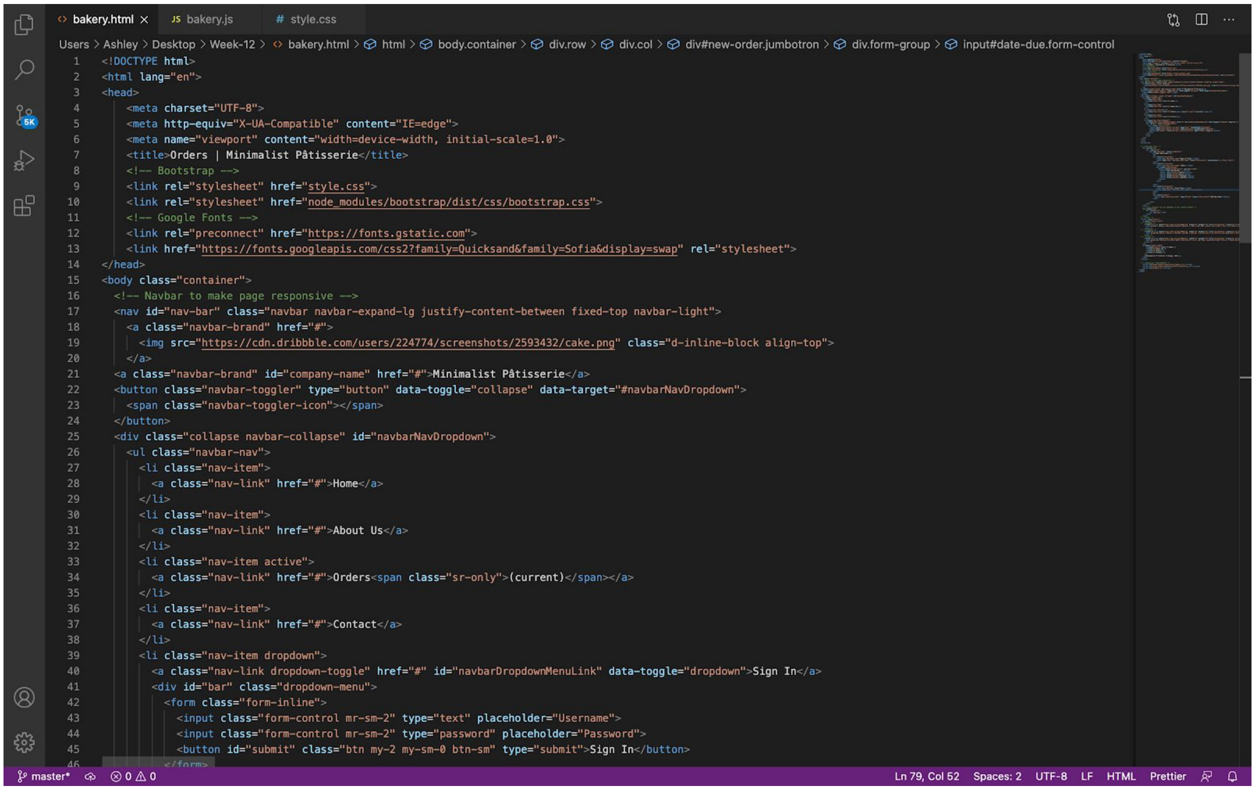
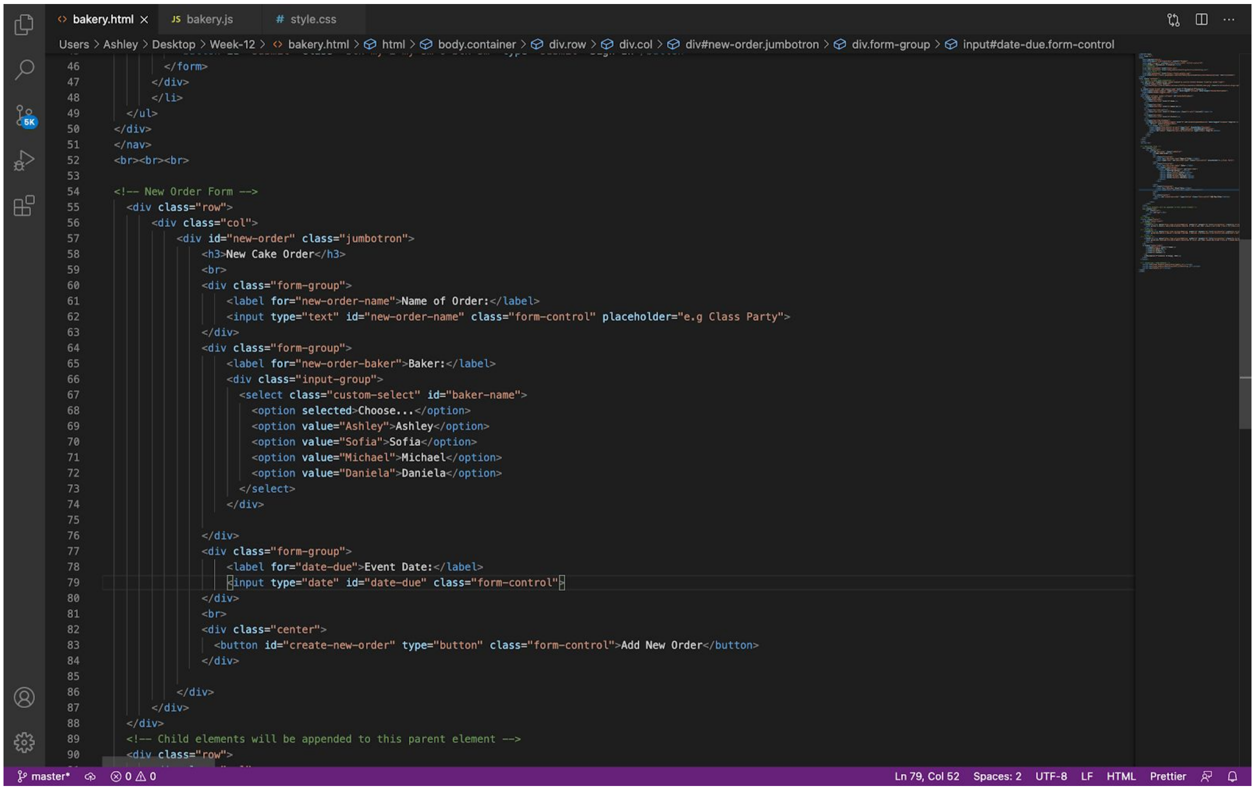
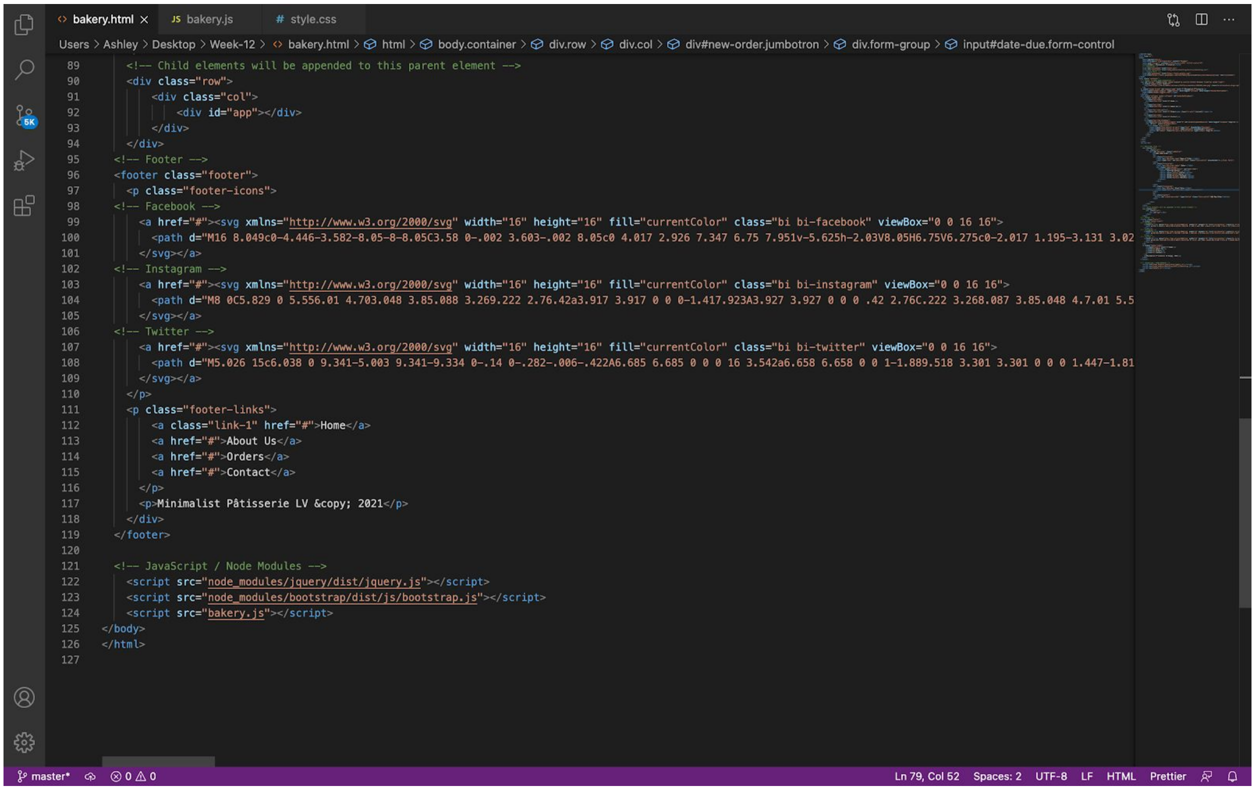
Screenshots of Code:


bakery.html
bakery.js
style.css
Users > Ashley > Desktop > Week-12 > bakery.html > html > body.container > div.row > div.col > div#new-order.jumbotron > div.form-group > input#date-due.form-control
5K
master*
0 0
Ln 79, Col 52
Spaces: 2
UTF-8
LF
HTML
Prettier

