

Lab #3 - ORF401 EL, SP, JJ

Name: HopStar



1. What are Migrations?
 - a. Based on the webpage and the fact that we were making changes to models.py, “Migrations” must refer to making changes to a model’s database and what exactly its saving. The migration process involves creation files that define how to alter the database structure in line with the application’s data models.
2. Create a couple of rides using your new model by simply adding them manually on the webapp. Once you have everything working, we don’t actually want the form to be on the same page as the search page. Create a new url and template (like we did with the splash page) and put the registration form on a new page. You have all the pieces to make a usable web app (adding new rides to a public list), make it usable by incorporating some CSS and JS as you did in Lab 2. Additionally, add some links to the pages to allow users to navigate to the search and to the “add ride” page. In your write up, include what your customizations for your site are (can’t be the same as Lab2). Make sure everything works on your live website after deploying it. (25 pts)
 - a. In the customization of our Ridesharing Registration page, enhancements were made that extended beyond just making an html page. First, we realized that we would like all three sections of the site so far (Homepage, Registration page, SearchPage) to be accessible at all times so we edited

Base.html and main.css to include links at the top of the page which can be clicked no matter what part of the site you are on which really makes the header look professional and more aesthetic than being an empty block of green as previously. We also enhanced the form fields to include placeholders like "MM/DD/YYYY" in date to show what is expected in each submission bar. We also used javascript to move checkcookie() to load on base.html rendering instead of each individual webpage. We tested the site and all aspects and saw that everything was working as intended with a sleek, aesthetic, and simple design using main.css.

4. Describe shortcomings of using a standard WWW browser as a client for this kind of application

As question 6 alludes to, WWW browsers are susceptible to security vulnerabilities like CSRF and XSS. These vulnerabilities can be exploited to manipulate sessions, steal information, or execute scripts. This client is likely worse than a mobile app which could garner more user data via a more robust cookies implementation as well as have a prettier or more accessible user interface. This is more of an issue of accessibility as more and more users move to mobile devices and mobile applications for a standard client for this application.

5. In your ideal app, what additional models would you introduce that would be helpful to building your version of a ride sharing app.

In our ideal ride sharing app, we would have separate user interfaces for drivers and riders as well as a maps integration so that people can see their route before they depart as well as in real time! Since it would be a b2b, we would not take payment via the apps and our revenue would not flow directly through the website. Despite this, we would still implement user feedback mechanisms into our current model.

6. The current webapp doesn't implement any user authentication. Part of the reason why is because implementing safe login requires countermeasures against Cross-Site Request Forgery (CSRF). You may have noticed the csrf-token in the new ride form. What is CSRF and how would you explain it to your 5 year old niece?

Cross-Site Request Forgery (CSRF) is like a trick where a bad wolf pretends to be a friendly sheep to sneak into a party he wasn't invited to. Imagine you're having a tea party with your toys, and you've told everyone that only friends who know

the secret handshake can share the tea and cookies. But what if a sneaky wolf dresses up like one of your toy friends and learns the secret handshake? He could then come to your party, pretend to be your friend, and eat all the cookies without you realizing he's not supposed to be there. To stop this from happening, we use something special called a "csrf-token" when we have parties on the internet. It's like a unique, super-secret handshake that changes every time we have a party. This way, even if the wolf learns the old handshake, he won't know the new one, and he can't trick us into giving him the cookies. So, with this csrf-token, we make sure that only our true friends can share our tea and cookies at the party.