Warehouse Program Documentation

Last Updated: February 15, 2025

Introduction

This document provides a step-by-step guide on how to run the Warehouse Program, a Python desktop application using FastAPI as the backend. The program utilizes PostgreSQL as its database management system (DBMS).

Important Updates

1. Removal of Virtual Environments

The venv and env folder has been removed from the repository due to the following reasons:

- It contains system-specific paths and binaries that work only on the original machine.
- It is not compatible across different operating systems (Windows, macOS, Linux).
- It significantly increases the repository size, causing slow cloning and pulling from GitHub.

2. Using environment.yml for Conda Environment Setup

- The environment.yml file has been added to the repository.
- This file allows developers to create a Conda environment with all necessary dependencies.

3. The Executable Program

- The **"Runnable Warehouse Program EXE"** folder contains the executable version of the application.
- This executable **only works on computers within the Masterbatch network** and will not function on external systems.

START RUNNING THE PROGRAM

The process for running the program on your machine for the first time:

1. Setup the Database and populate default data
2. Run the API SERVER
3. Execute the main.py from the frontend folder

SETUP THE DATABASE AND POPULATE DEFAULT DATA

1. Restoring the **RMManagementSystemDB_BACKUP File**

The **RMManagementSystemDB_BACKUP** file contains all essential data, including:

- Raw materials
- Statuses
- Warehouses

This backup must be used to import the default data, such as material codes, statuses, and warehouse information.

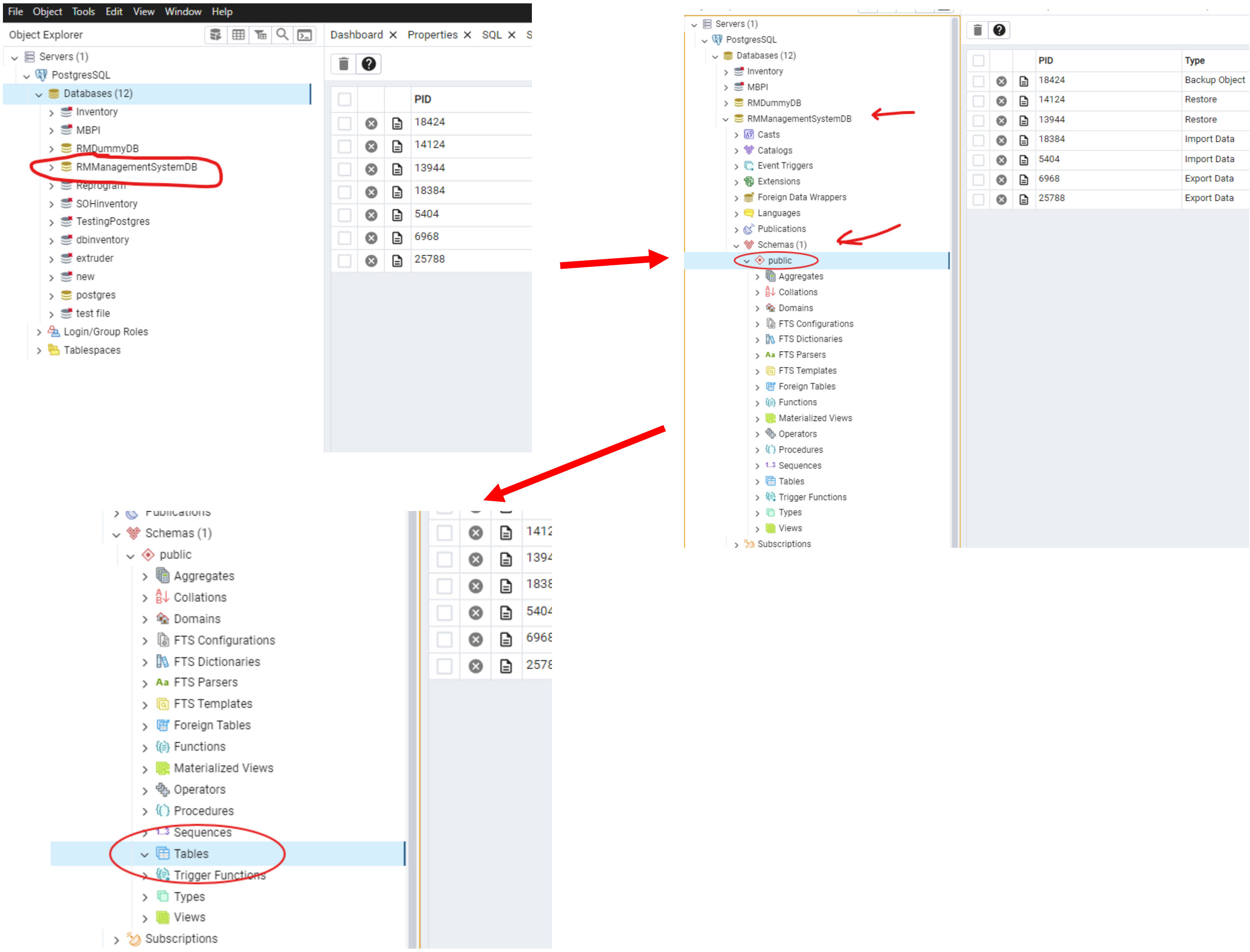
2. Steps to Restore the Database Using pgAdmin

