MOVIE ANALYSIS BASED ON OMDB DATABASE

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

MAY 2021

CONTACTS

DAGMARA ZADROŻNA
Author

DOMINIKA SAWICKA
Author


MOVIE

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.

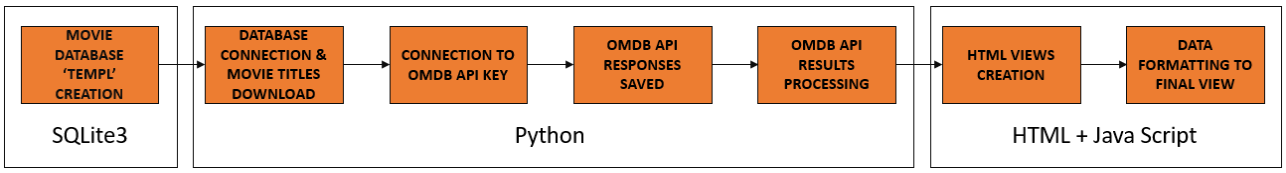

Image1: Project workflow

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.

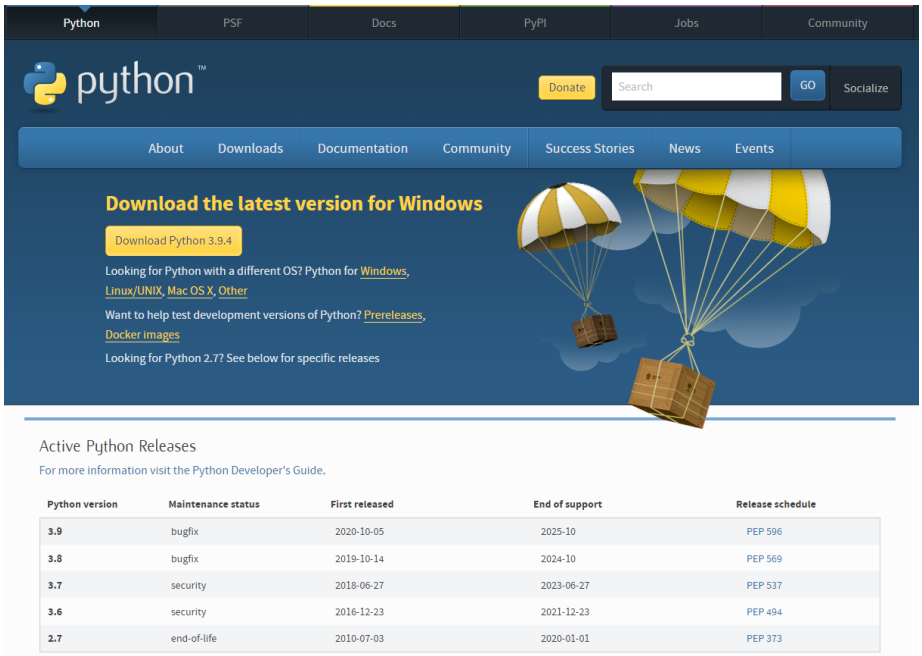

Image 3: Python download web page.

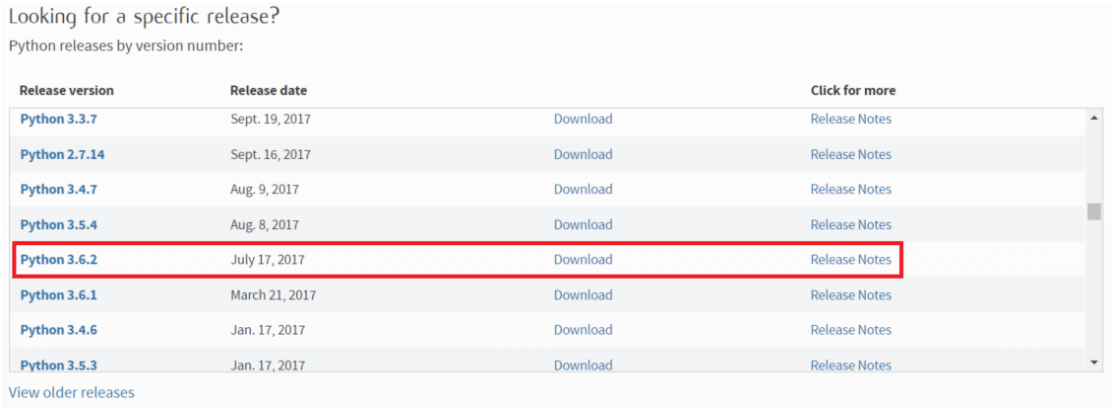

Image 4: Python 3.6.2 selection.

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

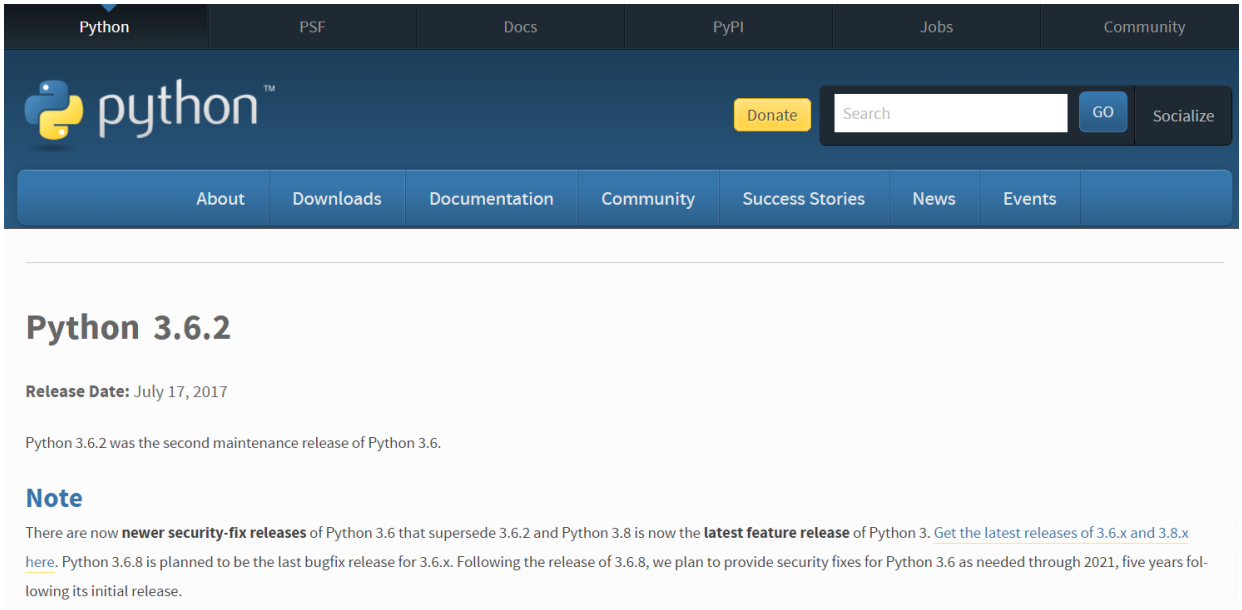

Image 5: Python 3.6.2 details.

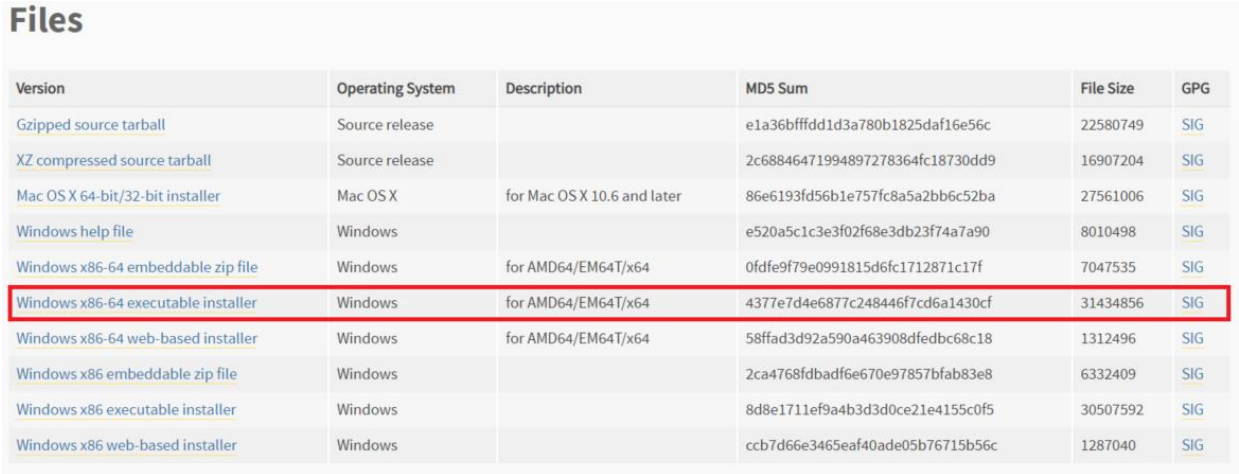

Image 6: Python 3.6.2 executable file selection.

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:

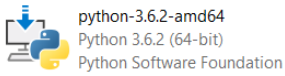

Image 7: Python 3.6.2 executable file icon.

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.

