**ORF401: eCommerce: Spring 2022**

**Lab 2, Due Wednesday, Feb. 16**

***Client-Side Processing with JavaScript: Making*** [***HandyRides, Inc.***](https://web.archive.org/web/20180615232159/http:/orfe.princeton.edu/courses/orf401/labs/lab1/aboutHandyRides.html) ***look nice***

***Background***

In Lab 1 you developed a simple proof of concept (POC) with the fictitious company HandyRides. In this assignment you will add client-side processing to the HTML page that you developed for that assignment and style the webpage. In particular, you should ensure that the new account information is only sent to the server if all of the required fields have been completed.

**Setup**

1) Use the same project from Lab 1. If you want to make a copy, do it now!

2) Add two files to HandyRides/static: “main.css” and “main.js”

3) Copy and paste the following JavaScript into main.js

function getCookie(c\_name) {

var i,x,y,ARRcookies=document.cookie.split(";");

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

x=ARRcookies[i].substr(0,ARRcookies[i].indexOf("="));

y=ARRcookies[i].substr(ARRcookies[i].indexOf("=")+1);

x=x.replace(/^\s+|\s+$/g,"");

if (x==c\_name) {

return unescape(y);

}

}

}

function setCookie(c\_name,value,exdays) {

var exdate=new Date();

exdate.setDate(exdate.getDate() + exdays);

var c\_value=escape(value) + ((exdays==null) ? "" : "; expires="+exdate.toUTCString());

document.cookie=c\_name + "=" + c\_value;

}

4) In HandyRides/templates/base.html, add the following immediately after the <html> tag

<head>

<script src="{% static 'main.js' %}"></script>

<link rel="stylesheet" href="{% static 'main.css' %}"/>

</head>

5) Edit HandyRides/urls.py to include:

from django.views.generic import TemplateView

and

path('', TemplateView.as\_view(template\_name="index.html"))

inside your urlpatterns

6) Add a file called “index.html” to HandyRides/templates

**Assignment**

1. Make a splash page in “index.html” that welcomes the user. It should include a link to the search page (/rides) (25pts)
2. Add CSS and more HTML to your webapp to make it look nice. Consider the following:
   * Adding images
   * Background color
   * Header/Footer
   * Spacing of elements
   * Fonts

Not looking for the class’ best designer, just play around with CSS and HTML. Looking

for 5 “elements”. (25pts)

1. Write and test:
   1. A JavaScript function checkForm that validates the user’s search when they submit.. Since our search is simple, we will have two requirements. If the user leaves the form blank (both inputs) then don’t submit the form. If the user searches for “Elon Musk” make a popup that says “He’s not here”. **Hint:** You need to add something like:

onclick=”return checkForm();”

To your form (10pts).

* 1. When a user visits your site for the first time, save a cookie and redirect them to the splash page. Using the setCookie and getCookie functions provided above will be helpful. Bonus points for not redirecting if the user’s first visit to the webpage (10+5pts).

1. Discuss how the JavaScript function checkForm could/should be made more generic (i.e., so that the same functions could be used with a variety of different HTML pages without having to be modified). (10pts)
2. Describe other ways in which JavaScript functions might be used to improve the functionality of HandyRides pages. (10pts)
3. Discuss the advantages and disadvantages of saving state information (e.g. information about previous usage) on the client versus saving it on the server. (5pts)