Full Stack Web Developer Nano-Degree Project (4): **Item Catalog**

Due Date: Dec 22, 2018

1. Project Overview

In this project, you will implement item catalog application by utilizing the knowledge you learned about SQLAlchemy, Flask Framework, OAuth 2.0 Protocol, APIs.

2. Project Prerequisites

In order to start implementation smoothly, take a moment to check that all below prerequisites are installed in your PC:

- Unix-Style Terminal Program
- Vagrant: https://www.vagrantup.com/downloads.html
- Virtual Machine: https://www.virtualbox.org/wiki/Downloads
- FSND Virtual Machine: https://github.com/udacity/fullstack-nanodegree-vm

Once you get the above installed, run below commands inside `fullstack-nanodegree-vm`:

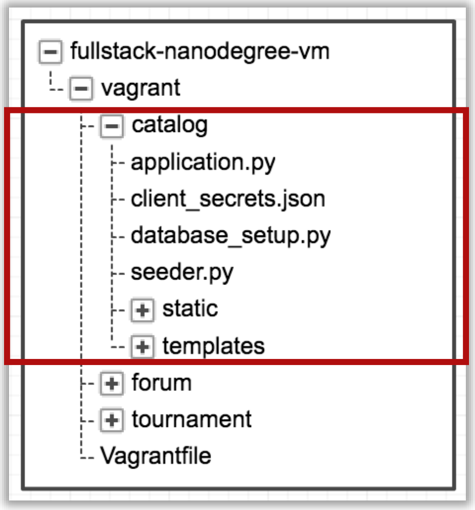
```
vagrant up

vagrant ssh

cd /vagrant/catalog
```

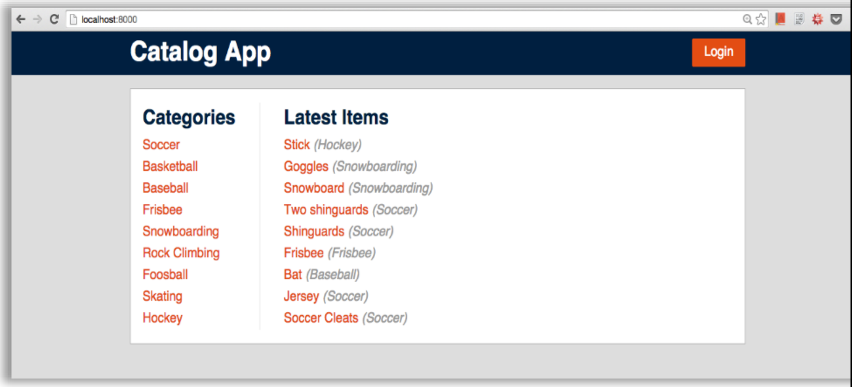
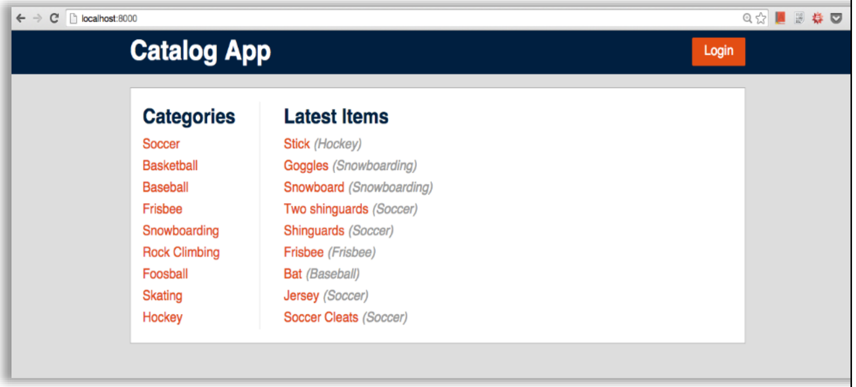
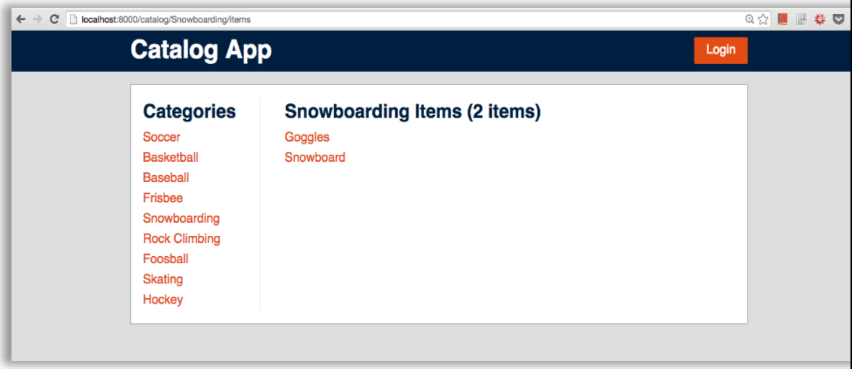
3. Project Structure

