

IMG 111: Final Project Report by Natalie Evans

URL: http://nevans.imgd.ca/IMG_111/project/dinner.html

username: Welcome **password:** IMG_111

It took me 20 hour(s) to complete the assignment. I have learned and feel good about 90 % of the skills listed in the Chapter Summary. From the assignment work I learnt the following:

Using local storage, working with event listeners, connecting JavaScript with CSS to create animations and transitions, working with the DOM selecting and manipulating elements inside it. Working with the BOM, to create a scroll to top function. I also did a lot of work with user-defined and pre-defined functions (like `setTimeout()`), and loops including a switch loop, and using a cool attribution method called toggle. I also learnt the difference between a node list and an HTML collection.

The difficulties I encountered with this assignment:

I made a lot of silly basic and common mistakes that lead to a lot of wasted time trouble shooting, I really should have taken my time and done everything step by step instead of rushing through trying to get everything done.

I had some difficulty trying to get a cookie to work, so I ended up learning about local storage instead.

And for the longest time my for loops were throwing an error saying that an element was undefined, but I realized that I was using `<=` with the length function on a node list where the index starts at 0, so it was looking for an extra element when it didn't exist.

I definitely struggled with putting the UI together, making sure everything was working properly and I was changing the right things and iterating through the tables properly.

The sections / topics that I would like to review further:

Other comments and suggestions:

About My Site:

So my site is a webpage with 2 pages one displaying a dinner menu, and a drink menu. Both pages have a menu filter, so users can select what kind of drinks they are looking for or if they have any dietary restrictions.

If a user were to click on a menu item, it would highlight it with a different background color, and the highlight will disappear when they move the mouse off of the element.

All images on the site are also interactive if the user clicks on them it will fade out and fade in with a different image. If the user scrolls past 800px a scroll to top bar will appear at the bottom of the window, and when clicked it will scroll to the top and disappear, and will also disappear if the user scrolls up past the 800px mark again.

When landing on a page if the user has not entered their name in the prompt before, it will prompt them and then store that information in local storage to display in the navigation bar with a welcome message, if the user's name is equal to null it will display them as a guest.

JavaScript user interaction features: selectable menu items, clickable images, and a scroll to top button.

JavaScript user interface feature: filters that allow you to customize the menu to your needs/wants.

Cookie/Local Storage: prompt and store user's name to display in navigation bar.

How I did it:

Selectable menu items: I selected all TR elements in my page and through a for loop I added event listeners to them. The function activated by the event listener added a class name to the element clicked on and in my style sheet there was matching css styling for that class that changed the background color and the color of the font. I then added another event listener to the element so that when the mouse left the element it would remove the class.

Clickable images: I put all of my images into multiple arrays separated out by dimensions. I then selected all my existing IMG tags in my HTML, added an event listener to it using a for loop again and the function named in the event listener contained a setTimeout function, with another function that would be called when the first fade out happened. When the second function was called it would fade the image out by applying a class name, then it would change

the source attribute to a different random image, but picking a random one out of the array. Then I removed the class name so the image would transition back in again.

Scroll to top: I had to do some research into BOM, and how to know when a user has scrolled in the window. I ended up using a lot of methods to select the window and using an if statement if they scrolled past a certain point it would display the button and take it away when they scrolled back up again or pressed the button to go up again. I added an event listener to a button in the element that when clicked it would scroll to the top of the page using the scrollTo method.

