

به نام حضرت دوست، که هر چه داریم از اوست

بیش تخصصی

# DJANGO 5

## با مثال

ساخت وب اپلیکیشن قدرتمند و قابل اطمینان پایتون از ابتدا

ویرایش پنجم

Antonio Melé

با مقدمه پائلو مرکوری (مشارکت کننده جنگو)

ترجمه: حسین یسوی

انتشارات پندار پارس

سرشناسه	: مله، آنتونیو - Melé, Antonio
عنوان و نام پدیدآور	: پیش تخصصی Django 5 با مثال: ساخت وب اپلیکیشن قدرتمند و قابل اطمینان پایتون از ابتدا / مولف آنتونیو مله؛ با مقدمه پائلو مرکوری (مشارکت کننده جنگو)؛ ترجمه حسین یعسوبی.
مشخصات نشر	: تهران: پندار پارس، ۱۴۰۳ = ۲۰۲۵ م.
مشخصات ظاهری	: ۸۴۰ ص.
شابک	: ۹۷۸-۶۲۲-۷۷۸۵-۴۴-۹
وضعیت فهرست نویسی	: فیبا
یادداشت	: عنوان اصلی: Django 5 by example, 5th.ed.
عنوان دیگر	: اپلیکیشن های قدرتمند و قابل اعتماد تحت وب پایتون را از ابتدا بسازید
موضوع	: جنگو (منبع الکترونیکی) - Django (Electronic resource)
موضوع	: وبگاهها - توسعه - Web site development
	: وبگاهها - برنامه های تالیفی - Web sites -- Authoring programs
شناسه افزوده	: ملکوره، پائولو، مقدمه نویس
شناسه افزوده	: Melchiorre, Paolo
شناسه افزوده	: یعسوبی، حسین، ۱۳۵۲ - مترجم
رده بندی کنگره	: ۸۸۷TK۵۱۰۵
رده بندی دیویی	: ۷/۰۰۶
شماره کتابشناسی ملی	: ۱۰۰۰۲۵۵۰
اطلاعات رکورد کتابشناسی	: فیبا

#### انتشارات پندارپارس



دفتر فروش: انقلاب، ابتدای کارگر جنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶ [www.pendarepars.com](http://www.pendarepars.com)

تلفن: ۶۶۵۷۲۳۳۵ - ۶۶۹۲۶۵۷۸ همراه: ۰۹۱۲۲۴۵۲۳۴۸ [info@pendarepars.com](mailto:info@pendarepars.com)



نام کتاب	: پیش تخصصی در Django 5 با مثال، ساخت وب اپلیکیشن قدرتمند و قابل اطمینان		
تالیف	: آلبرتو فراری، مارکو روسو	: ترجمه	: حسین یعسوبی
چاپ نخست	: فروردین ماه ۱۴۰۴ - پندارپارس	: شمارگان	: ۱۰۰ نسخه
طرح جلد	: سارا یعسوبی	: چاپ، صحافی	: روز
قیمت	: ۸۰۰.۰۰۰ تومان	: شابک:	: ۹۷۸-۶۲۲-۷۷۸۵-۴۴-۹

\* هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد \*

## پیش‌درآمد

جنگو یک چارچوب وب سطح بالا برای زبان پایتون است که توسعه سریع و طراحی تمیز و عمل‌گرایانه را تشویق می‌کند.

این توضیحی است که در وبسایت و مخزن گیت جنگو درباره آن نوشته شده و به نظر من این تعریف به شکلی مختصر و کامل آن را معرفی می‌کند.

در تقریباً ۲۰ ساله که از نخستین نسخه جنگو گذشته، چارچوب‌های وب بسیاری ظهور کرده‌اند و با سرعت رشد کرده‌اند؛ چارچوب‌هایی که بر اساس پایتون یا دیگر زبان‌ها هستند و هر کدام جایگاه خاص خود را پیدا کرده‌اند و از تکنولوژی‌های جدید بهره برده‌اند.

با این حال، جنگو به رشد خود ادامه داده است. این چارچوب ویژگی‌های خود را بهبود داده، ثبات آن را تضمین کرده و به شکلی گام به گام اما پیوسته نوآوری کرده است، نسخه به نسخه، تا به جدیدترین نسخه خود برسد.

من از نسخه ۱.۳ جنگو شروع به استفاده از آن کردم، پس از استفاده از Zope و Plone، و فوراً تحت تأثیر این موضوع قرار گرفتم که چقدر کمک می‌کند تا برنامه‌های دنیای واقعی را به شکلی سریع و مؤثر بسازم. با گذشت سال‌ها، ویژگی‌های برنامه‌های وبی که ساخته‌ام تغییر کرده‌اند، اما جنگو همیشه به‌عنوان ابزاری قابل اعتماد برای حل چالش‌های فنی، باقی مانده است.

در طول این سال‌ها سعی کرده‌ام تجربیاتم از ویژگی‌های جنگو که بیشتر از آن‌ها لذت می‌برم را بنویسم و آن‌ها را با مثال‌ها نشان دهم. وقتی این کتاب را کشف کردم، فوراً رویکرد آنتونیو را مشابه خودم دیدم، که در آن کشف ویژگی‌های جنگو را با ساخت برنامه‌های عملی همراه می‌کند.

علاوه بر ویژگی‌های پرکاربرد جنگو، قدردان این هستم که این کتاب ابزارهای مفید دیگری مثل جنگو REST Framework و نوار ابزار خطایابی جنگو را معرفی می‌کند، و اینکه همچنین مثال‌هایی از استفاده از ویژگی‌های پیشرفته‌تر ارائه می‌شود، مثل جستجوی متنی کامل با PostgreSQL (که علاقه زیادی به آن دارم) و یکپارچه‌سازی کش با Redis.

افزون بر این، در این نسخه جدید، بسیاری از ویژگی‌های جدید که در جنگو ۵.۰ معرفی شده‌اند را خواهید یافت، مانند مقادیر پیش‌فرض محاسبه‌شده توسط پایگاه داده، ویژگی‌های مدیریتی جدید و قالب‌های ساده‌شده برای رندر کردن فیلدهای فرم.

پروژه‌های نسخه قدیمی نیز با پایتون ۳.۱۲ و بسته‌های شخص ثالث به‌روزشده و دستورالعمل‌های نصب جدید، دوباره مورد بازبینی قرار گرفته‌اند. من اضافات جالبی مثل نمودارهای توضیحی و قابلیت‌های پیشرفته DRF را بسیار دوست داشتم.

وقتی جوان بودم، شروع به یادگیری HTML کردم و سعی داشتم صفحات وبی که بازدید می‌کردم را با استفاده از یک ویرایشگر متن ویرایش کنم و از آن زمان علاقه‌مند به یادگیری فناوری‌ها به روش عملی مانده‌ام. دوست دارم پیش از مطالعه یک فناوری، خودم دست به کار شوم و دوست دارم با ایجاد چیزی ملموس، آن را انجام دهم. این روش را به شما هم توصیه می‌کنم.

این کتاب شما را به یک سفر مشابه خواهد برد، جایی که بین مطالعه قابلیت‌های جنگو و استفاده از آن‌ها در مثال‌های عملی در حرکت خواهید بود. آماده باشید که خودتان دست به کار شوید!

*Paolo Melchiorre*

*Python developer / Django contributor*

*[www.paulox.net](http://www.paulox.net)*

## درباره نویسنده

آنتونیو ملئ به عنوان مدیر مهندسی در شرکت Backbase فعالیت می‌کند، و یک پیشتانز فین‌تک در تحول دیجیتال برای مؤسسات مالی است. او پس از اینکه این شرکت Nucoro، یک پلتفرم مدیریت ثروت دیجیتال که آنتونیو یکی از بنیان‌گذاران آن بود، را خریداری کرد، در سال ۲۰۲۳ به Backbase پیوست.

آنتونیو از سال ۲۰۰۶ پروژه‌های جنگو را برای مشتریان در صنایع مختلف توسعه داده است. در سال ۲۰۰۹، شرکت Zenx IT را تأسیس کرد، یک شرکت توسعه‌دهنده که در ساخت محصولات دیجیتال تخصص دارد. او به عنوان مدیر ارشد فناوری (CTO) و مشاور فناوری در استارت‌آپ‌های مختلف مبتنی بر فناوری فعالیت کرده و تیم‌های توسعه‌دهنده‌ای را مدیریت کرده که پروژه‌هایی برای کسب‌وکارهای دیجیتال بزرگ ساخته‌اند.

آنتونیو دارای مدرک کارشناسی ارشد علوم کامپیوتر از دانشگاه Pontificia Comillas است و برنامه مدیریت پیشرفته را در MIT Sloan به اتمام رسانده است. پدر او الهام‌بخش علاقه‌اش به کامپیوتر و برنامه‌نویسی بوده است.

من نمی‌توانستم این کار را بدون گروه فوق‌العاده‌ای از علاقه‌مندان و کارشناسان پایتون انجام دهم که وقت خود را صرف بررسی نسخه‌های اولیه این کتاب کردند. قدردان همه کسانی که این کتاب را مرور کردند و نسخه‌های بتا را مطالعه کردند، هستم. بینش‌ها، اصلاحات و اشتیاق شما واقعاً این پروژه را زنده کرد.

## درباره بازبین فنی کتاب

مارک واکر پیش‌تر در زمینه توسعه نرم‌افزارهای آموزش الکترونیکی در بخش‌های دولتی و دفاعی فعالیت کرده است. او توسعه پروژه‌های جنگو را با یک برنامه خاص آفلاین آغاز کرد و سپس به توسعه پروژه‌هایی برای بزرگ‌ترین رویدادهای دویدن در جهان پرداخت. پس از آن، وارد حوزه DevOps، آزمایش بارگذاری و تأمین امنیت برنامه‌ها برای تست نفوذ شد.

در سال‌های اخیر، او به عنوان سرپرست فنی انجمن Django CMS و عضو بنیاد نرم‌افزار جنگو فعالیت داشته است. مارک همچنین در زمینه بازی‌های فانتزی ورزشی کار کرده و از AWS به GCP مهاجرت کرده است. در طول این سال‌ها، مارک به نگهداری چندین بسته (packages) پرداخته و به عنوان یکی از هدایت‌گران برنامه فضایی Djangonaut فعالیت کرده است. این برنامه هدفش آموزش و راهنمایی توسعه‌دهندگانی است که می‌خواهند در اکوسیستم جنگو مشارکت کنند.

تقدیرم بہ تو  
کہ ہمراہم بودی...

