MOVIE ANALYSIS BASED ON OMDB DATABASE

Dagmara Zadrożna & Dominika Sawicka
PGDiP. Big Data. Data Engineering

MAY 2021

CONTACTS

DAGMARA ZADROŻNA
Author

DOMINIKA SAWICKA
Author

CONTENTS

IMAGES

1 INTRODUCTION

Movies are great way of visual communication, they use pictures and sounds to tell stories, ideas, feelings, perceptions or simply educate. There is no denying, that the internet has changed the movie industry significantly. With online activities becoming so integrated in people's daily routines, it can be difficult to remember times when movies were seen only on TV or in cinema. Today, we can reach to any website, social media or online streaming service to watch any movie on demand.

Netflix, HBO, Disney+ are just a few of those service providers. What goes behind all those streaming services are data and databases. Currently there are many online movie databases available for users. They include information about cast, production crew, diagrams, summaries, review and personal biographies. The most popular being the Internet Movie Database (IMDb), the Open Media Database (OMDb), TVDB, Rotten Tomatoes, the Complete Index To World Film (CITWF) and Internet Movie Firearms Database (IMFDb).

2 PROJECT SPECIFICATION

The aim of this project is to analyse specific number of movies based on certain attributes like Title, Year, Runtime, Genre, Director, Actors, IMDb ratings, etc. and visualise this information in easy and user-friendly manner. Project functionality allows the user to **see up to top 100 movie posters** and expand their types based earlier mentioned attributes. The data will be then visualised in form of **sortable table** or **graphical diagrams** to allow the user for better information interpretation. Use of '**random selection'** will allow for movie selection for the night and any issue regarding page can be raised through **contact us** page.

For the purpose of this project, we made data analysis of movies using one of the most frequently used data science programming language: Python with django framework, HTML, JavaScript, CSS and supporting it database engine SQLite3.

Outline of the project workflow can be seen below.

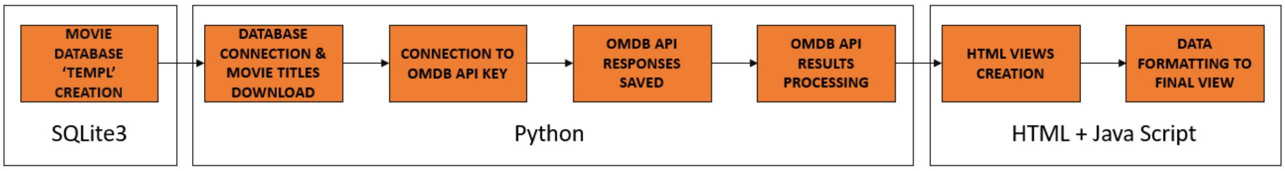

MOVIE DATABASE 'TEMPL' CREATION
DATABASE CONNECTION & MOVIE TITLES DOWNLOAD
CONNECTION TO OMDB API KEY
OMDB API RESPONSES SAVED
OMDB API RESULTS PROCESSING
HTML VIEWS CREATION
DATA FORMATTING TO FINAL VIEW
SQLite3
Python
HTML + Java Script

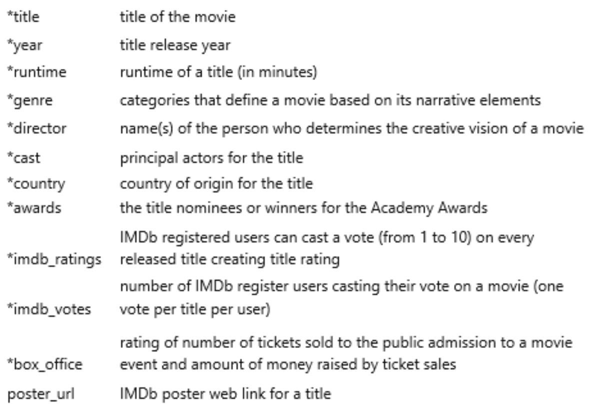
Image1: Project workflow

2.1 DATA SOURCES

Analysis was performed based on the Open Media Database (OMDb) API (1) with some additional information coming from IMDb database (2).

OMDb API is a crowdsourced web service to obtain information similar to that of IMDb's website. It retains all IMDb's features of listings and added bonuses while adding in the feature of allowing anyone to edit the entries. This allows it to potentially be more up to date than IMDb and for new information about any release to be added swiftly. (3)

Project database contains top 100 movie positions, contains following attributes (* primary attributes):



*title	title of the movie
*year	title release year
*runtime	runtime of a title (in minutes)
*genre	categories that define a movie based on its narrative elements
*director	name(s) of the person who determines the creative vision of a movie
*cast	principal actors for the title
*country	country of origin for the title
*awards	the title nominees or winners for the Academy Awards
*imdb_ratings	IMDb registered users can cast a vote (from 1 to 10) on every released title creating title rating
*imdb_votes	number of IMDb register users casting their vote on a movie (one vote per title per user)
*box_office	rating of number of tickets sold to the public admission to a movie event and amount of money raised by ticket sales
poster_url	IMDb poster web link for a title

Image2: Outline of database parameters.

Subsets of OMDb data are available for access to customers for personal and non-commercial use. You can hold local copies of this data, and it is subject to OMDb terms and conditions (4).

3 SOFTWARE REQUIREMENTS

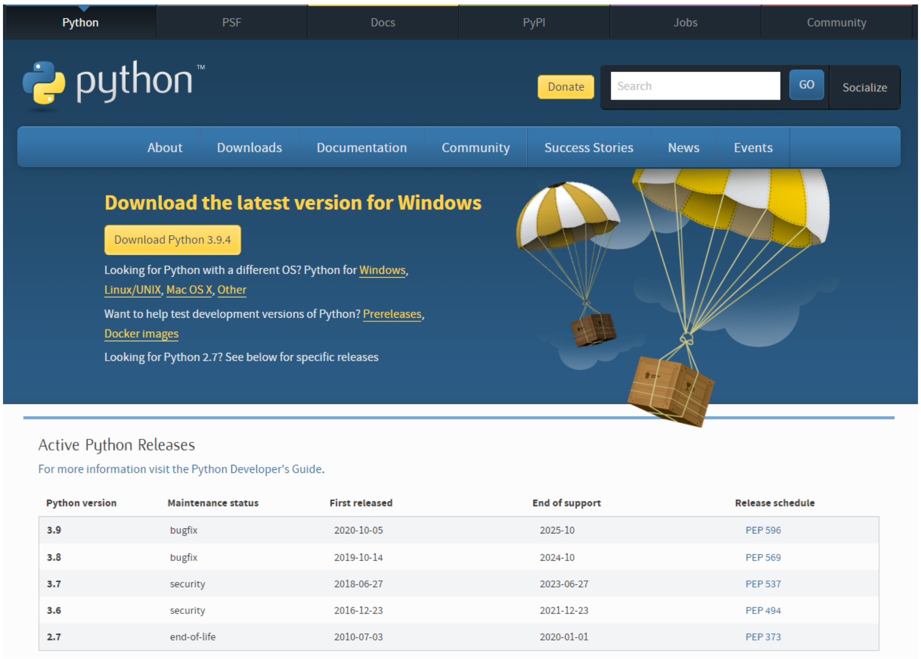
This section identifies particular software requirements used to execute the project code. It is assumed that reader will already have Windows based Operating System and latest Web Browsers installed on their hardware of choice. Below you will find details outlining software download and installation guidance.

3.1 PYTHON 3.6.2

Python is a dynamic object-oriented programming language that can be used for many kinds of software development. It offers strong support for integration with other languages and tools and comes with extensive standard libraries.

a. Click Python Download (5)

b. The following page will appear in your browser, scroll down to **'Looking for a specific release?'** to download Python version 3.6.2.



Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Prereleases, Docker images

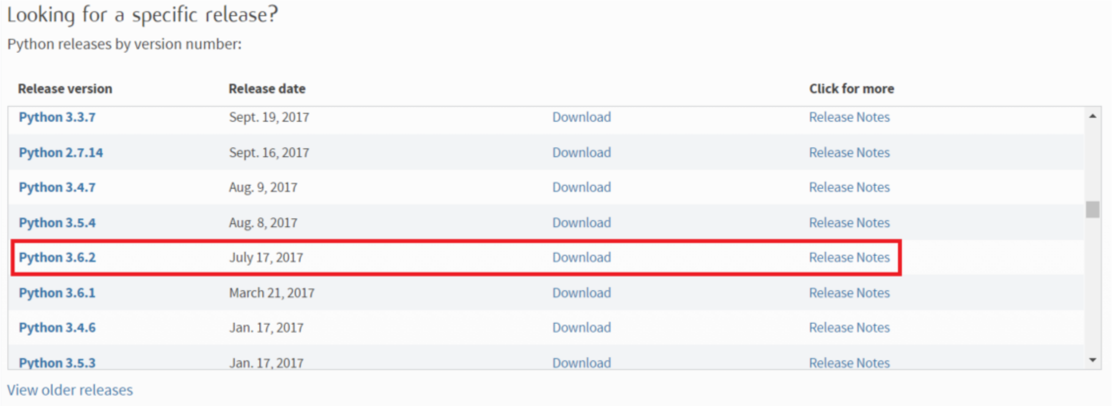
Looking for Python 2.7? See below for specific releases

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

Image 3: Python download web page.



Looking for a specific release?

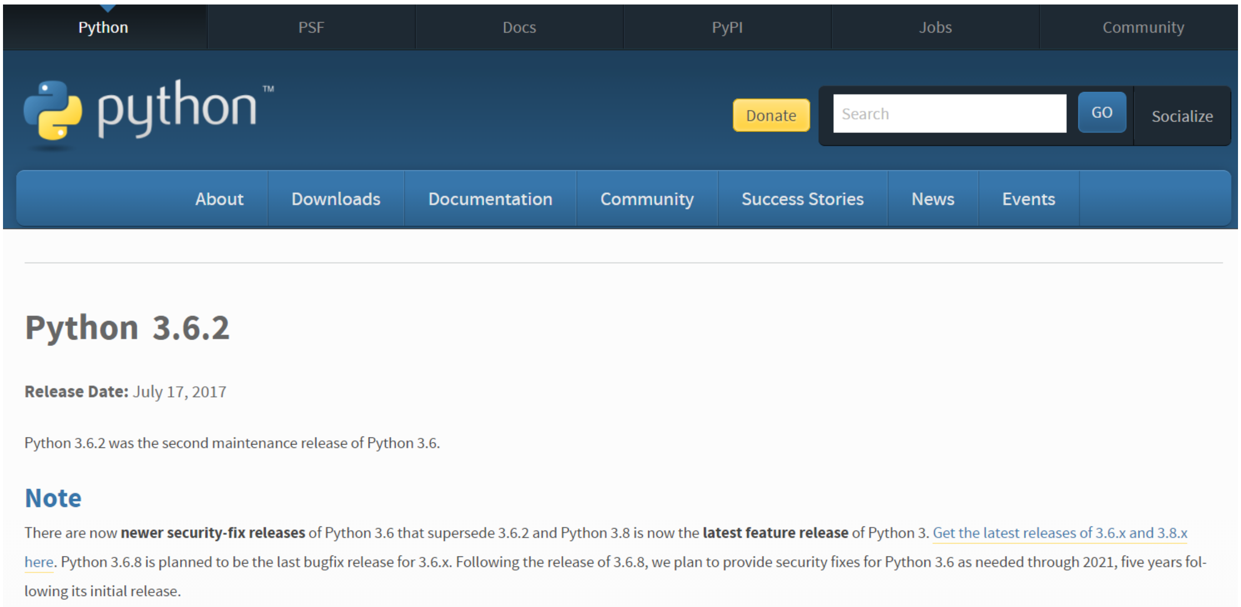
Python releases by version number:

Release version	Release date		Click for more
Python 3.3.7	Sept. 19, 2017	Download	Release Notes
Python 2.7.14	Sept. 16, 2017	Download	Release Notes
Python 3.4.7	Aug. 9, 2017	Download	Release Notes
Python 3.5.4	Aug. 8, 2017	Download	Release Notes
Python 3.6.2	July 17, 2017	Download	Release Notes
Python 3.6.1	March 21, 2017	Download	Release Notes
Python 3.4.6	Jan. 17, 2017	Download	Release Notes
Python 3.5.3	Jan. 17, 2017	Download	Release Notes

View older releases

Image 4: Python 3.6.2 selection.

c. Click the link **Download Python 3.6.2** button. The following page will appear in your browser.



Python 3.6.2

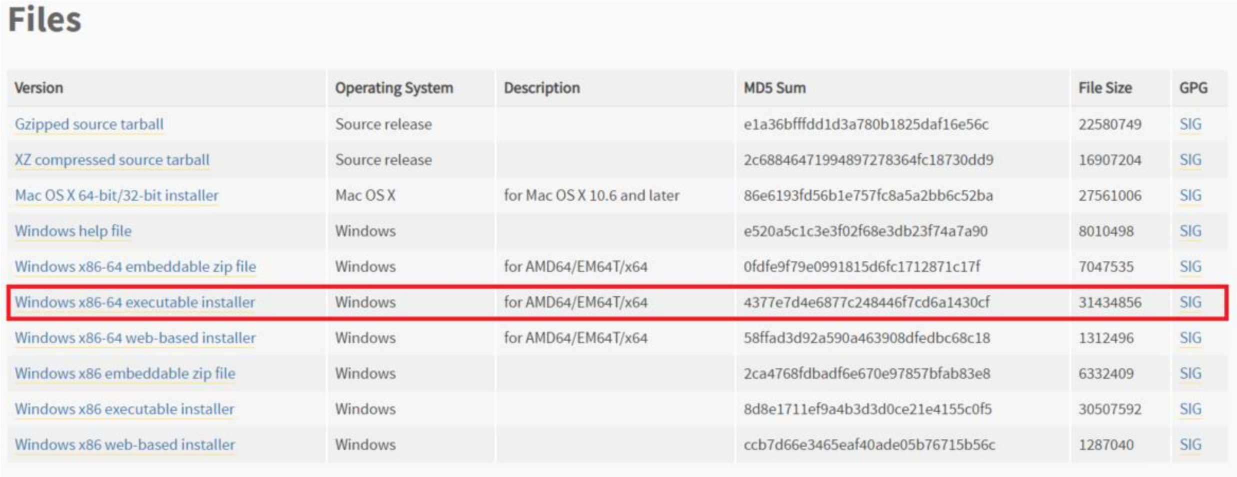
Release Date: July 17, 2017

Python 3.6.2 was the second maintenance release of Python 3.6.