Step 1: Ensure the public schema doesn’t have any tables (Important!)

Note: If you have just created the database and have not made any changes or started the API server yet, you can skip this step.

Before restoring the database, you must first ensure the public schema tables are empty to prevent errors during the restoration process.
To do this:

1. **Open pgAdmin 4.**
2. **Connect to your PostgreSQL server and locate the database where you will restore the backup.**
3. **Navigate to Schemas → Right-click on public → Select Delete/Delete (Cascade).**
4. **Confirm the deletion.**

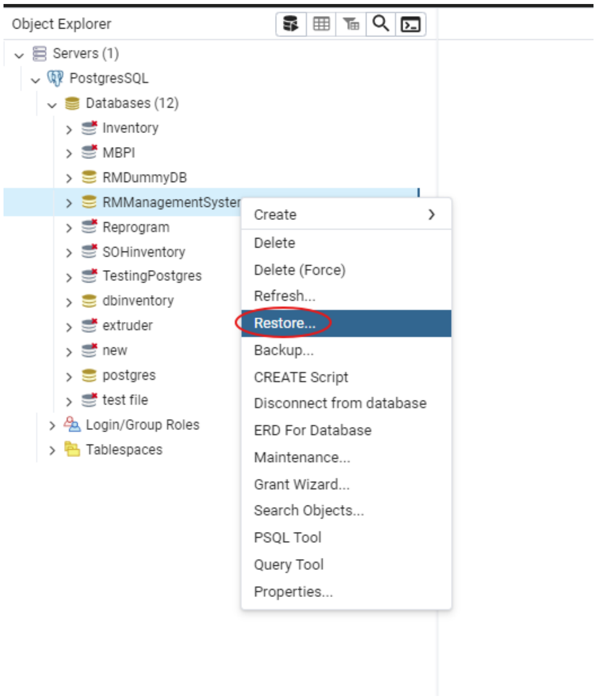


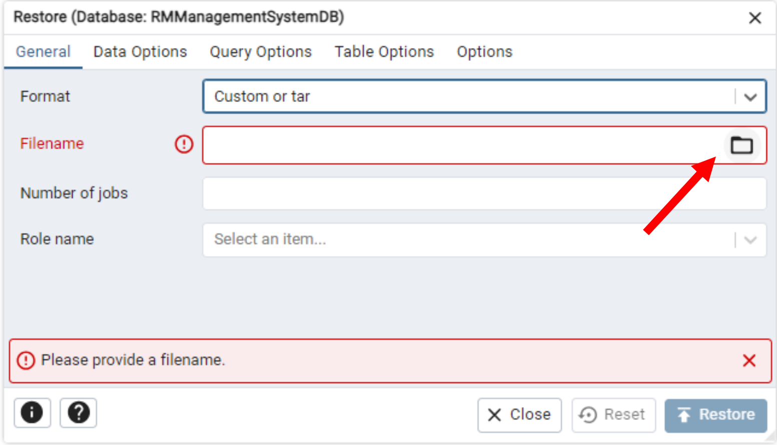
PID	Type
18424	Backup Object
14124	Restore
13944	Restore
18384	Import Data
5404	Import Data
6968	Export Data
25788	Export Data

Step 2: Restore the Database

1. **In pgAdmin, right-click on the target database and select Restore.**
2. **In the Filename section, browse and select the RMManagementSystemDB_BACKUP file.**
3. **Ensure that the format is set to Custom or tar.**
4. **Click Restore to start the process.**

