2️⃣

Phase 2

🎯 Phase 2: Backend with Python (3 Months = ±12 Weeks)

Goal: Learn FastAPI/Django, ORM, JWT Authentication, Pagination, Search, Unit Testing, and build a full **Task Management API (Trello-lite)**.

📅 Roadmap Phase 2 — Weekly Breakdown

📌 Month 1 → Python Backend Basics + FastAPI/Django

🗓️ Week 1: Python Backend Fundamentals

1. Review essential Python concepts:
    - Functions, modules, packages
    - Classes, objects, inheritance
    - Error handling
2. Learn to use virtual environments (`venv`) and dependency management (`pip`).
3. Understand Python backend project structure and folder organization.
4. Practice writing clean, modular code.

🗓️ Week 2: Intro to FastAPI or Django

1. Learn the purpose of backend frameworks and why FastAPI/Django are used.

2. Understand how routing works (GET, POST, PUT, DELETE).
3. Learn how to structure endpoints using the chosen framework.
4. Learn how the framework handles JSON responses and validations.
5. Understand how middleware works at a high level.

🗓️ Week 3: ORM (SQLAlchemy or Django ORM)

1. Learn what ORM is and why it's used.
2. Understand how to define models and their fields.
3. Study one-to-many and many-to-many relationships.
4. Learn migrations and database schema management.
5. Practice interacting with PostgreSQL using ORM queries.

🗓️ Week 4: CRUD API + Database Integration

1. Learn to connect your framework to the PostgreSQL database.
2. Understand how to implement full CRUD operations using ORM models.
3. Study data validation and request/response schemas (DTO/serializer concepts).
4. Learn proper error handling and HTTP status codes.
5. Understand how to structure backend layers (controller → service → repository).

📌 Month 2 → Authentication, Pagination, Search, Testing

🗓️ Week 5: JWT Authentication

1. Understand token-based authentication and JWT structure.

2. Learn password hashing best practices.
3. Study login, registration, and token generation flows.
4. Understand authentication middleware and protecting routes.
5. Learn about token expiration and refresh strategies.

🗓️ Week 6: Authorization + User Assignment System

1. Learn the difference between authentication and authorization.
2. Understand role-based access (admin/user) if needed.
3. Study how to restrict certain actions (e.g., only creators can edit).
4. Learn how to manage user–task/project assignment logic.
5. Handle permission errors and unauthorized access.

🗓️ Week 7: Pagination & Search

1. Understand why pagination is needed in APIs.
2. Learn common pagination patterns (limit, offset).
3. Study querying with filters (e.g., task title, status).
4. Learn how to implement full-text-like search using ORM.
5. Understand sorting (ascending/descending).

🗓️ Week 8: Unit Testing (pytest or unittest)

1. Learn the purpose of backend testing.
2. Understand the structure of test files and test cases.
3. Study how to test services, repositories, and API endpoints.
4. Learn how to mock database operations.
5. Practice writing tests for authentication and permissions.

📌 Month 3 → Build "Task Management API" (Trello-lite)

🗓️ Week 9: Project Setup

1. Create a clean project structure using best practices.
2. Design the ERD for:
    - User
    - Project
    - Task
    - Optional: Task assignment table
3. Configure database connections and migrations.
4. Set up GitHub repository and initial documentation.

🗓️ Week 10: Core Features Implementation

1. Implement full Project CRUD with validation.
2. Implement full Task CRUD linked to Project.
3. Add logic for task status, priority, and due date.
4. Implement assignment features (assign/unassign users).
5. Handle all edge cases (missing project, invalid data, etc.).

🗓️ Week 11: Authentication & Advanced API Features

1. Integrate full JWT authentication across all routes.
2. Add filtering:
    - Tasks by status
    - Tasks by assigned user

3. Implement pagination for lists of tasks and projects.
4. Add search functionality for project/task names.
5. Improve error messages and validation rules.

🗓️ Week 12: Testing, Documentation, Deployment (Optional)

1. Write unit tests for major functionalities (CRUD, auth, filtering).
2. Produce API documentation (Swagger/OpenAPI or manual docs).
3. Perform manual testing using Postman/Insomnia.
4. Optional: Deploy to free cloud platforms:
    - Render
    - Railway
    - Fly.io
5. Final review and polishing of the Task Management API.

🏁 Short Summary

- **Month 1:** Learn FastAPI/Django, ORM, CRUD fundamentals.
- **Month 2:** Master JWT Auth, authorization, pagination, search, and testing.
- **Month 3:** Build a complete **Task Management API** from scratch using industry standards.