# Python Web Frameworks: Django vs FastAPI

A Beginner's Guide to Modern Web Development

# Presentation Overview

- About Me & My Journey
- Web Development Basics
- Python Web Framework Landscape
- Django: The Complete Package
- FastAPI: The Speed Champion
- Live Demo Comparison
- Career Opportunities in the Philippines
- Which Framework Should You Choose?
- Resources & Getting Started
- Q&A

# About Me: From Accounting to Software Engineering

# About Me: From Accounting to Software Engineering

**Background & Education:**

- BS in Accountancy from Laguna University (2012)
- Pursuing Graduate Diploma in Computer Science (UP Open University)
- Started in finance at JPMorgan Chase (Financial Reporting for BlackRock funds & Alternative Investments Bank Loans)

**Career Evolution:**

- Financial Software Support (SS&C Advent)
- Technical Support Engineer (Oracle NetSuite)
- Junior Software Engineer (PDAX) - Cash in/Cash out payment systems for cryptocurrency exchange
- Mid Software Engineer (LawAdvisor Ventures) - Legal technology solutions
- Elixir/Phoenix Developer (AllocatorOne) - Financial technology for venture capital

# About Me: From Accounting to Software Engineering

**Technology Experience:**

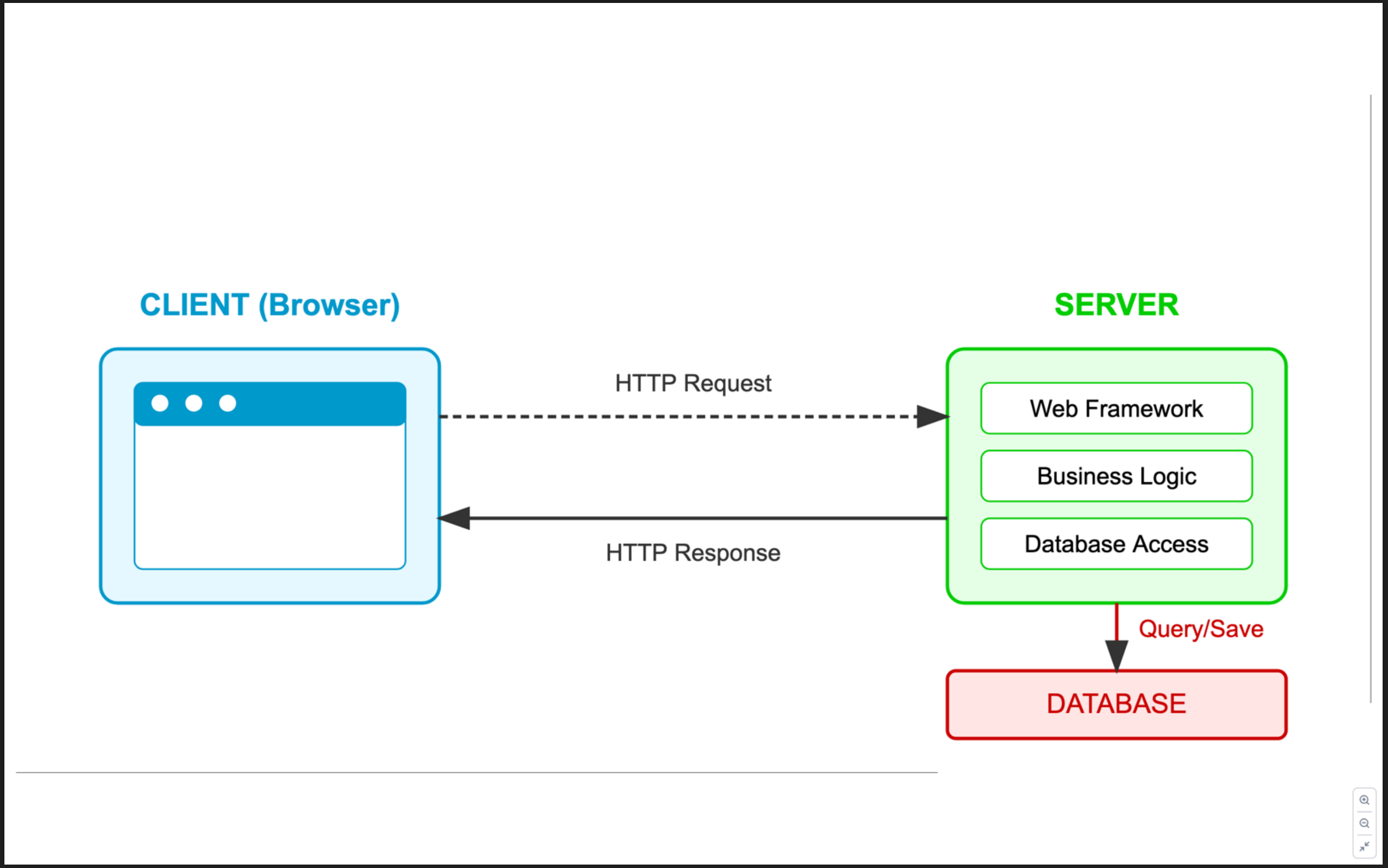
- **Languages:** Elixir, JavaScript, TypeScript, Dart, Python
- **Backend:** Phoenix Framework, Node.js
- **Frontend:** Phoenix LiveView, React, Flutter
- **Databases:** PostgreSQL
- **Cloud/DevOps:** AWS, GCP, Docker
- **Development Environment:** Neovim (previously VS Code)

