

HomeController:

Pt.1

```
package ca.sheridancollege.barnessp.controllers;

import java.util.List;

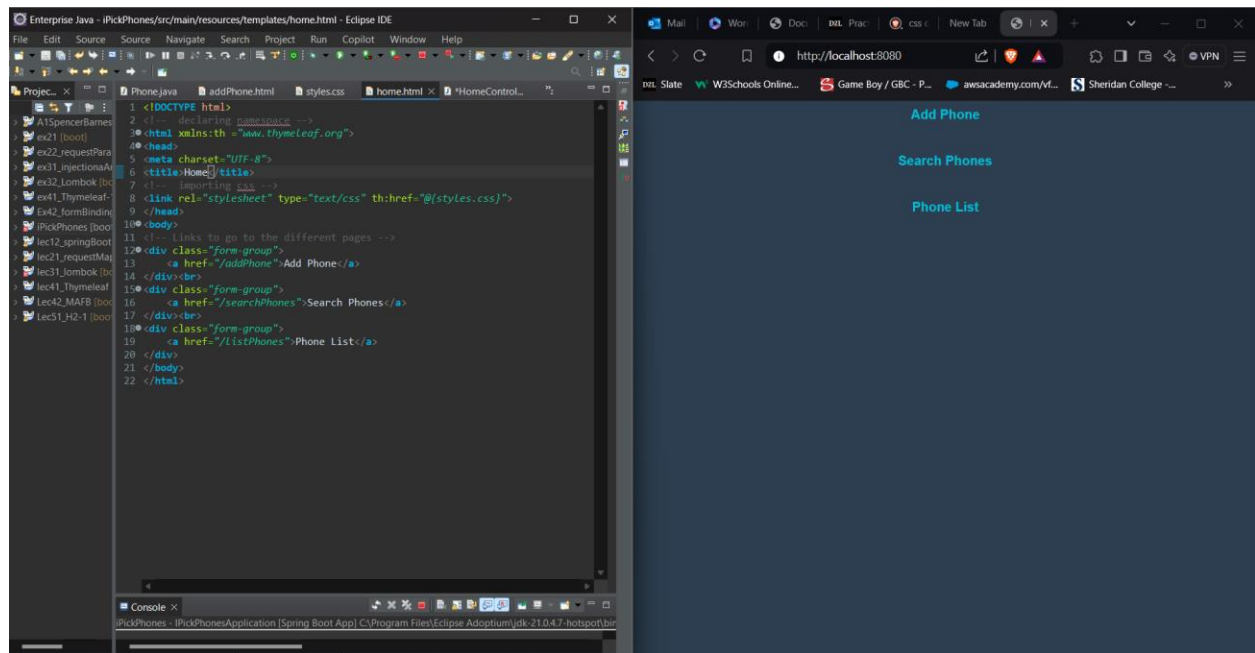
@Controller
public class HomeController {
    List<Phone> searchedPhoneList = new CopyOnWriteArrayList<Phone>();
    List<Phone> phoneList = new CopyOnWriteArrayList<Phone>();
    // maps to localhost:8080/ and displays the home page
    @GetMapping("/")
    public String goHome(Model model) {
        return "home";
    }
    // makes sure the phone object exists in addPhone and displays addPhone.html when a link is pressed or localhost:8080/addPhone is entered
    @GetMapping("/addPhone")
    public String addPhone(Model model) {
        model.addAttribute("phone", new Phone());
        return "addPhone";
    }
    // same for addPhone but instead of a phone object it makes sure it has the phoneList as an attribute
    @GetMapping("/listPhones")
    public String listPhones(Model model) {
        model.addAttribute("phoneList", phoneList);
        return "listPhones";
    }
    // same for above but has all three, phone, phoneList and searchedPhoneList as attributes
    @GetMapping("/searchPhones")
    public String searchPhones(Model model) {
        model.addAttribute("phone", new Phone());
        model.addAttribute("phoneList", phoneList);
        model.addAttribute("phoneList", searchedPhoneList);
        return "searchPhones";
    }
}
```

