# Developmental Testing

# Customer Relationship Management System (CRM)

**Kieran Williams**

## Some of the problems I encountered during the process

**This document outlines and provides evidence for syntactical, semantic and logical errors I encountered throughout the development process.**

**I have also given explanations of the problems and how I resolved the issues, providing proof that it fixed the error where appropriate.**

**Each problem will be separated by a line of '='**

# Experimentation And Prototype Development

When trying to call the comparisonOperator private static method from the bubble sort method I got an error stating that the method did not exist

```
    while(!sorted){
        sorted = true;
        for(let i = 0; i < n; i++){
            // if(array[i] > array[i+1]) is for ascending
            // uses this method so the method is dynamic in both
            // ascending and descending
            if (this[_comparisonOperator](comparisonOp, array[i], array[i+1])) {
                test = array[i];
                array[i] = array[i+1];
                array[i+1] = test;
                sorted = false;
            }
        }
    }
```

```
/Users/kieranwilliams/WebstormProjects/summerProject/classes/Sort.js:30
            if (this[_comparisonOperator](comparisonOp, array[i], array[i+1])) {
                                          ^

TypeError: this[_comparisonOperator] is not a function
    at Function.Sort.bubbleSort (/Users/kieranwilliams/WebstormProjects/summerProject/classes/Sort.js:30:46)
    at Object.<anonymous> (/Users/kieranwilliams/WebstormProjects/summerProject/classes/Sort.js:70:24)
    at Module._compile (module.js:652:30)
    at Object.Module._extensions..js (module.js:663:10)
    at Module.load (module.js:565:32)
    at tryModuleLoad (module.js:505:12)
    at Function.Module._load (module.js:497:3)
    at Function.Module.runMain (module.js:693:10)
    at startup (bootstrap_node.js:191:16)
    at bootstrap_node.js:612:3
```

```
if (Sort[_comparisonOperator](comparisonOp, array[i], array[i+1])) {
```

the value of keyword 'this' was not Sort. I had to call the static method class explicitly with ‘Sort’ instead of ‘this’ I realised that the keyword this in that instance will be the window object
And that I would have to be explicit in the future with any instances where I am creating a static Method without ECMAScript 6 syntax

==========

This code was surrounded by a for loop and the problem was that the array2D[i+1] in the final iteration did not exist

```
// uses the comparisonOperator method to make this
// method dynamic in sorting both ascending and descending
if (Sort[_comparisonOperator](comparisonOp,
    array2D[i][columnToSortBy], array2D[i+1][columnToSortBy])) {
    test = array2D[i];
    array2D[i] = array2D[i+1];
    array2D[i+1] = test;
    sorted = false;
}
```

```
/Users/kieranwilliams/WebstormProjects/summerProject/classes/Sort.js:65
              array2D[i][columnToSortBy], array2D[i+1][columnToSortBy])) {
                                                                      ^

TypeError: Cannot read property '2' of undefined
    at Function.Sort.bubbleSort2D (/Users/kieranwilliams/WebstormProjects/summerProject/classes/Sort.js:65:57)
    at Object.<anonymous> (/Users/kieranwilliams/WebstormProjects/summerProject/classes/Sort.js:86:24)
    at Module._compile (module.js:652:30)
    at Object.Module._extensions..js (module.js:663:10)
    at Module.load (module.js:565:32)
    at tryModuleLoad (module.js:505:12)
    at Function.Module._load (module.js:497:3)
    at Function.Module.runMain (module.js:693:10)
    at startup (bootstrap_node.js:191:16)
    at bootstrap_node.js:612:3
```

```
while(!sorted){
    sorted = true;
    for(let i = 0; i < n -1 ; i++){
```

There was one too many iterations in loop so I subtracted one from the terminating for loop condition. From this I learned that in the future I should account for the bounds when creating a loop that manipulates data as it is possible to generate an out of bounds error

==========

The compiler could not find the Sort class when I tried importing it into another file

```
const Sort = require("Sort");

let arr = [
```

```
module.js:549
    throw err;
    ^

Error: Cannot find module 'Sort'
    at Function.Module._resolveFilename (module.js:547:15)
    at Function.Module._load (module.js:474:25)
    at Module.require (module.js:596:17)
    at require (internal/module.js:11:18)
    at Object.<anonymous> (/Users/kieranwilliams/WebstormProjects/summerProject/classes/Search.js:82:14)
    at Module._compile (module.js:652:30)
    at Object.Module._extensions..js (module.js:663:10)
    at Module.load (module.js:565:32)
    at tryModuleLoad (module.js:505:12)
    at Function.Module._load (module.js:497:3)
```

```
const Sort = require("./Sort");
```

./' is needed before the filename when the file is in the same directory. From this point on I knew how to import classes from one file into another. To help circumvent this problem in the future, I also installed a plug in for the ide that would include ide features for the not natively supported Node js syntax