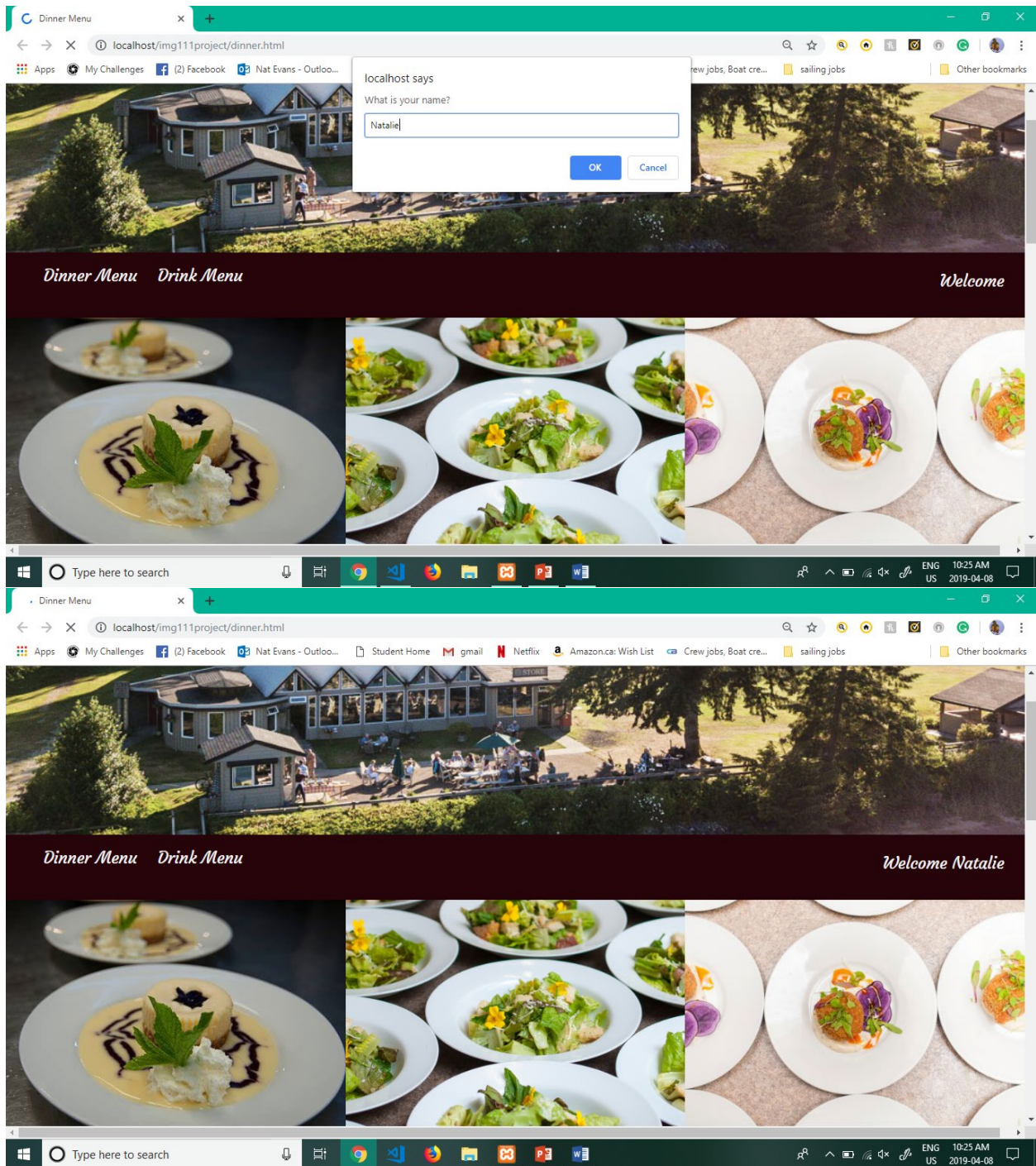
Customized your menu buttons: I selected a buttons on my page via getElementById and added event listener. I then created a function that contains a switch loop that looks at the event target and changes the table by looking at content attribute and if it doesn't contain the key word that its looking for it will apply a CSS rule to hide it, and I put the class list thing onto a toggle so you could go back and forth.

