

FastAPI File Management using PostgreSQL, MinIO, and Caching

By: Shrabana Paul

Mentor: Suprava Das

Institution: IDEAS-TIH, ISI Kolkata

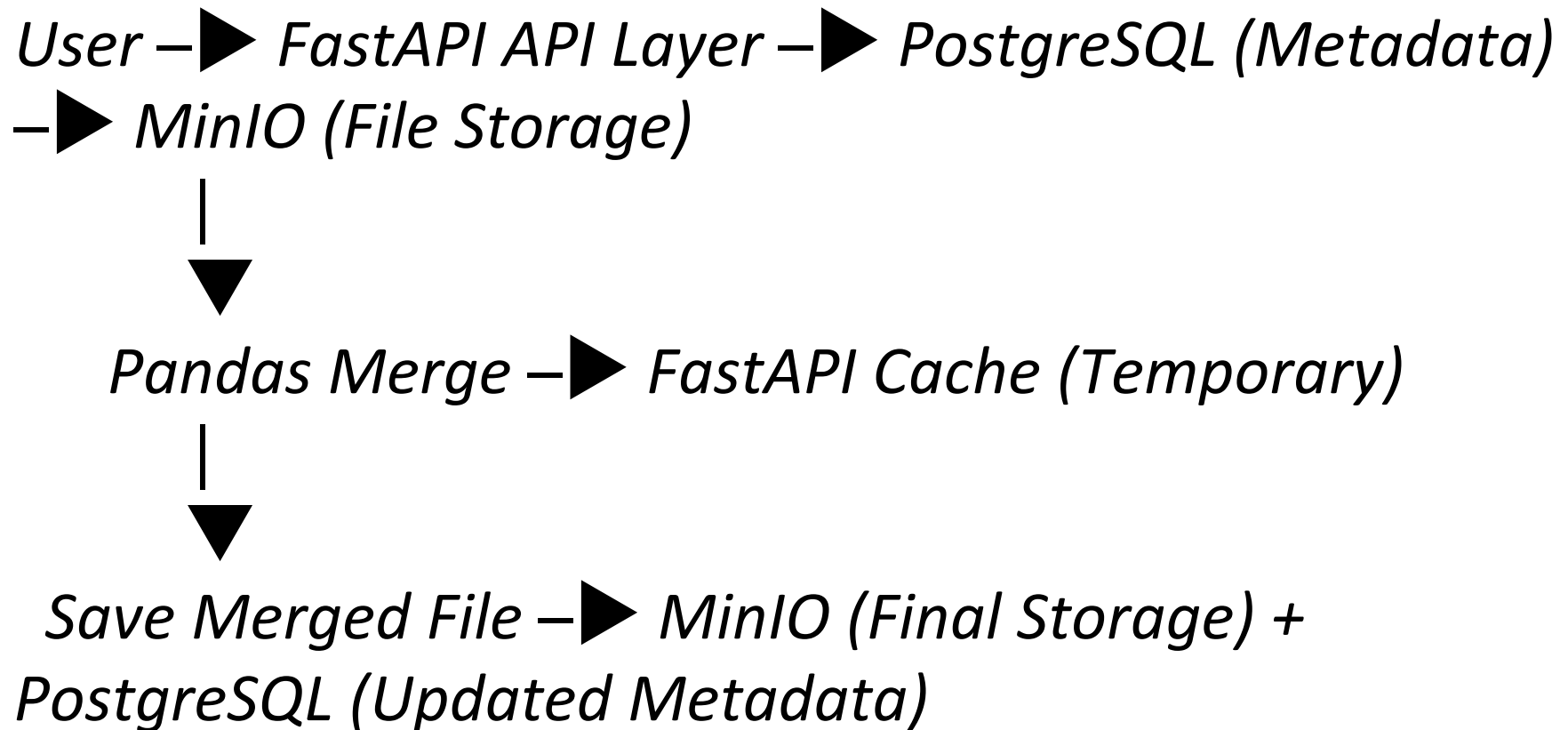
Internship Duration: Aug–Oct 2025

Abstract

- *Developed a scalable backend File Management System using FastAPI.*
- *Integrated PostgreSQL for metadata and MinIO for object storage.*
- *Implemented caching with FastAPI-Cache2 for performance.*
- *Supported real-time merging and retrieval of datasets.*
- *Ensures automation, reliability, and modular design.*