Note

There are now **newer security-fix releases** of Python 3.6 that supersede 3.6.2 and Python 3.8 is now the **latest feature release** of Python 3. Get the latest releases of 3.6.x and 3.8.x here. Python 3.6.8 is planned to be the last bugfix release for 3.6.x. Following the release of 3.6.8, we plan to provide security fixes for Python 3.6 as needed through 2021, five years following its initial release.

Image 5: Python 3.6.2 details.

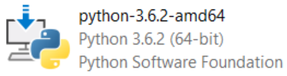


Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		e1a36bfffdd1d3a780b1825daf16e56c	22580749	SIG
XZ compressed source tarball	Source release		2c68846471994897278364fc18730dd9	16907204	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	86e6193fd56b1e757fc8a5a2bb6c52ba	27561006	SIG
Windows help file	Windows		e520a5c1c3e3f02f68e3db23f74a7a90	8010498	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	0fdfe9f79e0991815d6fc1712871c17f	7047535	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	4377e7d4e6877c248446f7cd6a1430cf	31434856	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	58ffad3d92a590a463908dfedbc68c18	1312496	SIG
Windows x86 embeddable zip file	Windows		2ca4768fdbadf6e670e97857bfab83e8	6332409	SIG
Windows x86 executable installer	Windows		8d8e1711ef9a4b3d3d0ce21e4155c0f5	30507592	SIG
Windows x86 web-based installer	Windows		ccb7d66e3465eaf40ade05b76715b56c	1287040	SIG

Image 6: Python 3.6.2 executable file selection.

d. Click on the **Download Windows x86-64 executable installer** link at the bottom of the page, under **File** heading. Download the installer. The file should appear as:



python-3.6.2-amd64
Python 3.6.2 (64-bit)
Python Software Foundation

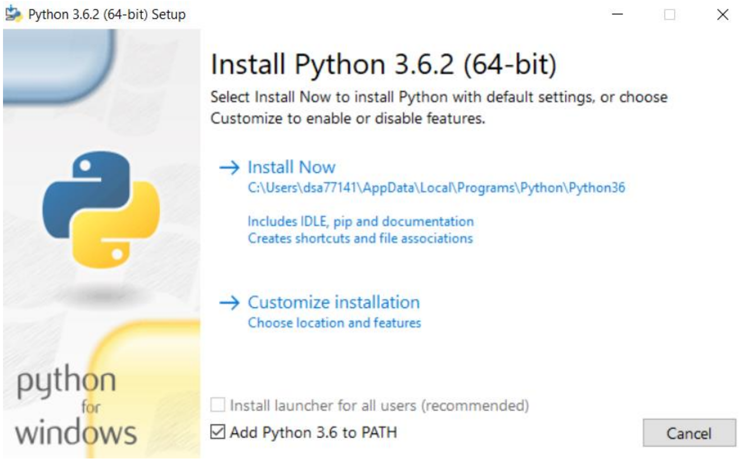
Image 7: Python 3.6.2 executable file icon.

e. Move this file to a more permanent location, so that you can install Python (and reinstall it easily later, if necessary).

f. Start the **Installing** instructions directly below.

g. Double-click the **icon** labelling the file python-3.6.2-amd64.exe.

h. A Python 3.6.2 (64-bit) Setup pop-up window will appear.



Install Python 3.6.2 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ Install Now
C:\Users\dsa77141\AppData\Local\Programs\Python\Python36

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

□ Install launcher for all users (recommended)
☑ Add Python 3.6 to PATH

Image 8: Python 3.6.2 installation setup.

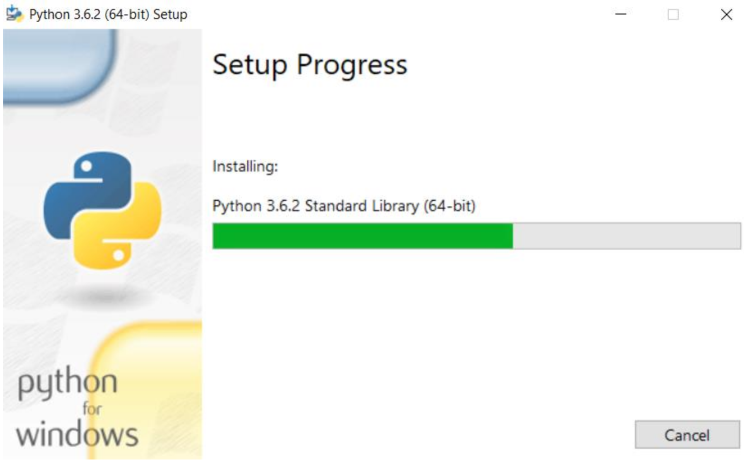
i. **Add** Python 3.6 to PATH checkboxes at the bottom are checked.

j. If the Python Installer finds an earlier version of Python installed on your computer, the **Install Now** message may instead appear as **Upgrade Now** (and the checkboxes will not appear).

k. Highlight the **Install Now** (or **Upgrade Now**) message, and then click it.

l. When it runs, a User Account Control pop-up window may appear on your screen. Do you want to allow this app to make changes to your device? Click the **Yes** button.

m. A new Python 3.6.2 (64-bit) Setup pop-up window will appear with a **Setup Progress** message and a progress bar.



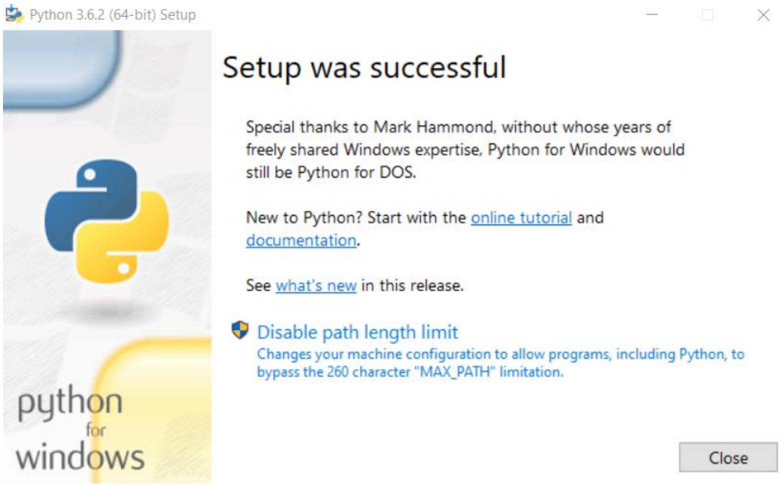
Setup Progress

Installing:

Python 3.6.2 Standard Library (64-bit)

Image 9: Python 3.6.2 installation setup progress.

n. During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new Python 3.6.2 (64-bit) Setup pop-up window will appear with a **Setup was successfully** message.



Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Image 10. Python 3.6.2 installation message.

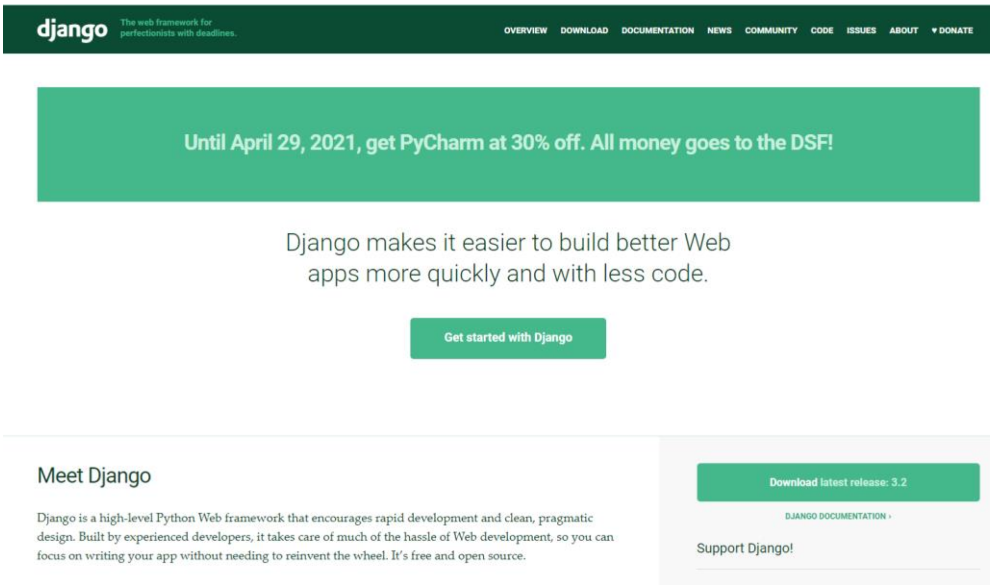
o. Click the **Close** button.

Python should now be installed.

3.2 DJANGO 3.1.7

Django is a Python-based free and open-source web framework that follows the model-template-views architectural pattern. Information related to django can be found on this web page django project (6).

The following page will appear in your browser, go to the **DOWNLOAD** tab in top right corner to see information related to Download and Installation of django.



Until April 29, 2021, get PyCharm at 30% off. All money goes to the DSF!

Django makes it easier to build better Web apps more quickly and with less code.

Meet Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Image 11: Django project web page.

Being a Python Web framework, django requires Python.

The development environment is an installation of django on local computer that can be used for developing and testing of django apps prior to deploying them to a production environment. The main tools that django itself provides are a set of Python scripts for creating and working with django projects that you can use to test local django web applications on your computer's web browser.

To install django, we first have to set up working environment for django using **Command Prompt.**

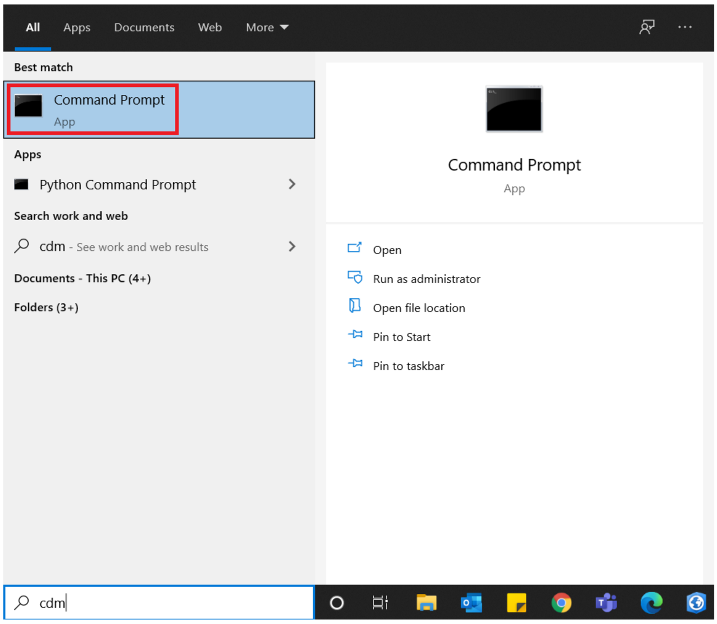
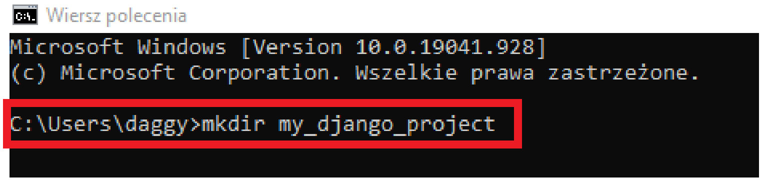


Image 12: Command Prompt access.

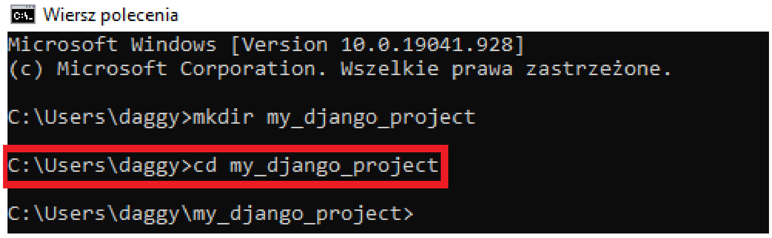
a. First, we have to create a **development directory** (**my django project**) for the project (note that project directory will be assigned based on authors location of choice) using command: *mkdir my_django_project*.



```
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\daggy>mkdir my_django_project
```

Image 13: Creation of the Development directory.

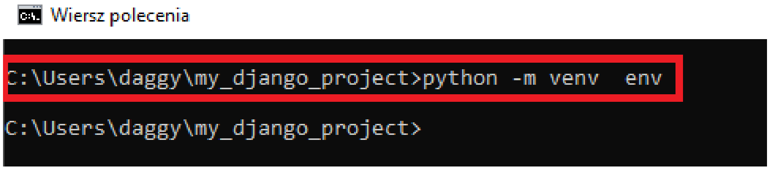
b. Within the directory we create **project catalog**, using command: *cd my_django_project*.



```
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\daggy>mkdir my_django_project

C:\Users\daggy>cd my_django_project

C:\Users\daggy\my_django_project>
```

Image 14: Creation of the project catalog.

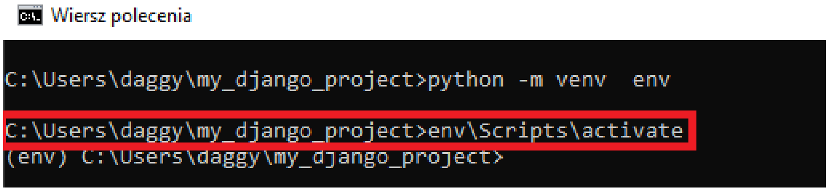
c. Create **virtual environment**, using command: *python -m venv env*. Here django related **libraries** will be installed.



```
C:\Users\daggy\my_django_project>python -m venv  env

C:\Users\daggy\my_django_project>
```

Image 15: Creation of the virtual environment.