## فهرست

۲۵	فصل ۱: ساخت یک BLOG APPLICATION
۲۶	نصب پایتون
۲۷	ایجاد یک محیط مجازی پایتون
۲۸	نصب جنگو
۲۹	نصب جنگو با pip
۲۹	بازبینی جنگو
۳۰	کامپوننت‌های اصلی فریمورک
۳۰	معماری جنگو
۳۲	فیچرهای جدید جنگو ۵
۳۳	ایجاد نخستین پروژه شما
۳۵	انجام مهاجرت‌های ابتدایی دیتابیس
۳۷	اجرای سرور توسعه
۳۸	تنظیمات پروژه
۳۹	پروژه‌ها و اپلیکیشن‌ها
۴۰	ایجاد یک اپلیکیشن
۴۱	ایجاد مدل‌های داده‌ای بلاگ
۴۱	ایجاد مدل Post
۴۳	افزودن فیلدهای datetime
۴۵	تعیین یک ترتیب چیدمان پیش‌فرض
۴۶	افزودن یک ایندکس دیتابیس
۴۷	فعال‌سازی اپلیکیشن
۴۸	افزودن یک فیلد status
۵۰	افزودن یک رابطه چند-به-یک
۵۲	ایجاد و اعمال مهاجرت‌ها
۵۵	ایجاد یک سایت ادمین برای مدل‌ها
۵۵	ایجاد یک سوپر یوزر
۵۶	سایت ادمین جنگو
۵۶	افزودن مدل‌ها به سایت ادمین
۵۸	سفارشی‌سازی نحوه نمایش مدل‌ها
۶۰	افزودن شمارش‌های ظاهری برای فیلترها
۶۰	کار با QuerySets و مدیران
۶۱	ایجاد اشیاء
۶۳	به‌روزرسانی اشیاء
۶۳	بازیابی اشیاء
۶۳	فیلتربندی اشیاء

۶۴	..... استفاده از لوکاپ‌های فیلد
۶۶	..... زنجیرکردن فیلترها
۶۶	..... مستثنی کردن اشیاء
۶۶	..... چیدمان اشیاء
۶۷	..... محدودسازی QuerySet ها
۶۷	..... شمارش اشیاء
۶۸	..... بررسی موجود بودن یک شیء
۶۸	..... حذف اشیاء
۶۸	..... لوکاپ‌های پیچیده با اشیاء Q
۶۹	..... زمان ارزیابی QuerySet ها
۶۹	..... درباره QuerySet بیشتر بدانیم
۶۹	..... ایجاد مدیران مدل
۷۱	..... ساختن نماهای list و detail
۷۱	..... ایجاد نماهای list و detail
۷۳	..... استفاده از میان‌بر get_object_or_404
۷۴	..... افزودن پترن‌های URL برای نماهای خود
۷۶	..... ایجاد الگوها برای نماها
۷۶	..... ایجاد یک الگوی پایه
۷۷	..... ایجاد الگوی لیست پست
۷۹	..... دسترسی به اپلیکیشن
۷۹	..... ایجاد الگوی جزئیات پست
۸۰	..... چرخه‌ی درخواست/پاسخ
۸۱	..... مدیریت فرمان‌های به‌کار رفته در این فصل
۸۲	..... خلاصه
۸۲	..... منابع آموزشی اضافه
۸۵	..... <b>فصل ۲: ارتقاء بلاگ و افزودن ابزارهای اجتماعی</b>
۸۵	..... بازبینی عملکردی
۸۶	..... استفاده از URLهای کانونی برای مدل‌ها
۸۹	..... ایجاد URLهای سئو پسند برای پست‌ها
۹۰	..... اصلاح پترن‌های URL
۹۱	..... تغییر نماها
۹۲	..... اصلاح URL کانونی برای پست‌ها
۹۳	..... افزودن صفحه‌بندی
۹۴	..... افزودن صفحه‌بندی به نمای لیست پست
۹۵	..... ایجاد یک الگوی صفحه‌بندی
۹۷	..... رسیدگی به خطاهای صفحه‌بندی
۱۰۰	..... ساخت نماهای کلاس‌محور
۱۰۰	..... دلیل استفاده از نماهای کلاس‌محور



۱۰۰	..... استفاده از نمای کلاس‌محور برای پست‌های لیست
۱۰۳	..... پیشنهاد پست‌ها با ایمیل
۱۰۴	..... ایجاد فرم‌ها با جنگو
۱۰۵	..... رسیدگی به فرم‌ها در نماها
۱۰۷	..... ارسال ایمیل‌ها با جنگو
۱۰۸	..... کار با متغیرهای محیطی
۱۱۲	..... ارسال ایمیل در نماها
۱۱۵	..... رندر کردن فرم‌ها در الگوها
۱۱۸	..... ایجاد یک سیستم کامنت
۱۱۸	..... ایجاد یک مدل برای کامنت‌ها
۱۲۰	..... افزودن کامنت‌ها به سایت ادمین
۱۲۱	..... ایجاد فرم‌ها از مدل‌ها
۱۲۲	..... رسیدگی به ModelForms در نماها
۱۲۵	..... ایجاد الگوها برای فرم کامنت
۱۲۶	..... افزودن کامنت‌ها به نمای جزئیات پست
۱۲۸	..... افزودن کامنت‌ها به الگوی جزئیات پست
۱۳۴	..... استفاده از الگوهای ساده‌شده برای رندرکردن فرم
۱۳۶	..... خلاصه
۱۳۷	..... <b>فصل ۳: گسترش اپلیکیشن بلاگ</b>
۱۳۷	..... بازیابی عملکردی
۱۳۹	..... پیاده‌سازی تگ‌کردن به وسیله django-taggit
۱۴۸	..... بازیابی پست‌های شبیه هم
۱۵۳	..... ایجاد تگ‌ها و فیلترهای الگوی سفارشی
۱۵۴	..... پیاده‌سازی تگ‌های الگوی سفارشی
۱۵۴	..... ایجاد یک تگ الگوی ساده
۱۵۶	..... ایجاد یک تگ الگوی گنجایشی (inclusion)
۱۵۸	..... ایجاد یک تگ الگو که یک کوئری‌ست برمی‌گرداند
۱۶۰	..... پیاده‌سازی فیلترهای الگوی سفارشی
۱۶۱	..... ایجاد یک فیلتر الگو برای پشتیبانی از سینتکس Markdown
۱۶۶	..... افزودن یک نقشه‌سایت به سایت
۱۷۰	..... ایجاد feed برای پست‌های بلاگ
۱۷۶	..... افزودن جست‌وجوی فول-متن به بلاگ
۱۷۷	..... نصب Docker
۱۷۸	..... نصب PostgreSQL
۱۷۹	..... تخلیه داده‌های موجود
۱۸۰	..... سوئیچ روی دیتابیس درون پروژه
۱۸۲	..... لود شدن داده‌ها در دیتابیس جدید
۱۸۳	..... لوکاپ‌های جست‌وجوی ساده

۱۸۴	جست‌وجو نسبت به چند فیلد
۱۸۴	ساخت یک نمای جست‌وجو
۱۸۸	ریشه‌یابی و رتبه‌بندی نتایج
۱۹۰	ریشه‌یابی و حذف واژه‌های توقف در زبان‌های مختلف
۱۹۱	کوئری‌های وزن‌دهی
۱۹۲	جست‌وجو با همسانی سه ضلعی
۱۹۵	خلاصه
۱۹۵	بسط پروژه با هوش مصنوعی
۱۹۷	<b>فصل ۴: ساخت یک وبسایت اجتماعی</b>
۱۹۷	بازبینی عملکردی
۱۹۹	ایجاد یک پروژه وبسایت اجتماعی
۱۹۹	آغاز پروژه وبسایت اجتماعی
۲۰۱	استفاده از فریمورک احراز هویت جنگو
۲۰۱	ایجاد یک نمای لاگین
۲۰۸	استفاده از نماهای پیش‌ساخته احراز هویت جنگو
۲۰۹	نماهای login و logout
۲۱۶	نماهای Change password
۲۱۸	نماهای Reset password
۲۲۷	ثبت‌نام کاربر و پروفایل‌های کاربری
۲۲۷	رجیستراسی کاربر
۲۳۵	توسعه مدل کاربر
۲۳۶	نصب Pillow و خدمت‌دهی فایل‌های رسانه‌ای
۲۳۷	ایجاد مهاجرت‌ها برای مدل پروفایل
۲۴۴	استفاده از یک مدل کاربری سفارشی
۲۴۴	خلاصه
۲۴۴	سایر منابع
۲۴۷	<b>فصل ۵: پیاده‌سازی احراز هویت اجتماعی</b>
۲۴۷	بازبینی کلی
۲۴۸	ملزومات فنی
۲۴۸	استفاده از فریمورک پیام‌ها
۲۵۲	ساخت یک بک‌اند سفارشی احراز هویت
۲۵۵	جلوگیری کاربران از به‌کاربردن یک نشانی ایمیل موجود
۲۵۷	افزودن احراز هویت اجتماعی به سایت
۲۶۰	اجرای سرور توسعه از طریق HTTPS
۲۶۳	احراز هویت با استفاده از Google
۲۶۸	پروفایل‌سازی برای کاربرانی که با احراز هویت اجتماعی ثبت‌نام می‌کنند
۲۷۱	خلاصه
۲۷۳	<b>فصل ۶: اشتراک‌گذاری محتوا روی وبسایت خود</b>

۲۷۳	..... بازبینی عملکردی
۲۷۵	..... ایجاد یک وبسایت بوکمارک‌گیری تصویر
۲۷۵	..... ساخت مدل image
۲۷۷	..... ایجاد روابط چند-به-چند
۲۷۹	..... ثبت مدل تصویر در سایت ادمین
۲۷۹	..... پست کردن محتوا از سایت‌های دیگر
۲۸۰	..... پاک کردن فیلدهای فرم
۲۸۱	..... نصب کتابخانه Requests
۲۸۲	..... بازنویسی متد save() از یک ModelForm
۲۸۷	..... ساخت یک بوکمارکلت به‌وسیله JavaScript
۳۰۰	..... ایجاد یک نمای جزئیات برای تصاویر
۳۰۳	..... ایجاد تصویرک‌ها با استفاده از easy-thumbnails
۳۰۶	..... افزودن اکشن‌های ناهمگام با جاوااسکریپت
۳۰۸	..... لود کردن JavaScript روی DOM
۳۰۹	..... جعل درخواست cross-site برای درخواست‌های HTTP در جاوااسکریپت
۳۱۱	..... انجام درخواست‌های HTTP با JavaScript
۳۱۸	..... افزودن صفحه‌بندی اسکروول نامحدود به فهرست تصاویر
۳۲۵	..... خلاصه
۳۲۷	..... <b>فصل ۷: پیگیری اعمال کاربر</b>
۳۲۷	..... بازبینی عملکردی
۳۲۹	..... ساخت یک سیستم دنبال‌کردن یا follow
۳۲۹	..... ایجاد روابط چند-به-چند با یک مدل واسط
۳۳۳	..... ایجاد نماهای لیست و جزئیات برای پروفایل‌های کاربر
۳۳۹	..... افزودن اکشن‌های user follow/unfollow با جاوااسکریپت
۳۴۲	..... ایجاد یک اپلیکیشن activity stream
۳۴۴	..... استفاده از فریمورک contenttypes
۳۴۵	..... افزودن روابط عمومی به مدل‌های خود
۳۴۹	..... پرهیز از اکشن‌های تکراری در جریان فعالیت
۳۵۱	..... افزودن اکشن‌های کاربر به جریان فعالیت
۳۵۵	..... نمایش جریان فعالیت
۳۵۶	..... بهینه‌سازی کوئری‌ست‌هایی که اشیاء وابسته را درگیر می‌کنند
۳۵۸	..... ایجاد الگوها برای اکشن‌ها
۳۶۰	..... استفاده از سیگنال‌ها برای غیرنرمال کردن حساب‌ها
۳۶۰	..... کار با سیگنال‌ها
۳۶۳	..... کلاس‌های پیکربندی اپلیکیشن
۳۶۶	..... استفاده از Django Debug Toolbar
۳۶۶	..... نصب Django Debug Toolbar
۳۶۹	..... پنل‌های Django Debug Toolbar

