

Tere Learning Roadmap

1. Jahan Dikkat Aayi Aur Kya Seekhna Hai

- **SQLAlchemy aur ORM:**
 - Tables ko classes banane ka concept.
 - Session management (`session`, `add`, `commit`).
- **FastAPI:**
 - Dependency injection (`Depends`), type hints, endpoints.
- **Code Structuring:**
 - Multiple files aur layers ka use.
- **Python Modern Features:**
 - Type hints (`: int`, `-> str`), OOP.
- **Logic Building aur Debugging:**
 - Khud se logic sochna, errors samajhna.

2. Kami Kahan Hai Aur Kaise Improve Karna Hai

- **Conceptual Clarity:** Core ideas ko samajhna.
- **Practice:** Hands-on projects banane ka experience.
- **Connecting Dots:** Python, SQLAlchemy, FastAPI ko jodna.
- **Debugging:** Errors ko khud solve karne ki skill.

3. Resources Aur Learning Plan

Python Basics Aur Modern Features

- **Kya Seekhna Hai:** Type hints, OOP, decorators.
- **Resources:**
 - Real Python: [Type Hints](#)
 - Real Python: [OOP in Python](#)
 - YouTube: Corey Schafer - [Python OOP Tutorials](#)
 - Book: *Fluent Python* by Luciano Ramalho (Chapter 1-5).
- **Practice:**
 - Ek `BankAccount` class bana (deposit, withdraw, balance).
 - Example:

```
class BankAccount:
    def __init__(self, owner: str, balance: float = 0.0):
        self.owner = owner
        self.balance = balance

    def deposit(self, amount: float) -> None:
        self.balance += amount

    def withdraw(self, amount: float) -> bool:
        if amount <= self.balance:
```

```
        self.balance -= amount
        return True
    return False
```

SQLAlchemy Aur ORM

- **Kya Seekhna Hai:** Tables, session, queries, relationships.
- **Resources:**
 - SQLAlchemy Docs: [ORM Quickstart](#)
 - YouTube: Tech With Tim - [SQLAlchemy Tutorial](#)
 - Miguel Grinberg: [SQLAlchemy Tutorial](#)
- **Practice:**
 - Blog app bana (User, Post tables).
 - Example:

```
from sqlalchemy import Column, Integer, String, ForeignKey
from sqlalchemy.orm import relationship, Session
from sqlalchemy.ext.declarative import declarative_base

Base = declarative_base()

class User(Base):
    __tablename__ = "users"
    id = Column(Integer, primary_key=True)
    name = Column(String)
    posts = relationship("Post", back_populates="user")

class Post(Base):
    __tablename__ = "posts"
    id = Column(Integer, primary_key=True)
    title = Column(String)
    content = Column(String)
    user_id = Column(Integer, ForeignKey("users.id"))
    user = relationship("User", back_populates="posts")
```

FastAPI

- **Kya Seekhna Hai:** Endpoints, Depends, request/response.
- **Resources:**
 - FastAPI Docs: [First Steps](#)
 - FastAPI Docs: [Dependencies](#)
 - YouTube: ArjanCodes - [FastAPI Tutorial](#)
 - FreeCodeCamp: [FastAPI Crash Course](#)
- **Practice:**
 - TODO app bana (CRUD endpoints).
 - Example:

```

from fastapi import FastAPI, Depends
from sqlalchemy.orm import Session

app = FastAPI()

def get_db():
    db = Session(...) # Assume setup
    try:
        yield db
    finally:
        db.close()

@app.post("/todos/")
def create_todo(title: str, db: Session = Depends(get_db)):
    # Logic here
    pass

```

PostgreSQL

- **Kya Seekhna Hai:** Tables, queries, foreign keys.
- **Resources:**
 - W3Schools: [SQL Tutorial](#)
 - YouTube: FreeCodeCamp - [PostgreSQL Tutorial](#)
 - PostgreSQL Docs: [Tutorial](#)
- **Practice:**
 - Terminal mein tables bana, queries likh.
 - Example: `CREATE TABLE users (id SERIAL PRIMARY KEY, name VARCHAR(50));`

Hands-On Projects

- **Ideas:**
 1. Expense Tracker (FastAPI + SQLAlchemy).
 2. E-commerce API (products, cart, orders).
 3. Chat App Backend (users, messages).
- **Approach:**
 - Logic paper pe likh.
 - Code kar, debug kar.

Debugging Aur Logic Building

- **Resources:**
 - YouTube: Corey Schafer - [Debugging in Python](#)
 - Real Python: [Debugging](#)
- **Practice:**
 - Errors print kar, debugger use kar.

4. Daily Plan

- 1-2 hrs: Topic padh.

- 1-2 hrs: Code likh.
- 30 min: Debug ya logic soch.

5. Extra Tips

- **GitHub:** Code save kar - github.com
- **Stack Overflow:** Help le - stackoverflow.com
- **AI Tools:** Mujhse (Grok) ya ChatGPT se pooch.