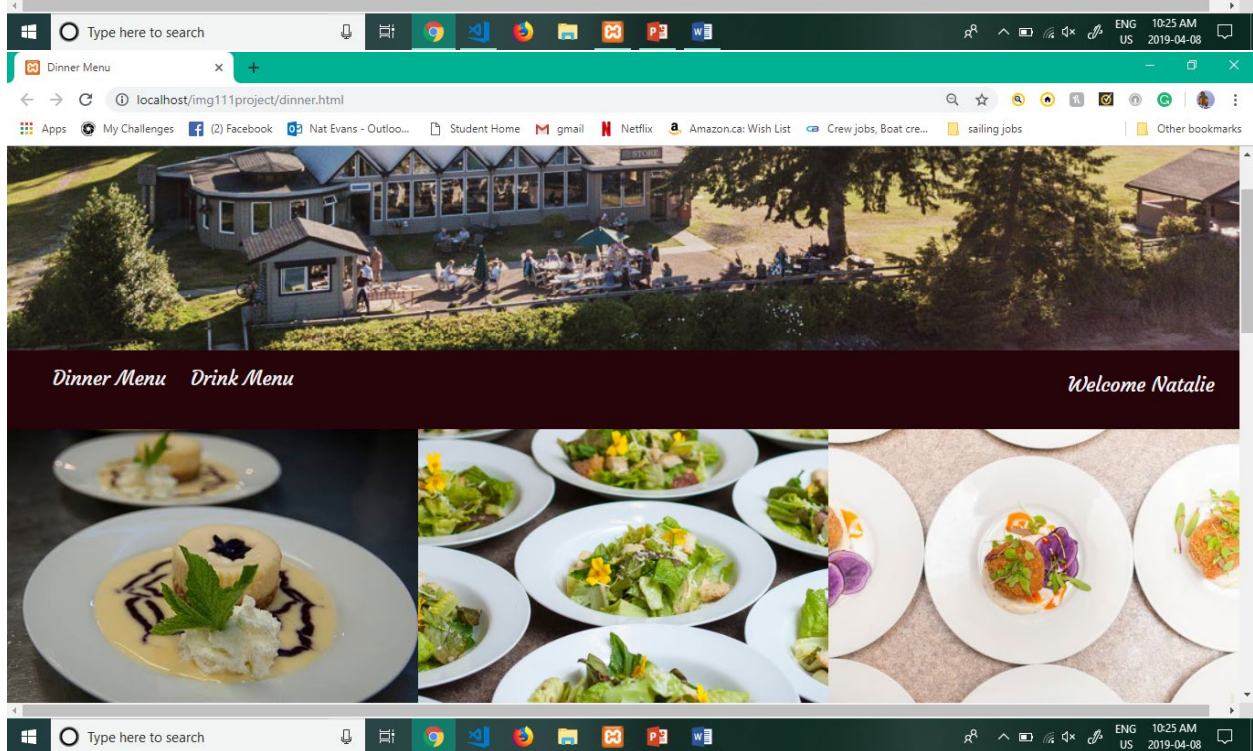
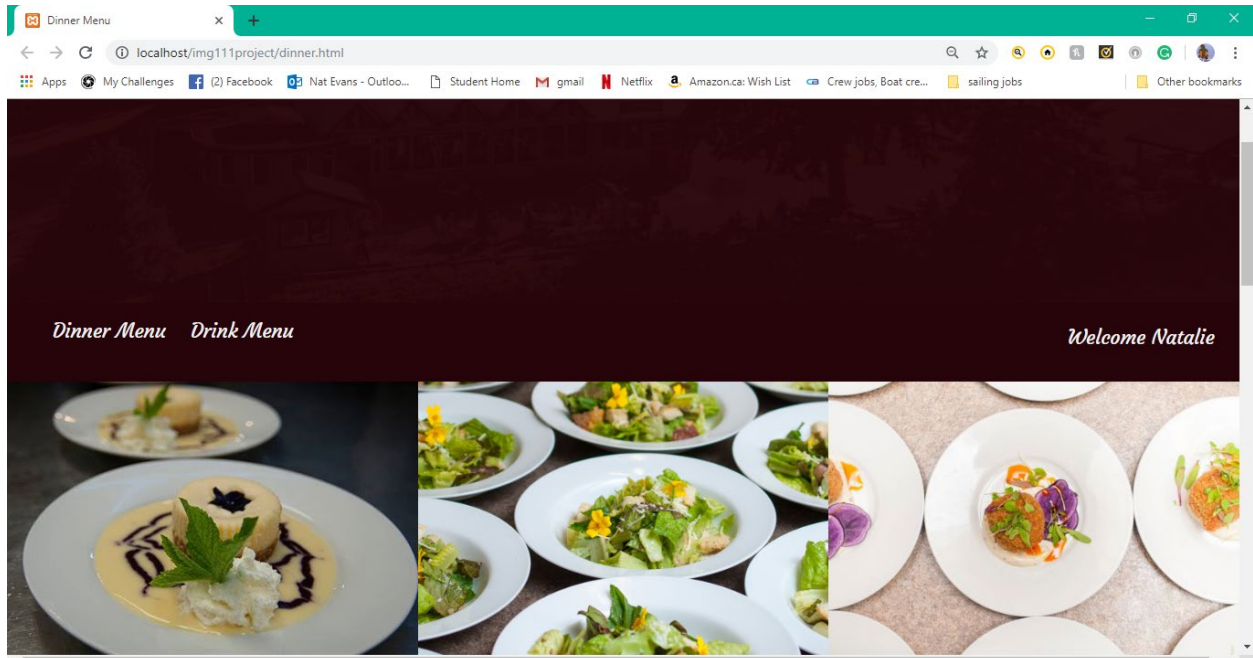
Local storage: so using an if statement I checked to see if local storage with a key username is null or not, if its null meaning that no key exists yet it will prompt you and then create local storage and assign your input to the key. If the key is set then it will display the user's name but if the user pressed cancel and didn't enter anything in it will say they are a guest.