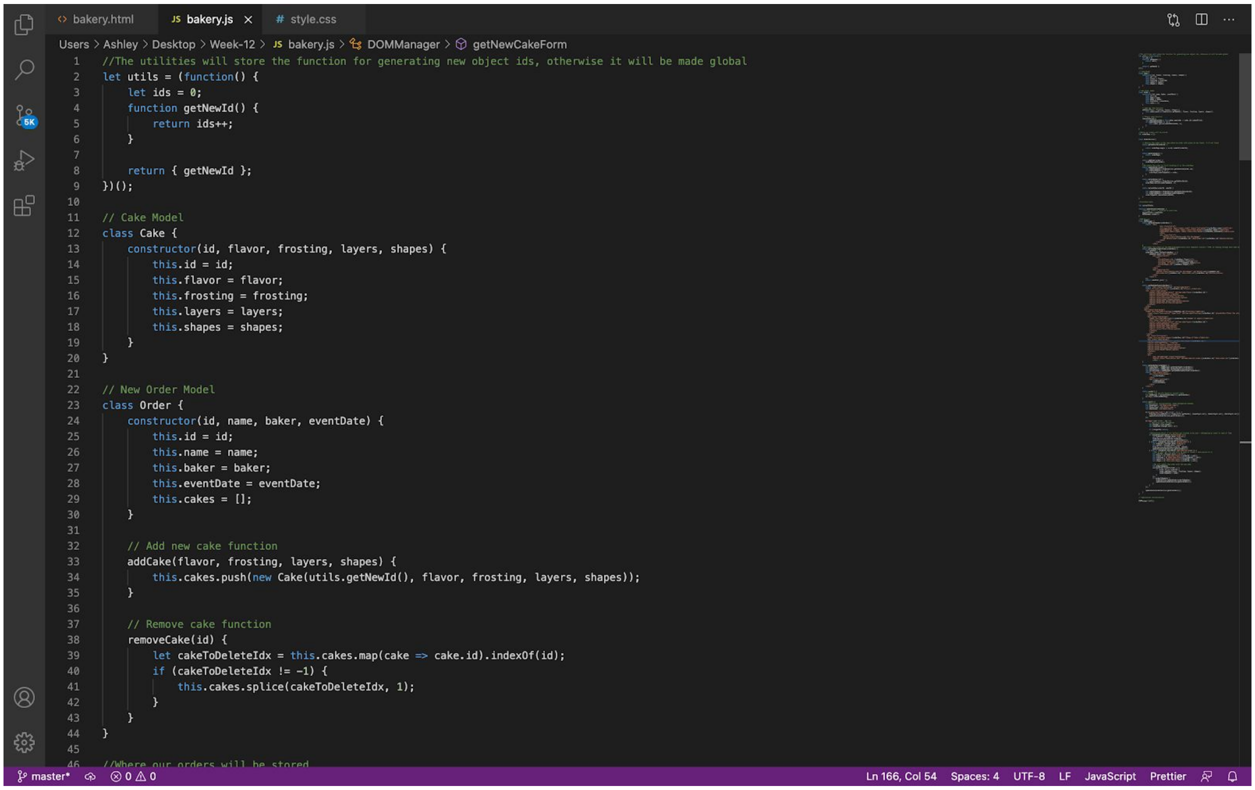
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Orders | Minimalist Pâtisserie</title>
    <!-- Bootstrap -->
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.css">
    <!-- Google Fonts -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Quicksand&family=Sofia&display=swap" rel="stylesheet">
</head>
<body class="container">
  <!-- Navbar to make page responsive -->
  <nav id="nav-bar" class="navbar navbar-expand-lg justify-content-between fixed-top navbar-light">
    <a class="navbar-brand" href="#">
      <img src="https://cdn.dribbble.com/users/224774/screenshots/2593432/cake.png" class="d-inline-block align-top">
    </a>
  <a class="navbar-brand" id="company-name" href="#">Minimalist Pâtisserie</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNavDropdown">
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="#">Home</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">About Us</a>
      </li>
      <li class="nav-item active">
        <a class="nav-link" href="#">Orders<span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Contact</a>
      </li>
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown">Sign In</a>
        <div id="bar" class="dropdown-menu">
          <form class="form-inline">
            <input class="form-control mr-sm-2" type="text" placeholder="Username">
            <input class="form-control mr-sm-2" type="password" placeholder="Password">
            <button id="submit" class="btn my-2 my-sm-0 btn-sm" type="submit">Sign In</button>
```


bakery.html
bakery.js
style.css
Users > Ashley > Desktop > Week-12 > bakery.html > html > body.container > div.row > div.col > div#new-order.jumbotron > div.form-group > input#date-due.form-control
5K
master*
0 0
Ln 79, Col 52
Spaces: 2
UTF-8
LF
HTML
Prettier

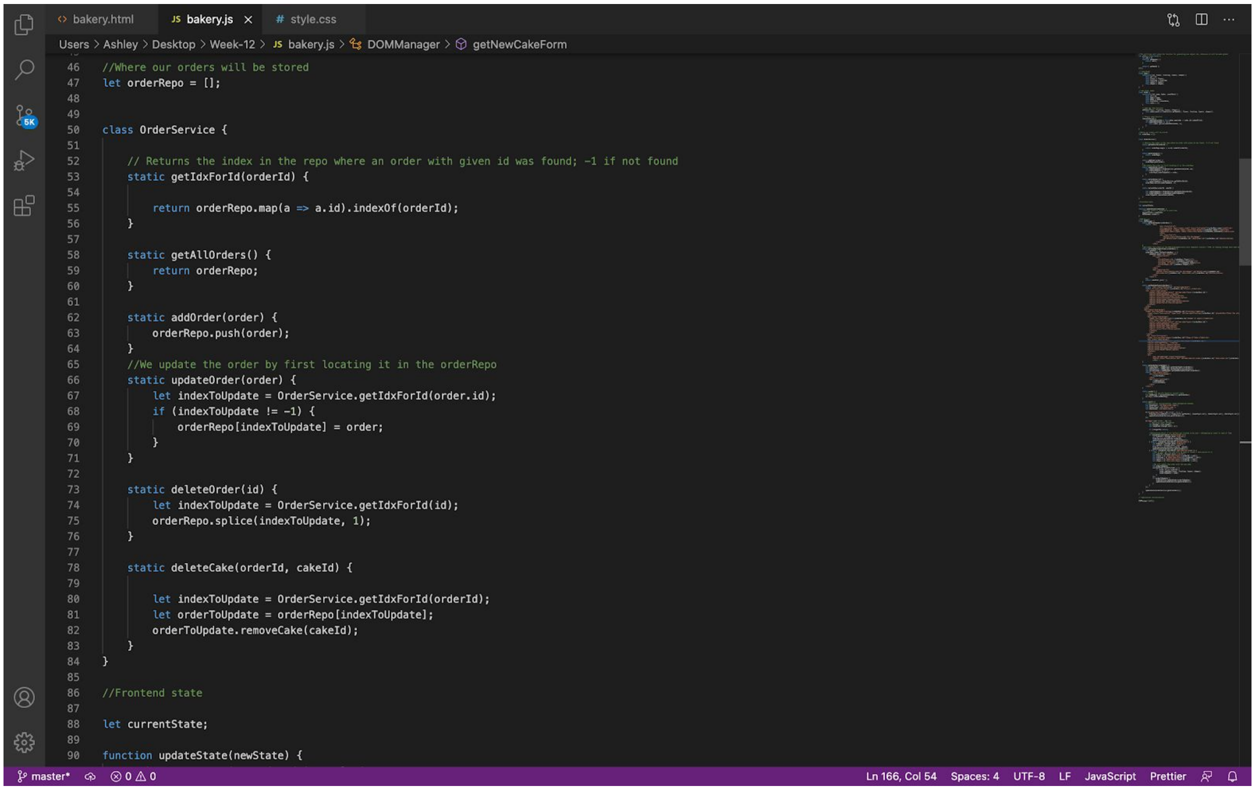
```html
          </form>
        </div>
        </li>
    </ul>
  </div>
  </nav>
  <br><br><br>

