Team NetCrew

Prepard By

Robert Geist

WEBSITE DEVELOPMENT PROPOSAL

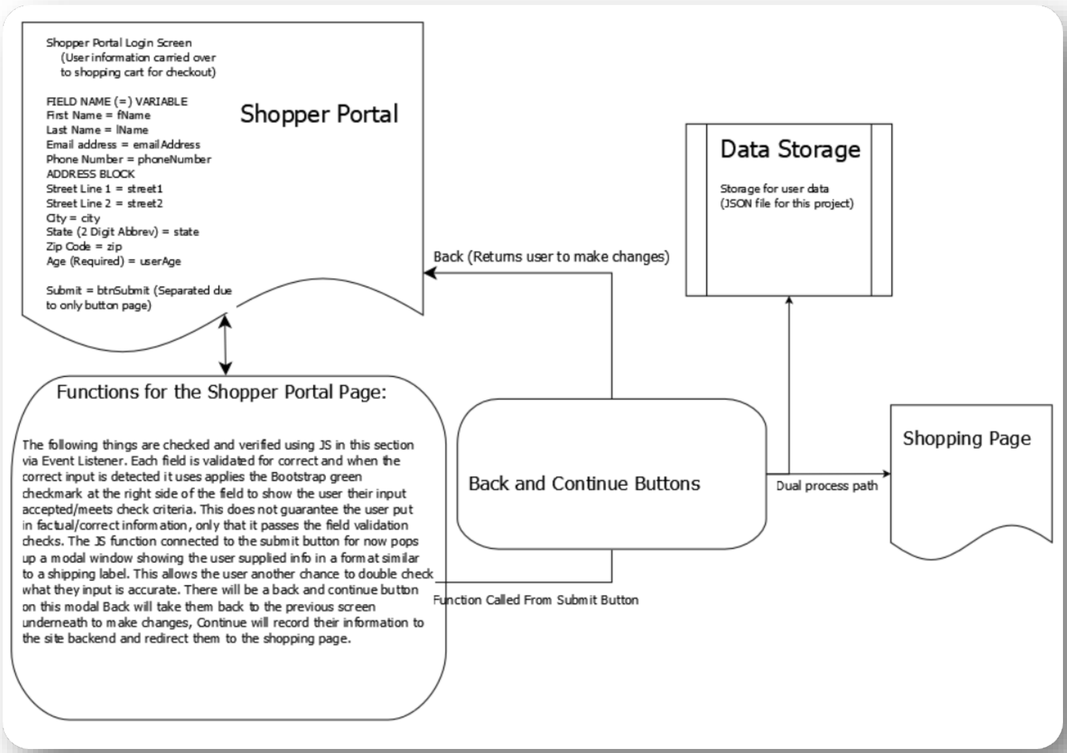
Team NetCrew:

Jamal Barker
James Do
Grace Kang
Robert Geist
Alajah Rivera

February 25, 2024

NetCrew Shopping Site (Deliverable 1)

Portal Flow Document Image:


Shopper Portal
Data Storage
Storage for user data
(JSON file for this project)
Back (Returns user to make changes)
Functions for the Shopper Portal Page:
Back and Continue Buttons
Dual process path
Shopping Page
Function Called From Submit Button

Shopper Portal Login Screen
(User information carried over to shopping cart for checkout)

FIELD NAME (=) VARIABLE
First Name = fName
Last Name = lName
Email address = emailAddress
Phone Number = phoneNumber
ADDRESS BLOCK
Street Line 1 = street1
Street Line 2 = street2
City = city
State (2 Digit Abbrev) = state
Zip Code = zip
Age (Required) = userAge

Submit = btnSubmit (Separated due to only button page)

The following things are checked and verified using JS in this section via Event Listener. Each field is validated for correct and when the correct input is detected it uses applies the Bootstrap green checkmark at the right side of the field to show the user their input accepted/meets check criteria. This does not guarantee the user put in factual/correct information, only that it passes the field validation checks. The JS function connected to the submit button for now pops up a modal window showing the user supplied info in a format similar to a shipping label. This allows the user another chance to double check what they input is accurate. There will be a back and continue button on this modal Back will take them back to the previous screen underneath to make changes, Continue will record their information to the site backend and redirect them to the shopping page.

