Mattia Danese
CS20 - Web Programming
Professor DiOrio
Assignment 14: Stock Ticker App

Questions

- I prefer Node.js over PHP because it is more popular and prevalent in the computer science industry. Not only is Node.js used more than PHP, there are also tons of additional modules that can be used with Node.js for added functionality or to make implementation easier and more efficient. Thus, I think there is just more you could do, or would want to do, with Node.js. Moreover, there are many web app frameworks, most notably React.js, where Javascript and Node.js is the backbone, so knowing/using Node.js would be beneficial if one would want to use a more distinguished framework. Lastly, I just prefer the syntax of Node.js (i.e. Javascript) over that of PHP.

Part One Video Link
https://drive.google.com/file/d/1SjgAzZj-O_FwdWN95gJq03P02S5AEqZj/view?usp=sharing

Part Two Online Link
https://stock-ticker-app-md.herokuapp.com

Part One

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Stock Ticker APP: Part One</title>
    <style>
        body {
            background-color: aliceblue;
            text-align: center;
        }
        input {
            font-size: large;
        }
    </style>
</head>
<body>
    <h1>Stock Ticker APP: Part One</h1>
    <br>
    <br>
    <h2>Upload a file below to insert data into the Stock Ticker
database</h2>
    <br>
    <form method="POST" action="/form_submit"
enctype="multipart/form-data">
        <input name="file" type="file" />
        <input type="submit" />
    </form>
</body>
</html>
```

error.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Stock Ticker APP: Part One</title>
    <style>
        body {
            background-color: aliceblue;
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>Stock Ticker APP: Part One</h1>
    <h2>Unknown Page Request</h2>
</body>
</html>
```

server.js

```
var http = require('http');
var fs = require('fs');
var formidable = require('formidable');
var readline = require('readline')
var pairs = [];

http.createServer(function (req, res) {
    if(req.url == "/"){
        let index="index.html";
        fs.readFile(index, function (err, txt) {
            res.writeHead(200, {'Content-Type': 'text/html'});
            res.write(txt);
            res.end();
        });
    }
    else if (req.url == "/form_submit"){
        if (req.method.toLowerCase() != "post"){
            let index = "index.html";
            fs.readFile(index, function (err, txt) {
                res.writeHead(200, {'Content-Type': 'text/html'});
                res.write(txt);
                res.end();
            });
        }
        else {
            const MongoClient = require("mongodb").MongoClient;
            const url =
"mongodb+srv://mattia:lunapark@cluster0.giy5w.mongodb.net/myFirstDatab
ase?retryWrites=true&w=majority";

            // extracting POST data
            var form = new formidable.IncomingForm();
            form.parse(req, function (err, fields, files) {
                if (err) {console.log(err);}

                // accessing file sent by user
                let path = files["file"]["filepath"];
                let newFile = readline.createInterface({
                    input: fs.createReadStream(path)
```

```
                });
                // extracting file contents
                newFile.on('line', function (line) {
                    // excludes column names
                    if (line != "Company,Ticker"){
                        let comp = line.split(',')[0];
                        let tick = line.split(',')[1];
                        pairs.push([comp, tick]);
                    }
                });
            });

            // connecting to mongo database
            MongoClient.connect(url, { useUnifiedTopology: true },
                async function (err, db) {
                    if (err) {console.log(err)}

                    let dbo = db.db("StockTickerApp");
                    let collection = dbo.collection("companies");

                    // Invariant: No two documents will have any of
the same company names or company tickers

                    for(i=0; i<pairs.length;i++){
                        let name = pairs[i][0];
                        let tick = pairs[i][1];

                        // avoids duplicate pairs
                        if(await collection.count({$and: [{names:
name}, {tickers: tick}]}) == 0){
                            if(await collection.count({names: name}) >
0){
                                // append ticker to existing document
                                await collection.updateOne({names:
name},
                                                {$push: {tickers:
tick}});
                            }
                            else if(await collection.count({tickers:
tick}) > 0){
```

```
                                // append company name to existing
document
                                await collection.updateOne({tickers:
tick},
                                                {$push: {names:
name}});
                            }
                            else {
                                // create new document for
company,ticker pair
                                await collection.insertOne({names:
[name], tickers: [tick]});
                            }
                        }
                    }
                    db.close();
                });

            // output displaying data was adding to database
            let index="index.html";
            fs.readFile(index, function(err, txt) {
                res.writeHead(200, {'Content-Type': 'text/html'});
                res.write(txt);
                res.write("<h4>Your data has been added to the
database!</h4>");
                res.end();
            });
        }
    }
    else {
        let error="error.html";
        fs.readFile(error, function(err, txt) {
            res.writeHead(200, {'Content-Type': 'text/html'});
            res.write(txt);
            res.end();
        });
    }
}).listen(8080);
```