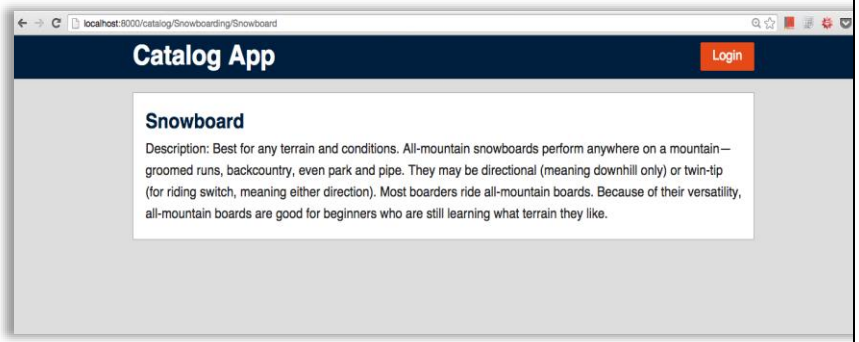
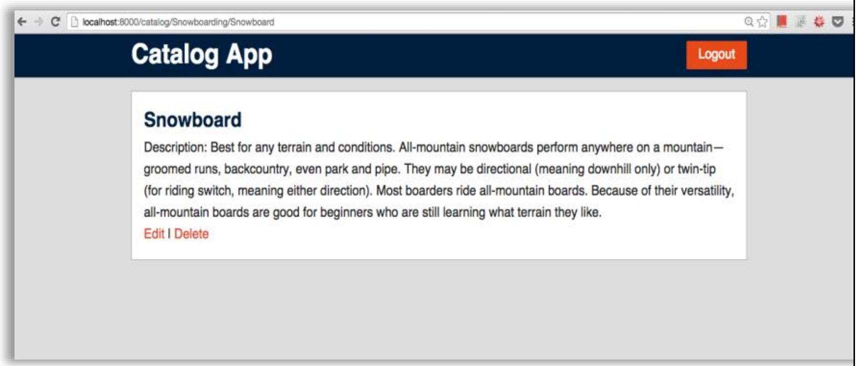
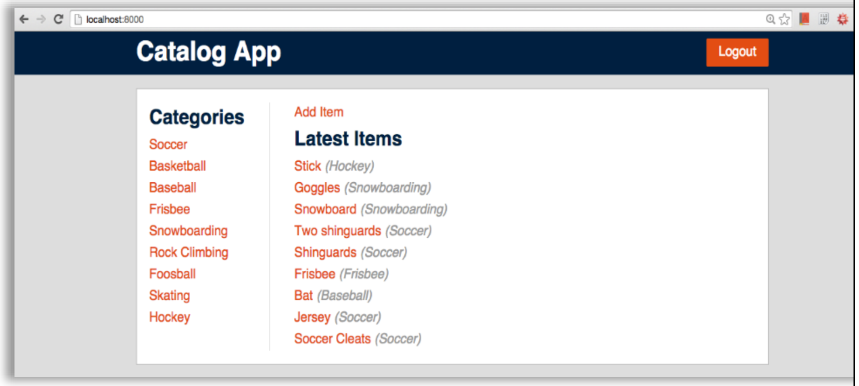
Your project should be structured as illustrated below, the project appears inside **catalog** folder:

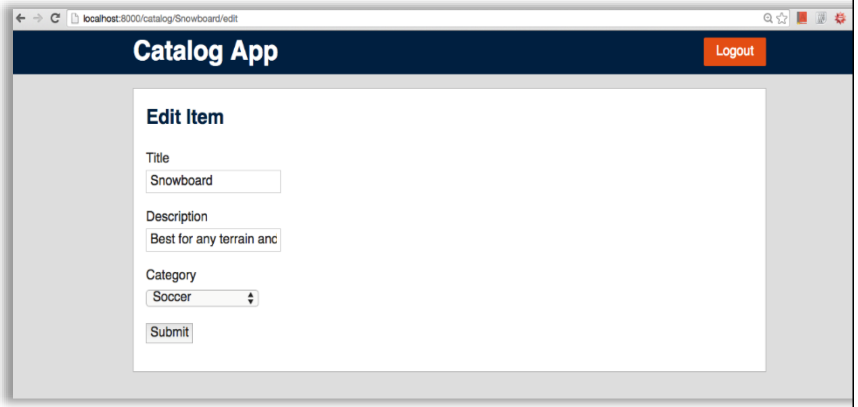
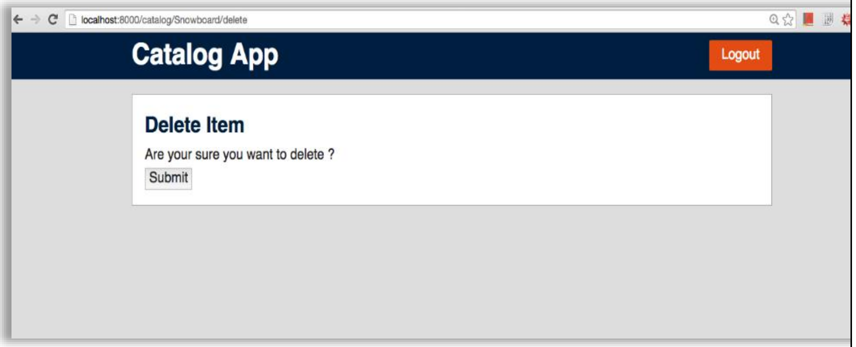
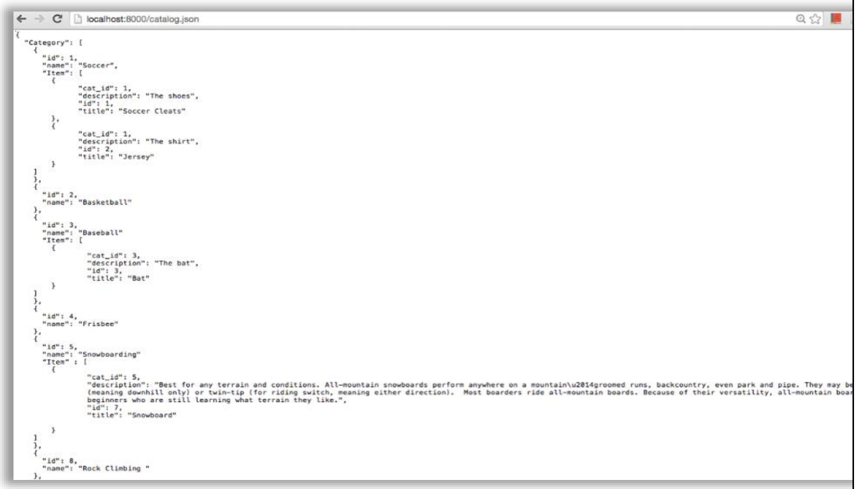


```
⊟ fullstack-nanodegree-vm
└-⊟ vagrant
    ├-⊟ catalog
    |   ├- application.py
    |   ├- client_secrets.json
    |   ├- database_setup.py
    |   ├- seeder.py
    |   ├-⊞ static
    |   └-⊞ templates
    ├-⊞ forum
    ├-⊞ tournament
    └- Vagrantfile
```

4. Project Functionalities

Function	Mockup	Description
Auth	 localhost:8000 Catalog App Login **Categories** Soccer Basketball Baseball Frisbee Snowboarding Rock Climbing Foosball Skating Hockey **Latest Items** Stick *(Hockey)* Goggles *(Snowboarding)* Snowboard *(Snowboarding)* Two shinguards *(Soccer)* Shinguards *(Soccer)* Frisbee *(Frisbee)* Bat *(Baseball)* Jersey *(Soccer)* Soccer Cleats *(Soccer)*	- Authentication / Authorization should be implemented via OAuth 2.0 - Whenever a user is logged in to the application for the first time, he should be added as a new User in the backed (DB) - Manipulating items are only available to authorized users (add item, edit item, delete item) - Item's owner is authorized to edit / delete the item he created, other users could not manipulate the item
Browse Catalog	 localhost:8000 Catalog App Login **Categories** Soccer Basketball Baseball Frisbee Snowboarding Rock Climbing Foosball Skating Hockey **Latest Items** Stick *(Hockey)* Goggles *(Snowboarding)* Snowboard *(Snowboarding)* Two shinguards *(Soccer)* Shinguards *(Soccer)* Frisbee *(Frisbee)* Bat *(Baseball)* Jersey *(Soccer)* Soccer Cleats *(Soccer)*	- This is considered as the main page of the application - Clickable categories are listed in the navigation bar, whenever a category is clicked Browse Category function will be executed - Clickable list of recent items (last added items) are shown along with the category name they belong to, whenever an item is clicked, Browse Item function will be executed
Browse Category	 localhost:8000/catalog/Snowboarding/items Catalog App Login **Categories** Soccer Basketball Baseball Frisbee Snowboarding Rock Climbing Foosball Skating Hockey **Snowboarding Items (2 items)** Goggles Snowboard	- Clickable categories are listed in the navigation bar, whenever a category is clicked, its corresponding items list will be shown - Clickable list of items for the selected category is shown, whenever an item is clicked, Browse Item function will be executed

