User Management

Skills Network

Estimated time needed: 1 hour and 20 mins

The admins of your app will need to manage users and user access based on users' types, such as, visitors and registered users. Your task is to add a standard user management feature to your app.

Environment setup

If your Theia workspace has been reset and you want to continue from where you left off previously, you can `git clone` or pull the latest code from your GitHub repository.

- Set up the Python runtime again if Theia workspace has been reset.

```
1. 1
1. python3 -m pip install -U -r requirements.txt
```

Copied!

Note: You may need to perform migrations for a new Theia environment.

Create a superuser for your app

Let's create a superuser first.

- Stop the server, if started. `cd` to `/server` folder and run:

```
1. 1
1. python3 manage.py createsuperuser
```

Copied!

With `Username`, `Email`, `Password` entered, you should see a message indicating that the superuser has been created:

```
1. 1
1. Superuser created successfully.
```

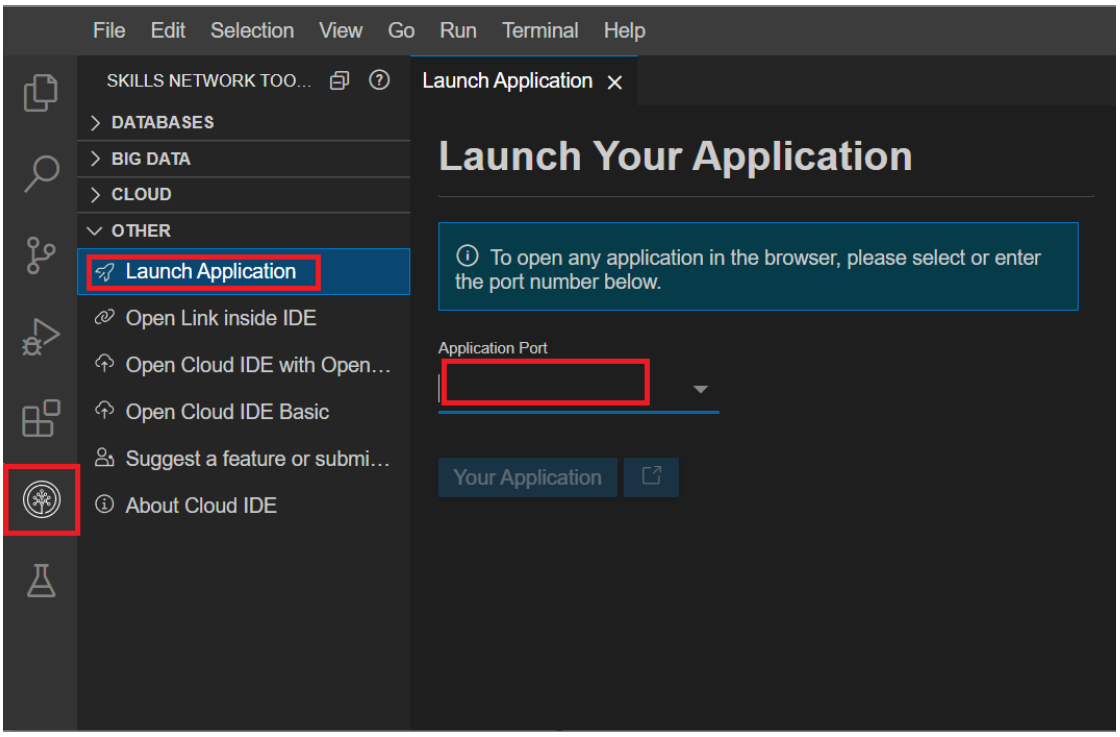
Copied!

Start the server again and log in to the Admin site as the superuser.

```
1. 1
1. python3 manage.py runserver
```

Copied!

- Click on the Skills Network button on the right, it will open the "Skills Network Toolbox". Then click `OTHER`, then `Launch Application`. From there you should be able to enter the port as `8000` and launch the development server.



After the browser tab opens, add the `/admin` path. Your full URL should look like the following:

`https://userid-8000.theiadocker-1.proxy.cognitiveclass.ai/admin`

- Log in to the admin site with the credentials you just created for the superuser.
- Click `Users` under the `AUTHENTICATION AND AUTHORIZATION` section. You should be able to view the superuser you just created.

