



Kontakt:

Uwaga

Wiadomości email sprawdzam raz w tygodniu – po wysłaniu wiadomości proszę przez tydzień jej nie ponawiać. W przypadku, kiedy nie odpowiem proszę o ponowne przesłanie wysłanie oraz wiadomość na MTeams (prywatną wiadomość)

Email: szymon.guzik@gdansk.merito.pl

LUB



Zasady zaliczenia przedmiotu:

- Zaliczenie laboratorium odbywa się z wykorzystaniem platformy Moodle
- Na każdym laboratorium wykonywane jest zadanie lub realizacja swojego projektu
- Z każdego ćwiczenia można zdobyć maksymalnie 100 pkt.
- Obecność na laboratoriach jest obowiązkowa
- Dodanie pliku z rozwiązaniem zadania z laboratorium lub przesłanie fragmentu projektu dot. omawianego zagadnienia (2 tygodnie od laboratorium z zadaniem do wykonania)
- Przesłane pliki będą sprawdzane w miarę na bieżąco - po sprawdzeniu od razu otrzymacie pkt za przesłane zadanie
- Ocena:
 - 150 pkt - 3.0
 - 180 pkt - 3.5
 - 210 pkt - 4.0
 - 240 pkt - 4.5
 - 270 pkt - 5.0

Dlaczego Django ?

Cały świat ▼

2004 – dziś ▼

Wszystko ▼

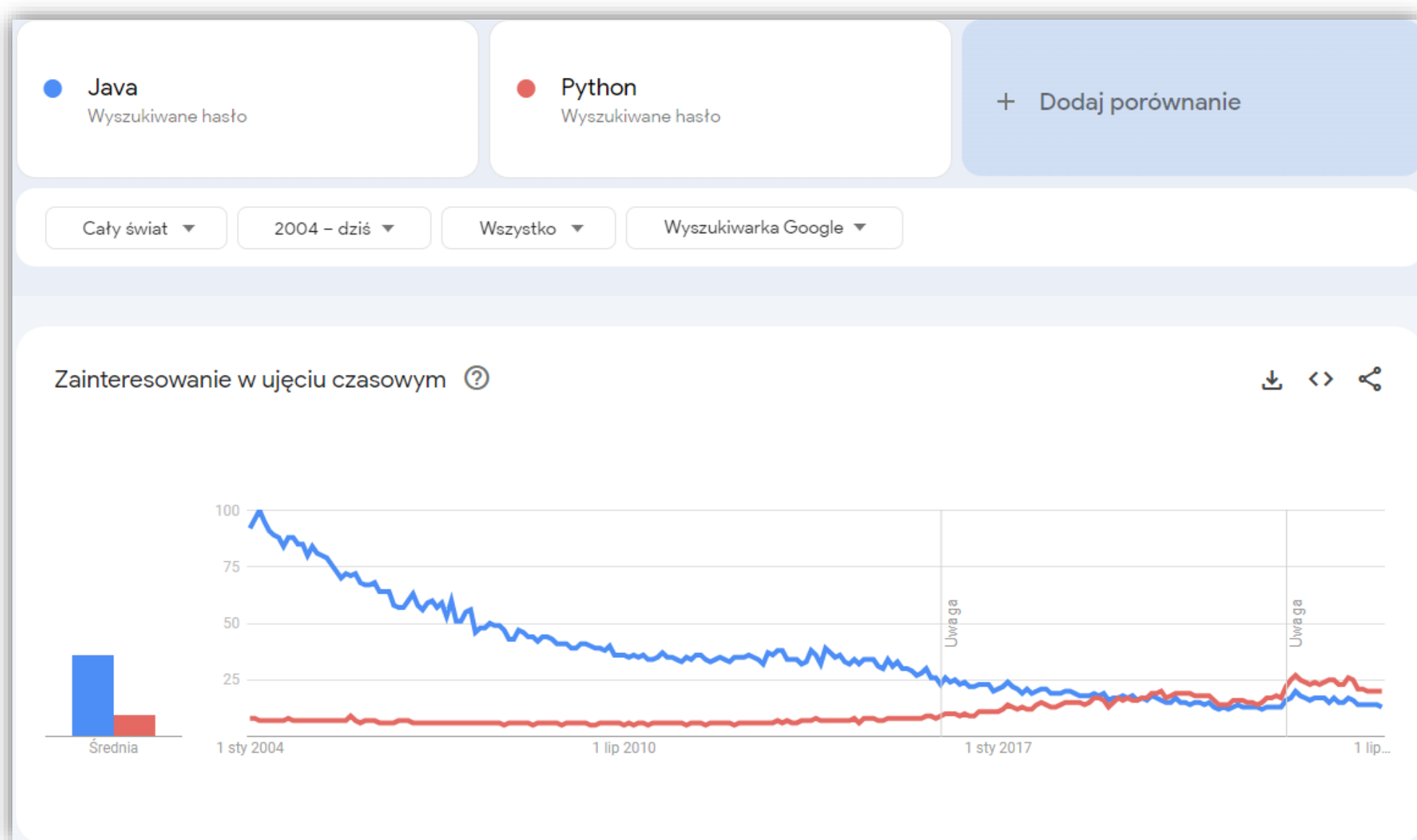
Wyszukiwarka Google ▼

Zainteresowanie w ujęciu czasowym ?



django

Dlaczego Django ?



Dlaczego Django ?

Python

[illegible]

Dlaczego Python ? Dlaczego Django ?

Python stał się jednym z najpopularniejszych języków programowania na świecie i ma wiele zalet, które sprawiają, że warto się go uczyć. Oto kilka powodów, dla których warto nauczyć się Pythona:

- **Łatwość nauce i czytelność:** Python jest znany z prostej składni, która jest bardzo czytelna. Dzięki temu jest doskonałym językiem dla początkujących programistów.
- **Wszechstronność:** Pythona można używać w wielu dziedzinach, od tworzenia stron internetowych (za pomocą Django lub Flask), przez analizę danych (pandas, numpy), uczenie maszynowe (TensorFlow, scikit-learn), aż po automatykę i scripting.
- **Duża społeczność:** Python ma ogromną społeczność, co oznacza, że są dostępne liczne zasoby, takie jak biblioteki, narzędzia oraz fora i grupy wsparcia, które pomogą w rozwiązaniu problemów.
- **Rozwój kariery:** Z powodu rosnącej popularności Pythona, umiejętność programowania w tym języku jest bardzo ceniona przez pracodawców, zwłaszcza w dziedzinach związanych z analizą danych, sztuczną inteligencją czy web developmentem.
- **Biblioteki i frameworki:** Python ma rozbudowany ekosystem bibliotek i narzędzi, które czynią go odpowiednim dla różnorodnych zastosowań. Niezależnie od tego, czy chodzi o analizę danych, tworzenie interfejsów użytkownika czy tworzenie gier, istnieją odpowiednie narzędzia dla Pythona.

Dlaczego Python ? Dlaczego Django ?

Python stał się jednym z najpopularniejszych języków programowania na świecie i ma wiele zalet, które sprawiają, że warto się go uczyć. Oto kilka powodów, dla których warto nauczyć się Pythona:

- **Przenośność:** Python jest wieloplatformowy, co oznacza, że można go uruchomić na różnych systemach operacyjnych.
- **Integracja:** Python może być łatwo zintegrowany z innymi językami, takimi jak C, C++ czy Java, co pozwala na korzystanie z najlepszych aspektów różnych technologii.
- **Rosnący rynek IoT:** Python jest często stosowany w rozwiązaniach Internetu Rzeczy (IoT) dzięki swojej prostocie i wszechstronności.
- **Wsparcie dla uczenia maszynowego i sztucznej inteligencji:** W ostatnich latach Python stał się dominującym językiem w dziedzinie uczenia maszynowego, dzięki bibliotekom takim jak TensorFlow, Keras czy scikit-learn.
- **Otwarte źródło:** Jako język open-source, Python jest bezpłatny do użytku i ma korzyści wynikające z aktywnej i zaangażowanej społeczności

Dlaczego Python ? Dlaczego Django ?

Django to jeden z najbardziej popularnych frameworków do tworzenia aplikacji internetowych opartych na Pythonie. Oto kilka powodów, dla których warto nauczyć się Django:

- **"Baterie w zestawie"**: Jednym z haseł Django jest "framework z bateriami w zestawie". Oznacza to, że Django dostarcza wiele wbudowanych narzędzi i funkcji, które przyspieszają rozwój aplikacji, takich jak panele administracyjne, uwierzytelnianie użytkowników czy formularze.
- **Skalowalność**: Django zostało zaprojektowane tak, aby można było tworzyć aplikacje skalowalne, od małych stron internetowych po duże platformy obsługujące miliony użytkowników.
- **Bezpieczeństwo**: Bezpieczeństwo jest jednym z priorytetów Django. Framework ten zawiera wiele zabezpieczeń przed powszechnymi atakami, takimi jak wstrzykiwanie SQL, cross-site scripting (XSS) czy fałszywe żądania między stronami (CSRF).
- **ORM (Object-Relational Mapping)**: Django posiada wbudowany system ORM, który pozwala na intuicyjne zarządzanie bazami danych przy użyciu obiektowego języka programowania, co upraszcza proces tworzenia i zarządzania bazami danych.
- **Wsparcie dla wielu baz danych**: Django wspiera wiele systemów baz danych, takich jak PostgreSQL, MySQL, SQLite czy Oracle.

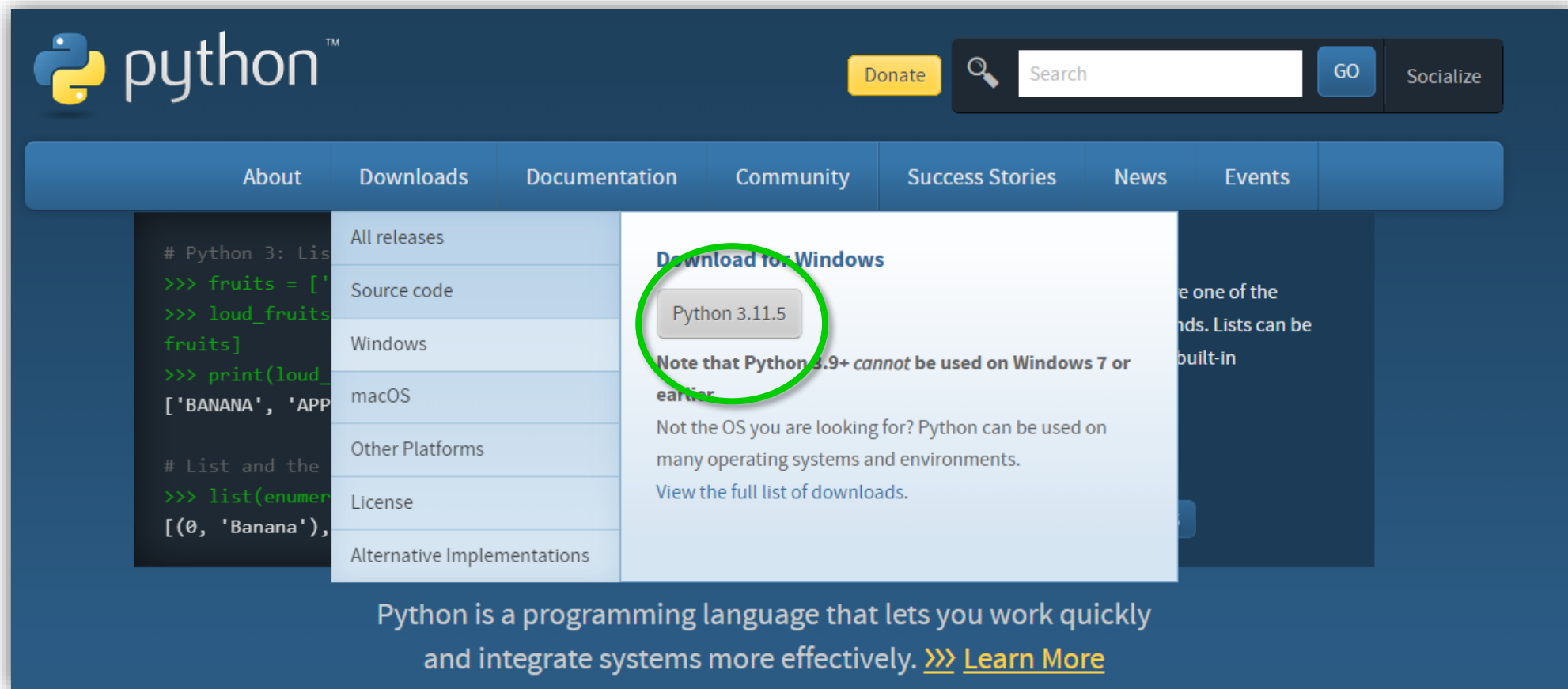
Dlaczego Python ? Dlaczego Django ?

