**All Austin Coop Nursery School**

**TABLE OF CONTENTS**

**1. Developer’s Material #**

1.1 Installation Guide #

1.1.1 Installing Python #

1.1.2 Installing Django And Other Packages #

1.1.3 Getting Started with Django Project

1.1.4 Go Live Process

1.2 Sandbox Environment #

1.3 Production Environment #

1.4 Diagrams #

1.4.1 Web Flow Diagram #

1.4.2 Entity Relationship Diagram (ERD) #

1.4.3 Use Case Diagram #

1.5 Coding and I/O Standards #

1.5.1 Overview #

1.5.2 Python and Django #

1.5.3 HTML #

1.6.4 MySQL #

1.6 Detailed Developer Documentation for Account Creation and Login #

Subsection 3.3.a #

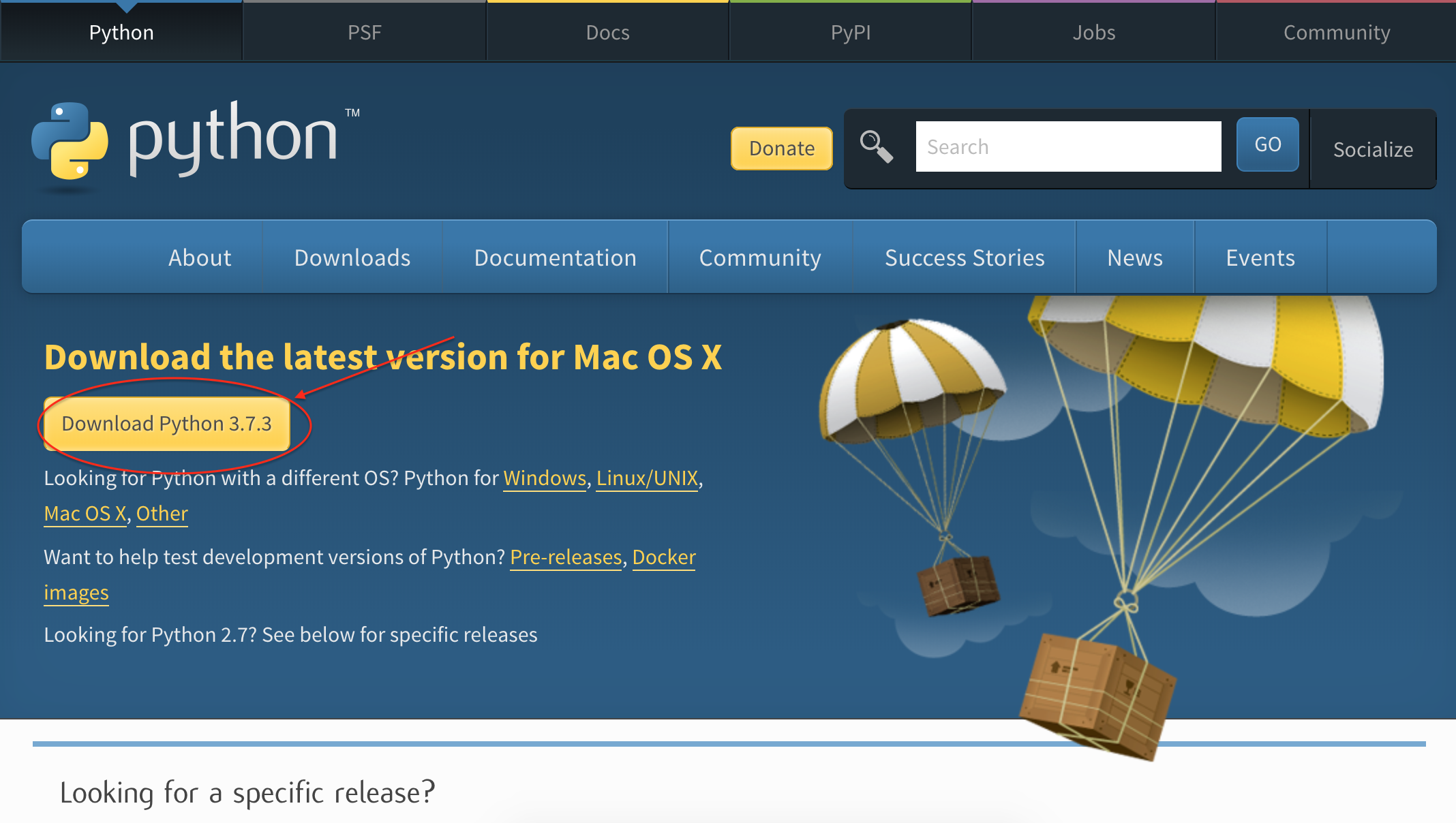
Subsection 3.3.b #

Subsection 3.3.c #

**1.1 Installation Guide**

The application we created is a Django Python App. In order to edit and update the application, the developer will need to install python, django and other required packages onto their computer. This section will cover how to install the appropriate software, how to get started with the program and go live process for the application.

