**ORF401: eCommerce: Spring 2021**

**Lab 1**

**Server-Side Processing with HTTP - Part I: Starting up** [**HandyRides, Inc.**](https://web.archive.org/web/20180615232159/http:/orfe.princeton.edu/courses/orf401/labs/lab1/aboutHandyRides.html)

**Background**

Originally, this assignment was to have you build a not-necessarily simple search engine and WWW front-end for the fictitious company [The Great DVD](https://web.archive.org/web/20180615232159/http:/orfe.princeton.edu/courses/orf401/labs/lab1/aboutGDVD.html). The objective is to let users search through a database of movie titles prior to reserving a particular movie. Searching for "needles in haystacks" may well be the most fundamental and most basic function of client-server systems, i.e. the World Wide Web. Part of the search process involves the creation of a "hash table" that contains pointers into the film catalog. The conversion of a given text string into a "hash index" is then a computationally efficient means of returning relevant film(s) from the film catalog.

None the less, that's OK, you can look at one approach and you can still salvage some benefit from this assignment by doing the following six (6) parts below, which are due, on Sunday, Feb, 14 end of day (11:59pm).

Some of you in the class may not very familiar with scripts. they are simply a collection of what otherwise would be "command-line" functions (line-at-a-time interpreted commands) that are assembled into a series of commands executed one after the other while the file containing these instructions is executed. At this point, learning Python will help you write your scripts. Additionally, Django, a web framework, helps turn your Python scripts and logic into a full fledged web app. This assignment is meant to introduce you to Django.

**Setup**

1) First, we need to install Python. You can check to see if Python is installed by opening your terminal (Command Prompt on Windows) and typing `python -V`. If this prints out a number, probably like 3.9.1, then you are all set. If your version is below 3.7, you should probably upgrade your Python.

**Windows:** Go here and download the installer (bottom of the page) https://www.python.org/downloads/release/python-391/

**Mac:** I recommend installing Homebrew first <https://brew.sh/> . Homebrew is a simple way to download software on Mac. Once you have it installed, simply type `brew install python`.

2) Next, we need to install Pip. Pip is a Python package that helps you install other Python packages. We need this to install Django. Follow the instructions here (you don’t need to install it if you already have it, check first!): [https://pip.pypa.io/en/stable/installing/#python-and-os-compatibility](https://pip.pypa.io/en/stable/installing/" \l "python-and-os-compatibility)

**Note:** If you have Anaconda installed already, you can replace `pip` commands with `conda`

3) The last thing we need to install is Django, if you successfully installed pip, it should be as simple as typing `pip install Django==3.1.6` in your terminal.

4) Unzip the file provided on Blackboard. This is our base webapp for our company. Inside, you will see a file called “manage.py”, this file is a script to launch your webapp. To test if you did everything correctly, open your terminal in the folder that has “manage.py” (or use cd to change the directory in your terminal). Then type `python manage.py runserver` which will launch your webapp. You should see something like “<http://127.0.0.1:8000/>” in the terminal message, you can put that in your browser to see the app. If you see a Django splash page, then you followed everything correctly and are ready to complete the assignment. Our webapp has a search page at “<http://127.0.0.1:8000/rides>” to view the potential riders of our app. If you want to shutoff the webapp, you need to press Ctrl+C to stop it.

**Assignment**

This semester we are going to do as we did last year, focus on a different fictitious company that hopefully we'll actually launch by the end of the semester. This company focuses on creating software plugins that others can use to easily enable their customers to share rides.

My name for this company is [HandyRides, Inc.](https://web.archive.org/web/20180615232159/http:/orfe.princeton.edu/courses/orf401/labs/lab1/aboutHandyRides.html) primarily because I was able to purchase HandyRides.com .

The company's initial product will focus on events that entice on-site participation from many individuals, have somewhat defined start and end times and require the participants to go through some pre-registration process.

An early partner/customer for this HandyRides should be [eVite](https://web.archive.org/web/20180615232159/http:/www.evite.com/) and similar companies.

1. In preparation for all of this, please download the CoPilot|Live, Uber and Lyft apps for your iOS or Android (or, heaven forbid, windowsMobile, does it still exist???) device.
2. Look around on-line and find an event that has these characteristics AND for which you can find a list of attendees to that event. (That may not be so easy; however, many conferences make available to its attendees a list of all others that attended. Unfortunately, these lists are usually password protected.) Find some event that has divulged its list of attendees. You may have to search legal records, like divorce proceedings, to find such a list. Be resourceful and creative!! Again, what you need is an event, its registration page (which ends up showing the scope of the personal information an attendee had to divulge in order to participate in the event). AND a list of the attendees and, at least, their addresses that registered/attended. (If you really can't find these data, then you will need to create by hand (or by any other means) a file that contains the registration information for the event that you have chosen. That file must have at least 30 registrants.)

There is a specific format that the data must be in. Use the format shown in “HandyRides/rides/fixtures/riders.json” to create your list of riders. To update your database, run `python manage.py loaddata`. (30pts)

The function of this first HandyRides plug-in is to find event registrants for which it may be feasible for them to travel together to and/or from the event thus saving money and possibly enjoying some companionship. Once found, the product must create an environment such that those paired individuals find it in their best interest to share a ride. Needless to say, the process will need to be very easy to use and enticing.

1. Suggest a better name that has its .com still available. (10 pts.)
2. Create a logo for either your named company (if you think that you have a name that is really as good or better than HandyRides.com) or for HandyRides.com (10 pts)
3. Create a search page that returns to the User all of the registrants from the 2-letter abbreviation of the State that the User typed into the Search Form. Do this by editing and testing the files named “HandyRides/rides/views.py” and “HandyRides/rides/forms.py” that you downloaded. As shown in the precept, these let a user enter a search term and display the results of a search. Edit it by at least using your logo and some other images and having it search the registration list that you found or created yourself.

Consider this page as the beginnings of the interface page that would be presented to the User upon registration to see if there is anyone else that he/she could share a ride to and/or from this event. This is but the beginning of the design. There are issues of privacy and follow-up that we will need to deal with.

* Your new search can check the state abbreviation (20 pts)
* Your new search is case insensitive. That is, if your user searches “ny” it will return results where “NY” in the database. (10 pts)
* Your search accepts two arguments, one for origination/destination city and another for state. That way, a user can specify a city in a particular state (10 pts)

1. We need a business case for this first product. Please begin making one. How will it be monetized? What is the scope/size of the market? What are the costs? (10 pts)

**Submission**

To submit your assignment, simply zip the top level of your Django app (if you unzipped the file on Blackboard, it should be called HandyRides) and include this zip file along with another file that includes your writeup on Blackboard. For grading, I will be launching your Django app and testing your search on my local machine.