==========

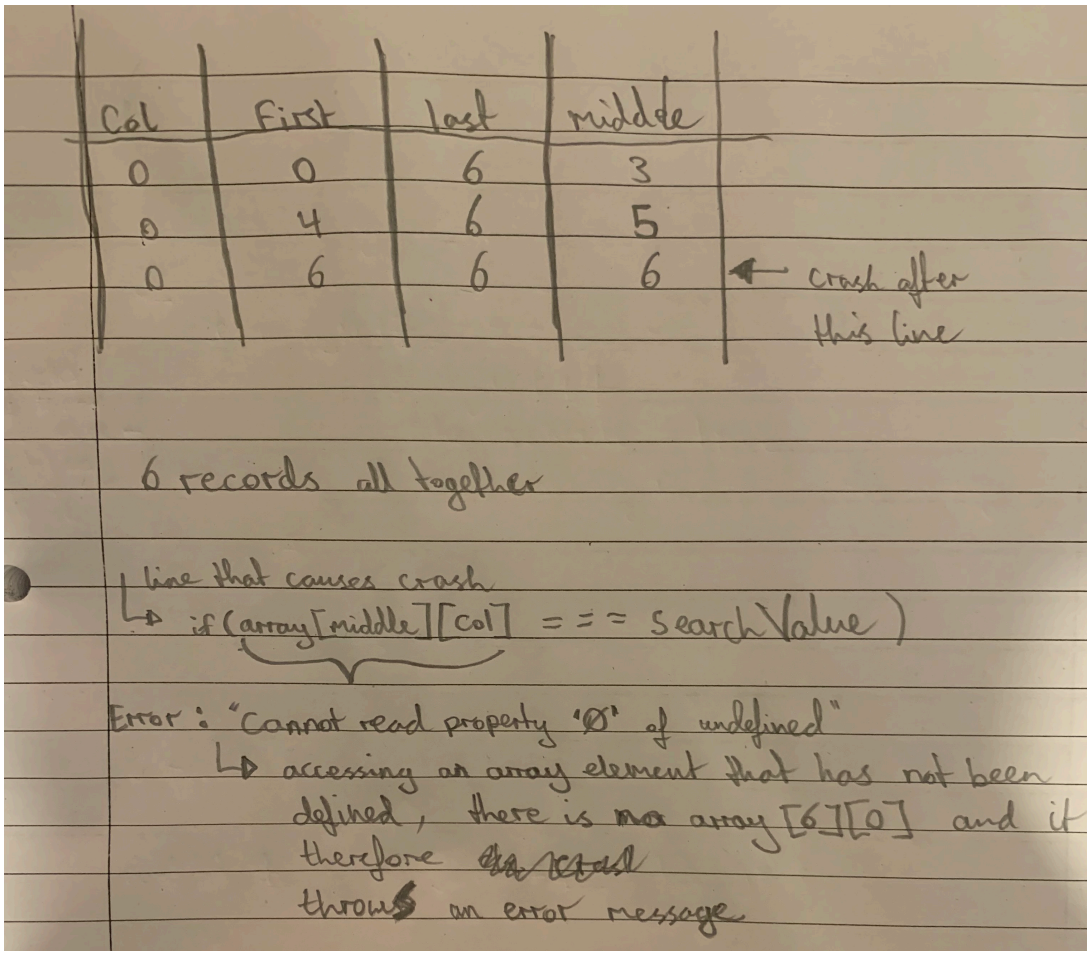
while trying to run the selectRecordByID method in the CSVHandler Class with an id that is not in the file, this error became apparent so set variable watches and drew up a trace table

```
80
81  let getRecordByIDTest = usersRecords.selectRecordByID("110");
82  console.log(getRecordByIDTest);
83
84  // Sort.bubbleSortAscending();
85  // CsvAndTxtHandler[_csvToArray]();
```

CSVHandler.js

```
/usr/local/bin/node /Users/kieranwilliams/WebstormProjects/summerProject/classes/CSVHandler.js
/Users/kieranwilliams/WebstormProjects/summerProject/classes/Search.js:112
        if(array[middle][col] === searchValue){
                         ^

TypeError: Cannot read property '0' of undefined
    at Function.Sort.(anonymous function) (/Users/kieranwilliams/WebstormProjects/summerProject/classes/Search.js:112:25)
    at Function.Search.binarySearch2D (/Users/kieranwilliams/WebstormProjects/summerProject/classes/Search.js:164:62)
    at CsvAndTxtHandler.selectRecordByID (/Users/kieranwilliams/WebstormProjects/summerProject/classes/CSVHandler.js:57:38)
    at Object.<anonymous> (/Users/kieranwilliams/WebstormProjects/summerProject/classes/CSVHandler.js:81:38)
    at Module._compile (module.js:652:30)
    at Object.Module._extensions..js (module.js:663:10)
    at Module.load (module.js:565:32)
    at tryModuleLoad (module.js:505:12)
    at Function.Module._load (module.js:497:3)
    at Function.Module.runMain (module.js:693:10)
```

4: Run  6: TODO  Terminal




```
// loop that will keep going until last > first or the matching records are found
// the array[middle][col] ---> middle is the row that is currently under inspection, col is a constant
while (first <= last) {

    // to avoid a crash, the program will check to see if middle is greater than the index of the last
    // record and that if it is a negative value. In either instance the loop will be terminated
    if (middle > array.length - 1 || middle < 0){
        break;
    }

    if(array[middle][col] === searchValue){
        recordsFound.push(array[middle]); // adds the record into records found array

        // finds any matches in the upper and lower bounds of the
        upperAndLowerBoundMatches = Search[_checkBoundsForMatches](array, middle, col, searchValue);

        upperAndLowerBoundMatches.forEach((match) => {
            recordsFound.push(match);
        });
        break; // gets out of loop
    }
    if(searchValue < array[middle][col]){
        last = middle - 1;
        middle = Math.floor((first + last)/2);
    }
    if(searchValue > array[middle][col]){
        first = middle + 1;
        middle = Math.floor((first + last)/2);
    }
}

if (recordsFound.length > 0){
    return recordsFound;
}
```

