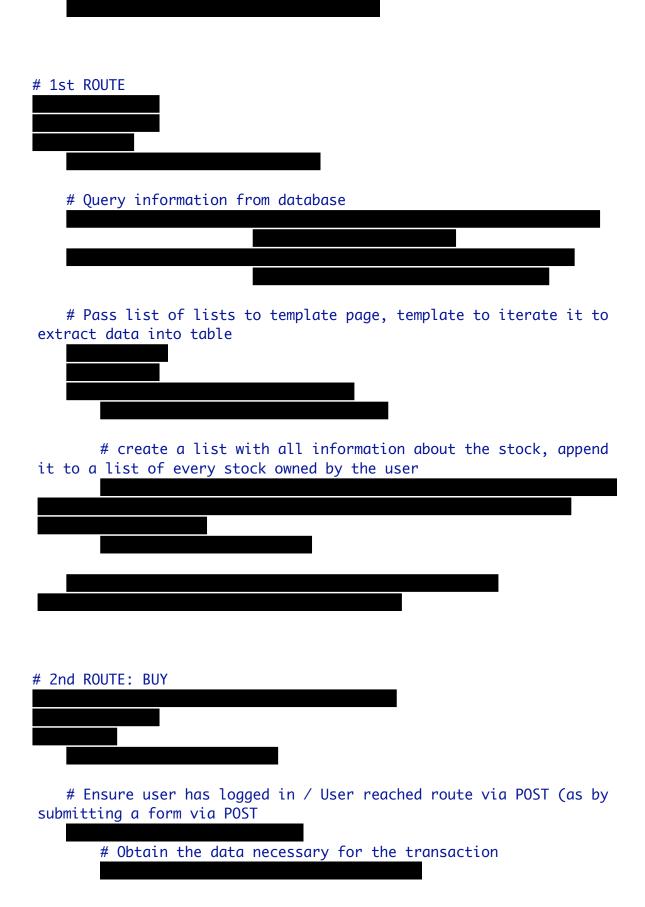
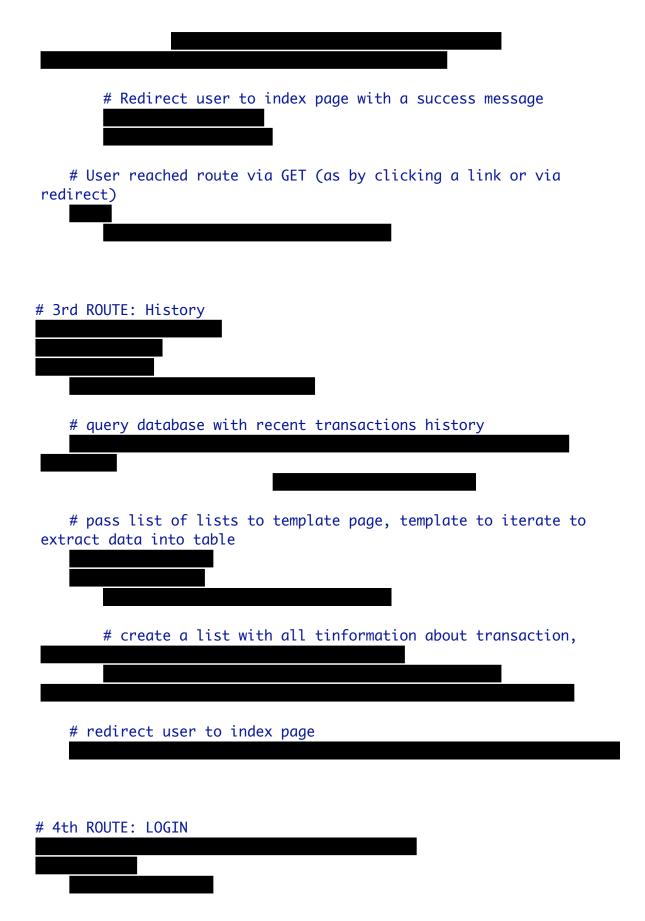
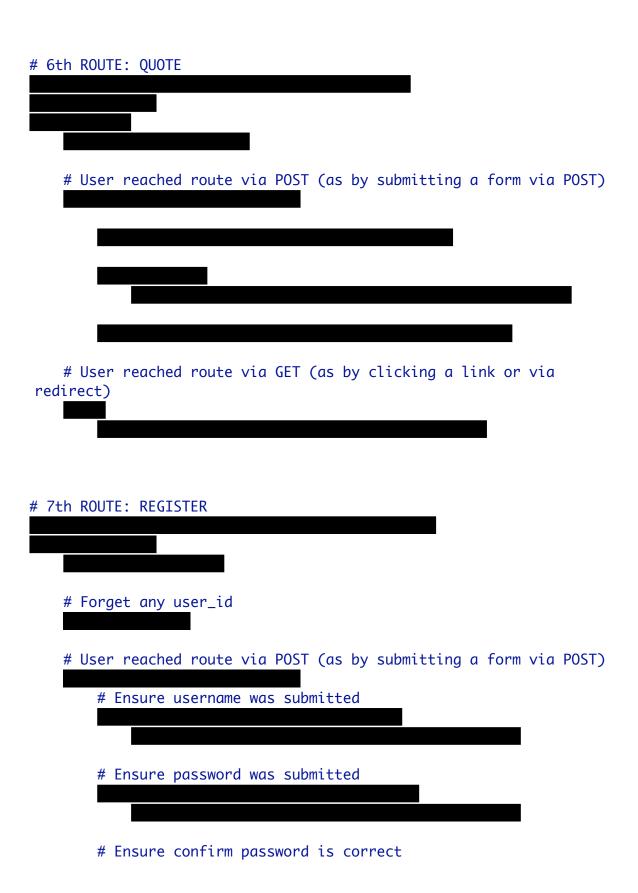
```
# Yannique Hecht
# HARVARD CS50 Week 8 - Finance - Implement a complex anf fully
functional website via which users can "buy" and "sell" stocks
#import
import os
from cs50 import SQL
from flask import Flask, flash, jsonify, redirect, render_template,
request, session
from flask_session import Session
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
# Ensure templates are auto-reloaded
# Ensure responses aren't cached
# Configure Jinja custom filter to display USD currency
# Configure session to use filesystem (instead of signed cookies)
# Configure CS50 Library to use SQLite database
# Make sure API key is set
```

