

```
# Yannique Hecht
# HARVARD CS50 Week 8 - Finance - Implement a complex and fully
  functional website via which users can “buy” and “sell” stocks

#import
import os
from cs50 import SQL
from flask import Flask, flash, jsonify, redirect, render_template,
  request, session
from flask_session import Session
from tempfile import mkdtemp
from werkzeug.exceptions import default_exceptions, HTTPException,
  InternalServerError
from werkzeug.security import check_password_hash,
  generate_password_hash
from helpers import apology, login_required, lookup, usd

# Configure Flask application
[REDACTED]

# Ensure templates are auto-reloaded
[REDACTED]

# Ensure responses aren't cached
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

# Configure Jinja custom filter to display USD currency
[REDACTED]

# Configure session to use filesystem (instead of signed cookies)
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

# Configure CS50 Library to use SQLite database
[REDACTED]

# Make sure API key is set
[REDACTED]
```

[REDACTED]

1st ROUTE

[REDACTED]
[REDACTED]
[REDACTED]

[REDACTED]

Query information from database

[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

Pass list of lists to template page, template to iterate it to
extract data into table

[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

create a list with all information about the stock, append
it to a list of every stock owned by the user

[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

2nd ROUTE: BUY

[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

Ensure user has logged in / User reached route via POST (as by
submitting a form via POST

[REDACTED]
Obtain the data necessary for the transaction
[REDACTED]

```
[REDACTED]
```

```
# Control if stock symbol is valid
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# Calculate total value of transaction
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# Check if current cash is enough for transaction
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# Check if user already has one or more stocks from the same  
company
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# Insert new row into stock table
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# update row into stock table
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# update user's cash
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# Update history table
```

```
[REDACTED]
```

```
# Redirect user to index page with a success message
```

```
[REDACTED]
```

```
# User reached route via GET (as by clicking a link or via  
redirect)
```

```
[REDACTED]
```

```
# 3rd ROUTE: History
```

```
[REDACTED]
```

```
# query database with recent transactions history
```

```
[REDACTED]
```

```
# pass list of lists to template page, template to iterate to  
extract data into table
```

```
[REDACTED]
```

```
# create a list with all tinformation about transaction,
```

```
[REDACTED]
```

```
# redirect user to index page
```

```
[REDACTED]
```

```
# 4th ROUTE: LOGIN
```

```
[REDACTED]
```

```
# Forget any user_id
```

```
[REDACTED]
```

```
# User reached route via POST (as by submitting a form via POST)
```

```
[REDACTED]
```

```
# Ensure username was submitted
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# Ensure password was submitted
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# Query database for username
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# Ensure username exists and password is correct
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# Remember which user has logged in
```

```
[REDACTED]
```

```
# Redirect user to home page
```

```
[REDACTED]
```

```
# User reached route via GET (as by clicking a link or via  
redirect)
```

```
[REDACTED]
```

```
return render_template("login.html")
```

```
# 5th ROUTE: LOGOUT
```

```
[REDACTED]
```

```
[REDACTED]
```

```
[REDACTED]
```

```
# Forget any user_id
```

```
[REDACTED]
```

```
# Redirect user to login form
```

[REDACTED]

6th ROUTE: QUOTE

[REDACTED]
[REDACTED]
[REDACTED]

[REDACTED]

User reached route via POST (as by submitting a form via POST)

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

User reached route via GET (as by clicking a link or via redirect)

[REDACTED]

[REDACTED]

7th ROUTE: REGISTER

[REDACTED]
[REDACTED]

[REDACTED]

Forget any user_id

[REDACTED]

User reached route via POST (as by submitting a form via POST)

[REDACTED]

Ensure username was submitted

[REDACTED]

[REDACTED]

Ensure password was submitted

[REDACTED]

[REDACTED]

Ensure confirm password is correct

```
[REDACTED]
```

```
# Query database for username if already exists
```

```
[REDACTED]
```

```
# Insert user and hash of the password into the table
```

```
[REDACTED]
```

```
# Query database for username
```

```
[REDACTED]
```

```
# Remember which user has logged in
```

```
[REDACTED]
```

```
# Redirect user to home page
```

```
[REDACTED]
```

```
# User reached route via GET (as by clicking a link or via  
redirect)
```

```
[REDACTED]
```

```
# 8th ROUTE: SELL
```

```
[REDACTED]
```

```
# User reached route via POST (as by submitting a form via POST)
```

```
[REDACTED]
```

```
# collect relevant informations
```

```
[REDACTED]
```

```
[REDACTED]

# Update stocks table
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

# delete stock from table if we sold every unit we had
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

# stop the transaction if the user does not have enough
stocks
[REDACTED]
[REDACTED]

# otherwise update with new value
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

# calculate and update user's cash
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

# Update history table
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

# Redirect user to home page with success message
[REDACTED]
[REDACTED]

# User reached route via GET (as by clicking a link or via
```


1150

© 2006 The Authors