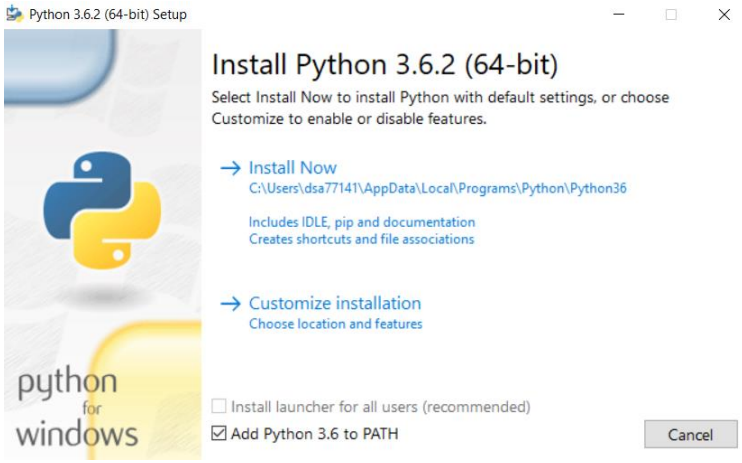

Image 8: Python 3.6.2 installation setup.

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.

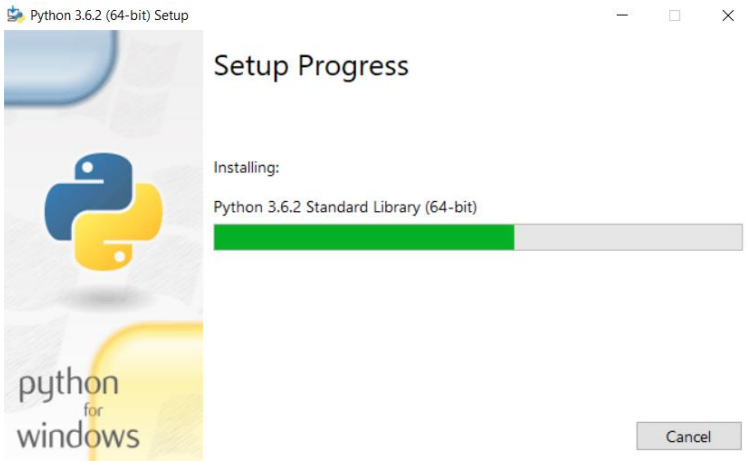

Image 9: Python 3.6.2 installation setup progress.

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.

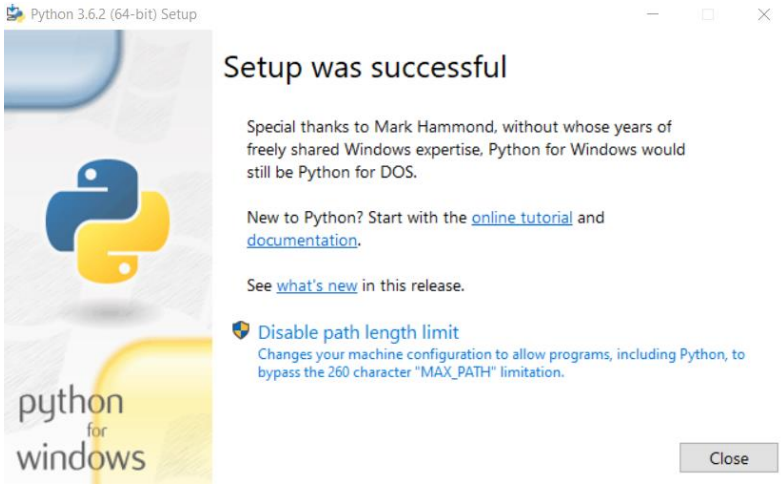

Image 10. Python 3.6.2 installation message.

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.

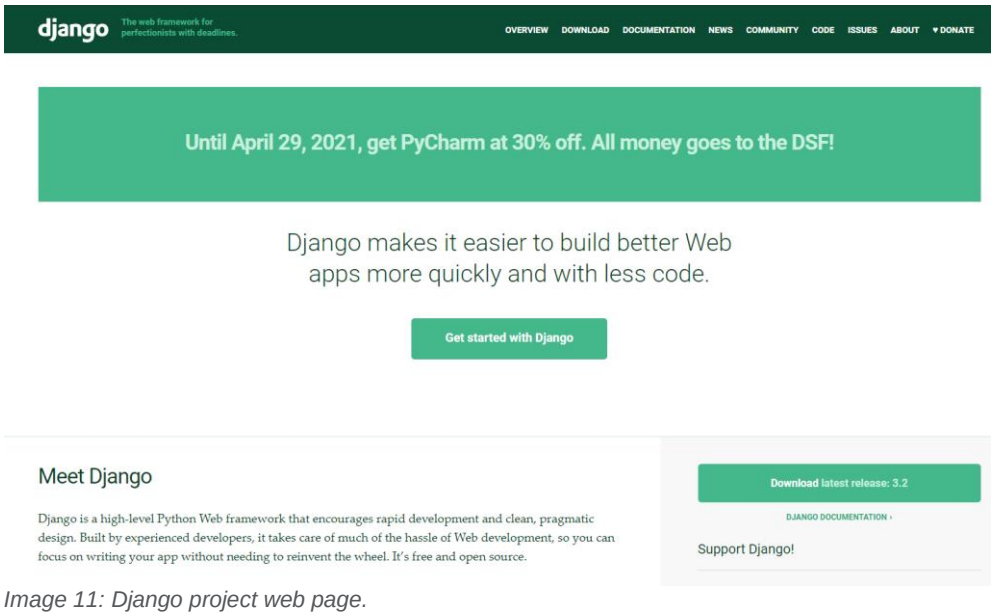

Image 11: Django project web page.

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.**

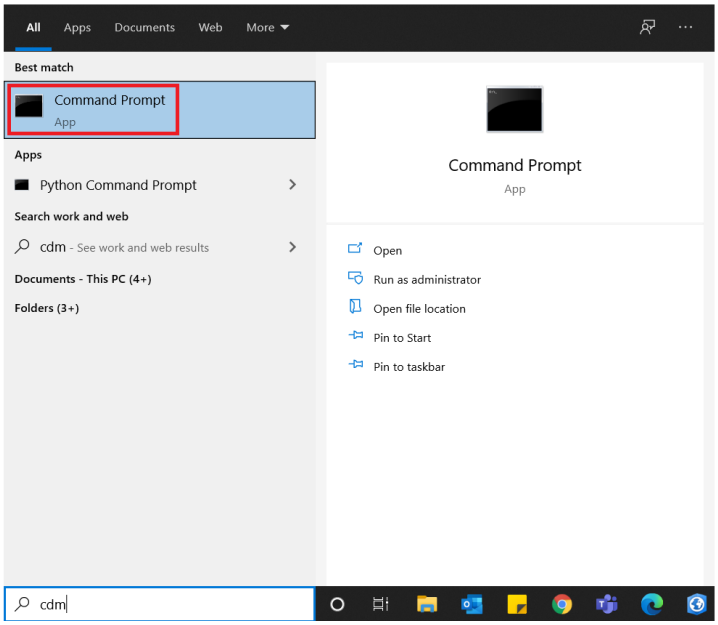

Image 12: Command Prompt access.

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*.

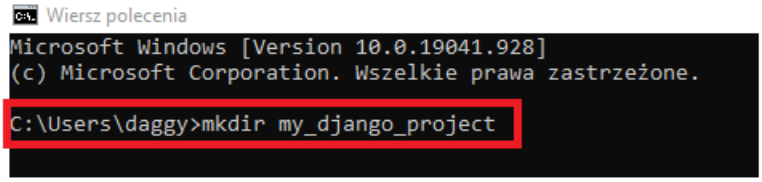

Image 13: Creation of the Development directory.

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

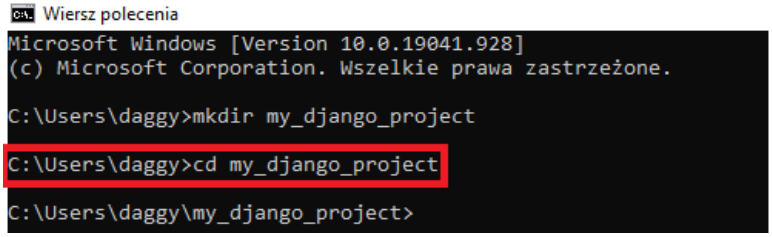

Image 14: Creation of the project catalog.

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

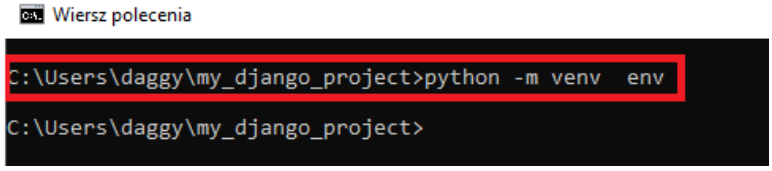

Image 15: Creation of the virtual environment.

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

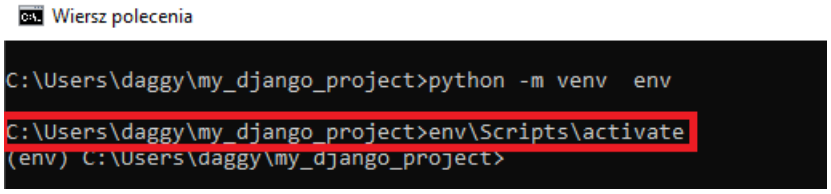

Image 16: Activation of the virtual environment in Command Prompt.

