COMP 3512
Assignment #1: PHP Pages and API

Version: 1.0 Sept 18
Due Wed October 15, 2025 at midnightish

Overview

This assignment provides an opportunity for you to demonstrate your ability to generate a PHP-based data-generated web site. **You can work in pairs or by yourself (I recommend you do work in pairs if at all possible).** Please don't ask me to be in a group of 3.

The SQLite database you have been provided with contains information about fictional users and real stocks. It has tables on users, companies/stocks, user portfolio, and historical stock price data for the first three months of 2019.

Beginning

I feel foolish saying this in a third-year university course, but it is your responsibility to read all the assignment instructions thoroughly and carefully. If you are unclear about something, ask me. But before you do, read the material in question again!

Grading

The grade for this assignment will be broken down as follows:

Visual Design and Styling	20%
Programming Design	10%
Functionality (follows requirements)	70%

Files

The starting files are provided on D2L.

Periodically, I may upload a revised version of the database if errors are found in the data.

GitHub

You need to have “a single source of truth” when it comes to your source code. That is, there must be a “master” location that contains the definitive version of your source code. If you are working as a pair, then one of you will have to decide whose github account has the master location. But even as a pair, each group member will need their own Github account.

Your repo on Github can be either private or public. The free GitHub account doesn’t allow private repos; if you sign up for the Student Developer Pack (https://education.github.com/pack) you can have free private repos while a student. The other way is to email me, and I can create a private repo under our department’s github organization (https://github.com/MountRoyalCSIS) for your group. You would need to supply the github names or emails for each member. You can also decide to use a public repo.

You will want to push out updates fairly frequently (1-2 times a day when working on it, or more frequently if lots of work is being done). I will be examining the commits when marking. You can push your content to GitHub via the terminal, using the following commands (not the stuff in square brackets though, as those are comments):

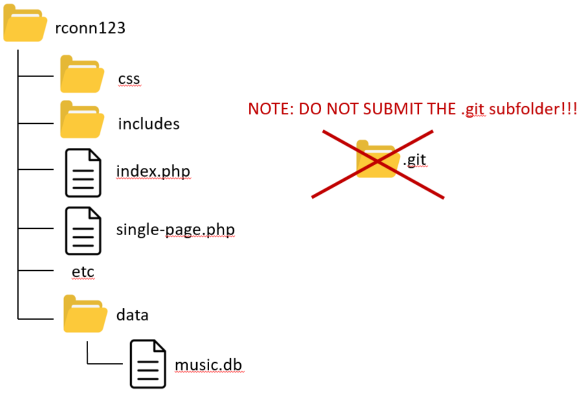
```
git init     [only need to do this one command once for your assignment]
git add *
git commit –m "Fixed the rocket launcher"     [alter message and name as appropriate]
git remote add origin https:... your-repo-url.git   [do this just once]
git push –u origin master   [login using your own individual github credentials]
```

For more information about Git and GitHub, read pages 571-577 of textbook (2nd Edition). There are many online guides to git (for instance, https://guides.github.com/introduction/git-handbook/).

Submitting

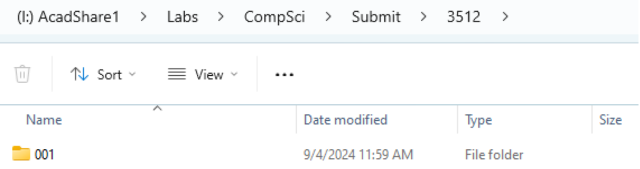
NOTE: I may make changes to these submission instructions closer to the due date. Right now, you will submit your assignment by copying a zip containing your assignment to MRU submit drive (it is the I: drive).

1. Your assignment **must** be in a folder whose name is the same as your user id (e.g., rconn123). If you are in a group, pick one your user ids for the folder name. This folder **must** contain your index.php pages and your other files. They must **not** be nested within some other folder! For instance, your folder must look something similar to the following (this is from a previous year's assignment, so don't worry about filenames here):



NOTE: DO NOT SUBMIT THE .git subfolder!!!