  <!-- New Order Form -->
    <div class="row">
        <div class="col">
            <div id="new-order" class="jumbotron">
                <h3>New Cake Order</h3>
                <br>
                <div class="form-group">
                    <label for="new-order-name">Name of Order:</label>
                    <input type="text" id="new-order-name" class="form-control" placeholder="e.g Class Party">
                </div>
                <div class="form-group">
                    <label for="new-order-baker">Baker:</label>
                    <div class="input-group">
                      <select class="custom-select" id="baker-name">
                        <option selected>Choose...</option>
                        <option value="Ashley">Ashley</option>
                        <option value="Sofia">Sofia</option>
                        <option value="Michael">Michael</option>
                        <option value="Daniela">Daniela</option>
                      </select>
                    </div>

                </div>
                <div class="form-group">
                    <label for="date-due">Event Date:</label>
                    <input type="date" id="date-due" class="form-control">
                </div>
                <br>
                <div class="center">
                  <button id="create-new-order" type="button" class="form-control">Add New Order</button>
                </div>

            </div>
        </div>
    </div>
    <!-- Child elements will be appended to this parent element -->
    <div class="row">
```


bakery.html
bakery.js
style.css
Users > Ashley > Desktop > Week-12 > bakery.html > html > body.container > div.row > div.col > div#new-order.jumbotron > div.form-group > input#date-due.form-control
5K
master*
0 0
Ln 79, Col 52 Spaces: 2 UTF-8 LF HTML Prettier

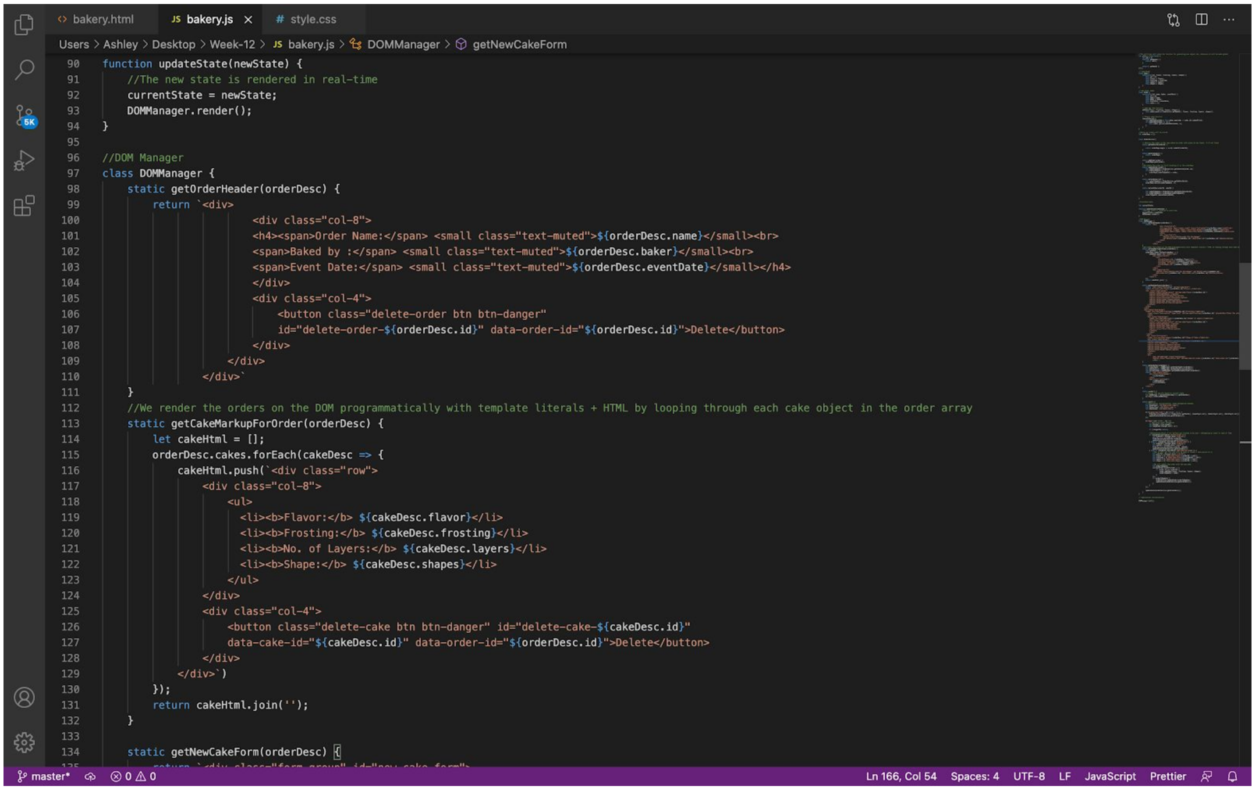
```html
    <!-- Child elements will be appended to this parent element -->
    <div class="row">
        <div class="col">
            <div id="app"></div>
        </div>
    </div>
  <!-- Footer -->
  <footer class="footer">
    <p class="footer-icons">
  <!-- Facebook -->
      <a href="#"><svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-facebook" viewBox="0 0 16 16">
        <path d="M16 8.049c0-4.446-3.582-8.05-8-8.05C3.58 0-.002 3.603-.002 8.05c0 4.017 2.926 7.347 6.75 7.951v-5.625h-2.03V8.05H6.75V6.275c0-2.017 1.195-3.131 3.02
      </svg></a>
  <!-- Instagram -->
      <a href="#"><svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-instagram" viewBox="0 0 16 16">
        <path d="M8 0C5.829 0 5.556.01 4.703.048 3.85.088 3.269.222 2.76.42a3.917 3.917 0 0 0-1.417.923A3.927 3.927 0 0 0 .42 2.76C.222 3.268.087 3.85.048 4.7.01 5.5
      </svg></a>
  <!-- Twitter -->
      <a href="#"><svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-twitter" viewBox="0 0 16 16">
        <path d="M5.026 15c6.038 0 9.341-5.003 9.341-9.334 0-.14 0-.282-.006-.422A6.685 6.685 0 0 0 16 3.542a6.658 6.658 0 0 1-1.889.518 3.301 3.301 0 0 0 1.447-1.81
      </svg></a>
    </p>
    <p class="footer-links">
        <a class="link-1" href="#">Home</a>
        <a href="#">About Us</a>
        <a href="#">Orders</a>
        <a href="#">Contact</a>
      </p>
      <p>Minimalist Pâtisserie LV &copy; 2021</p>
    </div>
  </footer>

