

Project 2: C\$75 Finance

Monday, 6 April 2009, noon ET

Goals.

- Introduce you to SQL, MySQL, and phpMyAdmin.
- Introduce you to JavaScript.

Recommended Reading.

- *JavaScript: Your visual blueprint for building dynamic Web pages*: chapters 1 – 7.
- *PHP & MySQL: Your visual blueprint for creating dynamic, database-driven Web sites*: chapters 7 and 8.
- <http://www.investopedia.com/university/stocks/>
- http://www.php-editors.com/articles/sql_phpmyadmin.php
- <http://www.w3schools.com/sql/>
- <http://dev.mysql.com/tech-resources/articles/intro-to-normalization.html>
- <http://dev.mysql.com/books/mysqlpress/mysql-tutorial/ch10.html>

Academic Honesty

All work that you do toward fulfillment of this course's expectations must be your own unless collaboration is explicitly allowed (*e.g.*, by some problem set or the final project). Viewing or copying another individual's work (even if left by a printer, stored in an executable directory, or accidentally shared in the course's virtual terminal room) or lifting material from a book, magazine, website, or other source—even in part—and presenting it as your own constitutes academic dishonesty, as does showing or giving your work, even in part, to another student.

Similarly is dual submission academic dishonesty: you may not submit the same or similar work to this course that you have submitted or will submit to another. Moreover, submission of any work that you intend to use outside of the course (*e.g.*, for a job) must be approved by the staff.

You are welcome to discuss the course's material with others in order to better understand it. You may even discuss problem sets with classmates, but you may not share code. If in doubt as to the appropriateness of some discussion, contact the staff.

You may even turn to the Web for instruction beyond the course's lectures and sections, for references, and for solutions to technical difficulties, but not for outright solutions to problems on problem sets or your own final project. However, failure to cite (as with comments) the origin of any code or technique that you do discover outside of the course's lectures and sections (even while respecting these constraints) and then integrate into your own work may be considered academic dishonesty.

All forms of academic dishonesty are dealt with harshly.

Grades.

Your code (CSS, PHP, SQL, XHTML, *etc.*) will be evaluated along the following axes.

Correctness. To what extent is your code consistent with our specifications and free of bugs?

Design. To what extent is your code written well (*i.e.*, clearly, efficiently, elegantly, and/or logically)?

Style. To what extent is your code readable (*i.e.*, commented and indented with variables aptly named)?

panel.cs75.net.

- ☐ Surf on over to `http://panel.cs75.net/`, log in, and follow the link to **Subdomain Management** under **Your Account**. On the page that appears, add a subdomain called **project2** to your domain. Not only will the panel automatically create A records for `project2.domain.tld` and `www.project2.domain.tld` for you, where `domain.tld` is your domain, it will also create a subdirectory called `project2` in your `public_html` directory. All of your work for this project must ultimately reside within that subdirectory.
- ☐ Return to your panel's home page and select **Password Protected Directories**, also under **Your Account**. On the page that appears, click **Find a Directory to Password Protect**. You should then see the contents of your `public_html` directory. In the row containing your `project2` subdirectory, click **Protect** in the column labeled **Action**. On the page that appears, check the box next to **Protection Enabled**, and then fill out the rest of the form. Any username and password are fine, as is any value for **Protected Directory Prompt**, which is just an aesthetic detail. When done, click **Save**. The panel will proceed to create (or edit) at least two files for you: `~/public_html/project2/.htaccess`, which tells Apache to password-protect the directory, and `~/domains/domain.tld/.htpasswd/public_html/project2/.htpasswd`, which contains your choice of username and password, separated by a colon, with the latter encrypted. Had you not a convenient panel, you could also have just created files like those by hand.
- ☐ Return to your panel's home page and select **MySQL Management**, also under **Your Account**. On the page that appears, click **Create a new Database**. Provide a name, username, and password for your database, then click **Create**. Make note of all details on the page that appears! You must use that database for this project.

investopedia.com.

- ☐ If you're not quite sure what it means to buy and sell stocks (*i.e.*, shares of a company), surf on over to the URL below for a tutorial.

`http://www.investopedia.com/university/stocks/`

You're about to implement C\$75 Finance, a Web-based tool with which you can manage portfolios of stocks! Not only will this tool allow you to check stocks' prices and portfolios' values, it will also let you buy (okay, "buy") and sell (fine, "sell") stocks!

finance.yahoo.com.