۳۷۱	.....	فرمان‌های Django Debug Toolbar
۳۷۲	.....	شمارش نماهای تصویر با Redis
۳۷۳	.....	نصب Redis
۳۷۵	.....	استفاده از Redis با پایتون
۳۷۶	.....	ذخیره نماهای تصویر در Redis
۳۷۸	.....	ذخیره‌سازی یک رتبه‌بندی در Redis
۳۸۲	.....	گام‌های بعدی با Redis
۳۸۲	.....	خلاصه
۳۸۳	.....	بسط پروژه با استفاده از AI
۳۸۵	.....	<b>فصل ۸: ساخت یک فروشگاه آنلاین</b>
۳۸۵	.....	بازبینی عملکردی
۳۸۷	.....	ایجاد یک پروژه فروشگاه‌های آنلاین
۳۸۸	.....	ایجاد مدل‌های کاتالوگ محصول
۳۹۲	.....	ثبت مدل‌های کاتالوگ در سایت ادمین
۳۹۴	.....	ساختن نماهای کاتالوگ
۳۹۷	.....	ایجاد الگوهای کاتالوگ
۴۰۳	.....	ساخت یک سبد خرید
۴۰۳	.....	استفاده از نشست‌های جنگو
۴۰۴	.....	تنظیمات نشست
۴۰۵	.....	انقضای نشست
۴۰۶	.....	ذخیره سبد خرید در نشست‌ها
۴۱۱	.....	ایجاد نماهای سبد فروشگاه‌های
۴۱۹	.....	ایجاد یک پردازشگر محتوا برای سبد جاری
۴۲۳	.....	رجیستر کردن سفارش مشتریان
۴۲۳	.....	ایجاد مدل‌های سفارش
۴۲۵	.....	درج مدل‌های سفارش در سایت ادمین
۴۲۷	.....	ایجاد سفارش‌های مشتری
۴۳۳	.....	ایجاد کارهای ناهمگام
۴۳۳	.....	کار با وظایف ناهمگام
۴۳۳	.....	واحدهای کاری، صف پیام‌ها، و کارگزاران پیام
۴۴۰	.....	افزودن وظایف ناهمگام به اپلیکیشن
۴۴۲	.....	مانیتور کردن Celery با Flower
۴۴۵	.....	خلاصه
۴۴۷	.....	<b>فصل ۹: مدیریت پرداخت‌ها و سفارش‌ها</b>
۴۴۷	.....	بازبینی عملکردی
۴۴۹	.....	ادغام یک گیت‌وی پرداخت
۴۵۰	.....	ایجاد یک حساب Stripe
۴۵۲	.....	نصب کتابخانه Stripe Python

۴۵۲	افزودن Stripe به پروژه
۴۵۴	ساخت فرایند پرداخت
۴۶۵	آزمایش فرایند چک اوت
۴۷۱	استفاده از webhooks برای دریافت اعلان‌های پرداخت
۴۷۹	ارجاع به پرداخت‌های استرایپ در سفارش‌ها
۴۸۳	پخش زنده
۴۸۴	ارسال سفارش‌ها به فایل‌های CSV
۴۸۴	افزودن اکشن‌های سفارشی به سایت ادمین
۴۸۷	گسترش سایت ادمین با نماهای سفارشی
۴۹۳	تولید فاکتورهای PDF به شکل پویا
۴۹۴	نصب WeasyPrint
۴۹۴	ایجاد یک الگوی PDF
۴۹۵	رندر کردن فایل‌های PDF
۴۹۹	ارسال فایل‌های PDF با ایمیل
۵۰۳	خلاصه
۵۰۳	منابع بیشتر
۵۰۵	<b>فصل ۱۰؛ توسعه فروشگاه</b>
۵۰۵	بازبینی عملکردی
۵۰۷	ایجاد یک سیستم کوپن
۵۰۷	ساخت مدل coupon
۵۱۰	اعمال یک کوپن به سبب خرید
۵۱۸	اعمال کوپن‌ها به سفارش‌ها
۵۲۳	ایجاد کوپن‌ها برای Stripe Checkout
۵۲۵	افزودن کوپن‌ها با سفارش‌ها در سایت ادمین و به فاکتورهای PDF
۵۲۹	ساخت یک موتور توصیه
۵۲۹	توصیه محصولات براساس خریدهای پیشین
۵۴۲	خلاصه
۵۴۲	منابع اضافی
۵۴۳	<b>فصل ۱۱؛ افزودن بین‌المللی‌سازی به فروشگاه</b>
۵۴۳	بازبینی عملکردی
۵۴۵	بین‌المللی‌سازی با جنگو
۵۴۵	تنظیمات بین‌المللی‌سازی و بومی‌سازی
۵۴۶	فرمان‌های مدیریت بین‌المللی‌سازی
۵۴۷	نصب تولکیت gettext
۵۴۷	چگونه ترجمه‌ها را به یک پروژه جنگو بیافزاییم
۵۴۸	جنگو چگونه زبان جاری را تعیین می‌کند
۵۴۸	آماده‌سازی پروژه برای بین‌المللی‌سازی
۵۵۰	ترجمه کد پایتون

۵۵۰	.....	ترجمه‌های استاندارد (Standard translations)
۵۵۱	.....	Lazy translations
۵۵۱	.....	ترجمه‌های شامل متغیرها
۵۵۲	.....	اشکال صیغه جمع در ترجمه‌ها
۵۵۲	.....	ترجمه کد شخصی خود
۵۵۷	.....	ترجمه الگوها
۵۵۷	.....	تگ الگوی {% translate %}
۵۵۷	.....	تگ الگوی {% blocktranslate %}
۵۵۸	.....	ترجمه الگوهای فروشگاه
۵۶۲	.....	استفاده از اینترفیس ترجمه Rosetta
۵۶۵	.....	ترجمه‌های Fuzzy
۵۶۶	.....	پترن‌های URL برای بین‌المللی‌سازی
۵۶۶	.....	افزودن یک پیشوند زبانی به پترن‌های URL
۵۶۷	.....	ترجمه پترن‌های URL
۵۷۲	.....	اجازه به کاربران برای تغییر زبان
۵۷۳	.....	ترجمه مدل‌ها با django-parler
۵۷۴	.....	نصب django-parler
۵۷۴	.....	ترجمه فیلدهای مدل
۵۷۷	.....	ادغام ترجمه‌ها در سایت ادمین
۵۷۸	.....	ایجاد مهاجرت‌ها برای ترجمه‌های مدل
۵۸۱	.....	استفاده از ترجمه‌ها در QuerySets
۵۸۲	.....	تطبیق دادن نماها با ترجمه‌ها
۵۸۴	.....	فرمت‌بندی بومی‌سازی
۵۸۶	.....	استفاده از django-localflower برای اعتبارسنجی فیلدهای فرم
۵۸۸	.....	توسعه پروژه با استفاده از AI
۵۸۸	.....	خلاصه
۵۸۹	.....	منابع اضافی
۵۹۱	.....	<b>فصل ۱۲؛ ساخت یک پلتفرم E-LEARNING</b>
۵۹۱	.....	بازبینی عملکردی
۵۹۲	.....	تنظیم پروژه e-learning
۵۹۳	.....	خدمت‌رسانی به فایل‌های رسانه
۵۹۴	.....	ساخت مدل‌های دوره
۵۹۷	.....	رجیستر کردن مدل‌ها در سایت ادمین
۵۹۸	.....	استفاده از فیکسچرها برای تهیه داده‌های اولیه برای مدل‌ها
۶۰۱	.....	ایجاد مدل‌ها برای محتوای چندرخی
۶۰۳	.....	استفاده از ارث‌بری مدل
۶۰۷	.....	ایجاد مدل‌های Content
۶۱۰	.....	ایجاد فیلدهای سفارشی مدل

۶۱۲	افزودن ترتیب‌دهی به اشیاء Content و Module
۶۱۷	افزودن نماهای احراز هویت
۶۱۷	افزودن یک سیستم احراز هویت
۶۱۸	ایجاد الگوهای احراز هویت
۶۲۱	خلاصه
۶۲۲	منابع اضافی
۶۲۳	<b>فصل ۱۳: ایجاد یک سیستم مدیریت محتوا</b>
۶۲۳	بازبینی عملکردی
۶۲۵	ایجاد یک CMS
۶۲۵	ایجاد نماهای کلاس‌محور
۶۲۶	استفاده از mixins برای نماهای کلاس‌محور
۶۲۹	کار با گروه‌ها و مجوزها
۶۳۷	مدیریت ماژول‌های دوره و محتویات آنها
۶۳۷	استفاده از فرم‌سرها برای ماژول‌های دوره
۶۴۲	افزودن محتوا به ماژول‌های دوره
۶۴۸	مدیریت ماژول‌ها و محتویات آنها
۶۵۳	چیدمان دوباره ماژول‌ها و محتویات آنها
۶۵۴	استفاده از mixins از Django-braces
۶۶۲	خلاصه
۶۶۲	منابع اضافی
۶۶۳	<b>فصل ۱۴: رندر و کش کردن محتوا</b>
۶۶۴	بازبینی عملکردی
۶۶۵	نمایش کاتالوگ دوره‌ها
۶۷۰	افزودن ثبت‌نام دانشجو
۶۷۱	ایجاد یک نمای ثبت‌نام دانشجو
۶۷۳	نام‌نویسی در دوره‌ها
۶۷۷	رندر کردن محتوای دوره‌ها
۶۷۸	دسترسی به محتوای دوره‌ها
۶۸۲	رندر کردن انواع مختلفی از محتوا
۶۸۴	استفاده از فریم‌ورک کش
۶۸۵	بک‌اندهای کش موجود
۶۸۶	نصب Memcached
۶۸۶	نصب Memcached Docher image
۶۸۷	نصب Memcached Python binding
۶۸۷	تنظیمات Cache جنگو
۶۸۸	افزودن Memcached به پروژه
۶۸۸	سطوح کش کردن
۶۸۹	استفاده از API کش سطح پایین

۶۹۰	.....	Django Debug Toolbar	کش با بررسی درخواست‌های
۶۹۴	.....		کش کردن پایه سطح پایین روی داده‌های پویا
۶۹۵	.....		کش کردن قطعات الگو
۶۹۶	.....		کش کردن نماها
۶۹۷	.....	per-site	استفاده از کش
۶۹۹	.....	Redis	استفاده از بک‌اندهای کش
۷۰۰	.....	Django Redisboard	مانیتورینگ با Redis
۷۰۱	.....		خلاصه
۷۰۲	.....		منابع اضافی
۷۰۳	.....		<b>فصل ۱۵: ساخت یک API</b>
۷۰۴	.....		بازبینی عملکردی
۷۰۵	.....	RESTful API	ساختن یک
۷۰۶	.....	Django REST	نصب فریمورک
۷۰۷	.....	serializerها	تعریف
۷۰۸	.....		شناخت پارسرها و رندرکننده‌ها
۷۱۰	.....		ساخت نماهای لیست و جزئیات
۷۱۱	.....	API	مصرف
۷۱۴	.....		بسط سریالایزرها
۷۱۴	.....		افزودن فیلدهای اضافی به سریالایزرها
۷۱۶	.....		پیاپی‌سازی فیلدهای متد سریالایزر
۷۱۷	.....		افزودن صفحه‌بندی به نماها
۷۲۰	.....		ساختن سریالایزر دوره
۷۲۱	.....		سریالایز کردن وابسته‌ها
۷۲۲	.....		ایجاد سریالایزهای تودرتو
۷۲۳	.....	ViewSets	ایجاد و روترها
۷۲۷	.....	API سفارشی	ساخت نماهای
۷۲۸	.....		رسیدگی به احراز هویت
۷۲۹	.....		پیاپی‌سازی احراز هویت پایه
۷۳۰	.....		افزودن مجوزها به نماها
۷۳۲	.....	ViewSetها	افزودن اکشن‌های اضافی به
۷۳۳	.....		ایجاد مجوزهای سفارشی
۷۳۴	.....		سریالایز کردن محتوای دوره
۷۳۷	.....	RESTful API	مصرف
۷۴۱	.....		خلاصه
۷۴۱	.....		منابع اضافی
۷۴۳	.....		<b>فصل ۱۶: ایجاد یک CHAT SERVER</b>
۷۴۳	.....		بازبینی عملکردی
۷۴۴	.....	chat	ایجاد یک اپلیکیشن



