CSC425 Module 2 Session Long Project – IMS WebApp Prototype

by Tom Parks

Trident University International

CSC425 Module 2 Session Long Project – IMS WebApp Prototype

This brief report outlines the prototype for my IMS WebApp project. This application is an Inventory Management System that is hosted on a web server with a Python backend and a database to store the raw data. We will discuss the basic setup, system requirements, key components, proof of concept and give a current update of the progress at the end.

# Web App Server Setup

There are three applications that will require setup prior to running this application, Python 3.8.5, Django 3.1.5, and a Django extension called crispy\_forms. I am developing this project on a Windows machine; however, I am remoting into my Ubuntu server running the 20.04 build. This machine runs the actual Python development server and will eventually host it on the installed Apache 2 server.

## Python 3.8.5

As stated, this application runs on a Python 3.8.5 backend, so it is recommended that you install the same version. For installation instructions and documentation, please navigate to [www.python.org](http://www.python.org) For Windows users, please click the “Downloads” tab and choose the distribution that is right for you. While writing this, I installed the latest release for Windows (3.9.1) and everything seemed operational.

For Linux based users, you will need to open a terminal and enter the following commands:

sudo apt update

sudo apt install python3.8

After the process completes, make sure you check the version by entering “python3 --version”. Your output should look similar to the below images. This check is a quick test to see if the installation was a success.

Figure 1: Python Version Output in Ubuntu Server 20.04

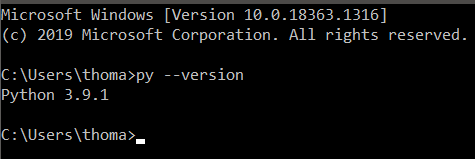
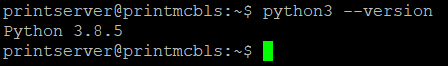


Figure 2: Python Version Output in Window Command Prompt

## Django 3.1.5

Next to install is Django. Again, you can use either Windows or Linux based machines. Both will use pip which is a package-management system written in Python. (Rodriguez, n.d.) To install Django in Windows, its best to go to the source. The Django website (<https://docs.djangoproject.com/en/3.1/howto/windows/>) has well documented step-by-step instructions.

For Linux users, you’ll need a terminal again. Here’s the command:

pip install Django==3.1.5

You can check the version by running “pip3 freeze”. This will produce an output of all installed modules. Here’s an example:

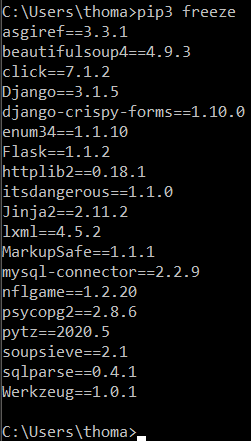


Figure 3: Django Version Check

Lastly, we will install crispy\_forms. To install the extension, enter the following commands in your console:

pip install Django-crispy-forms

If you were keen, you would have seen django-crispy-forms==1.10.0 directly under Django in the list. Use this method to ensure that the application was installed. Now that we’re past that, we can crank up the development server. Run the following command from your console:

/\* Windows \*/

py manage.py runserver

/\* Linux \*/

python3 manage.py runserver

You are now able to access the site through your web browser. I recommend using Google Chrome since that was what I used for development, however, Edge seems to work just fine as well. Just browse to “localhost:8000” or “127.0.0.1:8000”. You will be presented with the login page for the site.

You can login with either the admin account or a test user account. Here are the credentials:

* Admin Account:
  + Email: [admin@ims.webapp.com](mailto:admin@ims.webapp.com)
  + Password: password
* Test User:
  + Email: [testuser@email.com](mailto:testuser@email.com)
  + Password: testing321

# Minimum System Requirements

Python will run on Windows 10, Mac OS 10.9, and various Unix platforms (see <https://docs.python.org/3/using/unix.html> for more info).

# Key Components (functions)

This is an inventory management system, so communication with the database is crucial. With the prototype, I discussed the framework of the system. All of that is in place now, so we can look at it from a higher-level perspective. I will discuss the main functions of the application and how it has progressed for the last iteration.

Starting with the registration screen, there is a visual confirmation that a user has been registered and is awaiting admin approval. I also did not realize that after a user click the submit button from the registration form, an AttributeError exception is thrown. This has been fixed and was due to the fact that the redirect was sending them to a page that requires login. Because account registration requires admin approval, the user cannot login to view the site. However, after viewing the admin site, the user was registered.

The next big push was the Create page. A user can now add new End Items to their account(s). The only issue with this page is a problem with the user account query. Currently a user can see everyone’s equipment accounts instead of just their own. Otherwise, this page is fully operational.

I worked on the Update page as well, although I didn’t make much progress. I need to research how to push update queries to the database with Django.

The last changes that were made were somewhat cosmetic. The equipment listing in the right-hand column on the SnapShot page is now available in all pages. I have also included a “logged in as” section in the top right of the navbar. For the admin account, I have included an extra link under the “Account” section on the vertical navbar on the left side of the screen. Unfortunately, I have not been able to find a good way to route this link to the actual admin page. When I do, I will add it.

Lastly, I added a profile section in admin portion. I was able to create profiles with profile pics. I made a form (that still needs work) to allow users to update their profile and I linked it to the “Profile” link in the vertical navbar.

# Status

Still quite a bit of work to do. I did not reach my goals this time around, however, I’m still pushing hard. This week I will try to conquer the Update form and fix the queryset for the available accounts on the Create page.

If I have time, I will work on the removal portion and push to create at least one report. Unfortunately, I don’t think I can make it that far, but maybe luck will be on my side.

References

Last Name, F. M. (Year). *Book Title.* City Name: Publisher Name.

Rodriguez, I. (n.d.). *What Is Pip? A Guide for New Pythonistas*. Retrieved from Real Python: https://realpython.com/what-is-pip/