:

```
1 | python main.py
```

3. Follow the on-screen prompts to manage your stock portfolio.

Configure your MySQL Database

1. Log in as MySQL Administrator:
Open your command-line terminal or MySQL client tool and log in as the MySQL Administrator. You can use the following command, replacing `root` with your administrator username.

```
1 | mysql -u root -p
```

You will be prompted to enter your administrator password.

2. Create a New User:

Use the following SQL command to create a new user. Replace `<new_username>`, `<new_password>`, and `<host>` with the desired username, password, and host for the new user. In this case, we're creating a user named 'Dsci560' with the password 'Dsci560@1234' and allowing connections only from localhost ('localhost' as the host):

```
1 | CREATE USER 'Dsci560'@'localhost' IDENTIFIED BY 'Dsci560@1234';
```

3. Create a New Database:

Use the following SQL command to create a new database named `Lab3_NAH`:

```
1 | CREATE DATABASE Lab3_NAH;
```

4. Grant Permissions:

By default, newly created users have no privileges. You need to grant the appropriate permissions to the user to allow them to perform the required operations. Here's an example command to grant a user access to the newly created `Lab3_NAH` database:

```
1 | GRANT ALL PRIVILEGES ON Lab3_NAH.* TO 'Dsci560'@'localhost';
```

5. Refresh Privileges:

After assigning permissions to the user, you need to refresh the MySQL privileges cache to apply the changes:

```
1 | FLUSH PRIVILEGES;
```

6. Exit MySQL Client:

When you've finished creating the new user, creating the `Lab3_NAH` database, and granting permissions, you can exit the MySQL client:

```
1 | EXIT;
```

You have now successfully created a new user with the username 'Dsci560' and the password 'Dsci560@1234', along with creating a new database named `Lab3_NAH`. The user can use their credentials to connect to MySQL and access the specified database. Be sure to manage users and permissions carefully to ensure security and data integrity.