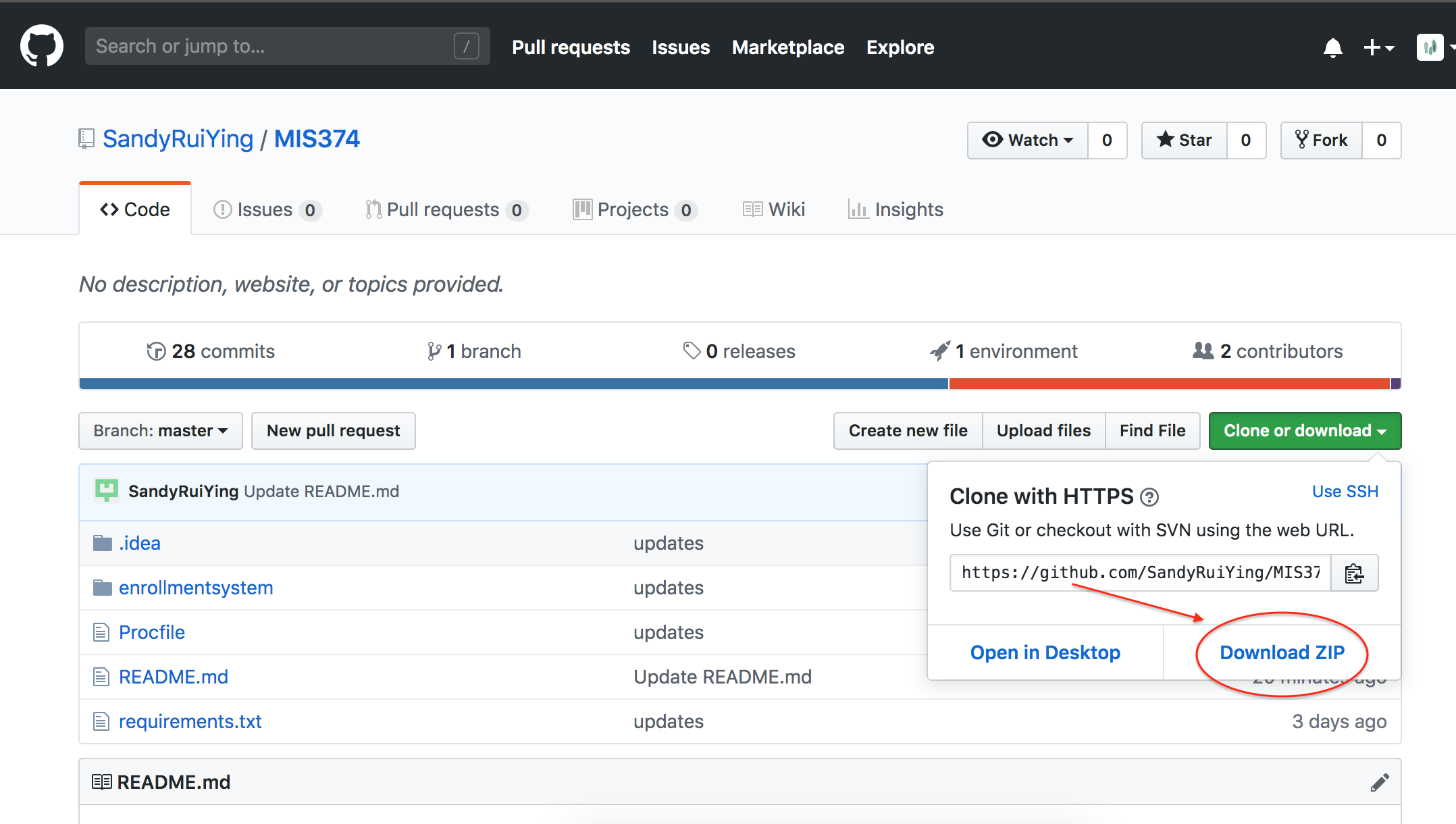
**1.1.1 Installing Python**



Navigate to [www.python.org/dowloads/](http://www.python.org/dowloads/) and click on the “Download Python 3.7.3” button to install python.