Django to jeden z najbardziej popularnych frameworków do tworzenia aplikacji internetowych opartych na Pythonie. Oto kilka powodów, dla których warto nauczyć się Django:

- **Społeczność:** Django ma silną i aktywną społeczność, co oznacza dostęp do licznych zasobów, wtyczek, narzędzi oraz wsparcia od innych deweloperów.
- **Dokumentacja:** Django jest znane z doskonałej dokumentacji, co ułatwia naukę i rozwiązywanie problemów.
- **Modularność:** Django zachęca do modularnego i ponownego użycia kodu poprzez system aplikacji, co ułatwia zarządzanie i rozwijanie skomplikowanych projektów.
- **Wszechstronność:** Django może być używane do tworzenia różnych rodzajów aplikacji internetowych, od blogów i stron firmowych po sklepy internetowe czy platformy społecznościowe.
- **Zawodowe możliwości:** Ze względu na popularność i wszechstronność Django, umiejętność programowania w tym frameworku może otworzyć wiele drzwi w dziedzinie rozwoju aplikacji internetowych.

Instalowanie Pythona

<https://www.python.org/>



Instalowanie Pythona

Terminal - sprawdzanie wersji Pythona



```
Szymon@DESKTOP-3A679F2 MINGW64
$ python --version
Python 3.11.3
```

IDE-zintegrowane środowisko programistyczne (Integrated Development Environment)



IDE-zintegrowane środowisko programistyczne (Integrated Development Environment)



PyCharm Professional

The Python IDE for Professional Developers

Download

.exe ▼

Free 30-day trial



PyCharm Community Edition

The IDE for Pure Python Development

Download

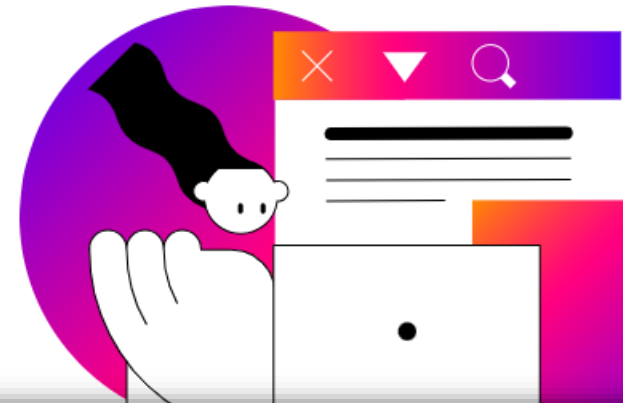
.exe ▼

Free, built on open source

IDE-zintegrowane środowisko programistyczne (Integrated Development Environment)

Free Educational Licenses

Learn or teach coding with best-in-class development tools from JetBrains!



Get free access to all developer tools from JetBrains!

[Apply now](#)

IDE-zintegrowane środowisko programistyczne (Integrated Development Environment)

Code editing. Redefined.

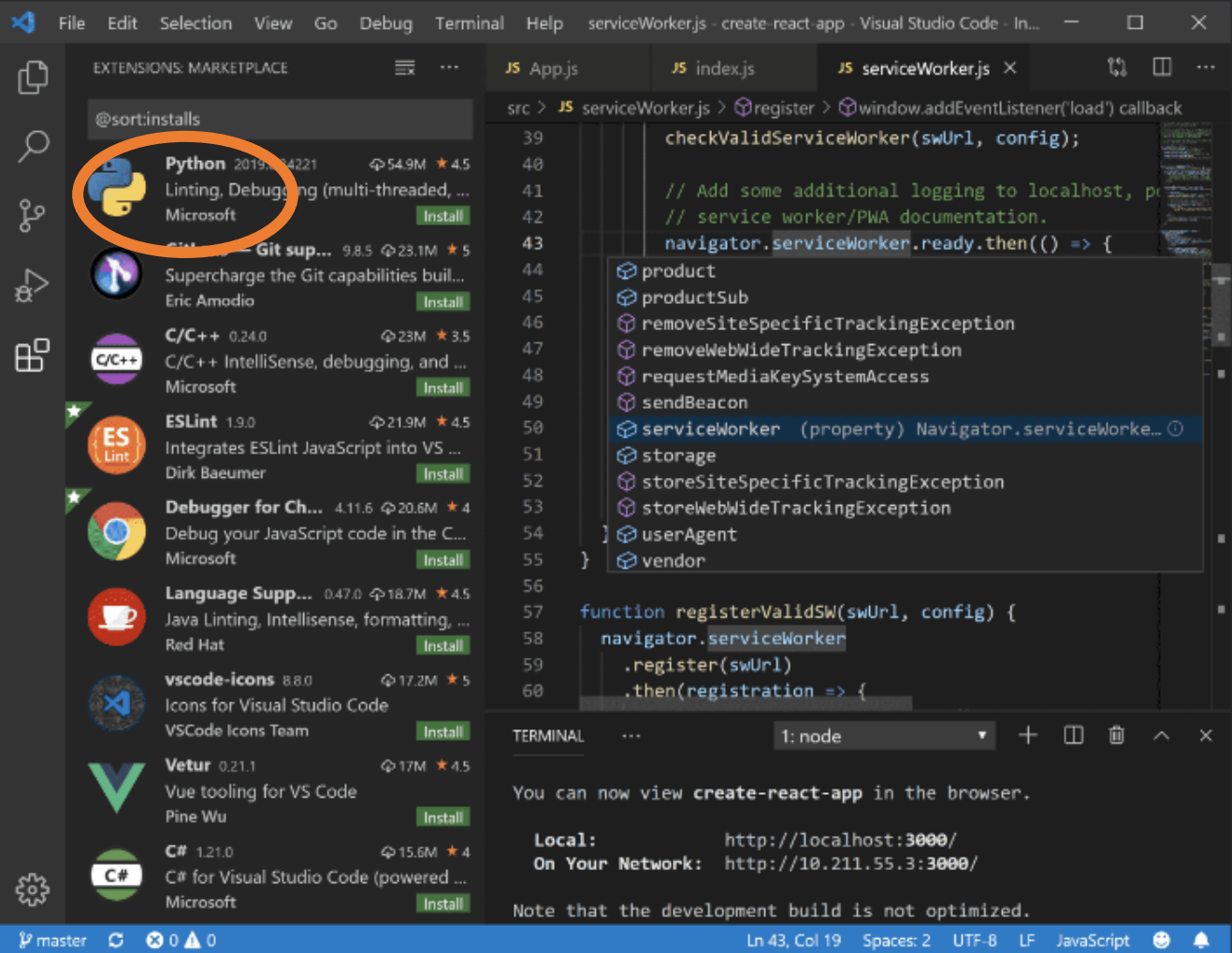
Free. Built on open source. Runs everywhere.

Download for Windows

Stable Build

Web, Insiders edition, or other platforms

By using VS Code, you agree to its
license and privacy statement.



The screenshot shows the Visual Studio Code interface. On the left, the 'EXTENSIONS: MARKETPLACE' panel is open, displaying a list of extensions. The 'Python' extension by Microsoft is highlighted with an orange circle. Other visible extensions include Git, C/C++, ESLint, Debugger for Chrome, Language Support for Java, vscode-icons, Vetur, and C#. The main editor area shows a JavaScript file named 'serviceWorker.js' with code for registering a service worker. A dropdown menu is visible over the code, showing various browser APIs like 'product', 'storage', and 'navigator'. The bottom status bar indicates the current file is 'Ln 43, Col 19' and the encoding is 'UTF-8'.

IDE-zintegrowane środowisko programistyczne (Integrated Development Environment)

Uwaga

Na zajęciach wykorzystywany będzie PyCharm Professional



PyCharm Professional

The Python IDE for Professional Developers

Virtual environment

<https://jaktestowac.pl/lesson/pt1-mk5-s01-l02/> [15.09.2023]

Python Virtual Environment (pot. venv – tego skrótu będziemy używać najczęściej) jest środowiskiem Pythona, które jest **odseparowane i całkowicie niezależne od głównej instalacji Pythona**. Każdy tworzony projekt powinien, zawierać swoje środowisko, dzięki czemu może składać się z unikalnego zestawu pakietów.



Virtual environment

<https://jaktestowac.pl/lesson/pt1-mk5-s01-l02/> [15.09.2023]

Czym jest Python Virtual Environment (venv)?

Python Virtual Environment (potocznie **venv** – tego skrótu będziemy używać najczęściej) jest środowiskiem Pythona, które jest **odseparowane** i **całkowicie niezależne** od głównej instalacji Pythona. Każdy tworzony przez nas projekt może, a nawet powinien, zawierać swoje środowisko, dzięki czemu może składać się z **unikalnego zestawu pakietów**.

Brzmi zawile? Bardzo upraszczając – **venv** jest **nowym katalogiem** na dysku, w którym znajduje się kopia Pythona. Zaraz po utworzeniu zawiera ona jedynie podstawowe pakiety. Zupełnie tak, jakbyśmy zainstalowali Pythona sami w nowej lokalizacji.

Różnica polega na tym, że **venv** tworzymy za pomocą jednego polecenia i oczywiście potrzebny jest już wcześniej zainstalowany Python. Nie zaśmiecamy sobie w żaden sposób systemu i możemy tworzyć dowolnie dużą liczbę takich środowisk.

W prosty sposób można aktywować **venv** (jak? – zrobimy to już w następnych lekcjach), doinstalować tylko te pakiety, których potrzebujemy, oraz uruchamiać stworzone przez nas skrypty.