Figure 1 Flow Diagram

Description:

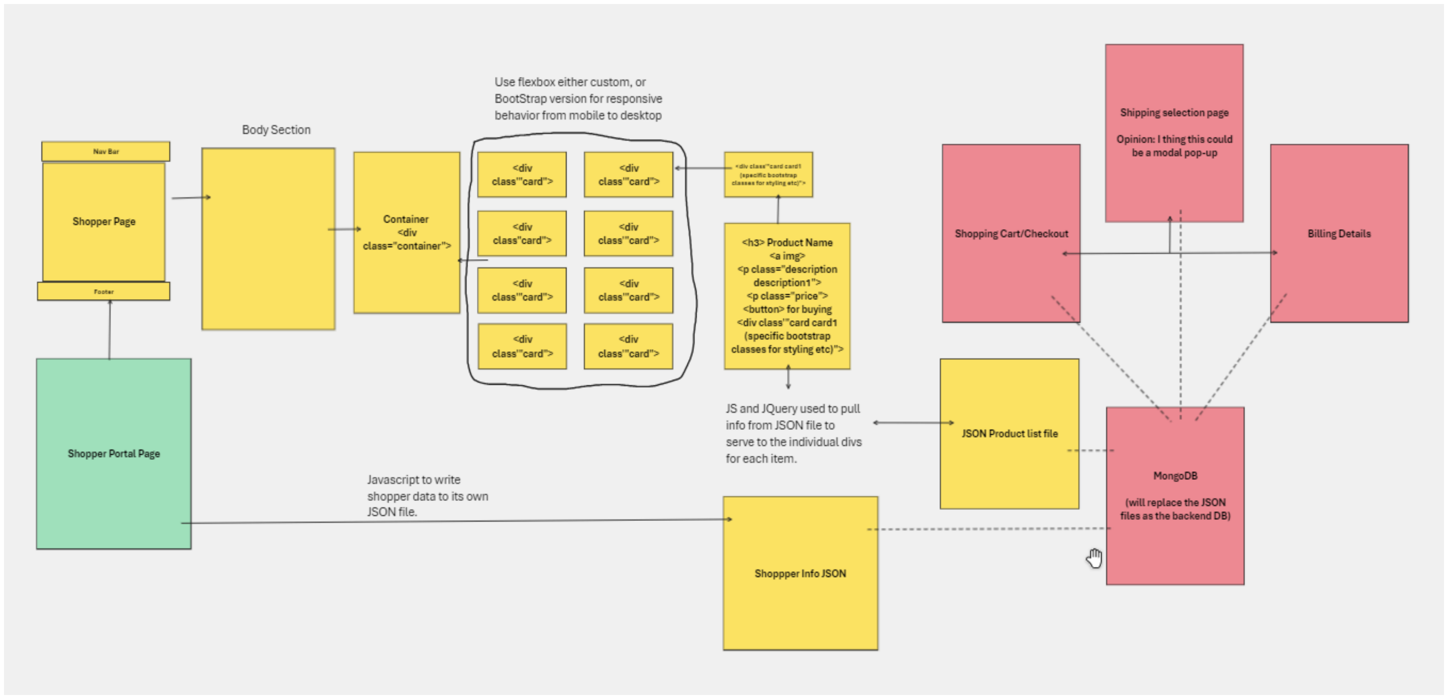
The above Screenshot gives an overview of the general flow of the initial document prototype. It shows how the input from the user is handled and where the user will end up when they correctly input the correct information to continue shopping. The block labeled Shopping Page is the next step in the site progress and will be represented in a different future diagram.

We will be incorporating multiple technologies in bringing this shopping site together, they are listed here along with the respective team members responsible for contributing to that area. It should also be noted that there is overlap in the process and method so any one team member could make some contribution in multiple areas.

Tech, and Team Member:

Technology	Team Member
HTML	James Do, Alajah Rivera
CSS	James Do, Alajah Rivera
BootStrap	Grace Kang
JavaScript	Rob Geist (Front and Back End)
JSON and other Database tech.	Jamal Barker

