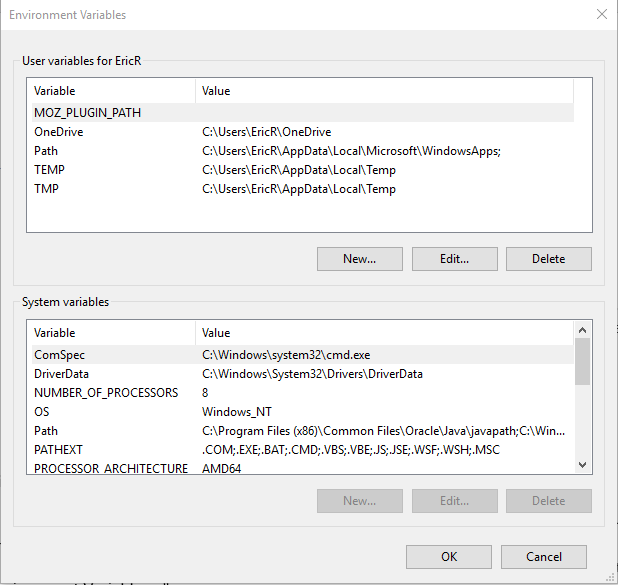
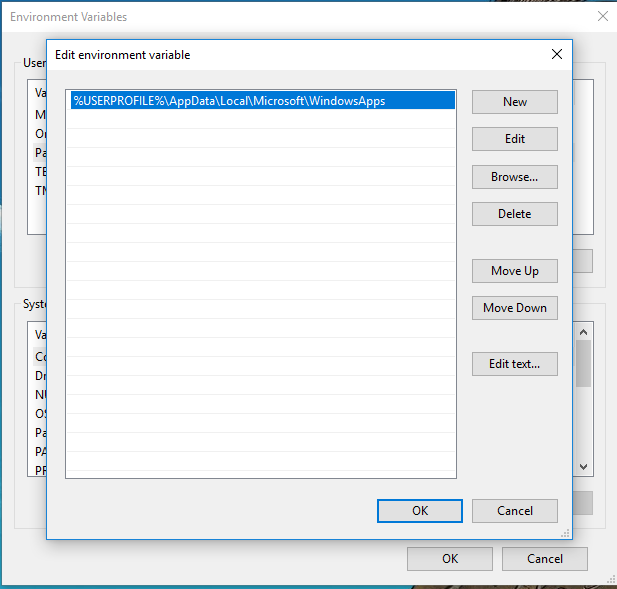
# Installing and Running Django on Windows 10

1. First, we must download and install Python.
   1. Go to <https://www.python.org/ftp/python/3.7.1/python-3.7.1-amd64.exe> and download Python 3.7.1.
   2. Once it is done downloading run the launcher and install. For this tutorial use the default installation path.
2. Now that Python is installed we need to add it, and its Pip utility, to our PATH. By doing this we can run Python and Pip on the command line without specifying their whole path.
   1. In Windows 10 type in the search bar on the taskbar “PATH” and select “edit the system environment variables”. The below window should now be open.



* 1. Select “Path” and then click the “Edit…” button. The window below should open.



* 1. In this window click the “New” button and input the full path to your Python installation and press the enter key.
     1. The default python location is:
     2. "C:\Users\< user name >\AppData\Local\Programs\Python\Python37\”
  2. Click “New” again and add the full path to Pip and other Python utilities.
     1. This location is by default:
     2. "C:\Users\< user name >\AppData\Local\Programs\Python\Python37\Scripts".
  3. Now that both paths are input click “OK” on the two windows.

1. Now we need to setup a virtual environment for our Django installation to be setup in.
   1. Open up CMD (typing “CMD” in the search and pressing enter).
   2. In CMD use the command “pip install virtualenv” to install in Python virtualenv.
   3. Setup the virtual environment:
      1. Create a new folder somewhere that you know the path to.
      2. Setup the virtual environment using this command:
         1. “virutalenv <path to where you want the virtual environment>”
2. Activate the virtual environment
   1. In CMD, change to the virtual environment directory. (use “cd < path >”)
   2. Then open the “Scripts” folder (cd Scripts)
   3. Now use the command “activate” to activate the virtual environment
3. Now that the virtual environment is activated we need to install Django and Boto3 in that virtual environment.
   1. Run the command “pip install Django” and “pip install boto3”
4. Change directory in CMD to your Django project. (The proper folder will have the “manage.py” in it.)
5. Run the command “pip -r requirements.txt” (this installs all the used libraries)
6. You will need to make a “keys.txt” file. Our program uses AWS credentials from an external file not on the GitHub to use AWS functions so transcribe and comprehend calls will not work unless we provide you with that file. This is a “keys.txt” file that has the credentials in the format “{"aws\_access\_key\_id":"”,aws\_secret\_access\_key":""}” (just remove outer quotation marks and fill in the blanks with the keys, this is just a dictionary) and is under the directory “SeniorDesign-SpeechCapture\Django-SpeechCapture\webapp\”
7. Need to change the bucket\_name variable in views.py to the name of your bucket in S3, also it is a good idea to make the bucket public so that our app can access the files (also need to change S# link in Django-SpeechCapture\webapp\templates\webapp\transcript.html to new bucket URL)
8. NLTK files are needed to run the analysis. To install these use command “python -m nltk.downloader all” (This may take a while as it is over 3GB of data.)
9. Run the Django application on your local machine by using the following command in CMD while in the Django project folder:
   1. “python manage.py runserver”
10. Now that our Django application is running we can access it by opening our browser and going to <http://127.0.0.1:8000/>
11. To end the program, click “ctrl-c” while focused on CMD

# Notes on Django and Our Program

1. In Django-SpeechCapture/webapp/urls.py file we describe all of the URLs of our application

a) e.g. path('transcript/', views.transcript1, name='transcript') describes the URL “<http://127.0.0.1:8000/>transcript/” and it will run the function transcript1 from views.py

2. In Django-SpeechCapture/webapp/views.py we describe all of the “views/pages” of our webapp

a) We described in 1a how we choose which function corresponds to which URL

b) The function “index(request)” in views.py takes a http request and returns a render (return render(request, 'webapp/home.html', {'directory': directory\_new, 'directory2': directory\_old})) that has the html to be displayed as well as the variables that will be passed into the HTML (to be able to dynamically show variables on the webpage)

4. To deploy to AWS Elastic beanstalk you need to use the Elastic Beanstalk CLI. More information here: <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-django.html>

5. When deploying to elastic beanstalk the command in “Django-SpeechCapture\.ebextensions” there is a “django.config” which has a command in it that you will need to change the red portion of “aws s3 cp s3://elasticbeanstalk-us-west-2-614262400641/nltk\_data /usr/local/share/nltk\_data/ --recursive” changing it to the name of the bucket/folder in your S3 that has a copy of the NLTK data you downloaded in the steps above (the default location for NLTK data on Windows 10 is “C:\Users\<USERNAME>\AppData\Roaming\nltk\_data”, change the username to the name of the account you are on, just copy from your local disk to S3 and change the path like said above.)

6. To edit javascript files and still keep your application working when uploaded and hosted on AWS edit the javascript files in directory “Django-SpeechCapture\static\_files\” then before deploying run command “python manage.py collectstatic” to update the JS files in “Django-SpeechCapture\static\”