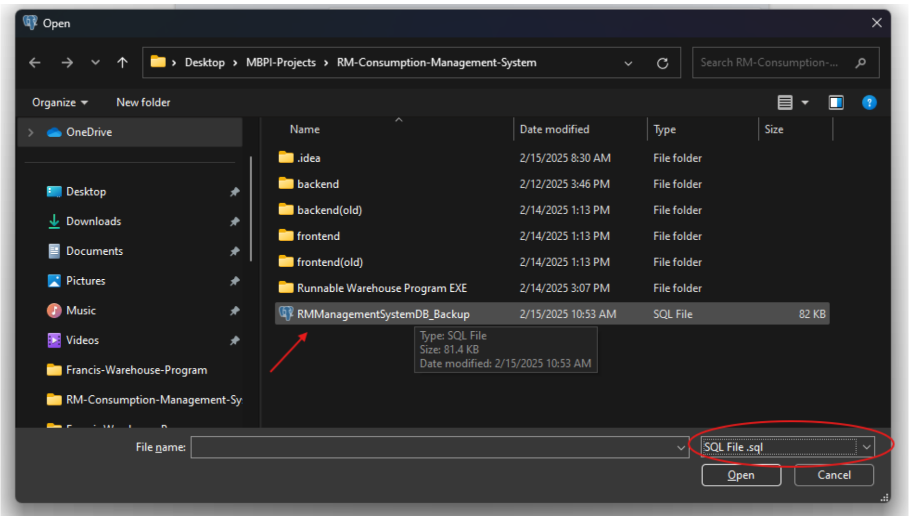
After successful restoration, the database will be populated with all necessary raw materials, statuses, and warehouse data.





Format	Custom or tar
Filename	
Number of jobs	
Role name	Select an item...

Please provide a filename.

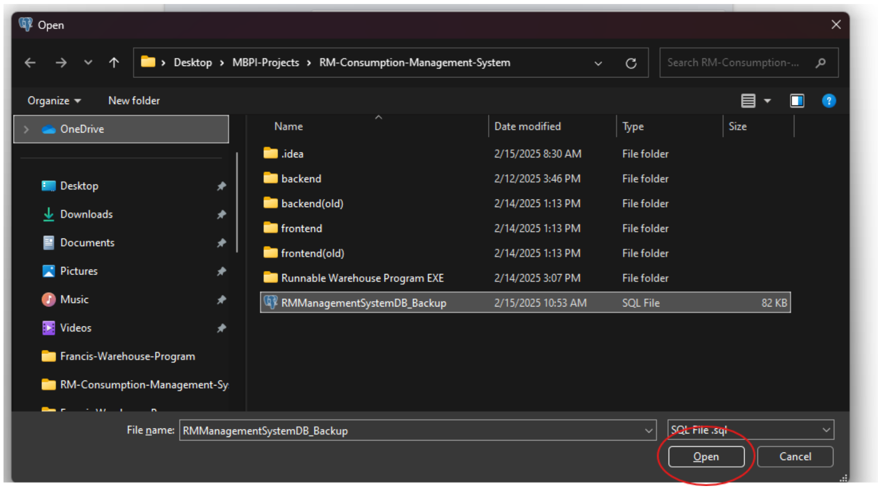

Open
Desktop › MBPI-Projects › RM-Consumption-Management-System
Search RM-Consumption-...
Organize
New folder
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
Francis-Warehouse-Program
RM-Consumption-Management-Sy
File name:
Open
Cancel

Name	Date modified	Type	Size
.idea	2/15/2025 8:30 AM	File folder	
backend	2/12/2025 3:46 PM	File folder	
backend(old)	2/14/2025 1:13 PM	File folder	
frontend	2/14/2025 1:13 PM	File folder	
frontend(old)	2/14/2025 1:13 PM	File folder	
Runnable Warehouse Program EXE	2/14/2025 3:07 PM	File folder	
RMManagementSystemDB_Backup	2/15/2025 10:53 AM	SQL File	82 KB

Type: SQL File
Size: 81.4 KB
Date modified: 2/15/2025 10:53 AM

SQL File .sql

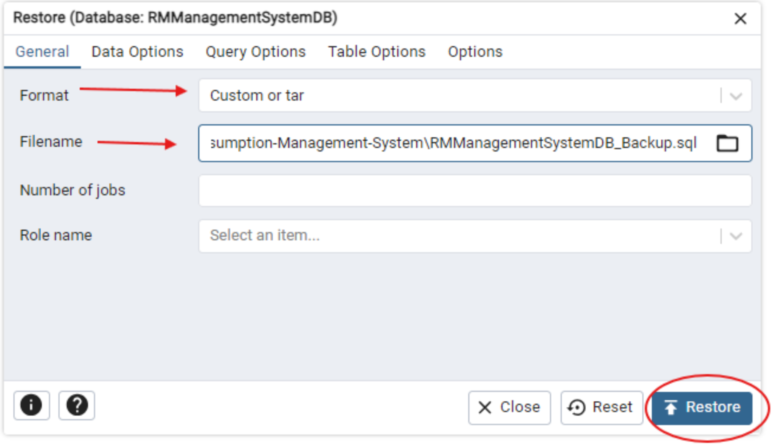
Select "All Files" or "SQL File .sql" to see the backup file