System Architecture / Workflow

FASTAPI FILE MANAGEMENT SYSTEM



Introduction

- *Organizations face challenges in handling large datasets.*
- *FastAPI + PostgreSQL + MinIO + Caching = Efficient backend for data ingestion.*
- *Simplifies ETL workflows with modular, microservice-based design.*

Project Objectives

- 1. Develop API-driven file management system.*
- 2. Integrate FastAPI, PostgreSQL, and MinIO.*
- 3. Implement caching for merged datasets.*
- 4. Automate data merging & validation.*
- 5. Ensure scalability & reproducibility.*

Methodology and tools used

Tool	Purpose
FastAPI	Backend framework
PostgreSQL	Metadata storage
MinIO	Object storage
Pandas	Data merging
SQLAlchemy	ORM for DB operations
FastAPI-Cache2	Temporary caching

Execution of the program

The screenshot displays the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure: Servers (1) > PostgreSQL 17 > Databases (2) > file_management > Schemas (1) > public > Tables (1). The main query editor shows the following SQL query:

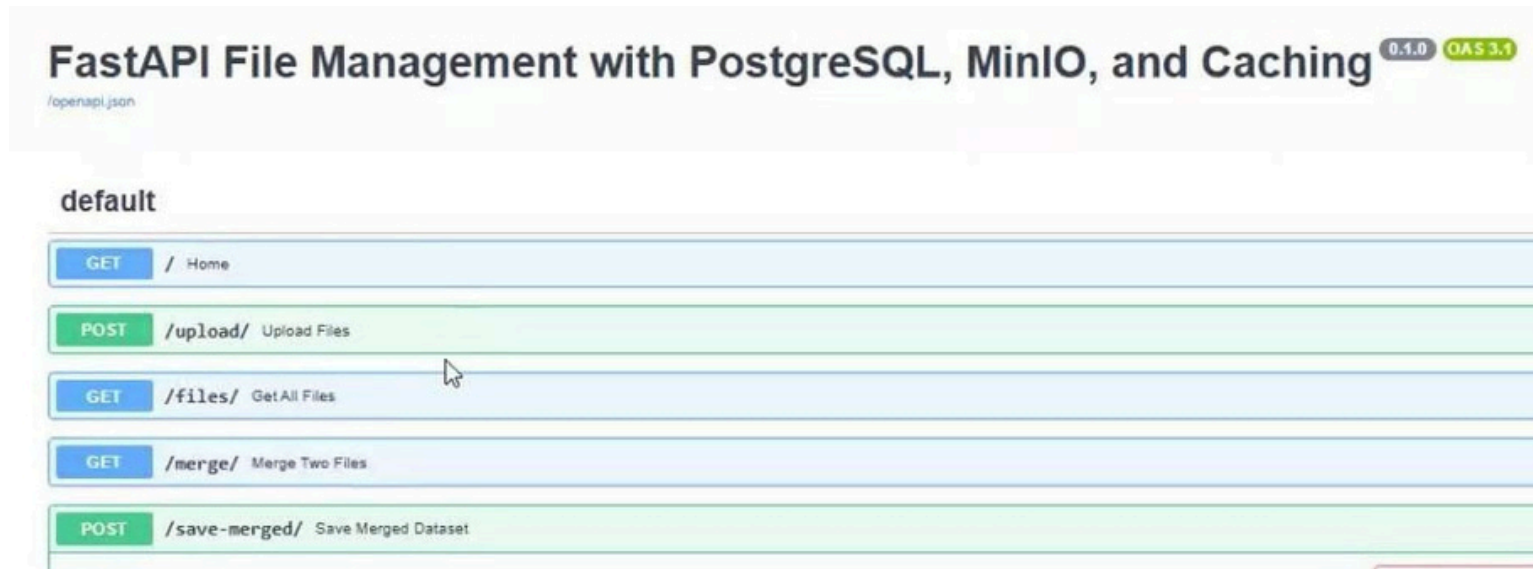
```
SELECT * FROM public.files ORDER BY id DESC;
```