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.

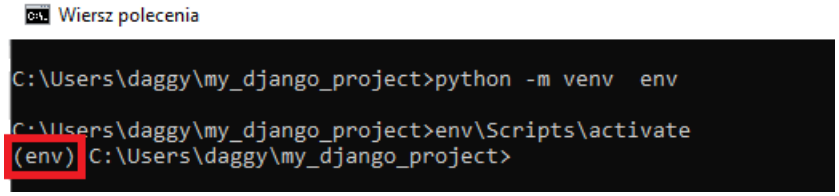

Image 17: Confirmation of the virtual environment activation in Command Prompt.

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

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

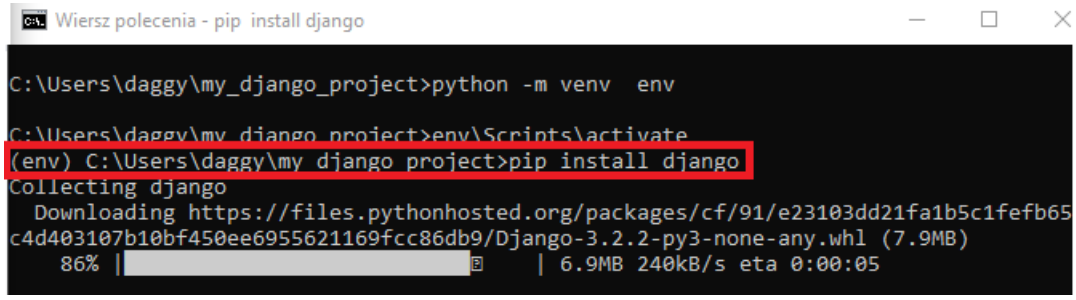

Image 18: Django installation command in Command Prompt.

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

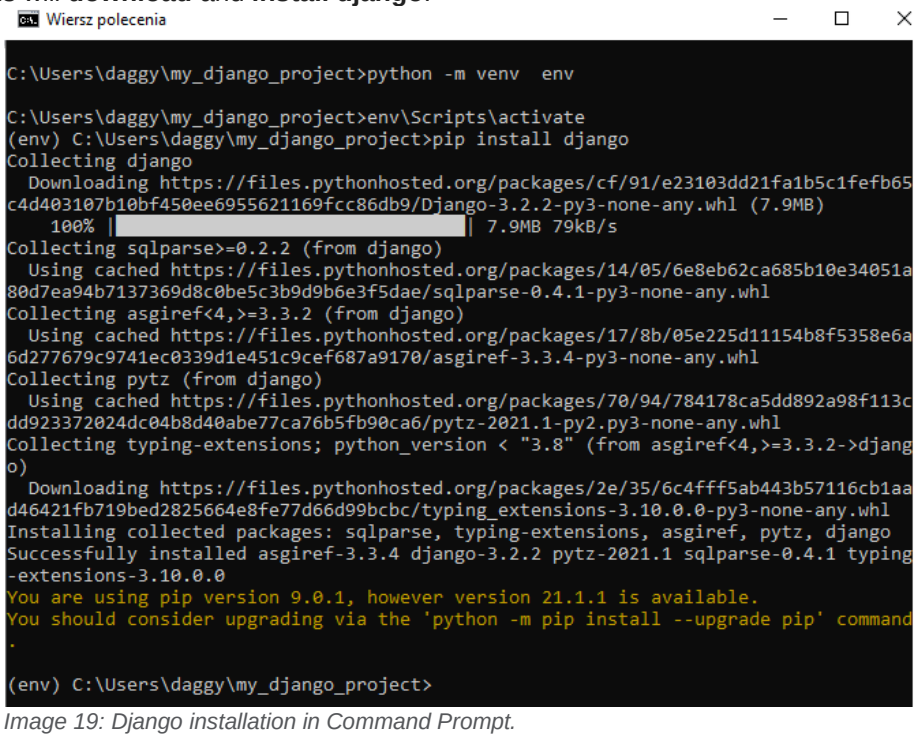

Image 19: Django installation in Command Prompt.

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.*

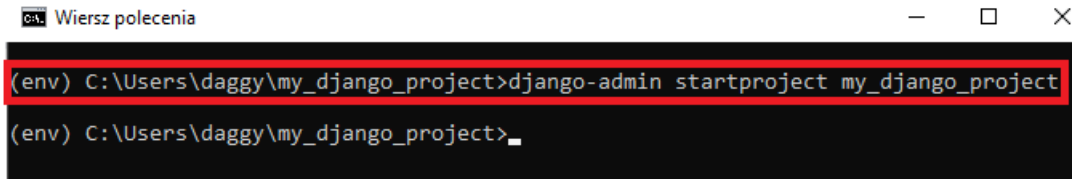

Image 20: Creation of django project.

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

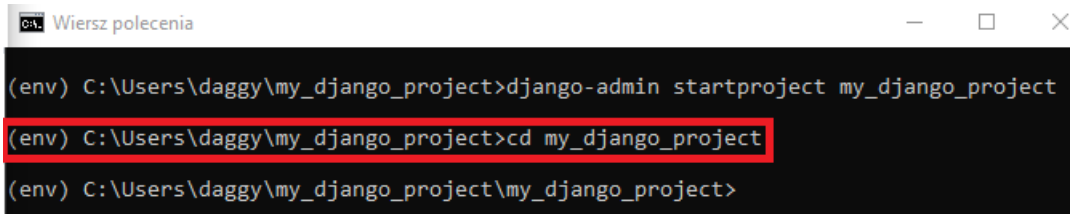

Image 21: Creation of django project folder.

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:

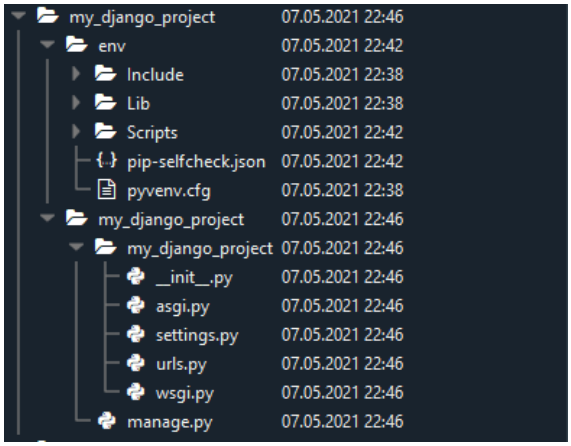

Image 22: Django project folder structure viewed in Spyder.

Exit using command: *cdm.*

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

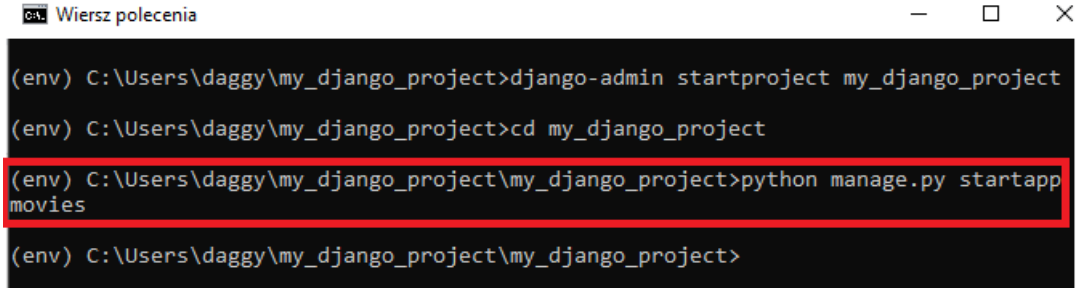

Image 23: Start of the django application.

e. Bring directory into text editor.

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

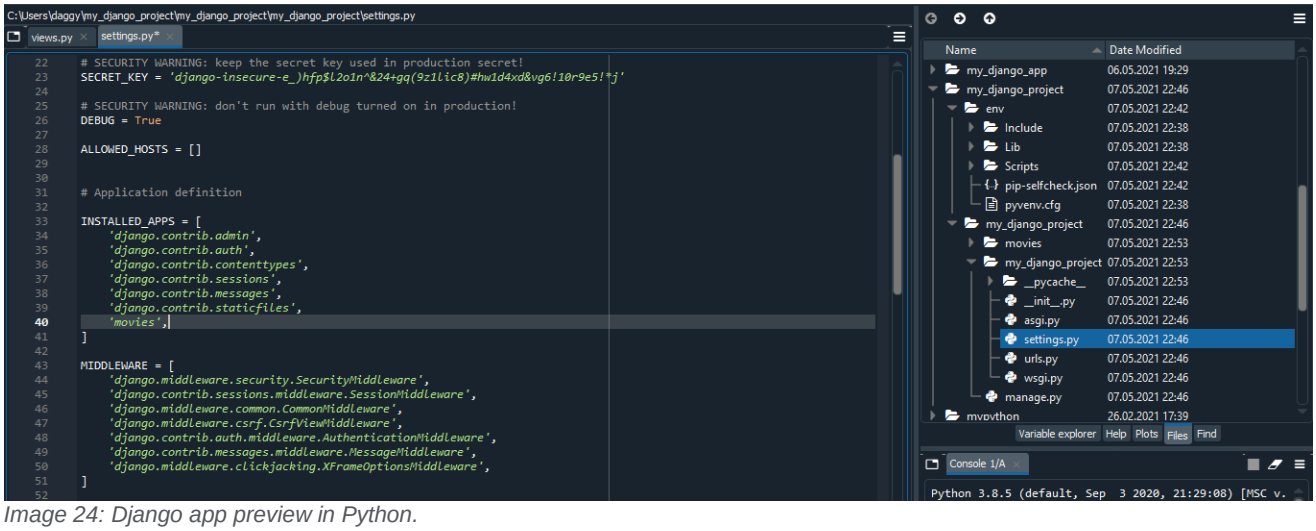

Image 24: Django app preview in Python.

