Database Project Assignment 7: Web Connectivity

Nicholas Weibler

**Connecting to Database:**

databseConnect.js File

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

On line 2, the mysql variable is set to use the mysql module.

A screenshot of a computer

Description automatically generated with medium confidence

Lines 5-11 creates a variable called connection and uses the mysql module to define the configuration that can be used to connect to my database. The configuration is described by lines 6-10 using the host, user, password, database and port specific to the eBay database.

A screenshot of a computer

Description automatically generated with medium confidence

Line 14 and 15 uses the connection variable that was set above to connect to the database. If there is an issue ( i.e. incorrect information ), an error will be displayed.

Lines 16-19 are commented out but were used to check if there was a connection to the database. It will uses the “SELECT \* FROM employee” to show the employee table on the console.

A screenshot of a computer

Description automatically generated with medium confidence

Line 23 allows the connection information available to be used with other modules, which is needed for the webserver.js file.

**Connecting to the webserver:**

webserver.js file:

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Lines 1 allows the webserver to use the databaseConnect file using a variable called connection.

Lines 2 and 3 uses Express which is a web framework that allows the following get requests to grab and send data to the webserver.

A screenshot of a computer

Description automatically generated with medium confidence

The first get request is shown on lines 6-15. This sets the route /employee. It will query the database using “SELECT \* FROM employee” which will grab the entire employee table. If there are any issues with the connection lines 8 and 9 will allow the program to throw an error, otherwise it will send the results of the query to the webserver.

A screenshot of a computer

Description automatically generated with medium confidence

Lines 17-27 are almost identical to the previous get request. The only change is that it will set a new route using /supplier and will query the supplier table.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Finally lines 29 and 30 will set the port to 8080. And will allow the program to listen for connections on that port. Line 31 will display the message “Connected on port: 8080” affirming that a good connection has been made.

localhost:8080/employee Screenshot

Graphical user interface, application, Word

Description automatically generated

The result from the employee table query is displayed on localhost:8080/employee in JSON format.

localhost:8080/supplier Screenshot

Graphical user interface, application, Word

Description automatically generated

The result from the supplier table query is displayed on localhost:8080/supplier in JSON format.