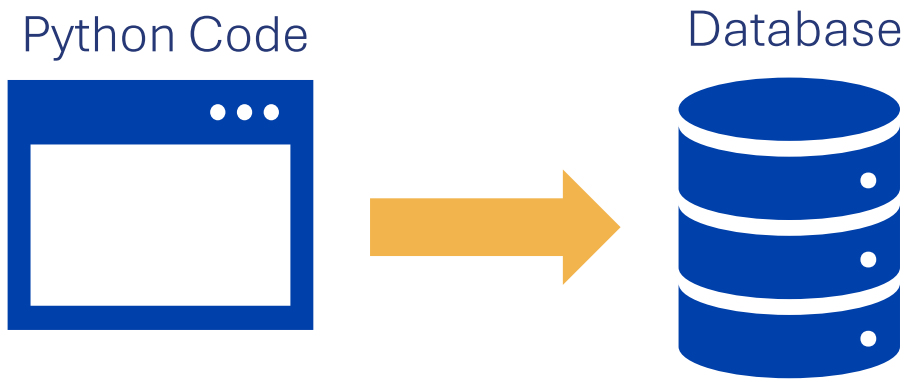
Object-Relational Mapping

The **ORM** allows developers to interact with the database using **Python** code, abstracting away the complexities of **SQL**.

Key features included:

- Model Definitions
- Migrations
- Migrate

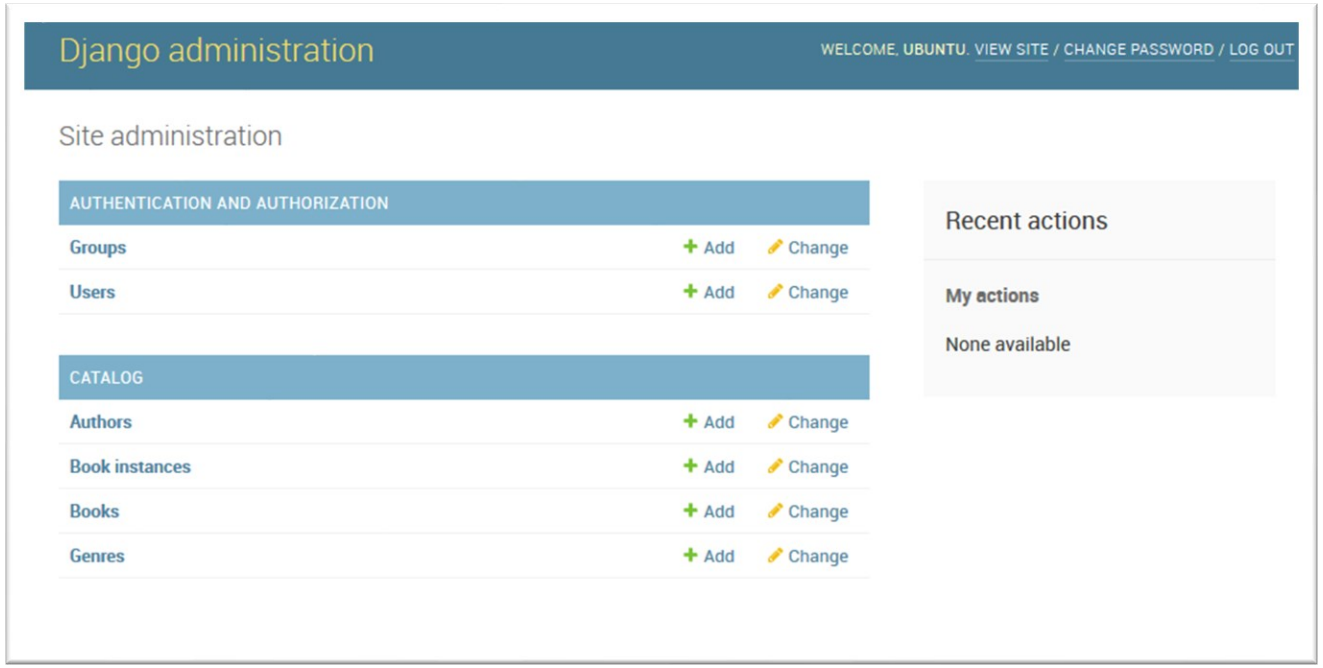


Admin Interface

The ORM allows developers to interact with the database using Python code, abstracting away the complexities of SQL.