Part Two

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Stock Ticker APP: Part Two</title>
    <style>
        body {
            background-color: aliceblue;
            text-align: center;
        }
        input {
            font-size: large;
        }
    </style>
</head>
<body>
    <h1>Stock Ticker APP: Part Two</h1>
    <br>
    <br>
    <h2>Search for a company either by name of stock ticker!</h2>
    <br>
    <form method="GET" action="/form_submit"
enctype="multipart/form-data">
        <input type="text" name="search" id="search" required/>
        <input type="radio" name="name_ticker" id="name" value="name"
required/>
        <label for="name">Company Name</label>
        <input type="radio" name="name_ticker" id="ticker"
value="ticker" required/>
        <label for="ticker">Ticker</label>
        <input type="submit" />
    </form>
</body>
</html>
```

error.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Stock Ticker APP: Part Two</title>
    <style>
        body {
            background-color: aliceblue;
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>Stock Ticker APP: Part Two</h1>
    <h2>Unknown Page Request</h2>
</body>
</html>
```

server.js

```
var http = require('http');
var fs = require('fs');

http.createServer(async function (req, res) {
    if(req.url == "/"){
        let index = "index.html";
        fs.readFile(index, function (err, txt) {
            res.writeHead(200, {'Content-Type': 'text/html'});
            res.write(txt);
            res.end();
        });
    }
    else if (req.url.substring(0,12) == "/form_submit"){
        if (req.url.indexOf('?') == -1){
            // console.log("here", req);
            let index = "index.html";
            fs.readFile(index, function (err, txt) {
                res.writeHead(200, {'Content-Type': 'text/html'});
                res.write(txt);
                res.end();
            });
        }
        else {
            const MongoClient = require("mongodb").MongoClient;
            const url =
"mongodb+srv://mattia:lunapark@cluster0.giy5w.mongodb.net/myFirstDatab
ase?retryWrites=true&w=majority";

            let data = req.url.split('?')[1].split('&');
            let search = data[0].split('=')[1];
            let n_or_t = data[1].split('=')[1];
            let result = "", content = "";

            // connecting to mongo database
            MongoClient.connect(url, { useUnifiedTopology: true },
                async function (err, db) {
                    if (err) {console.log(err)}

                    let dbo = db.db("StockTickerApp");
```

```
                    let collection = dbo.collection("companies");

                    // Invariant: No two documents will have any of
the same company names or company tickers

                    if(n_or_t == "name"){
                        // search for company by name
                        result = await collection.find({names:
search}).toArray();
                        if(result.length == 0){
                            content = `<h3>No information on company
'${search}' could be found</h3>`;
                        }
                        else {
                            content = `<h3>Company Name: ${search}
<br> Ticker(s): `;
                            for(i=0; i < result[0]["tickers"].length;
i++){
                                content += result[0]["tickers"][i];

                                if (i != result[0]["tickers"].length -
1)
                                    content += ", ";
                            }
                        }
                    }
                    else {
                        result = await collection.find({tickers:
search}).toArray();
                        if(result.length == 0){
                            content = `<h3>No information on company
with ticker '${search}' could be found</h3>`;
                        }
                        else {
                            content = "<h3>Company Name(s): ";
                            for(i=0; i < result[0]["names"].length;
i++){
                                content += result[0]["names"][i];
```

```
                                if (i != result[0]["names"].length -
1)
                                    content += ", ";
                            }

                            content += `<br> Ticker: ${search}</h3>`;
                        }
                    }
                    db.close();

                    // output displaying company data
                    let index="index.html";
                    fs.readFile(index, function(err, txt) {
                        res.writeHead(200, {'Content-Type':
'text/html'});
                        res.write(txt);
                        res.write(content);
                        res.end();
                    });
                });
        }
    }
    else {
        let error="error.html";
        fs.readFile(error, function(err, txt) {
            res.writeHead(200, {'Content-Type': 'text/html'});
            res.write(txt);
            res.end();
        });
    }
}).listen(process.env.PORT || 8080);
```