۷۴۵	.....	پیاده‌سازی نمای چت‌روم
۷۴۸	.....	زمان-واقعی جنگو با Channels
۷۴۹	.....	اپلیکیشن‌های ناهمگام با استفاده از ASCII
۷۵۰	.....	چرخه درخواست/پاسخ با استفاده از Channels
۷۵۲	.....	نصب Channels و Daphne
۷۵۴	.....	نوشتن یک مصرف‌کننده
۷۵۶	.....	مسیریابی
۷۵۸	.....	پیاده‌سازی کلاینت WebSocket
۷۶۵	.....	فعال‌سازی یک لایه کانال
۷۶۵	.....	کانال‌ها و گروه‌ها
۷۶۶	.....	تنظیم یک لایه کانال با Redis
۷۶۷	.....	به‌روزرسانی مصرف‌کننده برای انتشار پیام‌ها
۷۷۲	.....	افزودن محتوا به پیام‌ها
۷۷۶	.....	تغییر مصرف‌کننده به ناهمگام بودن کامل
۷۷۸	.....	ماندگاری پیام‌ها در دیتابیس
۷۷۸	.....	ایجاد یک مدل برای پیام‌های چت
۷۸۰	.....	افزودن مدل پیام به سایت ادمین
۷۸۱	.....	ذخیره پیام‌ها در دیتابیس
۷۸۳	.....	نمایش سابقه چت
۷۸۶	.....	ادغام اپلیکیشن چت با نماهای موجود
۷۸۷	.....	خلاصه
۷۸۷	.....	منابع اضافی
۷۸۹	.....	<b>فصل ۱۷؛ پخش زنده</b>
۷۹۰	.....	ایجاد یک محیط تولید
۷۹۰	.....	مدیریت تنظیمات چند محیط
۷۹۴	.....	استفاده از Docker Compose
۷۹۴	.....	نصب Docker Compose از طریق Docker Desktop
۷۹۵	.....	ایجاد یک Dockerfile
۷۹۶	.....	افزودن الزامات پایتون
۷۹۷	.....	ایجاد یک فایل Docker Compose
۸۰۱	.....	پیکربندی سرویس PostgreSQL
۸۰۴	.....	اعمال مهاجرت‌های دیتابیس و ایجاد یک سوپر یوزر
۸۰۵	.....	پیکربندی سرویس Redis
۸۰۷	.....	خدمت‌رسانی جنگو از طریق WSGI و NGINX
۸۰۷	.....	استفاده از uWSGI
۸۰۸	.....	پیکربندی uWSGI
۸۱۰	.....	استفاده از NGINX
۸۱۱	.....	پیکربندی NGINX

۸۱۳	.....	استفاده از یک hostname
۸۱۴	.....	سرویس‌دهی دارایی‌های ایستا و رسانه
۸۱۷	.....	سرویس‌دهی سایت با SSL/TLS
۸۱۷	.....	بررسی پروژه برای تولید
۸۱۹	.....	پیکربندی پروژه جنگو برای SSL/TLS
۸۲۰	.....	ایجاد گواهی‌نامه SSL/TLS
۸۲۱	.....	پیکربندی NGINX برای استفاده SSL/TLS
۸۲۳	.....	هدایت ترافیک HTTP روی HTTPS
۸۲۴	.....	پیکربندی Daphne برای Django Channels
۸۲۶	.....	استفاده از اتصالات امن برای WebSockets
۸۲۶	.....	درج Daphne در پیکربندی NGINX
۸۲۹	.....	ایجاد یک میان‌افزار سفارشی
۸۳۲	.....	ایجاد میان‌افزار زیردامین
۸۳۵	.....	پیاپیاده‌سازی فرمان‌های مدیریتی سفارشی
۸۳۸	.....	خلاصه
۸۳۸	.....	بسط پروژه با استفاده از هوش مصنوعی
۸۳۹	.....	منابع بیشتر

## پیش‌گفتار

جنگو، یک فریمورک وب متن‌باز پایتون است که توسعه سریع و تمیز، و طراحی عملی را انجام می‌دهد. همچنین، از بسیاری از درس‌های توسعه وب به‌دور است و یک منحنی آموزش نسبتاً کم عمق برای برنامه‌نویسان مبتدی ارائه می‌دهد. جنگو با حمل مجموعه‌ای غنی و متنوع از ماژول‌هایی که مشکلات مرسوم توسعه وب را حل می‌کند، از فلسفه‌ی "باتری‌های داخلی" پایتون پیروی می‌کند. سادگی جنگو توأم با فیچرهای مفیدش، آنرا برای برنامه‌نویسان تازه‌کار و حرفه‌ای، جذاب می‌کند. جنگو برای سادگی، انعطاف‌پذیری، قابل اتکا بودن، و مقیاس‌پذیری، طراحی شده است.

امروزه جنگو توسط تعداد بی‌شماری از استارت‌آپ‌ها و سازمان‌های بزرگی همچون اینستاگرام، Spotify، Pinterest، Udemy، Robinhood، و Coursera استفاده می‌شود. تصادفی نیست که چند سالی است در نظرسنجی سالانه توسعه‌دهندگان Stack Overflow، جنگو پی‌درپی توسط توسعه‌دهندگان در کل جهان، به‌عنوان محبوب‌ترین فریمورک وب انتخاب می‌شود.

این کتاب، شما را روی کل فرایند توسعه حرفه‌ای اپلیکیشن‌های وب با جنگو، راهنمایی می‌کند. تمرکز کتاب روی توضیح نحوه کار فریمورک وب جنگو با ساخت چندین پروژه از ابتدای کار است. نه تنها مناسب‌ترین جنبه‌های این فریمورک را توضیح می‌دهیم بلکه توضیح می‌دهیم چگونه آنرا در بسیاری از موقعیت‌های گوناگون جهان واقعی، به‌کار بندیم.

این کتاب نه تنها جنگو را می‌آموزد، بلکه سایر فناوری‌های مشهور همچون PostgreSQL، Redis، Celery، RabbitMQ، و Memcached را معرفی می‌کند. در خلال کتاب، شیوه تعامل با این فناوری‌ها را در پروژه‌های جنگوی خود می‌آموزیم تا عملکردهای پیشرفته را ایجاد کنیم و اپلیکیشن‌های پیچیده‌ای را بسازیم.

این کتاب، با استفاده از یک روش گام به گام که دنبال کردنش آسان باشد به ایجاد برنامه‌های کاربردی جهان واقعی، حل مشکلات مرسوم، و پیاده‌سازی بهترین تمرین‌ها می‌پردازد. پس از خواندنش، یک شناخت خوب از نحوه کار جنگو به‌دست خواهید آورد و می‌توانید اپلیکیشن‌های وب پایتون تمام عیار را بسازید.

### کاربران این کتاب

این کتاب باید به‌عنوان یک مقدمه‌ای برای برنامه‌نویسانی که به‌تازگی وارد جنگو شده‌اند ارائه شود. این کتاب، برای توسعه‌دهگانی با دانش پایتون که می‌خواهند جنگو را به شیوه‌ای عملی بیاموزند، یا

آنهایی که از پیش، کمی با آن آشنا هستند اما می‌خواهند بیشتر با آن آشنا شوند مناسب است. این کتاب با ساخت پروژه‌های عملی از ابتدا، شما را در نواحی مختلف این فریمورک، توانمند می‌سازد. برای درک بهتر کتاب، لازم است با مفاهیم برنامه‌نویسی آشنا باشید. افزون بر دانش پایه‌ای پایتون، فرض شده است که از کمی دانش اولیه HTML و JavaScript هم برخوردارید.

## مروری بر کتاب

این کتاب، حاوی بازه‌ای از عناوین توسعه برنامه‌های کاربردی وب با جنگو است؛ و شما را از طریق ساخت چهار برنامه کاربردی وب مختلف با فیچرهای کامل، که در ۱۷ فصل دسته‌بندی شده است آموزش می‌دهد:

- یک اپلیکیشن بلاگ (فصل ۱ تا ۳)
- یک وبسایت نشانه‌گذاری تصویر (فصل ۴ تا ۷)
- یک فروشگاه آنلاین (فصل ۸ تا ۱۱)
- یک پلتفرم یادگیری - الکترونیک یا همان ای-لرنینگ (فصل ۱۲ تا ۱۷)

هر فصل، چندین فیچر جنگو را آموزش می‌دهد.

فصل ۱، ایجاد یک Blog Application، شما را از طریق یک بلاگ اپلیکیشن با جنگو آشنا می‌کند. مدل‌های بلاگ پایه، نماها، الگوها، و URLها را برای نمایش پست‌های بلاگ ایجاد خواهیم کرد. نحوه ایجاد QuerySets را با Django object-relational mapper (ORM) خواهیم آموخت، و سایت ادمین جنگو را پیکربندی می‌کنیم.

در فصل ۲، ارتقاء بلاگ با فیچرهای پیشرفته، نحوه افزودن صفحه‌بندی به بلاگ خود را می‌آموزیم، و با چگونگی پیاده‌سازی نماهای class-based جنگو آشنا می‌شویم. چگونگی ارسال ایمیل با جنگو را می‌آموزیم و فرم‌ها و فرم‌های مدل را بررسی می‌کنیم. همچنین، یک سیستم کامنت را برای پست‌های بلاگ پیاده‌سازی می‌کنیم.

در فصل ۳، بسط اپلیکیشن بلاگ خود، شیوه تعامل با اپلیکیشن‌های شخص سوم را بررسی می‌کنیم. این فصل، شما را با فرایند ساخت یک سیستم تگ‌کردن آشنا می‌کند و می‌آموزید برای سفارش پست‌های مشابه، چگونه QuerySets پیچیده بسازید. با شیوه ساخت الگوی سفارشی تگ‌ها و فیلترها آشنا می‌شوید. همچنین با روش استفاده از فریمورک سایت‌مپ و ایجاد یک فید RSS برای پست‌های خود آشنا می‌شوید. بلاگ اپلیکیشن خود را با ساخت یک موتور جست‌وجو با استفاده از قابلیت‌های جست‌وجوی فول-متن PostgreSQL کامل می‌کنیم.

در فصل ۴، ایجاد یک Social Website، توضیح می‌دهیم چگونه یک وبسایت اجتماعی بسازیم. با نحوه پیاده‌سازی نماهای احراز هویت کاربر، و نیز استفاده از فریمورک احراز هویت جنگو آشنا می‌شویم. ثبت‌نام کاربر را پیاده‌سازی می‌کنیم و مدل کاربر را با یک مدل پروفایل سفارشی بسط می‌دهیم.

در فصل ۵، پیاده‌سازی احراز هویت اجتماعی، پیاده‌سازی احراز هویت را پوشش می‌دهیم و از فریمورک پیام‌ها استفاده می‌کنیم. یک بک‌اند احراز هویت سفارشی می‌سازیم و با استفاده از OAuth 2، احراز هویت اجتماعی را با گوگل یکپارچه می‌کنیم. می‌آموزیم چگونه از django-extensions برای اجرای توسعه سرور از طریق HTTPS و سفارشی‌سازی خط لوله احراز هویت اجتماعی برای خودکارسازی ایجاد پروفایل کاربر، استفاده کنیم.

در فصل ۶، اشتراک‌گذاری محتوا روی وبسایت خود، می‌آموزیم چگونه اپلیکیشن اجتماعی خود را به یک وبسایت بوک‌مارکینگ (نشانه‌گذاری) تصویر تبدیل کنیم. روابط چند-به-چند مدل‌ها را تعریف می‌کنیم و یک بوک‌مارکلت جاوااسکریپت را ایجاد می‌کنیم که با پروژه ما در تعامل باشد. در این فصل نشان می‌دهیم چگونه تصویرک‌ها (thumbnails) را تولید کنیم. همچنین می‌آموزیم چگونه درخواست‌های ناهمگام HTTP را به کمک جاوااسکریپت و جنگو پیاده‌سازی کنیم و با ساخت صفحه‌بندی نامحدود اسکرول هم آشنا خواهید شد.

در فصل ۷، پیگیری اعمال کاربر، می‌بینید چگونه یک سیستم دنبال‌کننده برای کاربران بسازید. با ایجاد یک اپلیکیشن جریان فعالیت کاربر، وبسایت نشانه‌گذاری تصویر خود را کامل می‌کنیم. همچنین می‌آموزیم چگونه یک روابط عادی میان مدل‌ها برقرار سازیم و QuerySets را بهینه سازیم. با سیگنال‌ها و پیاده‌سازی عادی‌سازی (denormalization) کار خواهیم کرد. از Django Debug Toolbar برای تعیین اطلاعات دیباگ مناسب استفاده می‌کنیم. در پایان، Redis را به پروژه خود می‌افزاییم تا دیده شدن‌های تصاویر را بشمارد و یک رتبه‌بندی بیشترین تصاویر دیده شده را با Redis ایجاد می‌کنیم.