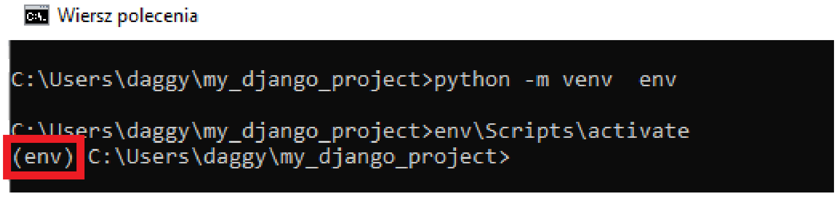
d. **Activate** the virtual environment using command: *env\Scripts\activate.*



```
C:\Users\daggy\my_django_project>python -m venv  env

C:\Users\daggy\my_django_project>env\Scripts\activate
(env) C:\Users\daggy\my_django_project>
```

Image 16: Activation of the virtual environment in Command Prompt.

When the command completes the new virtual environment will be active. This can be noted, as the prompt have the name of the environment in brackets.

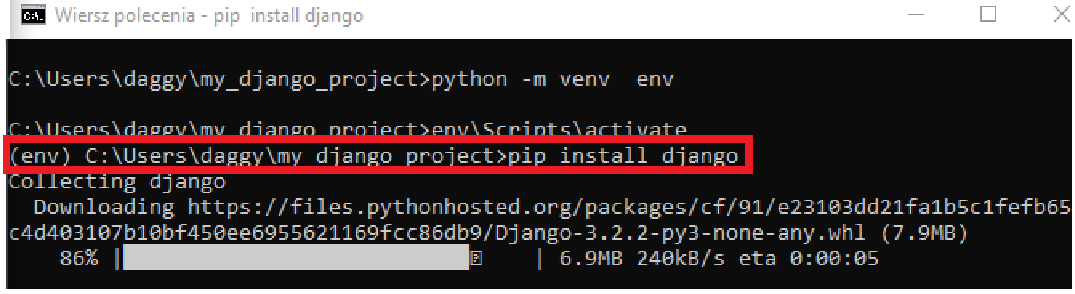


```
C:\Users\daggy\my_django_project>python -m venv  env

C:\Users\daggy\my_django_project>env\Scripts\activate
(env) C:\Users\daggy\my_django_project>
```

Image 17: Confirmation of the virtual environment activation in Command Prompt.

Virtual environment (env) is now ready for **django installation**.

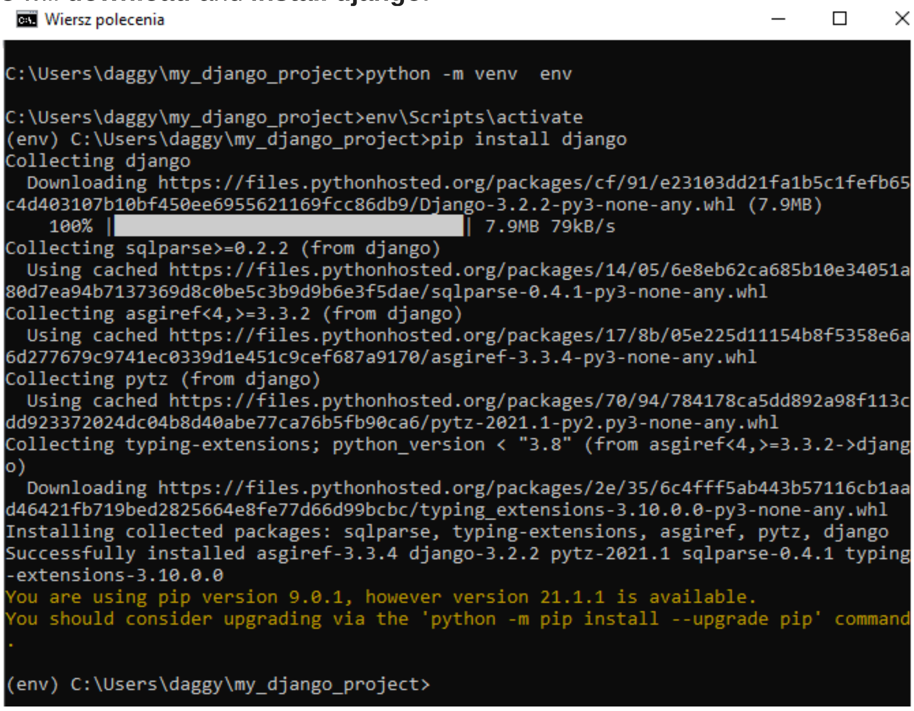
e. Django can be installed using by executing the following command: **pip install django**.



```
C:\Users\daggy\my_django_project>python -m venv  env

C:\Users\daggy\my_django_project>env\Scripts\activate
(env) C:\Users\daggy\my_django_project>pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/cf/91/e23103dd21fa1b5c1fefb65
c4d403107b10bf450ee6955621169fcc86db9/Django-3.2.2-py3-none-any.whl (7.9MB)
    86% |███████████████████████████     | 6.9MB 240kB/s eta 0:00:05
```

Image 18: Django installation command in Command Prompt.

This will **download** and **install django**.



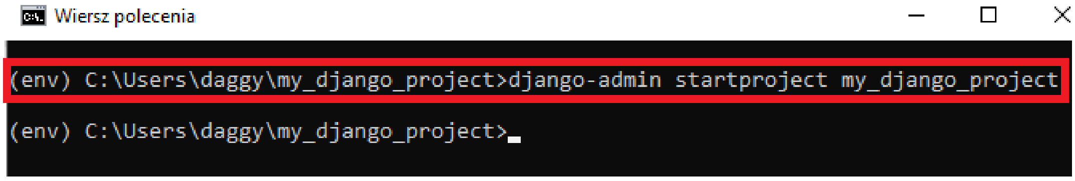
```
C:\Users\daggy\my_django_project>python -m venv  env

C:\Users\daggy\my_django_project>env\Scripts\activate
(env) C:\Users\daggy\my_django_project>pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/cf/91/e23103dd21fa1b5c1fefb65
c4d403107b10bf450ee6955621169fcc86db9/Django-3.2.2-py3-none-any.whl (7.9MB)
    100% |████████████████████████████████| 7.9MB 79kB/s
Collecting sqlparse>=0.2.2 (from django)
  Using cached https://files.pythonhosted.org/packages/14/05/6e8eb62ca685b10e34051a
80d7ea94b7137369d8c0be5c3b9d9b6e3f5dae/sqlparse-0.4.1-py3-none-any.whl
Collecting asgiref<4,>=3.3.2 (from django)
  Using cached https://files.pythonhosted.org/packages/17/8b/05e225d11154b8f5358e6a
6d277679c9741ec0339d1e451c9cef687a9170/asgiref-3.3.4-py3-none-any.whl
Collecting pytz (from django)
  Using cached https://files.pythonhosted.org/packages/70/94/784178ca5dd892a98f113c
dd923372024dc04b8d40abe77ca76b5fb90ca6/pytz-2021.1-py2.py3-none-any.whl
Collecting typing-extensions; python_version < "3.8" (from asgiref<4,>=3.3.2->djang
o)
  Downloading https://files.pythonhosted.org/packages/2e/35/6c4fff5ab443b57116cb1aa
d46421fb719bed2825664e8fe77d66d99bcbc/typing_extensions-3.10.0.0-py3-none-any.whl
Installing collected packages: sqlparse, typing-extensions, asgiref, pytz, django
Successfully installed asgiref-3.3.4 django-3.2.2 pytz-2021.1 sqlparse-0.4.1 typing
-extensions-3.10.0.0
You are using pip version 9.0.1, however version 21.1.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command
.

(env) C:\Users\daggy\my_django_project>
```

Image 19: Django installation in Command Prompt.

Django should now be installed.

Next step would be to create **django project** and **application** (app), which will be edited in python-based software.

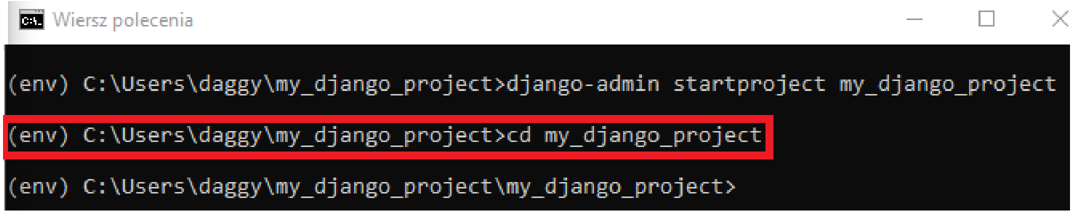
a. **Start django** project using command: *django-admin startproject my_django_app.*



```
(env) C:\Users\daggy\my_django_project>django-admin startproject my_django_project

(env) C:\Users\daggy\my_django_project>
```

Image 20: Creation of django project.

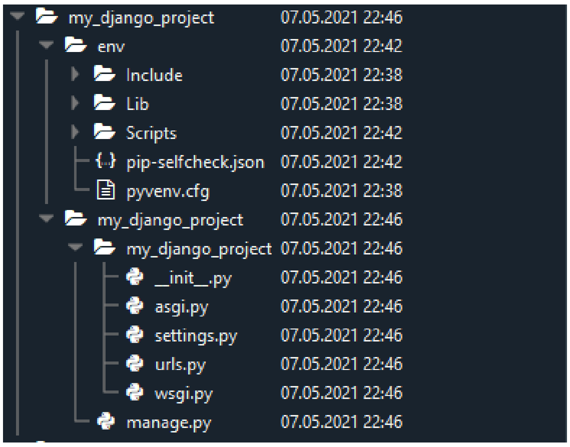
b. Enter the **project folder** using command: *cd my_django_project.*



```
(env) C:\Users\daggy\my_django_project>django-admin startproject my_django_project

(env) C:\Users\daggy\my_django_project>cd my_django_project

(env) C:\Users\daggy\my_django_project\my_django_project>
```

Image 21: Creation of django project folder.

c. To try to **verify** if project has been set up correctly view it structure using **any Python editor**.The following pop-up window should appear:

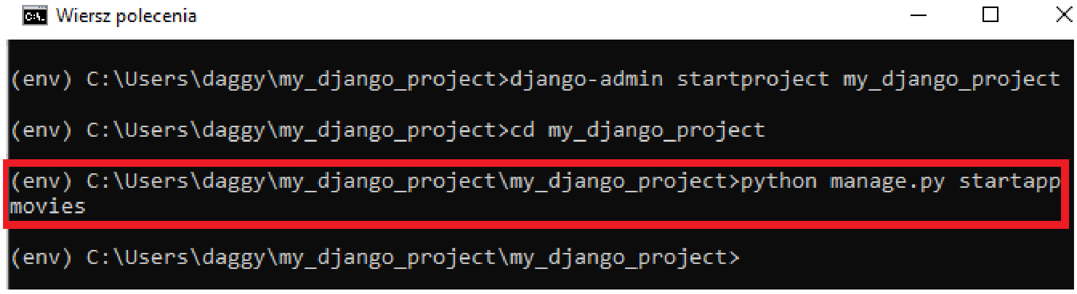


```
my_django_project           07.05.2021 22:46
  env                       07.05.2021 22:42
    Include                 07.05.2021 22:38
    Lib                     07.05.2021 22:38
    Scripts                 07.05.2021 22:42
    pip-selfcheck.json      07.05.2021 22:42
    pyvenv.cfg              07.05.2021 22:38
  my_django_project         07.05.2021 22:46
    my_django_project       07.05.2021 22:46
      __init__.py           07.05.2021 22:46
      asgi.py               07.05.2021 22:46
      settings.py           07.05.2021 22:46
      urls.py               07.05.2021 22:46
      wsgi.py               07.05.2021 22:46
    manage.py               07.05.2021 22:46
```

Image 22: Django project folder structure viewed in Spyder.

Exit using command: *cdm.*

d. Start **application (app)** for project using command: *python manage.py startapp movies.*

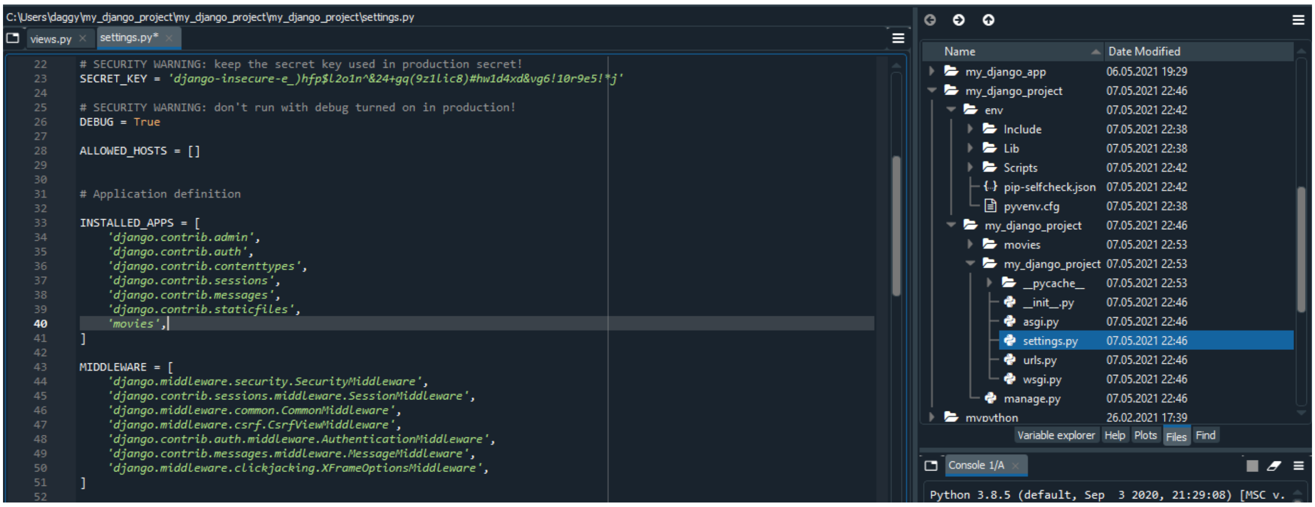


```
(env) C:\Users\daggy\my_django_project>django-admin startproject my_django_project

(env) C:\Users\daggy\my_django_project>cd my_django_project

(env) C:\Users\daggy\my_django_project\my_django_project>python manage.py startapp
movies

(env) C:\Users\daggy\my_django_project\my_django_project>
```

Image 23: Start of the django application.

e. Bring directory into text editor.