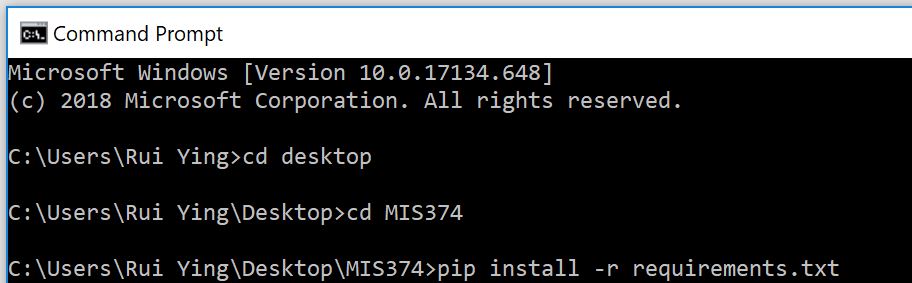
**1.1.2 Install Django and Other Packages**

Step1: Navigate to <https://github.com/SandyRuiYing/MIS374.git> to clone the repository of the application

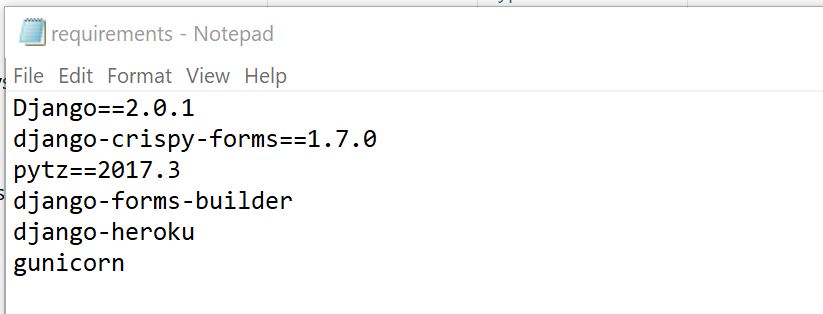


Step2: Move the downloaded ZIP file to Desktop and Unzip the file

Step3: Navigate to the root directory in the command prompt/PowerShell (Windows User) or Terminal (Mac User), and enter the command: “pip install -r requirements.txt” -- This will download Django and other required packages required for the application.

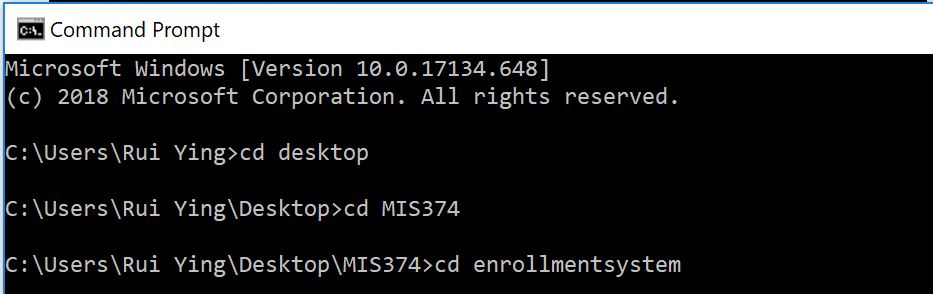


Note: “requirements.txt” should be updated if new package is added to the application