**Teaching Experience:**

- Conducted 1-week Flutter mobile development training for PCHRD-DOST researchers (2019)

# Web Development Fundamentals

# Web Development Fundamentals



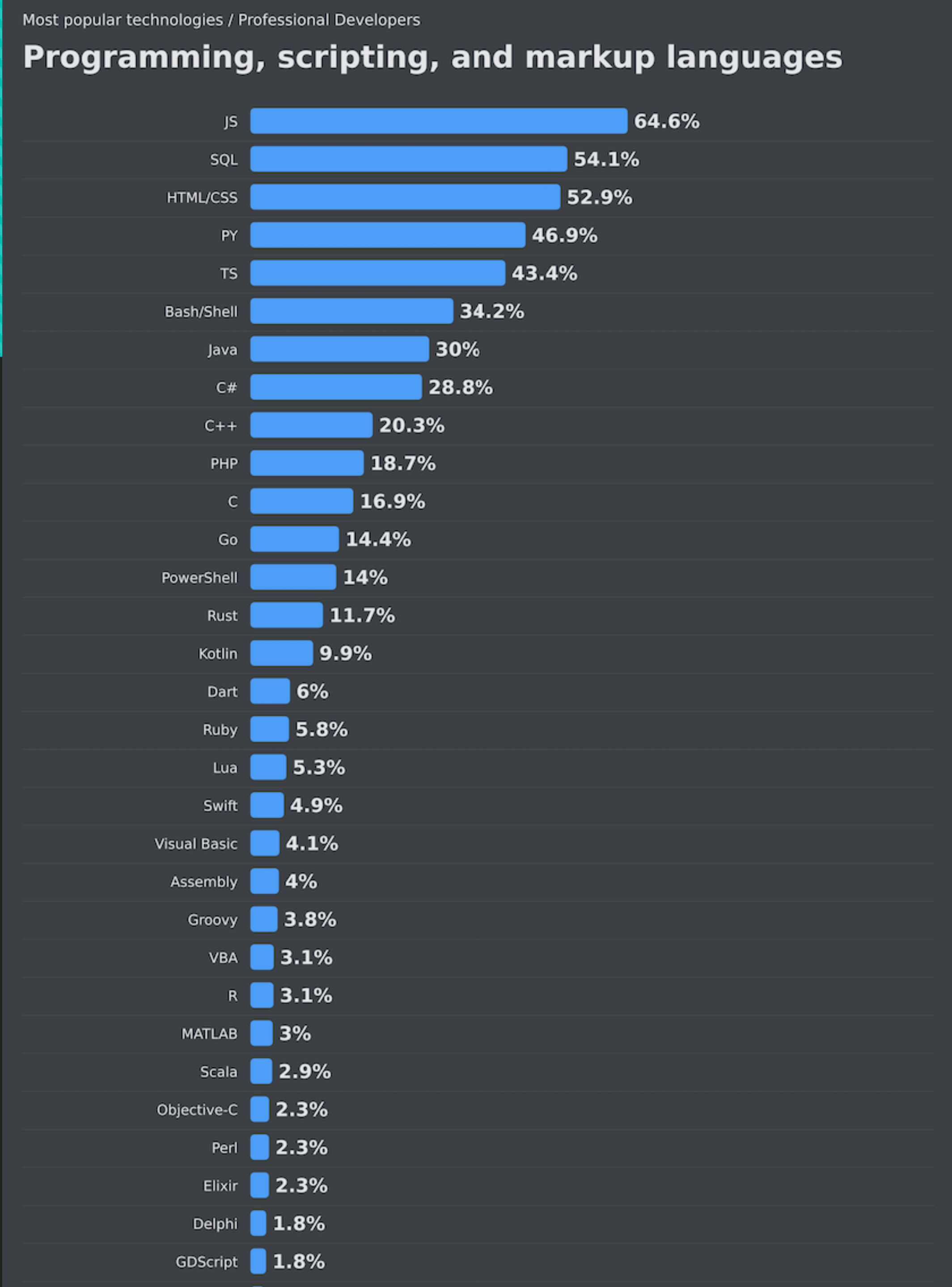
# Web Development Fundamentals

- **Client-Server Architecture:**
  - Client (browser) sends requests to server
  - Server processes requests and returns responses
  - Communication happens via HTTP protocol
- **The Web Stack:**
  - Frontend: What users see and interact with (HTML, CSS, JavaScript)
  - Backend: Server-side logic, database interactions, authentication
  - Database: Stores and manages application data
- **Why We Need Frameworks:**
  - Provides structure and organization for code
  - Handles common tasks (routing, database access, authentication)
  - Implements security best practices automatically
  - Speeds up development with reusable components

# Python Web Framework Landscape

# Python Web Framework Landscape

- Among top 3 languages for web development
- 46.9% of professional developers use Python (Stack Overflow 2024)
- Versatile across data science, AI, and web development
- Popular for readability and extensive libraries Growing ecosystem of web frameworks

