

# SymCheck Installation Manual

## About: What is SymCheck?

---

SymCheck is a self-diagnosing web application that allows users to upload a photo of a potential ailment that he/she may have to a webpage, in order to receive an estimated diagnosis probability on what that illness/ailment may be. The application itself utilizes a machine-learning algorithm for image classification with an accuracy of about 88%, and while it is in no way a *complete replacement* for physical medical diagnoses or care from trained medical professionals, it strives to enable users to save time and money by using the application to determine if a physical visit to a healthcare facility can be avoided.

## Accessing SymCheck via Cloud-Hosting (Desktop and Mobile)

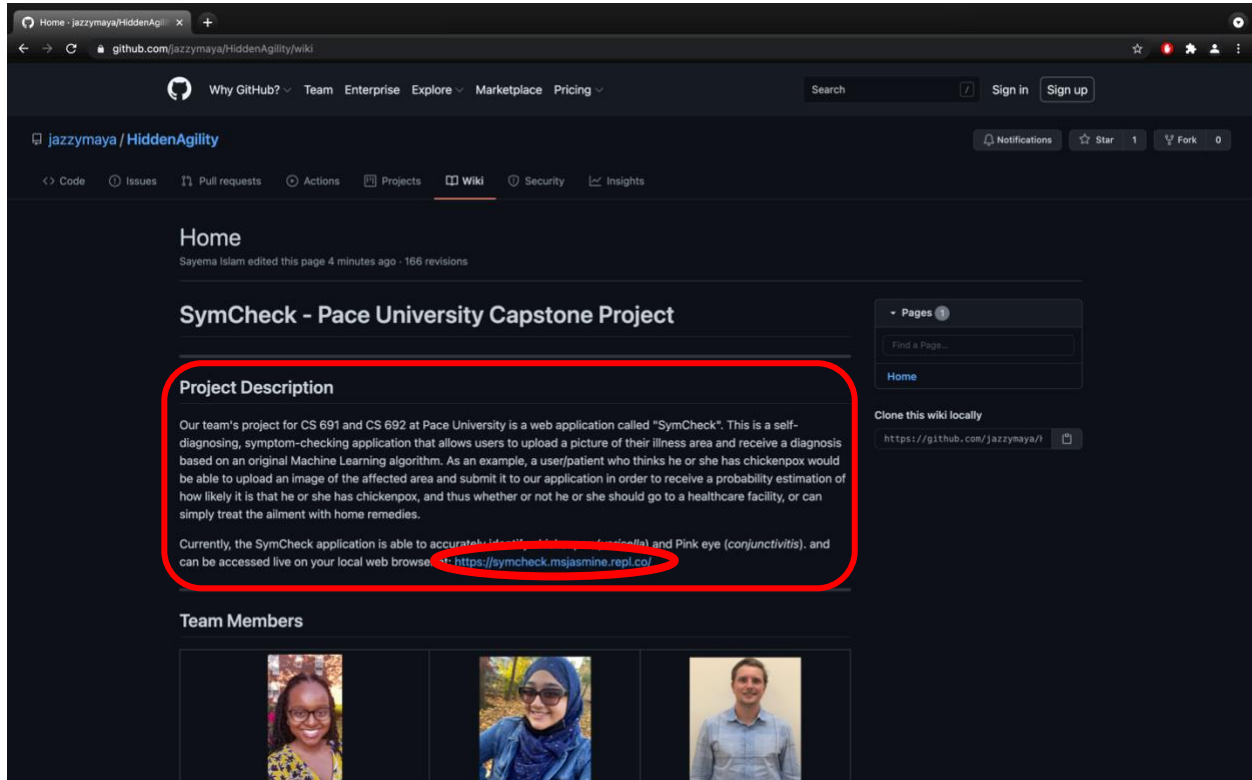
---

**Updated 04/24/2021**

Our team's SymCheck application can be accessed **without any additional installations** through its live Cloud instance hosted via AWS or repl.it, respectively, at: <http://symcheckapp.com/>\* or <https://symcheck.msjasmine.repl.co/>, or by simply navigating to the SymCheck wiki page on GitHub at: <https://github.com/jazzymaya/HiddenAgility/wiki> and then clicking a hyperlink to the website under the "Project Description" section.

\*Please note that after May 10<sup>th</sup>, 2021, the URL hosted on AWS will be deprecated due to cost concerns, but the same, complete SymCheck application can continue to be accessed via the repl.it instance.