  <!-- JavaScript / Node Modules -->
    <script src="node_modules/jquery/dist/jquery.js"></script>
    <script src="node_modules/bootstrap/dist/js/bootstrap.js"></script>
    <script src="bakery.js"></script>
</body>
</html>
```


bakery.html
bakery.js
style.css
Users > Ashley > Desktop > Week-12 > bakery.js > DOMManager > getNewCakeForm
5K
master*
0 0
Ln 166, Col 54 Spaces: 4 UTF-8 LF JavaScript Prettier

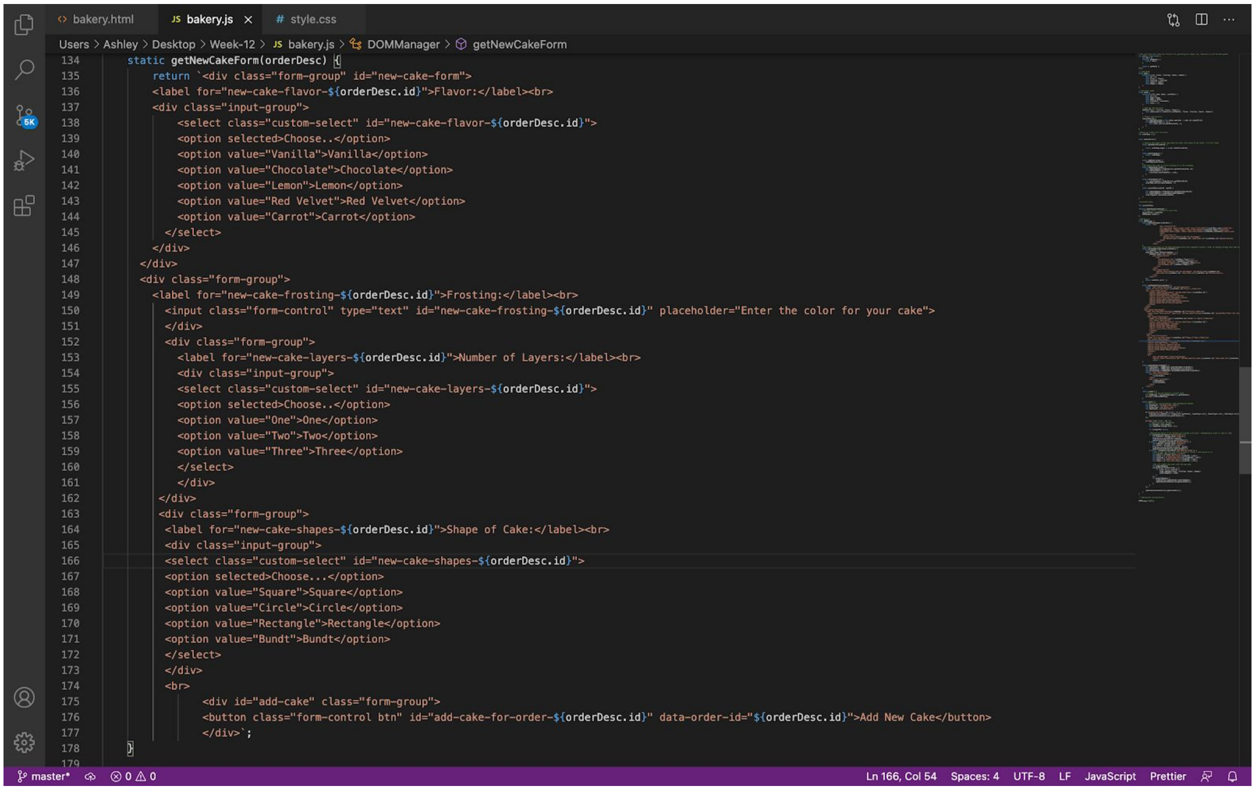
```javascript
//The utilities will store the function for generating new object ids, otherwise it will be made global
let utils = (function() {
    let ids = 0;
    function getNewId() {
        return ids++;
    }

    return { getNewId };
})();

// Cake Model
class Cake {
    constructor(id, flavor, frosting, layers, shapes) {
        this.id = id;
        this.flavor = flavor;
        this.frosting = frosting;
        this.layers = layers;
        this.shapes = shapes;
    }
}

// New Order Model
class Order {
    constructor(id, name, baker, eventDate) {
        this.id = id;
        this.name = name;
        this.baker = baker;
        this.eventDate = eventDate;
        this.cakes = [];
    }

    // Add new cake function
    addCake(flavor, frosting, layers, shapes) {
        this.cakes.push(new Cake(utils.getNewId(), flavor, frosting, layers, shapes));
    }

    // Remove cake function
    removeCake(id) {
        let cakeToDeleteIdx = this.cakes.map(cake => cake.id).indexOf(id);
        if (cakeToDeleteIdx != -1) {
            this.cakes.splice(cakeToDeleteIdx, 1);
        }
    }
}

//Where our orders will be stored
```



```javascript
//Where our orders will be stored
let orderRepo = [];


class OrderService {

    // Returns the index in the repo where an order with given id was found; -1 if not found
    static getIdxForId(orderId) {

        return orderRepo.map(a => a.id).indexOf(orderId);
    }

    static getAllOrders() {
        return orderRepo;
    }

    static addOrder(order) {
        orderRepo.push(order);
    }
    //We update the order by first locating it in the orderRepo
    static updateOrder(order) {
        let indexToUpdate = OrderService.getIdxForId(order.id);
        if (indexToUpdate != -1) {
            orderRepo[indexToUpdate] = order;
        }
    }

    static deleteOrder(id) {
        let indexToUpdate = OrderService.getIdxForId(id);
        orderRepo.splice(indexToUpdate, 1);
    }

    static deleteCake(orderId, cakeId) {

        let indexToUpdate = OrderService.getIdxForId(orderId);
        let orderToUpdate = orderRepo[indexToUpdate];
        orderToUpdate.removeCake(cakeId);
    }
}

//Frontend state

let currentState;

function updateState(newState) {
```

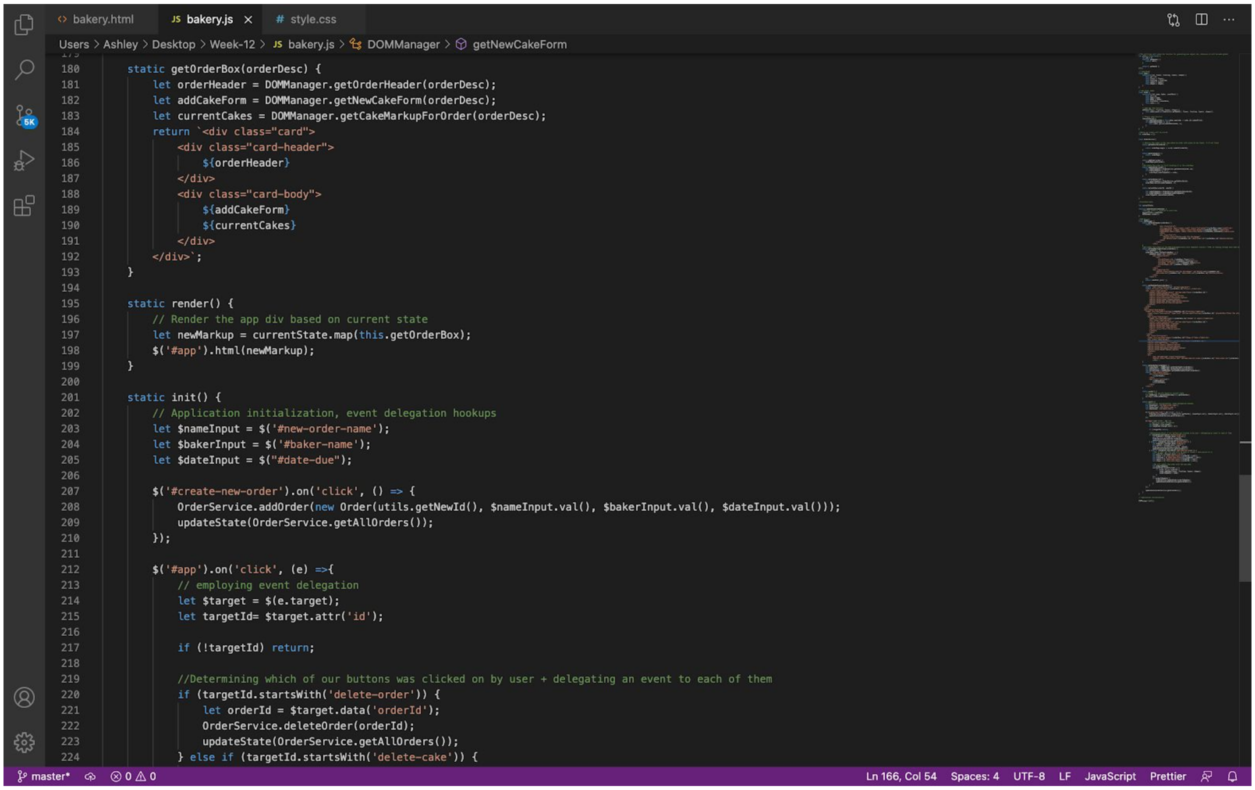


```javascript
function updateState(newState) {
    //The new state is rendered in real-time
    currentState = newState;
    DOMManager.render();
}

