OJT Project Design Template — Filled

Student Name: Shivam Chaudhary

Roll No: **240410700100**

Year & Section: 2nd year

Project Title : Sales Insights Backend

Project Type: Application Developer

Stack / Framework: FastAPI

1. Problem Understanding

1.1 Problem Statement (in your own words)

Businesses collect sales data every day, but most of it remains unorganized and unused. There is no simple backend system that can store, process, analyze, and generate insights from this sales data. My project aims to build a backend service that provides clean APIs to manage sales data and generate insights for better decision-making.

1.2 Why this problem matters?

- Helps business owners understand their sales performance.
- Saves time by avoiding manual Excel reports.
- Makes it easier for frontend developers to build dashboards.
- Good learning project for backend API development.

1.3 Inputs and Expected Outputs

Inputs	Process	Expected Outputs
Sales data (product, quantity, price, date)	CRUD operations	Add / update / view sales records
Filters (date, product)	Simple aggregation	Total sales, revenue, best product
API requests	Data validation	Clean JSON responses

2. Functional Scope

2.1 Core Features (Must-Haves)

1. Add new sales record
2. Get all sales records
3. Update/Delete sales record
4. API to calculate:
    - Total revenue
    - Total items sold
    - Best-selling product
5. Simple date filter (optional)

2.2 Stretch Goals (If time remains)

- CSV upload for bulk sales
- Authentication (simple JWT)
- Monthly revenue graph data

- Deployment on Render/Railway

2.3 Tools & Libraries

- FastAPI
- SQLAlchemy
- Pydantic
- SQL
- Uvicorn
- PyTest (for simple tests)

---

3. System & Design Thinking

3.1 App Flow (Simple Pipeline)

```
Client → FastAPI Routes → Controller → Database → JSON Output
```

3.2 Key Data Structures / Algorithms

- Sales model (id, product, quantity, price, date)
- Aggregation logic (sum, count, max)
- List filtering for date or product

3.3 Testing Methods

- Test API with Postman
- Unit tests for CRUD

- Verify revenue calculation manually

---

4. Timeline & Milestones (4 Weeks)

Week	Planned Deliverables	Mentor Checkpoint
W1	Understand problem, design DB, create FastAPI project	☐
W2	Build CRUD APIs	☐
W3	Build insights API + fix bugs	☐
W4	Documentation + final demo	☐

---

5. Risks & Dependencies

5.1 Hardest Part for Me

- Aggregation logic
- Database relationships (if added later)

5.2 Required Mentor Help

- Review DB design
- Suggest improvements for insight logic

---

6. Evaluation Readiness

6.1 How I Will Prove the Project Works

- Screenshots of API responses
- Postman collection
- GitHub link
- Demo video (1–2 mins)
- Simple documentation

6.2 Success Metrics

- CRUD APIs working properly
- Correct revenue and product insights
- API response within 200–300ms
- Clean code & modular structure

---

7. Responsibilities

Task	Student Name	Mentor Notes
Task 1: Project Setup	shivam	☐
Task 2: Database Design	shivam	☐
Task 3: CRUD API Development	shivam	☐
Task 4: Insights API	shivam	☐
Task 5: Testing & Documentation	shivam	☐

Signatures (Student):

Mentor Approval:

Date: