**Project: Stock Overflow**

**Team Name: Bits Plz**

**Edvin Dizdarevic, James Griffin, Phillip Griggs, Asheton Harrell, and Jonathan Waring**

The main goal of StockOverflow is to serve as a research tool for stocks based on a specified price range. Additionally, it serves as a gateway for young investors to educate themselves about trading. We provide some basic information about trading stocks, as well as new and interesting ways to visualize and look at stock data for algorithmic trading. We have found that very few online platforms organize stocks by user-specified price ranges. This makes finding “penny stocks” for amateur or new investors difficult and discouraging. Finding credible information and tools that are free is incredibly difficult as well. StockOverflow aims to make these obstacles for new traders obsolete.

We have web scraped approximately one million tuples that contains information for stocks in the S&P 500 index, as well as the FTSE 100 index. The tuples contain historical pricing data over a 5-year span on the 595 different stocks in these indexes. We have information on each stock’s volume, opening price, closing price, high price, and low price for each day dating back as far as 2010 up until March 31, 2017. It has been well established that 5 years of data is needed for meaningful stock trading analysis, and we wanted to ensure that our end product would allow our users to get the data they needed. The scraping was accomplished through the use of the [Yahoo! Finance API](https://pypi.python.org/pypi/yahoo-finance) in the Python programming language.

In StockOverflow we queried the data in a new and interesting way when compared to many websites that host data on stocks. Our queries were also innovative when compared to online brokers and the way they display information. On StockOverflow, the queries are primarily written to aid in the research of companies and perform in-depth analysis on their historical data. One new contribution is a query that allows the user to filter stock data by a specified date and price range. Another query shows the top five best performing stocks for a specific date, based on their perecent growth for the day. We also have a query that shows the top ten stocks based on their closing price for a specific date in a visually appealing way. There is also a query that gives the average closing price, high, and low for a company for each year since 2013 to present day. The final query that we implemented, shows the top five companies for each sector based on their average closing price for the year of 2017. All of these queries are not found together on any one platform, and many of them could not be found at all.

For our implementation, we used php, with a hint of ajax, for our backend. For the two graphing buttons (graph.php and bargraph.php), we used the d3js API to generate the two graphs. Once an on-change event occurred, an ajax callback function would fill an empty div container with data from the specific php file pertaining to the graph. This php file would query the database, grab the values needed, echo the javascript for the d3js graph, then loop through the query data to place the data into the graph x and y values. In the other files, we used the same approach, but instead of echoing back the javascript code for the graph, we just placed the query data into html tables.

In the end, we feel that we have created a fun, interactive, innovative, and useful tool for anyone looking to learn about stock trading. We provide plenty of ways for users to query historical stock data in a way that would be useful for building algorithms for trading stocks. Not only that, but we have provided beginner information and resources for further reading for anyone looking to get started in investing. This is a fully functional site that has plenty of real-world applications, and we have met our goal of creating a large scale database application within a multi-functional web framework.