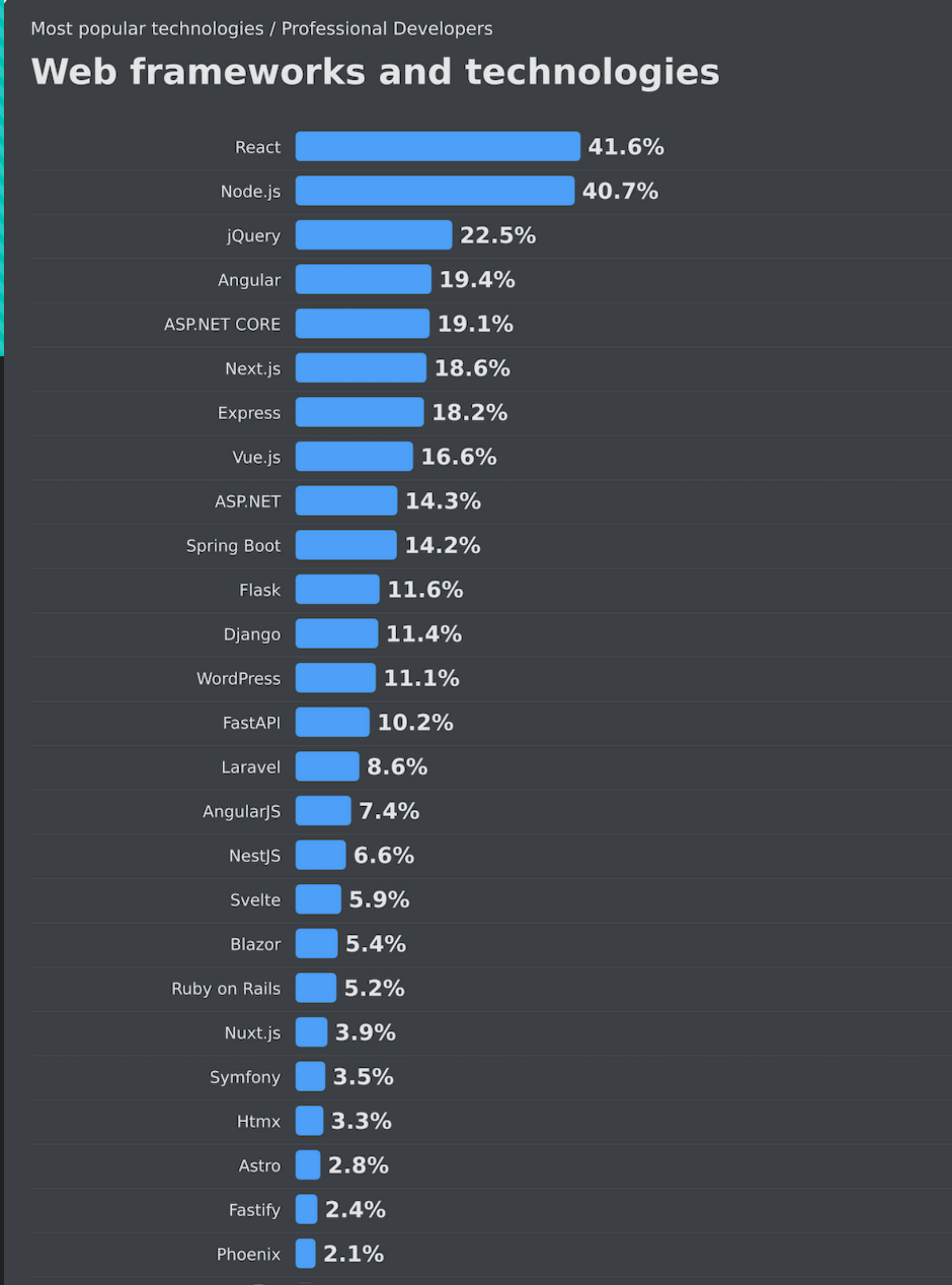


Most popular technologies / Professional Developers

Programming, scripting, and markup languages

# Python Web Framework Landscape

- **Most Popular Python Web Frameworks:**
  - Django (11.4%): Full-featured, "batteries included"
  - FastAPI (10.2%): Modern, high-performance, API-focused
  - Flask (11.6%): Lightweight, flexible microframework
- **Framework Comparison:**
  - Django: Established ecosystem (since 2005), used by Instagram, Pinterest
  - FastAPI: Fastest growing (since 2018), used by Netflix, Microsoft
  - Flask: Minimalist approach, popular for smaller projects