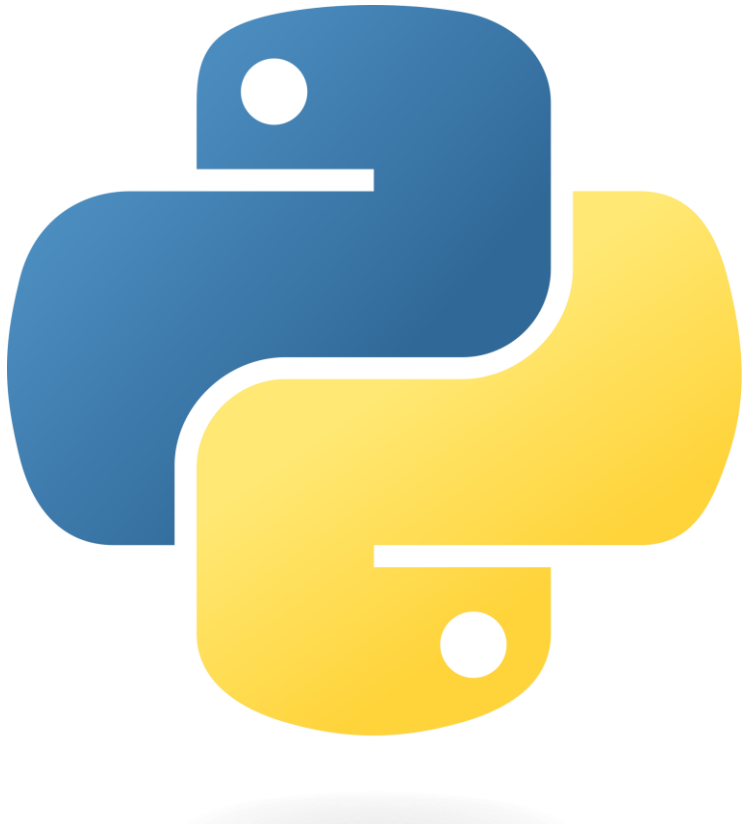
Key features included:

- Built-in Admin
- Customization
- Secure
- Data Management



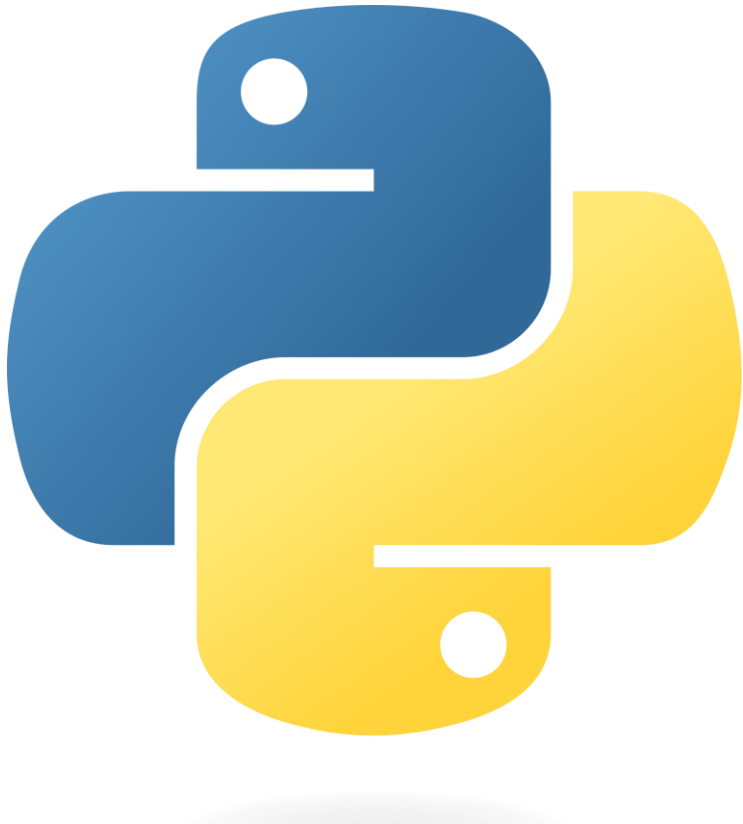
SQL

Structured Query Language, is a standard programming language used to manage and manipulate databases. It allows you to create, read, update, and delete data within a database. SQL is essential for querying and maintaining databases, making it a fundamental tool for database management and analysis.



Database

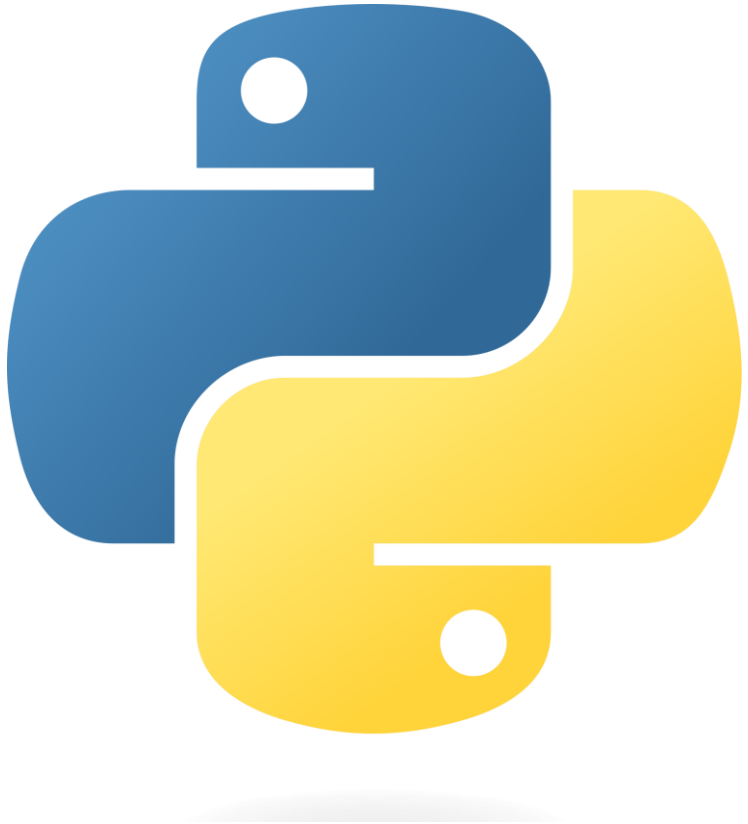
A database is an organized collection of data that is stored and accessed electronically. It allows you to store, retrieve, and manage large amounts of information efficiently.