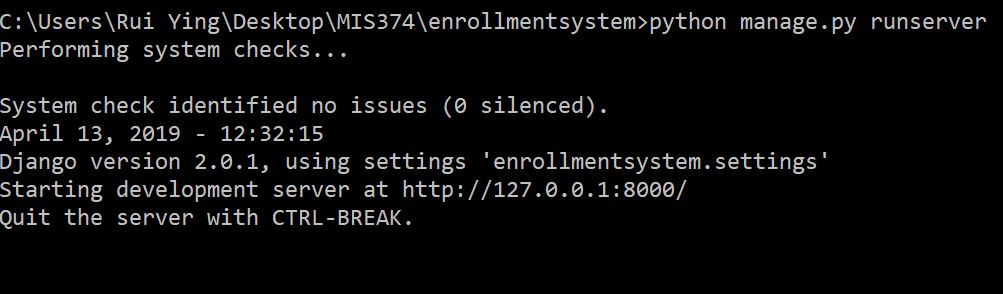


**1.1.3 Getting started**

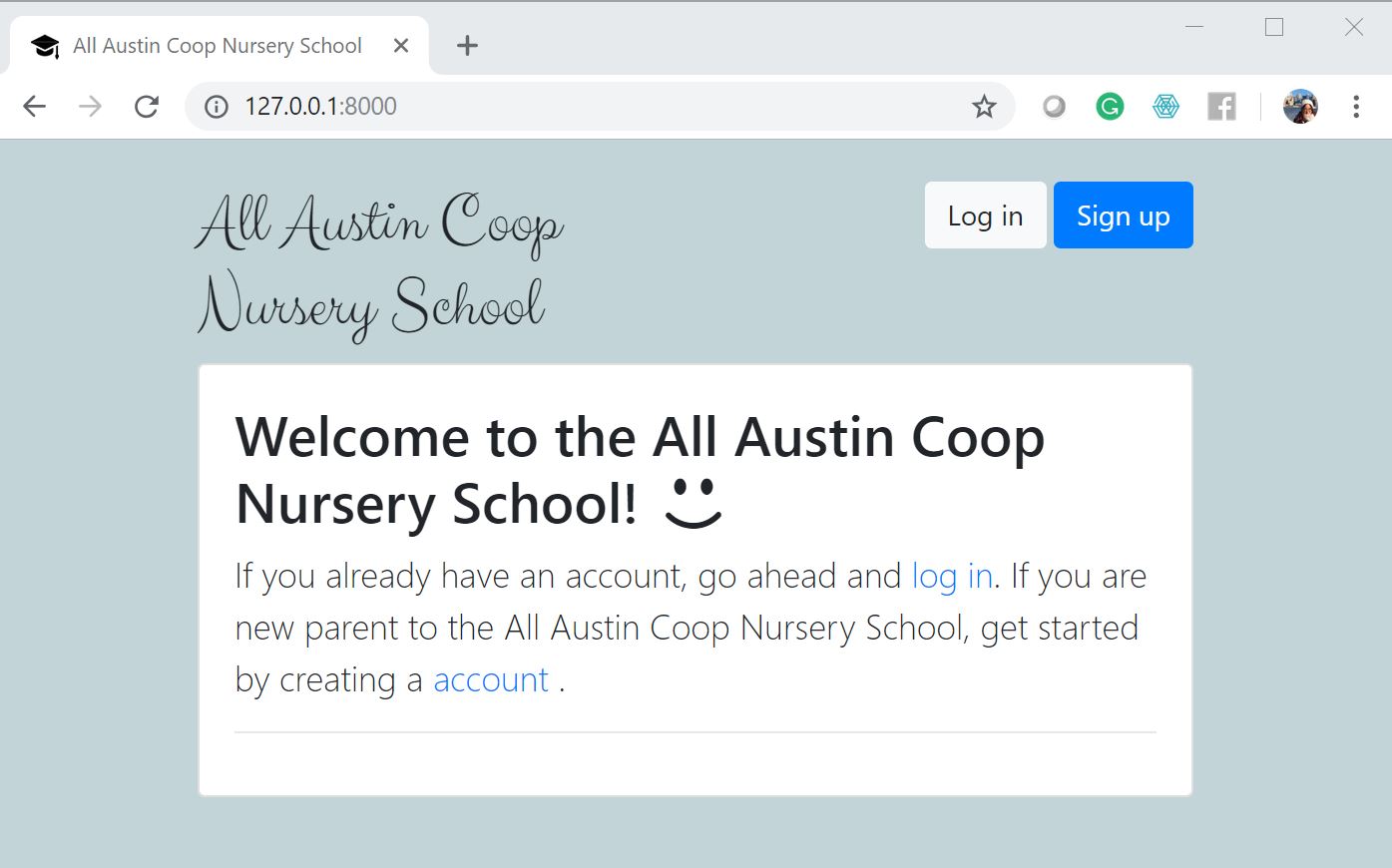
Step1: Open command prompt/PowerShell (Windows User) or Terminal (Mac User) and navigate to the folder contains manage.py



Step2: Enter the command “python manage.py runserver” for running the project locally