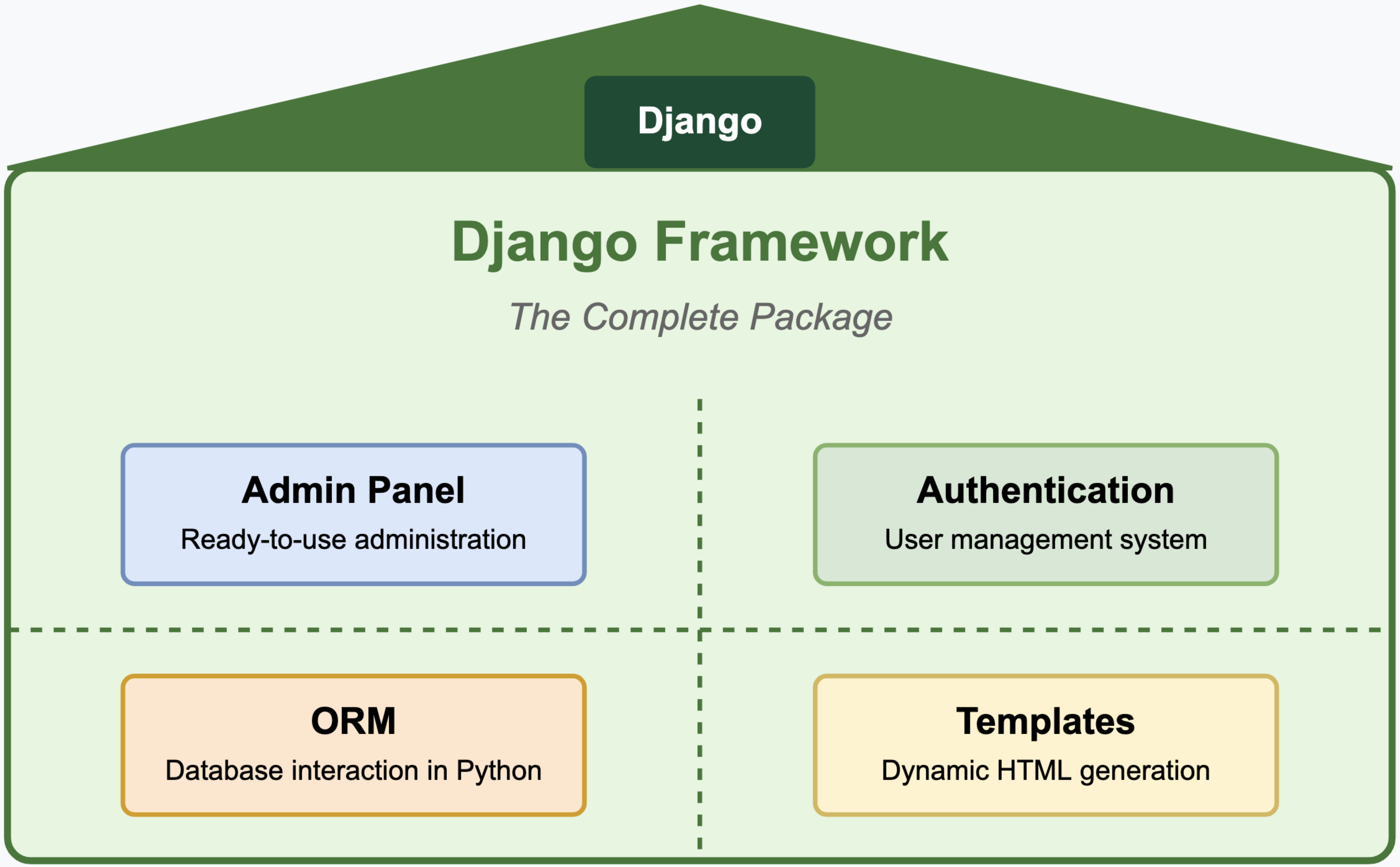


# Django: The Complete Package

# Django: The Complete Package

- **What is Django?**
  - "The web framework for perfectionists with deadlines"
  - Released in 2005, mature and battle-tested
  - Full-featured "batteries included" approach
  - Powers Instagram, Pinterest, Mozilla, Spotify
- **Core Philosophy:**
  - Rapid development
  - Don't Repeat Yourself (DRY)
  - Explicit over implicit
  - Consistency and security by default