Errors:

There are no errors in the JavaScript shown in the browser debugger, not in Visual Studio Code platform itself. There were some errors in my html but they didn't affect usability, and I could pinpoint what they were triggered by, it had something to do with my tables but looking at the html document gave me not insights on what I messed up in the HTML structure. I figured because they didn't affect usability or site structure that it could be a future issue for me to solve if this site were to ever go live. There was no errors in my CSS.

Screen Shots:





Dinner Menu

localhost/img111project/dinner.html

Apps My Challenges (2) Facebook Nat Evans - Outloo... Student Home gmail Netflix Amazon.ca: Wish List Crew jobs, Boat cre... sailing jobs Other bookmarks

Cedar Post's Dinner Menu

Appetizers

Market Salad	Island kale and greens, croutons, carrot ribbons, fresh jarlsberg, preserved lemon, creamy parsley.	16
Summer Caesar	Baby greens, island kale, roasted garlic, house croutons, fresh grated parmesan, creamy preserved lemon and garlic dressing	16
BC Spot Prawns	Garlic butter, island cherry tomatoes, and fresh island herbs	22
Seared Scallops and Braised Bacon	Garden micro greens, burnt orange glaze	24
Fresh Cortez Clams	Coconut-chili broth, cilantro pesto, baby bok choy, and grilled lime.	26
Saltspring Island Mussels	Coconut red curry broth, coriander roasted tomatoes, and cilantro oil.	26

To Top

Dinner Menu

localhost/img111project/dinner.html

Apps My Challenges (2) Facebook Nat Evans - Outloo... Student Home gmail Netflix Amazon.ca: Wish List Crew jobs, Boat cre... sailing jobs Other bookmarks

Cedar Post's Dinner Menu

Appetizers

Market Salad	Island kale and greens, croutons, carrot ribbons, fresh jarlsberg, preserved lemon, creamy parsley.	16
Summer Caesar	Baby greens, island kale, roasted garlic, house croutons, fresh grated parmesan, creamy preserved lemon and garlic dressing	16
BC Spot Prawns	Garlic butter, island cherry tomatoes, and fresh island herbs	22
Seared Scallops and Braised Bacon	Garden micro greens, burnt orange glaze	24
Fresh Cortez Clams	Coconut-chili broth, cilantro pesto, baby bok choy, and grilled lime.	26
Saltspring Island Mussels	Coconut red curry broth, coriander roasted tomatoes, and cilantro oil.	26