Open
Desktop › MBPI-Projects › RM-Consumption-Management-System
Search RM-Consumption-...
Organize
New folder
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
Francis-Warehouse-Program
RM-Consumption-Management-Sy

Name	Date modified	Type	Size
.idea	2/15/2025 8:30 AM	File folder	
backend	2/12/2025 3:46 PM	File folder	
backend(old)	2/14/2025 1:13 PM	File folder	
frontend	2/14/2025 1:13 PM	File folder	
frontend(old)	2/14/2025 1:13 PM	File folder	
Runnable Warehouse Program EXE	2/14/2025 3:07 PM	File folder	
RMManagementSystemDB_Backup	2/15/2025 10:53 AM	SQL File	82 KB

File name: RMManagementSystemDB_Backup

SQL File .sql


Restore (Database: RMManagementSystemDB)

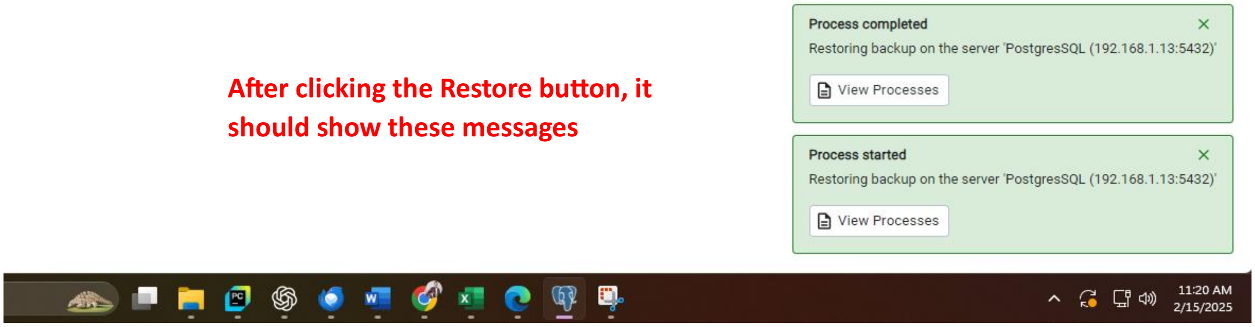
Format: Custom or tar

Filename: sumption-Management-System\RMManagementSystemDB_Backup.sql

Number of jobs:

Role name: Select an item...

After clicking the Restore button, it should show these messages

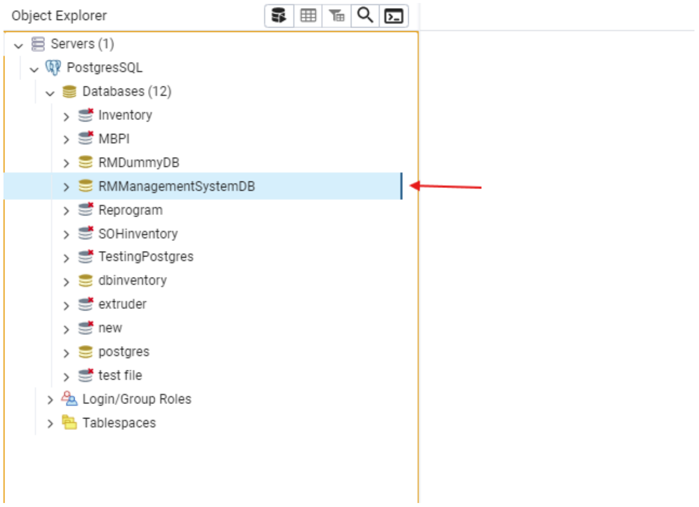

Process completed
Restoring backup on the server 'PostgresSQL (192.168.1.13:5432)'

Process started
Restoring backup on the server 'PostgresSQL (192.168.1.13:5432)'

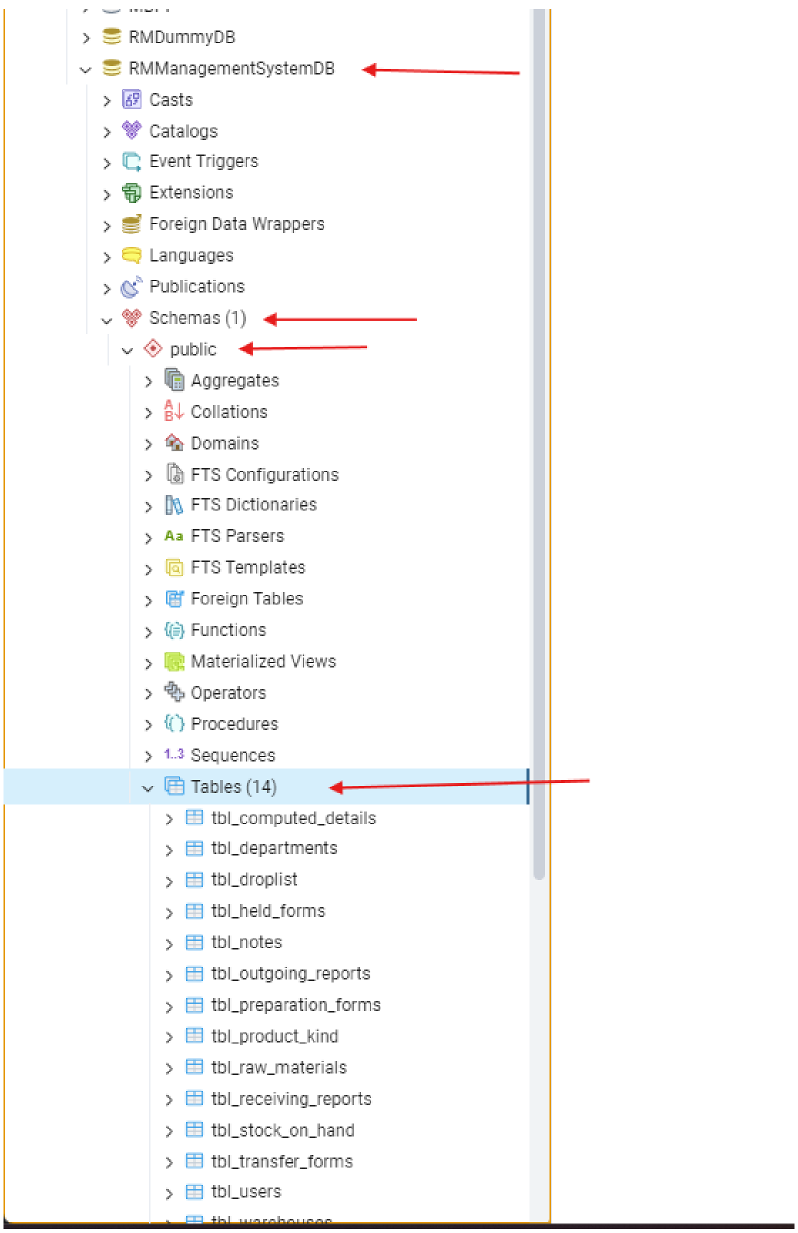
Step 3: Check the changes

1. **In pgAdmin, click on the target database > click the schemas > click the public > click the tables.**

2. **Clicking the tables should show the lists of the tables**



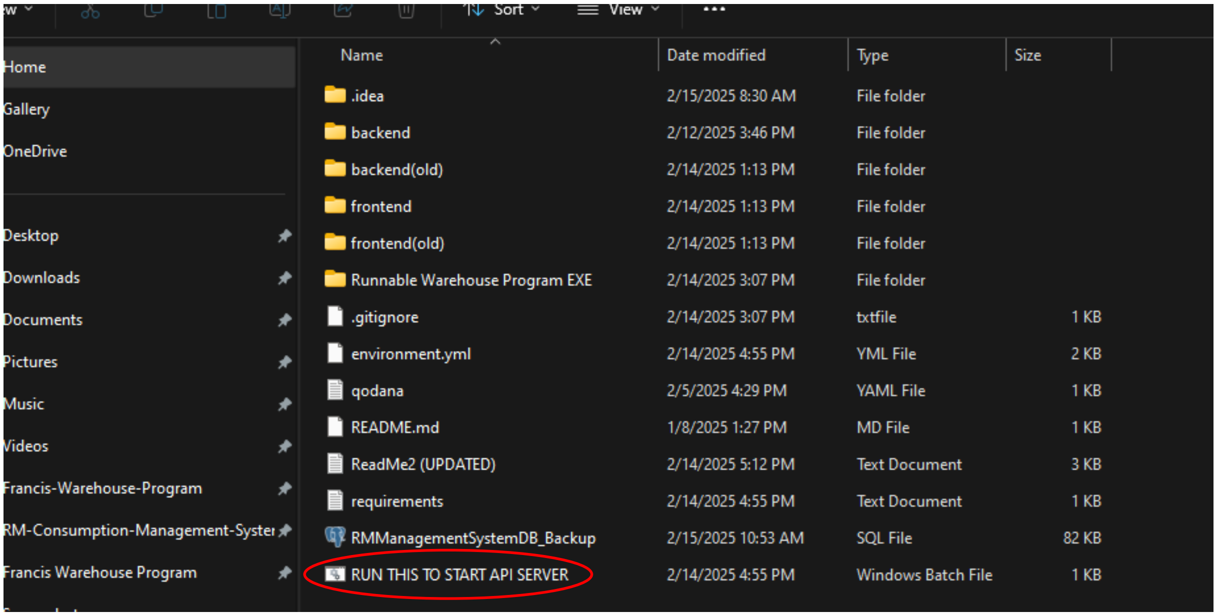
```
Object Explorer
⌄ Servers (1)
  ⌄ PostgresSQL
    ⌄ Databases (12)
      › Inventory
      › MBPI
      › RMDummyDB
      › RMManagementSystemDB
      › Reprogram
      › SOHinventory
      › TestingPostgres
      › dbinventory
      › extruder
      › new
      › postgres
      › test file
    › Login/Group Roles
    › Tablespaces
```



```
› MBPI
› RMDummyDB
⌄ RMManagementSystemDB
  › Casts
  › Catalogs
  › Event Triggers
  › Extensions
  › Foreign Data Wrappers
  › Languages
  › Publications
  ⌄ Schemas (1)
    ⌄ public
      › Aggregates
      › Collations
      › Domains
      › FTS Configurations
      › FTS Dictionaries
      › FTS Parsers
      › FTS Templates
      › Foreign Tables
      › Functions
      › Materialized Views
      › Operators
      › Procedures
      › Sequences
      ⌄ Tables (14)
        › tbl_computed_details
        › tbl_departments
        › tbl_droplist
        › tbl_held_forms
        › tbl_notes
        › tbl_outgoing_reports
        › tbl_preparation_forms
        › tbl_product_kind
        › tbl_raw_materials
        › tbl_receiving_reports
        › tbl_stock_on_hand
        › tbl_transfer_forms
        › tbl_users
        › tbl_warehouses
```

