**What is FastAPI?**

**FastAPI** is a **modern, high-performance web framework** for building **APIs** with **Python 3.7+**.  
It is based on:

* **Starlette** (for web handling, requests, responses)
* **Pydantic** (for data validation and serialization)

**🔹 Key Features**

1. **Fast** 🚀
   * One of the fastest Python frameworks (performance close to Node.js & Go).
   * Built on **ASGI (Asynchronous Server Gateway Interface)** for concurrency.
2. **Easy to Use** ✨
   * Very little boilerplate code.
   * You can build production-ready APIs in just a few lines.
3. **Automatic Documentation** 📑
   * Provides interactive API docs out of the box:
     + Swagger UI → /docs
     + ReDoc → /redoc
4. **Data Validation** ✅
   * Uses **Pydantic models** for request and response validation.
   * Automatically checks data types and formats (e.g., email, date).
5. **Async Support** 🔄
   * Supports both normal (def) and asynchronous (async def) functions.
   * Great for applications that handle many requests (like APIs, chat apps, IoT).

**🔹 When to Use FastAPI?**

* Building **REST APIs** or **GraphQL APIs**
* Backend for **web apps** or **mobile apps**
* **Machine Learning model deployment**
* **Microservices**
* Projects where **speed** and **scalability** matter

**🔹 Example (Hello World)**

from fastapi import FastAPI

app = FastAPI()

@app.get("/")

def read\_root():

return {"message": "Hello, FastAPI!"}

Run:

uvicorn main:app --reload

Open: 👉 <http://127.0.0.1:8000>  
Docs: 👉 <http://127.0.0.1:8000/docs>

✅ In short: **FastAPI = Fast, simple, and reliable way to build APIs with Python**.

11:25 fastapi

python -m venv venv

venv\Scripts\Activate

pip install fastapi uvicorn[standard]

in fastapi folder/main.py

from fastapi import FastAPI

app = FastAPI()

@app.get("/")

def read\_root():

return {"message": "Hello, FastAPI!"}

to run this main.py

uvicorn main:app –reload

pydantic model

to hit post request

so this eases the process of fastAPI to compare and validate the item object

**GITHUB**

git add .

git commit -m "Initial commit"

git branch -M main

git push -u origin main

30/09/25

Middleware

Crud operations

Working with Jinja templates and forms

Exception handling in FastAPI

Handling image uploads

Creating an HTML form

Uploading images in FastAPI

pip install fastapi uvicorn sqlalchemy passlib[bcrypt]