To Top

Dinner Menu

localhost/img111project/dinner.html

Apps My Challenges (2) Facebook Nat Evans - Outloo... Student Home gmail Netflix Amazon.ca: Wish List Crew jobs, Boat cre... sailing jobs Other bookmarks

Vegetarian Vegan Dairy Free Gluten Free

Cedar Post's Dinner Menu

Appetizers

Market Salad	Island kale and greens, croutons, carrot ribbons, fresh jarlsberg, preserved lemon, creamy parsley.	16
Summer Caesar	Baby greens, island kale, roasted garlic, house croutons, fresh grated parmesan, creamy preserved lemon and garlic dressing	16

Entrees

Samler Schnitzel	Crispy eggplant schnitzel, butter roasted wild mushrooms, spätzle, red cabbage.	29
------------------	---	----

Drink Menu

localhost/img111project/wine.html

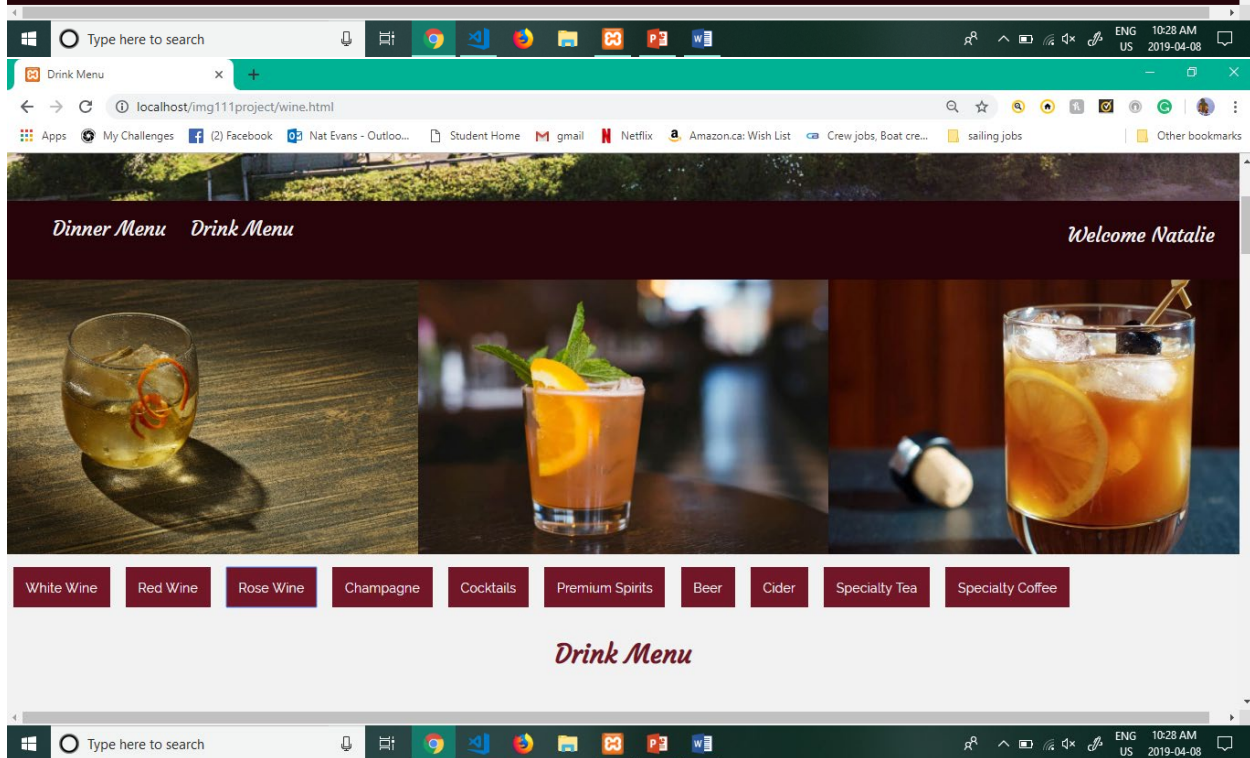
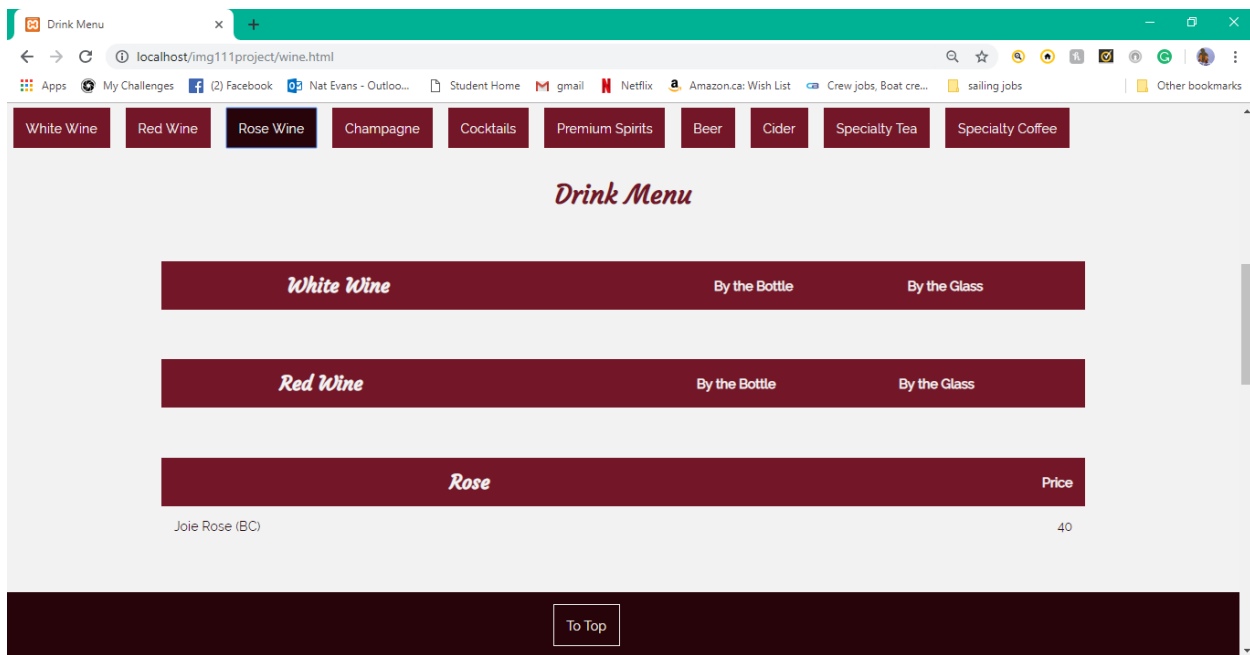
Apps My Challenges (2) Facebook Nat Evans - Outloo... Student Home gmail Netflix Amazon.ca: Wish List Crew jobs, Boat cre... sailing jobs Other bookmarks

