Stock App – Final Project

Questions on project:

1. Issue: every time I click on a link to a route the store gets reset (I lose the user login data)?? Have to use NavLink – not the nav links from Bootstrap
2. I may have to do quite a bit of data transformation for the charts. Should I have Rails do this work or should the reducer or action object / function do this work?
3. Should all of my fetch data go to the database? Since we are mainly displaying data not sure if it needs to be saved…

Todo List:

1. Display company data in company container (already added to store)
2. Fetch and display stock chart in company container
3. Hold Off for now: CONVERT TO STYLED-COMPONENTS INSTEAD OF BOOTSTRAP? MIGHT BE NICE TO USE A DIFFERENT CSS
4. X THUNK TO THE FETCH REQUESTS / ACTIONS!!
5. X Build seed data
   1. Add full list of tickers to database from Excel Files?
6. Code basic user interface / front end
   1. Homepage
   2. X Login form
      1. X For testing automatically log in user1 (andrew)
   3. X Navbar
   4. X Stock pick form
   5. Download chart data and load to database / view
   6. Watchlist view / form
      1. Watchlist listing (list of watchlists)
      2. Watchlist detail (table view of stocks on watchlist)
   7. New user form / route (/signup)
   8. Started: Download company data and load to database / view
7. Later: Code thunk ‘getting data’ actions to display to user that data is being downloaded…

Summary: It can be difficult to see the actual movement of stocks for custom periods of time. Most stock charts are set up to show price movement over a day, minute, or hour, but not over say a week or month. This app is designed for a user to be able to select any period of time and get the movement of a stock or set of stocks. This is useful to see trends in certain stocks or industries.

Features / User Experience:

* Modern and clean design
* Easy to use (reduce clutter and keep interface minimalistic)
  + Hide menus / options when not needed (drop down menus, etc)
* User Log in & Sign Up
* NavBar that dynamically updates various data and information
* Stock Selection Form
  + Have list of 500+ stocks to choose from?
  + Enter or select a stock code (or search by company name)
  + Select Dates or choose from several date range options
  + Ability to Save a ‘watchlist’ of stocks for quickly review
* Stock Data View
  + Summary of up to 25 stock data from watchlist with basic price movement
    1. Click to view charts and additional data
  + Candlestick chart with stock data (daily option only at this time)
  + Stock increase / decrease over period including dollar amount and percent
  + Current day stock price and normal statistics
    1. Low for year; high for year; increase YTD, etc
    2. **Need to research how other websites show this data**
  + Future: include comparable industry stock trend for comparison
* View company data including:
  + Summary of company (if possible): Use API: Company Profile2
    1. Country
    2. Exchange
    3. Market cap?
    4. Name
    5. Outstanding shares
    6. Ticker
    7. Weburl
    8. Logo
    9. industry
  + Company statistics (research other sites): Use API: Basic Financials
    1. 3 month avg trading volume
    2. 52 week high
    3. 52 week high date
    4. 52 week low
    5. 52 week low date
  + (future) Last 2 earnings (if possible): Use API: Financials As Reported
* Option to save data to user account? This will persist to database
* Future: Scan list of stocks and return sorted list by largest increase / decrease
  + Be able to look for 3 day trend, week trend, etc…

Pages:

1. Home
2. Login
3. Select\_stock
4. Company
5. Watchlist

Database: See Excel document

Redux Store:

* Questions:
  + Should watchlists be nested under user?
  + Should chart be nested under company?

User: {see fields in table},

Status: {loggedIn} 🡪 need to work on this…

Symbols: [{id, symbol, securityName}],

Watchlists: [{id, name, description, userId, watchlistsCompaniesId}],

Companies: [{see fields in table}],

Chart: {id, companyId, chartData: []} 🡪 just one chart loaded at a time?

Components & Local State:

* App (container): Contains the store and parent to all other containers
* User (container): handles login and other user functionality
  + Login: Login form
  + Select stock form
* Watchlists (container):
* NavBar: Render dynamic navigation bar
* Company (container):
  + Chart: Displays chart
  + Statistics: Displays company statistics