- ☐ Just the other day, I received the below.

To: malan@post.harvard.edu
Subject: Want to Turbocharge Your Portfolio?

Our weekly gift to you!
Big Press Release just out.

Inf inx Ven t ures (INFX.PK)
Price: 0.83
5 day expected price 1.90
This one will run for sure

Starting a new Marketing Campaign.
The World will know about this company
Should you wait until to late?

This one did very well during last marketing campaign. Very Well!

With this News we expect prices to exceed our projected price of 1.90

That's quite the gift! Let's get in on this opportunity now. Head on over to Yahoo! Finance at the URL below.

<http://finance.yahoo.com/>

Type the symbol for Infinex Ventures, **INFX.PK**, into the text field in that page's top-left corner and click **GET QUOTES**. Odds are you'll see a table like the below.



Wow, um, good thing I didn't buy. Anyhow, notice how Yahoo reports a stock's most recent price (**Last Trade**) and more. Notice, too, that Yahoo lets you download this data. Go ahead and click **Download Data** to download a file in CSV format (*i.e.*, as comma-separated values). Open the file in Excel or any text editor (*e.g.*, Notepad or TextEdit), and you should see a "row" of values, all excerpted from that table. It turns out that the link you just clicked led to the URL below.

`http://download.finance.yahoo.com/d/quotes.csv?s=INFX.PK&f=s11d1t1c1ohgv&e=.csv`

Notice how Infinex's symbol is embedded in this URL (as the value of the HTTP parameter called `s`); that's how Yahoo knows whose data to return. Notice also the value of the HTTP parameter called `f`; it's a bit cryptic (and apparently undocumented), but the value of that parameter tells Yahoo which fields of data to return to you. For an unofficial list of the fields that can appear in that otherwise cryptic string, surf on over to the URL below.

`http://www.gummy-stuff.org/Yahoo-data.htm`

It's worth noting that a lot of websites that integrate data from other websites do so via "screen scraping," a process that requires writing programs that parse (or, really, search) HTML or XHTML for data of interest (*e.g.*, air fares, stock prices, *etc.*). Writing a screen scraper for a site tends to be a nightmare, though, because a site's markup is often a mess, and if the site changes the format of its pages overnight, you need to re-write your scraper.

Thankfully, because Yahoo provides data in CSV, C\$75 Finance will avoid screen scraping altogether by downloading (effectively pretending to be a browser) and parsing CSV files instead. Of course, Yahoo's quotes aren't actually real-time but delayed (likely by a few minutes), but, hey, they're free.

`project2.domain.tld`.

- ☐ Your mission for this project is to implement C\$75 Finance, a website that allows users to manage portfolios of stocks. The overall design and aesthetics of this site are ultimately up to you, but we require that your site meet some requirements. All other details are left to your own creativity and interpretation.

Feature Requirements.

- ☐ Your site must require that a user log in with a username and password in order to see or do anything (except, obviously, log in or register).
- ☐ Your site must allow a user to register for an account. A user's username must be a syntactically valid email address.¹ A user's password must be at least six characters, and it cannot be entirely alphabetic or entirely numeric. Upon registering, a user should receive a free gift: \$10,000 in cash.
- ☐ Your site must allow a user to get a quote (*i.e.*, look up Last Trade) for a stock by providing its symbol.
- ☐ Your site must allow a user to buy shares of a stock by providing its symbol; your site must allow a user to buy more shares of a stock that he or she already owns. A user may not buy fractions of shares.
- ☐ Your site must allow a user to sell shares of a stock that he or she already owns; for simplicity, you may require that a user sell all or none rather than some. A user may not sell fractions of shares.
- ☐ Your site must allow a user to check the current value of his or her portfolio (*i.e.*, his or her cash plus his or her stocks' value based on Last Trade).
- ☐ Your site must allow a user to view his or her history of transactions (*i.e.*, the date and time of each buy or sell, the number of shares bought or sold, the price at which shares were bought or sold, and the total value bought or sold).
- ☐ Your site must perform client-side validation, where possible, of user input related to a buy or a sell. For instance, if some text field must contain a non-negative integer (*e.g.*, number of shares to buy or sell), you must reject attempts to submit invalid input (as by admonishing the user with an alert) or prevent them from typing anything non-numeric at all.
- ☐ On any page designed to take user input, you should give focus (via JavaScript) to the first field requiring the user's attention (*e.g.*, the username field on your login page).
- ☐ Your site must perform rigorous server-side error-checking. Under no circumstances should we be able to crash your site or induce unreasonable behavior. Letting us sell more shares than we own is not, shall we say, reasonable. We will bang on your code and try to find faults; do not let us succeed.