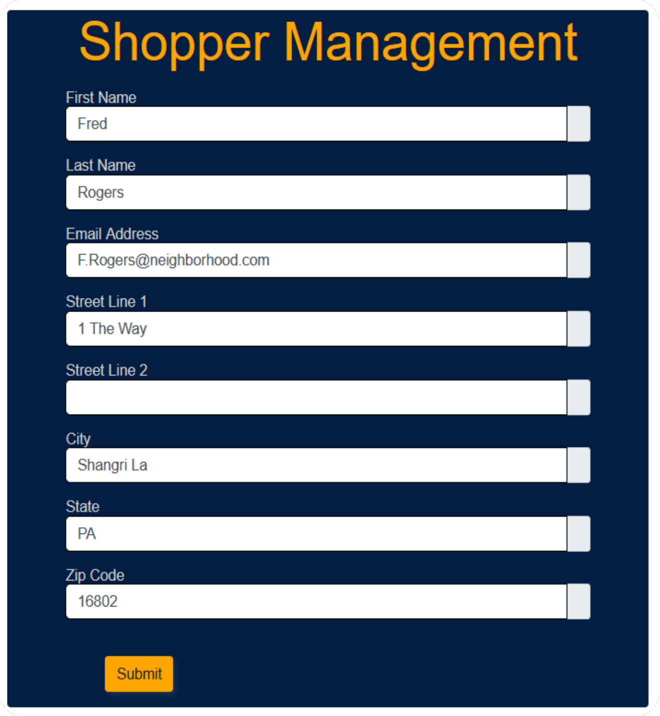
Project Layout:



Nav Bar
Shopper Page
Footer
Body Section
Container <div class="container">
Use flexbox either custom, or BootStrap version for responsive behavior from mobile to desktop
<div class'"card">
<div class"card">
<div class'"card">
<div class'"card">
<div class'"card">
<div class'"card">
<div class'"card">
<div class'"card">
<div class'"card card1 (specific bootstrap classes for styling etc)">
<h3> Product Name <a img> <p class="description description1"> <p class="price"> <button> for buying <div class'"card card1 (specific bootstrap classes for styling etc)">
JS and JQuery used to pull info from JSON file to serve to the individual divs for each item.
Shopper Portal Page
Javascript to write shopper data to its own JSON file.
Shoppper Info JSON
JSON Product list file
Shopping Cart/Checkout
Shipping selection page
Opinion: I thing this could be a modal pop-up
Billing Details
MongoDB
(will replace the JSON files as the backend DB)

Figure 2 Project Block Diagram

Initial Page Prototype Screen Shots:


Shopper Management

First Name
Fred

Last Name
Rogers

Email Address
F.Rogers@neighborhood.com

Street Line 1
1 The Way

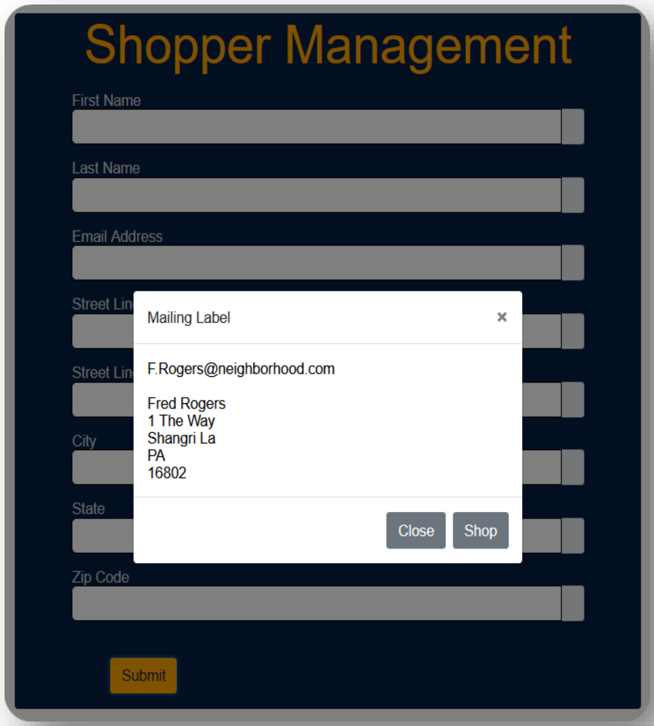
Street Line 2

City
Shangri La

State
PA

Zip Code
16802

Figure 2 Shopper page


Shopper Management

Mailing Label

F.Rogers@neighborhood.com

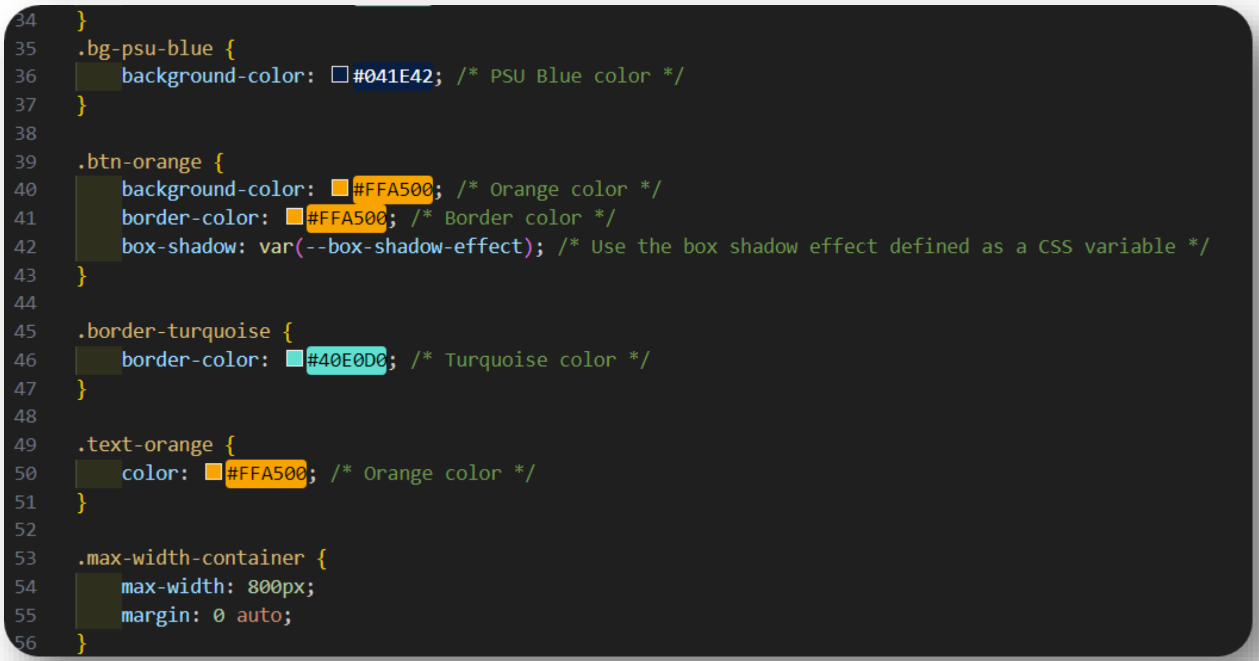
Fred Rogers
1 The Way
Shangri La
PA
16802

Figure 3 Shopper page with modal window

Description:

Sample Prototype Screenshot for initial page. Information was entered to show user input is in effect. Each of the fields will contain validation to ensure accuracy of the user input. There is also a hover element added to the submit button, so mouse users know when they are over the button. When the form is submitted it will present the user with a modal showing their entered information in a shipping label format to review for accuracy. There are two buttons on the modal: [Close], and [Shop]. If user info is correct, then click shop to continue on to the shopping page. If changes need made, then click close and edit the info and submit again.

Custom Site-Specific CSS Sample:



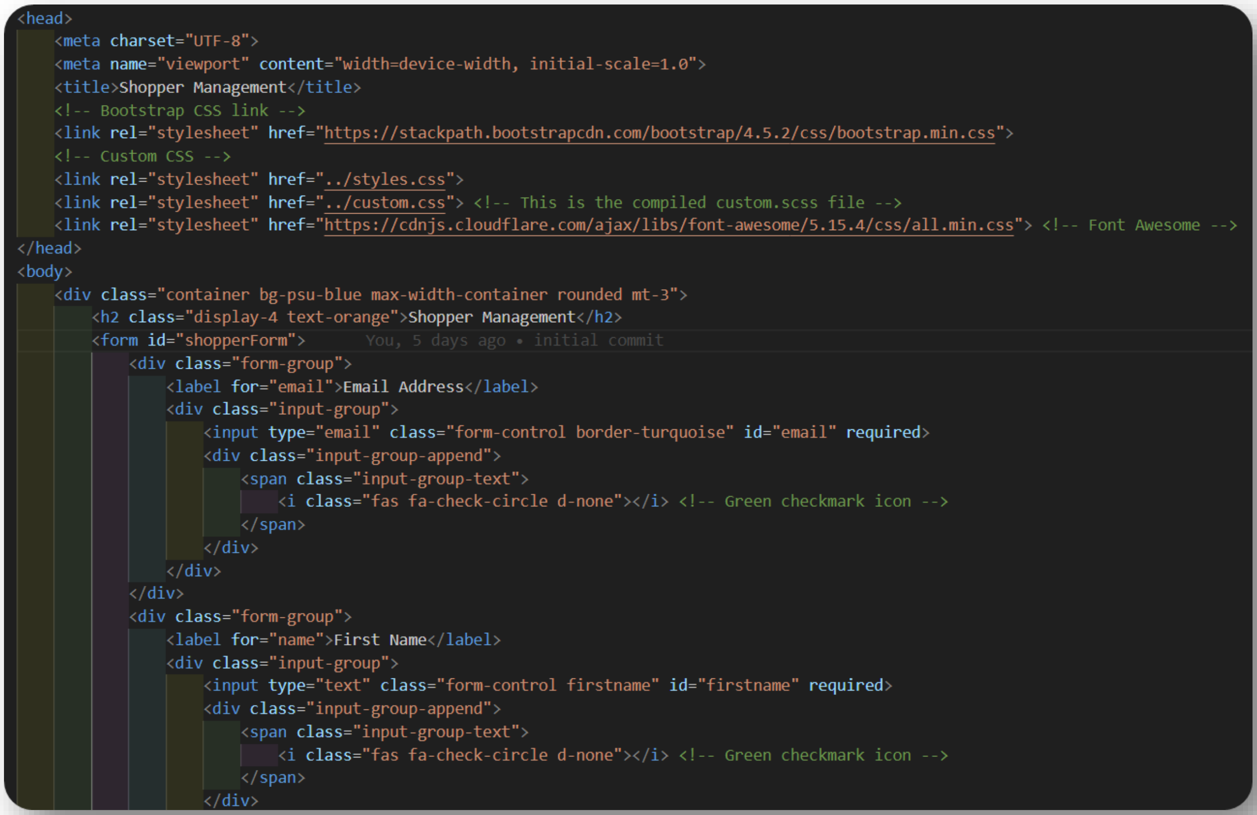
```css
}
.bg-psu-blue {
    background-color: #041E42; /* PSU Blue color */
}

.btn-orange {
    background-color: #FFA500; /* Orange color */
    border-color: #FFA500; /* Border color */
    box-shadow: var(--box-shadow-effect); /* Use the box shadow effect defined as a CSS variable */
}

.border-turquoise {
    border-color: #40E0D0; /* Turquoise color */
}

.text-orange {
    color: #FFA500; /* Orange color */
}

.max-width-container {
    max-width: 800px;
    margin: 0 auto;
}
```

Figure 4 CSS Sample Screen Shot

Description:

The CSS code provided displays the detail-oriented features for the webpage above. Within this code, the background color commented “PSU Blue” is displayed along the store’s webpage. The background color commented “Orange color” is used to fill in the radio button labeled “Submit”. The text color is also orange to display “Shopper Management”. We added shadow effects along the border of the page to create dimension.

Custom Site-Specific HTML Sample:



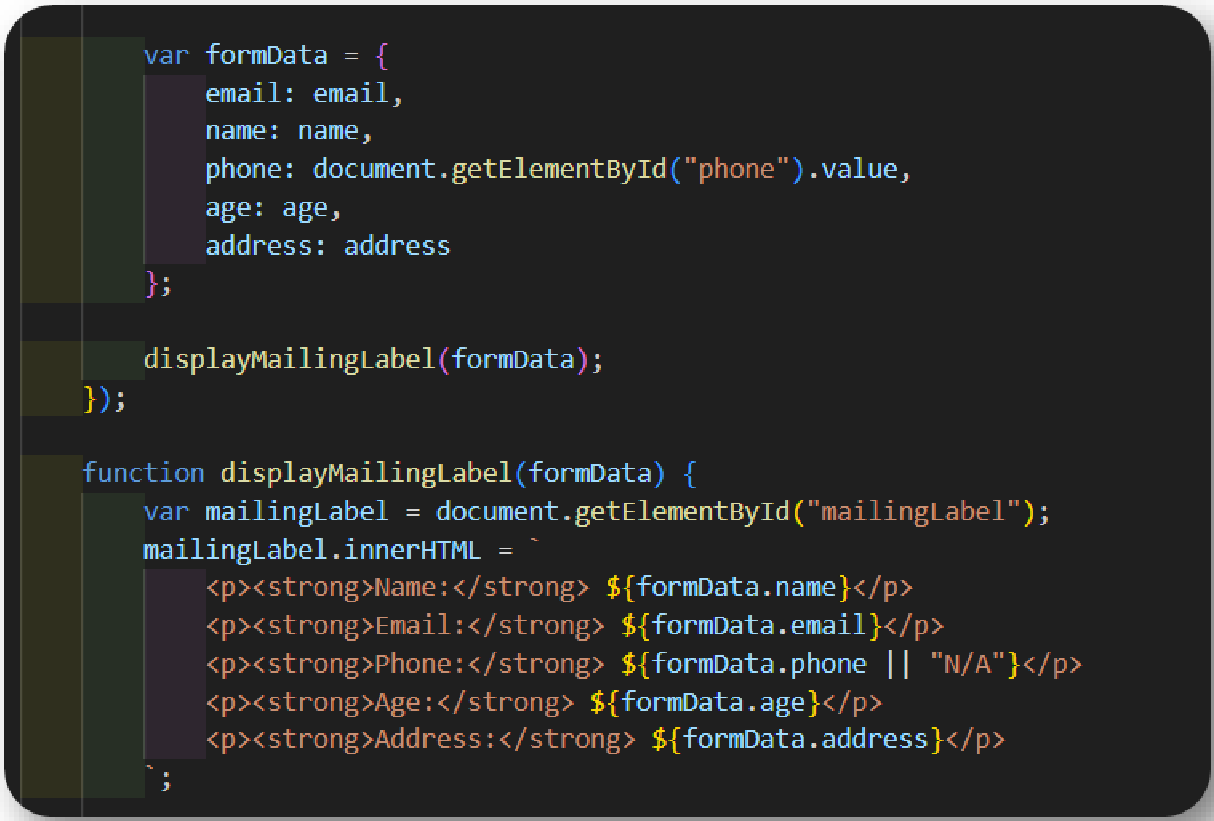
```html
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Shopper Management</title>
    <!-- Bootstrap CSS link -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <!-- Custom CSS -->
    <link rel="stylesheet" href="../styles.css">
    <link rel="stylesheet" href="../custom.css"> <!-- This is the compiled custom.scss file -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css"> <!-- Font Awesome -->
</head>
<body>
    <div class="container bg-psu-blue max-width-container rounded mt-3">
        <h2 class="display-4 text-orange">Shopper Management</h2>
        <form id="shopperForm">      You, 5 days ago • initial commit
            <div class="form-group">
                <label for="email">Email Address</label>
                <div class="input-group">
                    <input type="email" class="form-control border-turquoise" id="email" required>
                    <div class="input-group-append">
                        <span class="input-group-text">
                            <i class="fas fa-check-circle d-none"></i> <!-- Green checkmark icon -->
                        </span>
                    </div>
                </div>
            </div>
            <div class="form-group">
                <label for="name">First Name</label>
                <div class="input-group">
                    <input type="text" class="form-control firstname" id="firstname" required>
                    <div class="input-group-append">
                        <span class="input-group-text">
                            <i class="fas fa-check-circle d-none"></i> <!-- Green checkmark icon -->
                        </span>
                    </div>
```

Figure 5 HTML Sample Screen Shot

Description:

The HTML code below provides the foundation of our storefront webpage. The header was created to display “Shopper Management” and below, labels were created. This includes all of the labels needed for user input. We are asking the user to input “First Name, Last Name, Email Address, Street, City, State, and Zip code”. This will ensure a proper mailing label, when items need to get shipped out to the customer. Within the class, we added a character/string checker to ensure that the information being inputted by the user is accurate and that the code will except the data. If the data meets our codes requirements a green check mark icon will appear.

Custom Site-Specific JavaScript Sample Screen Shot:



```
        var formData = {
            email: email,
            name: name,
            phone: document.getElementById("phone").value,
            age: age,
            address: address
        };

        displayMailingLabel(formData);
    });

    function displayMailingLabel(formData) {
        var mailingLabel = document.getElementById("mailingLabel");
        mailingLabel.innerHTML = `
            <p><strong>Name:</strong> ${formData.name}</p>
            <p><strong>Email:</strong> ${formData.email}</p>
            <p><strong>Phone:</strong> ${formData.phone || "N/A"}</p>
            <p><strong>Age:</strong> ${formData.age}</p>
            <p><strong>Address:</strong> ${formData.address}</p>
        `;
```

Figure 6 JavaScript Sample Screen Shot

Description:

The purpose of this JavaScript code is to facilitate the collection and presentation of user-provided data, likely from a web form. It constructs an object named ‘formData’ to organize information such as email, name, phone number, age, and address. Additionally, the ‘displayMailingLabel’ function is designed to visually present this collected data, typically in a mailing label format on a web page. The function dynamically updates the content of an HTML element identified by the ‘mailingLabel’ id to display the gathered information in a structured manner. This code serves to streamline the process of capturing user input and presenting it in a user-friendly format for various applications, such as generating mailing labels or displaying user profiles.