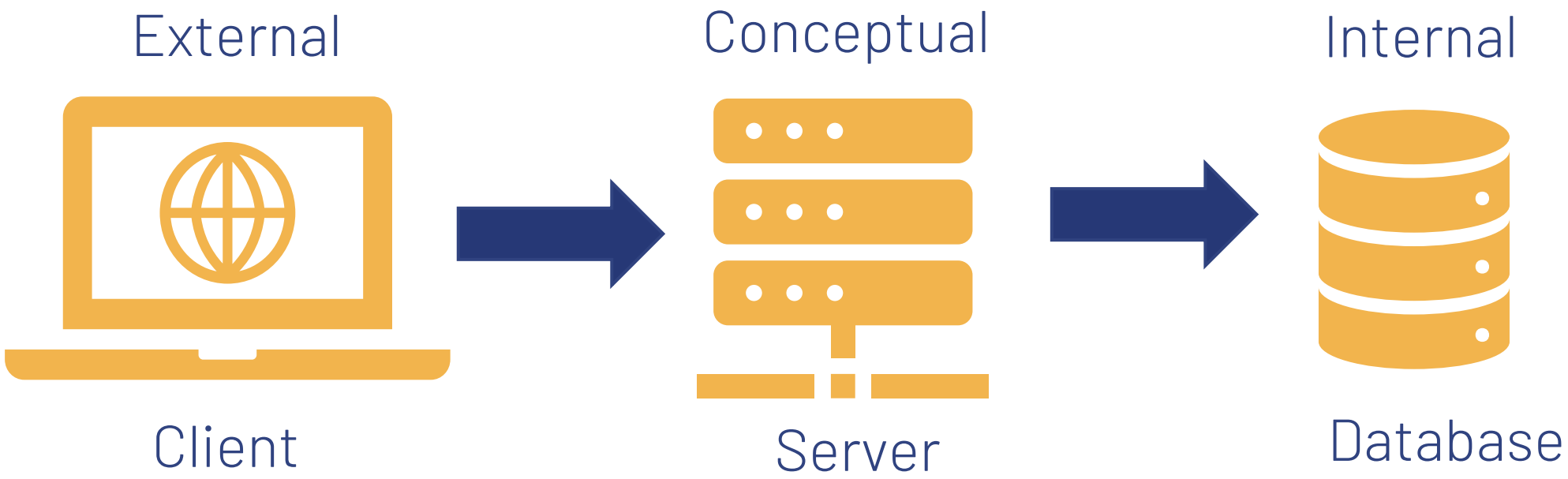
General Level

The database is implemented through three general levels. These levels are:

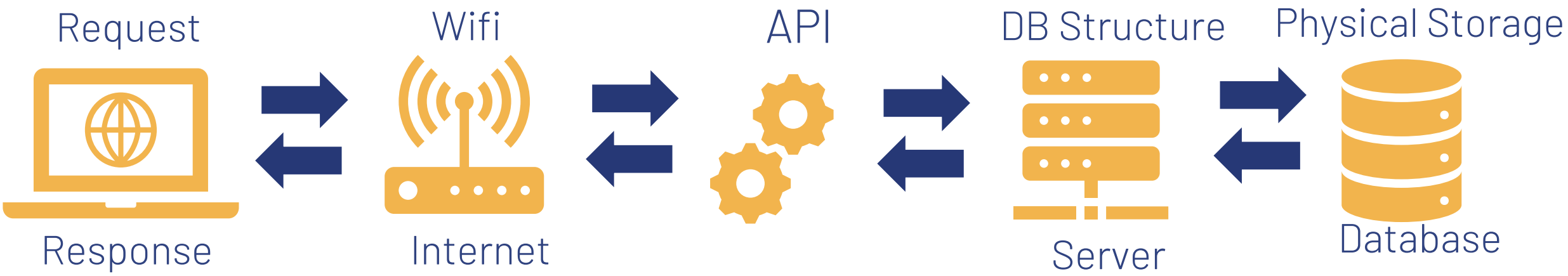
- **External Level** - Describes various user views
- **Conceptual Level** - Structure and constraints for the entire database
- **Internal Level -** Actual PHYSICAL storage structure and access paths.