Pt.2

```
// takes the filled out form, creates a new phone object and stores it inside the phoneList, returns to the same page in case you want to add more than one phone
@PostMapping("/addPhone")
public String phoneAdd(Model model, @RequestParam String brand, @RequestParam String modelName, @RequestParam String category, @RequestParam Long price,
    @RequestParam String features){
    Phone phone = new Phone(brand, modelName, price, category, features);
    phoneList.add(phone);
    model.addAttribute("phone", new Phone());
    model.addAttribute("phoneList", phoneList);
    return "addPhone";
}

// more complex, this one creates a dummy phone to use the categories attribute
@PostMapping("/searchPhones")
public String searchPhones(Model model, @RequestParam String category) {
    model.addAttribute("phone", new Phone("", "", Long.valueOf(0), category, ""));
    model.addAttribute("phoneList", phoneList);
    // clears the searched list every time its posted to make sure it doesn't get duplicate data
    searchedPhoneList.clear();
    // checks which category was chosen, for none adds all in the phone list
    for (Phone phone : phoneList) {
        if (category.equals("None")) {
            searchedPhoneList.add(phone);
        }
        // for any other category matches the given category with the category of each phone and if the same adds it to the searched list
        else if (phone.getCategory().equals(category)) {
            searchedPhoneList.add(phone);
        }
    }
    // add phone list to page
    model.addAttribute("phoneList", searchedPhoneList);
    return "searchPhones";
}
}
```

Home.html



Styles.css

Pt.1

```

1  @charset "UTF-8";
2
3
4  /* css for body changing font, adding flex, centering and changing colors*/
5  body {
6      display: flex;
7      flex-direction: column;
8      justify-content: center;
9      font-family: Arial, sans-serif;
10     background-color: #2c3e50;
11     color: white;
12     justify-content: center;
13     align-items: center;
14
15 }
16 /* input css adding border, text color and background color*/
17 input {
18     padding: 10px;
19     border: 2px solid #ccc;
20     border-radius: 4px;
21     background-color: #34495e;
22     color: white;
23 }
24 /*Just flex tings */
25 .content-container {
26
27     flex-direction: column;
28     text-align: center;
29 }
30
31 .form-container {
32     display: flex;
33     flex-direction: column;
34     align-items: center;
35     margin: 20px;
36 }
37

```

Pt.2

```

• .form-group {
    margin-bottom: 20px;
    width: 100%;
    text-align: center;
}

• textarea{padding: 10px;
    border: 2px solid #ccc;
    border-radius: 4px;
    background-color: #34495e;
    color: white;}
/* specific submit input colors for high visibility*/
• input[type="submit"] {
    padding: 10px 20px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}
/*Me trying to center the tables because it wont center on its own */
• .phone-table {

    width: 100%;
    max-width: 800px;
    min-width: 120px;

}

• .phone-table th {
    padding-left: 7.5px;
    padding-right: 7.5px;
    min-width: 120px;
    background-color: #1abc9c;
}

```

Pt.3

```

/*Purple links are ugly and low visibility */
a {
    text-decoration: none;
    color: #00bcd4;
    font-weight: bold;
}
a:hover {
    color: #00e5ff;

a:visited {
    color: #008ba3;
}

```

AddPhone.html

Pt.1

```

<!DOCTYPE html>
<!-- declaring namespace -->
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>Add Phone</title>
<!-- importing css -->
<link rel="stylesheet" type="text/css" th:href="@{styles.css}">
</head>

<body>
<!-- form which posts to the @PostMapping("/addPhone") in HomeController using the phone object -->
<form method="post" action="#" th:action="@{/addPhone}" th:object="${phone}">
<div class="form-group">
<label for="brand">Brand</label><br>
<input type="text" id="brand" th:field="*{brand}" required/>
</div>

<div class="form-group">
<label for="model">Model</label><br>
<input type="text" id="model" th:field="*{modelName}" required/>
</div>

<div class="form-group">
<label for="price">Price</label><br>
<input type="number" id="price" th:field="*{price}" required/>
</div>
<!-- A dropdown makes it easier for later in the searchPhones section -->
<div class="form-group">
<label for="category">Category</label><br>
<select th:field="*{category}">
<option th:each="aCat: *{categories}" th:value="${aCat}" th:text="${aCat}"/>
</select>
</div>