¹ Rather than validate email addresses per RFCs 2821 and 2822, you may, for simplicity, assume that an email address must be of the form `username@domain.tld` or `username@subdomain.domain.tld`, where `tld` contains only alphabetical characters, `subdomain` and `domain` contain only alphanumeric characters, and `username` contains only alphanumeric characters, dots, underscores, hyphens, and/or pluses.

Technical Requirements.

- ☐ Your site must live at `http://project2.domain.tld/`, where `domain.tld` is your own domain.
- ☐ Your site must have a file called `config.php` in `~/public_html/project2/` that contains any and all constants and other configuration options that we might want to examine or alter during evaluation of your site (*e.g.*, your database's name, username, and password).
- ☐ All PHP files must be `chmod'd 600`.
- ☐ Your site must pull its real-time data (prices, *etc.*) from Yahoo! Finance by parsing its CSVs; you may find `http://us3.php.net/fgetcsv` of interest.
- ☐ Your database tables must have constraints and indexes defined where appropriate.
- ☐ You must avoid race conditions by using SQL transactions or locks.
- ☐ You should avoid inefficiencies and redundancies in your database tables (*e.g.*, by keeping them in third normal form).
- ☐ You should avoid redundant XHTML; factor out XHTML common to multiple pages using PHP functions or "templates" (like we do for the course's website).
- ☐ Your XHTML must be valid (or "tentatively" valid), unless some feature of your site requires otherwise (for the sake of some browser); explain in XHTML comments any intentional invalidities. Your XHTML should also be as pretty-printed as possible. Your CSS need not be valid.
- ☐ Your PHP must be extensively commented and be as pretty-printed as possible.
- ☐ You may use a WYSIWYG editor to generate XHTML and/or CSS that you would like to use in your site.
- ☐ If you incorporate or adapt snippets of code from the Web into your project (*e.g.*, examples from `php.net`), cite the code's origins with PHP comments.
- ☐ If you incorporate images from the Web into your project, cite the images' with XHTML comments.
- ☐ Your website must appear and behave the same on at least two major browsers, namely:
 - ☐ Chrome 0.x
 - ☐ Firefox 3.x
 - ☐ Internet Explorer 7.x
 - ☐ Opera 9.x
 - ☐ Safari 3.x

Optional Features.

Just for fun, feel free to add zero or more additional features to your site!

- ☐ Empower users to change their passwords.
- ☐ Require users to confirm their email addresses in order to register.
- ☐ Empower users who've forgotten their password to receive reminders via email.
- ☐ Email users "receipts" anytime they buy or sell stocks.

- ☐ Keep users' passwords encrypted in your database.
- ☐ Present users with links to recent articles related to companies in which they own stock. See <http://finance.yahoo.com/rssindex> for inspiration.
- ☐ Incorporate into your site more data than just Last Trade from Yahoo.
- ☐ Implement the ability to short stocks.
- ☐ ...

THE BIG BOARD.

- ☐ If you'd like to try out your own (short-term) trading skills and compete against classmates, surf on over to

<http://cs75.net/finance/>

and you'll find that \$10K awaits you if you follow the link to play THE BIG BOARD. You're welcome to register with fake usernames and tinker, but realize that our version C\$75 Finance (intentionally) does not implement these same specs. Look to our implementation for inspiration but not a complete roadmap!

Submission.

- ☐ Once done with your site, put together a readme at <http://project2.domain.tld/readme/>. Treat this readme as your opportunity not only to explain but to justify your design decisions. Tell us why you modeled your database tables as you did. Tell us why you chose, say, select menus over radio buttons for some feature. Tell us with which two (or more) browsers we should evaluate your site. And give us an overall sense of how your site works (e.g., tell us which files do what). But still be succinct; keep this readme to just a few paragraphs in length.
- ☐ By this project's deadline, submit your website by SSHing to your domain and executing the command below in order to submit the contents of your `~/public_html/project2/` directory.

```
submit project 2
```

Thereafter, follow any on-screen instructions until you receive visual confirmation of your work's successful submission. You may re-submit as many times as you'd like, but take care not to re-submit after the project's deadline, lest your work be deemed late.