# Django: The Complete Package - Visual Metaphor



Django

## Django Framework

*The Complete Package*

**Admin Panel**
Ready-to-use administration

**Authentication**
User management system

**ORM**
Database interaction in Python

**Templates**
Dynamic HTML generation

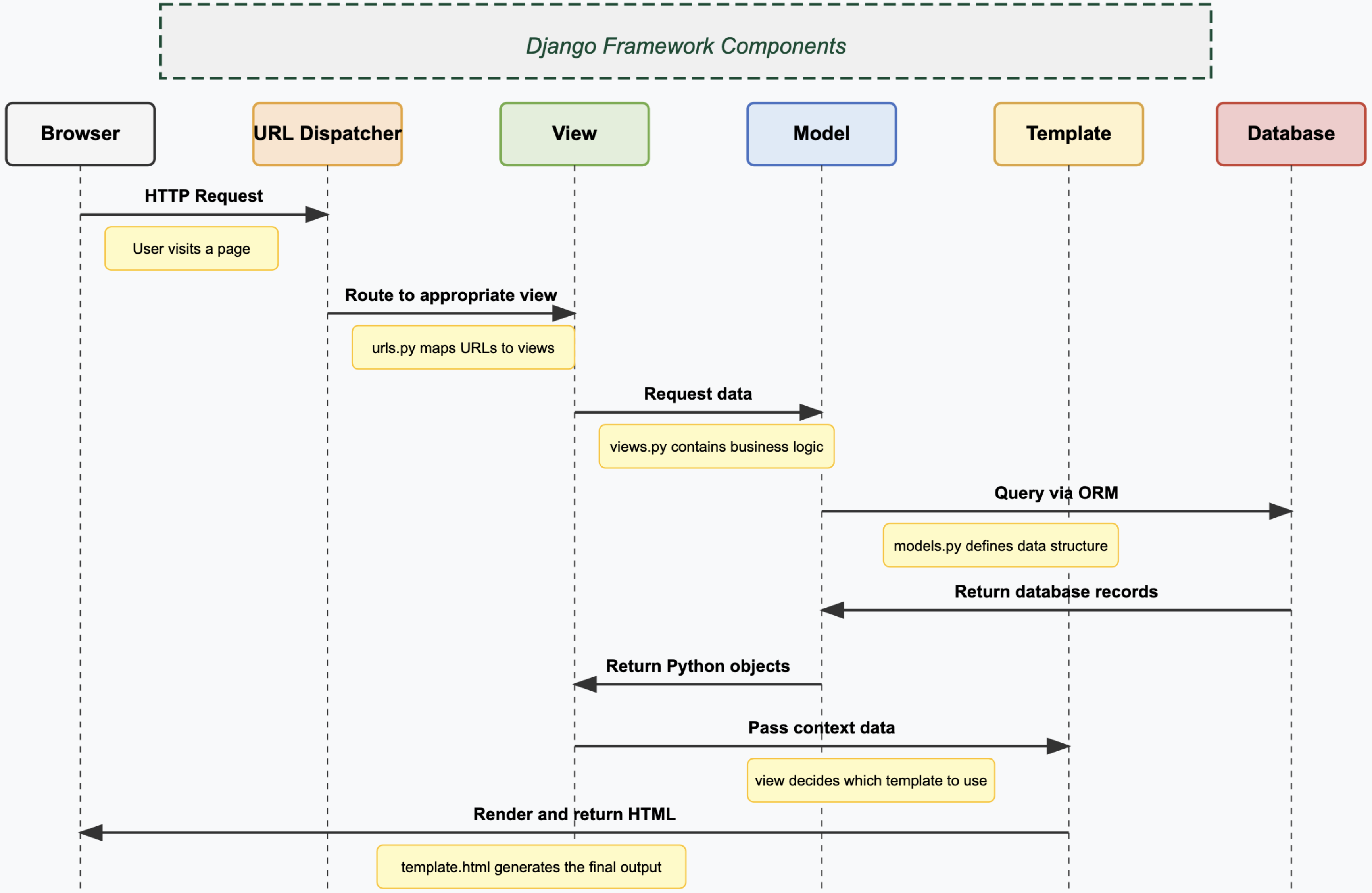
*"Like a fully-furnished house, everything you need is included"*

# Django Architecture

- **Key Components:**
  - **Models:** Define database structure using Python classes
  - **Views:** Handle business logic and control what users see
  - **Templates:** Generate HTML dynamically
  - **Admin Interface:** Auto-generated admin panel for content management
  - **Forms:** Handle user input with validation
  - **ORM (Object-Relational Mapper):** Interact with database using Python
- **Django's MVT Pattern:**
  - Similar to MVC (Model-View-Controller) but with different terminology
  - Model: Data structure and database interactions
  - View: Business logic and request handling
  - Template: Presentation layer (HTML generation)

# Django Request Flow

# Django Request Flow


Django Framework Components
Browser
URL Dispatcher
View
Model
Template
Database
HTTP Request
User visits a page
Route to appropriate view
urls.py maps URLs to views
Request data
views.py contains business logic
Query via ORM
models.py defines data structure
Return database records
Return Python objects
Pass context data
view decides which template to use
Render and return HTML
template.html generates the final output