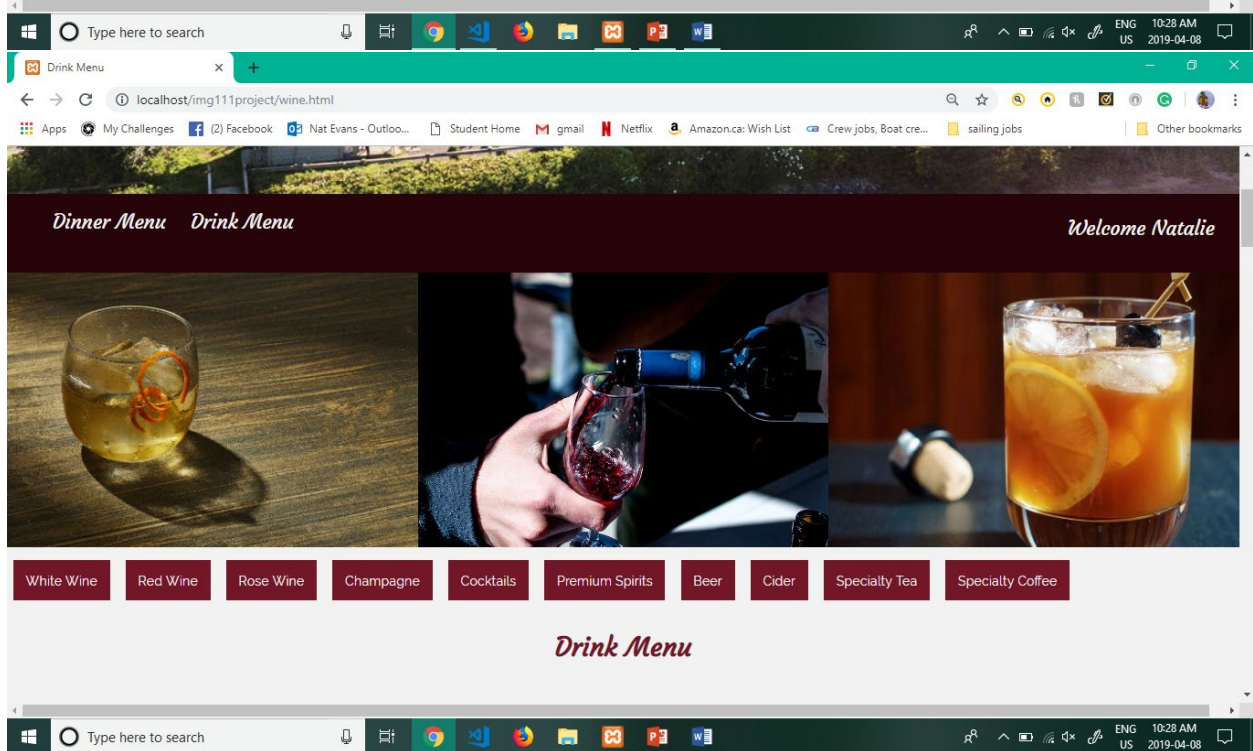
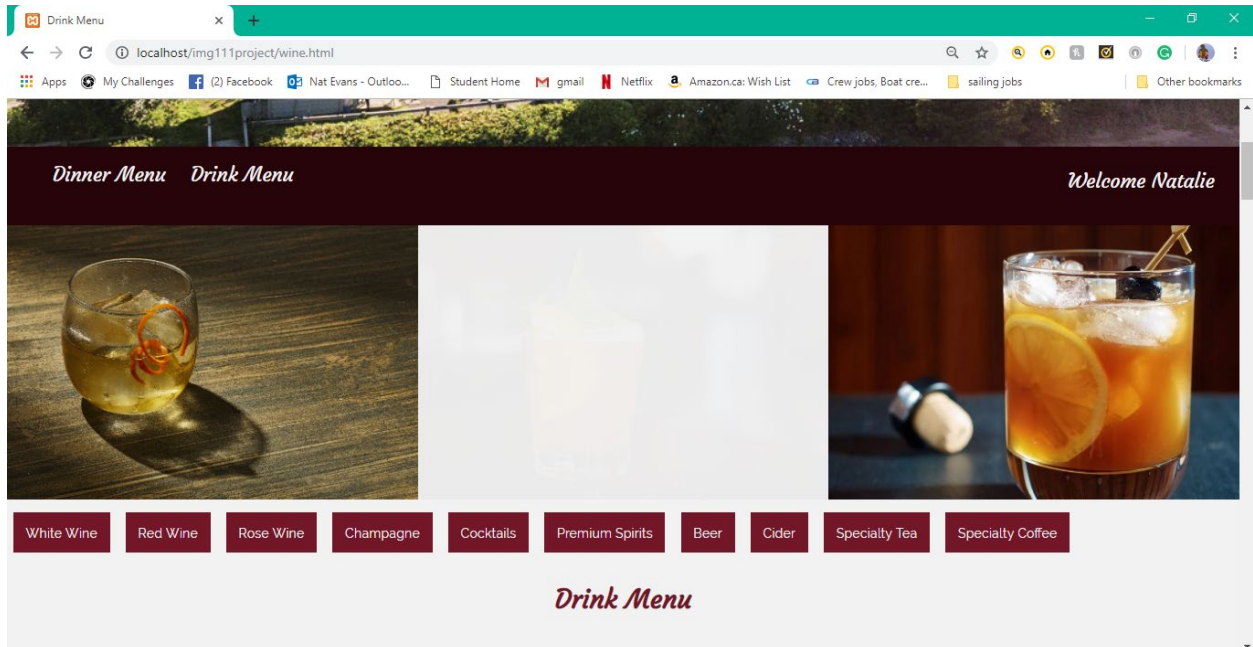