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

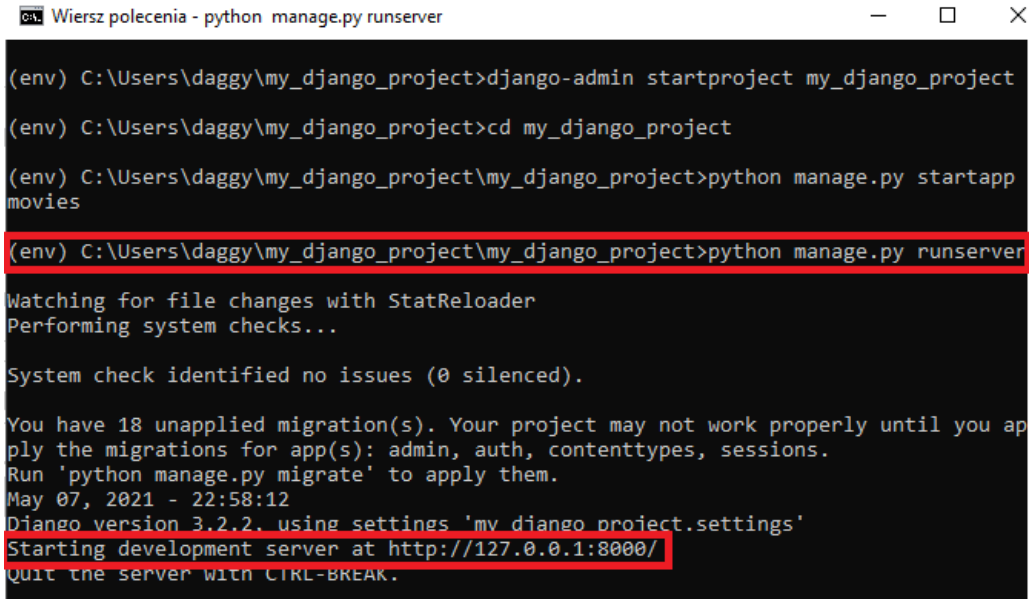

Image 25: Development Server command and location.

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

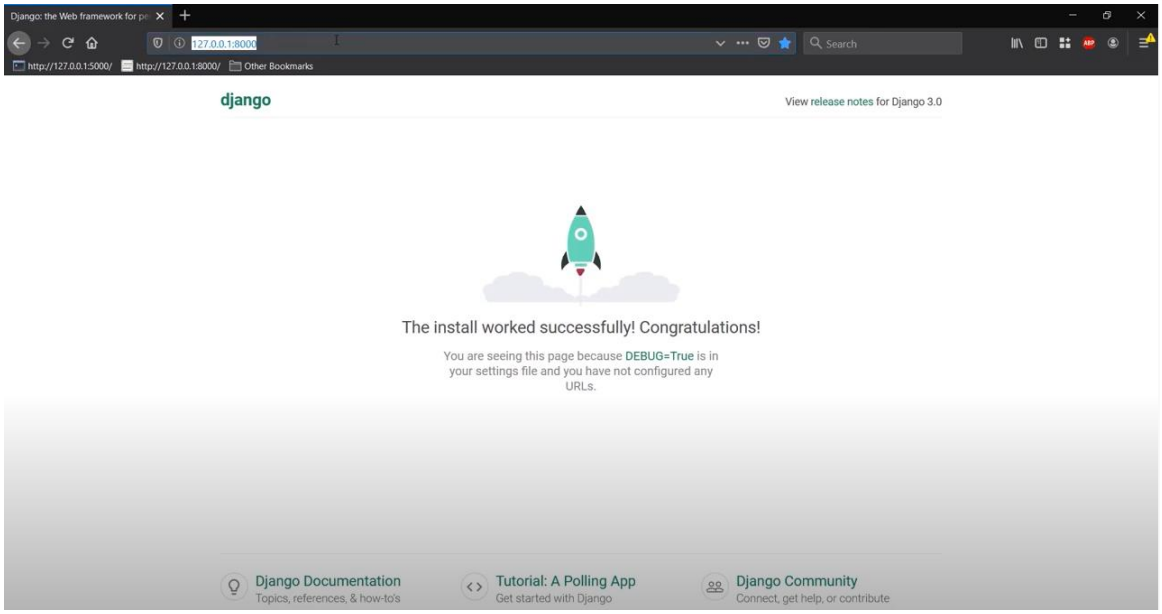

Image 26: Development Server location in web browser.

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**).

Image 27: Git application download.

b. The file should appear as.

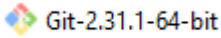

Image 28: Git 2.31.1-64-bit executable file icon.

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.

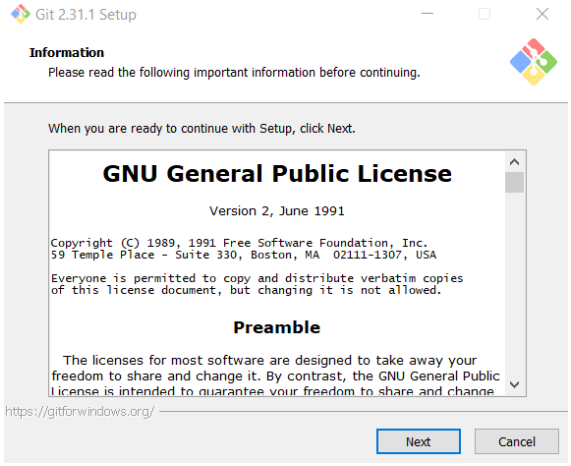

Image 29: Git-2.31.1 Setup information.

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

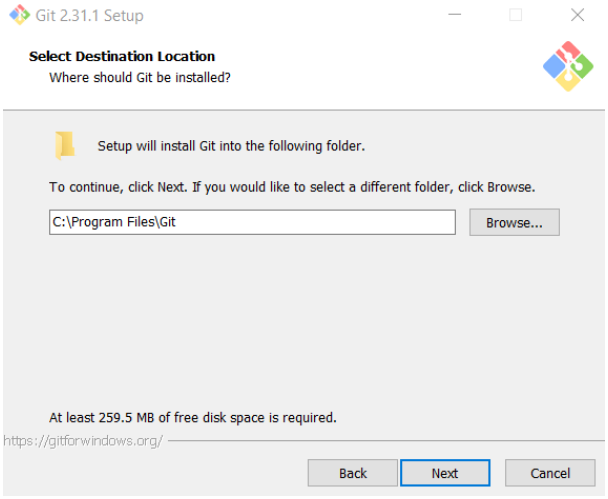

Image 30: Git-2.31.1 Setup Select Destination Location.

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

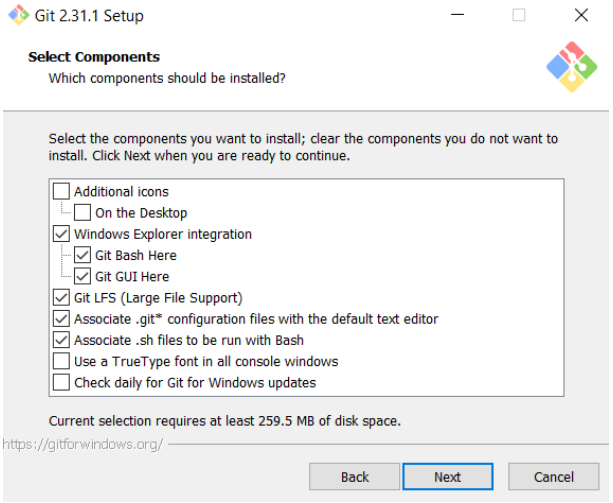

Image 31: Git-2.31.1 Setup Select Components.

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

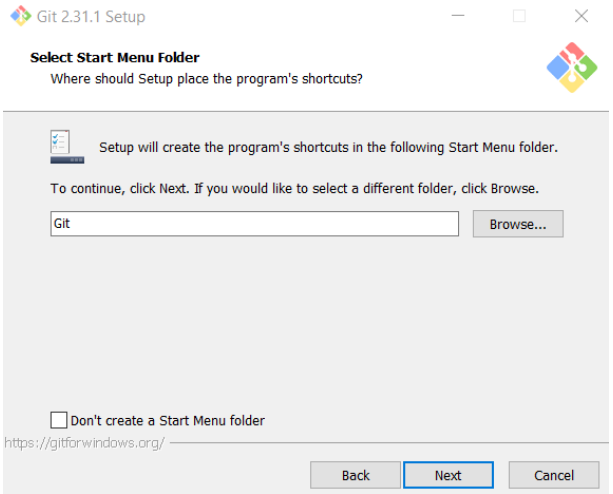

Image 32: Git-2.31.1 Setup Select Start Menu Folder.