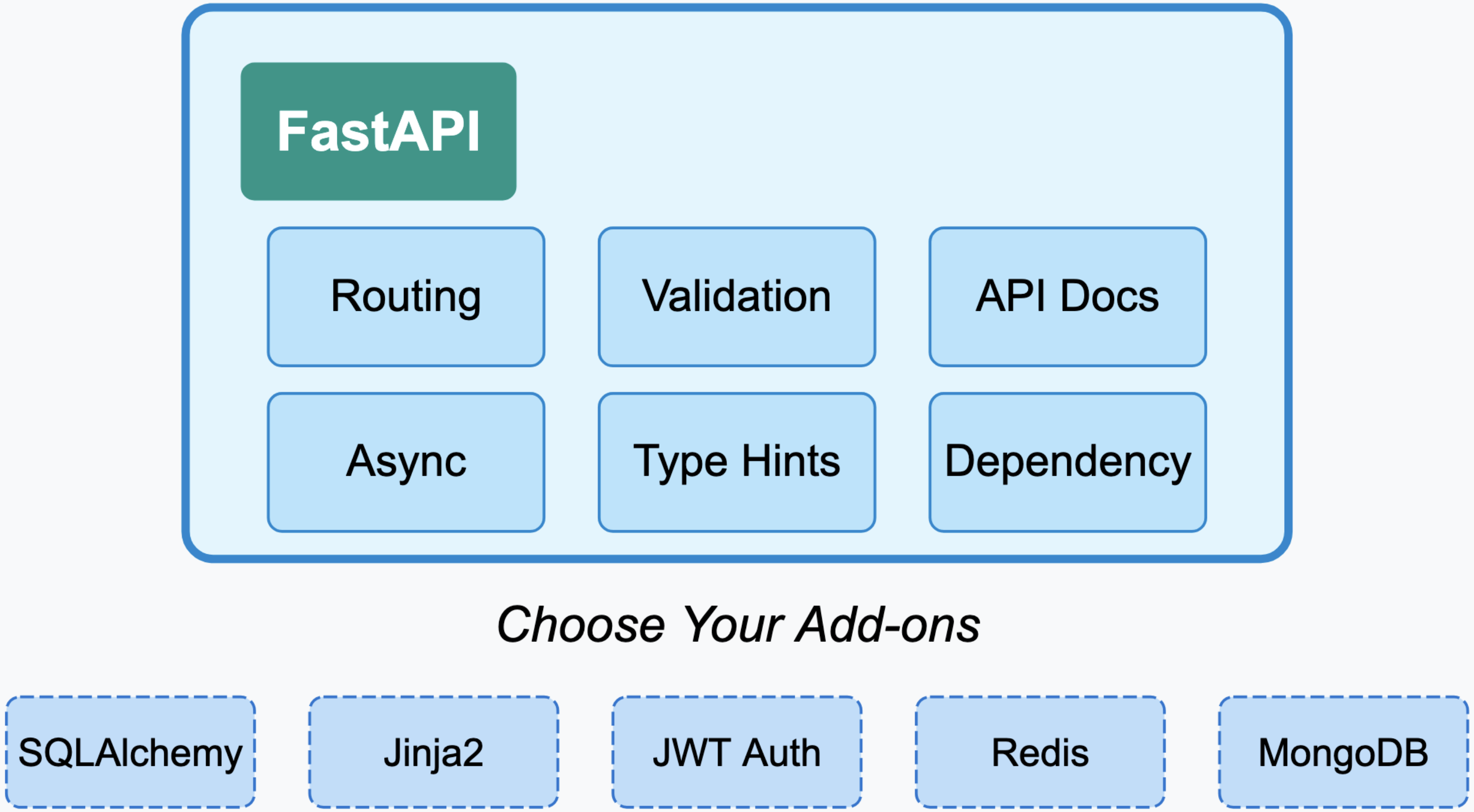
# FastAPI: The Speed Champion

# FastAPI: The Speed Champion

- **What is FastAPI?**
  - Modern, high-performance Python web framework
  - Released in 2018 by Sebastián Ramírez
  - Built for API development with automatic docs
  - Leverages Python type hints for validation
- **Core Philosophy:**
  - Performance: One of the fastest Python frameworks
  - Easy to use: Intuitive, minimal code required
  - Standards-based: Built on OpenAPI & JSON Schema
  - Modern Python: Uses Python 3.6+ type hints