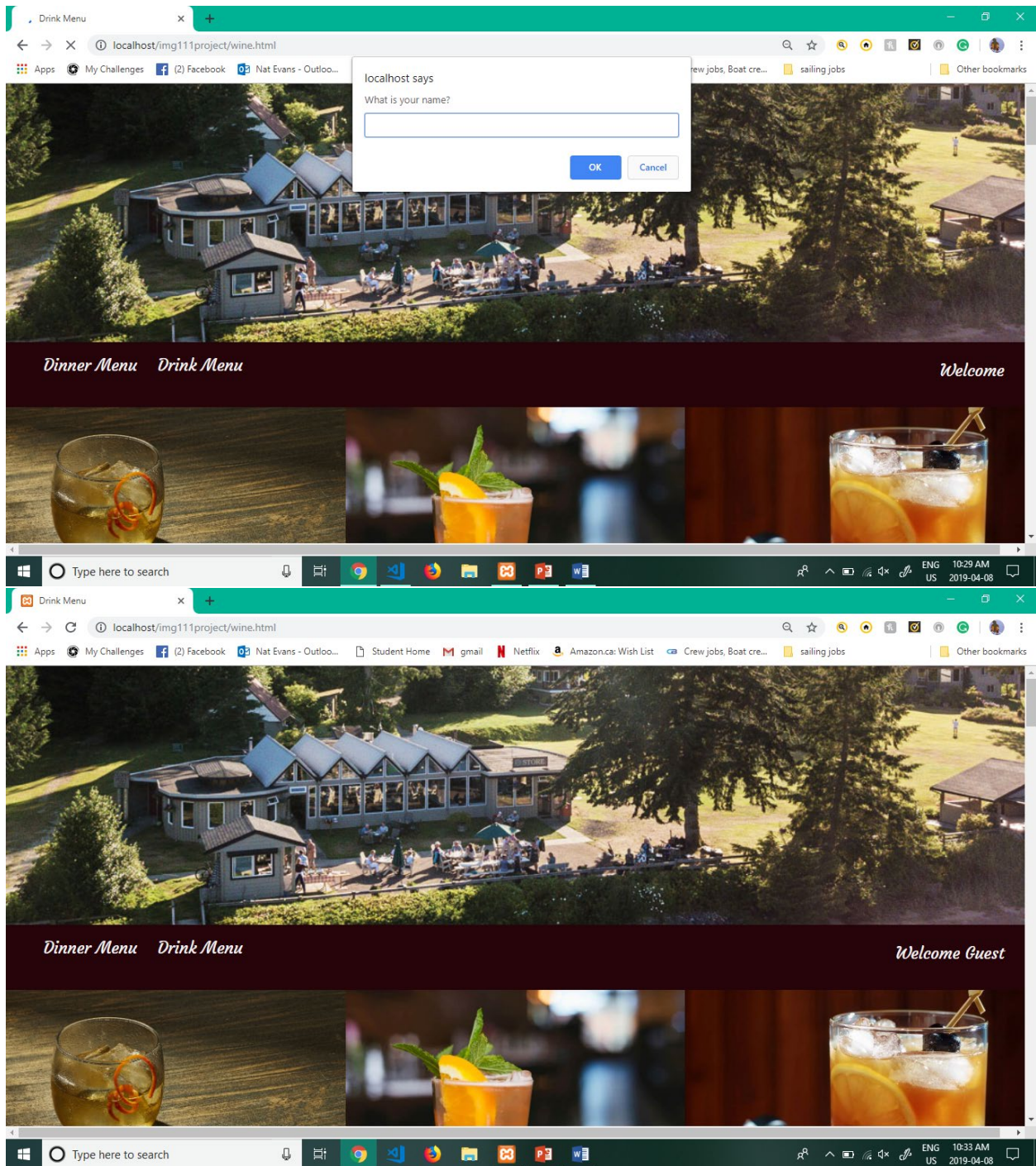
Dinner Menu Drink Menu

Welcome Natalie









Javascript Code:

```
/*  
page: cookie.js  
Make and store value in local storage, create a scroll to top function  
Author:Natalie Evans Date:2019-04-06*/
```

```

"use strict";
//scroll
window.onscroll = function() {scrollUp()};

function scrollUp() {
  if (document.body.scrollTop > 800 || document.documentElement.scrollTop > 800)
  {
    document.querySelector("address").classList.remove("display");
  } else{
    document.querySelector("address").classList.add("display");
  }
}

let toTop = document.getElementById("totop");
toTop.addEventListener("click", goTop);

function goTop(e){
  window.scrollTo(0,0);
  document.querySelector("address").classList.add("display");
}

//Local Storage
if (localStorage.getItem("username") == null) {
  let username = prompt("What is your name?");
  localStorage.setItem("username", username);

  if(username == null){
    document.getElementById("greeting").innerHTML = ` Welcome Guest`;
  } else {
    document.getElementById("greeting").innerHTML = ` Welcome ${username}`;
  }
}else{

  if(localStorage.getItem("username") == "null"){
    document.getElementById("greeting").innerHTML = ` Welcome Guest`;
  } else {
    document.getElementById("greeting").innerHTML = ` Welcome
${localStorage.getItem("username")}`;
  }

}
}

```

```

/*
page:wine.js
create a drink menu with filters, if they click change row color, clickable image
gallery.
Author: Natalie Evans Date: 2019-04-01
*/
"use strict";

//Fade images in and out on click
let drinkImages = [
    `redwine`,
    `whitewine`,
    `beer`,
    `mixeddrink`,
    `mixeddrink2`,
    `mixeddrink3`,
    `tea`,
    `coffee`,
    `coffee2`
];

let resortImages = [
    `resturant2`,
    `relaxing`,
    `marina`
];

let img = document.querySelector("img");
img.addEventListener("click", imgClicked);

let imgRest = document.querySelectorAll("img");

for(let i = 1 ; i < imgRest.length; i++){
    imgRest[i].addEventListener("click", imgRestClicked);
}

function imgClicked(e){
    e.target.classList.add("hide");
    setTimeout(showResortImages, 2000, e);
}

function showResortImages(e){
    let i = Math.floor(Math.random() * resortImages.length);
    let srcString = `images/${resortImages[i]}.jpg`;

```

```

e.target.src = srcString;
e.target.classList.remove("hide");
}

function imgRestClicked(e){
    e.target.classList.add("hide");
    setTimeout(showDrinkImages, 2000, e);
}

function showDrinkImages(e){
    let i = Math.floor(Math.random() * drinkImages.length);
    let srcString = `images/${drinkImages[i]}.jpg`;
    e.target.src = srcString;
    e.target.classList.remove("hide");
}

//create drink filter
let redWine = document.getElementById("redWine");
redWine.addEventListener("click", filterW);

let whiteWine = document.getElementById("whiteWine");
whiteWine.addEventListener("click", filterW);

let roseWine = document.getElementById("roseWine");
roseWine.addEventListener("click", filterW);

let champagne = document.getElementById("champagne");
champagne.addEventListener("click", filterW);

let cocktails = document.getElementById("cocktail");
cocktails.addEventListener("click", filterW);

let premiumSpirits = document.getElementById("premium");
premiumSpirits.addEventListener("click", filterW);

let beer = document.getElementById("beer");
beer.addEventListener("click", filterW);

let cider = document.getElementById("cider");
cider.addEventListener("click", filterW);

let tea = document.getElementById("tea");
tea.addEventListener("click", filterW);

```

```

let coffee = document.getElementById("coffee");
coffee.addEventListener("click", filterW);

function filterW(e){
let trFilter = document.querySelectorAll("tr");

switch(e.target){
  case redWine:
    e.target.classList.toggle("selected");
    for(let i = 0; i < trFilter.length; i++){
      if( !trFilter[i].getAttribute("content").includes("red")){
        trFilter[i].classList.toggle("display");
      }
    }
    break;

  case whiteWine:
    e.target.classList.toggle("selected");
    for(let i = 0; i < trFilter.length; i++){

      if( !trFilter[i].getAttribute("content").includes("white")){
        trFilter[i].classList.toggle("display");
      }
    }
    break;

  case roseWine:
    e.target.classList.toggle("selected");
    for(let i = 0; i < trFilter.length; i++){

      if( !trFilter[i].getAttribute("content").includes("rose")){
        trFilter[i].classList.toggle("display");
      }
    }
    break;

  case champagne:
    e.target.classList.toggle("selected");
    for(let i = 0; i < trFilter.length; i++){

      if( !trFilter[i].getAttribute("content").includes("champagne")){
        trFilter[i].classList.toggle("display");
      }
    }

