

Project Title:

Workout Progress Tracker & Arithmetic Calculator

Intern Information

- **Name:** [Yarramalla Manasa]
 - **Internship Duration:** [08-05-2025] – [30-08-2025]
 - **Company Name:** VirtuNexa
 - **Organization:** [Kandhula lakshumma memorial college of Engineering for women]
-

1. Introduction

The purpose of this project was to develop a Python-based application that allows users to track workout progress, perform basic arithmetic operations, and generate insightful reports using data visualization. The project helps users monitor performance and calculate data such as weight progress or calorie metrics using simple inputs.

2. Objectives

- Track user workout progress over time.
- Allow arithmetic calculations for custom metrics.
- Generate visual and textual reports.
- Handle errors like invalid inputs.
- Optionally provide GUI using Tkinter.
- Optionally store history using SQLite or Pandas.

3. Tools & Technologies Used

Component	Technology
Programming Language	Python
UI (Optional)	Tkinter
Data Visualization	Matplotlib
Data Analysis	Pandas (Optional)
Database (Optional)	SQLite
Logging	Python logging

4. System Structure

4.1 User Interface

- **Console-Based UI:** Command-line prompts for input/output.
- **Tkinter GUI (Optional):** User-friendly window-based interface for data entry and result display.

4.2 Core Features

- **Basic Arithmetic Operations:** Addition, subtraction, multiplication, division with validation.
- **Workout Data Tracking:** Inputs for exercises, sets, reps, weights.
- **Error Handling:** Catches non-numeric input, zero division, and invalid operations.
- **Suggestion Engine:** Simple logic to offer improvements based on past data.
- **Reports:** Graphs for weekly progress, summaries saved to files.

5. Project Structure

File Name	Description
main.py	Console UI for calculator and tracker
calculator.py	Handles all arithmetic logic
workout_tracker.py	Manages workout inputs and suggestions
db_manager.py	(Optional) Handles SQLite storage
report_generator.py	Creates visual reports using matplotlib
logger_config.py	Logs operations for audit/debugging
README.md	Setup and usage instructions

6. Sample Code Snippet (Matplotlib Report)

python

Copy code

```
import matplotlib.pyplot as plt

days = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
weights = [50, 55, 60, 62, 65, 67, 70]

plt.plot(days, weights, marker='o')
plt.title("Weekly Weight Lifting Progress")
plt.xlabel("Day")
plt.ylabel("Weight (kg)")
```

```
plt.grid(True)
```

```
plt.show()
```

7. Testing

Test Case	Status
Division by Zero	✓
Invalid Input (e.g., text)	✓
Suggestion Accuracy	✓
SQLite/Pandas Integration	✓
Report Generation	✓

8. Checklist

Task	Completed
Console calculator with error handling	✓
GUI with Tkinter (optional)	⚠
SQLite database (optional)	✓
Report generation with Matplotlib	✓
Workout data input and suggestions	✓
Logs for tracking operations	✓
User and developer documentation	✓

9. How to Run

Console Version:

bash

[Copy code](#)

```
python main.py
```

Dependencies:

bash

[Copy code](#)

```
pip install matplotlib pandas
```

GUI Version (if implemented):

bash

[Copy code](#)

```
python gui_main.py
```

10. Conclusion

This project enhanced my practical understanding of Python programming, especially in building modular applications that handle real-world data. It strengthened my skills in working with visualization tools, databases, and UI design. With potential for future improvements such as fitness API integration and mobile deployment, this app serves as a strong foundation for real-life workout tracking systems.