در فصل ۸، ساخت یک فروشگاه آنلاین، نحوه ایجاد یک فروشگاه آنلاین را بررسی می‌کنیم. مدل‌هایی را برای یک کاتالوگ محصول می‌سازیم، و یک کارت فروش را با استفاده از نشست‌های جنگو می‌سازیم. یک پروسیجر محتوا را برای کارت فروش می‌سازیم و می‌آموزیم چگونه سفارش‌های مشتری را مدیریت کنیم. در این فصل می‌آموزیم چگونه اعلان‌های غیرهمزمان یا ناهمگام را با استفاده از Celery و RabbitMQ ارسال کنیم. مانیتور کردن Celery به کمک Flower را نیز می‌آموزیم.

در فصل ۹، مدیریت پرداخت‌ها و سفارش‌ها، توضیح می‌دهیم چگونه یک مسیر پرداخت را به فروشگاه خود متصل کنیم. ما Stripe Checkout را یکپارچه می‌کنیم و اعلان‌های پرداخت غیرهمزمان را در اپلیکیشن خود دریافت می‌کنیم. نماهای سفارشی را در سایت ادمین پیاده‌سازی می‌کنیم و سایت ادمین

را نیز برای ارسال سفارش‌ها در فایل‌های CSV سفارشی‌سازی می‌کنیم. با نحوه تولید PDF فاکتورها به شکل پویا آشنا می‌شویم.

در فصل ۱۰، توسعه فروشگاه، به شما می‌آموزیم چگونه یک سیستم کوپن را برای اعمال تخفیف به سبد فروشگاه، ایجاد کنیم. برای پیاده‌سازی تخفیف کوپن‌ها، تعامل Stripe Checkout را به‌روز می‌کنیم و کوپن‌ها را به سفارش‌ها اعمال می‌کنیم. برای نگهداری محصولات که معمولاً با هم فروش رفته‌اند، از Redis استفاده می‌کنیم و از این اطلاعات برای ساخت یک موتور توصیه محصول استفاده می‌کنیم.

در فصل ۱۱، بین‌المللی‌سازی فروشگاه، نشان می‌دهیم چگونه بین‌المللی کردن را به پروژه خود بیافزاییم. نحوه تولید و مدیریت فایل‌های ترجمه‌ای و رشته‌های ترجمه‌ای در کد پایتون و الگوهای جنگو را می‌آموزیم. برای مدیریت ترجمه‌ها و پیاده‌سازی URL‌های زبان‌های مختلف، از Rosetta استفاده می‌کنیم. با شیوه ترجمه فیلدهای مدل با استفاده از django-parler و نحوه استفاده از ترجمه‌ها به کمک ORM آشنا می‌شوید. در آخر، یک فیلد فرم محلی‌شده را به کمک django-localflavor خواهیم ساخت.

در فصل ۱۲، ساخت یک پلتفرم e-learning، با شیوه ساخت یک پلتفرم ای-لرنینگ آشنا خواهید شد. وسایلی را در پروژه خود نصب می‌کنیم و مدل‌های ابتدایی را برای سیستم مدیریت محتوا می‌سازیم. از ارث‌بری مدل برای ساخت مدل داده برای محتوای چندشکلی استفاده می‌کنیم. با ساخت یک فیلد برای اشیاء سفارش، با نحوه ساخت فیلدهای مدل سفارشی آشنا می‌شویم. همچنین، نماهای احراز هویت برای CMS را پیاده‌سازی می‌کنیم.

در فصل ۱۳، ایجاد یک سیستم مدیریت محتوا، نحوه ایجاد یک CMS را با استفاده از نماهای کلاس-محور و میکسین‌ها می‌آموزیم. از گروه‌های جنگو و سیستم مجوزها برای محدودسازی دسترسی به نماها و پیاده‌سازی مجموعه فرم‌ها برای ویرایش محتوای دوره‌ها استفاده می‌کنیم. همچنین، برای چیدمان مجدد ماژول‌های دوره (course) و محتوای آنها با استفاده از جاوااسکریپت و جنگو، یک عملکرد درگ-اند-دروپ را نیز می‌سازیم.

در فصل ۱۴، رندر کردن و کش کردن محتویات، نشان می‌دهیم چگونه نماهای عمومی کاتالوگ دوره را پیاده‌سازی کنیم. یک سیستم ثبت‌نام دانش‌آموزی را ایجاد خواهیم کرد و نام‌نویسی دانش‌آموز روی دوره‌ها را مدیریت می‌کنیم. عملکرد (فانکشنالیتی) را برای رندرگیری انواع مختلف محتویات برای ماژول‌های دوره، ایجاد خواهیم کرد. می‌آموزیم چگونه به‌کمک فریمورک کش جنگو، محتویات را کش کنیم و بک‌اندهای کش Memcached و Redis را برای پروژه خود پیکربندی کنیم. در پایان، نحوه مانیتور کردن Redis را با استفاده از سایت ادمن، خواهیم آموخت.

فصل ۱۵، ساخت یک API، بررسی ساخت یک RESTful API برای پروژه با استفاده از فریمورک REST جنگو. در اینجا می‌آموزید چگونه سریال‌سازها را برای مدل‌های خود ایجاد کنید و نماهای API سفارشی را بسازید. تصدیق API را رسیدگی خواهیم کرد و مجوزها را برای نماهای API پیاده‌سازی می‌کنیم.

می‌آموزیم چگونه مجموعه نماهای API را برای روترها بسازیم. همچنین، یاد می‌گیریم چگونه به کمک کتابخانه Requests می‌توانیم API خود را مصرف کنیم.

فصل ۱۶، ساخت یک Chat Server، توضیح می‌دهد چگونه از Django Channels برای ساخت یک چت سرور ریل-تایم برای دانش‌آموزان استفاده کنیم. می‌آموزیم چگونه عملکردهایی را پیاده‌سازی کنیم که از طریق WebSockets به ارتباطات غیرهمزمان تکیه کنند. از Redis برای تنظیم یک لایه کانال استفاده می‌کنیم و می‌آموزیم چگونه مصرف‌کننده WebSocket را به شکل کاملاً غیرهمزمان بسازیم. همچنین، یک تاریخچه چت را با نگه داشتن پیام‌های چت در دیتابیس، پیاده‌سازی می‌کنیم.

فصل ۱۷، پخش زنده، نشان می‌دهد چگونه تنظیم‌هایی را برای چند محیط ایجاد کنیم و چگونه به کمک PostgreSQL، Redis، uWSGI، NGINX، و Daphne به همراه Docker Compose یک محیط تولید را تنظیم کنیم. می‌آموزیم چگونه از طریق HTTPS، امنیت پروژه را تأمین کنیم و از فریمورک سیستم چک جنگو استفاده کنیم. نحوه ایجاد یک میان‌افزار سفارشی و ایجاد فرمان‌های مدیریتی سفارشی را نیز می‌آموزید.

## بیشترین بهره را از کتاب ببرید

- باید دانش خوبی از کار کردن با پایتون داشته باشید.
- باید با HTML و جاوااسکریپت راحت باشید.
- پیشنهاد می‌شود به طریق بخش‌های ۱ تا ۳ راهنمای رسمی مستند جنگو در نشانی زیر مراجعه کنید:

<https://docs.djangoproject.com/en/5.0/intro/tutorial01>

## دانلود کدهای مثال‌های کتاب

مجموعه کدهای کتاب در گیت‌هاب به نشانی زیر موجود است:

<https://github.com/PacktPublishing/Django-5-by-Example>

در نشانی زیر نیز مجموعه کدها و ویدئوهایی داریم که خارج از حوصله این کتاب است:

<https://github.com/PacktPublishing>

برای دریافت تصاویر رنگی کتاب، به مسیر زیر مراجعه کنید:

<https://packt.link/gbp/9781805125457>





# فصل ۱

## ساخت یک Blog Application

در این کتاب نحوه ساخت پروژه‌های وب در سطح حرفه‌ای را به کمک جنگو خواهید آموخت. در این فصل ابتدایی، شما را از طریق ساخت الزامی بلوک‌های یک اپلیکیشن جنگو، از نصب تا استقرار آشنا می‌کنیم. اگر جنگو را هنوز روی سیستم خود نصب نکرده‌اید، به بخش نصب جنگو بروید که با فرایندهای نصب آن آشنا شوید.

پیش از آغاز نخستین پروژه جنگو، اجازه دهید ببینیم قرار است درباره چه بیاموزیم. در این فصل، یک بازیابی عمومی روی فریمورک خواهیم داشت. این شما را از طریق اجزای مختلف اصلی، برای ایجاد یک اپلیکیشن وب کاملاً کاربردی هدایت می‌کند: مدل‌ها، الگوها، نماها، و URLها. در پایان به شناختی از چگونگی کارکرد جنگو و نحوه تعامل آن با اجزای مختلف فریمورک، دست می‌یابید.

همچنین، تفاوت میان پروژه‌های جنگو و اپلیکیشن‌ها را می‌آموزید، و درباره مهم‌ترین تنظیمات جنگو خواهید آموخت. یک اپلیکیشن بلاگ ساده خواهید ساخت که به کاربران اجازه می‌دهد از طریق همه پست‌های منتشر شده، راهبری شوند و پست‌های فردی را بخوانند. همچنین، یک اینترفیس ادمین ساده می‌سازیم که پست‌ها را مدیریت و منتشر کنیم. در دو فصل بعدی، اپلیکیشن را با عملکردهای پیشرفته‌تر گسترش می‌دهیم.

این فصل را به عنوان نقشه راه خود برای ساخت اپلیکیشن جنگوی تمام عیار در نظر بگیرید. اگر برخی اجزا یا مفاهیم در ابتدا برایتان نامفهوم است ناامید نشوید. جزئیات اجزای مختلف جنگو را در این کتاب بررسی خواهیم کرد.

در این فصل، به عنوان‌های زیر می‌پردازیم:

- نصب پایتون
- ایجاد یک محیط مجازی پایتون
- نصب جنگو
- ایجاد و پیکربندی یک پروژه جنگو
- ساخت یک اپلیکیشن جنگو
- طراحی مدل‌های داده‌ای

- ایجاد و اعمال مهاجرت‌های مدل
- تنظیم یک سایت ادمین برای مدل‌های خود
- کار با QuerySets و مدیرهای مدل
- ساخت نماها، الگوها، و URLها
- شناخت چرخه درخواست/پاسخ جنگو

کار را با نصب پایتون روی سیستم آغاز می‌کنیم.

کد منبع این فصل را می‌توانید در نشانی زیر بردارید:

<https://github.com/PacktPublishing/Django-5-by-example/tree/main/Chapter01>

همه پکیج‌های پایتون به‌کار رفته در این فصل، در فایل `requirements.txt` در کد منبع این فصل موجود است. دستورالعمل‌های نصب هر پکیج پایتون را در بخش‌های زیر دنبال کنید، یا همه الزامات را یک باره با فرمان `python -m pip install -r requirements.txt` نصب کنید.

## نصب پایتون

جنگو ۵ از پایتون ۳.۱۰ و ۳.۱۱ و ۳.۱۲ پشتیبانی می‌کند. در مثال‌های کتاب، از پایتون ۳.۱۲ استفاده می‌کنیم. اگر از لینوکس یا macOS استفاده می‌کنید شاید پایتون را نصب شده داشته باشید. اگر از ویندوز استفاده می‌کنید می‌توانید یک نصب کننده پایتون را از وبسایت `python.org` دانلود کنید. پایتون را برای سیستم‌عامل خود می‌توانید از <https://www.python.org/downloads/> دانلود کنید.

خط فرمان پرامپت پوسته سیستم خود را باز کنید. اگر از macOS استفاده می‌کنید، `Command + spacebar` را بفشارید تا Spotlight باز شود و عبارت Terminal را بنویسید تا Terminal.app باز شود. اگر از ویندوز استفاده می‌کنید، وارد منوی Start شوید و powershell را در کادر جست‌وجو وارد کنید. سپس، روی اپلیکیشن Windows PowerShell کلیک کنید تا باز شود. یا اینکه، با تایپ `cmd` در کادر جست‌وجو، و کلیک روی اپلیکیشن Command Prompt برای باز شدن آن، از فرمان پرامپت پایه‌ای‌تر استفاده کنید.