```

```

    }
    break;

    case cocktails:
e.target.classList.toggle("selected");
    for(let i = 0; i < trFilter.length; i++){

        if( !trFilter[i].getAttribute("content").includes("cocktail")){
            trFilter[i].classList.toggle("display");
        }
    }
    break;

    case premiumSpirits:
e.target.classList.toggle("selected");
    for(let i = 0; i < trFilter.length; i++){

        if( !trFilter[i].getAttribute("content").includes("premium")){
            trFilter[i].classList.toggle("display");
        }
    }
    break;

    case beer:
e.target.classList.toggle("selected");
    for(let i = 0; i < trFilter.length; i++){

        if( !trFilter[i].getAttribute("content").includes("beer")){
            trFilter[i].classList.toggle("display");
        }
    }
    break;

    case cider:
e.target.classList.toggle("selected");
    for(let i = 0; i < trFilter.length; i++){

        if( !trFilter[i].getAttribute("content").includes("cider")){
            trFilter[i].classList.toggle("display");
        }
    }
    break;

    case tea:
e.target.classList.toggle("selected");

```

```

        for(let i = 0; i < trFilter.length; i++){

            if( !trFilter[i].getAttribute("content").includes("tea")){
                trFilter[i].classList.toggle("display");
            }
        }
        break;

        case coffee:
e.target.classList.toggle("selected");
        for(let i = 0; i < trFilter.length; i++){

            if( !trFilter[i].getAttribute("content").includes("coffee")){
                trFilter[i].classList.toggle("display");
            }
        }
        break;
    }

}

}

//create row highlight on click
let tr = document.getElementsByTagName("tr");

for(let i = 0; i < tr.length; i++){
    tr[i].addEventListener("click", underLine);
    tr[i].addEventListener("mouseleave", undo);
    function underLine(){
        tr[i].classList.add("Line");
    }

    function undo(){
        tr[i].classList.remove("Line");
    }
}

}

```

/*

page: dinner.js

create a dinner menu with filters, if they click change row color, clickable image gallery.

Author:Natalie Evans Date: 2019-04-02

```

*/

"use strict";

//fade images in and out on click
let dinnerImages = [
  `caesar`,
  `salad`,
  `beets`,
  `oysters`,
  `shrimp`,
  `seafood`,
  `brownie`,
  `desert1`,
  `icecream`,
];

let resortImages = [
  `restaurant2`,
  `relaxing`,
  `marina`
];

let img = document.querySelector("img");
img.addEventListener("click", imgClicked);

let imgRest = document.querySelectorAll("img");

for(let i = 1 ; i < imgRest.length; i++){
  imgRest[i].addEventListener("click", imgRestClicked);
}

function imgClicked(e){
  e.target.classList.add("hide");
  setTimeout(showResortImages, 2000, e);
}

function showResortImages(e){
  let i = Math.floor(Math.random() * resortImages.length);
  let srcString = `images/${resortImages[i]}.jpg`;
  e.target.src = srcString;
  e.target.classList.remove("hide");
}

```

```

function imgRestClicked(e){
    e.target.classList.add("hide");
    setTimeout(showDinnerImages, 2000, e);
}

function showDinnerImages(e){
    let i = Math.floor(Math.random() * dinnerImages.length);
    let srcString = `images/${dinnerImages[i]}.jpg`;
    e.target.src = srcString;
    e.target.classList.remove("hide");
}

//create food filter based on common dietary needs
let vegetarian = document.getElementById("vegetarian");
vegetarian.addEventListener("click", filter);

let vegan = document.getElementById("vegan");
vegan.addEventListener("click", filter);

let dairyFree = document.getElementById("dairy-free");
dairyFree.addEventListener("click", filter);

let glutenFree = document.getElementById("gluten-free");
glutenFree.addEventListener("click", filter);

function filter(e){

let trFilter = document.querySelectorAll("tr");

switch(e.target){
    case vegetarian:
        e.target.classList.toggle("selected");
        for(let i = 0; i < trFilter.length; i++){

            if( !trFilter[i].getAttribute("content").includes("vegetarian")){
                trFilter[i].classList.toggle("display");
            }
        }
        break;

        case vegan:
            e.target.classList.toggle("selected");

```

```

        for(let i = 0; i < trFilter.length; i++){

            if( !trFilter[i].getAttribute("content").includes("vegan")){
                trFilter[i].classList.toggle("display");
            }
        }
        break;

        case dairyFree:
            e.target.classList.toggle("selected");
            for(let i = 0; i < trFilter.length; i++){

                if( !trFilter[i].getAttribute("content").includes("dairy-free")){
                    trFilter[i].classList.toggle("display");
                }
            }
            break;

        case glutenFree:
            e.target.classList.toggle("selected");
            for(let i = 0; i < trFilter.length; i++){

                if( !trFilter[i].getAttribute("content").includes("gluten-free")){
                    trFilter[i].classList.toggle("display");
                }
            }
            break;
    }

}

}

//highlight row on click disappear on mouse leaving
let tr = document.getElementsByTagName("tr");

for(let i = 0; i < tr.length; i++){
    tr[i].addEventListener("click", underLine);
    tr[i].addEventListener("mouseleave", undo);
    function underLine(){
        tr[i].classList.add("Line");
    }

    function undo(){
        tr[i].classList.remove("Line");
    }
}

```

}