f. App can be added into any Python editor, into apps in **setting.py.**



C:\Users\daggy\my_django_project\my_django_project\my_django_project\settings.py

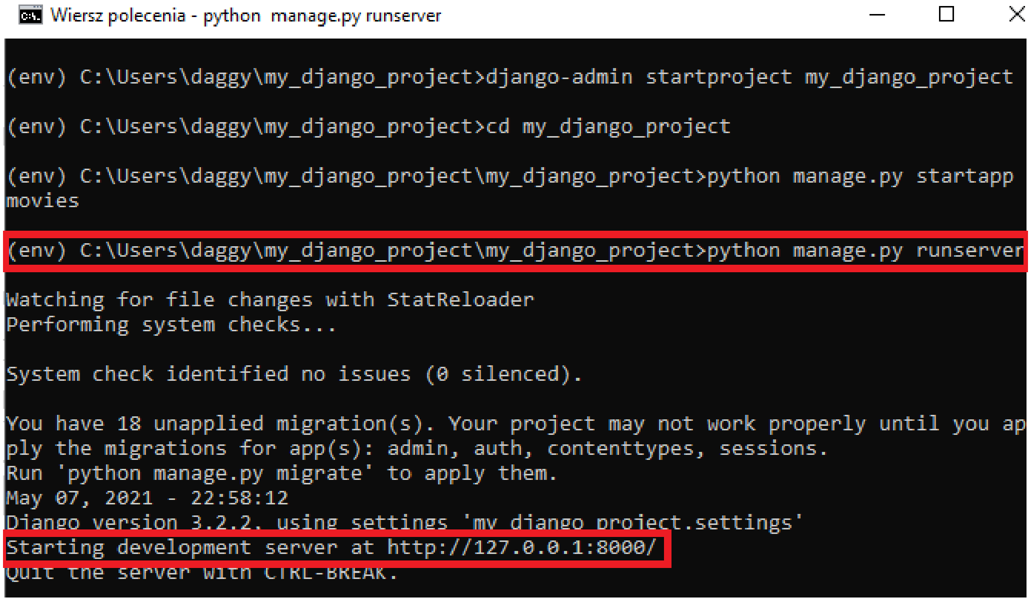
```
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-e_)hfp$l2o1n^&24+gq(9z1lic8)#hw1d4xd&vg6!10r9e5!*j'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'movies',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

```

Name	Date Modified
my_django_app	06.05.2021 19:29
my_django_project	07.05.2021 22:46
env	07.05.2021 22:42
Include	07.05.2021 22:38
Lib	07.05.2021 22:38
Scripts	07.05.2021 22:42
pip-selfcheck.json	07.05.2021 22:42
pyvenv.cfg	07.05.2021 22:38
my_django_project	07.05.2021 22:46
movies	07.05.2021 22:53
my_django_project	07.05.2021 22:53
__pycache__	07.05.2021 22:53
__init__.py	07.05.2021 22:46
asgi.py	07.05.2021 22:46
settings.py	07.05.2021 22:46
urls.py	07.05.2021 22:46
wsgi.py	07.05.2021 22:46
manage.py	07.05.2021 22:46
mypython	26.02.2021 17:39

```
Python 3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.
```

Image 24: Django app preview in Python.

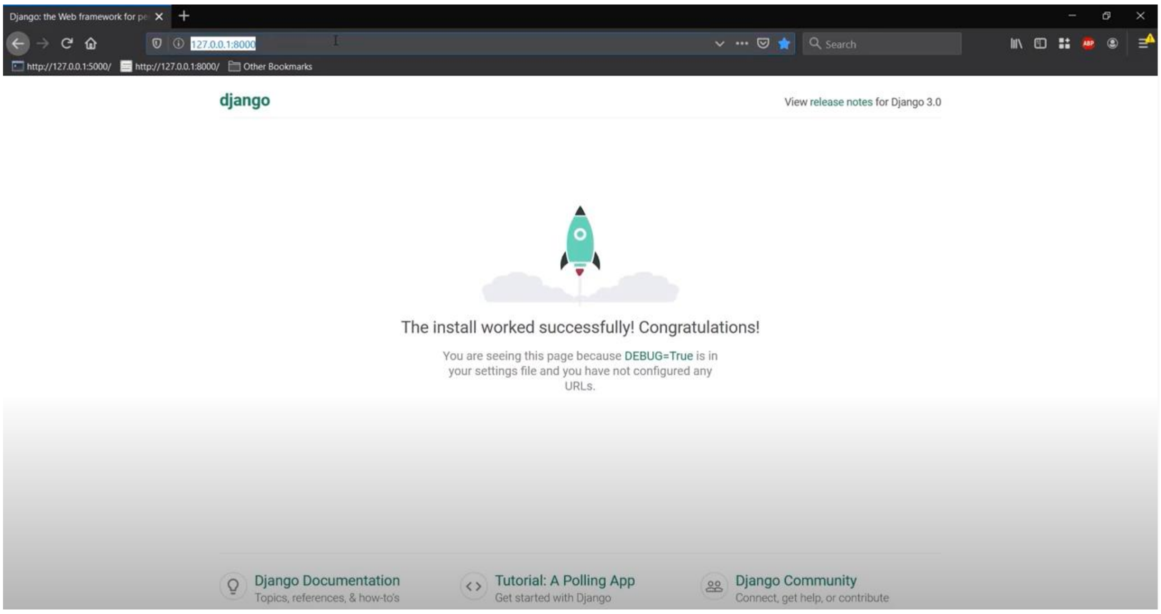
g. To verify if django is working, we can go into admin login to **run a server location** using command: *python manage.py runserver*



```
(env) C:\Users\daggy\my_django_project>django-admin startproject my_django_project

(env) C:\Users\daggy\my_django_project>cd my_django_project

(env) C:\Users\daggy\my_django_project\my_django_project>python manage.py startapp
movies

(env) C:\Users\daggy\my_django_project\my_django_project>python manage.py runserver

Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you ap
ply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
May 07, 2021 - 22:58:12
Django version 3.2.2, using settings 'my_django_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Image 25: Development Server command and location.

h. Copy and paste server location in browser: http://127.0.0.1:8000/



The install worked successfully! Congratulations!

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

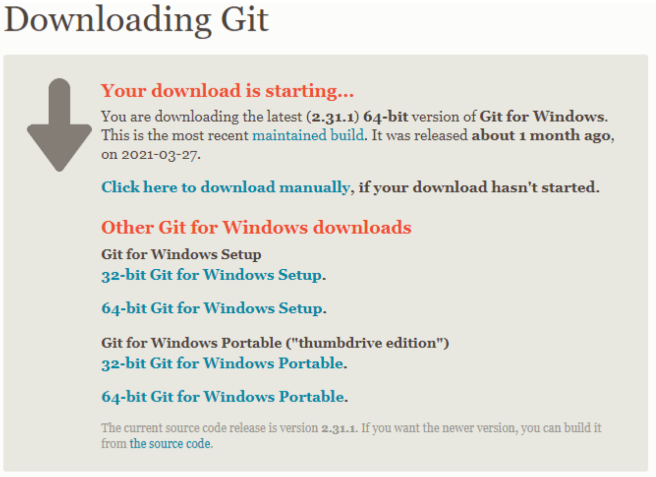
Image 26: Development Server location in web browser.

i. Work on your code in any Python editor.

3.3 OTHER SOFTWARE

1. HTML stands for HyperText Markup Language. It is the standard markup language for documents designed to be displayed in a web browser. It is used to structure and give purpose to our web content, for example defining paragraphs, headings, and data tables, or embedding images and videos in the page. (7)

   There is no need to download and install HTML. Open new file on Notepad or ++ or your favourite text editor and save as **filename. html**. It can be now open in any browser installed on PC.
2. JavaScript is a scripting or programming language that allows you to implement dynamic behaviour and complex features on web pages, like timely content updates, interactive maps, animated 2D/3D graphics, scrolling video, etc. (7) There is no need to download and install JavaScript. Any browser installed on PC has an interpreter that can execute the code.
3. CSS is a language of style rules that we use to apply styling to our HTML content, for example setting background colours and fonts, and laying out our content in multiple columns. (7) There is no need to download and install CSS. Any browser installed on PC has an interpreter that can execute the code.
4. Git is 'a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.' (8) It used to transfer **my django project** to Git Hub repository, which later is used by PythonAnywhere as a django application to push data it to external server.
   a. To Install click: Git download (9) and select appropriate operating system installation (Git for **Windows**).



Downloading Git

Your download is starting...

You are downloading the latest (2.31.1) 64-bit version of Git for Windows. This is the most recent maintained build. It was released about 1 month ago, on 2021-03-27.

Click here to download manually, if your download hasn't started.

Other Git for Windows downloads

Git for Windows Setup
32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Git for Windows Portable ("thumbdrive edition")
32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

The current source code release is version 2.31.1. If you want the newer version, you can build it from the source code.

Image 27: Git application download.

b. The file should appear as.

Git-2.31.1-64-bit

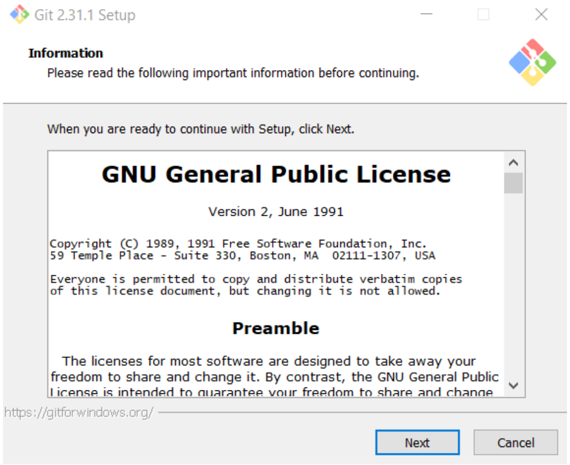
Image 28: Git 2.31.1-64-bit executable file icon.

c. Move this file to a more permanent location, so that you can install Git (and reinstall it easily later, if necessary).

d. Start the **Installing** instructions directly below.

e. Double-click the icon labelling the file **Git-2.31.1-64-bit.exe**

f. A **Git-2.31.1 Setup** pop-up window will appear.



Git 2.31.1 Setup

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

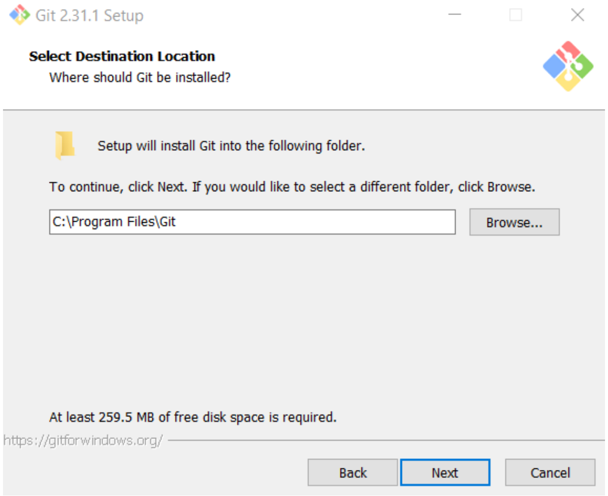
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

https://gitforwindows.org/

Image 29: Git-2.31.1 Setup information.

g. Click **Next** and select installation destination location (note that project directory will be assigned based on authors location of choice).



Git 2.31.1 Setup

Select Destination Location

Where should Git be installed?

Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

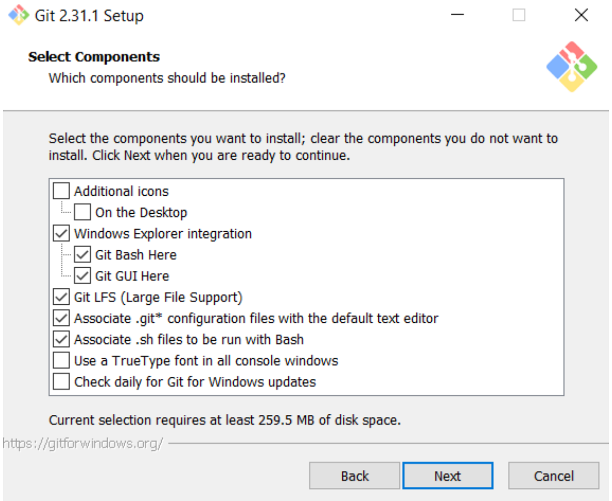
C:\Program Files\Git

At least 259.5 MB of free disk space is required.

https://gitforwindows.org/

Image 30: Git-2.31.1 Setup Select Destination Location.

h. Select components of installation and click **Next**.



Git 2.31.1 Setup

Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

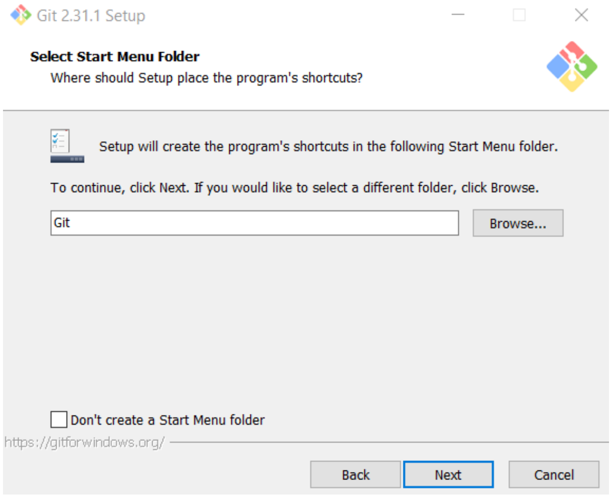
- □ Additional icons
  - □ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Git Bash Here
  - ☑ Git GUI Here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- □ Use a TrueType font in all console windows
- □ Check daily for Git for Windows updates

Current selection requires at least 259.5 MB of disk space.

https://gitforwindows.org/

Image 31: Git-2.31.1 Setup Select Components.

i. Select Start Menu folder and click **Next.**



Git 2.31.1 Setup

Select Start Menu Folder

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

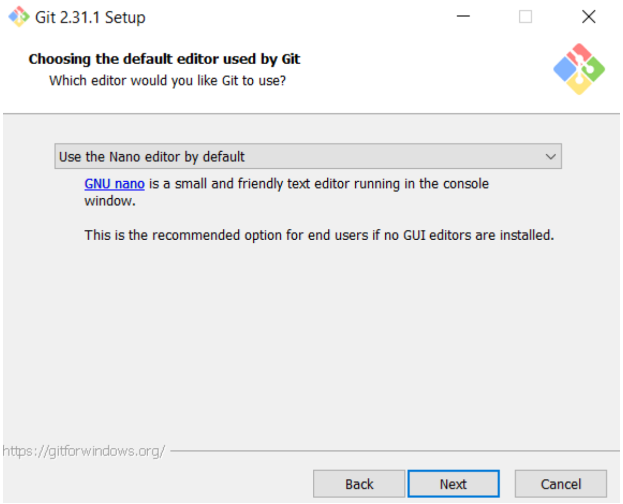
Git

□ Don't create a Start Menu folder

https://gitforwindows.org/

Image 32: Git-2.31.1 Setup Select Start Menu Folder.

j. Choose the default editor used by Git and click **Next**.



Git 2.31.1 Setup

Choosing the default editor used by Git

Which editor would you like Git to use?

Use the Nano editor by default

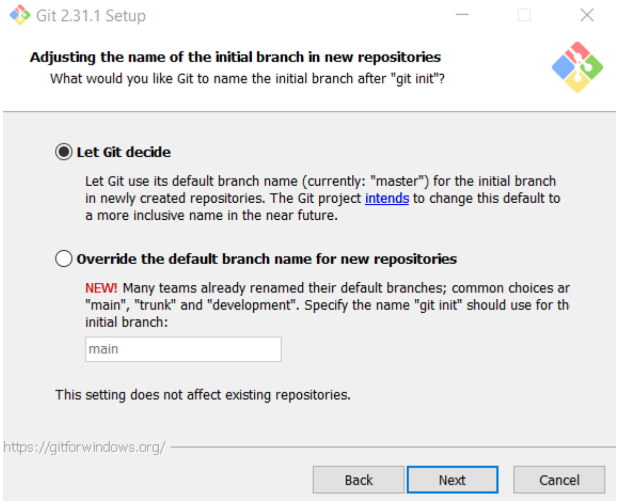
GNU nano is a small and friendly text editor running in the console window.

This is the recommended option for end users if no GUI editors are installed.

https://gitforwindows.org/

Image 33: Git-2.31.1 Setup Choose the default editor used by Git.

k. Adjust the name of the initial branch in new repositories and click **Next**.



Git 2.31.1 Setup

Adjusting the name of the initial branch in new repositories

What would you like Git to name the initial branch after "git init"?

● **Let Git decide**

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project intends to change this default to a more inclusive name in the near future.

○ **Override the default branch name for new repositories**

NEW! Many teams already renamed their default branches; common choices ar "main", "trunk" and "development". Specify the name "git init" should use for th initial branch:

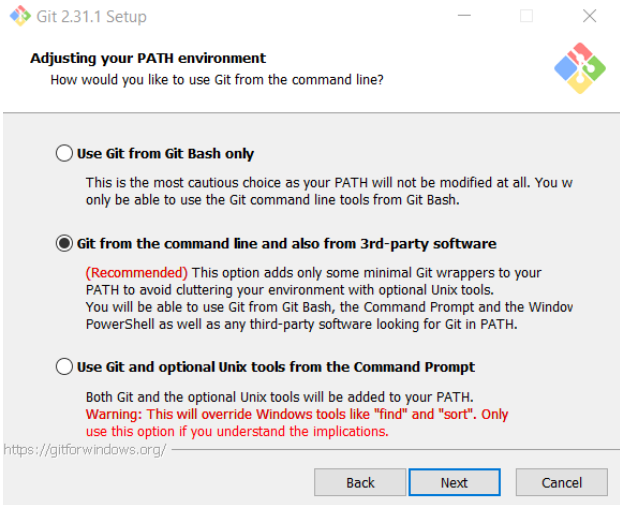
main

This setting does not affect existing repositories.

https://gitforwindows.org/

Image 34: Git-2.31.1 Setup Adjust the name of the initial branch in new repositories.

l. Adjust PATH environment and click **Next**.



Git 2.31.1 Setup

Adjusting your PATH environment

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You w only be able to use the Git command line tools from Git Bash.

● **Git from the command line and also from 3rd-party software**

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windov PowerShell as well as any third-party software looking for Git in PATH.

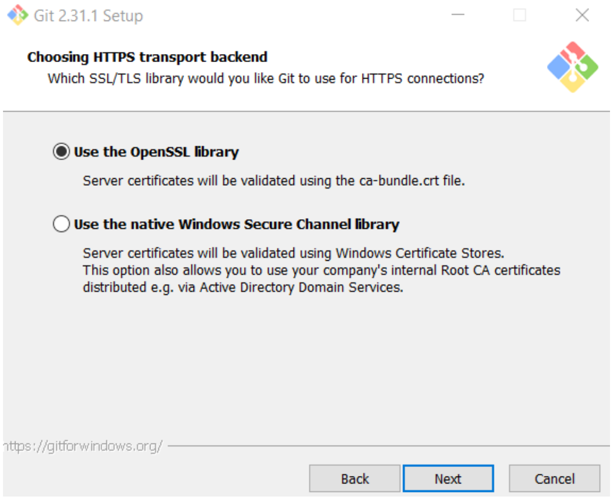
○ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH. Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://gitforwindows.org/

Image 35. Git-2.31.1 Setup Adjust your PATH environment.

m. Choose HTTPS transport backend and click **Next**.



Git 2.31.1 Setup

