

Spis treści

O autorze	17
O recenzentach	18
Słowo wstępne	19
Przedmowa	21
ROZDZIAŁ 1.	
Utworzenie aplikacji bloga	27
Instalacja Pythona	28
Tworzenie środowiska wirtualnego w Pythonie	29
Instalacja Django	30
Instalowanie Django za pomocą pip	30
Nowe funkcjonalności w Django 4	31
Ogólne informacje na temat frameworka Django	32
Główne komponenty frameworka	32
Architektura Django	33
Tworzenie pierwszego projektu	34
Stosowanie początkowych migracji bazy danych	35
Uruchamianie serwera programistycznego	36
Ustawienia projektu	38
Projekty i aplikacje	39
Utworzenie aplikacji	40
Tworzenie modeli danych dla bloga	41
Utworzenie modelu Post	41
Dodawanie pól daty i godziny	43
Definiowanie domyślnej kolejności sortowania	44
Dodawanie indeksu bazy danych	45
Aktywacja aplikacji	46
Dodawanie pola stanu	46
Dodanie relacji wiele do jednego	48
Tworzenie i stosowanie migracji	50

Tworzenie witryny administracyjnej dla modeli	52
Tworzenie superużytkownika	53
Witryna administracyjna Django	53
Dodawanie modeli do witryny administracyjnej	54
Personalizacja sposobu wyświetlania modeli	56
Praca z obiektami QuerySet i menedżerami	58
Tworzenie obiektów	59
Aktualizowanie obiektów	60
Pobieranie obiektów	61
Usuwanie obiektów	62
Kiedy następuje określenie zawartości kolekcji QuerySet?	62
Utworzenie menedżerów modelu	63
Przygotowanie widoków listy i szczegółów	64
Utworzenie widoków listy i szczegółów	65
Korzystanie ze skrótu get_object_or_404	66
Dodanie wzorców adresów URL do widoków	67
Utworzenie szablonów dla widoków	68
Tworzenie szablonu bazowego	69
Utworzenie szablonu listy postów	70
Uruchomienie aplikacji	71
Tworzenie szablonu szczegółów posta	71
Cykl żądanie-odpowiedź	72
Zasoby dodatkowe	74
Podsumowanie	75
ROZDZIAŁ 2.	
Usprawnienie bloga za pomocą funkcji zaawansowanych	76
Kanoniczne adresy URL dla modeli	77
Tworzenie dla postów adresów URL przyjaznych dla SEO	79
Modyfikowanie wzorców adresów URL	81
Modyfikowanie widoków	82
Modyfikowanie kanonicznego adresu URL dla postów	82

Dodanie stronicowania	83
Dodanie stronicowania do widoku listy postów	84
Tworzenie szablonu stronicowania	84
Obsługa błędów stronicowania	87
Tworzenie widoków opartych na klasach	90
Po co korzystać z widoków opartych na klasach?	90
Użycie widoku opartego na klasie do wyświetlania listy postów	91
Polecenie postów przez e-mail	93
Tworzenie formularzy w Django	94
Obsługa formularzy w widokach	95
Wysyłanie wiadomości e-mail w Django	97
Wysyłanie wiadomości e-mail w widokach	101
Renderowanie formularzy w szablonach	103
Utworzenie systemu komentarzy	107
Tworzenie modelu komentarzy	108
Dodawanie modeli do witryny administracyjnej	110
Utworzenie formularza na podstawie modelu	110
Obsługa klasy ModelForm w widoku	112
Tworzenie szablonów formularza komentarza	114
Dodawanie komentarzy do widoku szczegółów posta	116
Dodawanie komentarzy do szablonu szczegółów posta	117
Dodatkowe zasoby	123
Podsumowanie	124
ROZDZIAŁ 3.	
Rozbudowa aplikacji bloga	125
Dodanie funkcjonalności tagów	126
Pobieranie podobnych postów	134
Utworzenie własnych filtrów i znaczników szablonu	139
Utworzenie własnych znaczników szablonu	140
Utworzenie tagu szablonu typu simple_tag	140
Tworzenie znacznika szablonu typu inclusion_tag	143