Po co stosować venv? Jakie ma zalety?

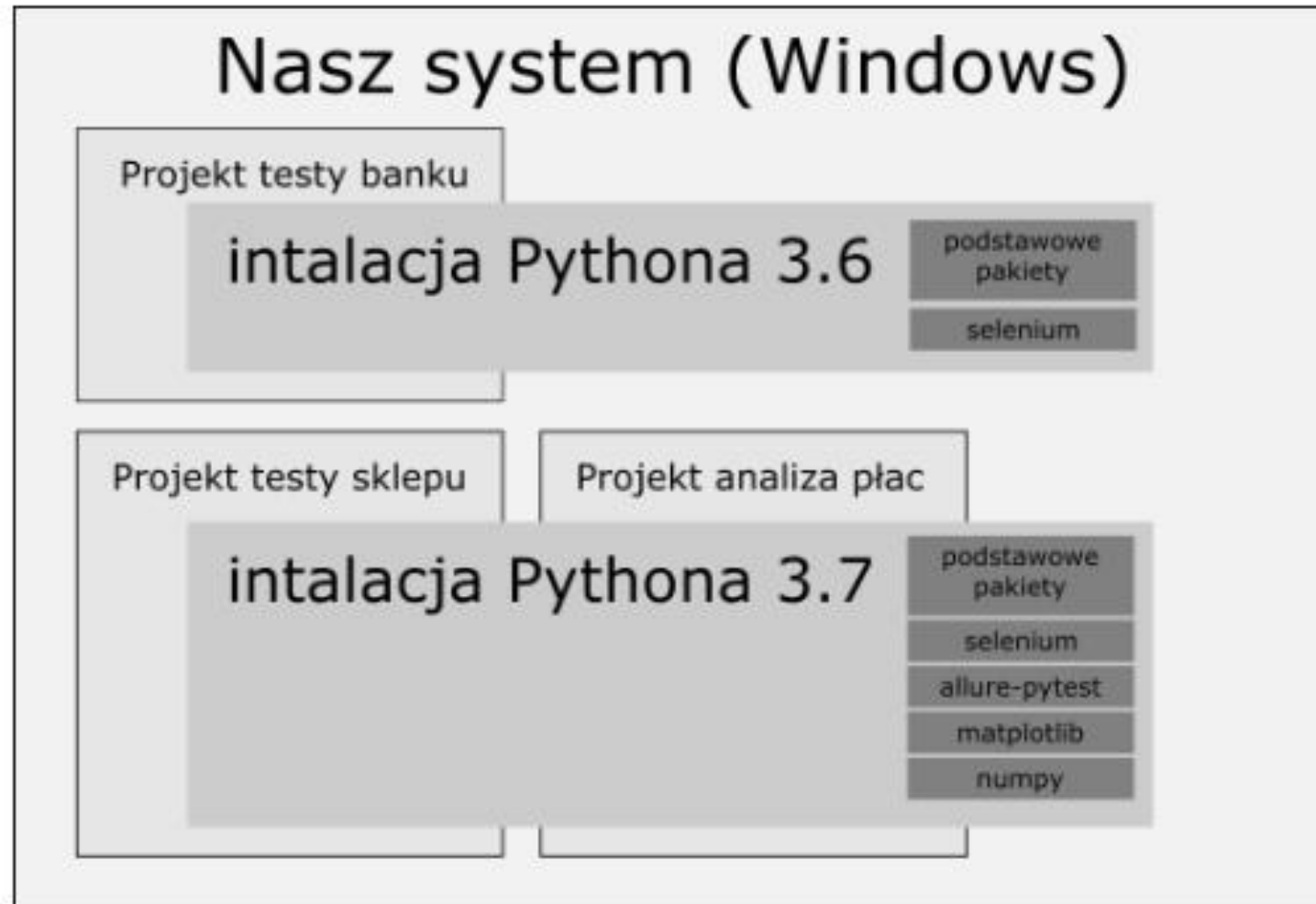
Największą i niewątpliwą zaletą **venv** jest **izolacja środowiska Python** dla danego projektu. Dzięki temu, każde stworzone przez nas środowisko, **może mieć własny zestaw pakietów**, który jest wymagany dla danego projektu.

Często dochodzi do sytuacji, gdy projekty nad którymi pracujemy wymagają różnych wersji pakietów, albo pakietów, które nie są kompatybilne ze sobą. **Venv** pozwala na pełną dowolność w komponowaniu zestawu potrzebnych bibliotek, dzięki czemu możemy mieć jednocześnie środowisko o nazwie **venv1** z pakietami w wersji 1.0 oraz **venv2** z pakietami w wersji 2.0 z możliwością szybkiego i bezproblemowego przelączania się między nimi.

Virtual environment

<https://jaktestowac.pl/lesson/pt1-mk5-s01-l02/> [15.09.2023]

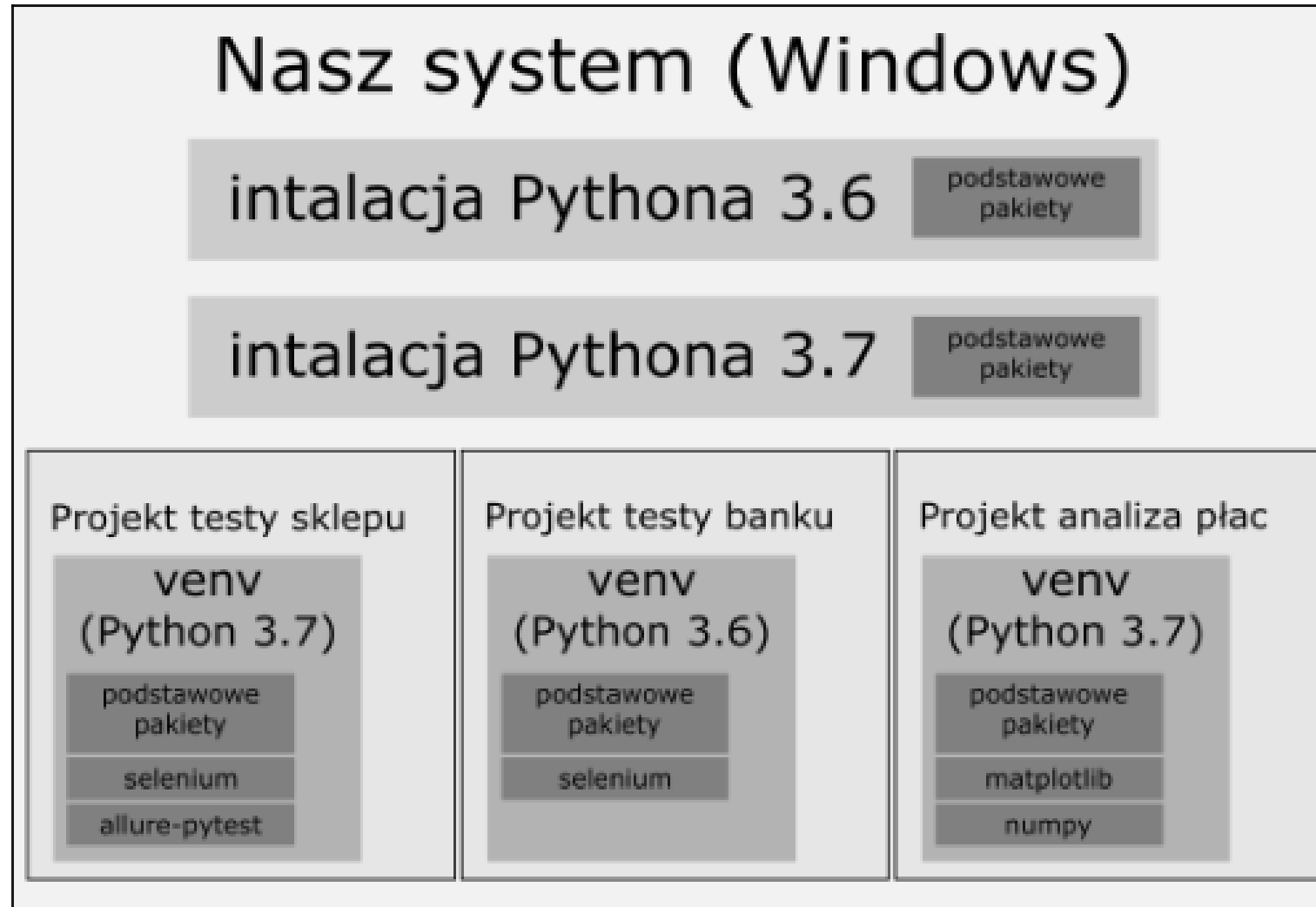
Zobacz sam jak może przykładowo wyglądać stan instalacji Pythona, pakietów i projektów bez zastosowania venv:



Virtual environment

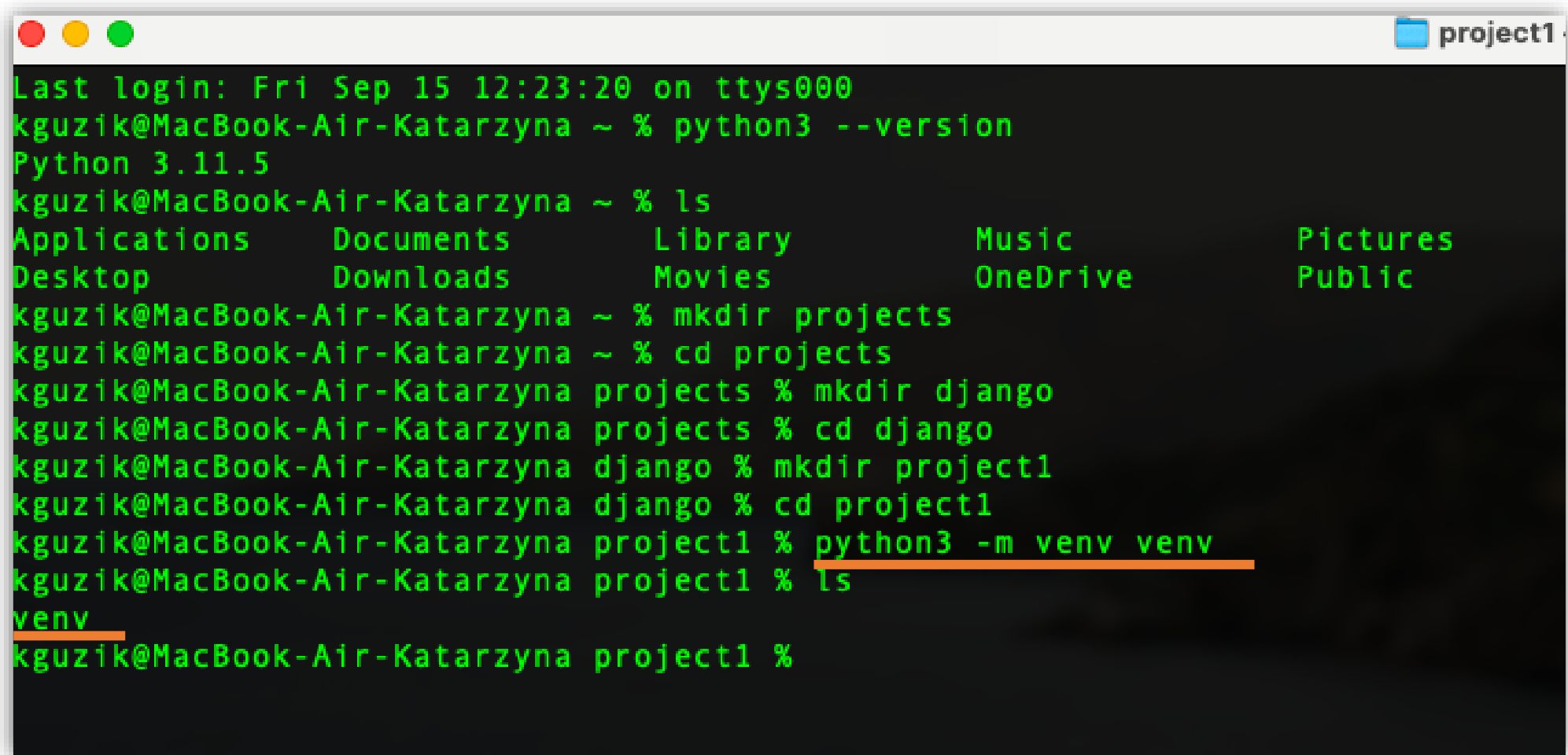
<https://jaktestowac.pl/lesson/pt1-mk5-s01-l02/> [15.09.2023]

