

COSC 349 Assignment 1

Name: Max Stewart

ID: 1086706

Project description:

My project is an online spending tracker that the user can track their spending and view this via a graph.

Build instructions:

To build my SpendTrack System you will need to download the GitHub repository files, these open a terminal window within the root of the repo and type 'vagrant up' followed by the return key. This will download and build the required VM's once this is complete you simply have to open a webpage to <http://127.0.0.1:10222/> and you will be taken to the front page. Even though there are two webservers running there are links within the webpages that will take you easily between the different web servers without needing to know the correct links.

NOTE: Some test data has been provided already but feel free to add some via the webserver pages as well to test the system out.

Download sizes

- Database server 30MB
- Webservers: 20MB each

In my project I have used three VM's these are as follows:

1. A web server that will host a page that will ask a user to input their spending data, this information will be stored in a database. This VM's job will be to write user data to a database and is one of the main client interaction points.
2. A database server that will store the information from the users, this information is the users spending info. This database will receive the information from webserver 1 and is needed as we want to store the data for long periods of time, and we wish for it to be available from another web server.
3. A webserver that will read the data from the database, this information is the spending data from the first web server that was saved in the database. This data will be read and converted into a graph for users to view. This is also one of the main client interaction points.

Since we have manually defined each VM's IP address we know where to look when we want to access them, this is seen in our webservers where we can add links between the VM's via the IP address and ports and this means the user can click on the link and be taken to the correct web server seamlessly. Our VM's communicate with one another via a private network this allows webserver to database access and vice versa and means we can take user input data and store this in the database described above.

Intended use:

I have designed my project to help track spending and to assist with budgeting. A user would input their spending data in via the first webserver. For example, they may go out for dinner and spend \$100, they would then input this purchase via the web server, and this would be stored in a database. When the user wants to view their spending, they would navigate to the second webserver and this data that has been stored will then be read and converted to a graph so another user can easily visualize their spending and figure out where they have spent too much or too little.

Useful modifications

- Having a system that could only accommodate one user is a big waste of resources. If I was hoping to make this system bigger and accessible by more users I would add a defined user system, by using the database and asking users to create a login we could expand the database and store information for multiple users at the same time. This change would require the "setup-db.sql" code to be modified, also the queries would need to be modified in the PHP pages as they would have to accommodate the users names, furthermore there would need to be a login/register user element added as either a new element on the existing pages or new pages altogether. Having a user table and having this relate to the different users and their own respective purchases would be important and this would require the developer to destroy the existing system with 'vagrant destroy' then after they had made the changes they would need to rebuild it using 'vagrant up'.
- With technology these days it could be possible and useful combine these three VM's to a single VM or container, as I am running two different web servers by simply combining these to one with different pages I would halve the resources required and the download size and time to bring the VM up. By also running the database server on the same box it would mean the unused capabilities on the one box would be put to better use.