The link to our cloud-hosted repl.it instance for SymCheck can be accessed from the Project Description section as shown below:



## Accessing SymCheck via Localhost

The complete SymCheck web application can also be successfully run on one's local machine via localhost, and requires the following dependencies and steps to install/run all dependencies needed for the project:

- **Python 3.5 – 3.8**
  - Please note that only *these specific* versions of Python work with the application as it is a dependency outlined in version 2.0 of TensorFlow (required for SymCheck's machine-learning-based algorithm of image classification)
  - The most recent version of Python, 3.9.0 is *not yet supported* by TensorFlow 2.0

- **Python pip 20.2.4**
  - This is a package-management system used to install and manage various software packages in Python
  - This is the most recent version of pip available, and will be used to install TensorFlow, and Pillow.
- **Django 3.1.3**
  - This is the most recent version of Django, a Python-based web framework used for the SymCheck application, available for use. This framework must also be installed on the user's computer in order to run the server used to access the SymCheck application via localhost.
- **Python Pillow 8.0.1**
  - Pillow is a user-friendly version of PIL, Python Imaging Library. This is another required install as it allows any images submitted to the SymCheck application's local media folder to be identified
- **TensorFlow 2.0**
  - This is the most recent version of TensorFlow available for use, TensorFlow 2.3.0, and must be installed to support the machine-learning-based classification of images for diagnosis

## Installation on MacOSx

(for instructions on installation for Windows, skip to page 9)

---

\*\*\*This installation guide assumes that you have Homebrew installed on your machine. If this is not the case, please go to <https://brew.sh/> and follow the steps to install Homebrew, a user-friendly package manager and software installation system for MacOS and Linux.

### Step 1: Install Python

Python is installed on all new Macs but it is an older version (Python 2.0) that is not supported by TensorFlow 2.0. Thus, we start by opening the Terminal app (you can do a spotlight search of Terminal if you are not sure where this application is located by pressing COMMAND + SPACE and typing in "Terminal"). Normally, we would install Python by simply entering the following command:

```
$ brew install python3
```

**DO NOT USE THIS COMMAND.** Because Homebrew is open-source, and by default installs the most up-to-date software, this will install Python 3.9.0 and we DO NOT WANT THIS as it will not work with TensorFlow 2.0. So instead, we use pyenv of Homebrew which is essentially a collection of different versions of Python to install the version we want:

```
$ brew install pyenv  
$ pyenv install <version>
```

For the purposes of this tutorial, we will be installing version 3.8.0 (insert the version in place of “<version>”).

```
$ pyenv local 3.8.0
```

The above command essentially sets the version that you desire. If you wish to list all available versions, simply use the command:

```
$ pyenv versions
```

You must also, however, update *PATH* to override the default version of Python on your machine using:

```
PATH="$~/.pyenv/versions/3.8.0/bin:${PATH}"
```

For any additional information on installing a specific version of Python using Homebrew, go to <https://gist.github.com/Bouke/11261620>.

---

Now, in the event that you have **trouble switching between versions of Python** (this happens when a user might have many different versions of Python installed on his/her computer in different locations, we recommend removing these versions (don't worry, you will, by default, not be able to remove the version of Python that comes with your Mac). To do this, you must first locate where you have Python 3 installed using the command:

```
$ which python3
```

Now, take the paths of these locations and delete the Python installations using:

```
$ sudo rm -R <pythonDirectoryPath>
```

And now you can go back and install a specific version of Python as outlined above.

---

Now that you have installed Python 3 on your machine, test the installation by running Python. Note, we must type in “python3” whenever running Python, otherwise the system will default to running the factory version of Python 2. See the screenshot below to confirm that you have successfully installed a version of Python 3:

```
→ ~ python3
Python 3.8.2 (default, Oct  2 2020, 10:45:42)
[Clang 12.0.0 (clang-1200.0.32.27)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Great! Now exit Python3 by entering:

```
>>>quit()
```

And now we are ready to install the pip package manager.

## Step 2: Install pip

Enter the following command to install pip:

```
$ sudo easy_install pip
```

## Step 3: Install a virtual environment for Python

Type the following command in your terminal after installing pip:

```
$ sudo pip install virtualenv
```

Now we need to create a folder, to hold all the files needed for our virtual environment so first, make a directory for your virtual environment (you can name it anything you like, for the purposes of this tutorial, it has been named “pyenv”), and then install the necessary folders as such:

```
$ mkdir pyenv
$ virtualenv pyenv
```

Now, go into the virtual environment folder with “cd” (change directory) and enter the following command to activate/start the virtual environment:

```
$ source bin/activate
```

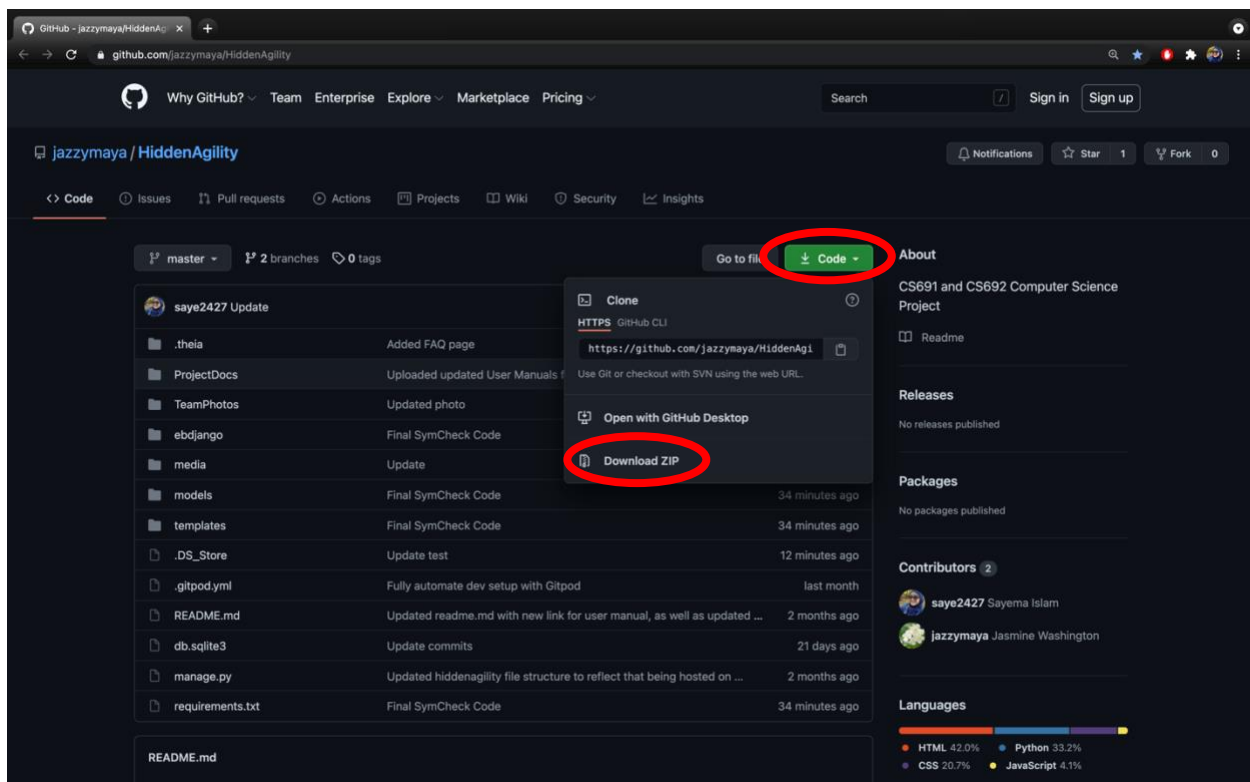
You should now see the name of your virtual environment as the prompt in your terminal as such:

```
→ ~ cd pyenv
→ pyenv source bin/activate
(pyenv) → pyenv
```

Note that you can exit the virtual environment at any time by simply typing the command “deactivate”. Go ahead and deactivate the virtual environment for now, and navigate to whichever directory you wish to locally save all of SymCheck’s application files.

## Step 4: Download SymCheck Files from GitHub

In order to run the SymCheck application, you will need to have a copy of the application’s files on your local machine. This can be done by navigating to HiddenAgility’s GitHub repository for SymCheck at: <https://github.com/jazzymaya/HiddenAgility> and clicking the green download button above the list of files, and clicking “Download ZIP” as shown below:



You can then unzip the downloaded folder, and voila! You now have all the required SymCheck files available on your computer.

Alternatively, you can pull a copy of the SymCheck code files from GitHub directly through your terminal using the following command:

```
$ git clone https://github.com/jazzymaya/HiddenAgility
```

This will save the SymCheck files to whatever directory you are in when executing said command:

```
Last login: Sat Nov 21 17:20:01 on ttys000
[oh-my-zsh] plugin 'zsh-nvm' not found
➔ ~
Applications  Documents  Library      Music         Public        get-pip.py    pyvenv
Desktop      Downloads  Movies       Pictures      VirtualBox   VMs          opt          wekafiles
➔ ~
➔ ~ cd downloads
➔ downloads git clone https://github.com/jazzymaya/HiddenAgility
Cloning into 'HiddenAgility'...
remote: Enumerating objects: 177, done.
remote: Counting objects: 100% (177/177), done.
remote: Compressing objects: 100% (145/145), done.
remote: Total 191 (delta 57), reused 188 (delta 23), pack-reused 14
Receiving objects: 100% (191/191), 79.85 MiB | 19.23 MiB/s, done.
Resolving deltas: 100% (57/57), done.
➔ downloads ls
HiddenAgility
➔ downloads
```

Now with the SymCheck files available on your local machine, we are ready to go forth with running the application!

### Step 5: Installing all other SymCheck Requirements (Django, TensorFlow, Pillow) using requirements.txt

Navigate to where you have saved your virtual environment, activate it, and then with your virtual environment activated, navigate back to and enter your folder with all of SymCheck's files, and enter the command:

```
(pyvenv) $ python3 -m pip install -r requirements.txt
```

Great! You now have all the dependencies necessary for running SymCheck installed within your virtual environment. Now just one more step before being able to run the application...

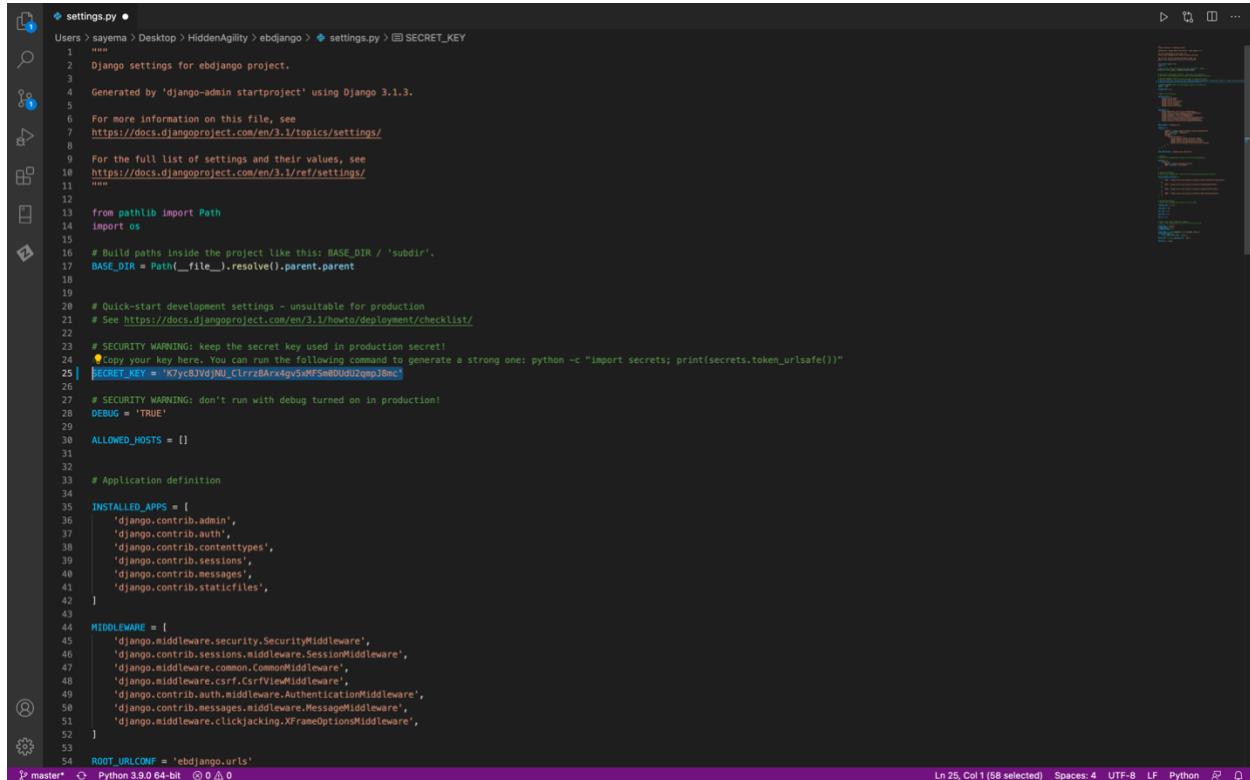
### Step 6: Generate a “Secret Code” for Securely Running Django

Within your terminal, enter the following command:

```
(pvenv) $ python3 -c "import secrets; print(secrets.token_urlsafe())"
```

```
(pyvenv) ➔ HiddenAgility git:(master) python3 -c "import secrets; print(secrets.token_urlsafe())"
K7yc8JVdjNU_Clrz8Arx4gy5xMF5m0DUdU2qmpJ8mc
(pyvenv) ➔ HiddenAgility git:(master)
```

Copy the code generated, and open the “settings.py” file located within the “ebdjango” folder of SymCheck's files (i.e. HiddenAgility > ebdjango > settings.py) in a code editor of your choice. Go to line 25 of the code where it says SECRET\_CODE= '', paste the code within the single quotes, and save the file as shown on the next page:



```

1  # Django settings for ebdjango project.
2
3  Generated by 'django-admin startproject' using Django 3.1.3.
4
5  For more information on this file, see
6  https://docs.djangoproject.com/en/3.1/topics/settings/
7
8  For the full list of settings and their values, see
9  https://docs.djangoproject.com/en/3.1/ref/settings/
10
11
12
13 from pathlib import Path
14 import os
15
16 # Build paths inside the project like this: BASE_DIR / 'subdir'.
17 BASE_DIR = Path(__file__).resolve().parent.parent
18
19
20 # Quick-start development settings - unsuitable for production
21 # See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/
22
23 # SECURITY WARNING: keep the secret key used in production secret!
24 # Copy your key here. You can run the following command to generate a strong one: python -c "import secrets; print(secrets.token_urlsafe())"
25 SECRET_KEY = 'K7yc3VdJNl_Clrz8Arx4q5w$W5m0k0z2qp1Bec'
26
27 # SECURITY WARNING: don't run with debug turned on in production!
28 DEBUG = 'True'
29
30 ALLOWED_HOSTS = []
31
32
33 # Application definition
34
35 INSTALLED_APPS = [
36     'django.contrib.admin',
37     'django.contrib.auth',
38     'django.contrib.contenttypes',
39     'django.contrib.sessions',
40     'django.contrib.messages',
41     'django.contrib.staticfiles',
42 ]
43
44 MIDDLEWARE = [
45     'django.middleware.security.SecurityMiddleware',
46     'django.contrib.sessions.middleware.SessionMiddleware',
47     'django.middleware.common.CommonMiddleware',
48     'django.middleware.csrf.CsrfViewMiddleware',
49     'django.contrib.auth.middleware.AuthenticationMiddleware',
50     'django.contrib.messages.middleware.MessageMiddleware',
51     'django.middleware.clickjacking.XFrameOptionsMiddleware',
52 ]
53
54 ROOT_URLCONF = 'ebdjango.urls'

```

Excellent! Now you are ready to run the SymCheck application!

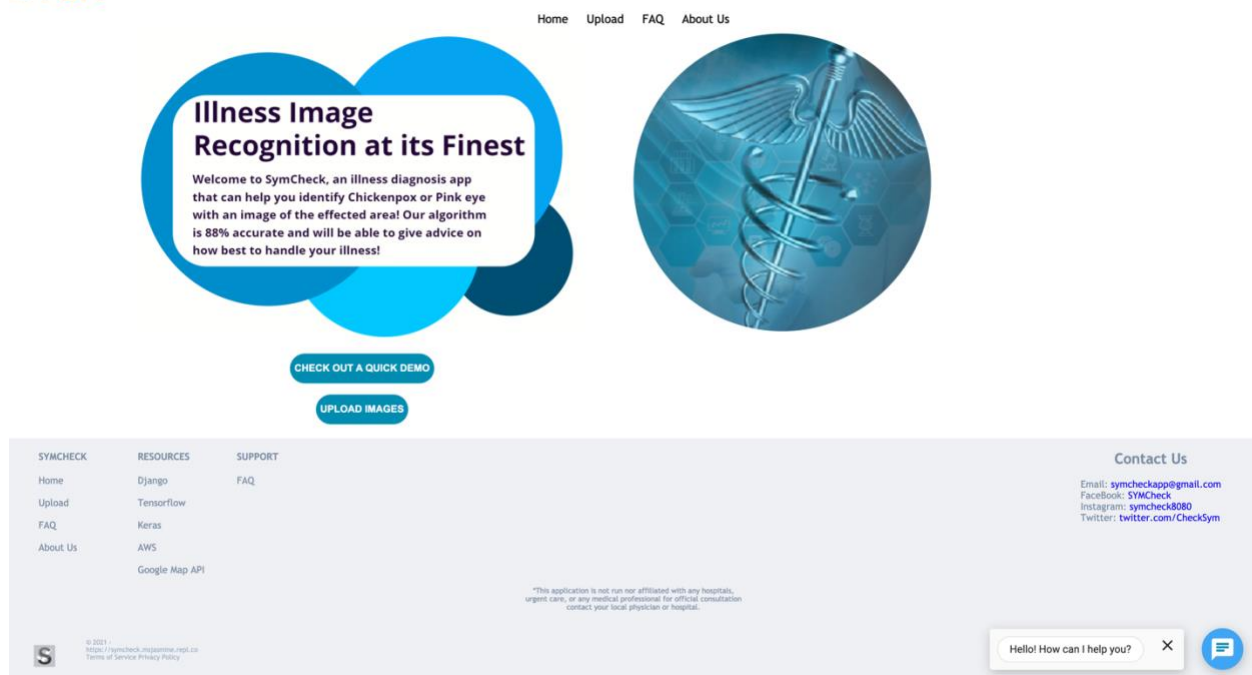
## Step 7: Running SymCheck via localhost

Ensuring that you are still within the root folder containing all of SymCheck's program files, enter the following command into your terminal:

```
(pyvenv) $ python3 manage.py runserver
```

Then navigate to <http://localhost:8000> or <http://127.0.0.1:8000> in your browser, and you should see the welcome page of SymCheck as shown on the next page:





**Congratulations! You can now use the SymCheck web application on your Mac to receive your own image-based diagnosis!**

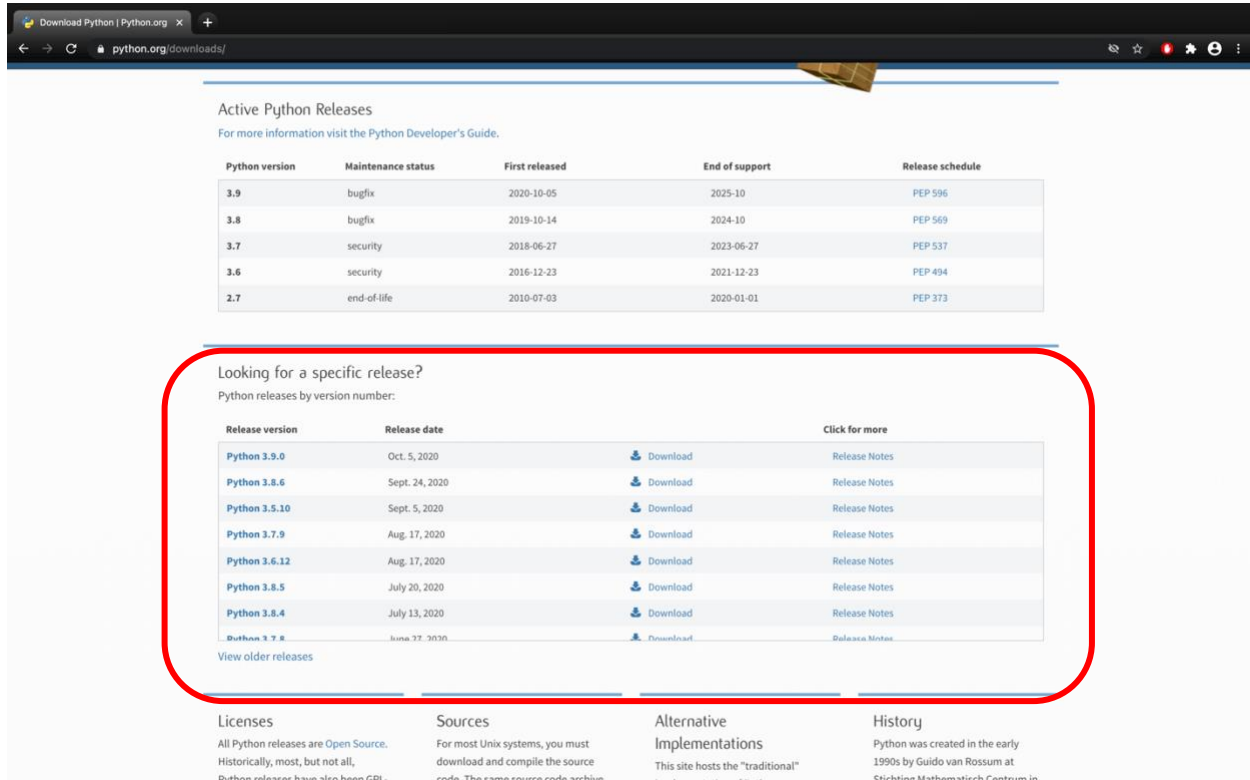
Check out our *SymCheck User Manual* next to learn more about how to navigate the website. And we hope our application serves you well in allowing you to make more informed decisions about your own care.

## Installation on Windows

\*\*\*Note: The setup of the environment for the SymCheck Application is more or less the same for Windows as it is for MacOSx, save the first step for Python 3 installation, and slightly different commands for command line vs. MacOSx's Terminal.

### Step 1: Install Python

To install Python on your Windows machine, navigate to <https://www.python.org/downloads/>. Be sure to download the files for a version of Python *between 3.5 and 3.8* by scrolling down to where it says "Looking for a specific release?"



Active Python Releases

For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.9	bugfix	2020-10-05	2025-10	PEP 596
3.8	bugfix	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537
3.6	security	2016-12-23	2021-12-23	PEP 494
2.7	end-of-life	2010-07-03	2020-01-01	PEP 373

Looking for a specific release?

Python releases by version number:

Release version	Release date	Download	Click for more
Python 3.9.0	Oct. 5, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
Python 3.8.6	Sept. 24, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
Python 3.5.10	Sept. 5, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
Python 3.7.9	Aug. 17, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
Python 3.6.12	Aug. 17, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
Python 3.8.5	July 20, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
Python 3.8.4	July 13, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
Python 3.7.8	June 27, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>

[View older releases](#)

**Licenses**  
All Python releases are [Open Source](#). Historically, most, but not all, Python releases have also been GPL.

**Sources**  
For most Unix systems, you must download and compile the source code. The same source code archive

**Alternative Implementations**  
This site hosts the "traditional" [implementation of Python](#)

**History**  
Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum in

Download the files for Python 3 by clicking the Download link.  
**Remember Python 3.9.0 is not supported by the latest version of TensorFlow!**

Once you have installed Python 3 on your machine, test the installation by running Python—simply type “python” into your command line as Mac users would type “python3”. You should see the same command line prompt beginning with “>>>”