I po zastosowaniu **venv**, czy dostrzegasz teraz zalety tego podejścia?:



Tworzenie katalogu projektu z wirtualnym środowiskiem (venv)

Tworzenie projektu pod Mac/Unix



```

Last login: Fri Sep 15 12:23:20 on ttys000
kguzik@MacBook-Air-Katarzyna ~ % python3 --version
Python 3.11.5
kguzik@MacBook-Air-Katarzyna ~ % ls
Applications      Documents          Library            Music              Pictures
Desktop           Downloads          Movies             OneDrive           Public
kguzik@MacBook-Air-Katarzyna ~ % mkdir projects
kguzik@MacBook-Air-Katarzyna ~ % cd projects
kguzik@MacBook-Air-Katarzyna projects % mkdir django
kguzik@MacBook-Air-Katarzyna projects % cd django
kguzik@MacBook-Air-Katarzyna django % mkdir project1
kguzik@MacBook-Air-Katarzyna django % cd project1
kguzik@MacBook-Air-Katarzyna project1 % python3 -m venv venv
kguzik@MacBook-Air-Katarzyna project1 % ls
venv
kguzik@MacBook-Air-Katarzyna project1 %

```

Tworzenie katalogu projektu z wirtualnym środowiskiem (venv)

Tworzenie projektu pod Mac/Unix

Uwaga

Trzeba aktywować zmienną środowiskową

```
[kguzik@MacBook-Air-Katarzyna project1 % source venv/bin/activate
(venv) kguzik@MacBook-Air-Katarzyna project1 %
```

Środowisko wirtualne, w którym projekt aktualnie „pracuje”

Uwaga

Dezaktywacja zmiennej środowiskowej

```
[(venv) kguzik@MacBook-Air-Katarzyna project1 % deactivate
kguzik@MacBook-Air-Katarzyna project1 % █
```

Tworzenie katalogu projektu z wirtualnym środowiskiem (venv)

Tworzenie projektu pod Windows

```
Szymon@DESKTOP-3A679F2 MINGW64 /e/projects/Django
```

```
$ python -m venv venv
```

```
Szymon@DESKTOP-3A679F2 MINGW64 /e/projects/Django
```

```
$ ls
```

```
venv
```

```
Szymon@DESKTOP-3A679F2 MINGW64 /e/projects/Django
```

```
$ |
```


Tworzenie katalogu projektu z wirtualnym środowiskiem (venv)

Tworzenie projektu pod Windows

Uwaga

Trzeba aktywować zmienną środowiskową

Środowisko wirtualne, w którym projekt aktualnie „pracuje”

```
Administrator: Windows PowerShell

PS E:\projects> cd .\Django\
PS E:\projects\Django> ls

Directory: E:\projects\Django

Mode                LastWriteTime         Length Name
----                -
d-----          15 wrz 2023     13:10         venv

PS E:\projects\Django> .\venv\Scripts\activate
(venv) PS E:\projects\Django>
```

Tworzenie katalogu projektu z wirtualnym środowiskiem (venv)

Tworzenie projektu pod Windows

Problem Windows

Brak możliwości uruchomienia zaznaczonego polecenia

Rozwiązanie

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine
```

Tworzenie katalogu projektu z wirtualnym środowiskiem (venv)

Tworzenie projektu pod Windows

Uwaga

Dezaktywacja zmiennej środowiskowej

```
(venv) PS E:\projects\Django> deactivate venv  
PS E:\projects\Django> _
```

Tworzenie projektu Django

```
pip install django
```

```
(venv) PS E:\projects\Django> pip install django
Collecting django
  Downloading Django-4.2.5-py3-none-any.whl (8.0 MB)
  ----- 8.0/8.0 MB 31.9 MB/s eta 0:00:00
Collecting asgiref<4,>=3.6.0
  Downloading asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting sqlparse>=0.3.1
  Downloading sqlparse-0.4.4-py3-none-any.whl (41 kB)
  ----- 41.2/41.2 kB ? eta 0:00:00
Collecting tzdata
  Downloading tzdata-2023.3-py2.py3-none-any.whl (341 kB)
  ----- 341.8/341.8 kB 20.7 MB/s eta 0:00:00
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.7.2 django-4.2.5 sqlparse-0.4.4 tzdata-2023.3

[notice] A new release of pip available: 22.3.1 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS E:\projects\Django>
```

Tworzenie projektu Django

```
django-admin startproject {nazwa} {lokalizacja}
```

```
(venv) PS E:\projects\Django> django-admin startproject crud_blog .  
(venv) PS E:\projects\Django> ls
```

Directory: E:\projects\Django

Mode	LastWriteTime	Length	Name
d-----	15 wrz 2023 14:06		crud_blog
d-----	15 wrz 2023 13:10		venv
-a-----	15 wrz 2023 14:06	687	manage.py

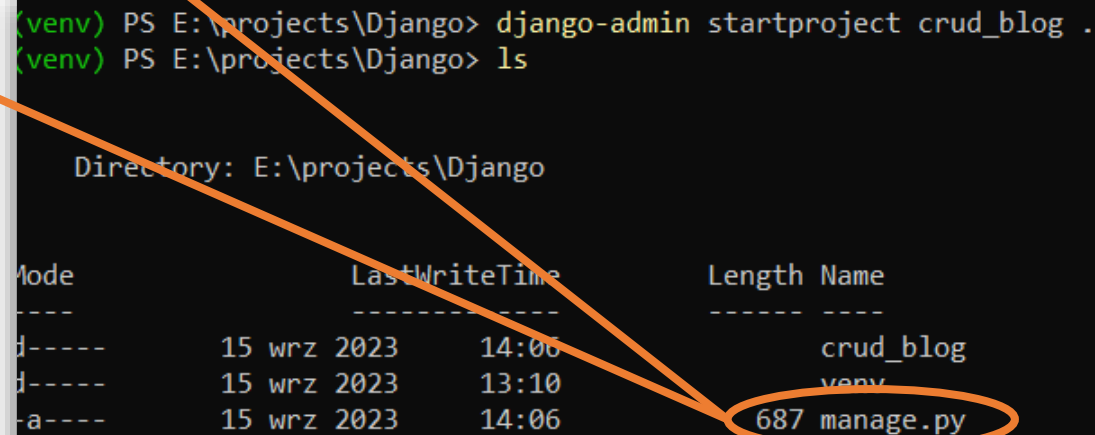
Uruchamianie lokalnego serwera

Linux/Mac

```
python3 manage.py runserver
```

Windows

```
python manage.py runserver
```



```
(venv) PS E:\projects\Django> django-admin startproject crud_blog .
(venv) PS E:\projects\Django> ls
```

Directory: E:\projects\Django

Mode	LastWriteTime	Length	Name
d-----	15 wrz 2023 14:06		crud_blog
d-----	15 wrz 2023 13:10		venv
-a----	15 wrz 2023 14:06	687	manage.py

Arrows from the code blocks point to the 'manage.py' file in the terminal output.

Uruchamianie lokalnego serwera

Linux/Mac/Windows

```
(venv) PS E:\projects\Django> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