```

Pt.2

```

35 <!-- wanted a larger text box so you could read all the features you're inputting at once -->
36 <div class="form-group">
37   <label for="features">Features</label><br>
38   <textarea id="features" th:field="*{features}" rows="4" cols="50"></textarea> <!-- Larger text box -->
39 </div>
40 <div class="form-group">
41   <input type="submit" value="Submit" />
42
43 </div>
44 <!-- link to home -->
45 <div class="form-group">
46   <a th:href="@{/}">Go Home</a>
47 </div>
48 </form>
49
50
51
52 </body>
53 </html>

```

Running

The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/addPhone`. The browser tabs include Mail, Work, Docu, D2L Pract, css c, and New Tab. The browser's address bar shows the URL `http://localhost:8080/addPhone`. The browser's address bar shows the URL `http://localhost:8080/addPhone`. The browser's address bar shows the URL `http://localhost:8080/addPhone`.

The form is displayed on a dark blue background. It contains the following elements:

- Brand:** A text input field containing the value "1231".
- Model:** A text input field containing the value "1231".
- Price:** A text input field containing the value "0111".
- Category:** A dropdown menu with the following options:
 - Flagship (selected)
 - Flagship
 - Mid-Range
 - Budget
- Features:** A large text area containing the text "1231231".
- Submit:** A green button labeled "Submit".
- Go Home:** A blue link labeled "Go Home".

SearchPhones.html

Pt.1

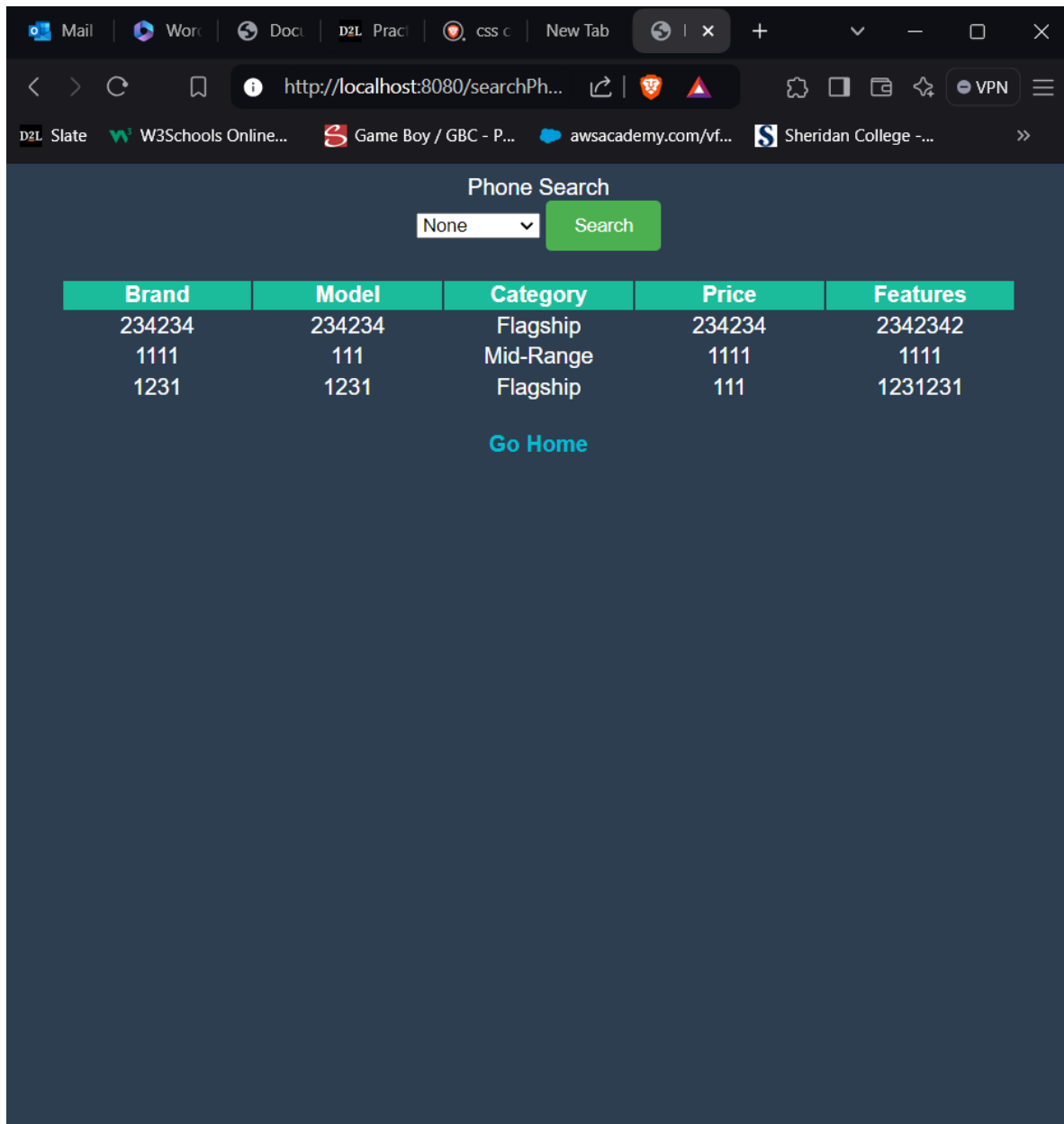
```
1 <!DOCTYPE html>
2 <!-- declaring namespace -->
3 <html xmlns:th="http://www.thymeleaf.org">
4 <head>
5 <meta charset="UTF-8">
6 <title>Phone Search</title>
7 <!-- importing CSS -->
8 <link rel="stylesheet" type="text/css" th:href="@{styles.css}">
9 </head>
10 <body>
11 <div class="content-container">
12 <div class="form-group">
13 <div class="form-group">
14 <!-- posting to @PostMapping("/searchphones") and using the object phone -->
15 <form method="post" th:action="@{/searchPhones}" th:object="${phone}">
16 <label>Phone Search</label><br>
17 <select th:field="*{category}">
18 <!-- using the drop down with the options from categories and the added None value to search for phones -->
19 <option th:each="c : *{categories}" th:value="{c}" th:text="{c}"></option>
20 <option th:value="None" th:text="None"></option>
21 </select>
22 <input type="submit" value="Search">
23 </form>
24 </div>
25
26 <!-- same phone table from list phones -->
27 <table class="phone-table">
28 <tr>
29 <th>Brand</th>
30 <th>Model</th>
31 <th>Category</th>
32 <th>Price</th>
33 <th>Features</th>
34 </tr>
```