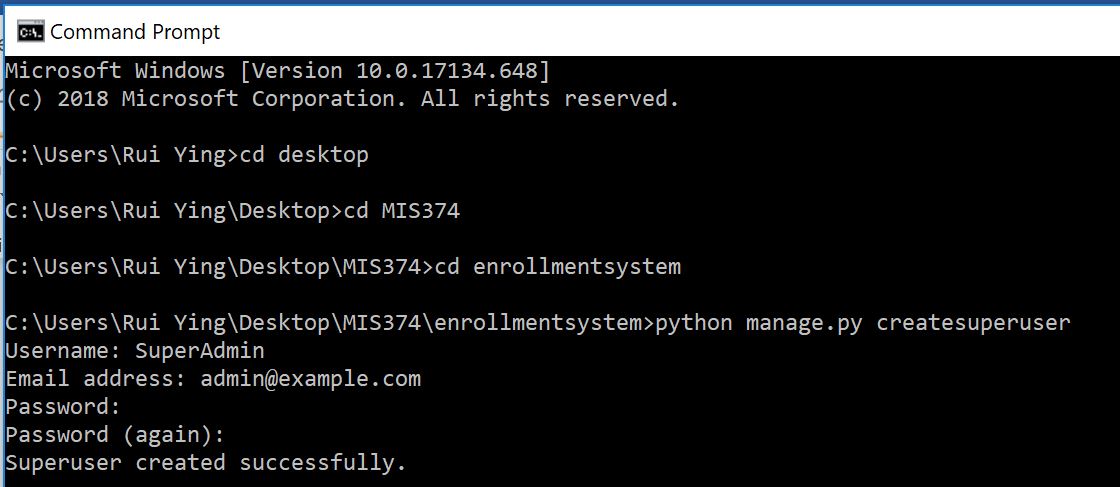


Step3: Copy and paste url <http://127.0.0.1:8000/> to the browser to open the application

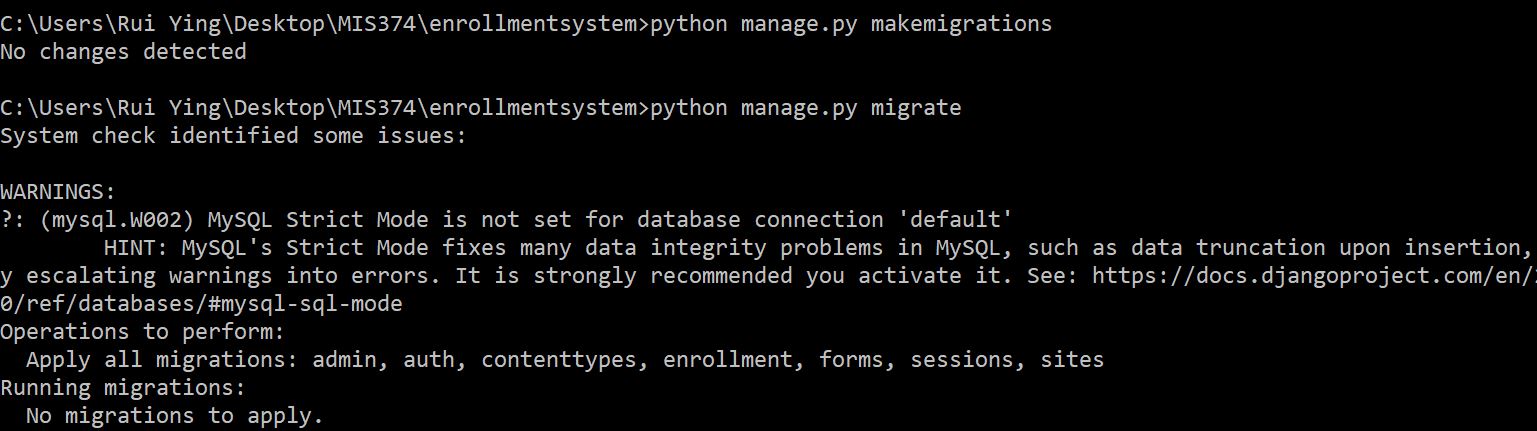


Note: Other useful commands

1. “python manage.py createsuperuser” allows you to create a superuser and gives developers the access to Django Administration page



1. “python manage.py makemigrations” and “python manage.py migrate” allow developers to update models