I added in a conditional to test for the case where the middle value becomes greater than the index of the last record and if it evaluates to a negative value it will terminate the loop

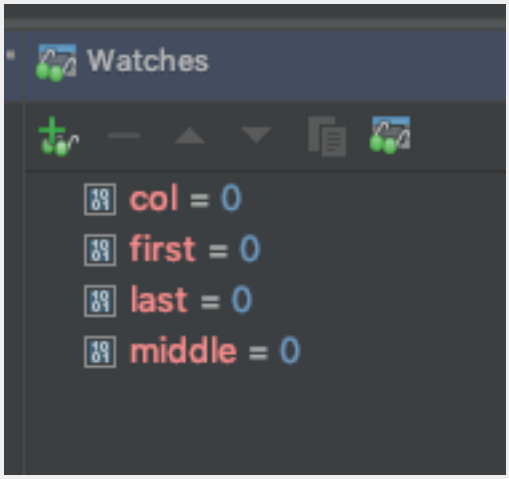
```
// to avoid a crash, the program will check to see if middle is greater than the index of the last
// record and that if it is a negative value. In either instance the loop will be terminated
if (middle > array.length - 1 || middle < 0){
    break;
}
```

A similar problem occurred when I used the id "0"

```
let usersRecords = new CsvAndTxtHandler("blah.csv");

let arrayOfRecords = usersRecords[_csvToArray]();

let getRecordByIDTest = usersRecords.selectRecordByID("0");
console.log(getRecordByIDTest);

// Sort.bubbleSortAscending();
// CsvAndTxtHandler[_csvToArray]();
```

```
CSVHandler.js
/usr/local/bin/node /Users/kieranwilliams/WebstormProjects/summerProject/classes/CSVHandler.js
/Users/kieranwilliams/WebstormProjects/summerProject/classes/Search.js:133
        if(searchValue > array[middle][col]){
                         ^

TypeError: Cannot read property '0' of undefined
    at Function.Search.(anonymous function) (/Users/kieranwilliams/WebstormProjects/summerProject/classes/Search.js:133:39)
    at Function.Search.binarySearch2D (/Users/kieranwilliams/WebstormProjects/summerProject/classes/Search.js:170:64)
    at CsvAndTxtHandler.selectRecordByID (/Users/kieranwilliams/WebstormProjects/summerProject/classes/CSVHandler.js:61:38)
    at Object.<anonymous> (/Users/kieranwilliams/WebstormProjects/summerProject/classes/CSVHandler.js:85:38)
    at Module._compile (module.js:652:30)
    at Object.Module._extensions..js (module.js:663:10)
    at Module.load (module.js:565:32)
    at tryModuleLoad (module.js:505:12)
    at Function.Module._load (module.js:497:3)
    at Function.Module.runMain (module.js:693:10)
```

So to get a better understanding of what was going on I set a variable watch to these variables



The problem was that middle would turn into a -1 value and crash the program as there is no way an array could have an index of -1 when the program was to subtract -1 from the middle and look for a value with that middle value in an array index

by checking after middle === search value evaluation, that the middle value is = 0 we can then terminate the loop to avoid crash

```
// terminates the loop if the middle value is 0 as it will cause a crash
// crash is due to the fact that middle will turn into a negative value
if (middle === 0) {
    break;
}

if(searchValue < array[middle][col]){
    last = middle - 1;
    middle = Math.floor((first + last)/2);
}
if(searchValue > array[middle][col]){
    first = middle + 1;
    middle = Math.floor((first + last)/2);
}
```

==========

When testing the rewrite method that clears a file of data, and then writes data from a 2d array to a csv file. I noticed that the data is getting re written to the file but not in the order I expected it to

```
let usersRecords = new CSVHandler("blah.csv");

// let arrayOfRecords = usersRecords[_csvToArray]();

let array = [
    ["101","kieran","williams"],
    ["102","jamie","lewis"],
    ["103","sally","alex"],
    ["104","rhodri","bevan"],
    ["105","callum","james"],
    ["106","kieran","james"],
    ["107","bob","james"],
    ["108","ryan","james"],
];

usersRecords[_rewriteDataToFileWithChanges](array);

// let getRecordByIDTest = usersRecords.selectRecordByField([0,1], "101");
// let recordMatches = Search.getRecords(getRecordByIDTest);
// console.log(recordMatches);
```

```
CSVHandler.js
/usr/local/bin/node /Users/kieranwilliams/WebstormProjects/summerProject/classes/CSVHandler.js
data has been deleted from the blah.csv
data rewritten successfully
data rewritten successfully
data rewritten successfully
data rewritten successfully
data rewritten successfully
data rewritten successfully
data rewritten successfully
data rewritten successfully
```

The data written into the csv file is shown below. I thought the records would be in order but from looking at the ids it is clear that they were being added to the text file in a random order

```
101,kieran,williams,
104,rhodri,bevan,
103,sally,alex,
105,callum,james,
102,jamie,lewis,
106,kieran,james,
107,bob,james,
108,ryan,james,
```

From looking online I realised the problem is that some data is being written to the file before others due to the asynchronous nature of javascript. It is a way of the single threaded language to be able do other process while the previous process is being carried out. To fix this I needed to tell the language to wait until the process is finished before moving on to the next task

```
fs.appendFileSync(this[_csvToTxt](), `${record},\n`);
```

I changed the append asynchronous method to a synchronous method , this allowed me to be able to tell the language to do one process at a time. This taught me to use synchronous methods when the option is there unless there is a reason for the async feature