Pt.2

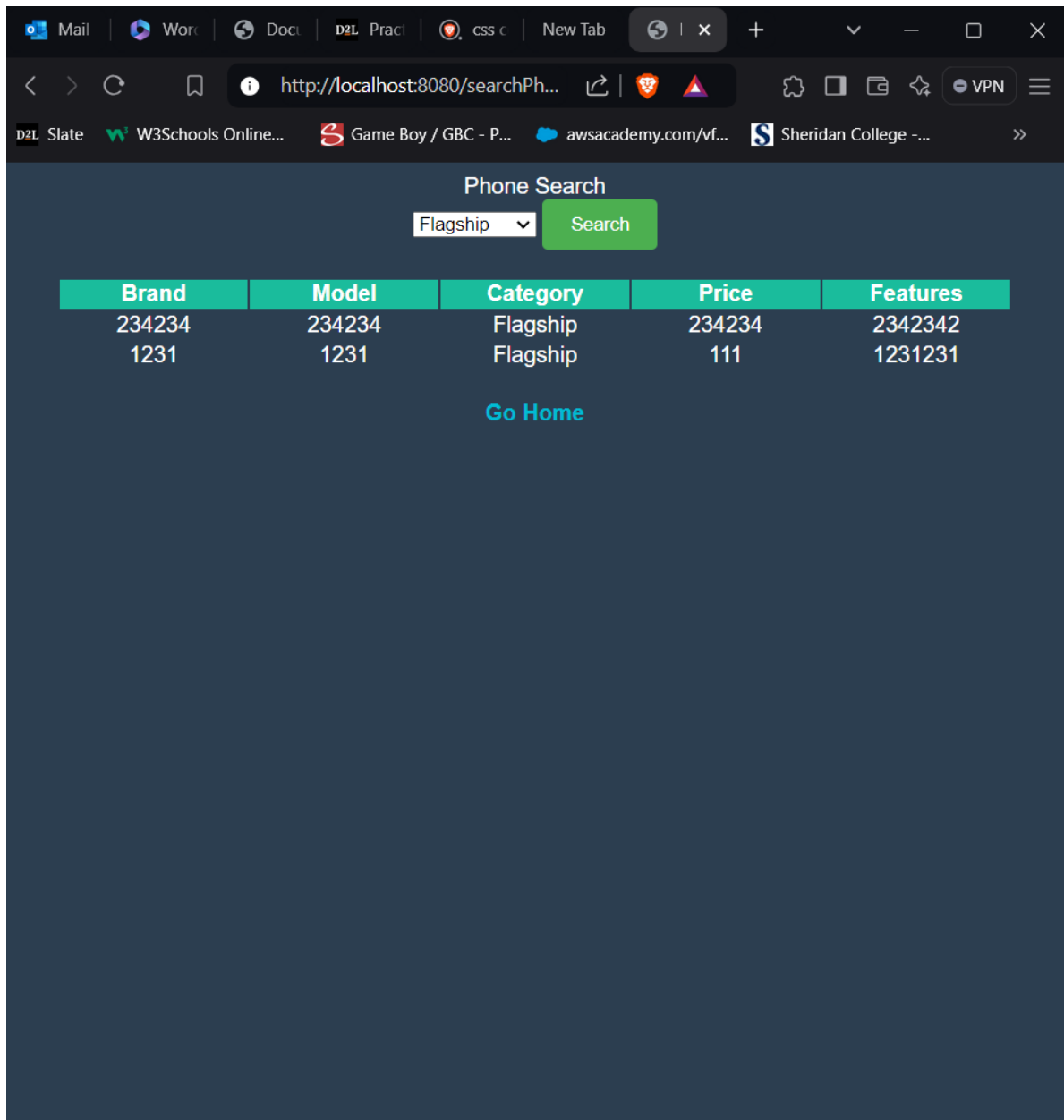
```
34 </tr>
35 <tr th:each="aPhone:${phoneList}">
36 <td th:text="${aPhone.brand}">Samsung</td>
37 <td th:text="${aPhone.modelName}">Galaxy S8</td>
38 <td th:text="${aPhone.category}">Mid-range</td>
39 <td th:text="${aPhone.price}">$1000</td>
40 <td th:text="${aPhone.features}">5MaH Battery, OLED Screen</td>
41 </tr>
42 </table>
43 </div>
44 <div class="form-group">
45 <!-- link to home -->
46 <a th:href="@{/}">Go Home</a>
47 </div>
48 </div>
49 </body>
50 </html>
```

Running

Pt.1



Pt.2



ListPhones.html

Pt.1

```

<!DOCTYPE html>
<!-- declaring namespace -->
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>Phone List</title>
<!-- importing css -->
<link rel="stylesheet" type="text/css" th:href="@{styles.css}">
</head>
<body>
<!-- making flex outer container -->
<div class="content-container">
<!-- inner flex group -->
<div class="form-group">
<h2>Phone List</h2>
<!-- table to display phones -->
<table class="phone-table">
<tr>
<!-- headings -->
<th>Brand</th>
<th>Model</th>
<th>Category</th>
<th>Price</th>
<th>Features</th>
</tr>

```

Pt.2

```

26 <!-- looping through each phone in phonelist and displaying the attributes in the table-->
27 <tr th:each="aPhone:${phoneList}">
28 <td th:text="${aPhone.brand}">Samsung</td>
29 <td th:text="${aPhone.modelName}">Galaxy S8</td>
30 <td th:text="${aPhone.category}">Mid-range</td>
31 <td th:text="${aPhone.price}">$1000</td>
32 <td th:text="${aPhone.features}">5MaH Battery, OLED Screen</td>
33 </tr>
34 </table>
35 </div>
36 <div class="form-group">
37 <!-- link to home -->
38 <a th:href="@{/}">Go Home</a>
39 </div>
40 </div>
41 </body>
42 </html>

```

Running