General Level

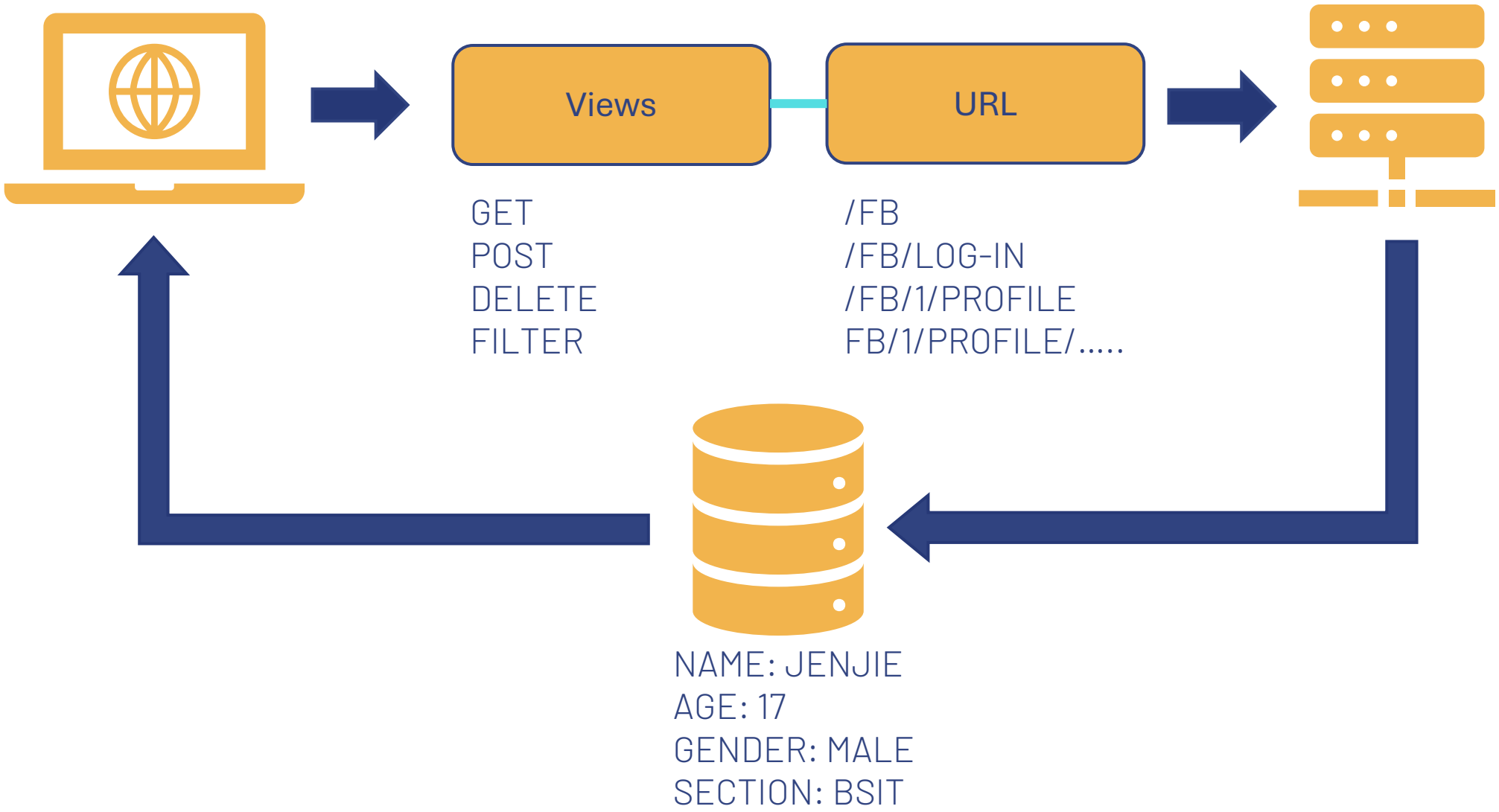


API Workflow



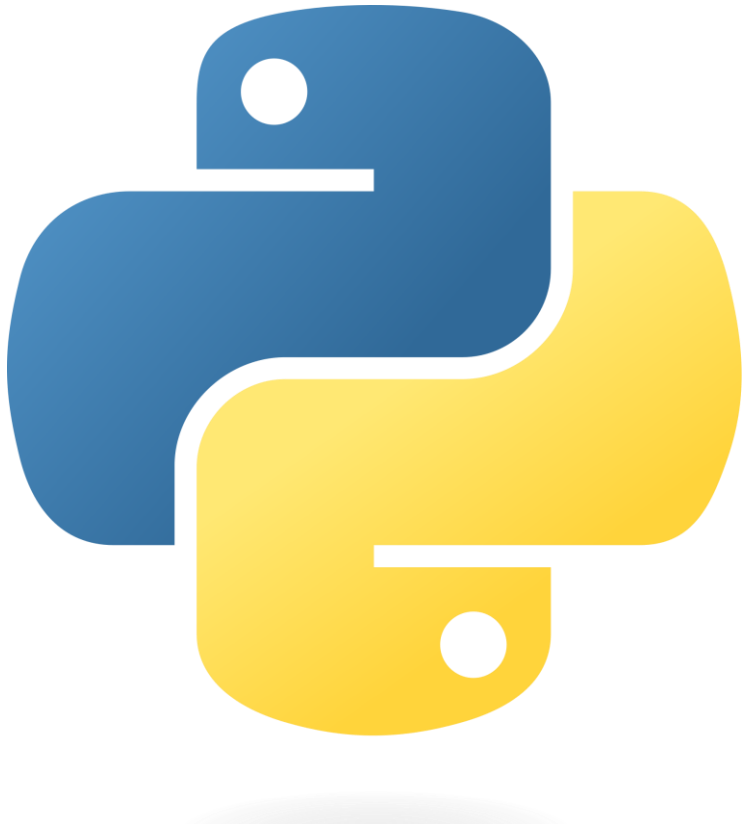
Lesson 3

Django API



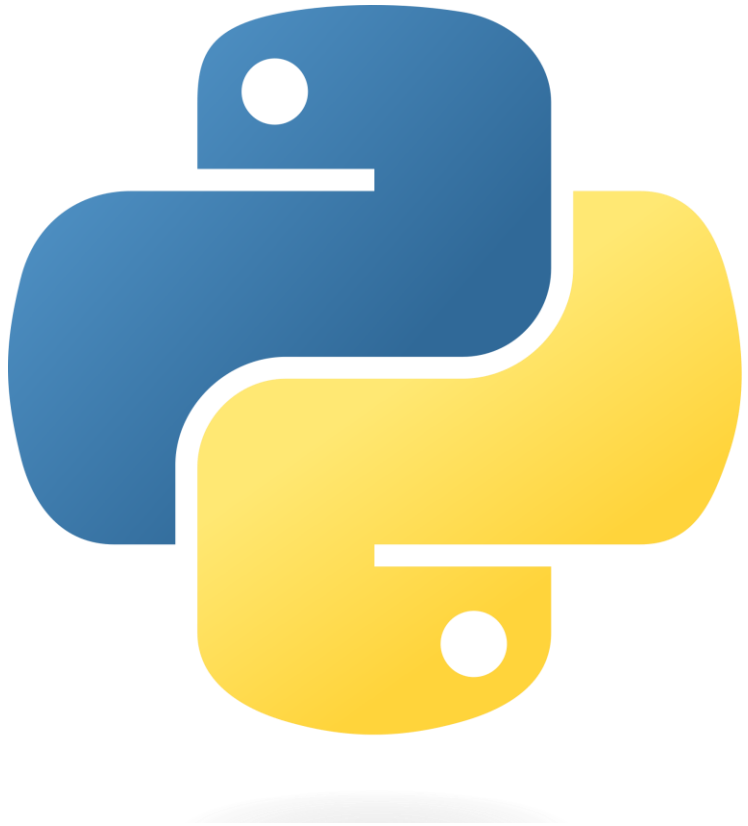
Django Database

Django databases are configured to store and manage your data. It supports multiple databases like SQLite, PostgreSQL, MySQL, and Oracle. You define your models (database tables) in