File writes in the correct order as shown above

==========

## Full Project Software Development

On the  client side javascript I was trying to get the toggle password to text in the password fields to work.

```
function showPassword() {
    console.log("running show password method");
    let passwordField = document.querySelector("#password");
    let confirmPasswordField = document.querySelector("#confirm-password");

    if(passwordField.type === "password" && confirmPasswordField === "password"){
        passwordField.type = "text";
        confirmPasswordField.type = "text";
    }
    else {
        passwordField.type = "password";
        confirmPasswordField.type = "password";
    }
}
```

function to toggle the password field's type, was not working because in one of the comparisons I forgot the .type property on the variable holding the confirm password text field in the comparison. To prevent making mistakes like this again I tried to make a conscious effort to pay more attention to what I was typing to avoid silly logic errors like this

==========

Here I was meant to check that the length of a value was equal to a specified length but it would not run

```
Validation.length = (value, length) => {
  return value.length === length;
};
```

```
if(passwordField.type === "password" && confirmPasswordField.type !== "password"){
```

```
Validation.length("ki",2);
           ^

TypeError: Validation.length is not a function
    at Object.<anonymous> (/Users/kieranwilliams/Desktop/CS5
    at Module._compile (module.js:652:30)
    at Object.Module._extensions..js (module.js:663:10)
    at Module.load (module.js:565:32)
    at tryModuleLoad (module.js:505:12)
    at Function.Module._load (module.js:497:3)
    at Function.Module.runMain (module.js:693:10)
    at startup (bootstrap_node.js:191:16)
    at bootstrap_node.js:612:3
```

The problem seems to be that js already has a .length global property and so it is calling that to be run on the Validation static class. By changing the name of the method to 'lengthCheck' it worked. From this I learned that when using properties of an object to attach functionality to an object to be careful that there isn't already a property assigned by the language with that name as it can result in the code not running as expected. To ensure this doesn't happen again I could look online what properties javascript automatically assigns to every object

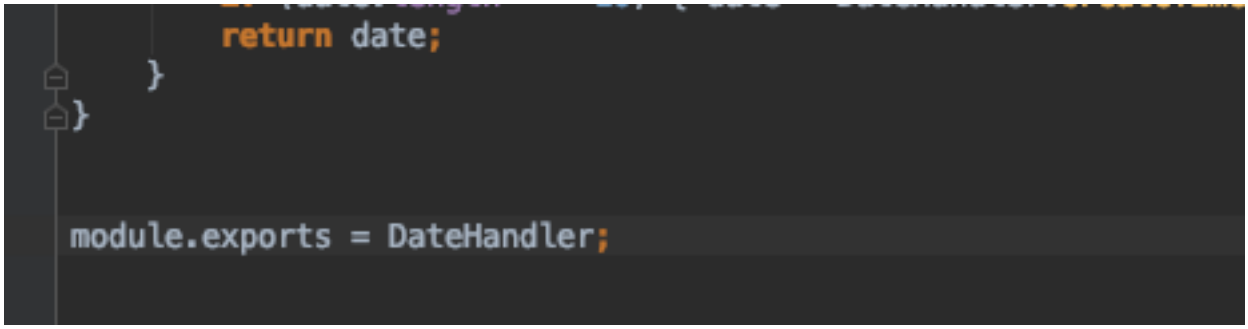
==========

While trying to call the getCurrentDateTime method, an error was getting thrown that it is not a method

```
const DateHandler = require('./DateHandler');

class GenerateDataPoints{
    static getDataPointsFromLast7Days (usersId, numOfWeeks, databaeFieldToQuery) {
        if ( numOfWeeks === undefined) numOfWeeks = 1;
        let dataPointsArray = [];
        let date = DateHandler.getCurrentDateTime();

        for (let i = numOfWeeks * 7; i > 0; i-- ){
            date = DateHandler.addDaysToDate(date, -i);
            console.log(date);
        }
    }
}


// GenerateDataPoints.getDataPointsFromLast7Days();
let date = GenerateDataPoints.getDataPointsFromLast7Days();
console.log(date);
```

```
        let date = DateHandler.getCurrentDateTime();
                               ^

TypeError: DateHandler.getCurrentDateTime is not a function
    at Function.getDataPointsFromLast7Days (/Users/kieranwilliams/Desktop/CS5
    at Object.<anonymous> (/Users/kieranwilliams/Desktop/CS5 Coursework/prograi
    at Module._compile (module.js:652:30)
    at Object.Module._extensions..js (module.js:663:10)
    at Module.load (module.js:565:32)
    at tryModuleLoad (module.js:505:12)
    at Function.Module._load (module.js:497:3)
    at Function.Module.runMain (module.js:693:10)
    at startup (bootstrap_node.js:191:16)
    at bootstrap_node.js:612:3

Process finished with exit code 1
```

the problem was that I had not exported the DateHandler Class so it could not be imported in the require statement at the top




Fixed by exporting the date handler class at the bottom of its file. To ensure this sort of error did not occur again I decided to import a plugin that would warn me whenever there was a require statement to which what was being required did not have an exports statement. This helped to prevent this error happening again