با تایپ فرمان زیر در پرامپت پوسته، بررسی کنید که پایتون ۳ روی سیستم نصب باشد:

```
python3 --version
```

اگر عبارت زیر را دیدید یعنی پایتون ۳ را نصب دارید:

```
Python 3.12.3
```

اگر با خطا مواجه شدید، به جای Python3، از عبارت python استفاده کنید. در ویندوز، به جای python از فرمان py استفاده کنید.

اگر پایتون روی سیستم شما نصب نیست یا نسخه‌ای پیش از ۳.۱۲ است، آنرا از <https://www.python.org/download/> دانلود کنید و مراحل نصب آنرا دنبال کنید. در سایت دانلود، دستورالعمل‌های نصب آن برای ویندوز و لینوکس و مک موجود است.

در خلال کتاب، هنگامی که در پرامپت پوسته به پایتون ارجاع می‌شود، از فرمان python استفاده می‌کنیم، چراکه در برخی سیستم‌ها ممکن است نیاز باشد از python3 استفاده کنند. اگر از لینوکس یا مک استفاده می‌کنید و پایتون سیستم شما، نسخه ۲ است، لازم است برای استفاده از نسخه ۳ پایتون که نصب کرده‌اید، از دستور python3 استفاده کنید. حواستان باشد که طول عمر پایتون ۲ در ژانویه ۲۰۲۰ به پایان رسید و دیگر نباید استفاده شود.

در ویندوز، عبارت python اجراکننده پایتون نصب شده پیش‌فرض شماست، درحالی که py لانچر پایتون است. لانچر پایتون برای ویندوز، در نسخه ۳.۳ پایتون معرفی شد. این تعیین می‌کند چه نسخه‌هایی از پایتون روی سیستم شما نصب است و به شکل خودکار، آخرین نسخه را تحویل می‌دهد. اگر از ویندوز استفاده می‌کنید، باید از فرمان py استفاده کنید.

اکنون شروع به ایجاد محیط پایتون برای پروژه خود و نصب کتابخانه‌های لازم می‌کنیم.

## ایجاد یک محیط مجازی پایتون

هنگامی که برنامه‌های کاربردی پایتون را می‌نویسیم، معمولاً از پکیج‌ها و ماژول‌هایی که در کتابخانه استاندارد پایتون موجود نیست استفاده می‌کنیم. ممکن است اپلیکیشن‌های پایتونی داشته باشید که نیاز به نسخه مختلفی از یک ماژول داشته باشند. هرچند، تنها یک نسخه مشخص از یک ماژول می‌تواند در گستره‌ی سیستم نصب شده باشد. اگر یک نسخه ماژول را برای یک اپلیکیشن ارتقاء می‌دهید، ممکن است سایر اپلیکیشن‌هایی که نیاز به یک نسخه قدیمی‌تر آن ماژول دارند از کار بیافتند.

برای حل این مشکل، می‌توانید از محیط‌های مجازی پایتون استفاده کنید. با محیط‌های مجازی می‌توان ماژول‌های پایتون را به جای نصب آنها در خود سیستم، در یک محل ایزوله شده نصب کرد. هر محیط مجازی، دارای کتابخانه پایتون خودش است و می‌تواند دارای مجموعه وابستگی خودش به پکیج‌های پایتون نصب شده در دایرکتوری site-packages خودش باشد.

از نسخه ۳.۳، کتابخانه venv همراه پایتون شده است که پشتیبانی برای ایجاد محیط‌های مجازی سبک را ارائه می‌دهد. با استفاده از ماژول venv پایتون برای ایجاد محیط‌های ایزوله پایتون، می‌توان از

نسخه‌های مختلف پکیج برای پروژه‌های مختلف استفاده کرد. مزیت دیگر استفاده از venv این است که نیازی به هیچ مزیت ادمینی برای نصب پکیج‌های نصب پایتون ندارید.

اگر از لینوکس یا مک استفاده می‌کنید، با فرمان زیر، یک محیط ایزوله شده بسازید:

```
python -m venv my_env
```

اگر سیستم شما پایتون ۲ داشته است و پایتون ۳ را نصب کردید فراموش نکنید به جای دستور python از python3 استفاده کنید.

اگر از ویندوز استفاده می‌کنید، به جای آن از فرمان زیر استفاده کنید:

```
py -m venv my_env
```

این از لانچر پایتون در ویندوز استفاده خواهد کرد.

فرمان بالا یک محیط پایتون در یک دایرکتوری جدید به نام my\_env ایجاد می‌کند. هر کتابخانه پایتونی که حین فعال بودن محیط مجازی خود نصب کنید به دایرکتوری my\_env/lib/python3.12/site-packages خواهد رفت.

اگر از لینوکس یا مک استفاده می‌کنید، برای فعال‌سازی محیط مجازی خود فرمان زیر را اجرا کنید:

```
source my_env/bin/activate
```

اگر از ویندوز استفاده می‌کنید از فرمان زیر استفاده کنید:

```
.\my_env\Scripts\activate
```

پرامپت پوسته، حاوی نام محیط مجازی فعال در درون پرانتز است، مانند این مورد:

```
(my_env) zenx@pc:~ zenx$
```

هر لحظه می‌توان با فرمان deactivate، محیط مجازی را غیرفعال کرد. اطلاعات بیشتر را در این نشانی ببینید: <https://docs.python.org/3/library/venv.html>

## نصب جنگو

اگر جنگو ۵ را نصب کرده‌اید از این بخش رد شوید و مستقیم به بخش ایجاد نخستین پروژه بروید.

جنگو به عنوان یک ماژول پایتون می‌آید و بنابراین می‌تواند در هر محیط پایتونی نصب شود. اگر هنوز آنرا نصب نکرده‌اید، راهنمای سریع نصب زیر را بخوانید.

## نصب جنگو با pip

سیستم مدیریت پکیج pip، متد ترجیحی نصب جنگو است. pip در پایتون ۳.۱۲ از پیش نصب شده است، اما می‌توانید دستور نصب pip را در نشانی زیر ببینید:

<https://pip.pypa.io/en/stable/installation/>

برای نصب جنگو به همراه pip، فرمان زیر را در پرامپت پوسته اجرا کنید:

```
python -m pip install Django~=5.0.4
```

این دستور، آخرین نسخه ۵ جنگو را در دایرکتوری site-packages محیط مجازی شما نصب می‌کند.

اینک بررسی می‌کنیم که آیا جنگو به شکل موفق نصب شده است یا خیر. فرمان زیر را در پرامپت پوسته اجرا کنید:

```
python -m django --version
```

اگر یک خروجی که با 5.0 آغاز می‌شود بگیرید، فرایند نصب روی سیستم با موفقیت انجام شده است. اگر پیام No module named Django را بگیرید، یعنی هنوز جنگو نصب نشده است. اگر مشکلی در نصب جنگو دارید می‌توانید گزینه‌های مختلف نصب را در نشانی زیر بازبینی کنید:

<https://docs.djangoproject.com/en/5.0/intro/install/>

همه پکیج‌های پایتون به کار رفته در این فصل، دارای فایل requirements.txt در کد منبع آن فصل است، که در بالا اشاره شد. شیوه‌نامه‌های نصب هر پکیج پایتون در بخش‌های زیر آمده است، یا می‌توانید همه ملزومات را یک بار با فرمان pip install -r requirements.txt نصب کنید.

## بازبینی جنگو

جنگو فریمورکی است که حاوی مجموعه‌ای از کامپوننت‌هاست که مشکلات مرسوم توسعه وب را حل می‌کنند. کامپوننت‌های جنگو، به شکلی سست جفت شده‌اند که یعنی می‌توانند به شکل مستقل مدیریت شوند. این به تفکیک مسئولیت‌های لایه‌های مختلف فریمورک کمک می‌کند؛ لایه دیتابیس چیزی درباره اینکه چه داده‌ای نمایش یابد نمی‌داند، سیستم الگو چیزی درباره درخواست‌های وب نمی‌داند، و غیره.

جنگو ماگزیمم کد قابل استفاده مجدد را با اصل DRY<sup>1</sup> زیر پیشنهاد می‌دهد. همچنین توسعه سریع را می‌پروراند و اجازه استفاده از کد کمتر با بهره بردن از مزیت قابلیت‌های پویای پایتون را می‌دهد، مانند درون‌گرایی.

<sup>1</sup> Don't repeat yourself

در نشانی زیر درباره فلسفه طراحی جنگو بخوانید:

<https://docs.djangoproject.com/en/5.0/misc/design-philosophies/>

### کامپوننت‌های اصلی فریمورک

جنگو از الگوی (Model-Template-View) MTV پیروی می‌کند. این تقریباً شبیه الگوی آشنای MVC (Model-View-Controller) است، جایی که الگو به‌عنوان نما عمل می‌کند و خود فریمورک به‌عنوان کنترلر عمل می‌کند.

مسئولیت‌ها در الگوی MTV به‌شکل زیر تقسیم شده است:

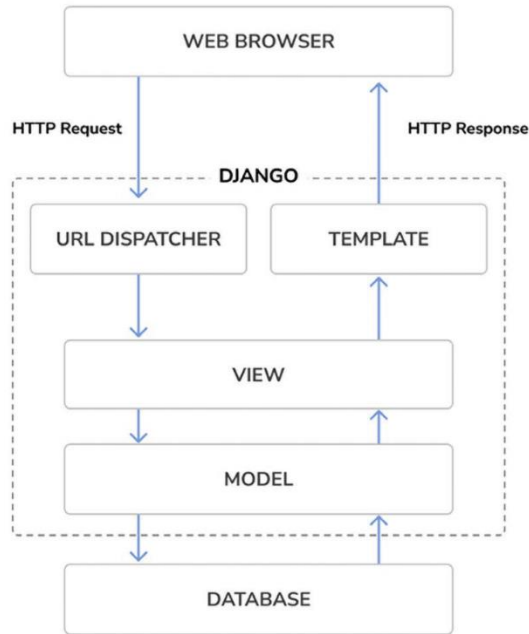
- **Model:** این ساختار منطقی داده‌ها را تعریف می‌کند و رسیدگی کننده داده‌ها میان دیتابیس و نما می‌باشد.
- **Template:** این لایه‌ی ارائه است. جنگو از یک سیستم الگوی plain-text استفاده می‌کند که هرچیزی که مرورگر رندر کند را حفظ می‌کند.
- **View:** این از طریق مدل با دیتابیس ارتباط برقرار می‌کند و داده‌ها را برای نمایش به الگو منتقل می‌کند.

خود فریمورک به‌شکل کنترلر عمل می‌کند. بسته به پیکربندی URL جنگو، یک درخواست به نمای مناسب ارسال می‌کند.

هنگام توسعه هر پروژه جنگو، همواره با مدل‌ها، نماها، الگوها، و URLها کار می‌کنید. در این فصل، می‌آموزیم چگونه آنها را با هم ترکیب کنیم.

### معماری جنگو

شکل زیر، درخواست‌های فرایندهای جنگو و نحوه مدیریت چرخه درخواست/پاسخ با کامپوننت‌های اصلی مختلف جنگو، یعنی URLها، نماها، مدل‌ها، و الگوها، را نشان می‌دهد:



چگونگی رسیدگی درخواست‌های HTTP جنگو و تولید پاسخ‌ها به شرح زیر است:

۱. یک مرورگر وب، درخواست یک صفحه با URL آنرا می‌دهد و وب‌سرور، درخواست HTTP را به جنگو پاس می‌دهد.
۲. جنگو از طریق الگوهای URL پیکربندی شده‌اش اجرا می‌شود و در نخستین موردی که با URL درخواستی تطابق دارد توقف می‌کند.
۳. جنگو نمایی که متناظر با الگوی URL تطبیقی است را اجرا می‌کند.
۴. این نما به شکل بالقوه از مدل‌های داده‌ای برای بازیابی اطلاعات از دیتابیس استفاده می‌کند.
۵. مدل‌های داده‌ای، تعاریف و رفتارهای داده‌ها را ارائه می‌دهند. آنها برای کوئری گرفتن دیتابیس به‌کار می‌روند.
۶. نما برای نمایش داده‌ها و بازگرداندن آن با یک پاسخ HTTP، یک الگو را رندر می‌کند (معمولاً HTML).