```
→ ~ python3
Python 3.8.2 (default, Oct 2 2020, 10:45:42)
[Clang 12.0.0 (clang-1200.0.32.27)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Great! Now exit Python by entering:

```
>>>quit()
```

And now we are ready to install the pip package manager.

## Step 2: Install pip

Enter the following command to install pip:

```
> python -m pip install -U pip
```

### Step 3: Install a virtual environment for Python

Type the following command in your terminal after installing pip:

```
> pip install virtualenv
```

Now we need to create a folder, to hold all the files needed for our virtual environment so first, make a directory for your virtual environment (you can name it anything you like, for the purposes of this tutorial, it has been named “pyenv”), and then install the necessary folders as such:

```
> mkdir pyenv  
> virtualenv pyenv
```

Now, go into the virtual environment folder with “cd” (change directory) and enter the following commands to activate/start the virtual environment:

```
> cd Scripts  
> activate
```

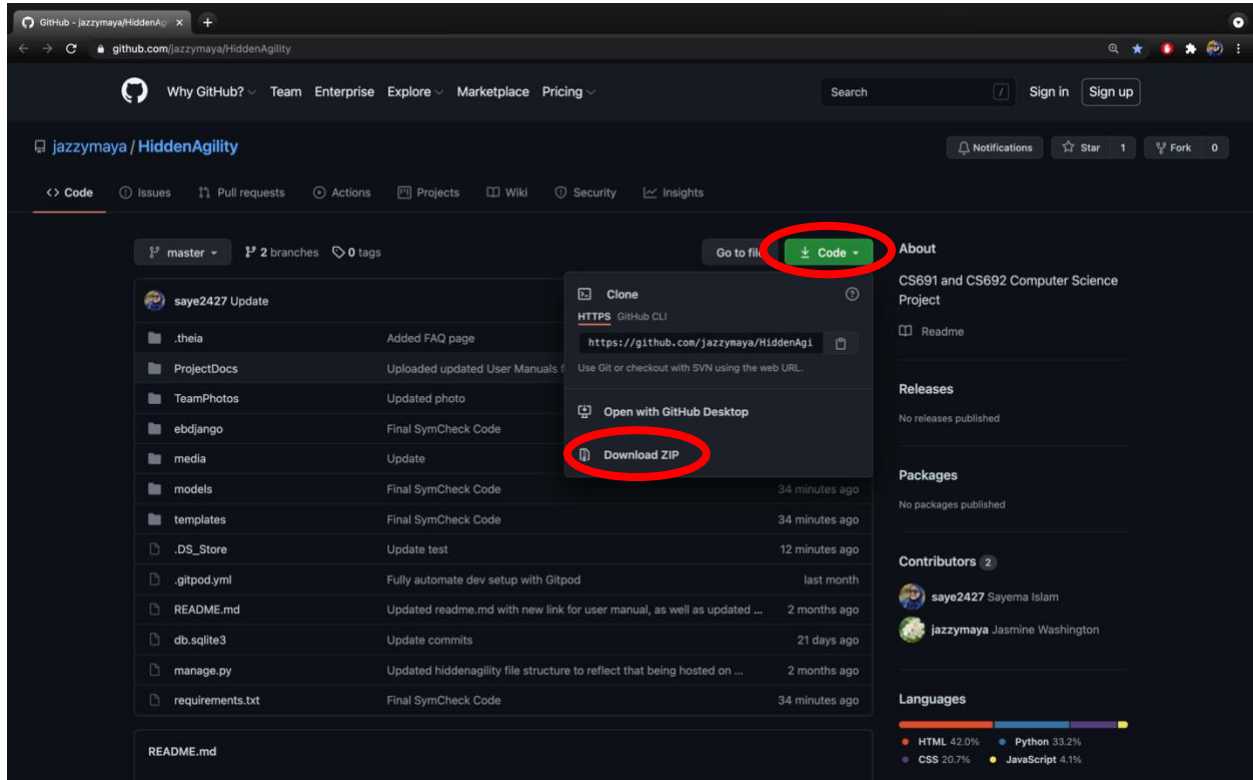
You should now see the name of your virtual environment as the prompt in your terminal as such:

```
(pyenv) [Folder_path]>
```

Note that you can exit the virtual environment at any time by simply typing the command “deactivate”. Go ahead and deactivate the virtual environment for now, and navigate to whichever directory you wish to locally save all of SymCheck’s application files.