//DOM Manager
class DOMManager {
    static getOrderHeader(orderDesc) {
        return `<div>
                        <div class="col-8">
                        <h4><span>Order Name:</span> <small class="text-muted">${orderDesc.name}</small><br>
                        <span>Baked by :</span> <small class="text-muted">${orderDesc.baker}</small><br>
                        <span>Event Date:</span> <small class="text-muted">${orderDesc.eventDate}</small></h4>
                        </div>
                        <div class="col-4">
                            <button class="delete-order btn btn-danger"
                            id="delete-order-${orderDesc.id}" data-order-id="${orderDesc.id}">Delete</button>
                        </div>
                    </div>
                </div>`
    }
    //We render the orders on the DOM programmatically with template literals + HTML by looping through each cake object in the order array
    static getCakeMarkupForOrder(orderDesc) {
        let cakeHtml = [];
        orderDesc.cakes.forEach(cakeDesc => {
            cakeHtml.push(`<div class="row">
                <div class="col-8">
                    <ul>
                      <li><b>Flavor:</b> ${cakeDesc.flavor}</li>
                      <li><b>Frosting:</b> ${cakeDesc.frosting}</li>
                      <li><b>No. of Layers:</b> ${cakeDesc.layers}</li>
                      <li><b>Shape:</b> ${cakeDesc.shapes}</li>
                    </ul>
                </div>
                <div class="col-4">
                    <button class="delete-cake btn btn-danger" id="delete-cake-${cakeDesc.id}"
                    data-cake-id="${cakeDesc.id}" data-order-id="${orderDesc.id}">Delete</button>
                </div>
            </div>`)
        });
        return cakeHtml.join('');
    }

    static getNewCakeForm(orderDesc) {
        return `<div class="form-group" id="new-cake-form">
```



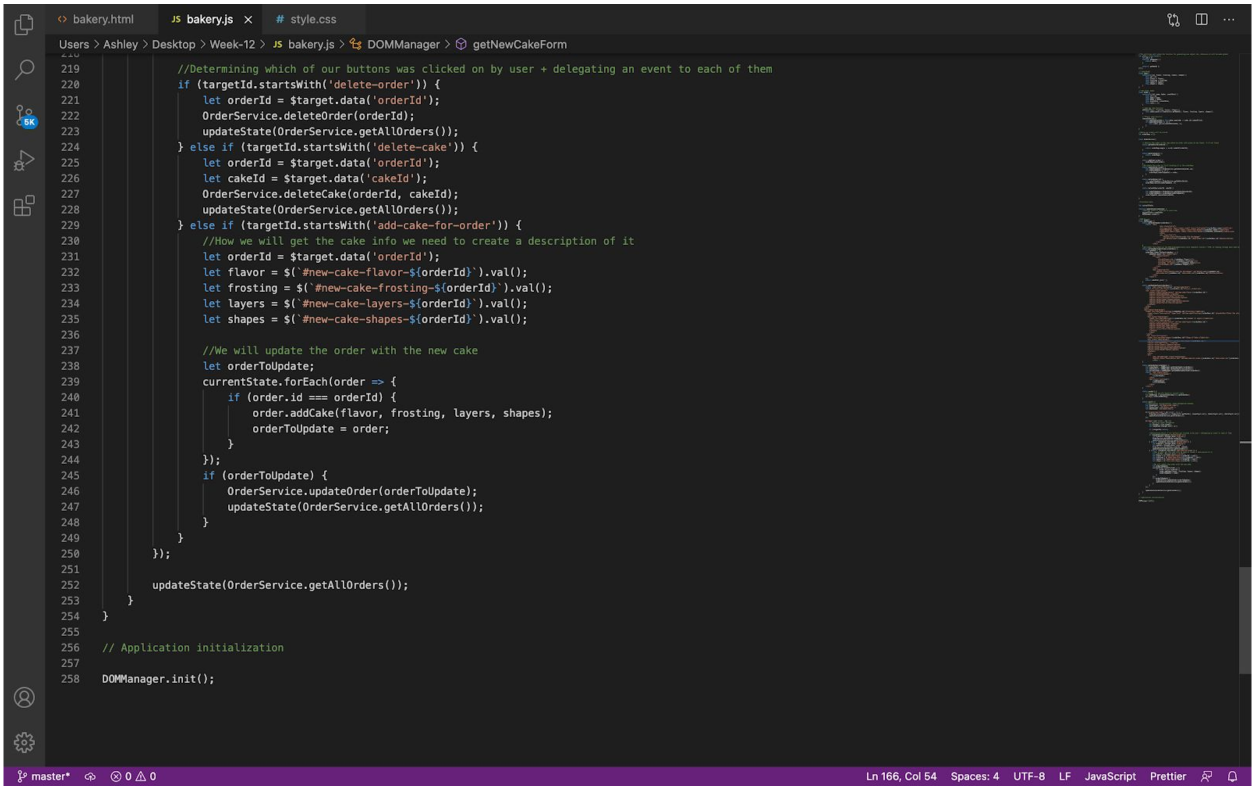
bakery.html
bakery.js
style.css
Users > Ashley > Desktop > Week-12 > bakery.js > DOMManager > getNewCakeForm
master*
0 0
Ln 166, Col 54 Spaces: 4 UTF-8 LF JavaScript Prettier

```
    static getNewCakeForm(orderDesc) {
        return `<div class="form-group" id="new-cake-form">
        <label for="new-cake-flavor-${orderDesc.id}">Flavor:</label><br>
        <div class="input-group">
            <select class="custom-select" id="new-cake-flavor-${orderDesc.id}">
            <option selected>Choose..</option>
            <option value="Vanilla">Vanilla</option>
            <option value="Chocolate">Chocolate</option>
            <option value="Lemon">Lemon</option>
            <option value="Red Velvet">Red Velvet</option>
            <option value="Carrot">Carrot</option>
          </select>
        </div>
      </div>
      <div class="form-group">
        <label for="new-cake-frosting-${orderDesc.id}">Frosting:</label><br>
          <input class="form-control" type="text" id="new-cake-frosting-${orderDesc.id}" placeholder="Enter the color for your cake">
          </div>
          <div class="form-group">
            <label for="new-cake-layers-${orderDesc.id}">Number of Layers:</label><br>
            <div class="input-group">
            <select class="custom-select" id="new-cake-layers-${orderDesc.id}">
            <option selected>Choose..</option>
            <option value="One">One</option>
            <option value="Two">Two</option>
            <option value="Three">Three</option>
            </select>
            </div>
         </div>
         <div class="form-group">
          <label for="new-cake-shapes-${orderDesc.id}">Shape of Cake:</label><br>
          <div class="input-group">
          <select class="custom-select" id="new-cake-shapes-${orderDesc.id}">
          <option selected>Choose...</option>
          <option value="Square">Square</option>
          <option value="Circle">Circle</option>
          <option value="Rectangle">Rectangle</option>
          <option value="Bundt">Bundt</option>
          </select>
          </div>
          <br>
                <div id="add-cake" class="form-group">
                <button class="form-control btn" id="add-cake-for-order-${orderDesc.id}" data-order-id="${orderDesc.id}">Add New Cake</button>
                </div>`;
    }
```

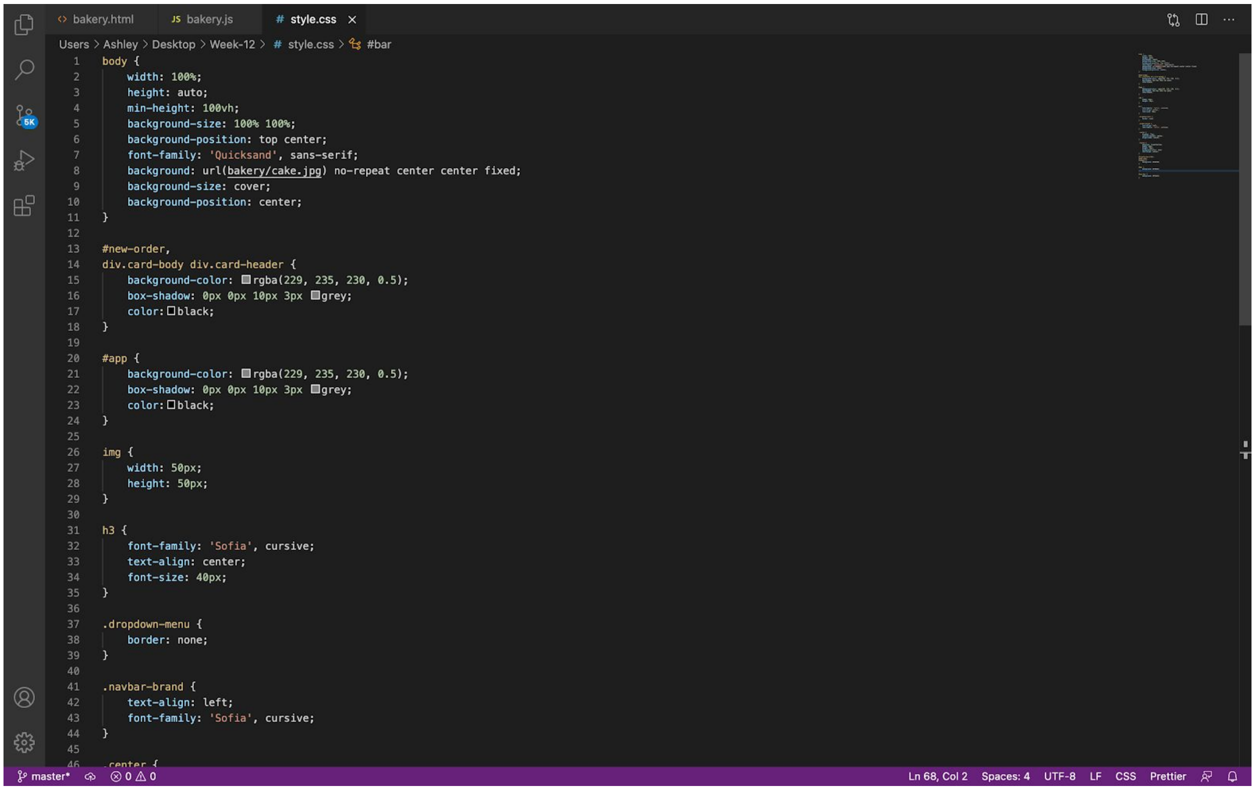


bakery.html
bakery.js
style.css
Users > Ashley > Desktop > Week-12 > bakery.js > DOMManager > getNewCakeForm
master*
0 0
Ln 166, Col 54 Spaces: 4 UTF-8 LF JavaScript Prettier

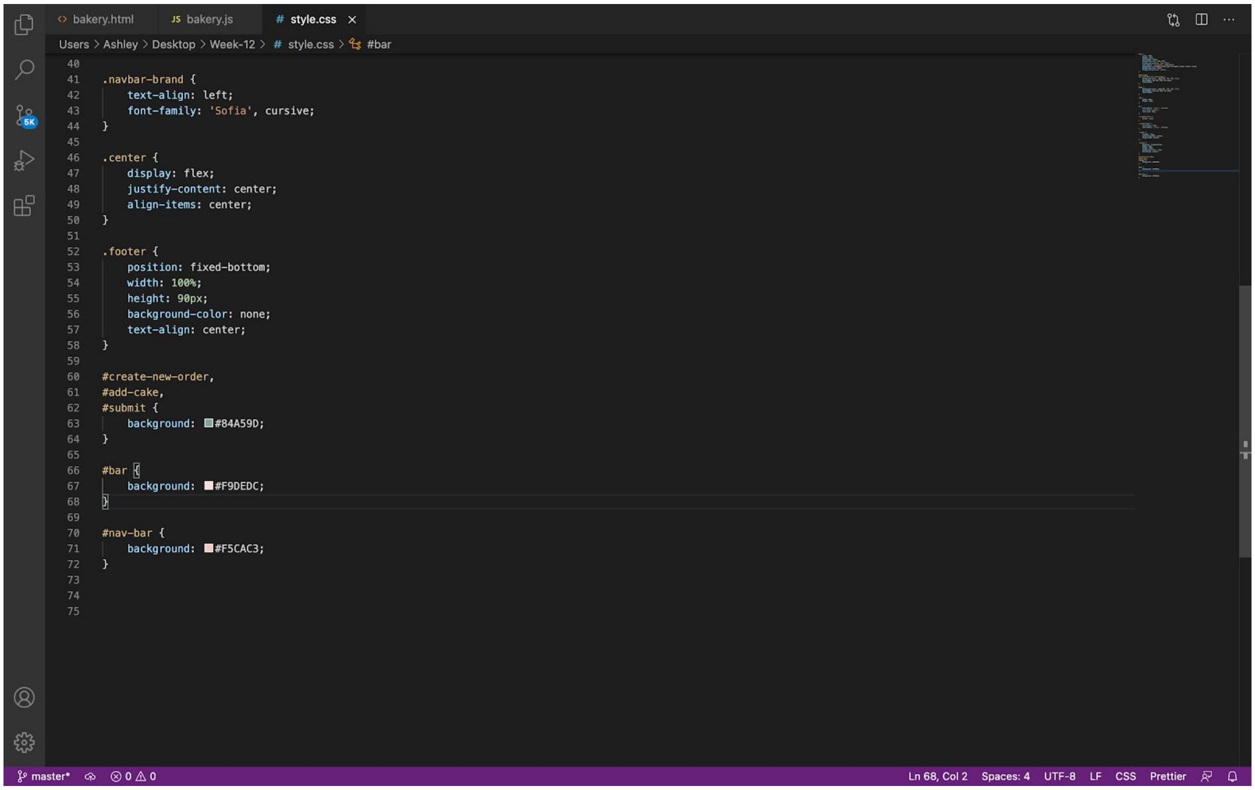
```
    static getOrderBox(orderDesc) {
        let orderHeader = DOMManager.getOrderHeader(orderDesc);
        let addCakeForm = DOMManager.getNewCakeForm(orderDesc);
        let currentCakes = DOMManager.getCakeMarkupForOrder(orderDesc);
        return `<div class="card">
            <div class="card-header">
                ${orderHeader}
            </div>
            <div class="card-body">
                ${addCakeForm}
                ${currentCakes}
            </div>
        </div>`;
    }