==========

```
class GenerateDataPoints{
    static getDataPointsFromLast7Days (usersId, numOfWeeks, databaeFieldToQuery) {
        if ( numOfWeeks === undefined) numOfWeeks = 1;
        let dataPointsArray = [];
        let date = DateHandler.getCurrentDateTime();
        let noDays = numOfWeeks *7;
        console.log(`current date: ${date}`);
        for (let i = noDays; i > 0; i-- ){
            date = DateHandler.subtractFromDate(date, {days: i});
            console.log(date);
        }
    }
}

GenerateDataPoints.getDataPointsFromLast7Days();
```

```
/usr/local/bin/node "/Users/kier
current date: 2019-02-03T13:35:0
2019-01-27T13:35:00+00:00
2019-01-21T13:35:00+00:00
2019-01-16T13:35:00+00:00
2019-01-12T13:35:00+00:00
2019-01-09T13:35:00+00:00
2019-01-07T13:35:00+00:00
2019-01-06T13:35:00+00:00
```

the outputs should be in order with the date increasing as the for loop progresses but it is decreasing the current date by 7, then decreases that date by 6, then decreases that date by 5 and so on until it decreases the date by 1 then the loop stops

the problem is that I'm subtracting from the same date and then storing that new date, then subtracting from it again in a loop. what I need to happen is for it to go back to the original date and then subtract a the i value from that repeatedly instead

```
const DATE = DateHandler.getCurrentDateTime();
let dateToBeChanged = DATE;
let noDays = numOfWeeks *7;
console.log(`current date: ${DATE}`);
for (let i = noDays; i > 0; i-- ){
    dateToBeChanged = DateHandler.subtractFromDate(DATE, {days: i});
    console.log(dateToBeChanged);
}
```

created a variable to be changed and a variable to hold the constant value and this resolve the issue

==========

time validator, should be returning false because the minutes were 80 and the max should be 59 but was returning true

```
        hour = valueArray[0] + valueArray[1];
        minute = valueArray[2] + valueArray[3];

        return ((Number(hour) >= 0 || Number(hour) <= 23 || Number(minute) >= 0 || Number(minute) <= 59));
    }
};


module.exports = Validation;

console.log(Validation.timeFormat_HHMM("20:80"));
```

the problem is that I'm using an OR logic gate meaning as long as one of the comparisons returns true, the return value will be true

changed the OR gate to an AND gate so that all conditions must be true for true to be returned. From this I learned the differences between AND and OR bitwise operators

==========

```
console.log(Validation.stringLookupCheck("KIERaN", {lowerCase: true}));
```

should return true as there is a lower case character

upon stepping through the code with the debugger I noticed the code never went through the loop because the comparison part of the for loop was comparing the increment against an array not the array length

```
for (let i = 0; i < stringArray; i++) {
    currentCharecter = stringArray[i];
    for (let j = 0; j < lowerCaseArray; j++) {
```

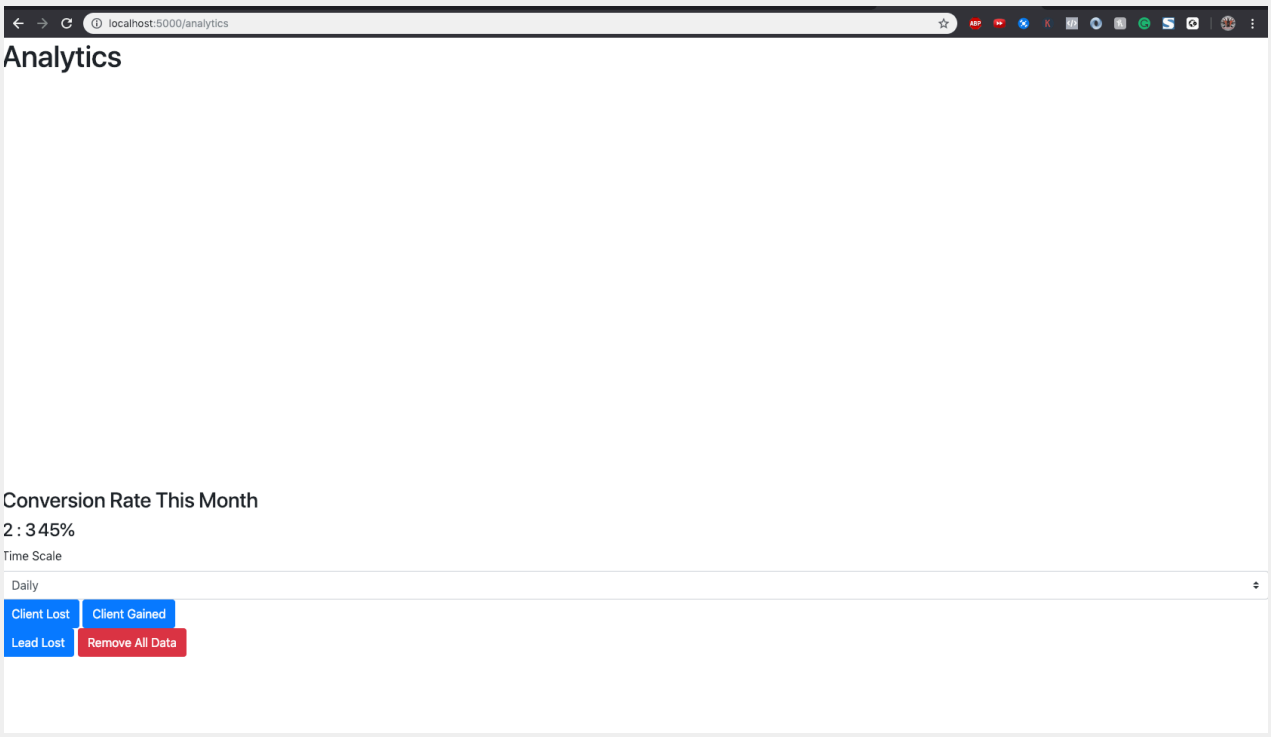
Changed lowerCaseArray to lowerCaseArray.length

```
for (let i = 0; i < stringArray.length; i++) {
    currentCharacter = stringArray[i];
    for (let j = 0; j < lowerCaseArray.length; j++) {
```

This resolved the issue as I needed to iterate through the array with a number value not an array as this would cause the loop to iterate only once. From this I learned to be more careful when it comes to writing syntax and not to rush as it is a lot easier to find an error if you proof read the code as you are going along instead of typing the entire thing as fast as possible and then trying to find the reason it wont run using the debugger.