Python, and Django takes care of generating the SQL for database schema creation, data querying, and more.



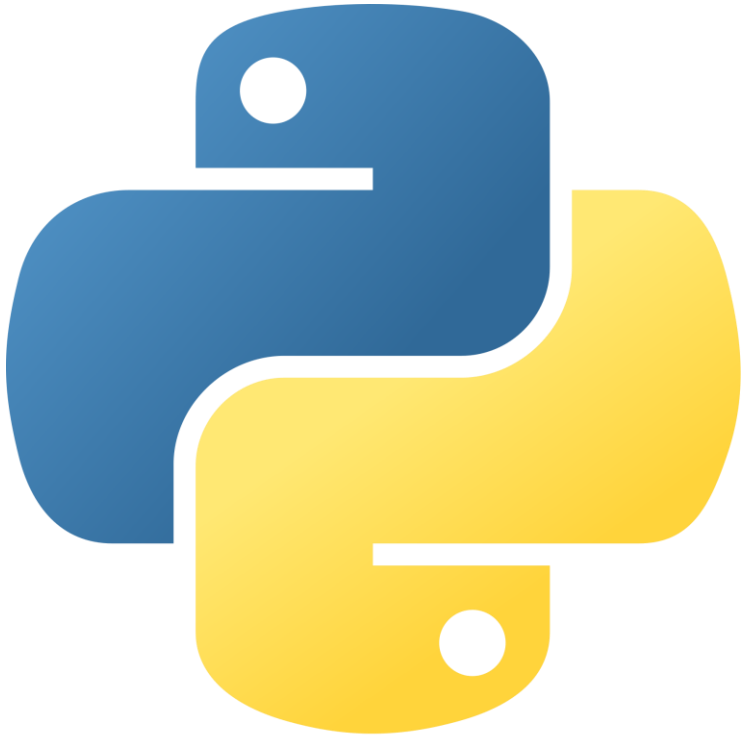
Generic Views

Django's generic views are ready-made views that handle common web development tasks, reducing the need for repetitive code. They make it easy to build views for listing, creating, updating, and deleting objects.