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

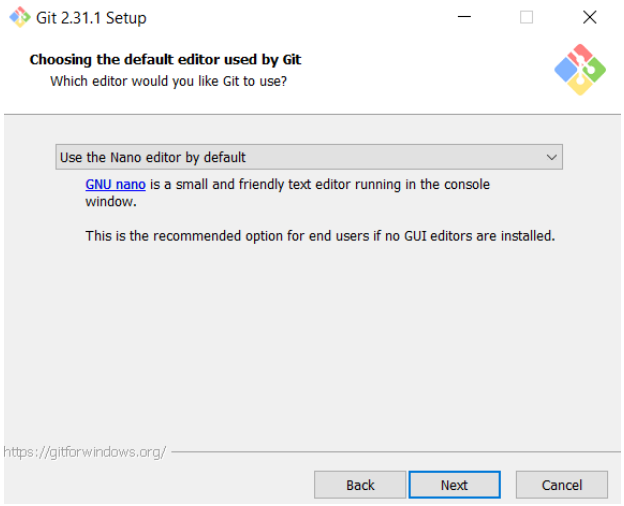

Image 33: Git-2.31.1 Setup Choose the default editor used by Git.

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

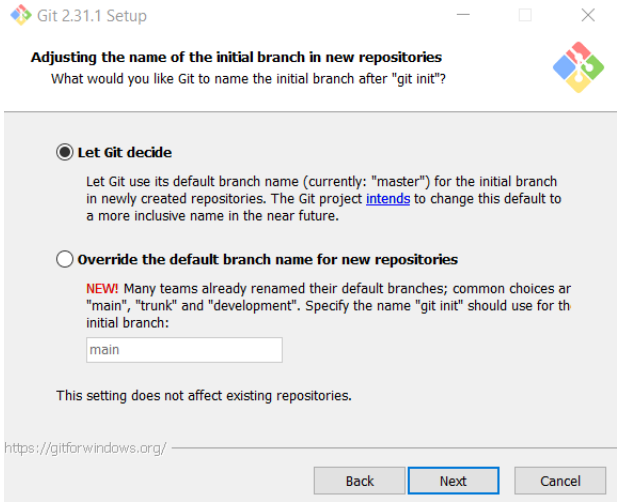

Image 34: Git-2.31.1 Setup Adjust the name of the initial branch in new repositories.

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

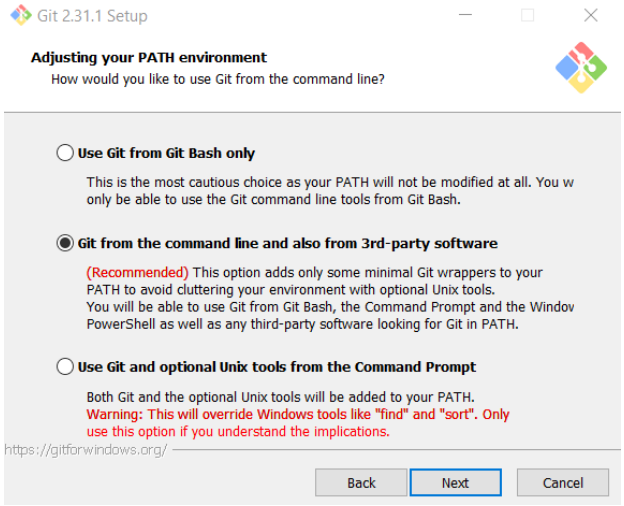

Image 35. Git-2.31.1 Setup Adjust your PATH environment.

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

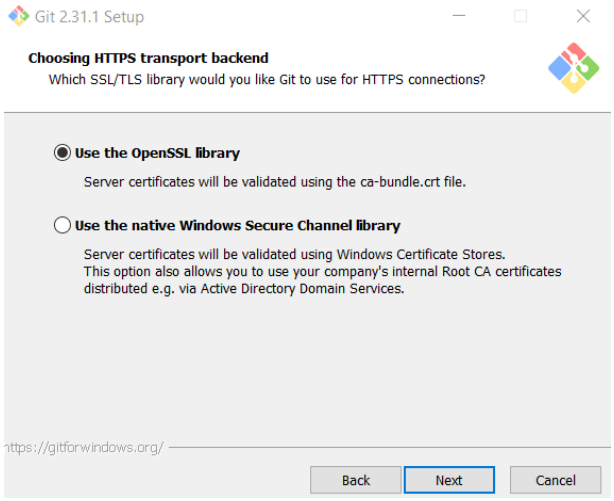

Image 36. Git-2.31.1 Choosing HTTPS transport backend.

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

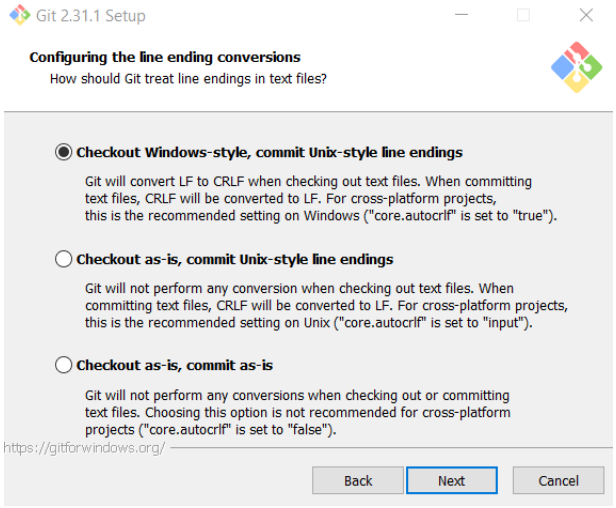

Image 37. Git-2.31.1 Setup Configurating the line ending conversation.

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

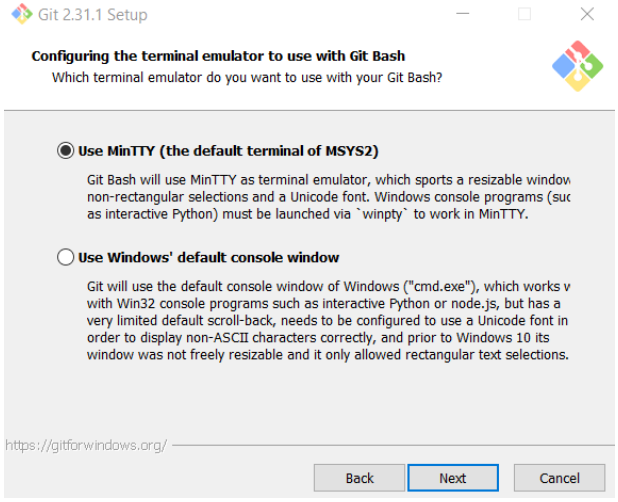

Image 38. Git-2.31.1 Setup Configurating the terminal emulator to use with Git Bash.

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

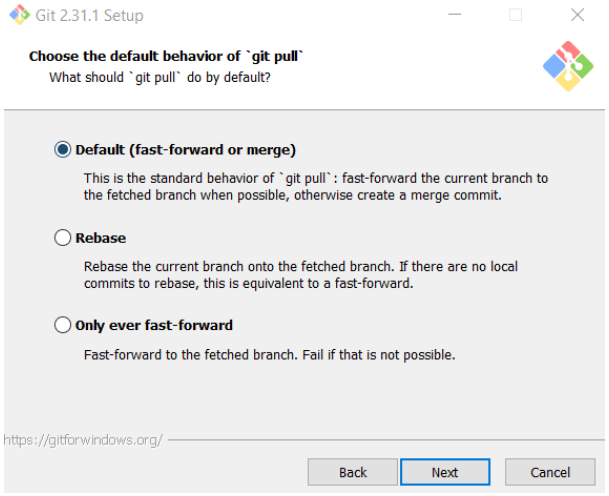

Image 39. Git-2.31.1 Setup Choose the default behaviour of the ‘git pull’.

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

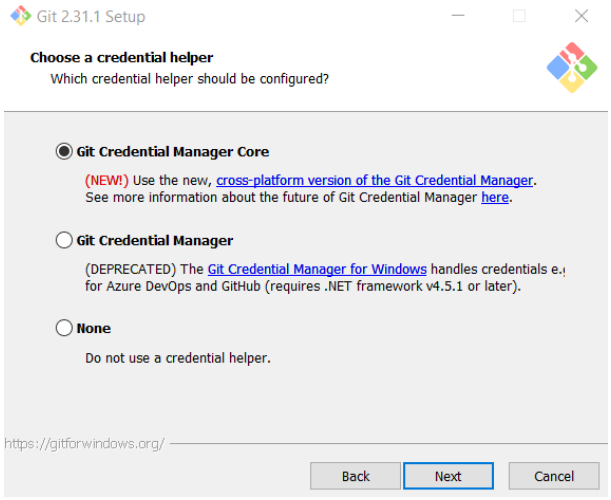

Image 40. Git-2.31.1 Setup Choose a credential helper.