RUNNING THE API SERVER

- A batch file named "RUN THIS TO START API SERVER" is provided for easy execution.
- Simply double-click the file to start the API server.



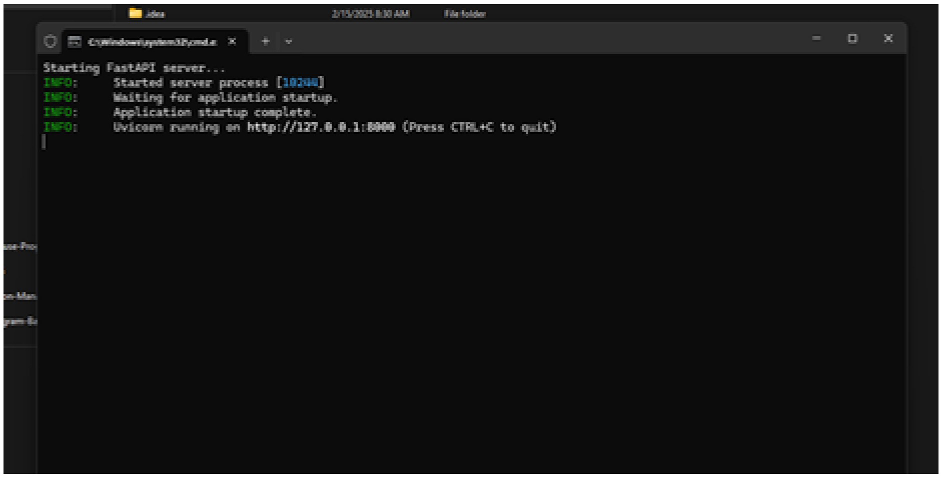
Sort
View
Home
Gallery
OneDrive
Desktop
Downloads
Documents
Pictures
Music
Videos
Francis-Warehouse-Program
RM-Consumption-Management-Syste
Francis Warehouse Program

Name	Date modified	Type	Size
.idea	2/15/2025 8:30 AM	File folder	
backend	2/12/2025 3:46 PM	File folder	
backend(old)	2/14/2025 1:13 PM	File folder	
frontend	2/14/2025 1:13 PM	File folder	
frontend(old)	2/14/2025 1:13 PM	File folder	
Runnable Warehouse Program EXE	2/14/2025 3:07 PM	File folder	
.gitignore	2/14/2025 3:07 PM	txtfile	1 KB
environment.yml	2/14/2025 4:55 PM	YML File	2 KB
qodana	2/5/2025 4:29 PM	YAML File	1 KB
README.md	1/8/2025 1:27 PM	MD File	1 KB
ReadMe2 (UPDATED)	2/14/2025 5:12 PM	Text Document	3 KB
requirements	2/14/2025 4:55 PM	Text Document	1 KB
RMManagementSystemDB_Backup	2/15/2025 10:53 AM	SQL File	82 KB
RUN THIS TO START API SERVER	2/14/2025 4:55 PM	Windows Batch File	1 KB

A terminal/command prompt window will open, indicating the server is running.

