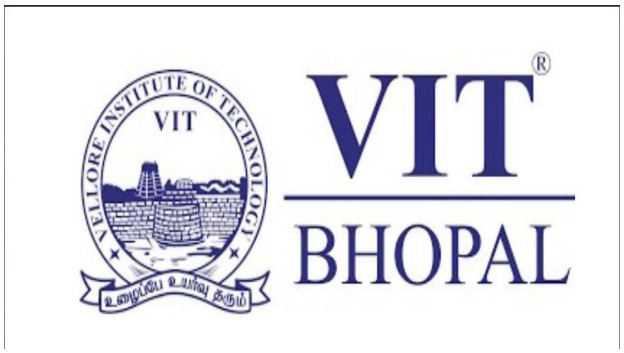


PROJECT REPORT

TITTLE: DYANAMIC ACADEMIC PERFORMANCE/PLANNER ASSISTANT

NAME: Uday Deval Chaturvedi

REG. NO. : 25BCE10300

COURSE: CSE

Date: NOVEMBER 2025

1. Introduction

The Dynamic Academic Performance/Planner Assistant (DAPA) is a command-line based academic support tool created to help students understand and manage their academic performance. It provides a simple way to track course information, calculate expected grades, estimate CGPA, and plan study time. The goal was to design a tool that feels practical not overly technical so that any student can use it comfortably.

2. Problem Statement

Academic evaluation in universities is distributed across multiple components such as assignments, quizzes, attendance, mid-semester examinations and final evaluations. However, students often do not have a clear picture of how each of these components influences their final course score or CGPA.

The lack of a unified system leads to:

- difficulty estimating required marks
- inconsistent academic planning
- misallocation of study time
- confusion about grade calculations

Therefore, there is a need for a simple, modular, and easily accessible tool that consolidates course information, tracks marks, predicts CGPA, and helps students plan their study efforts more effectively

3. Functional Requirements

• Course Manager to add/update course metadata

• Marks Entry for internal, mid, final and attendance

• CGPA Prediction based on converted weighted marks

• What-If Mode for hypothetical scenarios

• Study Planner that distributes daily study hours

• Unified summary of all the courses

• Summary Export (JSON/CSV)

4. Non-Functional Requirements

• Easy to use and understand

• Validates inputs such as marks and attendance

• Fast and responsive due to in-memory processing

• Modular design for future upgrades

5. System Architecture

The system follows a clean modular structure:

• data_manager.py —> handles storage

• planner.py — > calculations and CGPA logic

• visualizer.py — > analytical summaries

• main.py —> interactive CLI interface

6. Key Design Decisions

Python CLI was chosen due to its simplicity and familiarity for students. Data was stored in JSON-like dictionaries for convenience rather than using a database. The architecture is purposely minimal to ensure the tool remains lightweight.

7. Implementation Overview

Workflow:

1. Add courses

2. Enter marks

3. Convert raw marks → weighted marks

4. Compute CGPA

5. Generate study plan

6. Export summary

8. Challenges Faced

• Maintaining readability while separating logic

• Mapping attendance percentage to marks

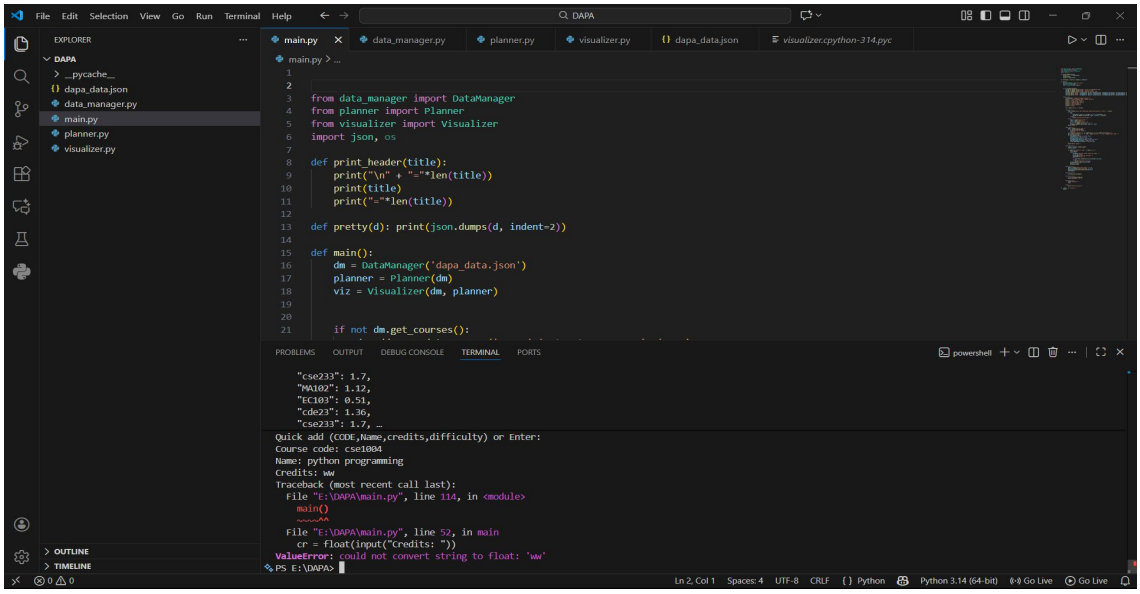
• Ensuring no invalid data breaks execution

