**Install Node.js**

It tries to find a combination of random integers that when entered into the quadratic formula yield real roots.

A black background with white numbers and symbols

Description automatically generated

**Node.js vs. BOM**

1. It does not work because Node.js is not run in a browser so it doesn’t have browser methods like window.prompt() and window.alert().
2. See NodeBOM.mjs
   1. What is different about the way that readline works over window.prompt() and window.alert(), is that readline relies on callback functions and runs asynchronously, while window.prompt() and window.alert() require interaction by the user before they can do anything else with the application.

**Parse a URL with Node.js**

1. **Research the query string - be sure to cite your sources**
   1. **What information does it contain?**

Query strings act as parameters that contain specific values that can later be interpreted as their own variables.

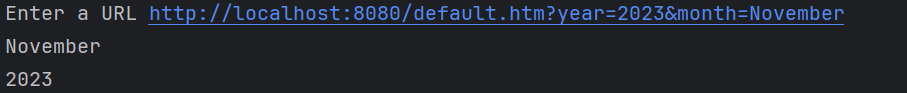
* 1. **Give an example of what a query string is used for?**

With the following URL: “https://www.w3schools.com/test/names.asp?n=John&n=Susan”, both the names “John” and “Susan” are able to be grabbed under the variable labeled “n”.

1. **Research documentation for the URL package. What does this package provide?**

The URL package provides lots of method that are able to parse and format URLs. Some of the most common and useful methods in this library are the parse(), format(), and resolve() methods.

1. **Using the URL package write a Node.js script that:**
   1. **Prompts the user to input a URL.**
   2. **Parses the URL to retrieve the query string.**
   3. **Print query string parameter keys and values to the console.**

****

**Accessing the File System in Node.js**

1. **Research the documentation for the fs package.**

The Node.js file system (fs) allows the ability to interact with the file explorer on your computer. The file system can read, create, update, delete, and rename any files that are specified to the file system.

1. **Using the fs package, write a Node.js script that:**
   1. **Prompts the user for a file name**
   2. **Reads the contents of the file and prints the contents to the console**
      1. **If the file does not exist, print an error to the console**
      2. **If the file size is greater than 50 bytes, print out only the first 50 bytes and then print an ellipsis (...)**

**A black screen with white text

Description automatically generated**

**A black screen with green and white text

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**Third Party Packages**

**Cropper.js |** [**cropperjs - npm (npmjs.com)**](https://www.npmjs.com/package/cropperjs)

A package that allows the ability manipulate and crop images in JavaScript.

**Shunter |** [**shunter - npm (npmjs.com)**](https://www.npmjs.com/package/shunter)

Shunter allows the ability to read JSON files and translate it into HTML.

**Express**

1. **Create a static web server script to serve pages in your project's public directory. Add at least one HTML file and at least one image file to the directory. Make sure your HTML file references your image.**

**A screen shot of a computer

Description automatically generated**A white background with black text

Description automatically generated

1. **What is the difference between a query parameter and a route parameter?**

Query parameters are parameters that can contain strings that can later be used as variables. Normally, query parameters are attached to the end of a URL after a question mark symbol. Routing parameters are parameters that segment a URL to a given route. The difference between these two parameters is that query parameters take in strings and use them as variables while routing parameters direct a user to a given place.

1. **How do you retrieve the query and route parameters using express? Provide a code example of each.**

To get a query parameter, the method “req.query.[ParamName]” will return the value.

Ex: GET /something?name1=John&name2=Doe

app.get('/something', (req, res) => {

const firstName = req.query.name1;

const lastName = req.query.name2;

});

To get the route parameter, we use the “req.params.[ParamName]” method, along with getting the name of the routing parameter.

Ex: GET /users/:userId/books/:bookId

app.get('/users/:userId/books/:bookId’, (req, res) => {

const userId = req.params.userId;

const bookId = req.params.bookId;

});

**Suggestions**

Overall, I enjoyed this lab, as it gave a great introduction into Node.js along with npm packages.

I enjoyed using express and setting up a basic web server.

**Sources**

<https://www.w3schools.com/asp/coll_querystring.asp>

<https://nodejs.org/api/url.html>

<https://www.abstractapi.com/api-glossary/query-parameters>

[Route Parameters | Routing & Navigation | Vaadin Docs](https://vaadin.com/docs/latest/routing/route-parameters#:~:text=Route%20parameters%20are%20variable%20parts%20of%20the%20URL,to%20pass%20extra%20information%20to%20a%20given%20route.)

<https://stackoverflow.com/questions/17007997/how-to-access-the-get-parameters-after-in-express>

<https://expressjs.com/en/guide/routing.html#route-parameters>