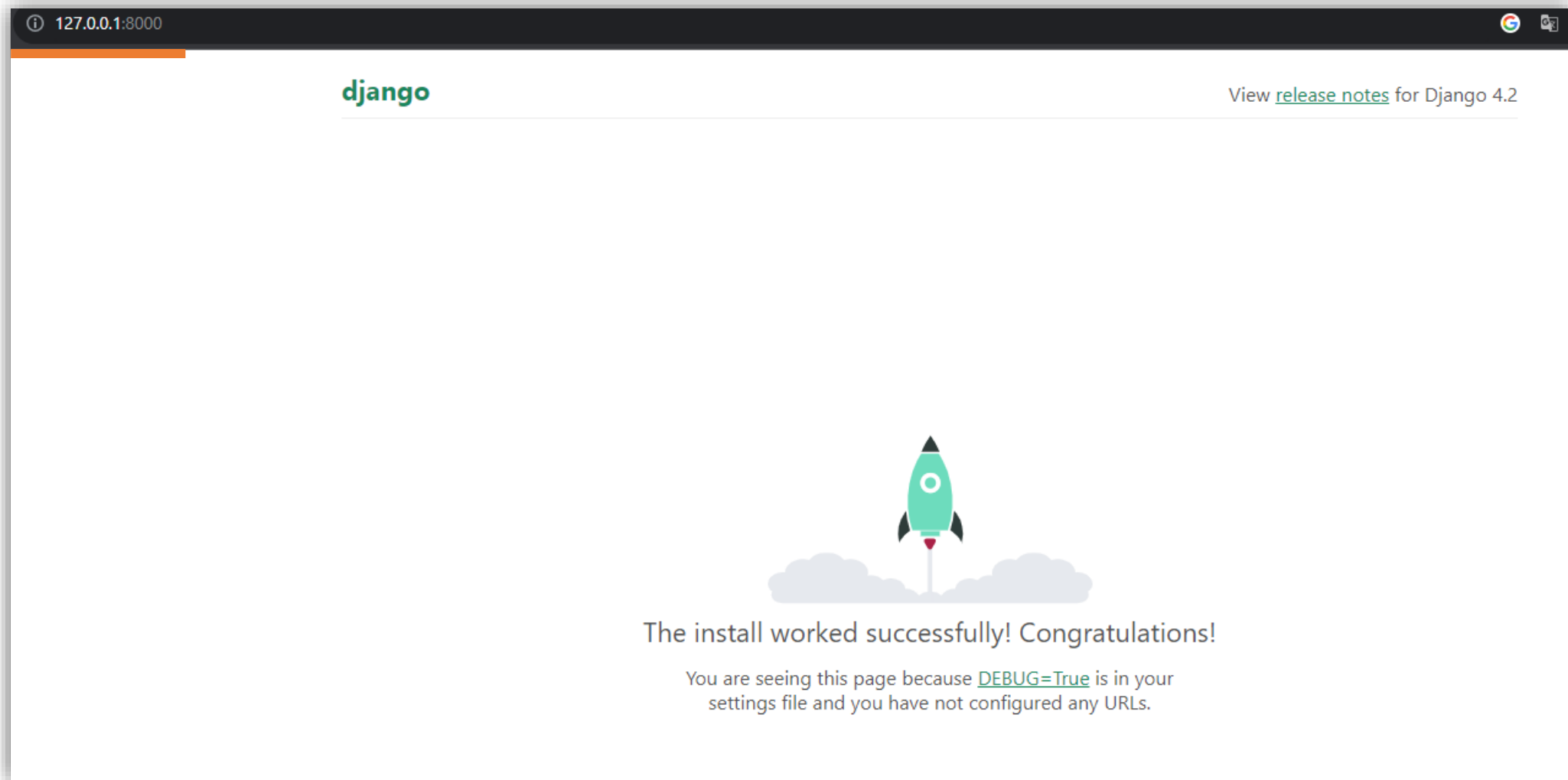
System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until
you have applied them.
Run 'python manage.py migrate' to apply them.
September 15, 2023 - 14:19:23
Django version 4.2.5, using settings 'crud_blog.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[15/Sep/2023 14:19:38] "GET / HTTP/1.1" 200 10664
Not Found: /favicon.ico
[15/Sep/2023 14:19:38] "GET /favicon.ico HTTP/1.1" 404 2113
```

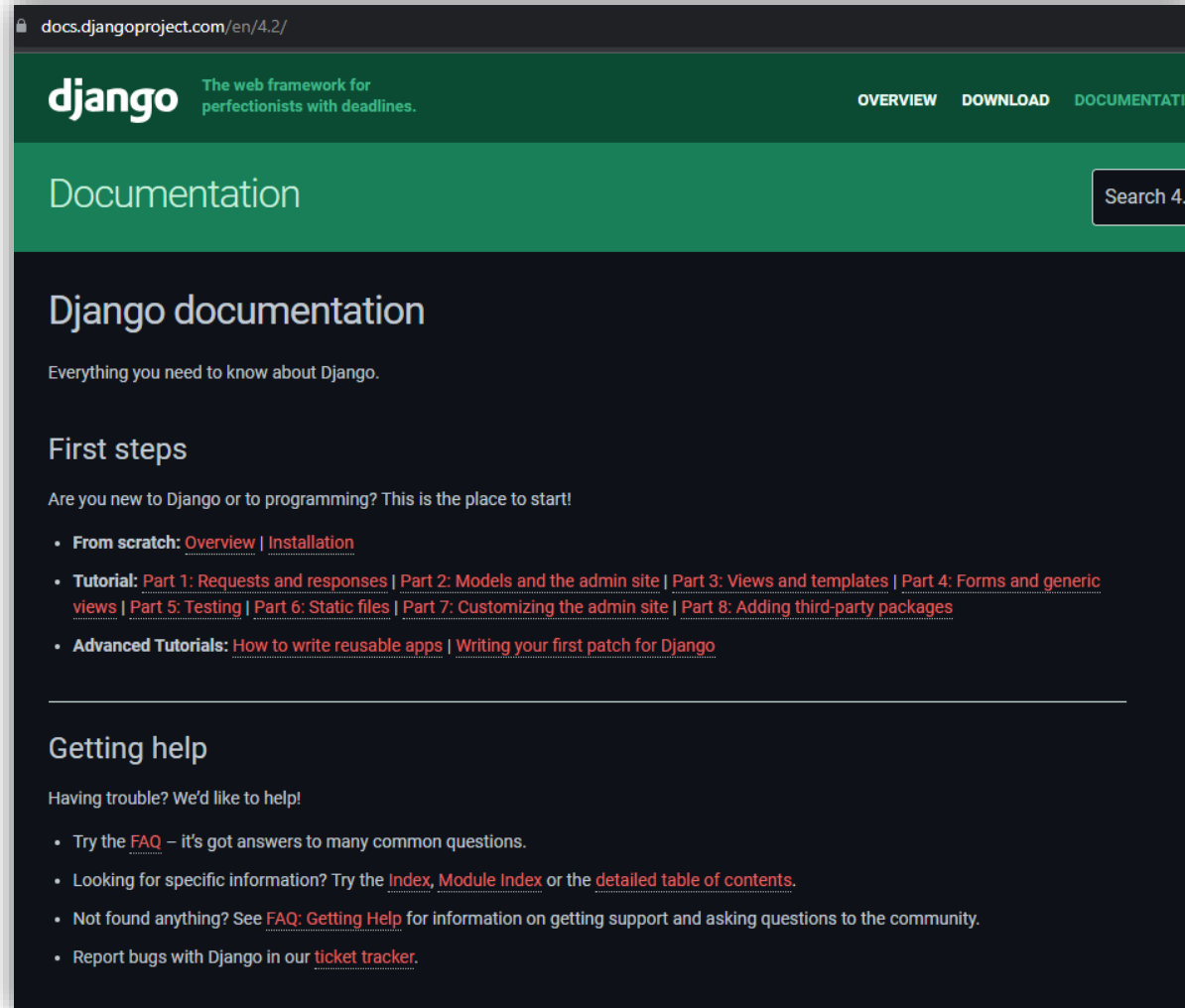
Uruchamianie lokalnego serwera

Linux/Mac/Windows



Dokumentacja Django

`https://docs.djangoproject.com/en/4.2/`



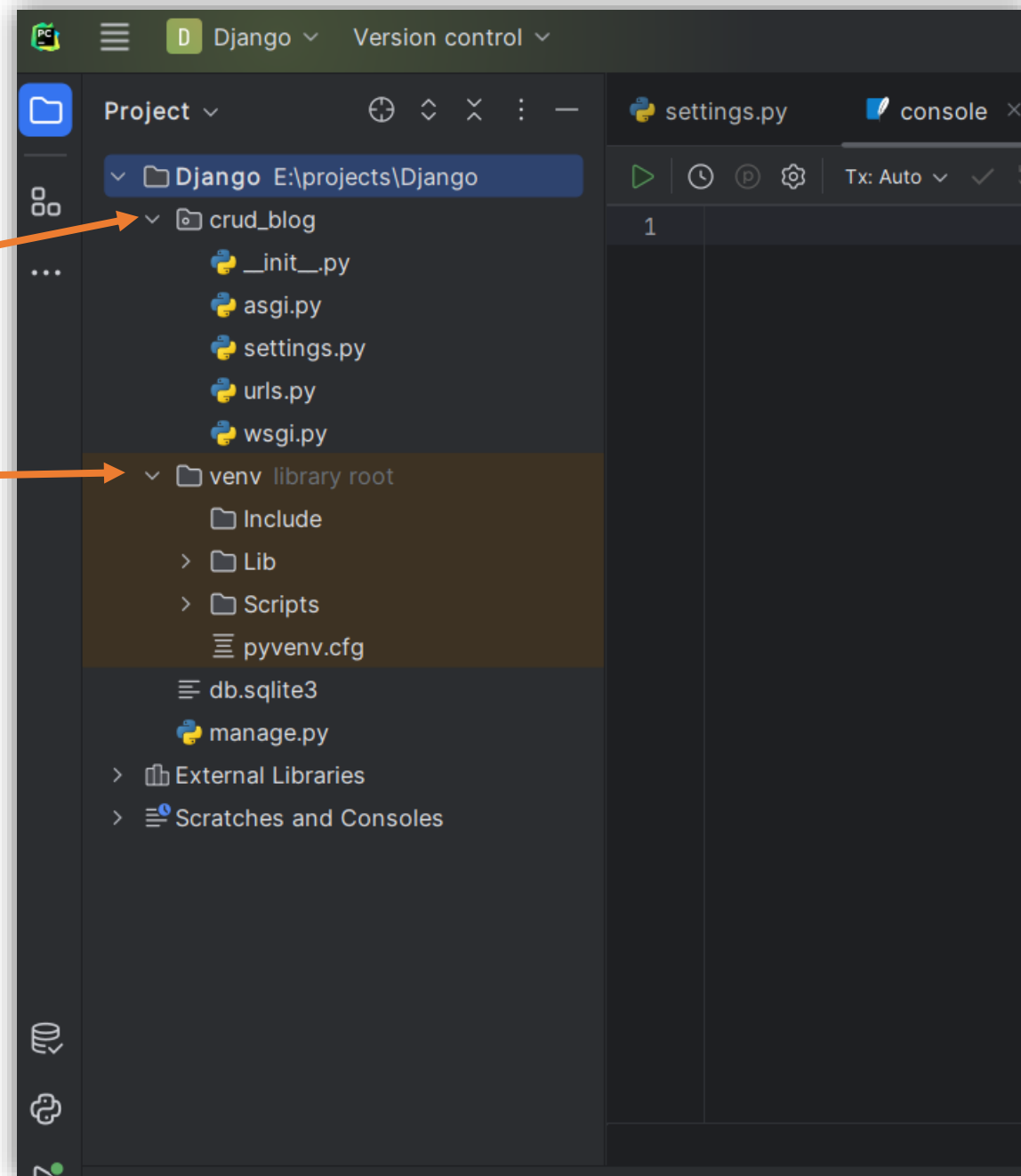
Omówienie plików

Uwaga

Projekt należy otworzyć w IDE

Aplikacja

Zmienna środowiskowa



Tworzenie nowej aplikacji

```
django-admin startapp {nazwa_aplikacji}
```

```
(venv) PS E:\projects\Django> django-admin startapp crud_blog_web  
(venv) PS E:\projects\Django> 
```

Aplikacje domyślne oraz „dedykowana”

The image shows a code editor interface with a project explorer on the left and a code editor on the right. The project explorer shows a Django project structure with folders like 'crud_blog' and 'crud_blog_web', and files like 'settings.py', 'urls.py', and 'wsgi.py'. The code editor displays the 'settings.py' file, which contains configuration for Django, including security warnings, debug mode, allowed hosts, and a list of installed applications.

Project Explorer (Left):

- Project Django E:\projects\Django
 - crud_blog
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py
 - crud_blog_web
 - venv library root
 - db.sqlite3
 - manage.py
 - External Libraries
 - Scratches and Consoles

Code Editor (Right): settings.py

```

25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'crud_blog_web',
41 ]

```

Annotations:

- A white bracket on the right side of the code editor groups the default Django applications (lines 34-39) and is labeled "Aplikacje domyślne".
- An orange bracket on the right side of the code editor groups the custom application 'crud_blog_web' (line 40) and is labeled "Aplikacje dedykowana".
- An orange arrow points from the 'crud_blog_web' folder in the project explorer to the 'crud_blog_web' entry in the 'INSTALLED_APPS' list.

Migracje

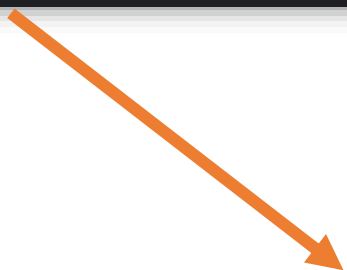
Migracje w Django to mechanizm, który pozwala na wprowadzanie zmian w schemacie bazy danych (tzn. strukturze tabel, indeksach, relacjach itp.) w sposób kontrolowany i wersjonowany. Dzięki temu, kiedy wprowadzasz zmiany w modelach aplikacji Django, nie musisz ręcznie modyfikować bazy danych ani pisać surowych skryptów SQL do dokonywania tych zmian.



Migracje

Po uruchomieniu projektu w terminalu widoczna jest informacja dot. migracji