    static render() {
        // Render the app div based on current state
        let newMarkup = currentState.map(this.getOrderBox);
        $('#app').html(newMarkup);
    }

    static init() {
        // Application initialization, event delegation hookups
        let $nameInput = $('#new-order-name');
        let $bakerInput = $('#baker-name');
        let $dateInput = $("#date-due");

        $('#create-new-order').on('click', () => {
            OrderService.addOrder(new Order(utils.getNewId(), $nameInput.val(), $bakerInput.val(), $dateInput.val()));
            updateState(OrderService.getAllOrders());
        });

        $('#app').on('click', (e) =>{
            // employing event delegation
            let $target = $(e.target);
            let targetId= $target.attr('id');

            if (!targetId) return;

            //Determining which of our buttons was clicked on by user + delegating an event to each of them
            if (targetId.startsWith('delete-order')) {
                let orderId = $target.data('orderId');
                OrderService.deleteOrder(orderId);
                updateState(OrderService.getAllOrders());
            } else if (targetId.startsWith('delete-cake')) {
```



```javascript
            //Determining which of our buttons was clicked on by user + delegating an event to each of them
            if (targetId.startsWith('delete-order')) {
                let orderId = $target.data('orderId');
                OrderService.deleteOrder(orderId);
                updateState(OrderService.getAllOrders());
            } else if (targetId.startsWith('delete-cake')) {
                let orderId = $target.data('orderId');
                let cakeId = $target.data('cakeId');
                OrderService.deleteCake(orderId, cakeId);
                updateState(OrderService.getAllOrders());
            } else if (targetId.startsWith('add-cake-for-order')) {
                //How we will get the cake info we need to create a description of it
                let orderId = $target.data('orderId');
                let flavor = $(`#new-cake-flavor-${orderId}`).val();
                let frosting = $(`#new-cake-frosting-${orderId}`).val();
                let layers = $(`#new-cake-layers-${orderId}`).val();
                let shapes = $(`#new-cake-shapes-${orderId}`).val();

                //We will update the order with the new cake
                let orderToUpdate;
                currentState.forEach(order => {
                    if (order.id === orderId) {
                        order.addCake(flavor, frosting, layers, shapes);
                        orderToUpdate = order;
                    }
                });
                if (orderToUpdate) {
                    OrderService.updateOrder(orderToUpdate);
                    updateState(OrderService.getAllOrders());
                }
            }
        });