2. You can also submit your zip file using the mymru web portal to upload your assignment.

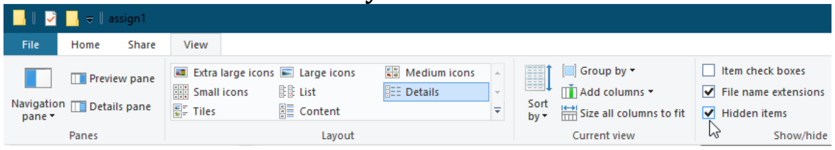


(I:) AcadShare1 > Labs > CompSci > Submit > 3512 >

Name	Date modified	Type	Size
001	9/4/2024 11:59 AM	File folder	

3. Make sure you use the SQLite version of the database rather than the MySQL version. That is, make sure your connection string is correct for the SQLite file which must be in a folder named data.

4. Do **NOT** submit the .git subfolder, if you have one!!! This folder contains literally hundreds of extra files, which means copying your submitted folder to my machine for marking takes 5 minutes instead of 30 seconds. If you do this, I will simply give you a mark of zero. You may have system files hidden, so be sure to make them visible so you do not submit this folder by mistake:



□ Item check boxes
☑ File name extensions
☑ Hidden items

5. When you are ready to submit, I recommend making a fresh folder with the appropriate name in your htdocs folder. Copy the relevant files. Test pages (still work with the SQLite connection?). Check folder structure. Check there is no .git folder. Copy/Submit it.

Data Files

Data has been provided with an SQLite database. If you install the SQLite Viewer extension you can view the tables directly in Visual Code. The database consists of four tables: you are only interested in this assignment in some of them:

- **users**: Contains ten fictional users
- **portfolio**: Each user has a portfolio of stocks they own. The value of a user's portfolio can be determined by adding the value (closing stock price * amount of stock owned) for each of the stocks in their portfolio.
- **stocks**: Each stock represents a real company.
- **history**: The value of a stock changes over time. This table contains the real opening and closing value of a stock on a given day (in this case between start of Jan 2019 and end of March 2019.

The tables are organized with primary keys and foreign keys so you will need to use `INNER JOINS` where necessary.

Requirements

1. Your assignment will also consist of four PHP pages. I have provided you with simple screen captures to show you roughly the functionality required, but needless to say I would expect your assignment to be a lot more visually styled than that shown in the screen captures.
2. You must use the PHP and PDO techniques found in Lab14a to retrieve data from the provided SQLite database for all the pages (except of course the about page) in this assignment. You will NOT use the web APIs you will create for your other pages.
3. If you make use of CSS recipes you found online or via AI, you must provide references (i.e., specify the URL where you found it) via comments within your CSS file. **Failure to properly credit other people's work in your CSS could result in a zero grade.** You can make use of a third-party CSS library; if you do, be sure to indicate this on the home page. Attractive styling with your own CSS will result in a higher style mark than if you use a third-party library; poor styling with your own CSS will likely result in a lower style mark than if you use a third-party library.
4. **No JavaScript on this assignment.** In the second assignment, you will be making a variety of improvements to this assignment using JavaScript.
5. Most of the functionality in the app can be found in the sketches shown on the next few pages. The sketches provided are used to indicate functionality not the layout. You can completely change the layout. Each of these is described in more details below.

6. Your assignment must create the following **web APIs** with the specified routes and functionality (see Lab12b exercise 10, and Lab14a exercise 14). The returned data must be JSON format. **NOTE: you will NOT use these APIs in the rest of this assignment; you will use these API routes in the second assignment.**