```
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.  
Run 'python manage.py migrate' to apply them.
```



```
You have 18 unapplied migration(s).  
Run 'python manage.py migrate' to a  
September 15, 2023 - 15:33:00
```

Migracje

Wykonywanie migracji

```
(venv) PS E:\projects\Django> python manage.py migrate
```

Operations to perform:

Apply all migrations: admin, auth, contenttypes, sessions

Running migrations:

Applying contenttypes.0001_initial... OK

Applying auth.0004_alter_user_username_opts... OK

Applying auth.0005_alter_user_last_login_null... OK

Applying auth.0006_require_contenttypes_0002... OK

Applying auth.0007_alter_validators_add_error_messages... OK

Applying auth.0008_alter_user_username_max_length... OK

Applying auth.0009_alter_user_last_name_max_length... OK

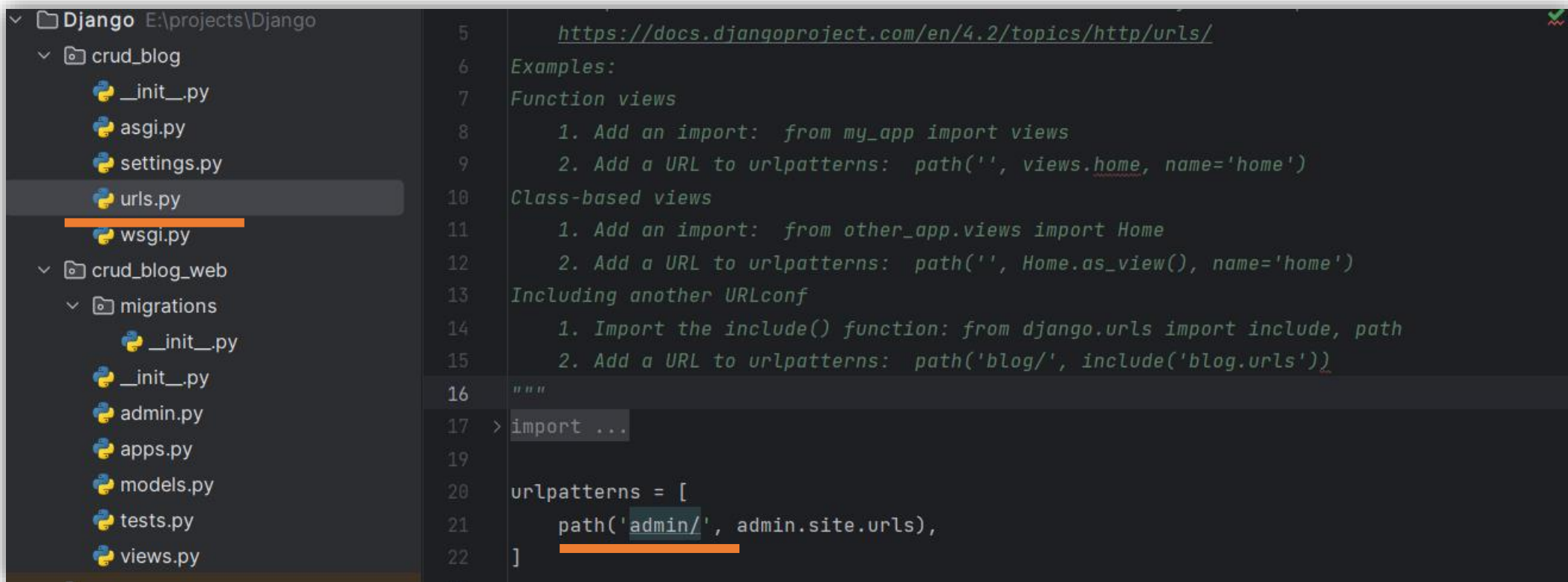
Applying auth.0010_alter_group_name_max_length... OK

Applying auth.0011_update_proxy_permissions... OK

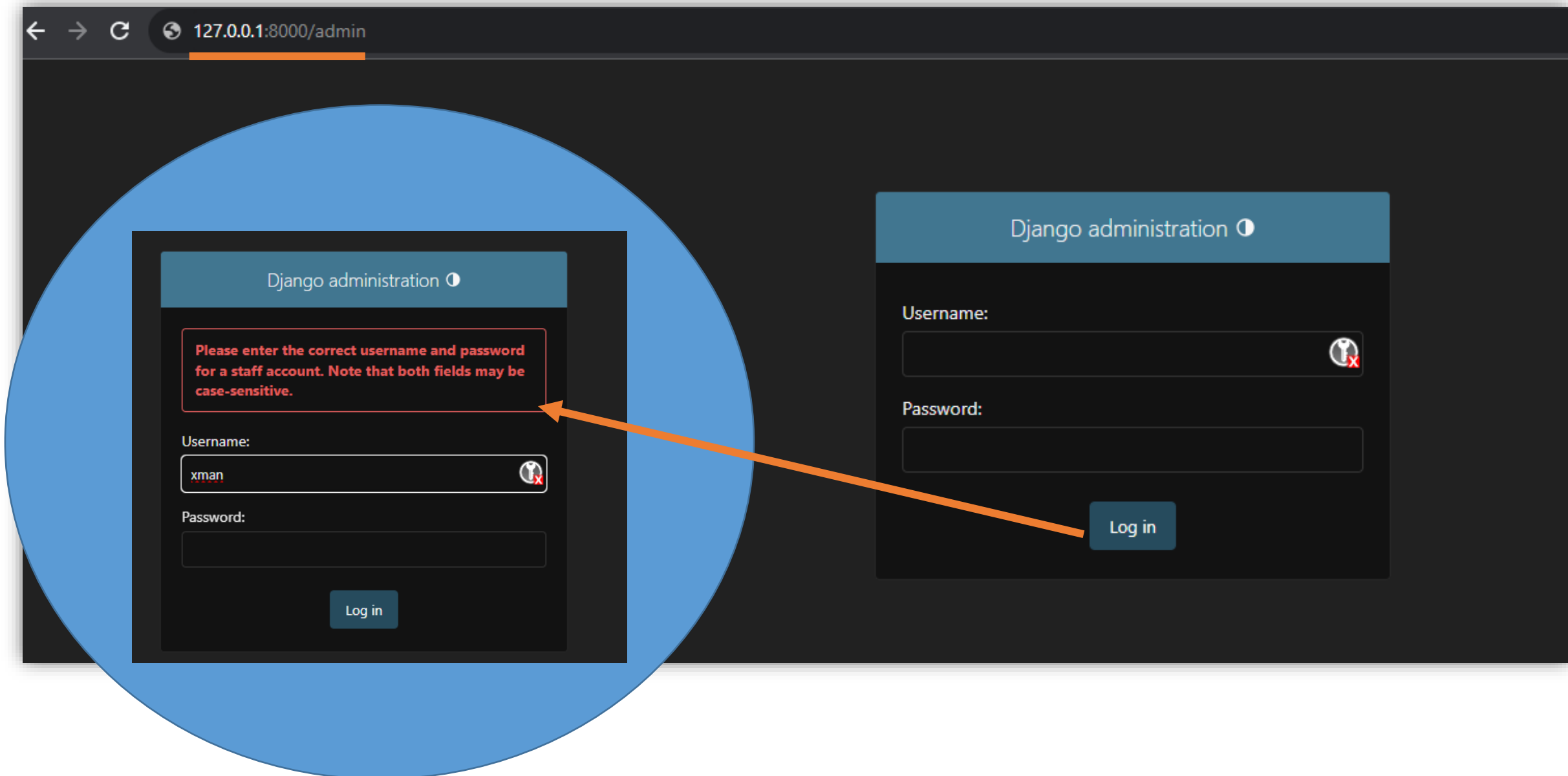
Applying auth.0012_alter_user_first_name_max_length... OK

Applying sessions.0001_initial... OK

Tworzenie użytkownika superuser oraz logowanie do admina



Tworzenie użytkownika superuser oraz logowanie do admina