Tworzenie znacznika szablonu, który zwraca kolekcję QuerySet	145
Utworzenie własnych filtrów szablonu	146
Tworzenie filtra szablonu obsługującego składnię Markdown	147
Dodanie mapy witryny	152
Utworzenie kanału wiadomości dla postów bloga	156
Dodanie do bloga wyszukiwania pełnotekstowego	163
Instalacja PostgreSQL	164
Utworzenie bazy danych PostgreSQL	165
Zrzucanie istniejących danych	165
Przełączanie bazy danych w projekcie	166
Ładowanie danych do nowej bazy danych	167
Proste wyszukiwania	168
Wyszukiwanie w wielu polach	169
Utworzenie widoku wyszukiwania	170
Stemming i ranking wyników	172
Stemming i usuwanie słów ze stoplisty dla różnych języków	175
Wagi zapytań	175
Wyszukiwanie z podobieństwem trygramu	176
Dodatkowe zasoby	178
Podsumowanie	178
ROZDZIAŁ 4.	
Utworzenie witryny społecznościowej	180
Utworzenie projektu witryny społecznościowej	181
Rozpoczęcie pracy nad aplikacją społecznościową	181
Użycie frameworka uwierzytelniania w Django	183
Utworzenie widoku logowania	184
Użycie widoków uwierzytelniania w Django	191
Widoki logowania i wylogowania	192
Widoki zmiany hasła	198
Widoki odzyskiwania hasła	199
Rejestracja użytkownika i profile użytkownika	208

Rejestracja użytkownika	208
Rozbudowa modelu User	214
Instalowanie modułu Pillow i udostępnianie plików multimedialnych	216
Tworzenie migracji dla modelu profilu	217
Użycie frameworka messages	223
Implementacja własnego backendu uwierzytelniania	226
Uniemżliwianie użytkownikom korzystania z istniejącego adresu e-mail	229
Dodatkowe zasoby	230
Podsumowanie	231
ROZDZIAŁ 5.	
Implementacja uwierzytelniania za pomocą witryn społecznościowych	232
Dodanie do witryny uwierzytelnienia za pomocą innej witryny społecznościowej	233
Uruchomienie serwera programistycznego za pośrednictwem HTTPS	236
Uwierzytelnienie za pomocą serwisu Facebook	239
Uwierzytelnienie za pomocą serwisu Twitter	246
Uwierzytelnienie za pomocą serwisu Google	256
Tworzenie profili dla użytkowników rejestrujących się za pomocą uwierzytelniania społecznościowego	264
Dodatkowe zasoby	266
Podsumowanie	267
ROZDZIAŁ 6.	
Udostępnianie treści w witrynie internetowej	268
Utworzenie witryny internetowej do kolekcjonowania obrazów	269
Utworzenie modelu Image	269
Zdefiniowanie relacji typu „wiele do wielu”	271
Rejestracja modelu Image w witrynie administracyjnej	272
Umieszczanie treści pochodzącej z innych witryn internetowych	273
Usunięcie zawartości pól formularza	274
Instalowanie biblioteki Requests	275
Nadpisanie metody save() egzemplarza ModelForm	275