### Step 4: Download SymCheck Files from GitHub

In order to run the SymCheck application, you will need to have a copy of the application’s files on your local machine. This can be done by navigating to HiddenAgility’s GitHub repository for SymCheck at: <https://github.com/jazzymaya/HiddenAgility> and clicking the green download button above the list of files, and clicking “Download ZIP” as shown on the next page:



You can then unzip the downloaded folder, and voila! You now have all the required SymCheck files available on your computer.

Alternatively, you can pull a copy of the SymCheck code files from GitHub directly through your terminal using the following command:

```
> git clone https://github.com/jazzymaya/HiddenAgility
```

This will save the SymCheck files to whatever directory you are in when executing said command:

```
Last login: Sat Nov 21 17:20:01 on ttys000
[oh-my-zsh] plugin 'zsh-nvm' not found
➔ ~ ls
Applications  Documents  Library  Music  Public  get-pip.py  pyvenv
Desktop      Downloads  Movies   Pictures  VirtualBox VMs  opt  wekafiles
➔ ~ cd downloads
➔ downloads git clone https://github.com/jazzymaya/HiddenAgility
Cloning into 'HiddenAgility'...
remote: Enumerating objects: 177, done.
remote: Counting objects: 100% (177/177), done.
remote: Compressing objects: 100% (145/145), done.
remote: Total 191 (delta 57), reused 108 (delta 23), pack-reused 14
Receiving objects: 100% (191/191), 70.85 MiB | 19.23 MiB/s, done.
Resolving deltas: 100% (57/57), done.
➔ downloads ls
HiddenAgility
➔ downloads
```

Now with the SymCheck files available on your local machine, we are ready to go forth with running the application!



### **Step 5: Installing all other SymCheck Requirements (Django, TensorFlow, Pillow) using requirements.txt**

Navigate to where you have saved your virtual environment, activate it, and then with your virtual environment activated, navigate back to and enter your folder with all of SymCheck's files, and enter the command:

```
(pyvenv) > python3 -m pip install -r requirements.txt
```

Great! You now have all the dependencies necessary for running SymCheck installed within your virtual environment. Now just one more step before being able to run the application...

### **Step 6: Generate a "Secret Code" for Securely Running Django**

Within your terminal, enter the following command:

```
(pvenv) > python3 -c "import secrets; print(secrets.token_urlsafe())"
```

Copy the code generated, and open the "settings.py" file located within the "ebdjango" folder of SymCheck's files (HiddenAgility > ebdjango > settings.py) in a code editor of your choice. Go to line 25 of the code where it says SECRET\_CODE= ' ', and paste the code within the single quotes, and save the file as shown below:

Excellent! Now you are ready to run the SymCheck application!

### **Step 7: Running SymCheck via localhost**

Ensuring that you are still within the root folder containing all of SymCheck's program files, enter the following command into your terminal:

```
(pyvenv) > python3 manage.py runserver
```

Then navigate to <http://localhost:8000> or <http://127.0.0.1:8000> in your browser, and you should see the welcome page of SymCheck as shown on the next page:

## Illness Image Recognition at its Finest

Welcome to SymCheck, an illness diagnosis app that can help you identify Chickenpox or Pink eye with an image of the effected area! Our algorithm is 88% accurate and will be able to give advice on how best to handle your illness!


[CHECK OUT A QUICK DEMO](#)
[UPLOAD IMAGES](#)

<b>SYMCHECK</b>	<b>RESOURCES</b>	<b>SUPPORT</b>
<a href="#">Home</a>	<a href="#">Django</a>	<a href="#">FAQ</a>
<a href="#">Upload</a>	<a href="#">Tensorflow</a>	
<a href="#">FAQ</a>	<a href="#">Keras</a>	
<a href="#">About Us</a>	<a href="#">AWS</a>	
	<a href="#">Google Map API</a>	

### Contact Us

Email: [symcheckapp@gmail.com](mailto:symcheckapp@gmail.com)  
 Facebook: [SYMCheck](#)  
 Instagram: [symcheck8080](#)  
 Twitter: [twitter.com/CheckSym](#)

\*This application is not run nor affiliated with any hospitals, urgent care, or any medical professional for official consultation. contact your local physician or hospital.

© 2021 - <https://symcheck.mqgame.net>  
 Terms of Service Privacy Policy



**Congratulations! You can now use the SymCheck web application on your Mac to receive your own image-based diagnosis!**

Check out our *SymCheck User Manual* next to learn more about how to navigate the website. And we hope our application serves you well in allowing you to make more informed decisions about your own care.