The query has been executed, and the results are displayed in the Data Output tab. The results show 6 rows of data:

id	file_name	file_format	upload_time
1	merged_20251109_20470...	csv	2025-11-09 20:47:07.424061
2	Retail_Data_Transactions.c_	csv	2025-11-09 20:40:27.758175
3	Retail_Data_Response.csv	csv	2025-11-09 20:40:27.758175
4	merged_20251108_22042...	csv	2025-11-08 22:04:22.18222
5	Retail_Data_Transactions.c_	csv	2025-11-08 22:02:52.083722
6	Retail_Data_Response.csv	csv	2025-11-08 22:02:52.083722

At the bottom of the window, a notification from 'Awesome Screen Recorder & Screenshot' is visible, stating 'is sharing your screen.' with buttons for 'Stop sharing' and 'Hide'.

API Testing Environment



Swagger UI showing all FastAPI endpoints.

File Upload Verification

12	Retail_Data_Transactions.c...	csv	2025-11-09 20:40:27.758175
11	Retail_Data_Response.csv	csv	2025-11-09 20:40:27.758175

Uploaded files stored with metadata in PostgreSQL.

Merge Endpoint

The screenshot shows a REST client interface for the `GET /merge/` endpoint, titled "Merge Two Files". Under the "Parameters" tab, there is a table with two columns: "Name" and "Description". The first row shows a required integer query parameter `file_id_1` with the value `11` entered in the input field. The second row shows a required integer query parameter `file_id_2` with the value `12` entered in the input field. At the bottom of the interface is a blue "Execute" button.

Name	Description
<code>file_id_1</code> * required integer (query)	11
<code>file_id_2</code> * required integer (query)	12

Execute

File IDs 11 and 12 selected for merging.

Cached Merge Preview

Response body

```
{
  "cache_key": "d29dbfff-381a-4eb4-8fa0-64b9591ab47f",
  "preview": [
    {
      "customer_id": "CS1112",
      "response": 0,
      "trans_date": "14-Jan-15",
      "tran_amount": 39
    },
    {
      "customer_id": "CS1112",
      "response": 0,
      "trans_date": "16-Jul-14",
      "tran_amount": 90
    },
    {
      "customer_id": "CS1112",
      "response": 0,
      "trans_date": "29-Apr-14",
      "tran_amount": 63
    },
    {
      "customer_id": "CS1112",
      "response": 0,
      "trans_date": "04-Dec-14",
      "tran_amount": 39
    },
  ]
}
```

FastAPI returns preview of merged cached dataset.