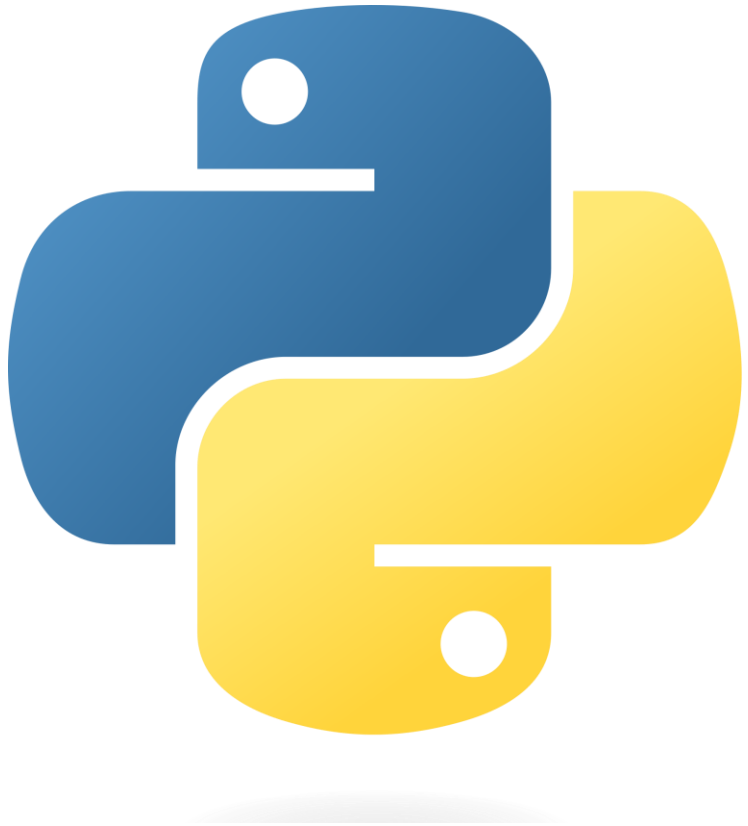
Benefits

- **Pre-built Views:** Django generic views provide ready-to-use, common view patterns.
- **Customizable:** They can be easily customized to fit specific requirements.
- **Time-saving:** Using them reduces the amount of code you need to write.



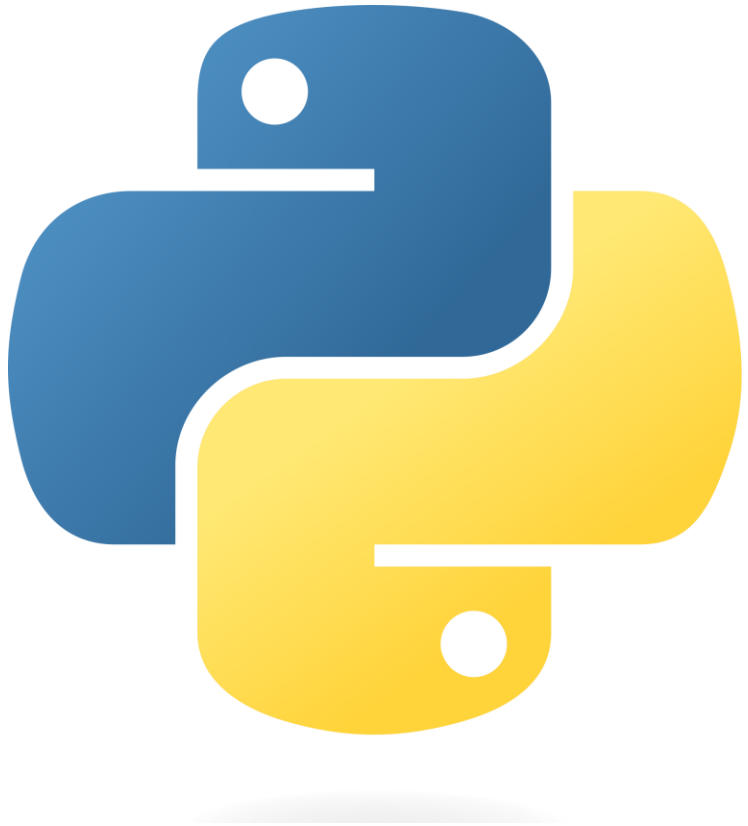
Benefits

- **Rapid Development:** They speed up the development process.
- **Code Reusability:** You can reuse them across different projects or apps.
- **Consistency:** They promote consistent coding practices across your application.