Tworzenie użytkownika superuser oraz logowanie do admina

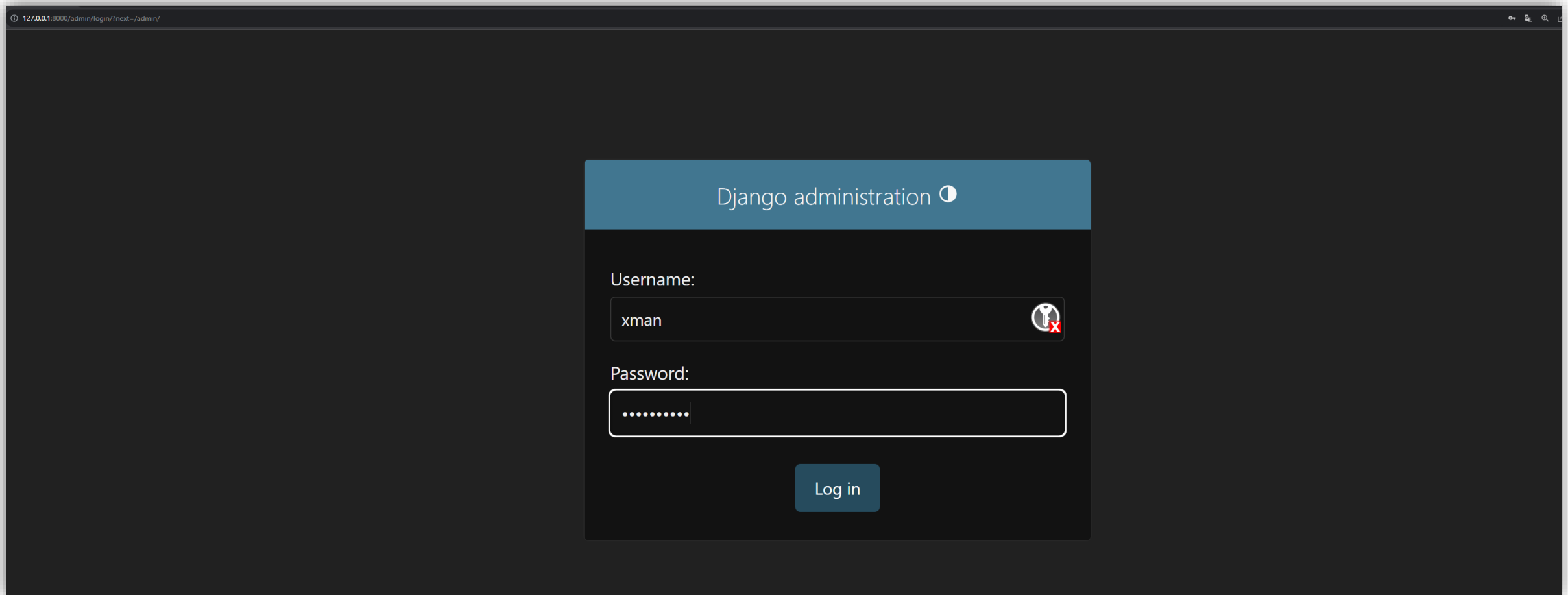
```
(venv) PS E:\projects\Django> python manage.py createsuperuser
Username (leave blank to use 'szymon'): xman
Email address: szymon.guzik@gdansk.merito.pl
Password:
Password (again):
Superuser created successfully.
```

Username: xman

Email: szymon.guzik@gdansk.merito.pl

Password: qwerty2023

Tworzenie użytkownika superuser oraz logowanie do admina



127.0.0.1:8000/admin/login/?next=/admin/

Django administration

Username:

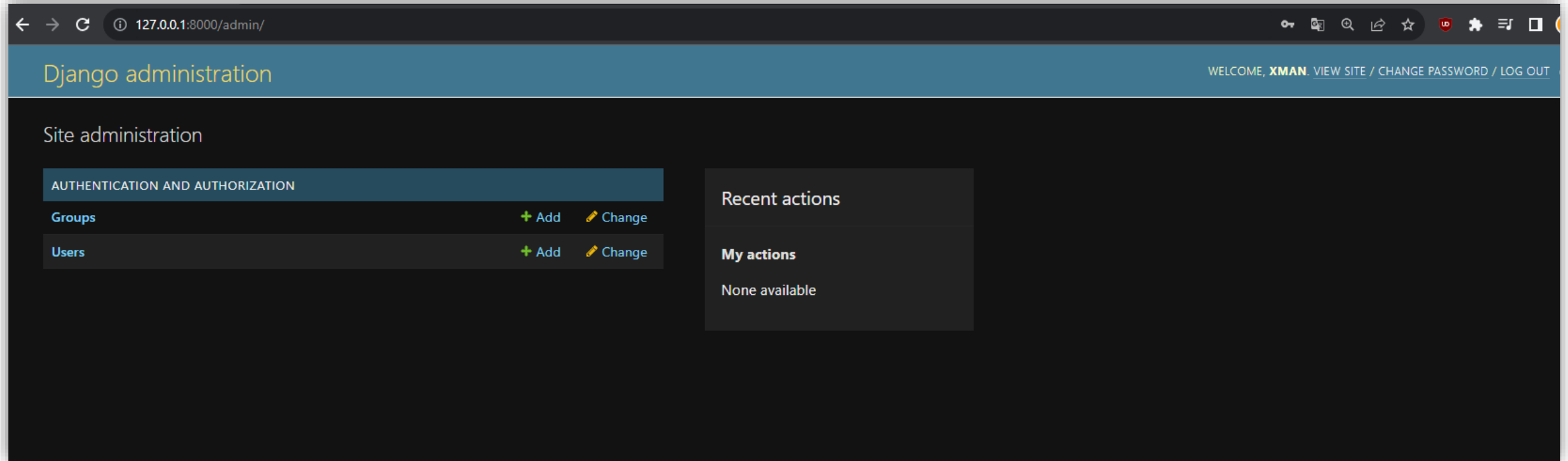
xman

Password:

.....

Log in

Tworzenie użytkownika superuser oraz logowanie do admina

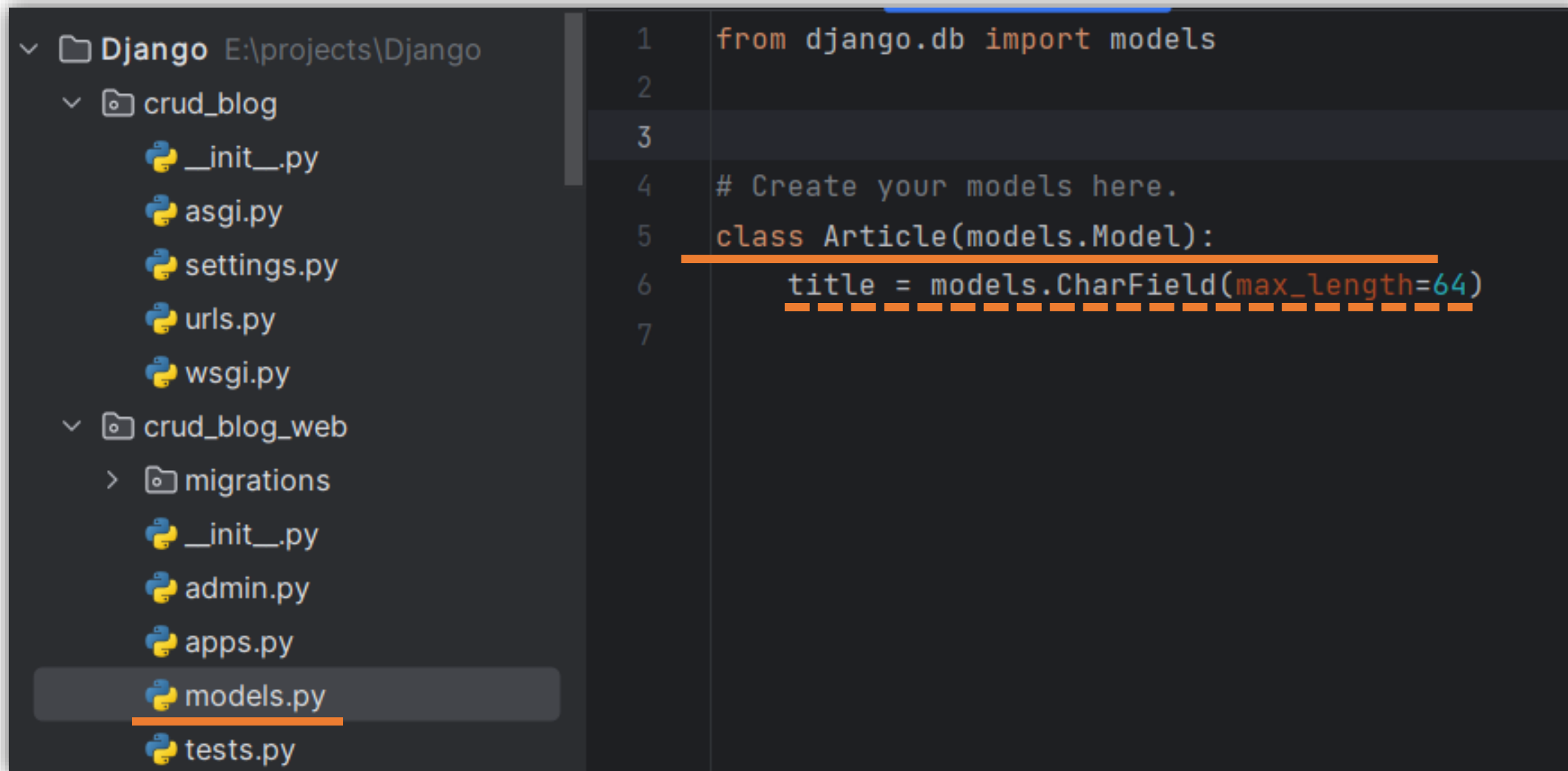


Modele, migracje, rejestracja

W frameworku Django, modele są kluczowym elementem, który pozwala na **definiowanie struktury bazy danych oraz zachowania związane z danymi**. Modele w Django to w zasadzie klasy Pythona, które definiują i reprezentują tabele w bazie danych. Modele dostarczają również metody i narzędzia do tworzenia, odpytywania, modyfikowania i usuwania rekordów w bazie danych



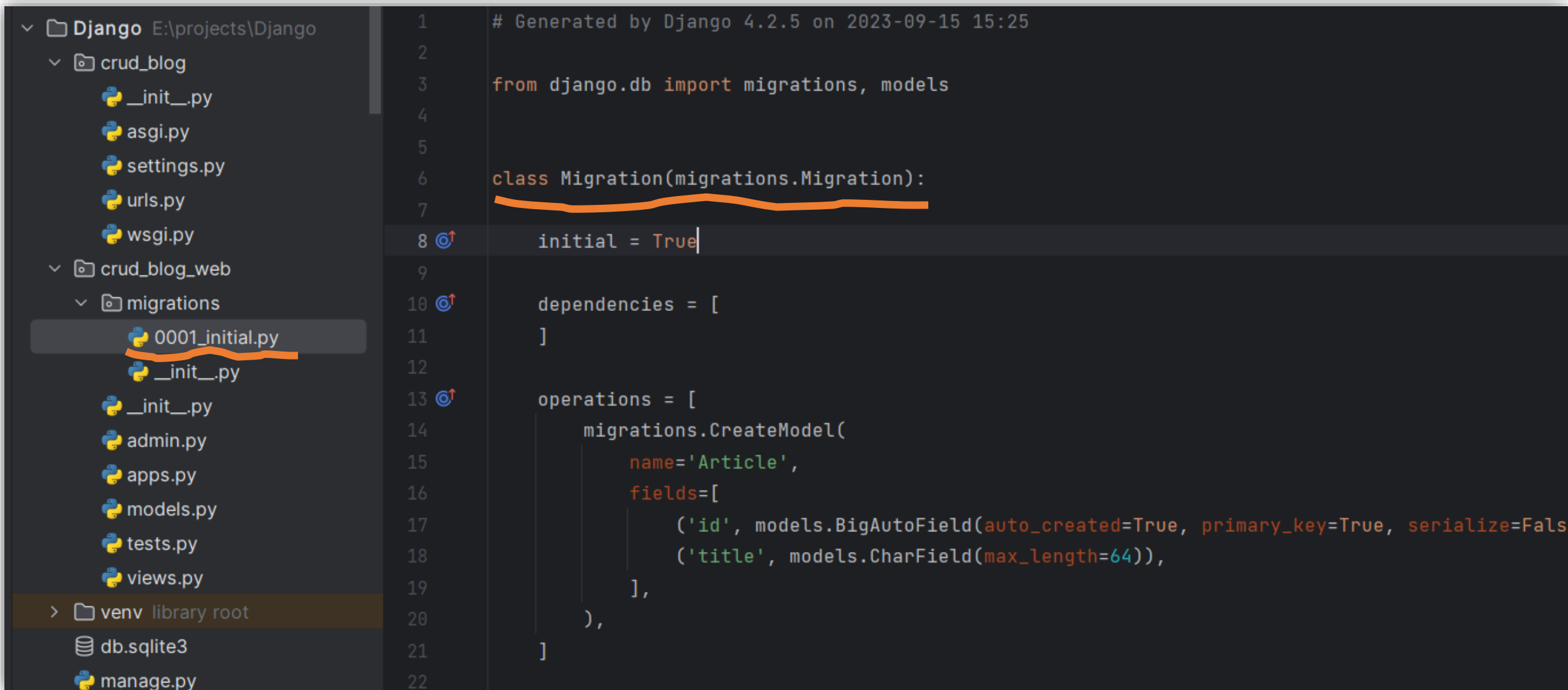
Modele, migracje, rejestracja



Modele, migracje, rejestracja