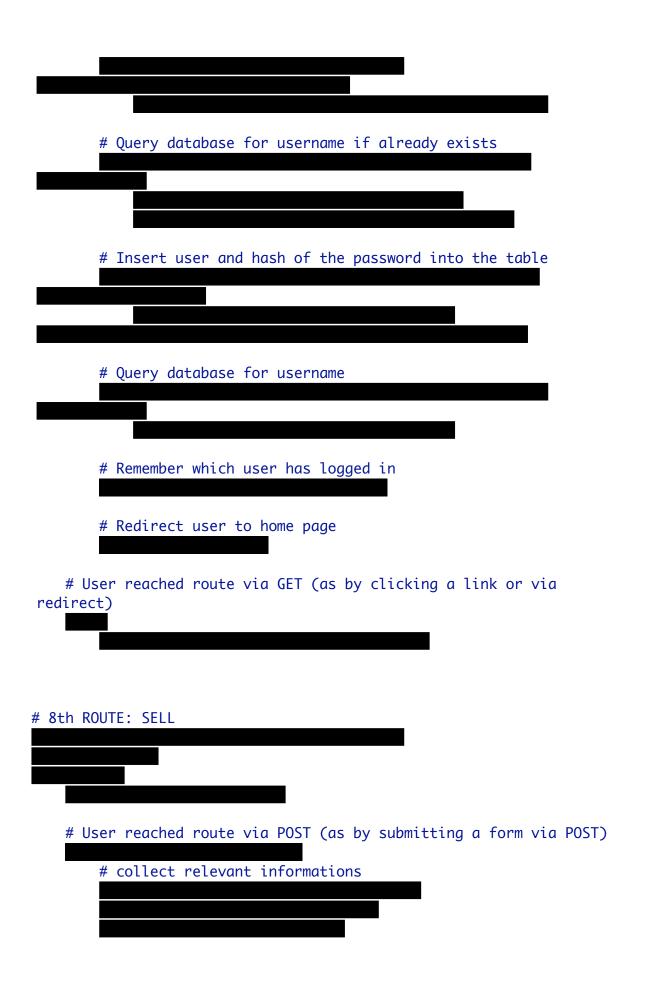


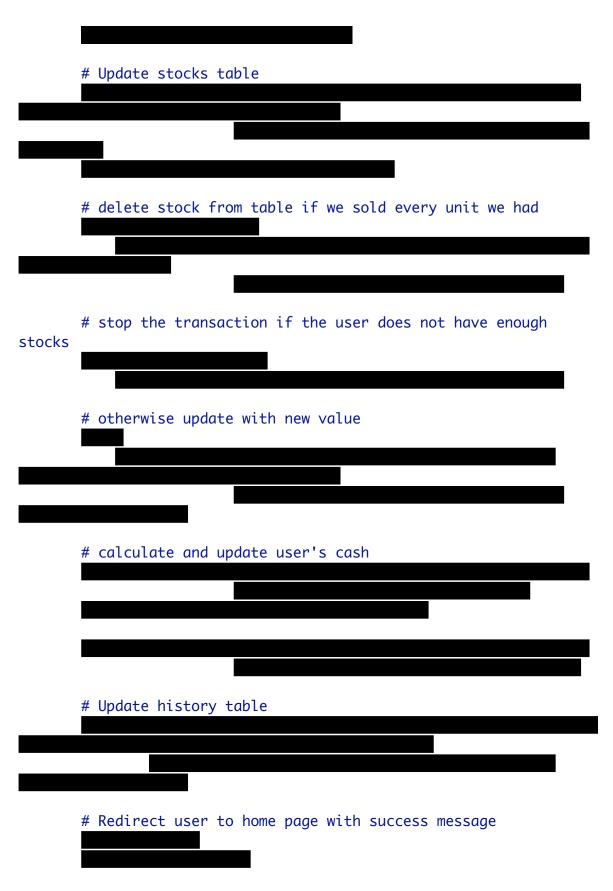
```
# Control if stock symbol is valid
       # Calculate total value of transaction
       # Check if current cash is enough for transaction
       # Check if user already has one or more stocks from the same
company
       # Insert new row into stock table
       # update row into stock table
       # update user's cash
       # Update history table
```



```
# Forget any user_id
   # User reached route via POST (as by submitting a form via POST)
        # Ensure username was submitted
        # Ensure password was submitted
        # Query database for username
        # Ensure username exists and password is correct
        # Remember which user has logged in
        # Redirect user to home page
    # User reached route via GET (as by clicking a link or via
redirect)
       return render_template("login.html")
# 5th ROUTE: LOGOUT
   # Forget any user_id
    # Redirect user to login form
```







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