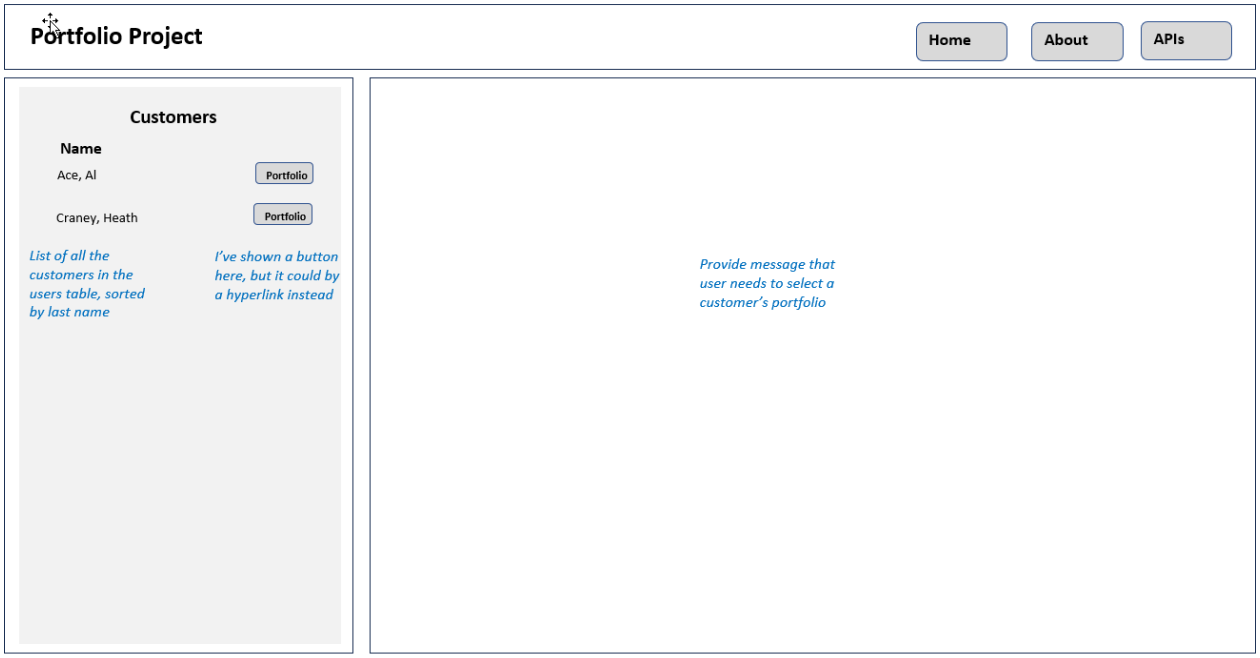
`/api/companies.php`	Returns all the companies
`/api/companies.php?ref=?`	Returns just the specified company (use the `symbol` field), e.g., `/api/companies.php?ref=ads`
`/api/portfolio.php?ref=?`	Returns the portfolio info for the specified user (use the `userId` field), e.g., `/api/portfolio.php?ref=8`
`/api/history.php?ref=?`	Returns the history information for the specified company (use the `symbol` field) sorted by ascending date, e.g., `/api/history?ref=ads`

7. You will provide a PHP page with sample links (described below) that demonstrate your API works properly. This page wouldn't be for a "real" user; they are for me only and don't require any fancy styling.

8. **Header**. The page title can be whatever you'd like. Add working links or buttons for the home page, about page, and the API tester page.

9. **Github Repo.** You must provide me with your source code on a github repo. If working in pairs, I would recommend that you each create an identical repo in your own github account.

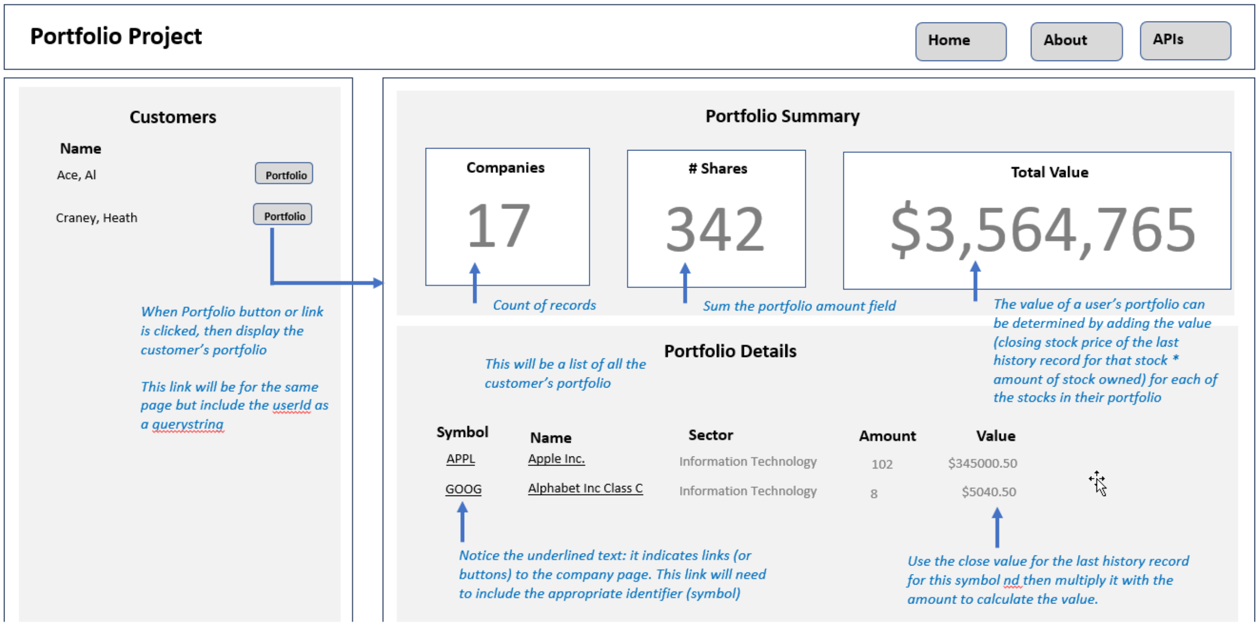
10. The first shown below is the **Home Page**. This will be what the user sees first. It's filename must be `index.php`. The prototype shown below is purposively simple and focuses on functionality. It's up to you to make it look good.



