

Stock Manager Task

Background

You want to buy and sell some shares on the stock market and need a simple tool for personal use to help you keep track of them.

Task Summary

Your task is to build a simple web app that will allow a user to manage a fake stock portfolio and a cash balance. As a minimum, users will need to be able to:

- Add or remove amounts to their cash balance, which is then used for buying shares.
- Execute “buy” or “sell” orders for a quantity of stock. For example, the user may wish to buy 3 GOOG shares. The amount of shares times the current price-per-share would then be deducted from their cash balance.
- Current cash and share holdings should be visible to the user.

Requirements

- The stock prices must use real data - this is readily available on the internet for free through many sources.
- The application must persist state across web page refreshes. (It need not, however, persist state across server restarts).

Optional

These are just some ideas. They may not all be suitable or implementable in the short time frame available. You may have other better ideas, too.

- Dynamic client side (web 2.0, single page app, no full page reloads)
- Give the user a live quote as they type a ticker
- Persist the server state (e.g. to a file or db) so the user’s information isn’t lost on server restart.
- Net total asset value based on current stock prices

Please timebox this task - try to think of what key features you can do in 2 - 5 hours, and implement those features well. This is not meant to be a production quality system. Document shortcuts and simplifying assumptions.

Resources

- Use any language / libraries / framework you like.
- If you have any questions about the nature of this application please ask.

Assessment

- Think independently, build something good based on a fairly vague spec.
- Ability to work independently & leverage resources at your disposal to save effort.
- Product completeness, clean UI.
- Clear, simple, readable, correct, not overly-engineered code.
- **Ability to prioritise, to make good decisions about what to leave out versus what to include, given the time and scope constraints.**

Deliverables

- A **zip/tar.gz file** containing the full source code and any other resources
- **Clear instructions for running the application and installing any dependencies.**
- A hosted demo we can play with. This should be as simple as pushing something to heroku, app-engine or amazon tiny instance (all free). **Please send us the zip of your application first.**
- A **short** report (half page, email is fine) outlining:
 - Overall process you went through, any key decisions made.
 - Shortcuts, simplifying assumptions, known bugs, etc.
 - What slowed you down, what was unexpected
 - Any problems you ran into, and how you solved them