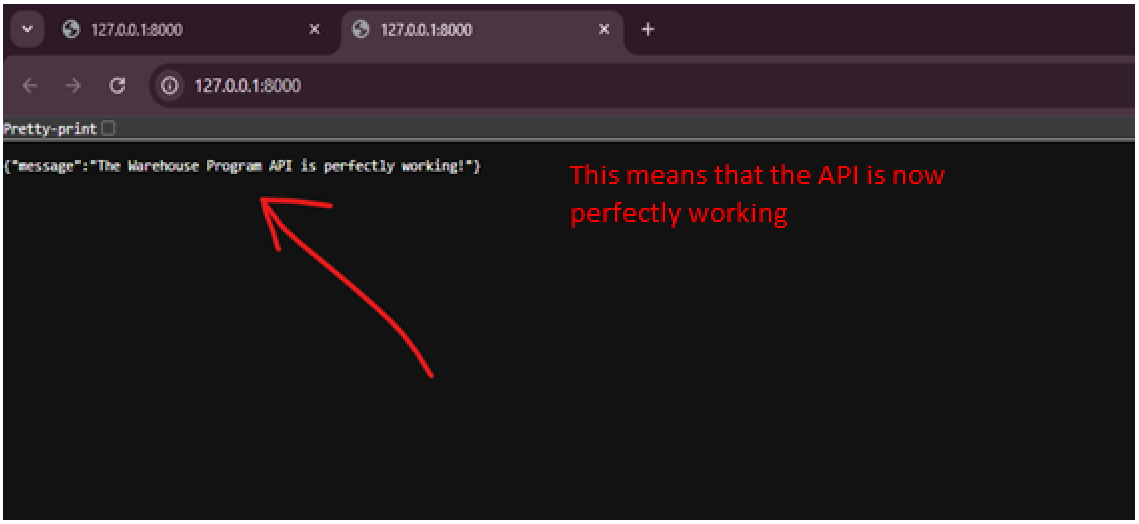
Important: Closing this terminal will stop the API server, which will cause the entire program to malfunction.

- Make sure that the API Server is running before running the main.py inside the frontend folder. You can see if this is running in the terminal if it looks like this:



```
Starting FastAPI server...
INFO:     Started server process [10200]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
```

- To test the API if its working properly, enter this to your url to your browser: http://127.0.0.1:8000. The browser should return this page:

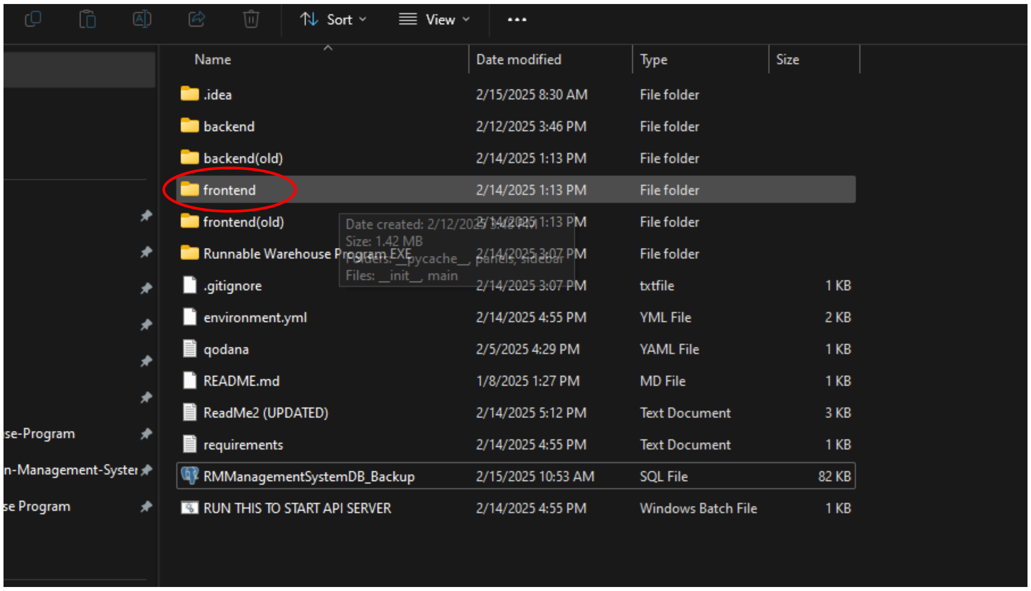


```
{"message":"The Warehouse Program API is perfectly working!"}
```

