Student Name: Prince Agrawal

Roll Number: 25BAI11117

Course: Introduction to Problem Solving and Programming

Institution: VIT

1. INTRODUCTION

This project titled Smart Study Planner with Analytics is developed using Python programming language. The main aim of this project is to help students manage their study tasks in an organized and efficient manner. Many students struggle to keep track of their assignments, deadlines, and study schedules. This application provides a simple solution to manage these tasks digitally.

The system allows users to add new study tasks, view them, mark them as completed, and also receive a suggested study plan. Along with this, the program also displays progress analytics which helps the user understand how much work is completed and how much is still pending.

The project is designed using basic programming concepts such as functions, loops, conditional statements, file handling and modular programming.

2. PROBLEM STATEMENT

Students often forget important deadlines and find it difficult to manage multiple subjects simultaneously. There is no simple system to track study progress and monitor task completion. Hence, there is a need for a system that:

- **Stores study tasks safely**
- **Tracks deadlines and priorities**
- **Displays task completion status**
- **Suggests effective use of study time Provides progress report**

3. OBJECTIVES

The main objectives of this project are:

- **To develop a task management system for students**
- **To enable users to add, view and update tasks**
- **To track completed and pending tasks**
- **To provide suggestive study planning**
- **To analyze task progress and display statistics**
- **To store tasks persistently using file handling**

4. TOOLS AND TECHNOLOGIES USED

- **Python Programming Language**
- **JSON Data Storage**
- **VS Code**
- **GitHub Version Control**
- **Terminal Program Execution**

5. SYSTEM DESIGN

The project follows a modular structure for better organization and readability.

Main files and their roles:

- **app.py – Main program that controls the flow**
- **models.py – Defines the Task structure**
- **storage.py – Handles data saving and loading**
- **logic.py – Contains core processing logic**
- **ui.py – Manages user interaction**
- **tasks.json – Stores task data**

This separation of files improves maintainability and makes the program easier to understand.

6. METHODOLOGY

The program follows the steps below:

1. Program displays a menu to the user

2. User selects an operation

3. Corresponding function is executed

4. Data is updated and saved

5. Output is displayed

6. Program repeats until exit

Task data is saved in tasks.json so that it remains available even after closing the program.

7. ALGORITHM

General Algorithm:

1. Start program

2. Display menu

3. Accept user choice

4. Perform action based on choice

5. Save updated data

6. Display result

7. Repeat until user chooses Exit

8. Stop

Study Plan Algorithm:

Calculate task importance based on priority

Distribute available hours accordingly

Display recommended hours per task

8. IMPLEMENTATION

The system uses Python with file handling to manage task data. Tasks are stored in JSON format containing fields such as title, subject, estimated hours, deadline, priority, and status.The user interacts through the command prompt where options are provided to manage tasks easily.

9. TESTING

The program was tested using the following steps:

Adding new tasks

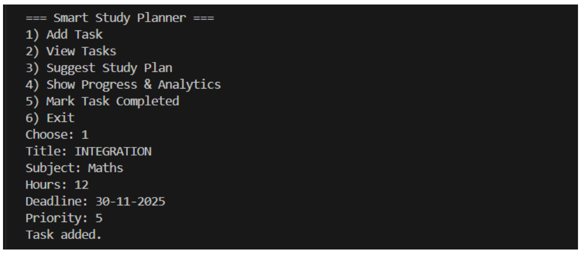
Viewing all tasks

Marking tasks as completed

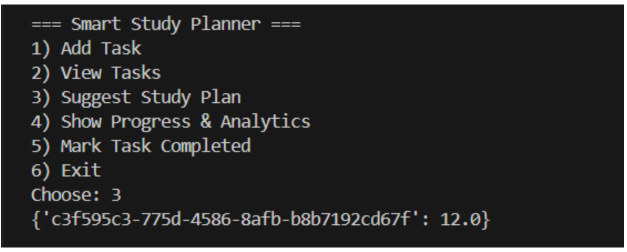
Checking analytics output

Running the program multiple times

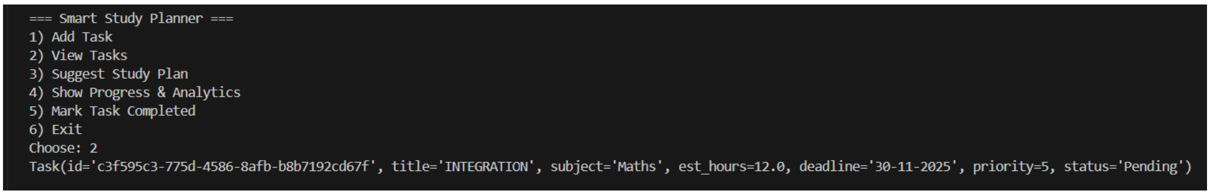
Screenshots of successful testing are shown below:



```
=== Smart Study Planner ===
1) Add Task
2) View Tasks
3) Suggest Study Plan
4) Show Progress & Analytics
5) Mark Task Completed
6) Exit
Choose: 1
Title: INTEGRATION
Subject: Maths
Hours: 12
Deadline: 30-11-2025
Priority: 5
Task added.
```



```
=== Smart Study Planner ===
1) Add Task
2) View Tasks
3) Suggest Study Plan
4) Show Progress & Analytics
5) Mark Task Completed
6) Exit
Choose: 3
{'c3f595c3-775d-4586-8afb-b8b7192cd67f': 12.0}
```



```
=== Smart Study Planner ===
1) Add Task
2) View Tasks
3) Suggest Study Plan
4) Show Progress & Analytics
5) Mark Task Completed
6) Exit
Choose: 2
Task(id='c3f595c3-775d-4586-8afb-b8b7192cd67f', title='INTEGRATION', subject='Maths', est_hours=12.0, deadline='30-11-2025', priority=5, status='Pending')
```

10. RESULTS

The project worked successfully and achieved all the desired objectives:

- **Tasks were saved correctly**
- **Tasks could be modified**
- **Progress analytics were displayed**
- **Study suggestions were generated**
- **Data remained persistent**
- **This confirms the system performs as expected.**

11. LIMITATIONS

- **Some current limitations of the system:**
- **No graphical interface**
- **Works only in terminal**
- **No online backup**
- **No automatic notifications**

12. FUTURE SCOPE

The project can be enhanced by:

- **Adding GUI using Tkinter**
- **Mobile application version**
- **Reminder notifications**
- **Cloud storage integration**
- **Calendar integration**

13. CONCLUSION

The Smart Study Planner is an effective tool for managing study tasks. It demonstrates the practical application of Python programming in solving real-life problems. The project helped in understanding concepts such as modular programming, file handling and task automation. Overall, the project fulfills its purpose and can be further enhanced in future.

14. REFERENCES

- **Python Official Documentation**
- **Programming Textbooks**
- **Online Python Tutorials**