در پایان این فصل، به چرخه درخواست/پاسخ جنگو بازمی‌گردیم.

جنگو در فرایند درخواست/پاسخ، قلاب‌ها<sup>۱</sup> را دارد که middleware یا میان‌افزار نامید می‌شود. به‌منظور سادگی، عمداً میان‌افزار از این نمودار خارج شده است. در مثال‌های دیگر کتاب از میان‌افزار استفاده خواهیم کرد و می‌آموزیم چگونه میان‌افزار سفارشی ایجاد کنیم (فصل ۱۷؛ بخش زنده). عناصر کاربردی جنگو را توضیح دادیم و اینکه چگونه درخواست‌ها را پردازش می‌کند. اکنون اجازه دهید ویژگی‌های جدید نسخه ۵ جنگو را بررسی کنیم.

### فیچرهای جدید جنگو ۵

جنگو ۵ چندین ابزار کلیدی معرفی کرده که در مثال‌های کتاب استفاده خواهیم کرد. این نسخه، همچنین ابزارهای مشخصی را منع کرده و کارکردهای آنها را حذف کرده است. جنگو ۵ ابزارهای اصلی جدید زیر را ارائه می‌کند:

- فیلترهای Facet در سایت ادمین: این فیلترهای ظاهری می‌تواند اکنون به سایت ادمین افزوده شود. در صورت اعمال، شماری صورتک ظاهری، برای فیلترهای اعمال شده در لیست شیء ادمین نمایش داده می‌شود. این ابزار در بخش `Added facet counts to filters` در همین فصل آمده است.
- الگوهای ساده شده برای رندرکردن فیلد فرم: رندرکردن فیلد فرم، با این قابلیت برای تعریف گروه‌های فیلد با الگوهای تخصیصی، ساده‌سازی شده است. این به ایجاد فرایند رندگیری عناصر مربوط به یک فیلد فرم جنگو، مانند لیبل‌ها، ویجت‌ها، متن `help`، و خطاها کمک می‌کند که ساده‌تر و کارآمدتر شود. یک نمونه از کاربرد گروه‌های فیلد را می‌توان در فصل ۲ و بخش *ایجاد الگوها برای فرم کامنت مشاهده کرد*.
- مقادیر پیش‌فرض `database-computed`: جنگو مقادیر پیش‌فرض محاسبه‌شده توسط دیتابیس را اضافه می‌کند. مثالی از این ویژگی را در بخش *افزودن فیلدهای `datetime`* در همین فصل خواهید دید.
- فیلدهای مدل `database-generated`: این یک نوع فیلد جدید است که امکان ایجاد ستون‌های تولید شده در دیتابیس را می‌دهد. هر بار مدل تغییر می‌کند برای تنظیم خودکار مقدار فیلد، یک عبارت به‌کار می‌رود. مقدار فیلد با استفاده از سینتکس `GENERATED ALWAYS` در SQL تنظیم می‌شود.
- گزینه‌های بیشتر برای اعلام انتخاب‌های فیلد مدل: فیلدهایی که انتخاب‌ها را پشتیبانی می‌کند دیگر نیاز به دسترسی به صفت<sup>۲</sup> `choices`. برای دسترسی به انواع شمارش ندارند. یک مپینگ یا

<sup>1</sup> hooks

<sup>2</sup> attribute



قابل فراخوانی بودن به جای یک قابل تکرارپذیری می تواند مستقیماً برای بسط انواع شمارش به کار رود. انتخاب‌ها با انواع شمارش‌ها در این کتاب به روزرسانی شده است تا این تغییرات را منعکس کند. یک نمونه از این را می‌توانید در بخش *افزودن یک فیلد وضعیت* در این فصل ببینید.

در جنگو ۵، چندین ارتقاء در پشتیبانی غیرهمزمان نیز صورت پذیرفته است. پشتیبانی از ASGI<sup>1</sup> نخستین بار در جنگو ۳ ارائه شد و در نسخه ۴.۱ با رسیدگی‌کننده‌های غیرهمزمان برای نماهای کلاس-محور و یک اینترفیس ORM غیرهمزمان، بهبود یافت. جنگو ۵، توابع غیرهمزمان را به فریمورک تصدیق هویت می‌افزاید، از توزیع سیگنال غیرهمزمان، پشتیبانی می‌کند، و پشتیبانی غیرهمزمان از چند دکوراتور درون‌ساخت را می‌افزاید.

جنگو ۵ از پایتون ۳.۸ و ۳.۹ پشتیبانی نمی‌کند.

برای بررسی فهرست کامل تغییرات جنگو ۵ به نشانی زیر سری بزنید:

<https://docs.djangoproject.com/en/5.0/releases/5.0/>

در زمان چاپ کتاب، هیچ تغییر جدی در جنگو ۵ ایجاد نشده و اپلیکیشن‌های جنگو ۴ را می‌توان به راحتی و سرراستی، به جنگو ۵ به‌روزرسانی کرد.

اگر می‌خواهید یک پروژه جنگو موجود را به سرعت به نسخه ۵ ارتقاء دهید، می‌توانید از ابزار `django-upgrade` استفاده کنید. این پکیج، فایل‌های پروژه شما را با اعمال واسطه‌هایی برای یک نسخه هدف، بازنویسی می‌کند. برای راهنمایی بیشتر به نشانی زیر سری بزنید:

<https://github.com/adamchainz/django-upgrade>

ابزار `django-upgrade` با پکیج `pyupgrade` نفس می‌کشد. از `pyupgrade` می‌توان برای به‌روزرسانی خودکار سینتکس به جدیدترین نسخه پایتون استفاده کرد. اطلاعات بیشتر را در نشانی زیر ببینید:

<https://github.com/asottile/pyupgrade>

## ایجاد نخستین پروژه شما

نخستین پروژه جنگوی شما یک اپلیکیشن بلاگ است. این یک آشنایی ابتدایی از توانمندی‌های جنگو و کارکردهایش به شما می‌دهد.

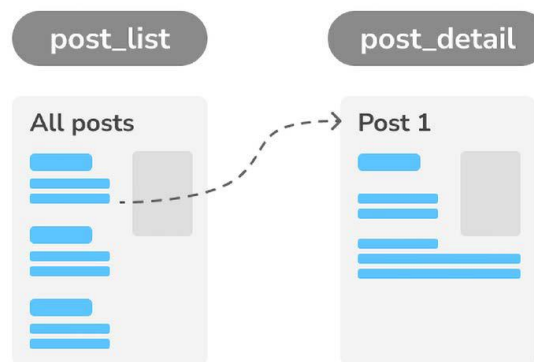
بلاگ‌نویسی، نقطه آغاز مناسبی برای ساخت یک پروژه کامل جنگو است، چون طیف گسترده‌ای از ابزارهای لازم را دربر می‌گیرد، از مدیریت محتوای پایه تا کارکردهای پیشرفته‌ای همچون

<sup>1</sup> Asynchronous Server Gateway Interface

کامنت نویسی، به اشتراک‌گذاری پست، جست‌وجو، و پست‌های پیشنهادی. پروژه بلاگ در سه فصل آغازین کتاب پوشش داده خواهد شد.

در این فصل، کار را با ایجاد پروژه جنگو و یک اپلیکیشن جنگو برای بلاگ آغاز خواهیم کرد. سپس مدل‌های خود را می‌سازیم و آنها را با دیتابیس، همگام می‌کنیم. در پایان، یک سایت ادمین برای بلاگ می‌سازیم و نماها، الگوها، و URLها را خواهیم ساخت.

شکل زیر، یک ارائه‌ای از صفحات اپلیکیشن بلاگ که قرار است بسازید را نشان می‌دهد.



اپلیکیشن بلاگ، حاوی فهرستی از پست‌ها، شامل عنوان پست، داده‌های انتشاری، نویسنده، یک پست منتخب، و یک پیوند برای خواننده پست است. صفحه لیست پست، یا نمای `post_list` پیاده‌سازی خواهد شد. در این فصل با نحوه ایجاد نماها آشنا خواهید شد.

هنگامی‌که خوانندگان روی پیوند یک پست در صفحه پست لیست کلیک کنند، به یک نمای واحد (با جزئیات) از یک پست هدایت می‌شوند. نمای جزئیات، عنوان، داده‌های انتشاری، نویسنده، و بدنه کامل پست را نمایش خواهد داد.

اجازه دهید کار را با ایجاد پروژه جنگو برای بلاگمان آغاز کنیم. جنگو یک فرمان ارائه می‌دهد که امکان ایجاد یک ساختار فایل پروژه ابتدایی را می‌دهد.

فرمان زیر را در پرامپت پوسته خود اجرا کنید:

```
django-admin startproject mysite
```

توجه: به دلیل وضوح چاپ، کادر سیاه فرمان‌های پرامپت حذف شده است.

این موجب ایجاد یک پروژه جنگو با نام `mysite` می‌شود.

**هشدار:** به منظور جلوگیری از ناسازگاری، از نام‌گذاری پروژه‌ها پس از ماژول‌های درون‌ساخت پایتون یا جنگو پرهیز کنید.

اجازه دهید نگاهی به ساختار پروژه تولید شده بیاندازیم:

```
mysite/
  manage.py
  mysite/
  __init__.py
  asgi.py
  settings.py
  urls.py
  wsgi.py
```

بیرونی‌ترین پوشه mysite/ نگهدارنده پروژه ماست، و حاوی فایل‌های زیر است:

- `manage.py`: این یک خط-فرمان سودمند به کار رفته برای تعامل با پروژه است. معمولاً نیازی به ویرایش این فایل ندارید.
- `mysite/`: این پکیج پایتون برای پروژه شماست که حاوی فایل‌های زیر است:
  - `__init__.py`: یک فایل خالی که به پایتون می‌گوید با پوشه `mysite` به شکل یک ماژول پایتون رفتار کند.
  - `asgi.py`: این پیکربندی برای اجرای پروژه به عنوان یک اپلیکیشن ASGI با وب‌سرورهای `ASGI-compatible` است. ASGI، ظهور پایتون استاندارد برای غیرهمزمانی وب‌سرورها و اپلیکیشن‌هاست.
  - `settings.py`: این شامل تنظیمات و پیکربندی پروژه شماست و حاوی تنظیمات پیش‌فرض آغازین است.
  - `urls.py`: این جایی است که الگوهای URL زندگی می‌کنند. هر URL تعریفی در اینجا به یک نما نگاشت می‌شود.
  - `wsgi.py`: این پیکربندی برای اجرای پروژه شما به شکل یک اپلیکیشن WSGI<sup>1</sup> با وب‌سرورهای سازگار با WSGI است.

### انجام مهاجرت‌های ابتدایی دیتابیس

اپلیکیشن‌های جنگو نیاز به یک دیتابیس برای نگهداری داده‌ها دارند. فایل `settings.py` حاوی پیکربندی دیتابیس برای پروژه شما با تنظیمات `DATABASES` است. پیکربندی پیش‌فرض، یک دیتابیس `SQLite3` است. `SQLite3` با مجموعه پایتون ۳ می‌آید و می‌تواند در هر یک از اپلیکیشن‌های پایتون

<sup>1</sup> Web Server Gateway Interface

شما به کار رود. SQLite یک دیتابیس سبک وزن است که می‌تواند با جنگو برای توسعه به کار رود. اگر برنامه‌ریزی شما، گسترش اپلیکیشن خود در یک محیط تولید است باید از یک دیتابیس پر ابزار همچون PostgreSQL، MySQL یا Oracle استفاده کنید. برای کسب اطلاعات بیشتر درباره چگونگی رسیدن به دیتابیس اجرایی با جنگو به نشانی زیر بروید:

<https://docs.djangoproject.com/en/5.0/topics/install/#database-installation>

فایل settings.py نیز شامل لیستی به نام INSTALLED\_APPS است که حاوی اپلیکیشن‌های مرسوم جنگو است که به طور پیش فرض به پروژه شما افزوده می‌شود. در ادامه، در بخش تنظیمات پروژه، به سراغ این اپلیکیشن‌ها خواهیم رفت.

