

Group: Team TBD
De La Rosa, Adrian 916835002 Centroids
Robles, Marco 921282632 CaliPrerunner
Jian, Shengwei 921151618 Sheng333
Cintas, Alfonso 922293720 alfonsoCintas

Team TBD: Storefront Final Assignment

<https://github.com/CSC317-Fall-2021/csc317-group-html-storefront-CaliPrerunner>

Findings:

The final additions we made to our website was setting up and adding a database that was using node.js and mySQL. We then had to migrate all of our products and js files over to the database. We also added a service to our contact us page that sent the administrator an email once someone filled out the form with all the details of the form and also sent a confirmation email to the customer. We also cleaned up the sign in and create account form to make it look cleaner.

Issues:

We had issues during the presentation which was on Adrian's laptop, there was not enough disk storage to pull all the resources on the VM so we had to run it straight from the file system. We also had an initial problem with the emailing logic. We looked at the BLOB packages built into node.js. but we found a more simple package that was built into Js that allowed us to serve emails from a gmail account.

We had an issue with the css id tags, each tag was tied to the wrong class making the colors and sizes unproportional. To fix that we had to create new ID names and classes to sort things out.

We also ran into issues incorporating old redundant structures from our previous versions of the site and then having to brainstorm and implement our final storefront with these same deprecated forms. The solution turned out to be instead simplifying and applying some polymorphism to the div element id's that would allow for our javascript to easily traverse many iterations of the same or somewhat same forms. Ex of this in our website structure can be seen in the login and contact us pages where we began with large blobs to be filled out by us but no actual way of moving our data from the front end to our backend. These forms at least had some bits of skeleton code to help us see where to go from 0 but nevertheless the uphill climb to implement functionality in our old structures came down to needing a simplified version or versions of the websites forms, data input, plugins, you name it, but there needed to be a central structure and vision to propel the site to working demo range.

Group: Team TBD
De La Rosa, Adrian 916835002 Centroids
Robles, Marco 921282632 CaliPrerunner
Jian, Shengwei 921151618 Sheng333
Cintas, Alfonso 922293720 alfonsoCintas

Solutions:

To fix the email service we found a more simple package that was built into JS that allowed us to serve emails from a gmail account.

Database implementation began with small pieces of linker code from node js. We used the packages for accessibility into mySQL. These libraries would prove extremely useful in parsing back user data from the website and back into our server and this made it easy as well to grab whichever website data we decided to keep on the backend. We created a few javascript files to test the functionality of our database and to begin the process of filling in our data from the site. The example below is of our configuration file on successful load into mySQL.


```
JS config.js M X JS app.js M <> paymentPage.html
JS config.js > ...
1  var mysql = require('mysql');
2
3  //credentials into config file for the localhost mysql server
4  let config = {
5      host: "127.0.0.1",
6      port: '3306',
7      user: 'root',
8      password: 'student123',
9      database: 'sys'
10 };
11 //export those credentials
12 module.exports = config;
13
14 //open connection to mySQL db 'sys' where our website DB is
15 let con = mysql.createConnection({
16     host: "127.0.0.1",
17     port: "3306",
18     user: "root",
19     password: "student123",
20     database: "sys"
21 });
22
23 //function for successful connection to mySQL server, used for testing
24 con.connect(function(err) {
25     if (err) {
26         return console.error('error' + err.message);
27     }
28     console.log("Connected to MySQL server!");
29 });
30
31 //function for closing connection to mySQL
32 con.end(function(err) {
33     if (err) {
34         return console.log('error:' + err.message);
35     }
36     console.log('Close the database connection.');
```

All loaded using command line 'node config.js' which would parse the js to mySQL.

Group: Team TBD
De La Rosa, Adrian 916835002 Centroids
Robles, Marco 921282632 CaliPrerunner
Jian, Shengwei 921151618 Sheng333
Cintas, Alfonso 922293720 alfonsoCintas

Contact Us Email Service:


Sent to administrator:

 barkt140@gmail.com
Mon 12/6/2021 10:47 AM
To: Marco A Robles

A customer has contated us!! First Name: Marco Last Name: Robles Phone Number: 3109229156 Email: marco@marcorobles.com Comment Box: This is now working!!

[Reply](#) | [Forward](#)

Sent to customer:

 barkt140@gmail.com <barkt140@gmail.com>
to adelaro2 ▾

2:28 PM (6 hours ago) ☆ ↶ ⋮

Thank you for reaching out to us. We will contact you in 2-3 business days. If your need is urgent please call us at (888)-888-8888

[↶ Reply](#) [➡ Forward](#)