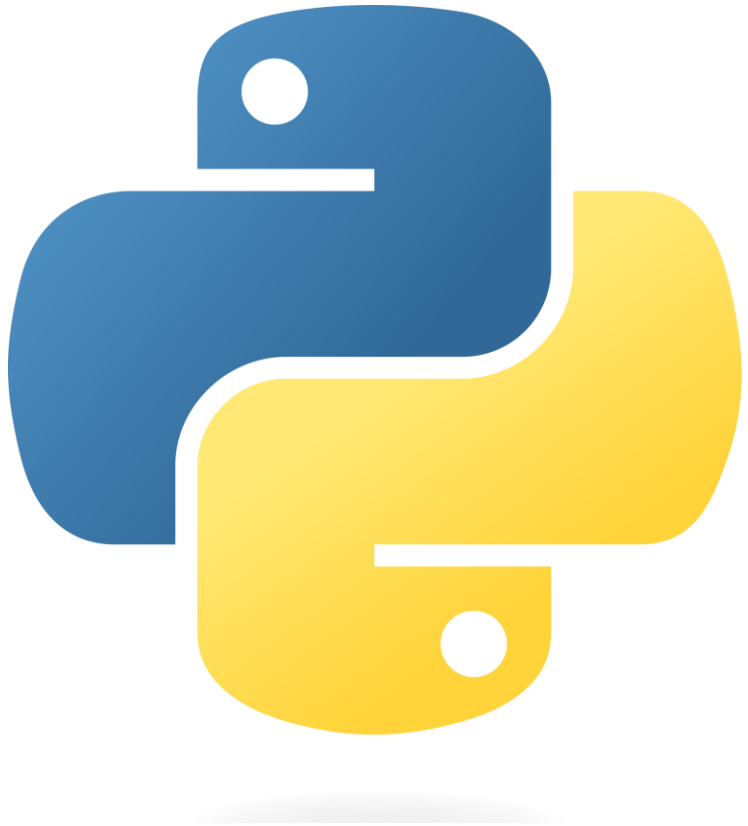
Benefits

- **Rapid Development:** They speed up the development process.
- **Code Reusability:** You can reuse them across different projects or apps.
- **Consistency:** They promote consistent coding practices across your application.



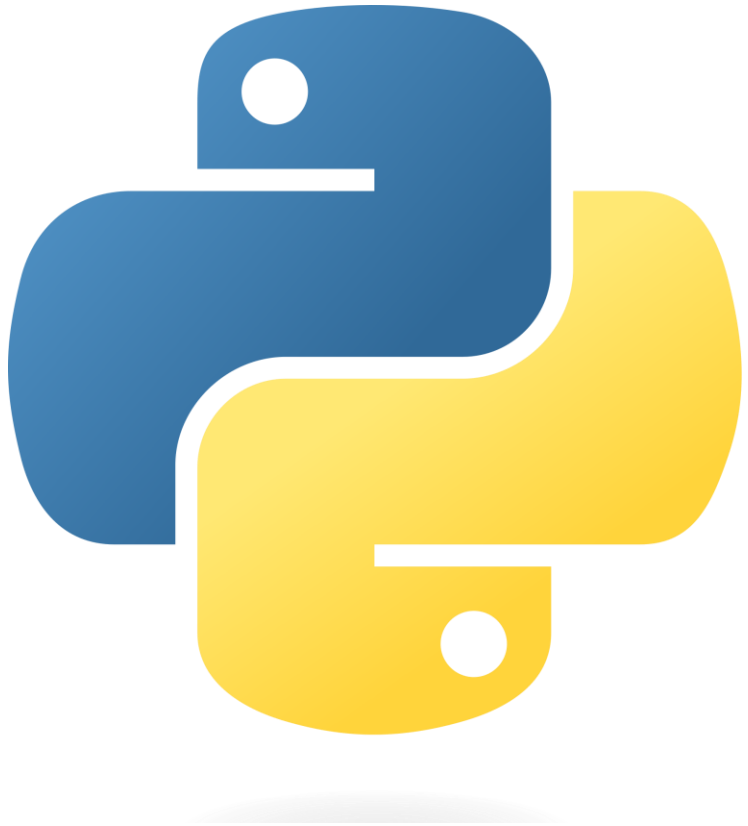
TemplateView

TemplateView is used to render a template without needing a model or queryset in **Django**. It's great for displaying static pages or simple views that don’t require complex data processing.



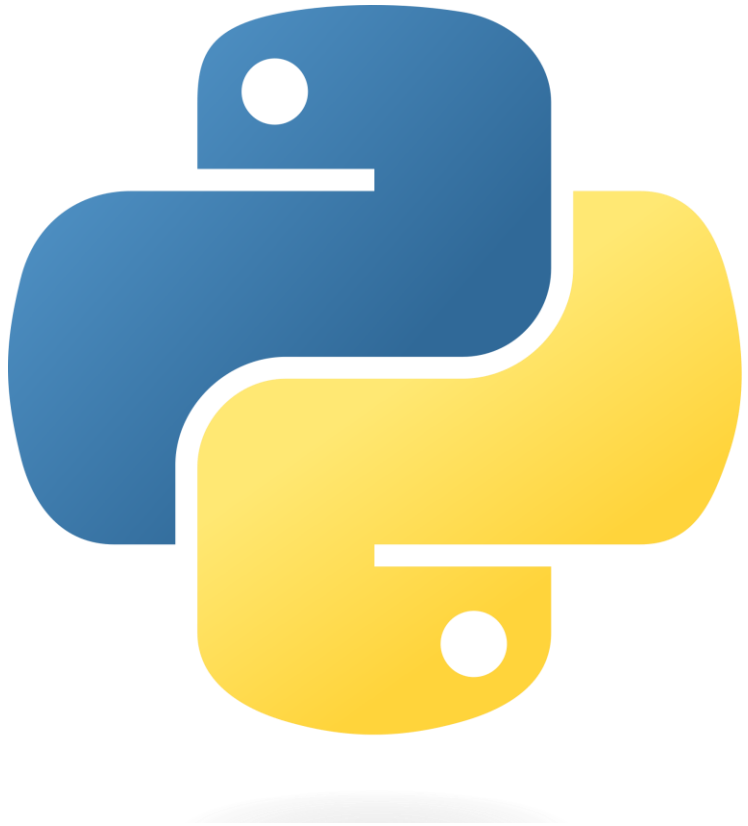
ListView

ListView displays a list of objects from a model or queryset in **Django**. It supports pagination, filtering, and custom queryset definition out of the box, making it easy to present multiple records efficiently.