برنامه‌های کاربردی جنگو حاوی مدل‌های داده‌ای است که به جدول‌های دیتابیس نگاشت می‌شود. در بخش ایجاد مدل‌های داده‌ای بلاگ، مدل‌های خود را ایجاد خواهیم کرد. برای تکمیل ستاپ پروژه، باید جدول‌های مرتبط با مدل‌های اپلیکیشن‌های پیش فرض جنگو مشتمل بر تنظیم INSTALLED\_APPS را ایجاد کنید. جنگو دارای سیستمی است که در مدیریت مهاجرت‌های دیتابیس به شما کمک می‌کند.

پرامپت پوسته را باز کنید و فرمان‌های زیر را اجرا کنید:

```
cd mysite
python manage.py migrate
```

یک خروجی را خواهید دید که با خطوط زیر پایان می‌یابد:

```
Applying contenttypes.0001_initial... OK
Applying auth.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Applying admin.0003_logentry_add_action_flag_choices... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... 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
```

خطوط بالا، مهاجرت‌های دیتابیس است که توسط جنگو اعمال می‌شود. با اعمال مهاجرت‌های ابتدایی، جدول‌های اپلیکیشن‌های فهرست شده در تنظیم INSTALLED\_APPS در دیتابیس ایجاد می‌شود. در بخش *ایجاد و اعمال مهاجرت‌ها* در همین فصل بیشتر درباره فرمان مهاجرت `migrate` خواهید آموخت.

### اجرای سرور توسعه

جنگو به همراه یک وب‌سرور سبک وزن می‌آید که بدون نیاز به صرف زمان برای پیکربندی یک سرور تولیدی، کدهای شما را سریع‌تر اجرا کند. هنگامی که سرور توسعه‌ای جنگو را اجرا می‌کنید، کار چک کردن تغییرات در کد شما را انجام می‌دهد. به شکل خودکار لود می‌شود، و شما را از لود کردن دوباره آن به شکل دستی پس از تغییرات کد، خلاص می‌کند. هرچند، شاید متوجه برخی کارها همچون افزودن فایل‌های جدید به پروژه نشود، پس در موارد این چنینی مجبورید سرور را به شکل دستی ریستارت کنید.

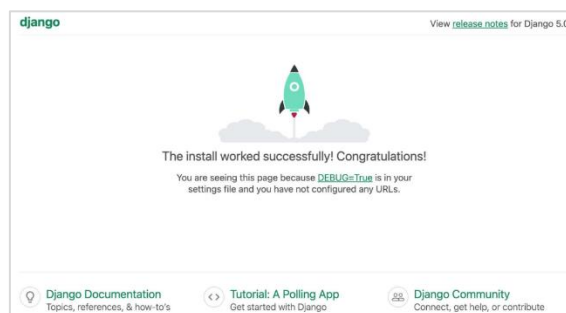
با تایپ فرمان زیر در پرامپت پوسته، توسعه سرور را آغاز می‌کنیم:

```
python manage.py runserver
```

باید چیزهایی را ببینید؛ همچون پیام‌های زیر:

```
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
January 01, 2024 - 10:00:00
Django version 5.0, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

اینک `http://127.0.0.1:8000/` را در مرورگر خود باز کنید. باید یک صفحه ایستا را ببینید که می‌گوید پروژه به شکل موفق اجرا شده است:



شکل بالا اشاره می‌کند که جنگو در حال اجرا است. اگر نگاهی به کنسول خود بیاندازید درخواست GET را خواهید دید که توسط مرورگر انجام شده است:

```
[01/Jan/2024 10:00:15] "GET / HTTP/1.1" 200 16351
```

هر درخواست HTTP توسط سرور توسعه در کنسول ثبت می‌شود. هر خطایی که حین اجرای سرور توسعه رخ دهد نیز در کنسول آشکار خواهد شد.

سرور توسعه جنگو را روی یک میزبان و پورت سفارشی می‌توان اجرا کرد یا به جنگو گفت که یک فایل تنظیمات مشخص را همانند زیر لود کند:

```
python manage.py runserver 127.0.0.1:8001 --settings=mysite.settings
```

**نکته:** هنگامی که مجبورید با چند محیط سروکار داشته باشید که پیکربندی‌های مختلفی لازم دارند، می‌توانید یک فایل تنظیمات مشخص را برای هر محیط ایجاد کنید.

این سرور تنها برای توسعه است و برای استفاده در تولید مناسب نیست. برای توسعه جنگو در یک محیط تولیدی باید آنرا با استفاده از یک وب‌سروری همچون Apache، Gunicorn، یا uWSGI، به‌عنوان یک اپلیکیشن WSGI اجرا کنید یا با استفاده از یک سروری همچون Daphne یا Uvicorn به‌شکل یک اپلیکیشن ASGI اجرا کنید. اطلاعات بیشتر درباره نحوه گسترش جنگو با وب‌سرورهای مختلف را در نشانی زیر ببینید:

<https://docs.djangoproject.com/en/5.0/howto/deployment/wsgi/>

در فصل ۱۷ توضیح می‌دهیم چگونه یک محیط تولید را برای پروژه‌های جنگوی خود ستاپ کنید.

## تنظیمات پروژه

اجازه دهید فایل `settings.py` را باز کنیم و نگاهی به پیکربندی پروژه بیاندازیم. چندین تنظیم وجود دارد که جنگو در این فایل گنجانده است، اما تنها بخشی از همه تنظیمات جنگو در دسترس است. در نشانی زیر می‌توانید همه تنظیمات و مقادیر پیش‌فرض آنها را ببینید:

<https://docs.djangoproject.com/en/5.0/ref/settings/>

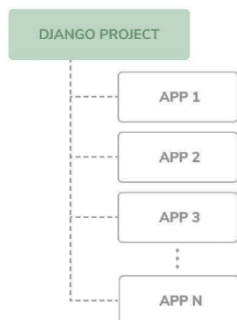
اجازه دهید برخی از تنظیمات پروژه را بازبینی کنیم:

- `DEBUG`، یک `Boolean` است که حالت دیباگ پروژه را روشن و خاموش می‌کند. اگر روی `True` باشد هنگامی که یک استثناء مشخص توسط اپلیکیشن صادر شده باشد صفحات جزئیات خطا را نمایش خواهد داد. هنگامی که به یک محیط تولید می‌روید یا دتان باشد که مجبورید آنرا روی `False` تنظیم کنید. هرگز یک سایت را با روشن بودن `DEBUG` در تولید گسترش ندهید زیرا داده‌های حساس مرتبط با پروژه را افشا خواهید کرد.

- ALLOWED\_HOSTS درحالی که حالت دیباگ فعال است یا وقتی تست‌ها اجرا می‌شود اعمال نمی‌گردد. به محضی که سایت خود را به تولید انتقال دهید و DEBUG را روی False تنظیم کنید مجبورید دامین/هاست خود را به این تنظیم بیافزایید تا برای سرور سایت جنگو مجاز شمرده شود.
  - INSTALLED\_APPS تنظیمی است که برای همه پروژه‌ها مجبورید ویرایش کنید. این تنظیم به جنگو می‌گوید کدام اپلیکیشن برای این سایت فعال است. به حالت پیش‌فرض، جنگو شامل اپلیکیشن‌های زیر است:
    - django.contrib.admin: یک سایت ادمین
    - django.contrib.auth: یک فریمورک تصدیق هویت
    - django.contrib.contenttypes: یک فریمورک برای رسیدگی به انواع محتوا
    - django.contrib.sessions: یک فریمورک نشست
    - django.contrib.messages: یک فریمورک پیام دادن
    - django.contrib.staticfiles: یک فریمورک برای مدیریت فایل‌های استاتیک، همچون CSS، JavaScript و تصاویر.
  - MIDDLEWARE لیستی است حاوی میان‌افزار برای اجرا شدن.
  - ROOT\_URLCONF اشاره به ماژول پایتون می‌کند جایی که الگوهای URL ریشه اپلیکیشن شما تعریف شده باشد.
  - DATABASE یک دیکشنری است که حاوی تنظیماتی برای همه دیتابیس‌ها برای استفاده در پروژه است. همواره باید یک دیتابیس پیش‌فرض موجود باشد. پیکربندی پیش‌فرض از یک دیتابیس SQLite3 استفاده می‌کند.
  - LANGUAGE\_CODE زبان پیش‌فرض کد این سایت جنگو را تعریف می‌کند.
  - USE\_TZ به جنگو می‌گوید پشتیبانی از زمان محلی (timezone) را فعال یا غیرفعال کند. جنگو از زمان و تاریخ‌های timezone-aware پشتیبانی می‌کند. هنگامی که یک پروژه جدید را به‌کمک فرمان مدیریت startproject ایجاد می‌کنید این گزینه روی True تنظیم می‌شود.
- نگران اینکه چیز زیادی درباره آنچه تا اینجا دیدید نفهمید نشوید. در فصل‌های بعدی بیشتر به تنظیمات مختلف جنگو خواهیم پرداخت.

### پروژه‌ها و اپلیکیشن‌ها

در سرتاسر کتاب، بارها و بارها با واژه‌های پروژه و اپلیکیشن برخورد می‌کنید. در جنگو، هر پروژه، یک نصب جنگو یا برخی تنظیمات را دربر می‌گیرد. هر اپلیکیشن، گروهی از مدل‌ها، نماها، الگوها، و URLهاست. اپلیکیشن‌ها با فریمورک تعامل می‌کنند که عملکردهای مشخصی را ارائه دهند و می‌توانند



در پروژه‌های گوناگون، دوباره به کار گرفته شوند. به یک پروژه می‌توانید به‌عنوان وبسایت خود فکر کنید، که حاوی چندین اپلیکیشن است، مانند یک بلاگ، ویکی، یا فروم، که می‌تواند توسط سایر پروژه‌های جنگو نیز استفاده شوند.

شکل مقابل ساختار یک پروژه جنگو را نشان می‌دهد:

### ایجاد یک اپلیکیشن

اجازه دهید نخستین اپلیکیشن جنگوی خود را بسازیم. می‌خواهیم یک اپلیکیشن بلاگ را از آغاز بسازیم. فرمان زیر را در پرامپت پوسته از پوشه ریشه پروژه اجرا کنید:

```
python manage.py startapp blog
```

این ساختار پایه‌ای اپلیکیشن را ایجاد خواهد کرد که مانند این به نظر خواهد رسید:

```
blog/
  __init__.py
  admin.py
  apps.py
  migrations/
    __init__.py
  models.py
  tests.py
  views.py
```

توضیح این فایل‌ها از این قرار است:

- `__init__.py`: این یک فایل خالی است که به پایتون می‌گوید با پوشه `blog` به‌شکل یک ماژول پایتون رفتار کند.
- `admin.py`: این جایی است که ماژول‌ها را رجیستر می‌کنید که در سایت ادمن جنگو درج شوند- استفاده از این سایت اختیاری است.
- `apps.py`: این شامل پیکربندی اصلی اپلیکیشن `blog` است.
- `migrations`: این پوشه حاوی مهاجرت‌های دیتابیس اپلیکیشن است. مهاجرت‌ها به جنگو اجازه پیگیری تغییرات مدل و همگام‌سازی مطابق با دیتابیس را می‌دهد. این پوشه حاوی یک فایل `__init__.py` خالی است.
- `models.py`: این شامل مدل‌های داده‌ای اپلیکیشن شماست؛ همه اپلیکیشن‌های جنگو نیاز دارند یک فایل `models.py` داشته باشند اما این می‌تواند خالی بماند.