List of all the customers in the users table, sorted by last name

I've shown a button here, but it could by a hyperlink instead

Provide message that user needs to select a customer's portfolio



When Portfolio button or link is clicked, then display the customer's portfolio

This link will be for the same page but include the userId as a querystring

Count of records

Sum the portfolio amount field

*The value of a user's portfolio can be determined by adding the value (closing stock price of the last history record for that stock * amount of stock owned) for each of the stocks in their portfolio*

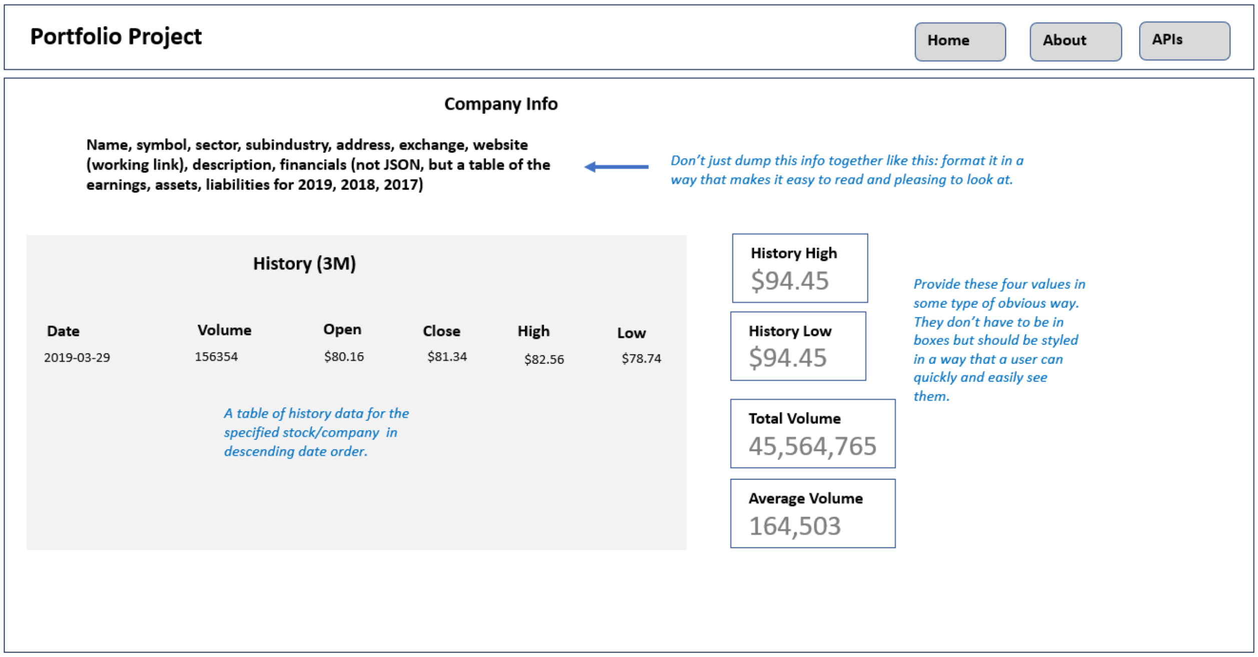
This will be a list of all the customer's portfolio

Symbol	Name	Sector	Amount	Value
APPL	Apple Inc.	Information Technology	102	$345000.50
GOOG	Alphabet Inc Class C	Information Technology	8	$5040.50

Notice the underlined text: it indicates links (or buttons) to the company page. This link will need to include the appropriate identifier (symbol)

Use the close value for the last history record for this symbol nd then multiply it with the amount to calculate the value.