==========

I turned the html file into a templating language ejs to work with the http framework express. refactored all the files perfectly but my ejs document was not applying my css I had written no was it applying my client side js

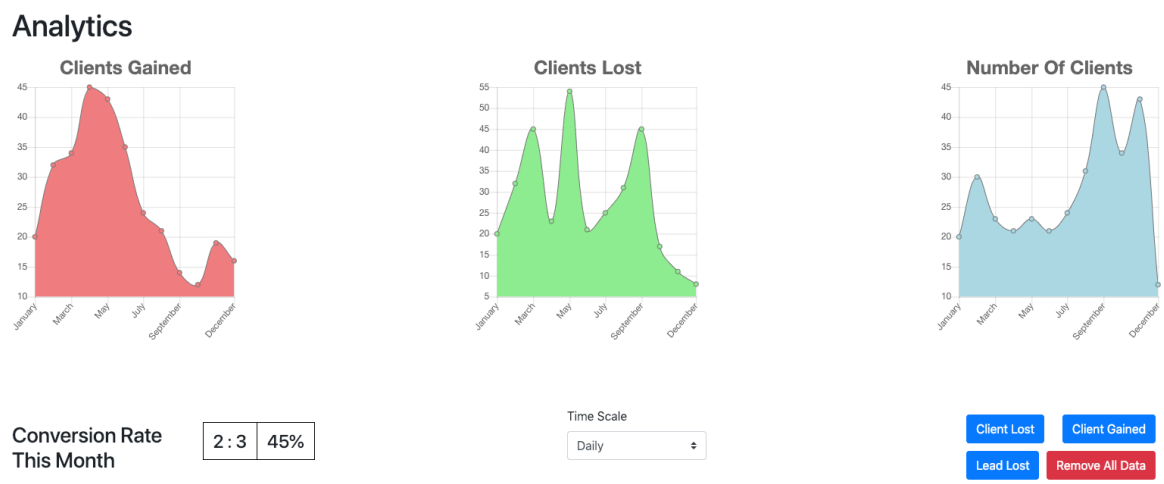



I forgot I had used this line of code in the index file which sets the root of all static files to the public folder

```
app.use(express.static(__dirname + "/public"));
```

so I changed the file path from to this which goes back out of one directory first with the ../

```
<link rel="stylesheet" type="text/css" href="../public/stylesheets/AnalyticsStyles.css">
```




And then it worked

==========

I tried to add a record to the csv file containing contact records

And got this error

```
//renders the create contact page
router.post("/", (req, res) => {
    let businessName = req.body.businessName;
    let firstName = req.body.firstName;
    let lastName = req.body.lastName;
    let email = req.body.email;
    let telephone = req.body.telephone;
    let city = req.body.city;
    let postcode = req.body.postcode;
    let address = req.body.address;
    let status = req.body.status;
    contactsFile.addRecord([businessName, firstName, lastName, email, telephone, city, postcode, address, status]);


});
```

```
server started on port 5000
TypeError: Path must be a string. Received undefined
    at assertPath (path.js:28:11)
    at Object.basename (path.js:1380:5)
    at CSVHandler.[csvToTxt] (/Users/kieranwilliams/Desktop/experimental coursework/programming/kwcrmCourseworkProject/classes/CSVHandler.js:38:21)
    at CSVHandler.[txtInCsvFormatToArray] (/Users/kieranwilliams/Desktop/experimental coursework/programming/kwcrmCourseworkProject/classes/CSVHandler.js:47:51)
    at CSVHandler.addRecord (/Users/kieranwilliams/Desktop/experimental coursework/programming/kwcrmCourseworkProject/classes/CSVHandler.js:86:47)
    at router.post (/Users/kieranwilliams/Desktop/experimental coursework/programming/kwcrmCourseworkProject/routes/contactsRoutes.js:38:18)
    at Layer.handle [as handle_request] (/Users/kieranwilliams/Desktop/experimental coursework/programming/kwcrmCourseworkProject/node_modules/express/lib/router/layer.js:95:5)
    at next (/Users/kieranwilliams/Desktop/experimental coursework/programming/kwcrmCourseworkProject/node_modules/express/lib/router/route.js:137:13)
    at Route.dispatch (/Users/kieranwilliams/Desktop/experimental coursework/programming/kwcrmCourseworkProject/node_modules/express/lib/router/route.js:112:3)
    at Layer.handle [as handle_request] (/Users/kieranwilliams/Desktop/experimental coursework/programming/kwcrmCourseworkProject/node_modules/express/lib/router/layer.js:95:5)
```

the problem was that I hadn't set the constructor with the file path

```
const CSVHandler = require("../classes/CSVHandler");
let contactsFile = new CSVHandler();
```

fixed this by passing the file path of the contacts csv file to the constructor method of the contactsFile

