**Caleb Ogbonnaya**

**Database II: Project Report**

**Dr Xia**

**Project Title: EOD “**Stockify” Page - Data Warehouse Project

**Project Description and Scope:** The goal of this project is to develop a data warehouse for analyzing the trend of financial market and maximizing users’ investments. The type of stocks I chose to first analyze was the US stock prices of the End of Day (EOD) data for a couple companies. Here is what the data warehousing schema looks like below (from my initial design \*\*this can be modified) for financial market analysis. Finally, I have implemented a web application of analytical models for forecasting information to help decision making of the users called “Stockify”.

**General Features and Functions will include:**

1. Average of High-Low and Open-Close
2. Show entries over date ranges with step of 1 day
3. Show entries by company or by multiple companies with search option
4. Show entries by year or by multiple years with search option
5. Show entries by quarter or by multiple quarters
6. Show entries by month or by multiple months with search option
7. Pagination of data using data-table
8. Search to show entries by specific keyword
9. Shows S&P 500 dataset

**Specific Requirements | functionalities [all] | Usability [installation, interface and manual]**

**Main Functionalities/Features**

* The system will allow users to visualize the End-of-Day (EOD) of USA Stocks Exchange of S&P 500
* The system will allow users to visualize current gold prices in the market in the form of chart
* The system will provide search fields and filters to extract the data according to given parameters.
* The system will allow users to apply customized search using filters. Multiple filters can be applied to fetch the records on interface
* The system will call APIs to get the real-time (live data) from the authentic sources e.g. Quandl.com
* The system will store the data into database
* The system will generate CSV file or files of stocks, gold price data for each refresh request
* Mobile Responsive - Easy to use on desktop and mobile
* Also contains the price, market capitalization, earnings, price/earnings ratio, price to book and etc; of all the S&P 500 companies.

**Specific Requirements**

**1.0 Project Plan**

* Understanding requirements and class design
* Creating project
* Making APIs calls to fetch the data from Quandl.com server
* Creating Data access layer in DB framework in Laravel. Database first approach will be used to create database schema
* Creating Business logic layer with PHP
* Creating Presentation logic layer in Laravel MVC
* Prediction of results
* Designing models, controllers and views
* jQuery Plugins to generates charts/graphs
  1. **Functional Requirements**
* Live data from APIs calls
* Presenting data using charts and data-table
* Customized search to fetch the records
* A single or multiple filters to fetch the records
* Generate CSV files to maintain data for each new request for data from API
* Charts/Graphs for visualizing the data
  1. **Database Requirements**

A database will need to be created and connected to the application through MySQL connector so that all data from the APIs can be stored for later use. It would have the following entities:

* Stocks

Contains stocks data of S&P 500

* GoldPrice

Contains data of different currency in which gold trades

* 1. **Platform Requirements**

To develop this application Laravel framework, has been chosen as it is deployable on cross platforms (Windows, Linux, MacOS). Applications developed on Laravel may also be deployed even on cloud giving it more flexibility. Visual Studio Code, MySQL, JavaScript and PHP, has been used in creating this application.

**System Designing**

In designing this system, the “**Project Plan**” was put in execution, and the following tools and technologies were utilized.

**Tools and Technologies**

Backend Technologies

* PHP as programming language
* Laravel as a MVC framework

Frontend Technologies - GUI

* HTML5
* CSS3
* Bootstrap 4
* JS/jQuery 3
* Datatable.js + jQuery DateRangeSlider.js (for choosing range of date)
* C3.js and D3.js (for charts/graphs)

Database

* MySQL

Development tools

* VS-Code
* XAMPP Server (for MySQL database)
* Composer and Laravel Server for Support Laravel MVC project

Data Source for APIs

* Quandl.com

**Discussion: Why is the solution better than others out there?**

Yes, agreeable - there are many solutions out there but this solution in particular is better than others because our system is fully responsive which means it can be used it on the desktop and mobile side by side. The system will provide a quick access to real-time stocks data at the end of day. This solution is more advantageous in comparison to other sites where one would have to keep an eye on interested stocks, this solution gives the users a full range to searches to perform which in-turn makes decision making easier. This web application will let users to visualize the stocks data, gold prices and performance-summary of all S&P 500 companies. All this are accessible just with click anywhere from anywhere.

JSON

Presenting

Processing

User Interface

Quandl.com

API

Stockify

Request

**Platform and System Setup:**

* Linux
* Windows

**Steps to setup:**

* Install composer

[**https://getcomposer.org/download/1.8.4/composer.phar**](https://getcomposer.org/download/1.8.4/composer.phar)

* Install Laravel

After installation of composer, on command prompt type: composer global require laravel/installer

* Install XAMPP

Start database – MySQL

Start Apache Server for phpMyAdmin

Visit Link on browser: localhost:8080/ and open phpMyAdmin

Create Database and Run SQL file for schema creation

* .env Setup

Open .env file in the zipped file and set the database name by default *stockset* is set

* Run Laravel Application

Open command prompt or Terminal and run following commands simultaneously:

composer update

php artisan serve

Visit link which will be prompted after executing *php artisan serve* command e.g. localhost:8000/ or 127.0.0.1/8000