Choosing HTTPS transport backend

Which SSL/TLS library would you like Git to use for HTTPS connections?

● **Use the OpenSSL library**

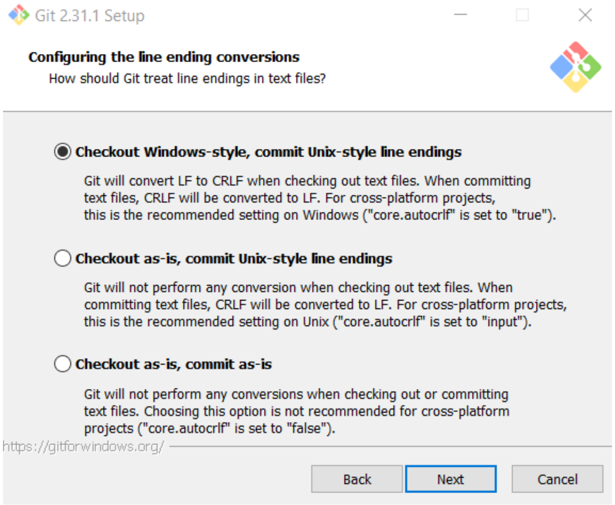
Server certificates will be validated using the ca-bundle.crt file.

○ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores. This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

Image 36. Git-2.31.1 Choosing HTTPS transport backend.

n. Configure the line ending conversions and click **Next**.



Git 2.31.1 Setup

Configuring the line ending conversions

How should Git treat line endings in text files?

● **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

○ **Checkout as-is, commit Unix-style line endings**

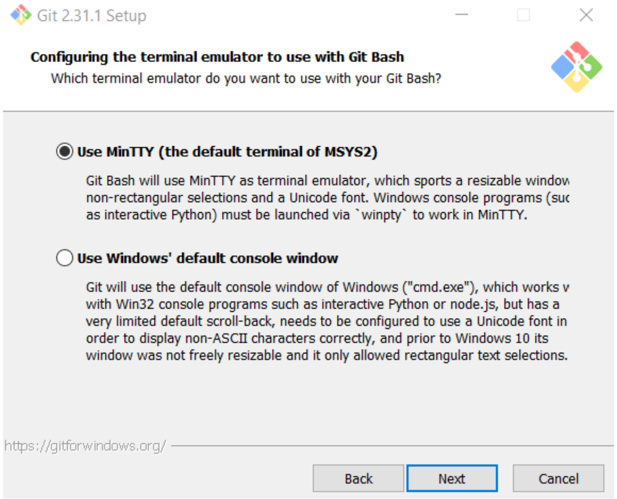
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

Image 37. Git-2.31.1 Setup Configurating the line ending conversation.

o. Configure the terminal emulator to use with Git Bash and click **Next**.



Git 2.31.1 Setup

Configuring the terminal emulator to use with Git Bash

Which terminal emulator do you want to use with your Git Bash?

● **Use MinTTY (the default terminal of MSYS2)**

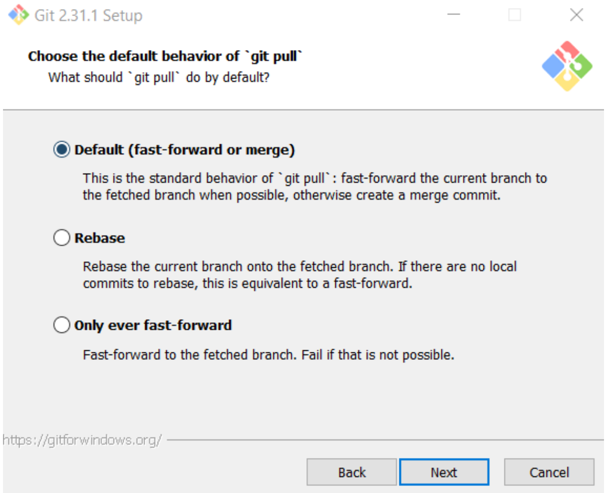
Git Bash will use MinTTY as terminal emulator, which sports a resizable window non-rectangular selections and a Unicode font. Windows console programs (suc as interactive Python) must be launched via `winpty` to work in MinTTY.

○ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works v with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

Image 38. Git-2.31.1 Setup Configurating the terminal emulator to use with Git Bash.

p. Choose the default behaviour of the ‘git pull’ and click **Next**.



Git 2.31.1 Setup

Choose the default behavior of `git pull`

What should `git pull` do by default?

● **Default (fast-forward or merge)**

This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

○ **Rebase**

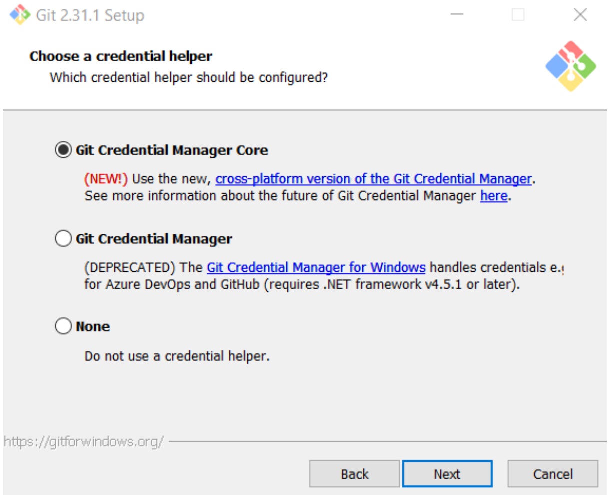
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

○ **Only ever fast-forward**

Fast-forward to the fetched branch. Fail if that is not possible.

Image 39. Git-2.31.1 Setup Choose the default behaviour of the ‘git pull’.

q. Choose a credential helper and click **Next.**



Git 2.31.1 Setup

Choose a credential helper

Which credential helper should be configured?

● **Git Credential Manager Core**

(NEW!) Use the new, cross-platform version of the Git Credential Manager. See more information about the future of Git Credential Manager here.

○ **Git Credential Manager**

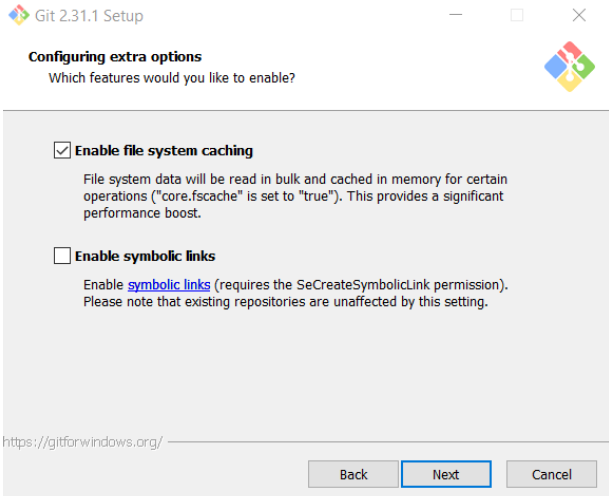
(DEPRECATED) The Git Credential Manager for Windows handles credentials e.g for Azure DevOps and GitHub (requires .NET framework v4.5.1 or later).

○ **None**

Do not use a credential helper.

Image 40. Git-2.31.1 Setup Choose a credential helper.

r. Configure extra options and click **Next.**



Git 2.31.1 Setup

Configuring extra options

Which features would you like to enable?

☑ **Enable file system caching**

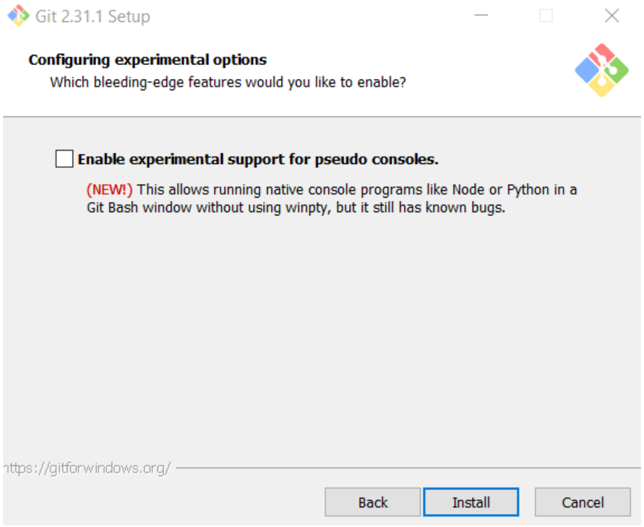
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

□ **Enable symbolic links**

Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

Image 41. Git-2.31.1 Setup Configuring extra options.

s. Configure experimental options and click **Install.**



Git 2.31.1 Setup

Configuring experimental options

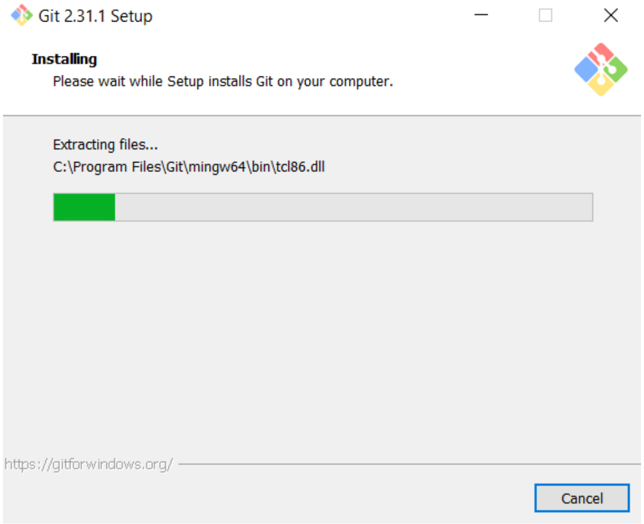
Which bleeding-edge features would you like to enable?

□ **Enable experimental support for pseudo consoles.**

(NEW!) This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.

Image 42. Git-2.31.1 Setup Configuring experimental options.

t. A pop-up window will appear with a **Setup Progress** message and a progress bar.



Git 2.31.1 Setup

Installing

Please wait while Setup installs Git on your computer.

Extracting files...

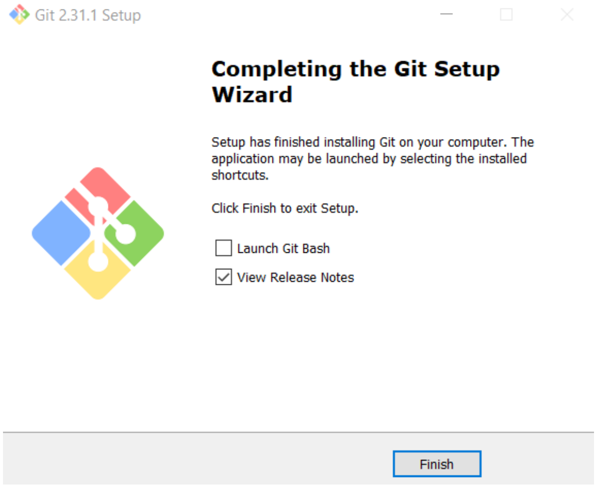
C:\Program Files\Git\mingw64\bin\tcl86.dll

Image 43. Git-2.31.1 Setup Installation.

u. During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new Git-2.31.1 Setup pop-up window will appear with a **Setup was successfully** message.

v. Click the **Finish** button.

w. A Release Note will appear with further information:



Git 2.31.1 Setup

Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

□ Launch Git Bash

☑ View Release Notes

Image 44. Git-2.31.1 Setup Wizard installation completion.

Git should now be installed.

x. Create a public repository on the GitHub.

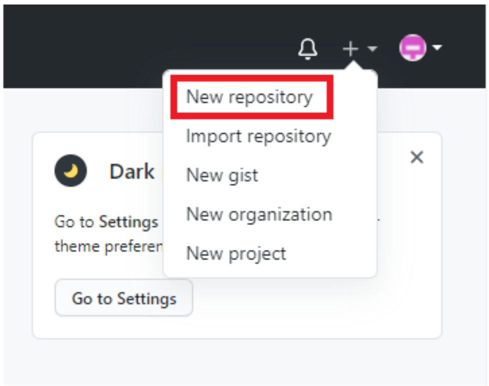
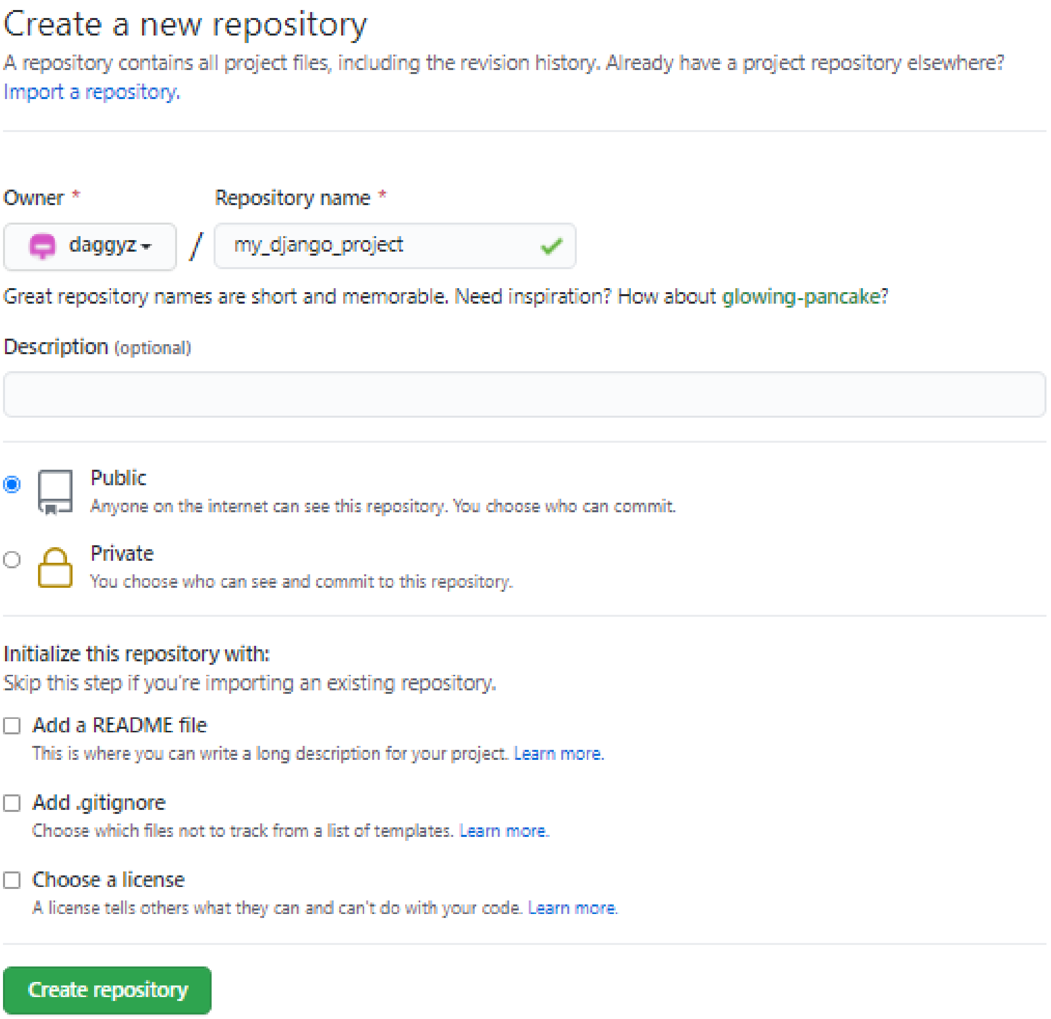


Image 45. Creation of public repository in GitHub.

y. Fill in details about the repository.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Repository name *

daggyz / my_django_project ✓

Great repository names are short and memorable. Need inspiration? How about glowing-pancake?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

□ Add a README file
This is where you can write a long description for your project. Learn more.

□ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

Image 46. Public repository details in GitHub.

3.4 LIBRARIES

Libraries are a set of useful functions that eliminate the need for writing codes from scratch. For the project purposes we used following libraries for Python, Java and CSS. Those can be installed on Windows using **pip** in **Command Prompt** or using **Anaconda** software.

PIP is a package management system used to install and manage software packages/libraries written in Python. These files are stored in a large “on-line repository” termed as Python Package Index (PyPI).

1. **Requests** is an abstract the complexities of making requests behind an API so that user can focus on interacting with services and consuming data in the application. To install package in Python 3.6.2 use following command: **pip install requests.**
2. **Pandas** is fast, powerful and flexible data manipulation and analysis tool. In particular, it offers data structures and operations for manipulating numerical tables and time series. To install package in Python 3.6.2 use following command: **pip install pandas.**
3. **NumPy** is adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. To install package in Python 3.6.2 use following command: **pip install numpy.**

4. **SQLite3** is a software library that provides a relational database management system. It is lightweight in terms of setup, database administration and required resources (10). There is no need to download and install SQLite3, as it is bundled with Python.

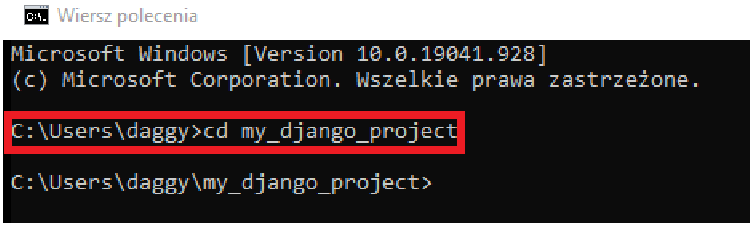
4 WEB PAGE

A web page (or webpage) is a hypertext document provided by a website and displayed to a user in a web browser. A website typically consists of many web pages linked together in a coherent fashion. (11) It is commonly written in HTML and viewed in any Internet browser.

For the purposes of our project, we hosted our project on the external web hosting service **PythonAnywhere**. It features WSGI-based web hosting (Web Server Gateway Interface) and allows to run django applications free for **3 months** with a limit of up to 5 GB. In order to use python anywhere, repository on **GitHub** was created to upload our project. This method allows for efficient and smooth running of the web page.

To upload project to python anywhere **Command Prompt** was used.

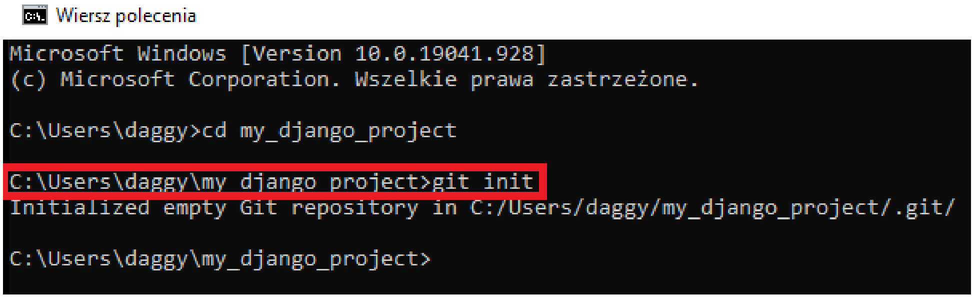
a. First, we go to the project location, using command: *cd my_django_project.*