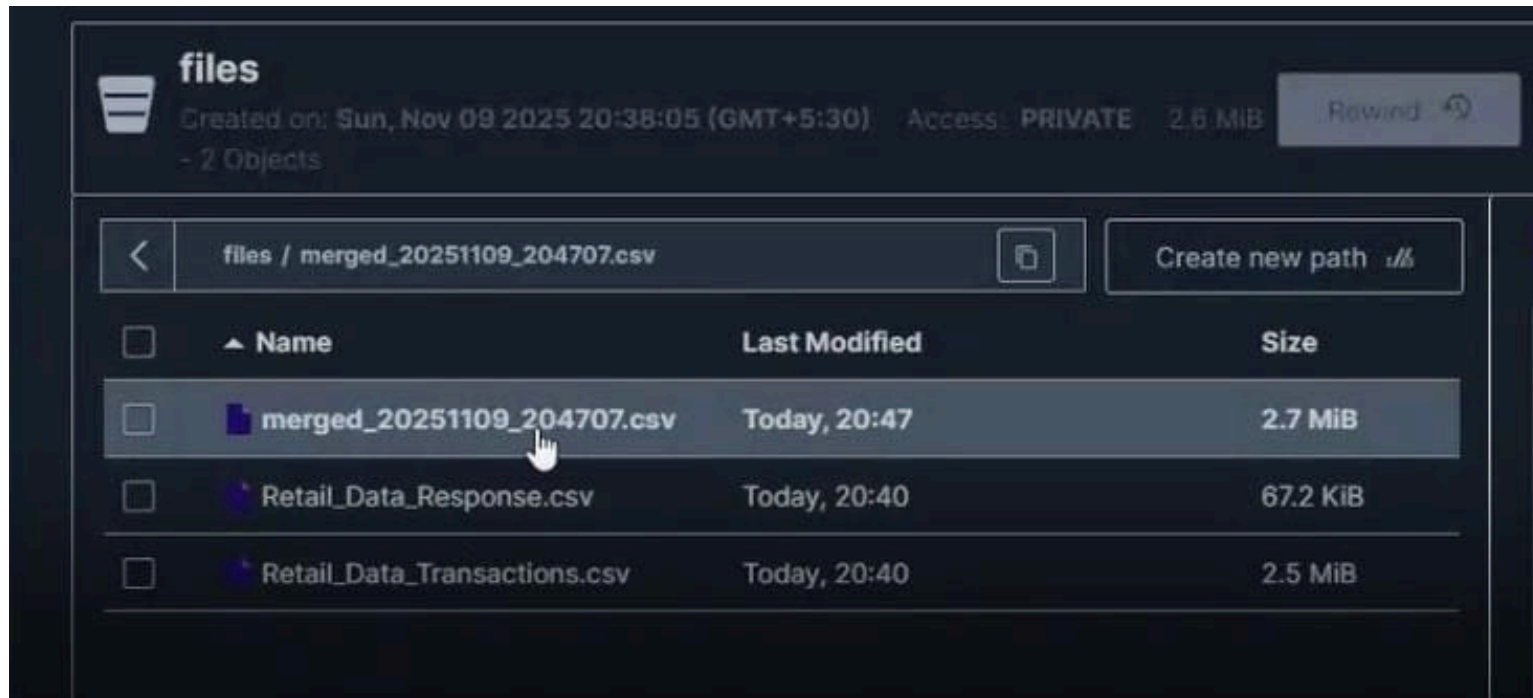
Save Merged Dataset

Response body

```
{  
  "message": "Merged dataset saved successfully to MinIO!",  
  "file_name": "merged_20251109_204707.csv"  
}
```

Merged dataset saved successfully to MinIO.

Final Files in MinIO



MinIO storage shows merged and raw datasets.

Results

- All FastAPI endpoints (/upload/, /files/, /merge/, /save-merged/) executed successfully through Swagger UI.
- Uploaded datasets were stored correctly in MinIO and their metadata (ID, filename, format, timestamp) was inserted into PostgreSQL.
- Merge operation (file_id_1 = 11, file_id_2 = 12) executed smoothly:
 - Files fetched from MinIO
 - Merged via Pandas on `customer_id`
 - Preview returned with a unique cache_key
- Cached merged dataset was successfully saved back to MinIO as: merged_20251109_204707.csv
- PostgreSQL updated with new metadata entry for the merged file.
- Full backend workflow validated end-to-end:
Upload → Retrieve → Merge (Cached) → Save → Verify

References & GitHub

- • *FastAPI Docs* — <https://fastapi.tiangolo.com>
- • *MinIO Docs* — <https://min.io/docs>
- • *PostgreSQL Docs* — <https://www.postgresql.org/docs>
- • *SQLAlchemy Docs* — <https://docs.sqlalchemy.org>
- • *GitHub Repo* — https://github.com/shrabanapaul9/fastapi_file_manager