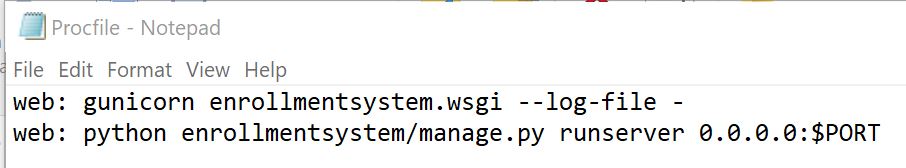


**1.1.3 Go Live Process**

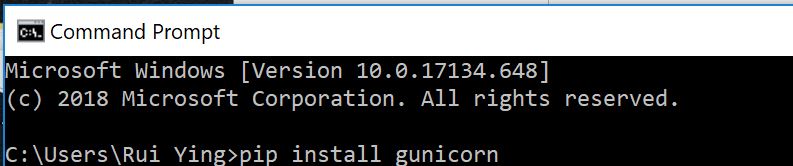
Before final deployment, Heroku can be to host the website for testing purpose so that it will not disrupt the web server that the school’s main website currently runs on. Heroku is a free platform for hosting the website and here are the steps to deploy an application on Heroku:

Step1: Install Git <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git> and Heroku CLI <https://devcenter.heroku.com/articles/heroku-cli#download-and-install>

Step2: Create Profile (without ‘txt’ exntension). Procfile specified application's process types and entry points and should be positioned at the root of the repository.

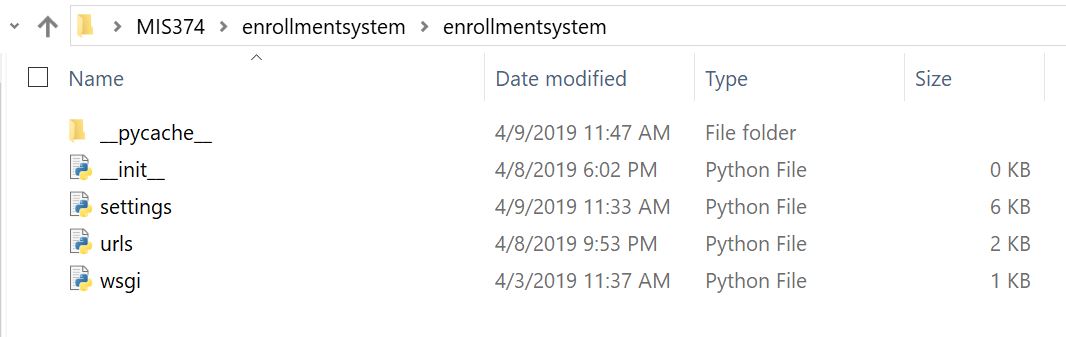


Step3: Install gunicorn package and django-heroku through comman prompt. The Procfile requires gunicorn, which is a production web server. (Note: add gunicorn and Django-heroku to requirements.txt)





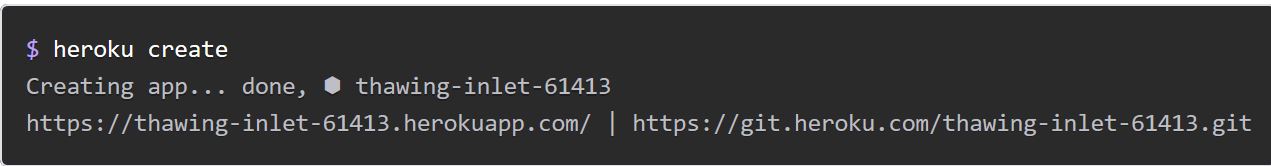
Step4: Navigate to the folder contain settings.py and add “import django\_heroku” to the top of settings.py and add “django\_heroku.settings(locals())” to the bottom of settings.py.



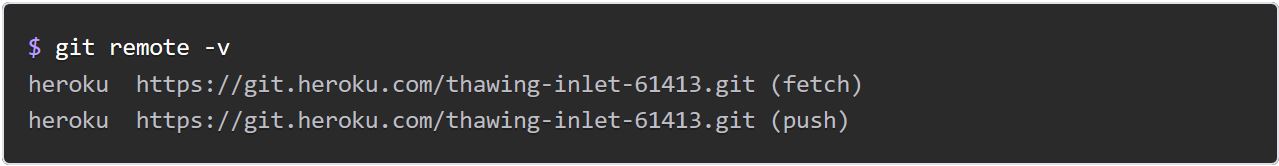
Step5: Initialize a Git repository for the app and commit code to the repository. Navigate to the root of the repository using command prompt