Browse Item	 localhost:8000/catalog/Snowboarding/Snowboard Catalog App Login Snowboard Description: Best for any terrain and conditions. All-mountain snowboards perform anywhere on a mountain—groomed runs, backcountry, even park and pipe. They may be directional (meaning downhill only) or twin-tip (for riding switch, meaning either direction). Most boarders ride all-mountain boards. Because of their versatility, all-mountain boards are good for beginners who are still learning what terrain they like.  localhost:8000/catalog/Snowboarding/Snowboard Catalog App Logout Snowboard Description: Best for any terrain and conditions. All-mountain snowboards perform anywhere on a mountain—groomed runs, backcountry, even park and pipe. They may be directional (meaning downhill only) or twin-tip (for riding switch, meaning either direction). Most boarders ride all-mountain boards. Because of their versatility, all-mountain boards are good for beginners who are still learning what terrain they like. Edit | Delete	- Item title and description are shown - If the user is logged in, edit and delete functionalities will be provided
Add Item	 localhost:8000 Catalog App Logout Categories Soccer Basketball Baseball Frisbee Snowboarding Rock Climbing Foosball Skating Hockey Add Item Latest Items Stick *(Hockey)* Goggles *(Snowboarding)* Snowboard *(Snowboarding)* Two shinguards *(Soccer)* Shinguards *(Soccer)* Frisbee *(Frisbee)* Bat *(Baseball)* Jersey *(Soccer)* Soccer Cleats *(Soccer)*	- In the main page, add functionality will be provided if the user is logged in

Edit Item	 localhost:8000/catalog/Snowboard/edit Catalog App Logout **Edit Item** Title: Snowboard Description: Best for any terrain anc Category: Soccer Submit	- If the user is authorized to edit an item (he is logged in and he is the owner), edit functionality will be provided
Delete Item	 localhost:8000/catalog/Snowboard/delete Catalog App Logout **Delete Item** Are your sure you want to delete ? Submit	- If the user is authorized to delete an item (he is logged in and he is the owner), delete functionality will be provided
API Endpoints	 localhost:8000/catalog.json	- JSON Endpoint will be provided serving all categories along with their items - JSON Endpoint will be provided serving a specific item in a specific category

```
{
  "Category": [
    {
      "id": 1,
      "name": "Soccer",
      "Item": [
        {
              "cat_id": 1,
              "description": "The shoes",
              "id": 1,
              "title": "Soccer Cleats"
        },
        {
              "cat_id": 1,
              "description": "The shirt",
              "id": 2,
              "title": "Jersey"
        }
      ]
    },
    {
      "id": 2,
      "name": "Basketball"
    },
    {
      "id": 3,
      "name": "Baseball"
      "Item": [
        {
                "cat_id": 3,
                "description": "The bat",
                "id": 3,
                "title": "Bat"
        }
      ]
    },
    {
      "id": 4,
      "name": "Frisbee"
    },
    {
      "id": 5,
      "name": "Snowboarding"
      "Item" : [
        {
                "cat_id": 5,
                "description": "Best for any terrain and conditions. All-mountain snowboards perform anywhere on a mountain\u2014groomed runs, backcountry, even park and pipe. They may be
                (meaning downhill only) or twin-tip (for riding switch, meaning either direction).  Most boarders ride all-mountain boards. Because of their versatility, all-mountain boar
                beginners who are still learning what terrain they like.",
                "id": 7,
                "title": "Snowboard"

        }
      ]
    },
    {
      "id": 8,
      "name": "Rock Climbing "
    },
```

5. Development Approach

It is advised to use iterative development explained throughout the lessons:

- Mockups are already provided
- Routing, have a look at the Mockups and try to obtain routing associated with each functionality. Go ahead to your application.py file, define a method for each functionality and associate it with the suitable router
- Templates, inside templates folder create html files as required. Some functionalities are only available for authorized users, you may create publicFile.html and file.html for such cases
- CRUD Functionalities, take a moment to familiarize yourself with the Mockups, define the DB structure accordingly and implement it in database_setup.py. Seed your DB with some data by implementing seeder.py. Go ahead to your application.py and implement below CRUD functionalities inside the methods you defined before in **Routing Phase**:
    - Create Item
    - Read Catalog, Read Category, Read Item
    - Update Item
    - Delete Item
- API Endpoints, inside application.py implement JSON Endpoints for all categories and for a specific item

6. Coding Style, almost there!

For your Python code:

- Use Python PEP8 style guide to test your code quality **`pycodestyle`**
- In terminal run **`pip install pycodestyle`** to install the package
- In terminal run **`pycodestyle fileName.py`** to check the style, you should get 0 errors

7. Submission

- Review Project Rubric to ensure all requirements are satisfied
- Push catalog folder to GitHub along with Readme.md file containing all the steps to start the application