==========

This code was meant to update a record but instead started to delete fields and data within fields

```
let contactsFile = new CSVHandler(__dirname + '/dataStoreFiles/contacts.csv');
let record = req.body;
let recordArray = Object.values(record); // turns the record object into an array

console.log(recordArray);
let arrayOfEdits = [];

// populate an array of changes to be made
for (let i = 2; i < recordArray.length; i++) {
    let recordChange = [i.toString(), recordArray[i]];
    arrayOfEdits.push(recordChange);
}

res.send(arrayOfEdits);
```

This is the original file without the id that was meant to be updated

```
[ 'Dans Tree Services',
  'dan',
  'hughes',
  'dannyboi@gmail.com',
  '017927836654',
  'neath',
  'sa94bw',
  '89 lane street',
  'lead' ]
```

As you can see lots of data is randomly missing


the problem is that I am going from index 2 of the record because I thought I had to account for the id and time stamp but they aren't even in the array because the array is only composed from what comes back from the edit form

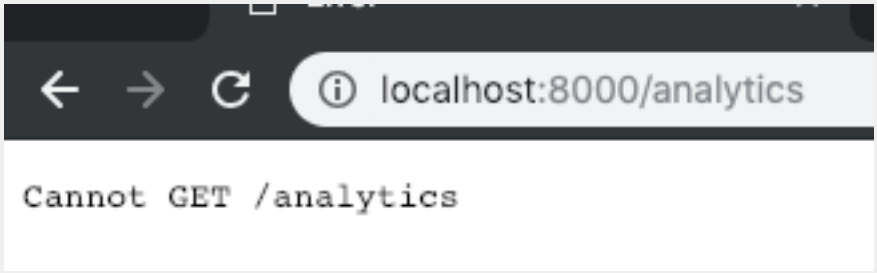
what I should have done is start the index value that increments to 0 and then only +2 on the value that represents the column needing changing

By making this change it worked

```
// populate an array of changes to be made
for (let i = 0; i < recordArray.length; i++) {
    let recordChange = [(i+2).toString(), recordArray[i]];
    arrayOfEdits.push(recordChange);
}
```

==========

could not get to the /analytics get route.




The route existed but could not be found

```
router.get("/analytics", (req, res) => {
    res.render("Analytics");
});
```

I was also requiring the route in the index file too so I couldn't see why it would not be able to find the route

```
    analyticsRoutes = require("./routes/analyticsRoutes"),
```

the problem was that, I had already set the path to /analytics in the index entry point js file. And I was then putting /analytics again in the analyticsRoutes page

this was resulting in the route being /analytics/analytics to resolve this I made the route in the analytics page just /

```
app.use("/analytics", analyticsRoutes);
```

Changed the base route for the analytics routes to just / as the / gets seen by the index file as /analytics/

```
////////////////////////////////////////////////////////////////////

//                                    GET ROUTES

////////////////////////////////////////////////////////////////////

router.get("/", (req, res) => {
    res.render("Analytics");
});
```

This resolved the problem and it made me decide to read the express framework documentation so that I wouldn't run into any other similar problems in the future.

==========

I experience a logical error in that binary strings are different lengths as some bytes are 8 bits and other bytes don't show the last bit as it is a 0 which messes up the lengths. Need them to be the same length to get them to be able to XOR all the bits and produce accurate xor values that can be used to get the original text back

```
console.log(`type of input Bin: ${typeof inputBin}`);
console.log(`input Bin: ${inputBin}`);
```

```
Encryption.js
/usr/local/bin/node /Users/kieranwilliams/WebstormProjects/summerProject/classes/Encryption.js
key: kierankieranasdof
type of input Bin: string
input Bin: 1101000 1100101 1101100 1101100 1101111 100000 1110100 1101000 1100101 1110010 1100101 100000 1110111 1101111 1110010 1101100 1100100
key Bin: 1101011 1101001 1100101 1110010 1100001 1101110 1101011 1101001 1100101 1110010 1100001 1101110 1100001 1110011 1100100 1101111 1100110


 XOR Encryption Result with:
 key = kieran
 input = hello there world
 result = 0

Process finished with exit code 0
```

Logical error the xor is being done on two strings not binary digits. result is string XOR string which gives 0

needed to xor each bit individually (firstly need to parse the digit into an integer)

This would have been inefficient and would have been very time consuming

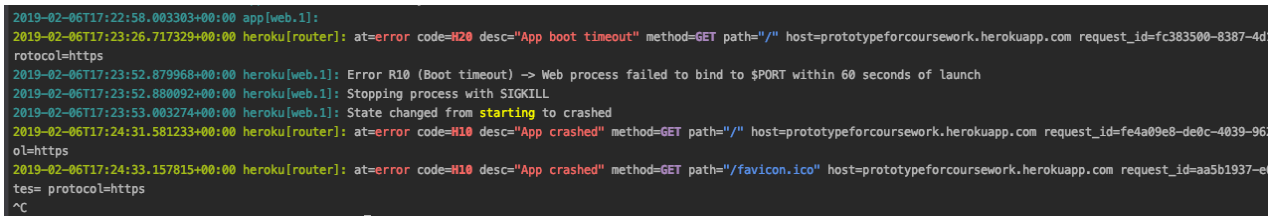
Decided to just xor the utf character codes instead and that takes care of the bits problem