Utworzenie bookmarkletu za pomocą JavaScript	280
Utworzenie szczegółowego widoku obrazu	294
Utworzenie miniatur za pomocą modułu easy-thumbnails	296
Dodawanie asynchronicznych operacji za pomocą JavaScript	299
Załadowanie JavaScript w modelu DOM	301
Ataki CSRF w żądaniach HTTP w JavaScript	302
Wykonywanie żądań HTTP za pomocą JavaScript	303
Dodanie do listy obrazów nieskończonego stronicowania	309
Dodatkowe zasoby	316
Podsumowanie	317
ROZDZIAŁ 7.	
Śledzenie działań użytkownika	319
Utworzenie systemu obserwacji	320
Utworzenie relacji typu „wiele do wielu” za pomocą modelu pośredniego	320
Utworzenie widoków listy i szczegółowego dla profilu użytkownika	324
Dodanie działań obserwowania i rezygnacji z obserwowania użytkownika za pomocą JavaScript	329
Budowa aplikacji z ogólnym strumieniem aktywności	332
Użycie frameworka contenttypes	333
Dodanie do modelu relacji generycznych	334
Uniknięcie powielonych działań w strumieniu aktywności	338
Dodanie działania użytkownika do strumienia aktywności	339
Wyświetlanie strumienia aktywności	341
Optymalizacja kolekcji QuerySet dotyczącej powiązanych obiektów	342
Tworzenie szablonów dla działań użytkowników	344
Użycie sygnałów dla denormalizowanych zliczeń	347
Praca z sygnałami	347
Definiowanie klas konfiguracyjnych aplikacji	350
10 Django 4	
Korzystanie z paska narzędzi Django Debug Toolbar	352
Instalacja paska narzędzi Django Debug Toolbar	352

Panele paska narzędzi Django Debug Toolbar	355
Polecenia paska narzędzi Django Debug Toolbar	358
Zliczanie wyświetleń obrazu za pomocą bazy danych Redis	359
Instalacja Dockera	359
Instalacja bazy danych Redis	360
Użycie bazy danych Redis z Pythonem	362
Przechowywanie różnych elementów widoków w bazie danych Redis	363
Przechowywanie rankingu w bazie danych Redis	365
Kolejne kroki z bazą danych Redis	367
Dodatkowe zasoby	369
Podsumowanie	369
ROZDZIAŁ 8.	
Utworzenie sklepu internetowego	371
Utworzenie projektu sklepu internetowego	372
Utworzenie modeli katalogu produktów	373
Rejestracja modeli katalogu w witrynie administracyjnej	377
Utworzenie widoków katalogu	379
Utworzenie szablonów katalogu	381
Utworzenie koszyka na zakupy	386
Użycie sesji Django	387
Ustawienia sesji	388
Wygaśnięcie sesji	389
Przechowywanie koszyka na zakupy w sesji	389
Utworzenie widoków koszyka na zakupy	394
Utworzenie procesora kontekstu dla bieżącego koszyka na zakupy	401
Rejestracja zamówień klienta	405
Utworzenie modeli zamówienia	405
Dołączenie modeli zamówienia w witrynie administracyjnej	407
Utworzenie zamówień klienta	408
Zadania asynchroniczne	414
Wykorzystywanie zadań asynchronicznych	414

Wątki robocze, kolejki komunikatów i brokery komunikatów	414
Dodatkowe zasoby	426
Podsumowanie	426
ROZDZIAŁ 9.	
Zarządzanie płatnościami i zamówieniami	427
Integracja bramki płatności	428
Tworzenie konta Stripe	429
Instalowanie biblioteki Pythona do obsługi serwisu Stripe	431
Dodanie do projektu obsługi serwisu Stripe	432
Budowanie procesu płatności	433
Testowanie płatności	442
Korzystanie z webhooków do otrzymywania powiadomień o płatnościach	449
Odwoływanie się do płatności Stripe w zamówieniach	457
Wdrożenie do produkcji	461
Eksport zamówień do plików CSV	461
Dodanie własnych działań do witryny administracyjnej	461
Rozbudowa witryny administracyjnej za pomocą własnych widoków	464
Dynamiczne generowanie faktur w formacie PDF	469
Instalacja WeasyPrint	470
Utworzenie szablonu PDF	470
Generowanie pliku w formacie PDF	471
Wysyłanie dokumentów PDF za pomocą poczty elektronicznej	474
Dodatkowe zasoby	478
Podsumowanie	479
ROZDZIAŁ 10.	
Rozbudowa sklepu internetowego	480
Utworzenie systemu kuponów	480
Utworzenie modeli kuponu	481
Zastosowanie kuponu w koszyku na zakupy	483
Zastosowanie kuponu w zamówieniu	492
Zastosowanie kuponów w sesji Stripe Checkout	496