11. The next shown is the **Company Page**. It includes details about the company as well as historical stock market results. How will the page “know” which company to display? It will be passed the `symbol` value for the company.



Portfolio Project

Company Info

Name, symbol, sector, subindustry, address, exchange, website (working link), description, financials (not JSON, but a table of the earnings, assets, liabilities for 2019, 2018, 2017)

Don’t just dump this info together like this: format it in a way that makes it easy to read and pleasing to look at.

History (3M)

Date	Volume	Open	Close	High	Low
2019-03-29	156354	$80.16	$81.34	$82.56	$78.74

A table of history data for the specified stock/company in descending date order.

History High
$94.45

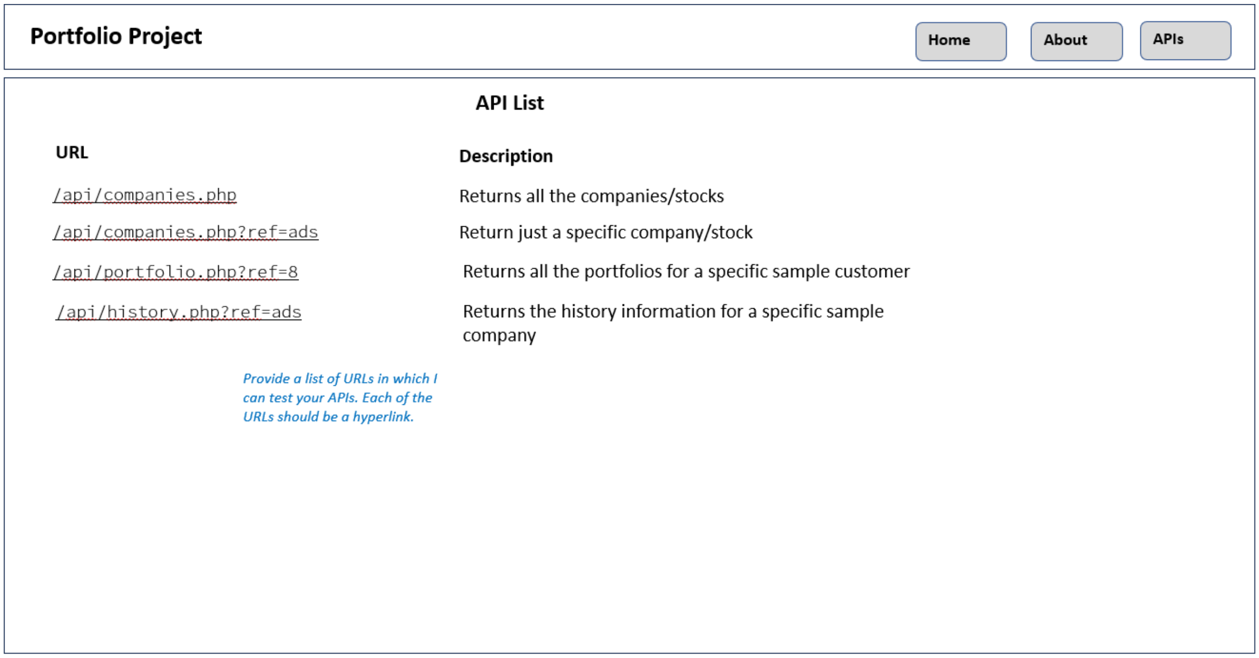
History Low
$94.45

Total Volume
45,564,765

Average Volume
164,503

Provide these four values in some type of obvious way. They don’t have to be in boxes but should be styled in a way that a user can quickly and easily see them.

12. The next page is the **API Tester Page**. This page should include sample hyperlinks for each of the routes described on page 5. When one of these links is clicked, the user will see the JSON data returned from the API.