        updateState(OrderService.getAllOrders());
    }
}

// Application initialization

DOMManager.init();
```

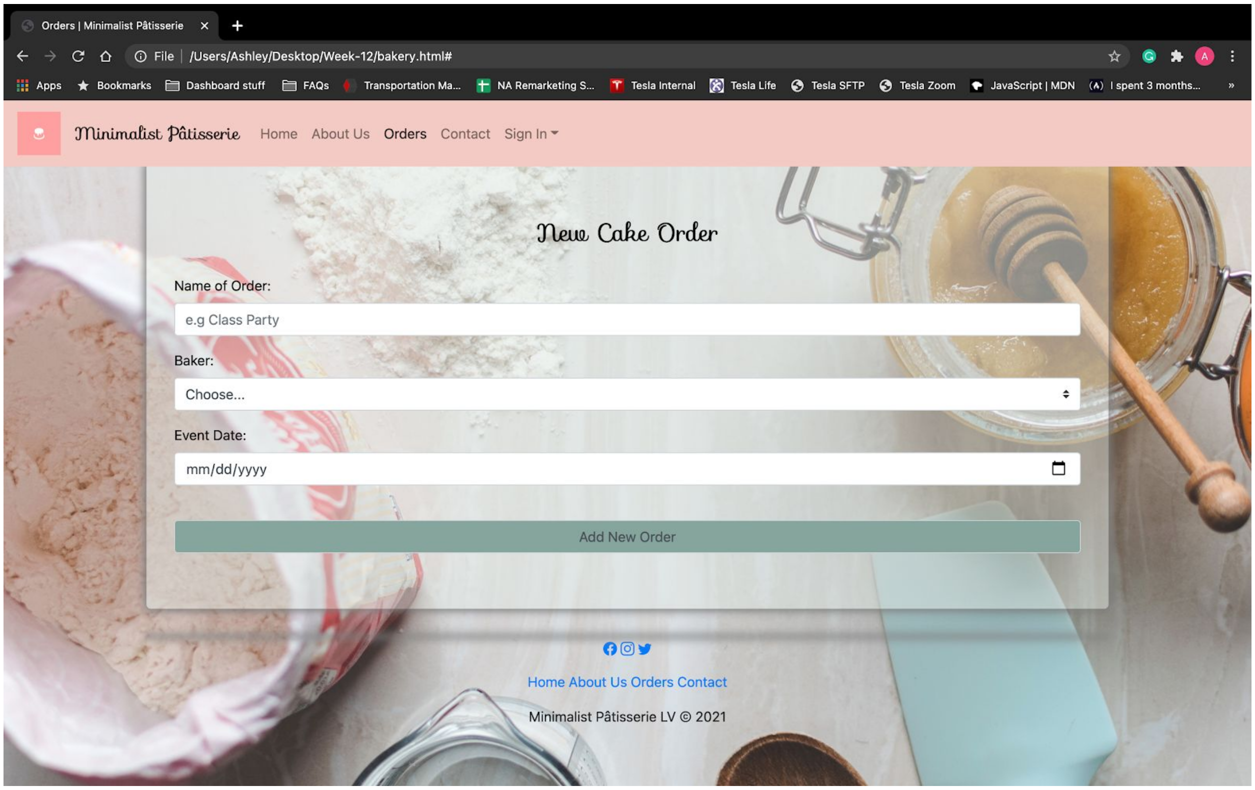


```css
body {
    width: 100%;
    height: auto;
    min-height: 100vh;
    background-size: 100% 100%;
    background-position: top center;
    font-family: 'Quicksand', sans-serif;
    background: url(bakery/cake.jpg) no-repeat center center fixed;
    background-size: cover;
    background-position: center;
}

#new-order,
div.card-body div.card-header {
    background-color: rgba(229, 235, 230, 0.5);
    box-shadow: 0px 0px 10px 3px grey;
    color: black;
}

#app {
    background-color: rgba(229, 235, 230, 0.5);
    box-shadow: 0px 0px 10px 3px grey;
    color: black;
}

img {
    width: 50px;
    height: 50px;
}

h3 {
    font-family: 'Sofia', cursive;
    text-align: center;
    font-size: 40px;
}

.dropdown-menu {
    border: none;
}

.navbar-brand {
    text-align: left;
    font-family: 'Sofia', cursive;
}

.center {
```


bakery.html
bakery.js
style.css
Users > Ashley > Desktop > Week-12 > style.css > #bar
master*
0 0
Ln 68, Col 2 Spaces: 4 UTF-8 LF CSS Prettier

```css

.navbar-brand {
    text-align: left;
    font-family: 'Sofia', cursive;
}

.center {
    display: flex;
    justify-content: center;
    align-items: center;
}

.footer {
    position: fixed-bottom;
    width: 100%;
    height: 90px;
    background-color: none;
    text-align: center;
}