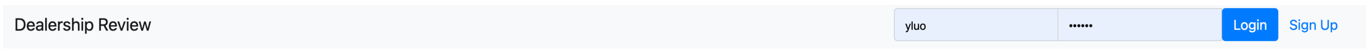
Add user login/logout and signup menu items

First, let's add login/logout and signup menu items to the navigation bar `<nav>` created in the previous lab.

- Open `./templates/djangoapp/index.html` and find the created navigation bar `<nav>` to the html template.

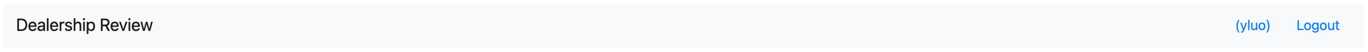
Add the following menu items to `<nav>`:

The created `<nav>` bar for login and signup may look like the following screenshot:



Add the following navigation bar items after user has been authenticated:

The `<nav>` bar for an authenticated user may look like the following screenshot:



More details about adding Bootstrap navigation bar with login/logout and signup menu items could be found in this lab:

[Bootstrap Integration]

Add a new login view

Next, you need to create a new login Django view to handle a login request.

- Open `djangoapp/views.py` and add a new login view to authenticate user.
- Configure a route for the login view by adding a path entry in `djangoapp/urls.py`.

You may refer to this lab to get more details about Django authentication:

[Django Authentication System]

Take a screenshot for peer-review:

After you have created the login view, please take a screenshot and name it as `login.jpg` or `login.png` for peer-review.

Add a new logout view

After the user can login, you need to create a new logout Django view to handle the logout request.

- Open `djangoapp/views.py` and add a new logout view to handle a logout request.
- Configure a route for the logout view by adding a path entry in `djangoapp/urls.py`.

You may refer to this lab to get more details about Django authentication:

[Django Authentication System]

Take a screenshot for peer-review:

After you have created the logout view, please take a screenshot and name it as `logout.jpg` or `logout.png` for peer-review.

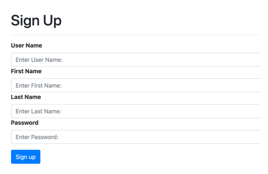
Add a sign-up template

At this point, you should be able to log in and log out with the superuser you created earlier.

Next, you will create a signup or registration page to create new regular users.

- Open `./templates/djangoapp/registration.html` and add a registration form with:
  - A text input to receive first name.
  - A text input to receive last name.
  - A text input to receive username.
  - A password text input to receive password.
  - A submission button pointing to a signup view, to be created later.

The new signup page may look like the following screenshot:


Sign Up

User Name

Enter User Name:

First Name

Enter First Name:

Last Name

Enter Last Name:

Password

Enter Password:

Add a new sign-up view

You also need to create a new signup Django view in order to register(create) a new user and then log in the user.

- Open `djangoapp/views.py`, add a new sign-up view.
- Configure a route for the signup view by adding a path entry in `djangoapp/urls.py`.

You may refer to this lab to get more details about Django authentication:

Django Authentication System

Take a screenshot for peer-review:

After you have created the sign-up view, please take a screenshot and name it
as `sign-up.jpg` or `sign-up.png` for peer-review.

Test the updated app

Now you may want to test the updated Django app with signup, login, and logout end-to-end.

Submission

Commit and push the changes to your GitHub repository.

If you need to refresh your memory on how to commit and push to GitHub in Theia lab environment, please refer to this lab Working with git in the Theia lab environment

Summary

In this lab, you have added user management related templates and views to the app.
In the next lab, you will start to create car/make related models, templates, and views.

Author(s)

Yan Luo

Other Contributor(s)

Upkar Lidder

Changelog

Date	Version	Changed by	Change Description
2022-09-20	1.3	K Sundararajan	Updated pip (packages installation) command
2022-09-14	1.2	K Sundararajan	Updated instructions
2022-09-01	1.1	K Sundararajan	Updated `Launch Application` instructions as per the new Theia IDE
2021-01-28	1.0	Yan Luo	Created new instructions for Capstone project