```
(venv) PS E:\projects\Django> python .\manage.py makemigrations
Migrations for 'crud_blog_web':
  crud_blog_web\migrations\0001_initial.py
    - Create model Article
```

Modele, migracje, rejestracja



Modele, migracje, rejestracja

Uwaga

W celu wykonania migracji należy
ponownie użyć komendy

```
(venv) PS E:\projects\Django> python manage.py migrate
```

```
Migrations for 'crud_blog_web':
```

```
  crud_blog_web\migrations\0001_initial.py
```

```
    - Create model Article
```

```
(venv) PS E:\projects\Django> python manage.py migrate
```

```
Operations to perform:
```

```
  Apply all migrations: admin, auth, contenttypes, crud_blog_web, sessions
```

```
Running migrations:
```

```
  Applying crud_blog_web.0001_initial... OK
```

Modele, migracje, rejestracja

Uwaga

Rejestracja modelu pozwala między innymi na zarządzanie tabelami z poziomu panelu admina

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders 'project-blog' and 'crud_blog_web'. The 'crud_blog_web' folder contains 'migrations', 'admin.py', 'apps.py', 'models.py', 'tests.py', and 'views.py'. The 'admin.py' file is selected and highlighted with an orange line. The code editor shows the content of 'admin.py' with the following code:

```

1 from django.contrib import admin
2 from .models import Article
3 # Register your models here.
4 admin.site.register(Article)

```

The code is highlighted with an orange line. The file explorer also shows a 'venv' folder at the bottom, which is the virtual environment root.

Modele, migracje, rejestracja

Uwaga

Rejestracja modelu pozwala między innymi na zarządzanie tabelami z poziomu panelu admina

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

CRUD_BLOG_WEB

Articles + Add

Change user

xman

Username:

xman

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:


algorithm: pbkdf2_sha256 iterations: 600000 salt: anU4Lk***** hash: +DrcaW*****

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using [this form](#).


Personal info

Model fields


<https://docs.djangoproject.com/en/4.2/ref/models/fields/>



The web framework for perfectionists with deadlines.


[OVERVIEW](#)
[DOWNLOAD](#)
[DOCUMENTATION](#)
[NEWS](#)
[COMMUNITY](#)
[CODE](#)
[ISSUES](#)
[ABOUT](#)
[♥ DONATE](#)


Documentation

Search 4.2 documentation (Ctrl + K) 

Model field reference


This document contains all the API references of **Field** including the [field options](#) and [field types](#) Django offers.



See also

If the built-in fields don't do the trick, you can try [django-localflavor \(documentation\)](#), which contains assorted pieces of code that are useful for particular countries and cultures.

Also, you can easily [write your own custom model fields](#).




Note

Technically, these models are defined in `django.db.models.fields`, but for convenience they're imported into `django.db.models`; the standard convention is to use `from django.db import models` and refer to fields as `models.<Foo>Field`.

Field options

The following arguments are available to all field types. All are optional.

Support Django!



Martin De Wulf donated to the Django Software Foundation to support Django development. Donate today!

Contents

- [Model field reference](#)
 - [Field options](#)
 - `null`
 - `blank`
 - `choices`
 - [Enumeration types](#)
 - `db_column`
 - `db_comment`
 - `db_index`

Model fields

Aktualizacja modelu

```
from django.db import models

# Create your models here.
# 2 usages
class Article(models.Model):
    # pole 150 znaków
    # pole nie może być puste
    # Mogę stworzyć inny artykuł o tym samym tytule
    title = models.CharField(max_length=150, blank=False, unique=False)
    # Jeżeli nikt nie poda roku powstania artykułu to pole zostanie uzupełnione rokiem 2023
    year = models.PositiveSmallIntegerField(default=2023)
```

```
(venv) PS E:\projects\Django> python .\manage.py makemigrations
```

```
(venv) PS E:\projects\Django> python manage.py migrate
```

Model fields

Aktualizacja modelu

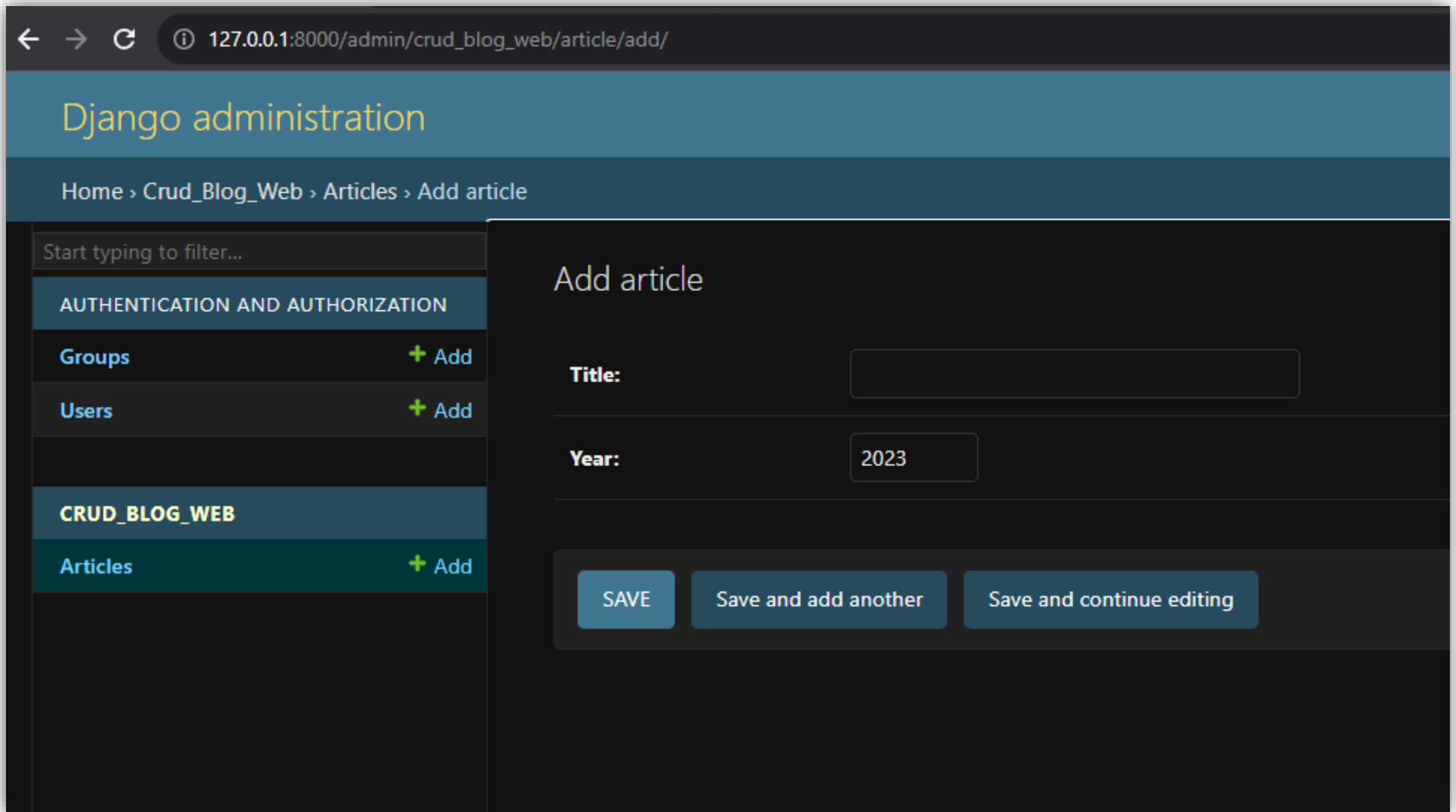
```
from django.db import models

# Create your models here.
# 2 usages
class Article(models.Model):
    # pole 150 znaków
    # pole nie może być puste
    # Mogę stworzyć inny artykuł o tym samym tytule
    title = models.CharField(max_length=150, blank=False, unique=False)
    # Jeżeli nikt nie poda roku powstania artykułu to pole zostanie uzupełnione rokiem 2023
    year = models.PositiveSmallIntegerField(default=2023)
```

```
(venv) PS E:\projects\Django> python .\manage.py makemigrations
```

```
(venv) PS E:\projects\Django> python manage.py migrate
```

Model fields



The screenshot displays the Django administration interface for adding a new article. The browser address bar shows the URL `127.0.0.1:8000/admin/crud_blog_web/article/add/`. The page title is "Django administration". The breadcrumb navigation path is "Home > Crud_Blog_Web > Articles > Add article".

On the left sidebar, the "Articles" link under the "CRUD_BLOG_WEB" section is highlighted. The main content area is titled "Add article" and contains the following form fields:

- Title:** A text input field.
- Year:** A text input field containing the value "2023".

At the bottom of the form, there are three buttons: "SAVE", "Save and add another", and "Save and continue editing".

Rzutowanie pobieranych danych

```

1  from django.db import models
2
3
4  # Create your models here.
5  2 usages
6  class Article(models.Model):
7      # pole 150 znaków
8      # pole nie może być puste
9      # Mogę stworzyć inny artykuł o tym samym tytule
10     title = models.CharField(max_length=150, blank=False, unique=False)
11     # Jeżeli nikt nie poda roku powstania artykułu to pole zostanie uzupełnione rokiem 2023
12     year = models.PositiveSmallIntegerField(default=2023)
13
14     def __str__(self):
15         return self.title

```


Rzutowanie pobieranych danych

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

CRUD_BLOG_WEB

Articles + Add

Select article to change

Action: [-----] Go 0 of 1 selected

☐ ARTICLE

☐ Article object (1)

1 article

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

CRUD_BLOG_WEB

Articles + Add

Select article to change

Action: [-----] Go 0 of 1 selected

☐ ARTICLE

☐ Uniwersytet Merito

1 article

Praca samodzielna studenta na laboratorium oraz w wolnym czasie

Proszę odszukać narzędzie pozwalające na połączenie się z bazą danych typu SQLite lub skonfigurować połączenie w wybranym IDE

Proszę zapoznać się z dokumentacją Django (zadanie do domu)

<https://docs.djangoproject.com/>

Zadanie 1. – czas wykonania 1h

Należy stworzyć projekt w Django w wybranej tematyce.

Jeżeli nie masz pomysłu na temat to Twoim zadaniem jest stworzenie portalu (do wyboru)

- Sprzedaż nieruchomości
- Salon samochodowy
- Portal z wycieczkami (podobne do booking.com)
- Portal gastronomiczny (podobne do pyszne.pl)
- Portal podobny do allegro

Proszę przemyśleć jakie dane (modele, tabele, kolumny) będą potrzebne do realizacji takiego zadania.

Następnie należy stworzyć modele, wykonać migracje, zarejestrować aplikacje oraz przetestować obsługę po stronie panelu admina. W tym zadaniu nie ma znaczenia na obecną chwilę tworzenie powiązań pomiędzy tabelami w bazie danych.

Zadanie należy spakować w całości i umieścić na platformie moodle.

Oceniana jest jakość wykonania zadania.

Za to zadanie można uzyskać maksymalnie 100 pkt.

Pomagam w zadaniu o ile nie wybiega to z tematyką obecnych laboratoriów