#create-new-order,
#add-cake,
#submit {
    background: #84A59D;
}

#bar {
    background: #F9DEDC;
}

#nav-bar {
    background: #F5CAC3;
}
```

Screenshots of Running Application:


Orders | Minimalist Pâtisserie
File | /Users/Ashley/Desktop/Week-12/bakery.html#
Apps Bookmarks Dashboard stuff FAQs Transportation Ma... NA Remarketing S... Tesla Internal Tesla Life Tesla SFTP Tesla Zoom JavaScript | MDN I spent 3 months...
Minimalist Pâtisserie
Home
About Us
Orders
Contact
Sign In

New Cake Order

Name of Order:

e.g Class Party

Baker:

Choose...

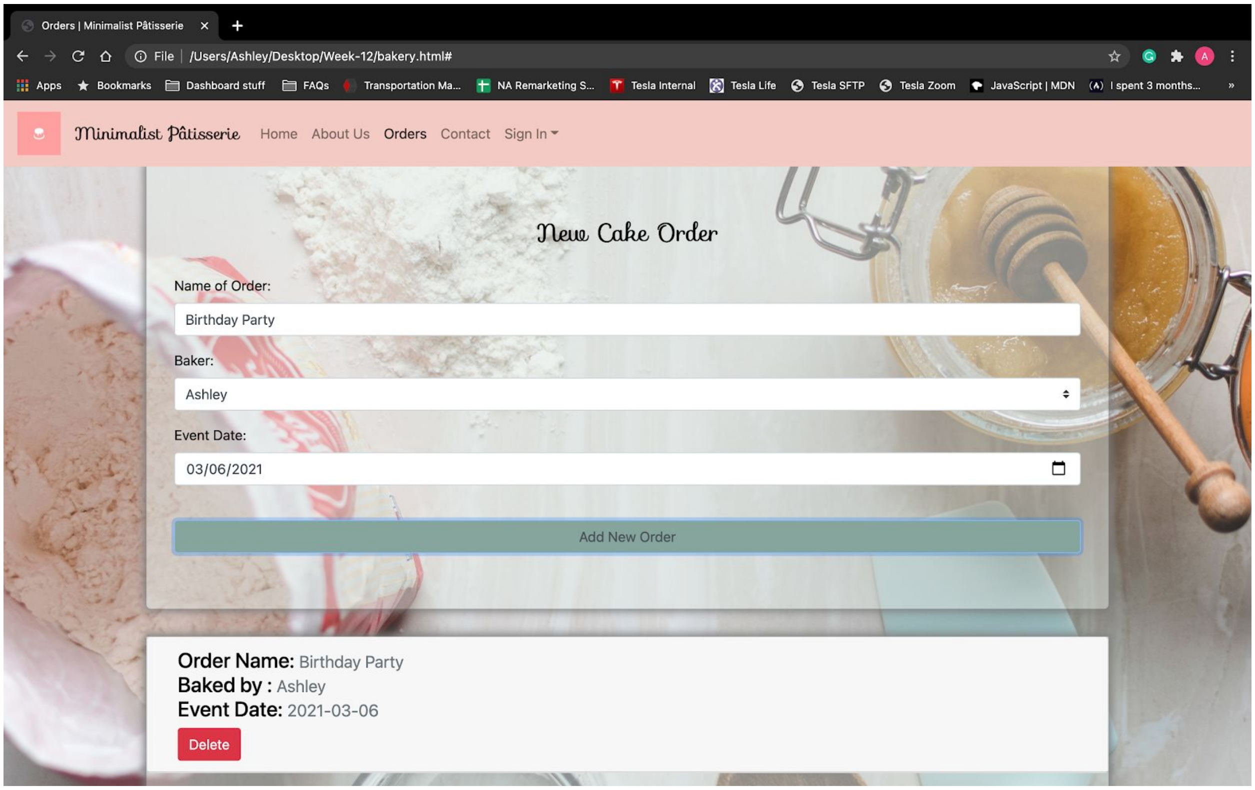
Event Date:

mm/dd/yyyy

Add New Order

Home About Us Orders Contact

Minimalist Pâtisserie LV © 2021



New Cake Order

Name of Order: Birthday Party

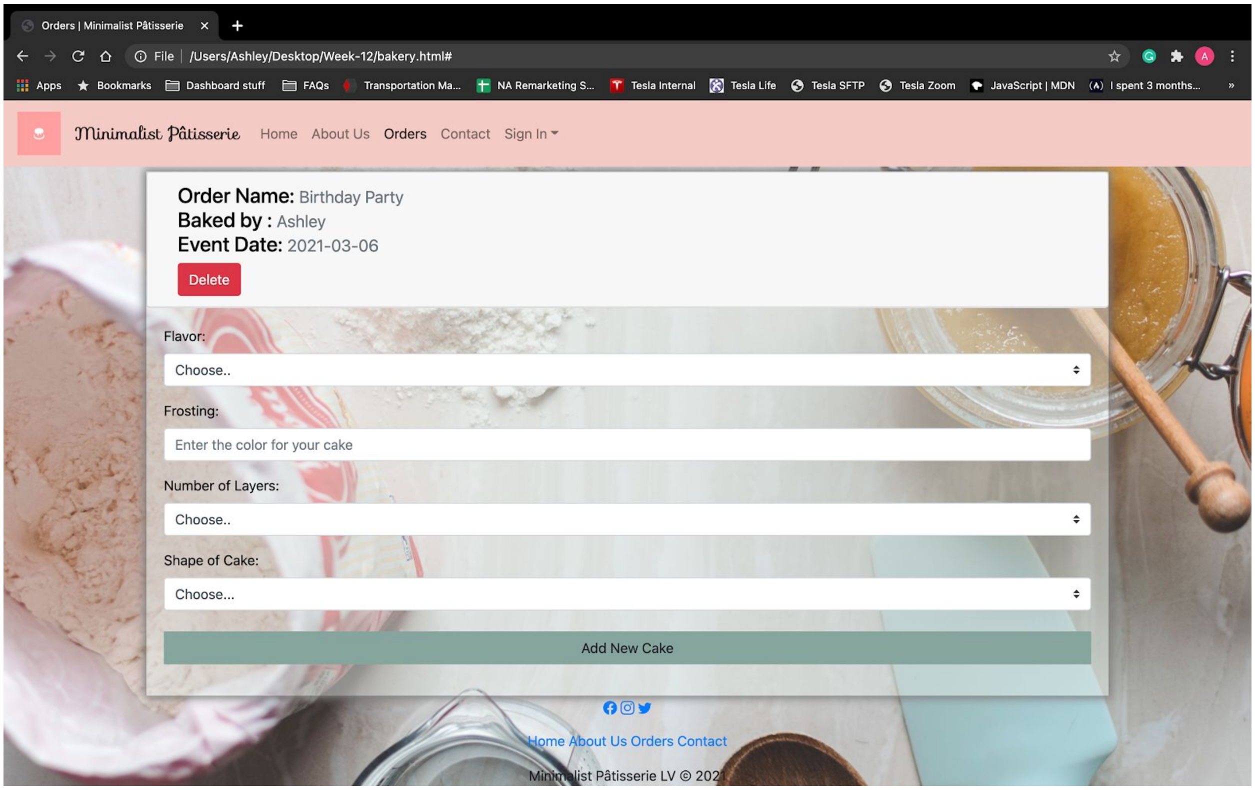
Baker: Ashley

Event Date: 03/06/2021

Add New Order

Order Name: Birthday Party
Baked by : Ashley
Event Date: 2021-03-06

Delete



Order Name: Birthday Party
Baked by : Ashley
Event Date: 2021-03-06

Delete

Flavor: Choose..

Frosting: Enter the color for your cake

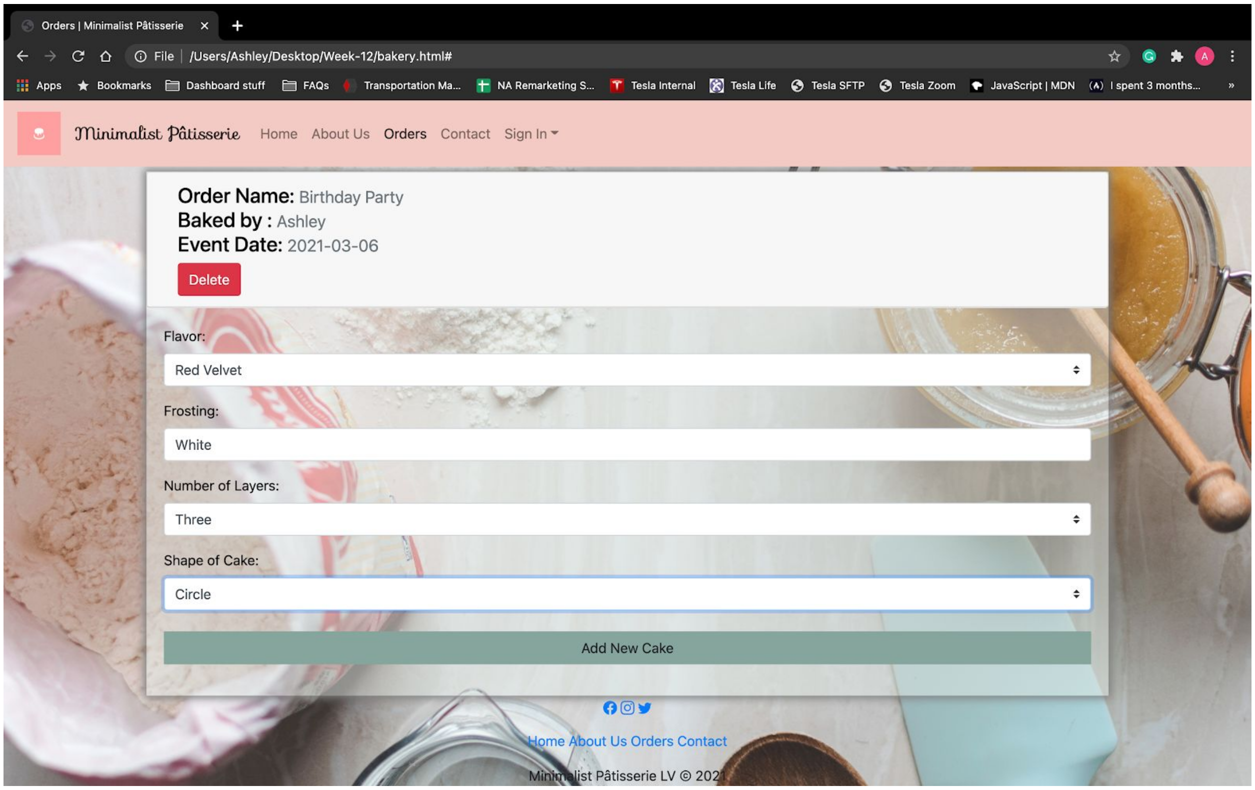
Number of Layers: Choose..

Shape of Cake: Choose...

Add New Cake

Home About Us Orders Contact

Minimalist Pâtisserie LV © 2021



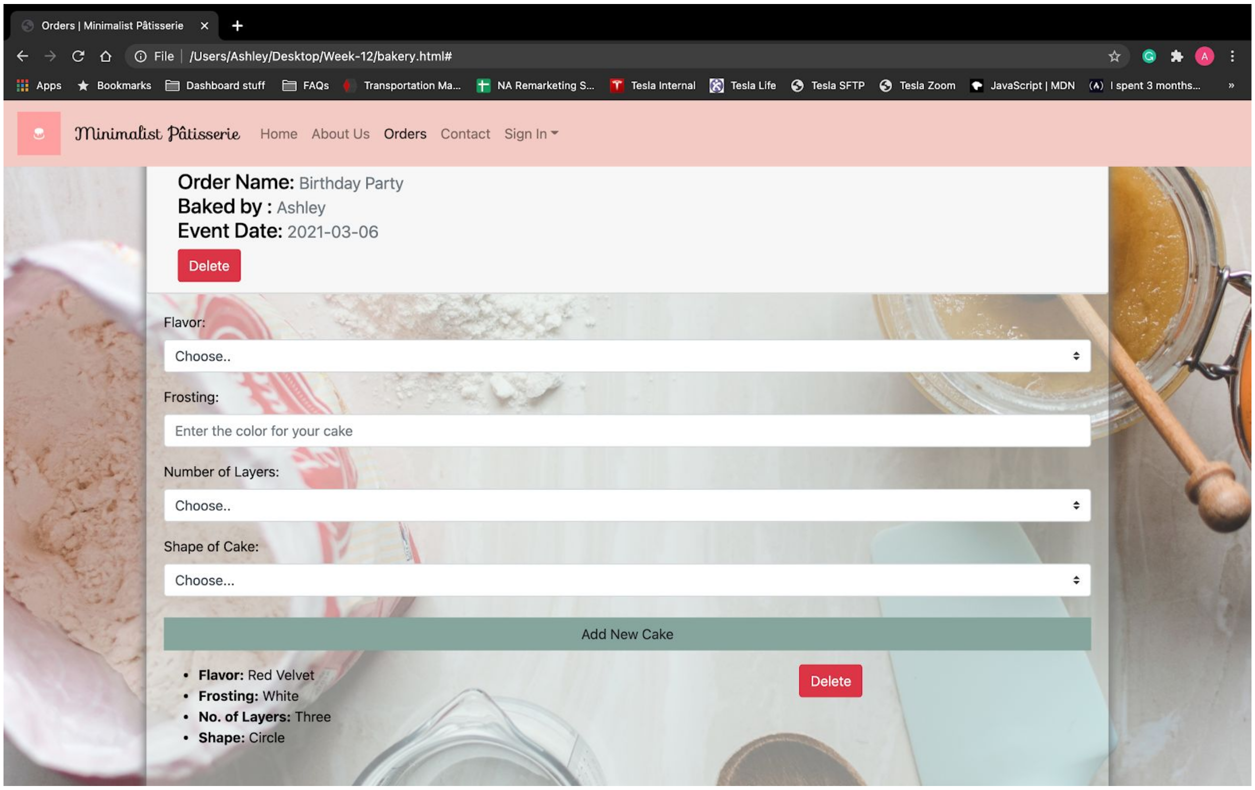
Order Name: Birthday Party
Baked by : Ashley
Event Date: 2021-03-06

Flavor: Red Velvet

Frosting: White

Number of Layers: Three

Shape of Cake: Circle



Order Name: Birthday Party
Baked by : Ashley
Event Date: 2021-03-06

Flavor: Choose..

Frosting: Enter the color for your cake

Number of Layers: Choose..

Shape of Cake: Choose...

- **Flavor:** Red Velvet
- **Frosting:** White
- **No. of Layers:** Three
- **Shape:** Circle