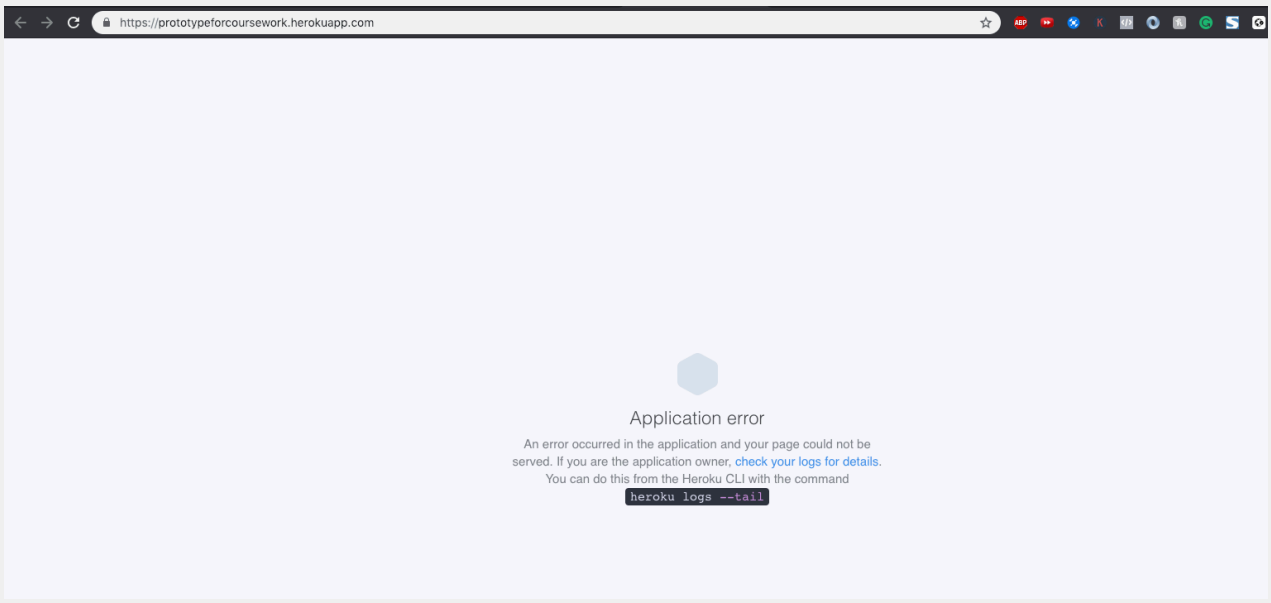
And meant I didn't have to code a parser

==========

uploaded the web file to the heroku server

the app would crash every time I would try to enter the site and in the heroku CLI logger it gave the error code of H10 which is representative for app crashing from time out




the problem was that I was using a hard coded port for the app to run on but for it to work on heroku I need it to be an environment variable so that the server can set it

```
app.listen(8000, (err) => {
    if (err) {
        console.log(err);
    }
    else {
        console.log("server started on port 8000");
    }
});
```

changed the port to use the environment variable if given one if not to use port 8000 which i can use on my local host connection

```
app.listen(process.env.PORT || 8000, (err) => {
    if (err) {
        console.log(err);
    }
    else {
        console.log("server started on port 8000");
    }
});
```

At this point I did not know about environment variables an I learned that they are a good way of allowing an external service provide the value to a variable

==========

In my encryption method there was an encryption error. The password key was not splitting into an array

```
TypeError: key.split is not a function
    at Function.Encryption.(anonymous function) (/Users/kieranwilliams/Desktop/experimental_coursework/programming/kwcrmCourseworkProject/Classes/Encryption.js:14:20)
    at Function.Encryption.xorEncrypt (/Users/kieranwilliams/Desktop/experimental_coursework/programming/kwcrmCourseworkProject/Classes/Encryption.js:31:40)
    at router.post (/Users/kieranwilliams/Desktop/experimental_coursework/programming/kwcrmCourseworkProject/routes/contactsRoutes.js:416:36)
    at Layer.handle [as handle_request] (/Users/kieranwilliams/Desktop/experimental_coursework/programming/kwcrmCourseworkProject/node_modules/express/lib/router/layer.js:95:5)
    at next (/Users/kieranwilliams/Desktop/experimental_coursework/programming/kwcrmCourseworkProject/node_modules/express/lib/router/route.js:137:13)
    at Route.dispatch (/Users/kieranwilliams/Desktop/experimental_coursework/programming/kwcrmCourseworkProject/node_modules/express/lib/router/route.js:112:3)
    at Layer.handle [as handle_request] (/Users/kieranwilliams/Desktop/experimental_coursework/programming/kwcrmCourseworkProject/node_modules/express/lib/router/layer.js:95:5)
    at /Users/kieranwilliams/Desktop/experimental_coursework/programming/kwcrmCourseworkProject/node_modules/express/lib/router/index.js:281:22
    at param (/Users/kieranwilliams/Desktop/experimental_coursework/programming/kwcrmCourseworkProject/node_modules/express/lib/router/index.js:354:14)
    at param (/Users/kieranwilliams/Desktop/experimental_coursework/programming/kwcrmCourseworkProject/node_modules/express/lib/router/index.js:365:14)
```

the problem was that I was using req.body which was sending in a object not the string piece of data

fixed by adding the password property to the end of the object.

```
encryptedData = Encryption.xorEncrypt(extraNotesData, req.body.password);
console.log(encryptedData);
```

==========

| |
|---|
| |
| |

==========

| |
|---|
| |
| |
| |
| |
| |
| |

==========

| |
|---|
| |
| |
| |
| |
| |
| |

==========

| |
|---|
| |
| |

==========

==========

==========

| |
|---|
| |
| |
| |
| |

==========

| |
|---|
| |
| |
| |
| |
| |
| |

==========

| |
|---|
| |
| |
| |
| |
| |
| |

==========