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

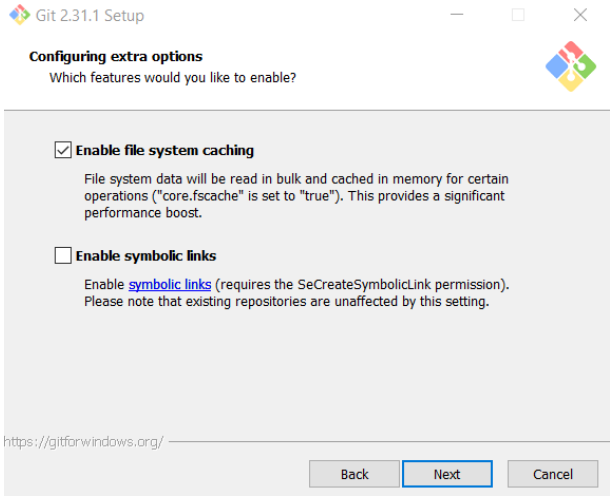

Image 41. Git-2.31.1 Setup Configuring extra options.

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

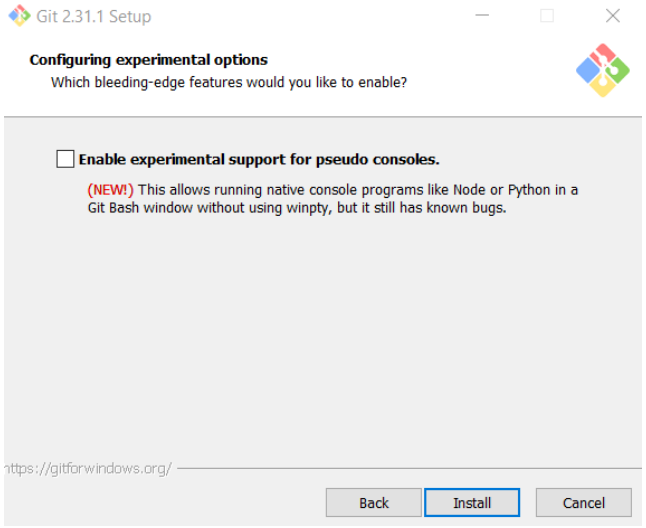

Image 42. Git-2.31.1 Setup Configuring experimental options.

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

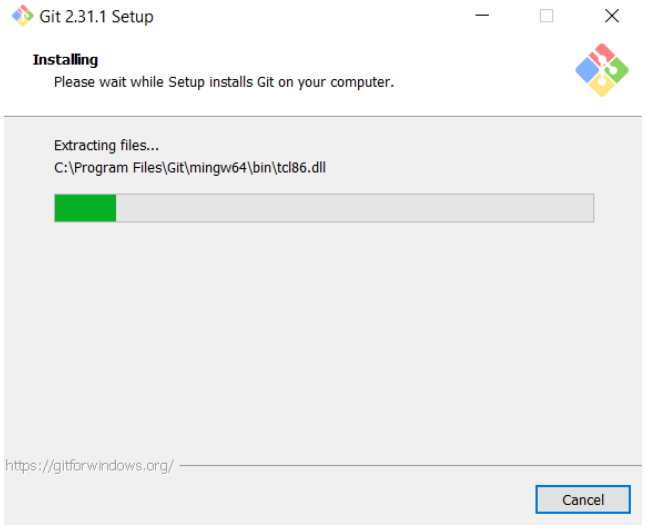

Image 43. Git-2.31.1 Setup Installation.

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:

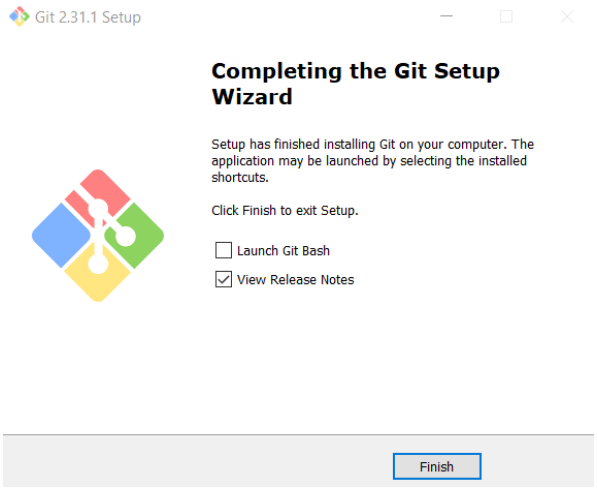

Image 44. Git-2.31.1 Setup Wizard installation completion.

Git should now be installed.

x. Create a public repository on the GitHub.

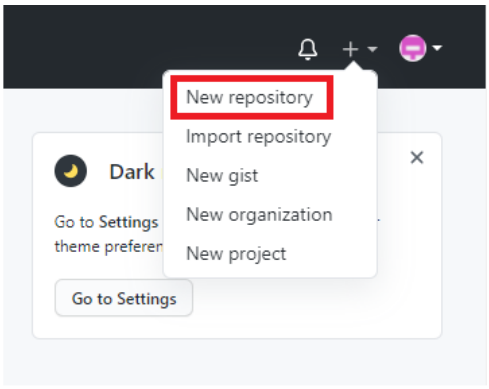

Image 45. Creation of public repository in GitHub.

y. Fill in details about the repository.

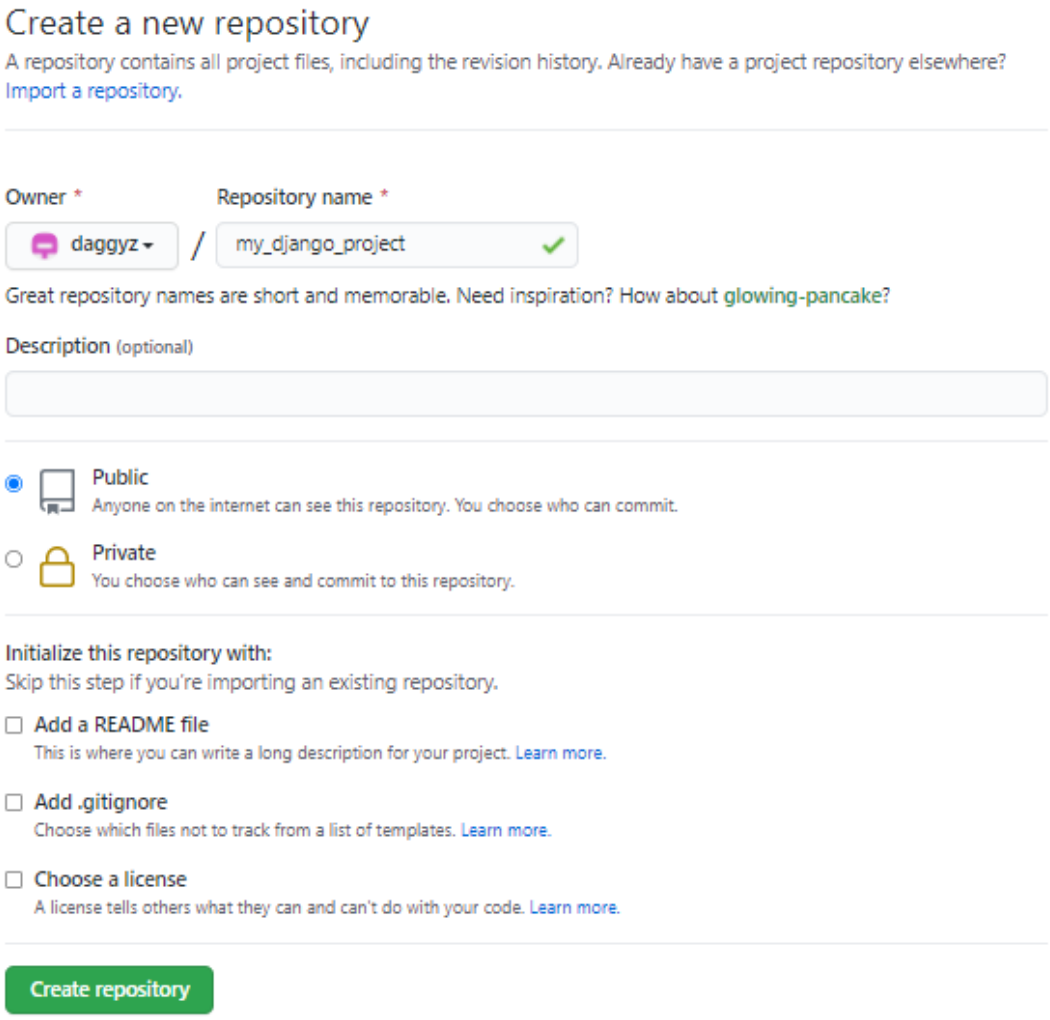

Image 46. Public repository details in GitHub.

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 or website) 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.*

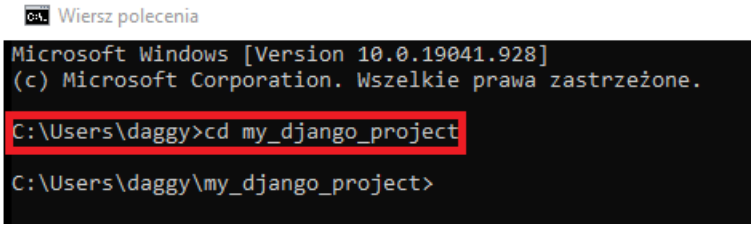

Image 47. Project location selection in Command Prompt.