# FastAPI: The Customizable Toolkit

# FastAPI: The Customizable Toolkit



FastAPI

Routing

Validation

API Docs

Async

Type Hints

Dependency

*Choose Your Add-ons*

SQLAlchemy

Jinja2

JWT Auth

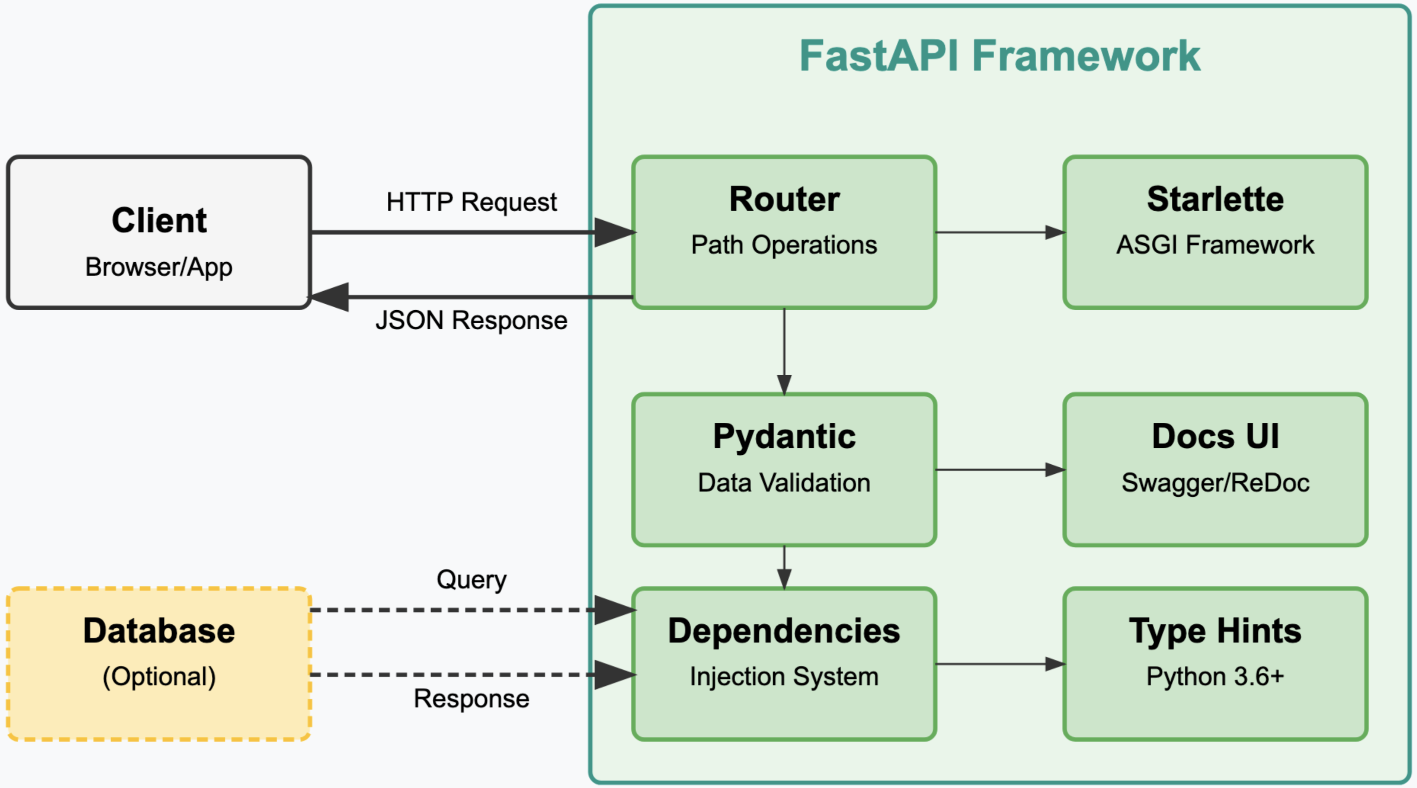
Redis

MongoDB

*"Like a customizable toolkit, add only what you need"*

# FastAPI Architecture

# FastAPI Architecture



FastAPI Framework

Client
Browser/App

HTTP Request

JSON Response

Router
Path Operations

Starlette
ASGI Framework

Pydantic
Data Validation

Docs UI
Swagger/ReDoc

Database
(Optional)