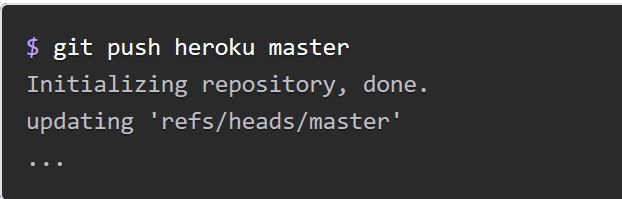
Step6: Create a Heroku remote using “heroku remote”



“git remote -v” allows developers to see if remote “heroku” exists



Step7: Deploy the code



Step8: Apply migrations

“heroku run python enrollmentsystem/manage.py makemigrations”

“heroku run python enrollmentsystem/manage.py migrate”

Step9: Create Superuser in Heroku app

“heroku run python enrollmentsystem/manage.py createsuperuser”

Note:

For more information about how to create Procfile and how to deploy application on Heroku can be found here:

<https://devcenter.heroku.com/articles/django-app-configuration>

<https://devcenter.heroku.com/articles/git>

Open source used in this application can be found in GitHub repositories:

<https://github.com/stephenmcd/django-forms-builder> for the form builder

<https://github.com/suhailvs/django-schools/tree/master/django_school> for the basic structure of the application

**1.2 Sandbox Environment**

Pycharm is used to write or edit the code and Django. The website runs locally through Command prompt/PowerShell(Windows User) and Terminal(Mac User). Bugs can be fixed as soon as errors found.

The code is pushed to Github as appropriate.

**1.3 Production Environment**

As soon as the local project passed all the test cases and approved by the client, the website is published to Heroku. The same URL is used, thus the previous version will be removed upon deployment. The migration is re-apply to construct the models for database hosted by Heroku.

**1.4 Diagrams**

**1.4.1 Web Flow Diagram**

**1.4.2 Entity Relationship Diagram (ERD)**

**1.4.3 Use Case Diagram**

**1.5 Coding and I/O Standards**

**1.5.1 Overview**

The website is developed utilizing the Django web framework and is hosted using the server provided by the client, Victor (the website is hosted on Heroku before final deployment). Python and HTML are used for creating this website. Data is stored in MYSQL database

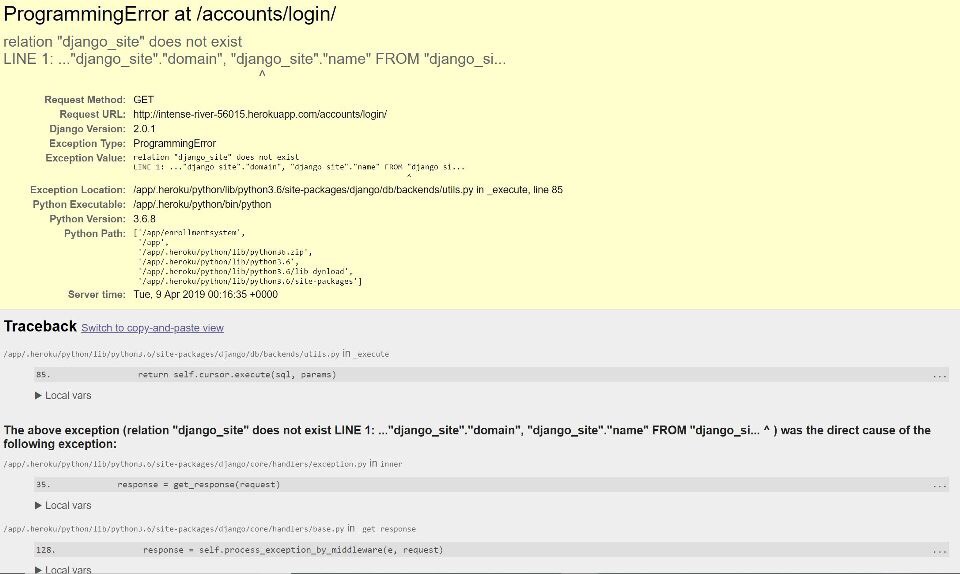
**1.5.2 Logic Layer: Python and Django**

Technology: Python and Django packages

**1.5.3 Presentation Layer: HTML**

**1.6.4 Database Layer: MySQL**

**Common Bugs:**

****

No migration is applied after deployment, thus there is no tables available on Heroku database. See 1.1.3 Step8 to apply migrations.