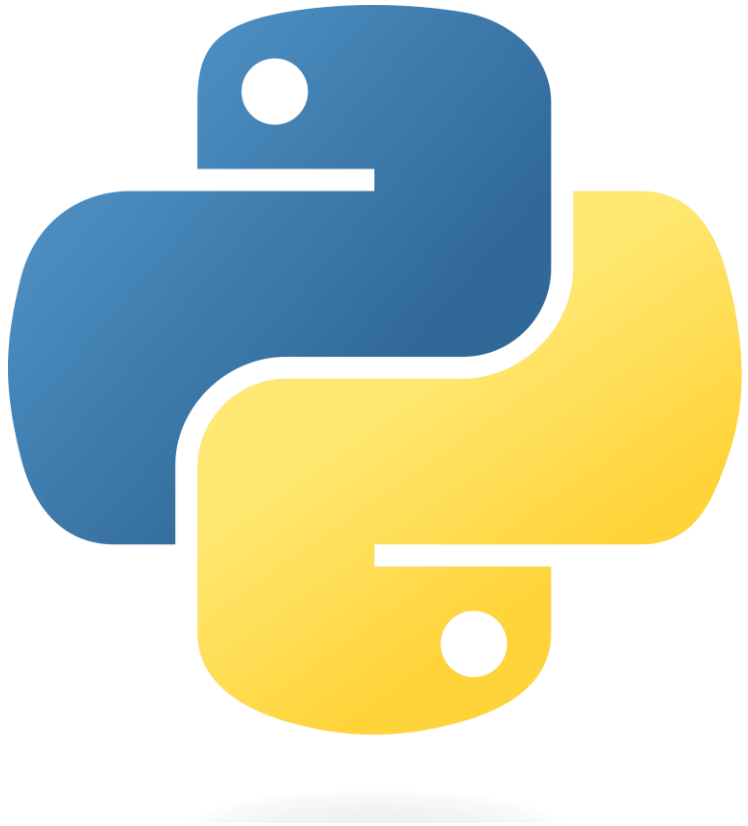
DetailView

DetailView displays a single object's details in **Django**. It retrieves an object based on the primary key and presents its fields, allowing users to see the full data of a specific record.



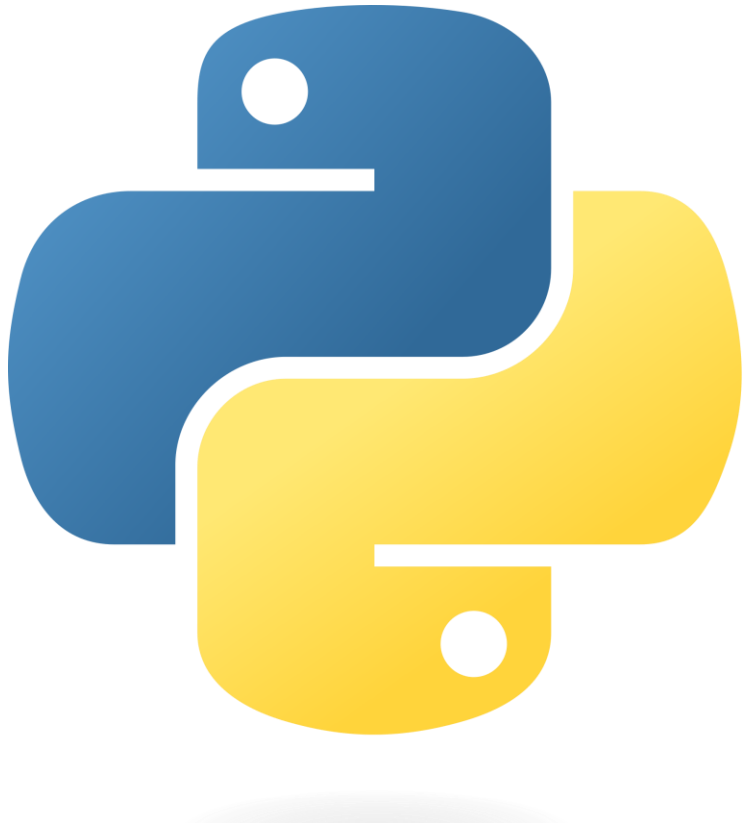
CreateView

CreateView handles the creation of new objects in **Django**. It displays a form for inputting object data and processes the form submission, simplifying the creation process of new database entries.



UpdateView

UpdateView handles editing existing objects in **Django**. It displays a form pre-filled with the object's current data and processes the changes upon form submission, making object updates smooth and efficient.



DeleteView

DeleteView handles the deletion of objects in **Django**. It displays a confirmation page and processes the deletion upon user confirmation, streamlining the removal of objects from the database.