```
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\daggy>cd my_django_project

C:\Users\daggy\my_django_project>
```

Image 47. Project location selection in Command Prompt.

b. Initiate connection with GitHub using command: *git init.*

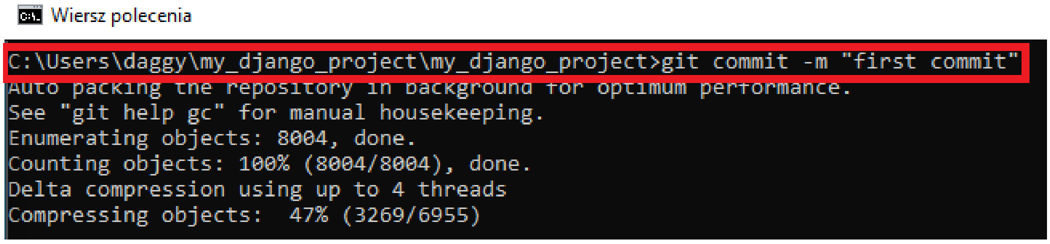


```
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\daggy>cd my_django_project

C:\Users\daggy\my_django_project>git init
Initialized empty Git repository in C:/Users/daggy/my_django_project/.git/

C:\Users\daggy\my_django_project>
```

Image 48. Connection initiation with GitHub in Command Prompt.

c. Add all file to the GitHub repository with command: *git add* .

d. Capture the state of a project at finish point in time with command: *git commit -m* “first commit”.