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

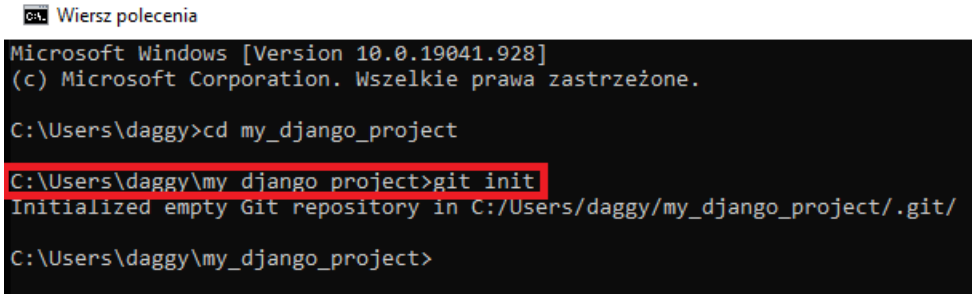

Image 48. Connection initiation with GitHub in Command Prompt.

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”.

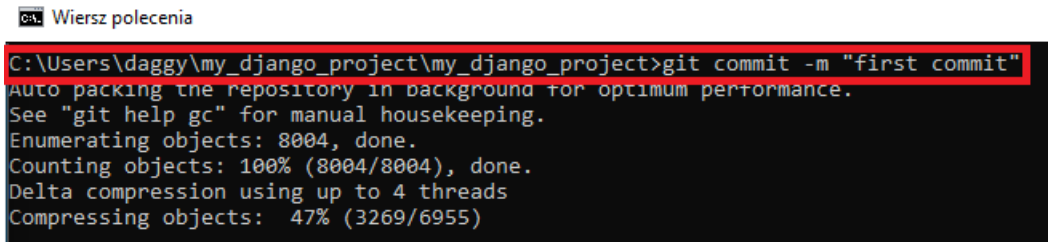

Image 49. State of a project in Command Prompt.

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

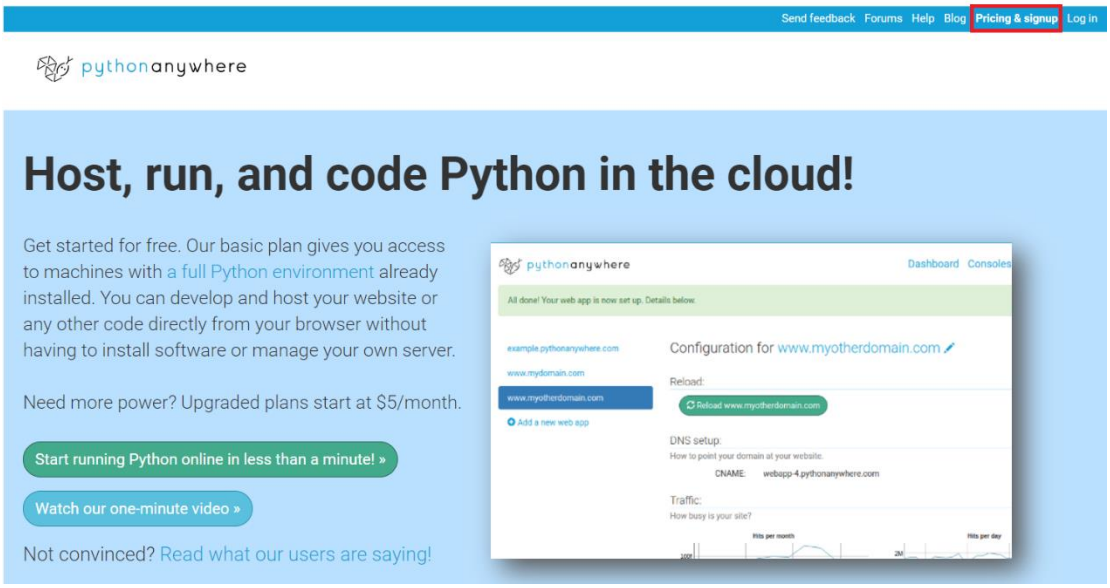

Image 50. PythonAnywhere web site.

c. Create an account:

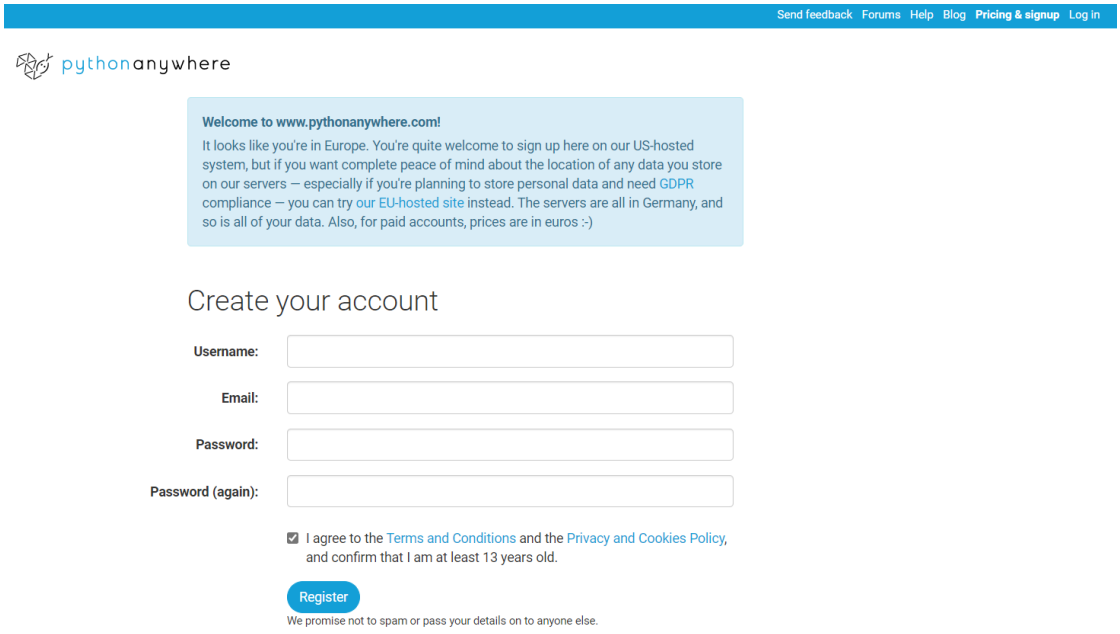

Image 51. PythonAnywhere account creation.

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

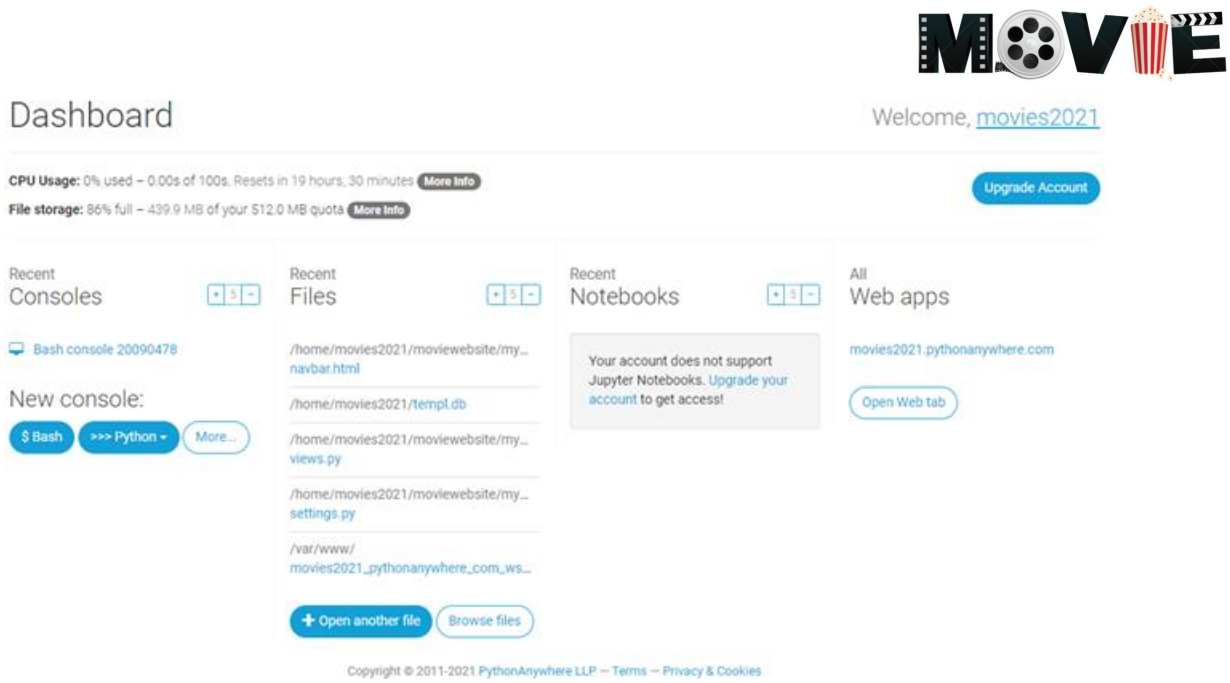

Image 52. PythonAnywhere Dashboard configuration.