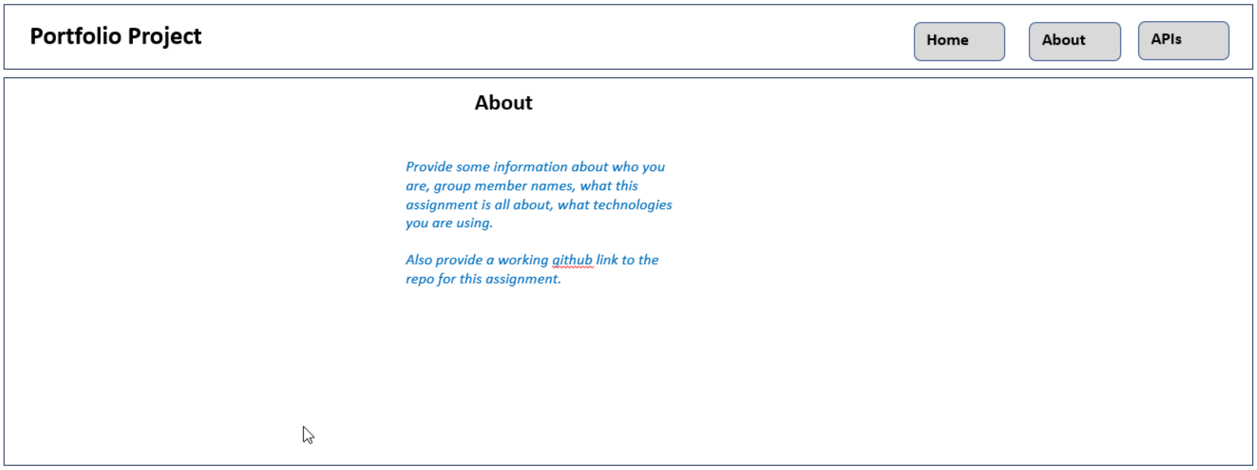
Portfolio Project

API List

URL	Description
/api/companies.php	Returns all the companies/stocks
/api/companies.php?ref=ads	Return just a specific company/stock
/api/portfolio.php?ref=8	Returns all the portfolios for a specific sample customer
/api/history.php?ref=ads	Returns the history information for a specific sample company

Provide a list of URLs in which I can test your APIs. Each of the URLs should be a hyperlink.

13. The next shown is the **About Page**. It includes details about your assignment. You need to provide a brief description (be sure to mention it is assignment #1 for COMP3512 at Mount Royal University), styled nicely, about what this site is about, what technologies you are using, the group member names, and the URL for the github repo. Notice also there are buttons or links that will appear on every PHP page.



Provide some information about who you are, group member names, what this assignment is all about, what technologies you are using.

Also provide a working github link to the repo for this assignment.

Hints

1. Begin by getting your data access from the database working. Complete Lab14a as soon as possible! You will find that this lab provides a library of functions that will simply your code. Most of Lab14a uses MySQL but Exercise 14a.8 shows you that you can use SQLite simply by changing the connection string.
2. I would recommend writing up your SQL for each of the APIs and save this in a separate text file, as you will be using these queries for your pages and for your APIs. You might find SQLiteStudio or the SQLite extension for Visual Code helpful for examining the database.

   I would recommend you complete the **Company Page** first. It will be passed the `symbol` as a query string parameter. Don't worry about styling at this point, just get the data display working. Test it with a sample `driverRef`, such as `aapl`.
3. Complete the **Home Page** next.
4. Begin implementing the different API pages. Look at Exercise 12.10 in Lab12b and Exercise 14a.14 in Lab14a to see how to create a page that returns JSON instead of HTML.
5. Finish off by implementing the **About Page**.
6. Most of the visual design mark will be determined by how much effort you took to make the pages look well designed. Simply throwing up the data with basic formatting will earn you minimal marks for this component. At this point, start improving the style of your existing pages.
7. If working in pairs, I would strongly recommend that each member is involved with retrieving and working with data. This will improve your midterm mark significantly.
8. Before submitting, test every single bit of functionality and cross-check with this document to make sure it has all the functionality I've requested. Carefully read the specifications for each bit of functionality. Every year, students lose all sorts of marks because they didn't read this document carefully enough!!