This means that the API is now perfectly working

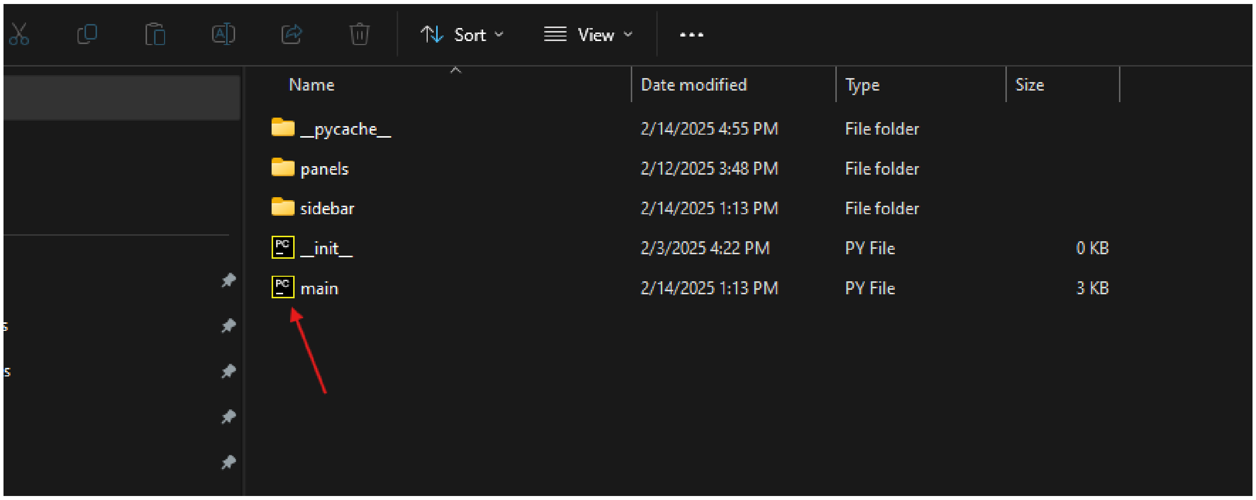
EXECUTE THE MAIN.PY FROM THE FRONTEND FOLDER

- After we are done with the API server and to manually run the program, navigate to: frontend/main.py and execute main.py using your IDE.



Name	Date modified	Type	Size
.idea	2/15/2025 8:30 AM	File folder	
backend	2/12/2025 3:46 PM	File folder	
backend(old)	2/14/2025 1:13 PM	File folder	
frontend	2/14/2025 1:13 PM	File folder	
frontend(old)	2/14/2025 1:13 PM	File folder	
Runnable Warehouse Program EXE	2/14/2025 3:07 PM	File folder	
.gitignore	2/14/2025 3:07 PM	txtfile	1 KB
environment.yml	2/14/2025 4:55 PM	YML File	2 KB
qodana	2/5/2025 4:29 PM	YAML File	1 KB
README.md	1/8/2025 1:27 PM	MD File	1 KB
ReadMe2 (UPDATED)	2/14/2025 5:12 PM	Text Document	3 KB
requirements	2/14/2025 4:55 PM	Text Document	1 KB
RMManagementSystemDB_Backup	2/15/2025 10:53 AM	SQL File	82 KB
RUN THIS TO START API SERVER	2/14/2025 4:55 PM	Windows Batch File	1 KB

Date created: 2/12/2025 3:46 PM
Size: 1.42 MB
Folders: __pycache__, panels, sidebar
Files: __init__, main



Name	Date modified	Type	Size
__pycache__	2/14/2025 4:55 PM	File folder	
panels	2/12/2025 3:48 PM	File folder	
sidebar	2/14/2025 1:13 PM	File folder	
__init__	2/3/2025 4:22 PM	PY File	0 KB
main	2/14/2025 1:13 PM	PY File	3 KB

Project Structure

Backend Folders (API)

- backend/ → Latest version of the API (**bug-fixed and stable**)
- backend(old)/ → Older version of the API (**contains bugs but kept for reference**)

Frontend Folders (GUI)

- frontend/ → Latest version of the GUI (**bug-fixed and improved UI/UX**)
- frontend(old)/ → Older version of the GUI (**contains bugs and lacks UI/UX features**)

Executable Program Folder

- Runnable Warehouse Program EXE/ → Contains the packaged application.
- **Note:** All subfiles and subfolders inside this folder are essential for the program to function properly.

The application currently uses **backend/ and frontend/** for normal operation, while the backend(old)/ and frontend(old)/ folders are retained for reference or restoring previous features.

Troubleshooting

1. API Server Fails to Start

- Ensure you have activated the Conda environment:
- conda activate warehouse_program_env
- Check if PostgreSQL is running.

2. The Application Crashes on Startup

- Ensure the API server is running before launching the frontend application.