Dodawanie kuponów do zamówień w serwisie administracyjnym oraz do faktur w formacie PDF	499
Utworzenie silnika rekomendacji produktu	502
Rekomendacja produktu na podstawie wcześniejszych transakcji	503
Dodatkowe zasoby	510
Podsumowanie	511
ROZDZIAŁ 11.	
Internacjonalizacja sklepu internetowego	512
Internacjonalizacja za pomocą Django	513
Ustawienia internacjonalizacji i lokalizacji	513
Polecenia przeznaczone do zarządzania internacjonalizacją	514
Instalowanie zestawu narzędzi gettext	514
Jak dodać tłumaczenie do projektu Django?	515
W jaki sposób Django określa bieżący język?	515
Przygotowanie projektu do internacjonalizacji	516
Tłumaczenie kodu Pythona	517
Tłumaczenie standardowe	518
Tłumaczenie leniwe	518
Tłumaczenia zawierające zmienne	518
Liczba mnoga w tłumaczeniu	519
Tłumaczenie własnego kodu	519
Tłumaczenie szablonów	523
Znacznik szablonu {% trans %}	523
Znacznik szablonu {% blocktrans %}	524
Tłumaczenie szablonów sklepu internetowego	524
Użycie interfejsu do tłumaczeń o nazwie Rosetta	528
Opcja fuzzy	531
Wzorce adresów URL dla internacjonalizacji	531
Dodanie prefiksu języka do wzorców adresów URL	532
Tłumaczenie wzorców adresów URL	533
Umożliwienie użytkownikowi zmiany języka	537

Tłumaczenie modeli za pomocą django-parler	539
Instalacja django-parler	539
Tłumaczenie pól modelu	540
Integracja tłumaczy w witrynie administracyjnej	542
Utworzenie migracji dla tłumaczy modeli	543
Używanie tłumaczy z mechanizmem ORM	545
Adaptacja widoków dla tłumaczy	546
Format lokalizacji	548
Użycie modułu django-localflavor do weryfikacji pól formularza	549
Dodatkowe zasoby	551
Podsumowanie	552
ROZDZIAŁ 12.	
Budowa platformy e-learningu	553
Utworzenie platformy e-learningu	554
Obsługa plików multimedialnych	555
Utworzenie modeli kursu	556
Rejestracja modeli w witrynie administracyjnej	558
Użycie fikstur w celu dostarczenia początkowych danych dla modeli	559
Utworzenie modeli dla zróżnicowanej treści	562
Wykorzystanie dziedziczenia modelu	563
Utworzenie modeli treści	565
Utworzenie własnych kolumn modelu	568
Dodawanie porządkowania do modułów i obiektów treści	570
Dodanie widoków uwierzytelniania	574
Dodanie systemu uwierzytelniania	574
Utworzenie szablonów uwierzytelniania	575
Dodatkowe zasoby	578
Podsumowanie	578
ROZDZIAŁ 13.	
Tworzenie systemu zarządzania treścią	580
Utworzenie systemu zarządzania treścią	581

Utworzenie widoków opartych na klasach	581
Użycie domieszek w widokach opartych na klasach	582
Praca z grupami i uprawnieniami	584
Zarządzanie modułami kursu i treścią	592
Użycie zbiorów formularzy dla modułów kursów	592
Dodanie treści do modułów kursów	596
Zarządzanie modułami i treścią	602
Zmiana kolejności modułów i treści	607
Dodatkowe zasoby	615
Podsumowanie	616
ROZDZIAŁ 14.	
Renderowanie i buforowanie treści	617
Wyświetlanie kursów	618
Dodanie rejestracji uczestnika	623
Utworzenie widoku rejestracji uczestnika	623
Zapisanie się na kurs	625
Uzyskanie dostępu do treści kursu	629
Renderowanie różnych rodzajów treści	633
Użycie frameworka buforowania	636
Dostępne mechanizmy buforowania	636
Instalacja Memcached	637
Instalowanie obrazu Dockera mechanizmu Memcached	637
Instalacja powiązania Memcached dla języka Python	638
Ustawienia buforowania we frameworku Django	638
Dodanie Memcached do projektu	639
Poziomy buforowania	639
Użycie niskopoziomowego API buforowania	640
Sprawdzanie żądań pobierających dane z bufora za pomocą paska narzędzi	
Django Debug Toolbar	642
Buforowanie fragmentów szablonu	646
Buforowanie widoków	647