```
C:\Users\daggy\my_django_project\my_django_project>git commit -m "first commit"
Auto packing the repository in background for optimum performance.
See "git help gc" for manual housekeeping.
Enumerating objects: 8004, done.
Counting objects: 100% (8004/8004), done.
Delta compression using up to 4 threads
Compressing objects:  47% (3269/6955)
```

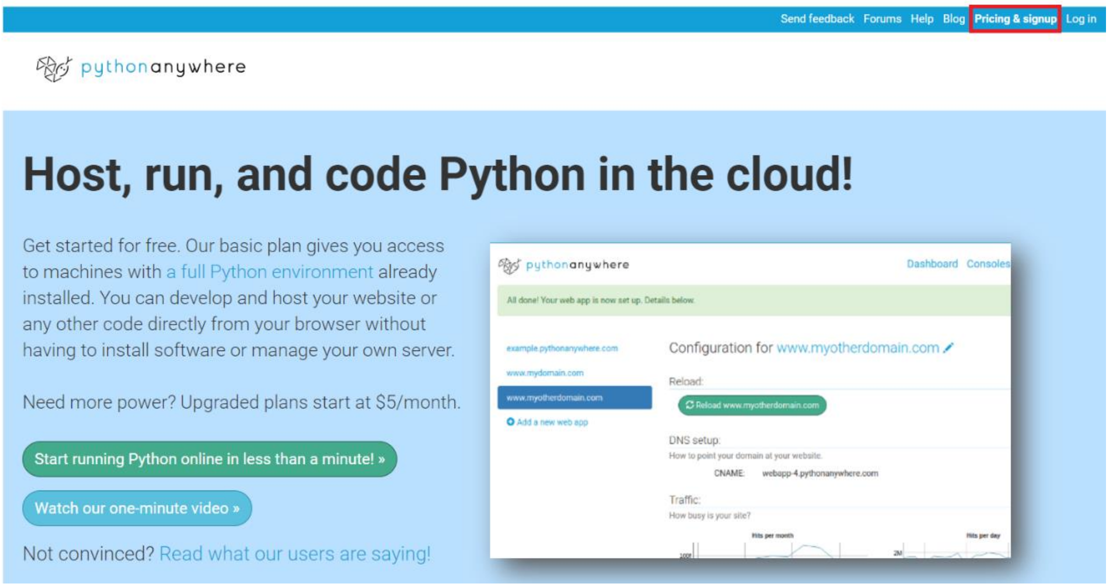
Image 49. State of a project in Command Prompt.

e. Send project to GitHub using command: *git push.*

PythonAnywhere is an online integrated development environment and web hosting service based on the Python programming language. (12) It is used as a tool that allows us to push our django application to the external server.

a. To Install go to: pythonanywhere (13)

b. Within Pricing & signup click on **Create a Beginner** account



Host, run, and code Python in the cloud!

Get started for free. Our basic plan gives you access to machines with a full Python environment already installed. You can develop and host your website or any other code directly from your browser without having to install software or manage your own server.

Need more power? Upgraded plans start at $5/month.

Not convinced? Read what our users are saying!

All done! Your web app is now set up. Details below.

Configuration for www.myotherdomain.com

Reload:

DNS setup:

How to point your domain at your website:

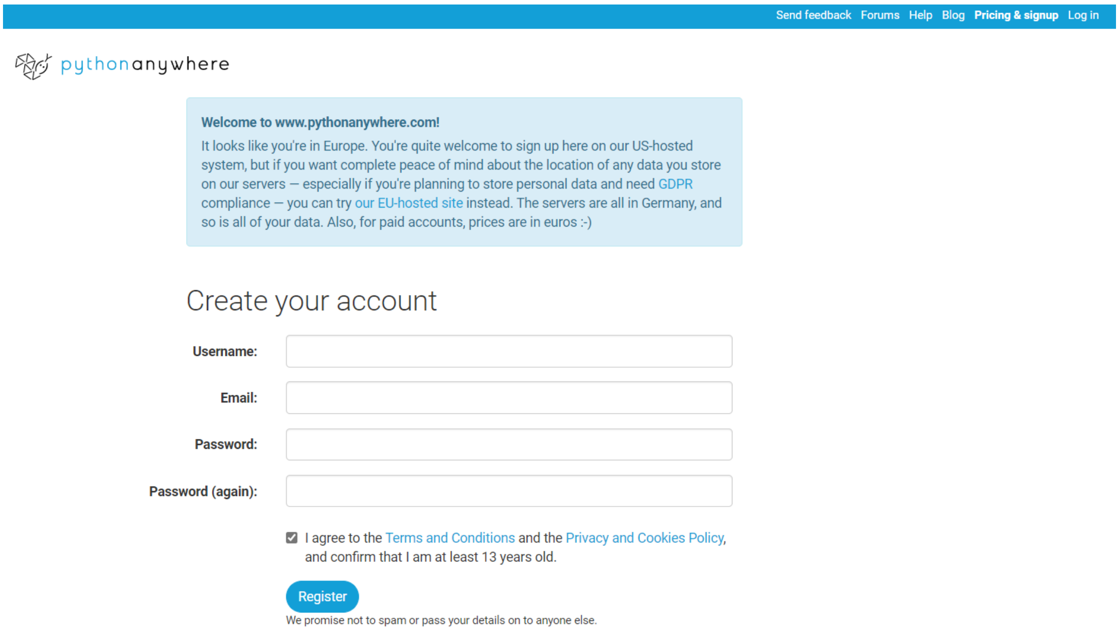
CNAME: webapp-4.pythonanywhere.com

Traffic:

How busy is your site?

Image 50. PythonAnywhere web site.

c. Create an account:



Welcome to www.pythonanywhere.com!

It looks like you're in Europe. You're quite welcome to sign up here on our US-hosted system, but if you want complete peace of mind about the location of any data you store on our servers — especially if you're planning to store personal data and need GDPR compliance — you can try our EU-hosted site instead. The servers are all in Germany, and so is all of your data. Also, for paid accounts, prices are in euros :-)

Create your account

Username:

Email:

Password:

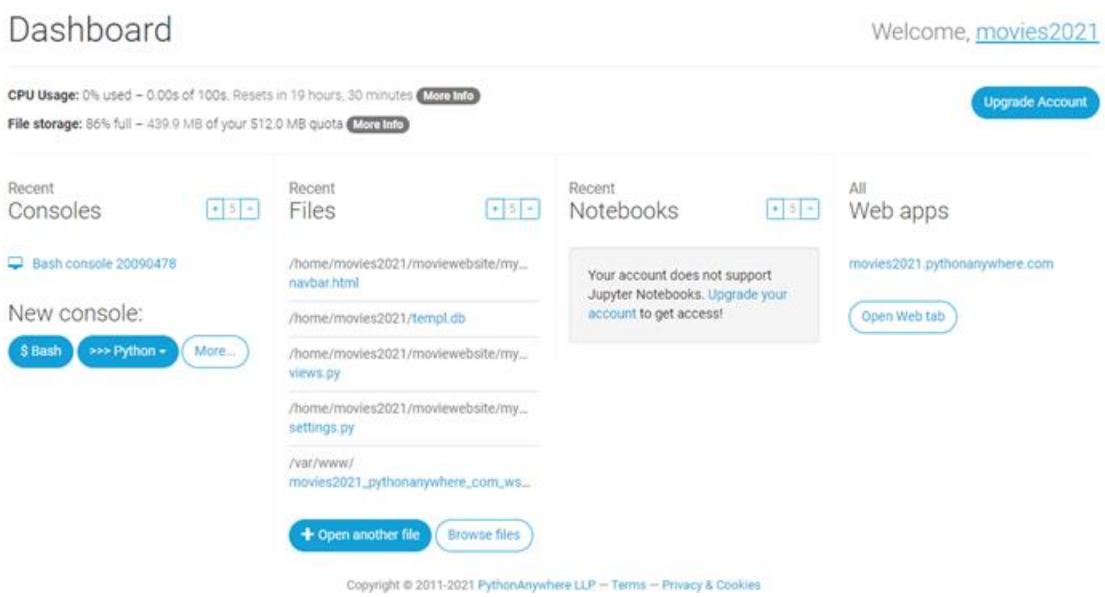
Password (again):

☑ I agree to the Terms and Conditions and the Privacy and Cookies Policy, and confirm that I am at least 13 years old.

We promise not to spam or pass your details on to anyone else.

Image 51. PythonAnywhere account creation.

d. In the **Dashboard** tab configure web page, so that it can be place o the PythonAnywhere server:



Dashboard

Welcome, movies2021

CPU Usage: 0% used – 0.00s of 100s. Resets in 19 hours, 30 minutes More Info

File storage: 86% full – 439.9 MB of your 512.0 MB quota More Info

Upgrade Account

Recent
Consoles

Bash console 20090478

New console:

$ Bash · >>> Python · More...

Recent
Files

/home/movies2021/moviewebsite/my... navbar.html

/home/movies2021/templ.db

/home/movies2021/moviewebsite/my... views.py

/home/movies2021/moviewebsite/my... settings.py

/var/www/ movies2021_pythonanywhere_com_ws...

Open another file · Browse files

Recent
Notebooks

Your account does not support Jupyter Notebooks. Upgrade your account to get access!

All
Web apps

movies2021.pythonanywhere.com

Open Web tab

Copyright © 2011-2021 PythonAnywhere LLP — Terms — Privacy & Cookies

Image 52. PythonAnywhere Dashboard configuration.

Our web page was launched: movies2021.pythonanywhere.com (14).

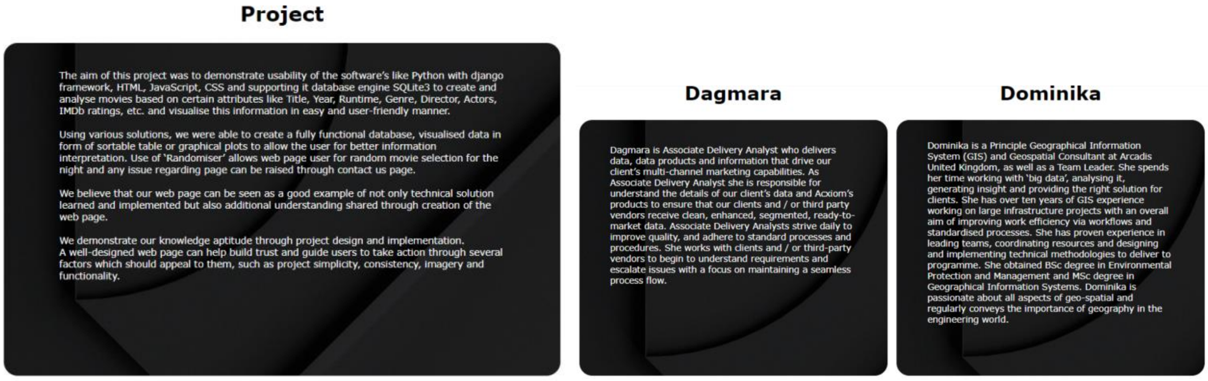
about
posters
movies
diagrams
randomiser
contact
MOVIE

Image 53: movies2021.pythonanywhere.com web page.

4.1 FUNCTIONALITY

To see and test web page functionality, tabs were created. Those allow users to view multiple panels of information within a single window. Using tabs users can switch between different sets of information. Those are:

1. **about** – this page contains all information about the project and creators of the project.



Project

The aim of this project was to demonstrate usability of the software's like Python with django framework, HTML, JavaScript, CSS and supporting it database engine SQLite3 to create and analyse movies based on certain attributes like Title, Year, Runtime, Genre, Director, Actors, IMDb ratings, etc. and visualise this information in easy and user-friendly manner.

Using various solutions, we were able to create a fully functional database, visualised data in form of sortable table or graphical plots to allow the user for better information interpretation. Use of 'Randomiser' allows web page user for random movie selection for the night and any issue regarding page can be raised through contact us page.

We believe that our web page can be seen as a good example of not only technical solution learned and implemented but also additional understanding shared through creation of the web page.

We demonstrate our knowledge aptitude through project design and implementation.
A well-designed web page can help build trust and guide users to take action through several factors which should appeal to them, such as project simplicity, consistency, imagery and functionality.

Dagmara

Dagmara is Associate Delivery Analyst who delivers data, data products and information that drive our client's multi-channel marketing capabilities. As Associate Delivery Analyst she is responsible for understand the details of our client's data and Acxiom's products to ensure that our clients and / or third party vendors receive clean, enhanced, segmented, ready-to-market data. Associate Delivery Analysts strive daily to improve quality, and adhere to standard processes and procedures. She works with clients and / or third-party vendors to begin to understand requirements and escalate issues with a focus on maintaining a seamless process flow.

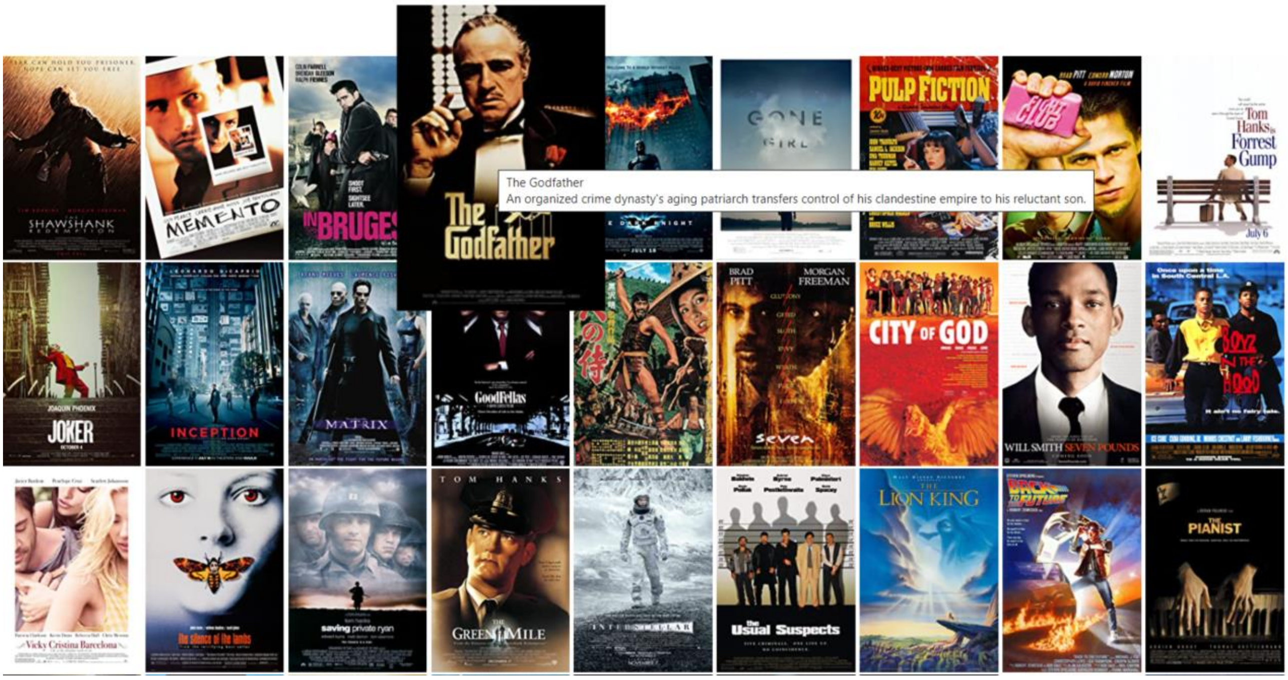
Dominika

Dominika is a Principle Geographical Information System (GIS) and Geospatial Consultant at Arcadis United Kingdom, as well as a Team Leader. She spends her time working with 'big data', analysing it, generating insight and providing the right solution for clients. She has over ten years of GIS experience working on large infrastructure projects with an overall aim of improving work efficiency via workflows and standardised processes. She has proven experience in leading teams, coordinating resources and designing and implementing technical methodologies to deliver to programme. She obtained BSc degree in Environmental Protection and Management and MSc degree in Geographical Information Systems. Dominika is passionate about all aspects of geo-spatial and regularly conveys the importance of geography in the engineering world.

Image 54: 'About' tab.

2. **posters**. In this tab of our website, you will find posters and short descriptions of the analysed movies. In our opinion, these are up to 100 movies that every cinema fan should watch. We utilise all movie genres - thrillers, dramas, comedies and even horrors. Our list includes movie classics such as The Matrix, The Godfather, Gladiator and Forrest Gump. The list also includes surprising titles such as In Bruges, which

has captured critics with the beautiful landscapes of Bruges (known as the Flemish Venice), and The Room, which was announced the world's worst film. **Which of the movies have you already watched?**



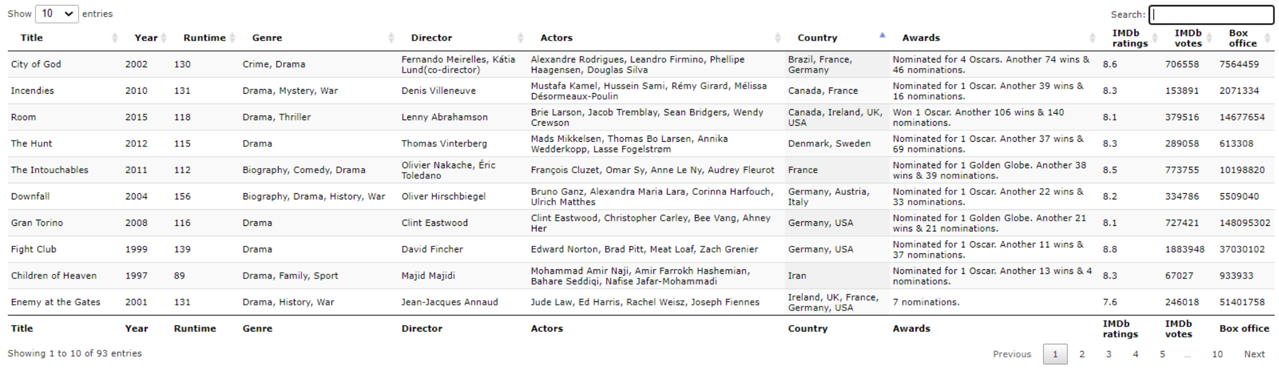
The Godfather
An organized crime dynasty's aging patriarch transfers control of his clandestine empire to his reluctant son.

Image 55: 'Posters' tab.

3. **movies**. The movies tab contains a sortable table containing detailed data of the movies analysed by us. By presenting the data in this format, you can easily sort movies according to the attributes. The attributes used by us are: **Title**, **Year** (of production), **Runtime** (duration in minutes), **Genre** (movie style), **Director**, main **Actors**, **Country** (of production), most important **Awards**, **IMDb ratings**, number of **IMDb votes** and **Box office**. It allows uses to see the most important or popular responses, visual information in ascending or descending order or sort it alphabetically.

   Number of the tabular entries can be manipulated to display movie value as 10, 25, 50 or 100. Alternatively, other movie selection can be viewed with help from **Next** or **Previous** buttons.

   The **search box** above the table allows for user input to be searched for in a database.



Show 10 entries

Search:

Title	Year	Runtime	Genre	Director	Actors	Country	Awards	IMDb ratings	IMDb votes	Box office
City of God	2002	130	Crime, Drama	Fernando Meirelles, Kátia Lund(co-director)	Alexandre Rodrigues, Leandro Firmino, Phellipe Haagensen, Douglas Silva	Brazil, France, Germany	Nominated for 4 Oscars. Another 74 wins & 46 nominations.	8.6	706558	7564459
Incendies	2010	131	Drama, Mystery, War	Denis Villeneuve	Mustafa Kamel, Hussein Sami, Rémy Girard, Mélissa Désormeaux-Poulin	Canada, France	Nominated for 1 Oscar. Another 39 wins & 16 nominations.	8.3	153891	2071334
Room	2015	118	Drama, Thriller	Lenny Abrahamson	Brie Larson, Jacob Tremblay, Sean Bridgers, Wendy Crewson	Canada, Ireland, UK, USA	Won 1 Oscar. Another 106 wins & 140 nominations.	8.1	379516	14677654
The Hunt	2012	115	Drama	Thomas Vinterberg	Mads Mikkelsen, Thomas Bo Larsen, Annika Wedderkopp, Lasse Fogelstrøm	Denmark, Sweden	Nominated for 1 Oscar. Another 37 wins & 69 nominations.	8.3	289058	613308
The Intouchables	2011	112	Biography, Comedy, Drama	Olivier Nakache, Éric Toledano	François Cluzet, Omar Sy, Anne Le Ny, Audrey Fleurot	France	Nominated for 1 Golden Globe. Another 38 wins & 39 nominations.	8.5	773755	10198820
Downfall	2004	156	Biography, Drama, History, War	Oliver Hirschbiegel	Bruno Ganz, Alexandra Maria Lara, Corinna Harfouch, Ulrich Matthes	Germany, Austria, Italy	Nominated for 1 Oscar. Another 22 wins & 33 nominations.	8.2	334786	5509040
Gran Torino	2008	116	Drama	Clint Eastwood	Clint Eastwood, Christopher Carley, Bee Vang, Ahney Her	Germany, USA	Nominated for 1 Golden Globe. Another 21 wins & 21 nominations.	8.1	727421	148095302
Fight Club	1999	139	Drama	David Fincher	Edward Norton, Brad Pitt, Meat Loaf, Zach Grenier	Germany, USA	Nominated for 1 Oscar. Another 11 wins & 37 nominations.	8.8	1883948	37030102
Children of Heaven	1997	89	Drama, Family, Sport	Majid Majidi	Mohammad Amir Naji, Amir Farrokh Hashemian, Bahare Seddiqi, Nafise Jafar-Mohammadi	Iran	Nominated for 1 Oscar. Another 13 wins & 4 nominations.	8.3	67027	933933
Enemy at the Gates	2001	131	Drama, History, War	Jean-Jacques Annaud	Jude Law, Ed Harris, Rachel Weisz, Joseph Fiennes	Ireland, UK, France, Germany, USA	7 nominations.	7.6	246018	51401758
Title	Year	Runtime	Genre	Director	Actors	Country	Awards	IMDb ratings	IMDb votes	Box office

Showing 1 to 10 of 93 entries

Image 56: 'Movies' tab.

4. **diagrams.** It is a graphical representation of the top movie selection, based of carefully selected categories, allowing for quick and easy information visualization. This technique allows us to clarify, interpret and analyse data and relationships between them to significantly reduce time and effort put into the overall

analysis of this information. In this section you will find graphs related to all 100 movies analysed by us. You can browse through the most interesting suggestions and learn about their basic parameters.

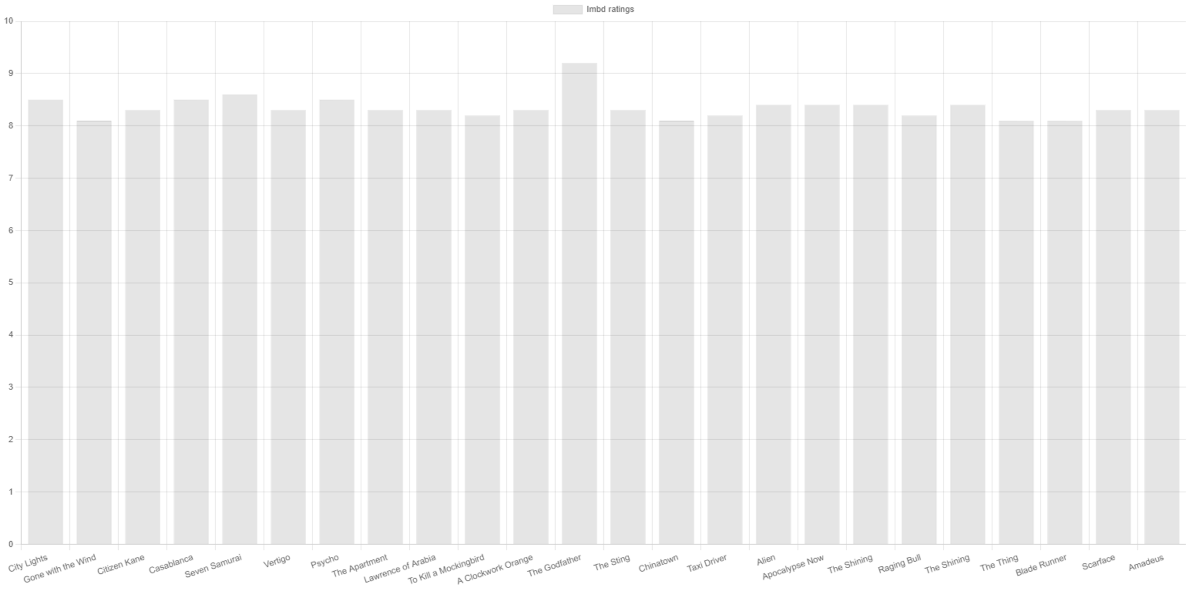


Image 57: 'Diagrams' tab with plot for IMDb year <1985.

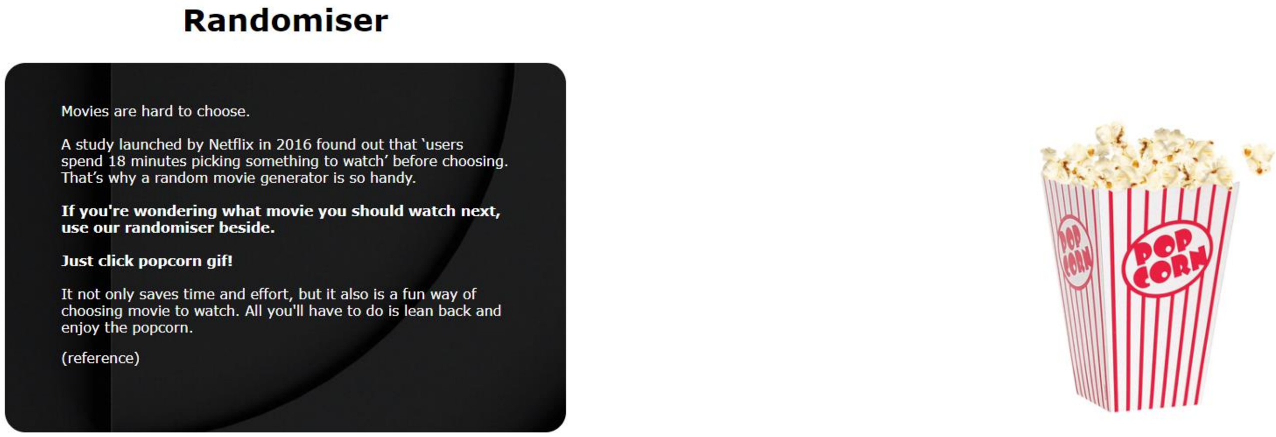
Diagrams are categories by year related ranges:

- <1985
- > = 1985 and <2005
- > = 2005

We also have graphs related to overall scores, for each of these year ranges, based on:

- imdb rating (IMDb registered users can cast a vote (from 1 to 10) on every released title in the database. Individual votes are then aggregated and summarized as a single IMDb rating)
- imdb votes (number of IMDb register users casting their vote on a movie, one vote per title per user)
- box office (rating of number of tickets sold to the public admission to a movie event and amount of money raised by ticket sales).

5. **randomiser** – it allows users to pick a random movie to watch. By clicking on icon, a movie will be selected in a random order.



Randomiser

Movies are hard to choose.

A study launched by Netflix in 2016 found out that 'users spend 18 minutes picking something to watch' before choosing. That's why a random movie generator is so handy.

If you're wondering what movie you should watch next, use our randomiser beside.

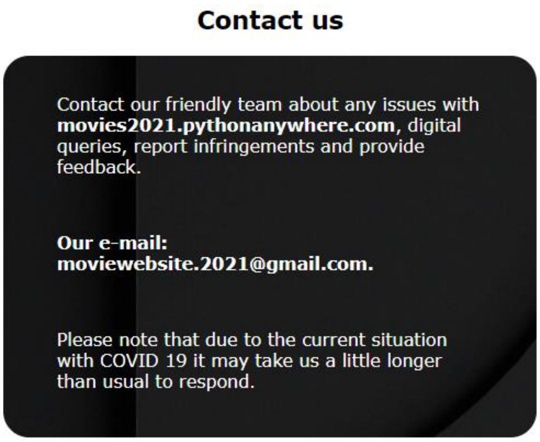
Just click popcorn gif!

It not only saves time and effort, but it also is a fun way of choosing movie to watch. All you'll have to do is lean back and enjoy the popcorn.

(reference)

Image 58: 'Randomiser' tab.

6. **contact (us)** – is a page where users can about ways in which to contact out team and post or raise an issue, ask digital queries, report infringements and provide feedback about our web page.



Contact us

Contact our friendly team about any issues with **movies2021.pythonanywhere.com**, digital queries, report infringements and provide feedback.

Our e-mail:
moviewebsite.2021@gmail.com.

Please note that due to the current situation with COVID 19 it may take us a little longer than usual to respond.

Image 59: Contact us tab.

5 SUMMARY

The aim of this project was to demonstrate usability of **software** like Python with django framework, HTML, JavaScript, CSS and supporting it database engine SQLite3 to **create and analyse movies** based on certain **attributes** like Title, Year, Runtime, Genre, Director, Actors, IMDb ratings, etc. and **visualise** this information in easy and user-friendly manner.

Using various solutions, we were able to create a fully functional database, visualised data in form of **sortable table** or **graphical diagrams** to allow the user for better information interpretation. Use of '**Randomiser'** allows **web page** user for random movie selection for the night and any issue regarding page can be raised through **contact us** page.

We believe that our web page can be seen as a good example of not only technical solution learned and implemented but also additional understanding shared through the creation of this web page.

We demonstrate our knowledge aptitude through project design and implementation. A well-designed web page can help build trust and guide users to take action through factors which should appeal to them, such as **project simplicity**, **consistency**, **imagery** and **functionality**.

6 REFERENCES

(1) OMDb API source: http://www.omdbapi.com/

(2) IMDb Wikipedia source: https://datasets.imdbws.com/

(3) OMDb API source: https://rapidapi.com/blog/list-of-online-movie-databases/

(4) OMDB API licensing disclaimer source: https://creativecommons.org/licenses/by-nc/4.0/

(5) Python 3.6.2 download source: https://www.python.org/downloads/

(6) Django framework source: https://www.djangoproject.com/

(7) JavaScript, HTML, CSS source: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

(8) Git source: https://git-scm.com/

(9) Git download source: https://git-scm.com/downloads

(10) SQLite3 source: https://www.sqlite.org/index.html

(11) Web page source: Web page - Wikipedia

(12) PythonAnywhre source: https://en.wikipedia.org/wiki/PythonAnywhere

(13) PythonAnywhere source: https://www.pythonanywhere.com/

(14) Project Movies source: http://movies2021.pythonanywhere.com/about

MOVIE