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

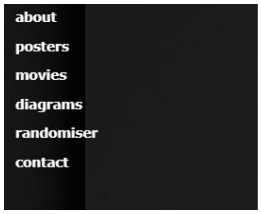

Image 53: movies2021.pythonanywhere.com web page.

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.

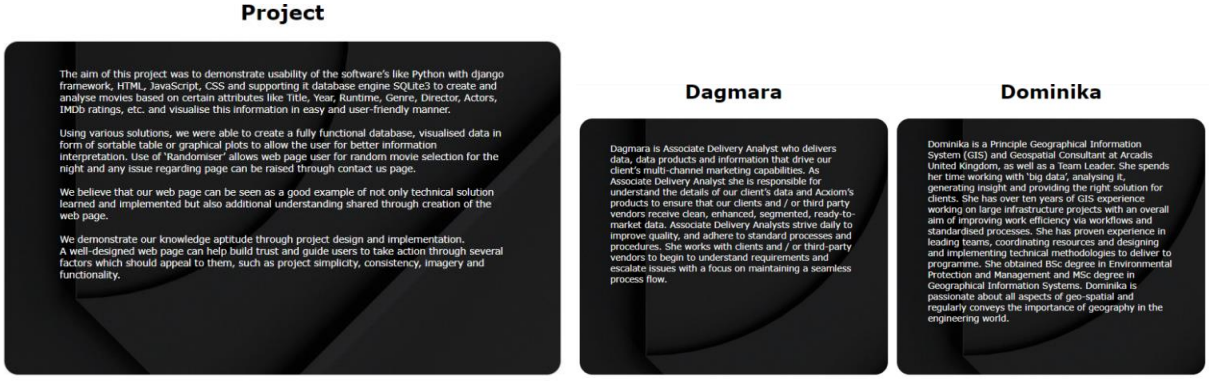

Image 54: 'About' tab.

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?**

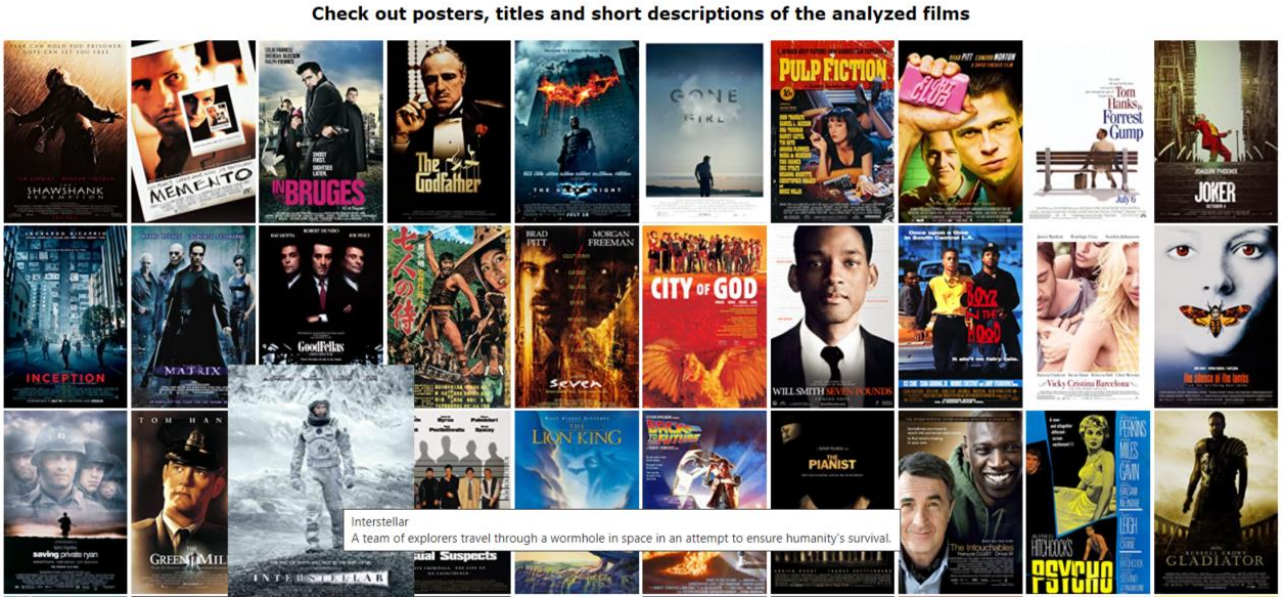

Image 55: 'Posters' tab.

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 Previous 1 2 3 4 5 … 10 Next

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.

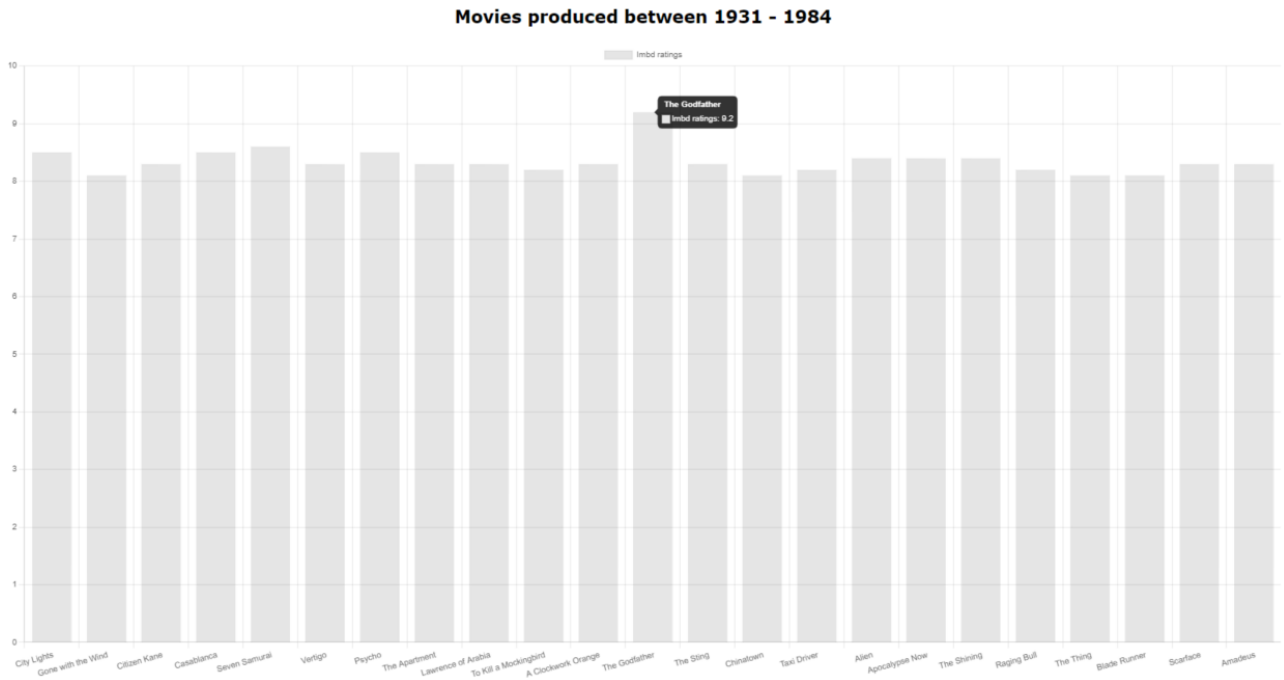

Image 57: 'Diagrams' tab with plot. IMDb rating for movies between 1931-1984.

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.

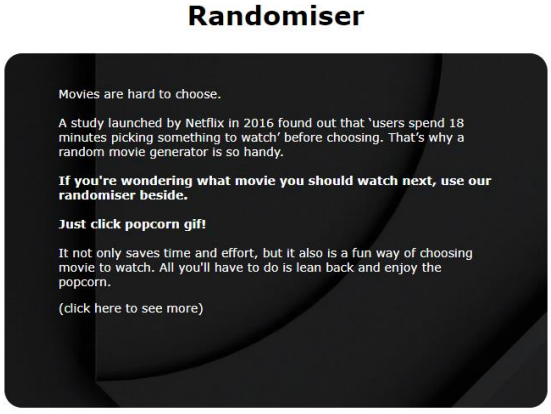

Image 58: 'Randomiser' tab.

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.

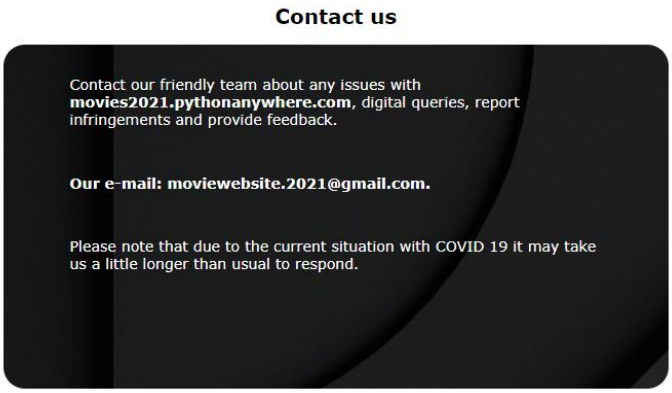

Image 59: Contact us tab.

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 guide users to the source of information and build trust in the data presented. We believe that our project web page will appeal to users due to **simplicity** of the design, **functionality**, **consistency** and basic **aesthetic values**.

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