01/28 Finish reading project document and tutorial part #1-4.

01/29 Both of us did not know how to start;

Went to office hour and found Corey SQL YouTube channel useful.

01/30 Finished Corey Django Tutorial up to part 10.

01/31 Done user register and log in.

Changed css settings: it looks better now.

02/01 Done driver registration, same logic as user registration basically.

Issue: Driver class has all fields as CharField. It turns out that using request.user.username to search for objects is the most convenient way.

Solved: by making one field called ‘user’ a OneToOneField to model User. As a result, there is only one driver object could exist with one username.

02/02 Created model called Ride, it has all info related to or needed to locate one ride object, such as owner, driver, sharer, destination and etc.

Issue: we have to delete the database many times because whenever there is issue, causing a 404 error, the object is created and when we came back and tried to do the action again, the UNIQUE constraint is violated.

02/03 Previous issue solved by making driver, sharer, owner, who all have one field called ‘user’ a OneToOneField to User model, a charfield. And we, at the backend, set the value manually. By doing so, we can ensure that user cannot input any wrong value in their forms and do not have to delete the objects from the database which is very bad.

02/04 Ride search function finished.

Design strategy: if the current logged-in user has not registered as driver, he/she could search and take any rides that is open as a sharer; if the user has already registered as a driver, his action, same as before, will become a ride confirmation.

Also finished ride finish functionality: created a link, only visible to drivers, to finish current ride. After driver clicks finish button, the ride object is deleted from the database.

02/05 Done email notifications.

Issue met: email could not be sent.

Solution: go to google account settings, and allow login request from less secured apps.

02/06 Tried out in Docker container.

Issue met: docker does not know port 8000.

Solution: forgot to do a chmod.

02/07 It could be a problem to put all the applications in one Docker

Issue met: from class lecture

Solution: need further learning of how to create more Docker and arrange the application