Query

Response

Dependencies
Injection System

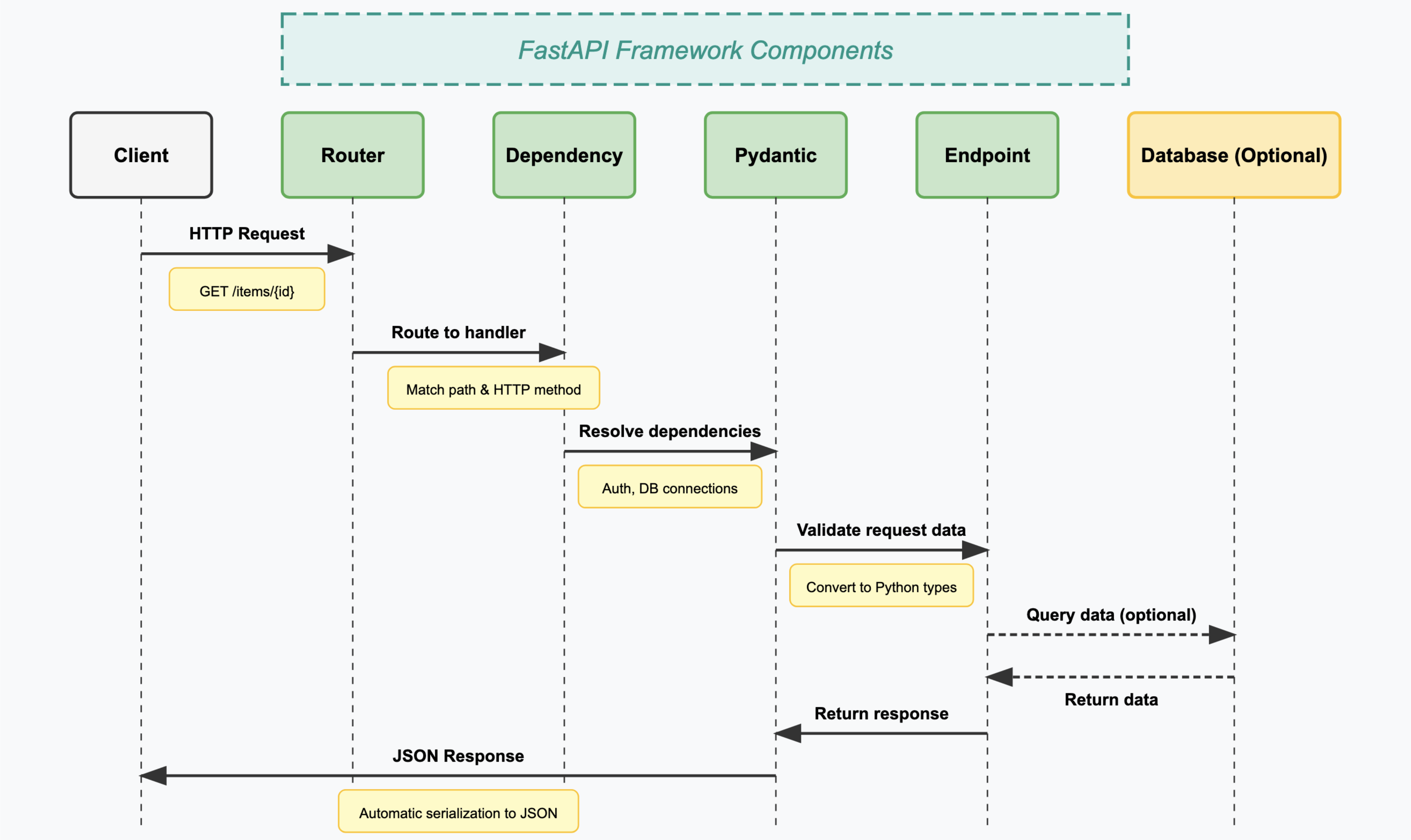
Type Hints
Python 3.6+

**Key Differences from Django:**

- No built-in ORM - Can use any database library (SQLAlchemy, databases, etc.)
- API-first design - JSON is the primary response format, not HTML templates

# FastAPI Request Flow

# FastAPI Request Flow



*FastAPI Framework Components*

Client

Router

Dependency

Pydantic

Endpoint

Database (Optional)

HTTP Request

GET /items/{id}

Route to handler

Match path & HTTP method

Resolve dependencies

Auth, DB connections

Validate request data

Convert to Python types

Query data (optional)

Return data

Return response

JSON Response

Automatic serialization to JSON

# Live Demo Comparison

# Career Opportunities in the Philippines

# Python Web Development Careers in the Philippines

- Example Salary Ranges (From Job Listings):
  - Entry Level: ₱32,000 - ₱35,000/month (ASG Platform)
  - Mid-Level: ₱35,000 - ₱50,000/month (AI Fullstack)
  - Experienced: ₱50,000 - ₱80,000/month (Django & React)
  - Senior Positions: ₱120,000+ opportunities exist
- Most In-Demand Skills:
  - Django: Enterprise, E-commerce, CMS
  - FastAPI: AI/ML applications, High-performance APIs
  - Common: PostgreSQL, Docker, Git, React

# Python Framework Job Market Insights

- Companies Hiring:
  - Multinational: Accenture, IBM, CITCO, Amdocs
  - Local: PayMaya, GCash, Various startups
- Entry Points:
  - Fresh Graduate Programs (₱25K-₱35K)
  - 2-Year Residency Programs (₱32K-₱35K → ₱100K)
  - Junior Developer Roles (₱30K-₱45K)
- Career Growth Factors:
  - Framework specialization (+20-30% salary premium)
  - DevOps knowledge (Docker, CI/CD)
  - Full-stack capabilities (React, Vue.js)

# Job Market Data Sources (JobStreet Listings)

- **Django Jobs:**
  - Full-Stack Developer (Django & React) – Remote - Roko Home Services
  - Full Stack Engineer - CITCO International Support Services
- **Entry-Level Opportunities:**
  - Software Engineer - Fresh Graduates Accepted! - Amdocs Philippines Inc.
  - (WFH) Software Developers - 2 Year Residency Program - ASG Platform
- **AI & Python Jobs:**
  - Artificial Intelligence Developers - Offshore Outsource Operations, Inc.
- **Salary Research:**
  - ₱120,000+ Salary Range for FastAPI Engineers

# Which Framework Should You Choose?

# Which Framework Should You Choose?

- **Choose Django When:**
  - Building complete websites with user accounts, admin panels
  - Need built-in features like authentication, admin panel, forms
  - Creating content-heavy sites (blogs, e-commerce, social media)
  - Want a "batteries-included" approach (like a fully furnished house)
- **Choose FastAPI When:**
  - Building APIs for mobile apps or frontend frameworks
  - Performance is critical
  - Need automatic API documentation
  - Prefer a lightweight, modern approach (like a custom tiny home)

# Recommendations for Beginners

- **For Complete Beginners:** Start with Django (more structure, tutorials)
- **For AI/Data Science Interest:** Consider FastAPI (popular for ML APIs)
- The Good News:
  - Both are in high demand in the Philippines
  - Skills from one transfer to the other
  - You can (and should) eventually learn both!
- **Most Important:** Just start building something!

# Resources & Getting Started

# Resources & Getting Started

- Official Documentation:
  - Django: https://docs.djangoproject.com/
  - FastAPI: https://fastapi.tiangolo.com/
- Beginner-Friendly Tutorials:
  - Django: Django Girls Tutorial,
  - Mozilla Django Tutorial FastAPI: FastAPI's Interactive Docs,
  - TestDriven.io guides
  - **freeCodeCamp.org Certifications:** (Highly Recommended!)
- Practice Projects:
  - Todo App (like our demo!)
  - Personal Blog
  - Simple API for a mobile app
- GitHub Repository: https://github.com/jeryldev/python-web-frameworks-2025

# Q&A