Mechanizm buforowania bazy danych Redis	650
Monitorowanie bazy danych Redis za pomocą Django Redisboard	650
Dodatkowe zasoby	652
Podsumowanie	653
ROZDZIAŁ 15.	
Utworzenie API	654
Utworzenie API typu RESTful	655
Instalacja frameworka REST Django	655
Definiowanie serializatorów	656
Klasy parserów i renderowania formatów	657
Utworzenie widoków listy i szczegółowego	658
Wykorzystanie API	660
Opracowanie zagnieżdżonych serializatorów	662
Tworzenie własnych widoków API	664
Obsługa uwierzytelniania	665
Określenie uprawnień do widoków	666
Utworzenie kolekcji ViewSet i routerów	668
Dołączanie dodatkowych operacji do kolekcji ViewSet	670
Tworzenie niestandardowych uprawnień	671
Serializacja treści kursu	671
Wykorzystanie API RESTful	674
Dodatkowe zasoby	677
Podsumowanie	678
ROZDZIAŁ 16.	
Budowanie serwera czatu	679
Utworzenie aplikacji czatu	679
Implementacja widoku pokoju czatu	680
Obsługa czasu rzeczywistego w Django za pomocą frameworka Channels	683
Aplikacje asynchroniczne z wykorzystaniem ASGI	683
Cykl żądanie-odpowiedź z wykorzystaniem frameworka Channels	684
Instalacja frameworka Channels	686

Pisanie konsumenta	688
Routing	690
Implementacja klienta WebSocket	691
Warstwa kanału komunikacyjnego	698
Kanały komunikacyjne i grupy	698
Konfiguracja warstwy kanału komunikacyjnego z wykorzystaniem Redis	698
Aktualizacja konsumenta w celu rozgłaszania wiadomości	700
Dodawanie kontekstu do wiadomości	704
Modyfikacja konsumenta w celu uzyskania pełnej asynchroniczności	707
Integracja aplikacji czatu z istniejącymi widokami	709
Dodatkowe zasoby	711
Podsumowanie	711
ROZDZIAŁ 17.	
Wdrożenie	712
Tworzenie środowiska produkcyjnego	713
Zarządzanie ustawieniami dla wielu środowisk	713
Korzystanie z systemu Docker Compose	716
Korzystanie z systemu Docker Compose	717
Tworzenie pliku Dockerfile	717
Dodanie wymagań Pythona	719
Tworzenie pliku Docker Compose	720
Konfigurowanie usługi PostgreSQL	723
Stosowanie migracji bazy danych i tworzenie superużytkownika	726
Konfigurowanie usługi Redis	726
Serwowanie Django za pomocą WSGI i NGINX	728
Korzystanie z uWSGI	728
Konfiguracja uWSGI	729
Korzystanie z NGINX	731
Konfiguracja NGINX	732
Korzystanie z nazwy hosta	734
Udostępnianie zasobów statycznych i multimedialnych	735

Zabezpieczanie witryny za pomocą protokołu SSL/TLS	738
Sprawdzenie gotowości projektu do wdrożenia do produkcji	738
Konfiguracja projektu do obsługi SSL/TLS	739
Utworzenie certyfikatu SSL/TLS	740
Konfiguracja serwera NGINX do wykorzystania SSL/TLS	741
Przekierowywanie ruchu HTTP do HTTPS	744
Wykorzystanie serwera Daphne z frameworkiem Django Channels	745
Wykorzystanie bezpiecznych połączeń dla gniazd WebSocket	747
Uwzględnienie Daphne w konfiguracji NGINX	747
Utworzenie własnej warstwy middleware	751
Utworzenie oprogramowania middleware do obsługi subdomeny	752
Implementacja własnych poleceń administracyjnych	754
Dodatkowe zasoby	757
Podsumowanie	758