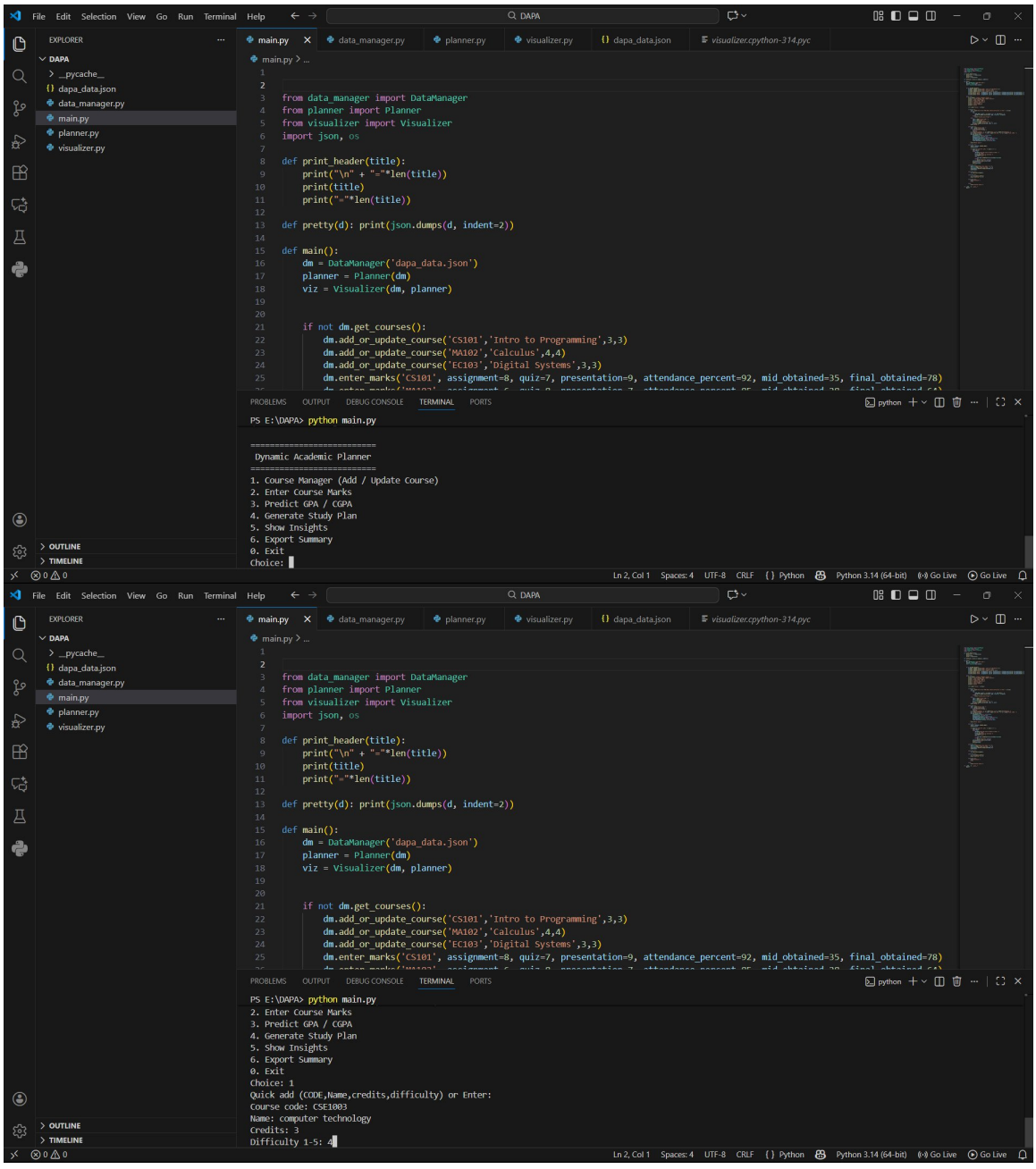
• Designing a friendly CLI interface

9. Testing

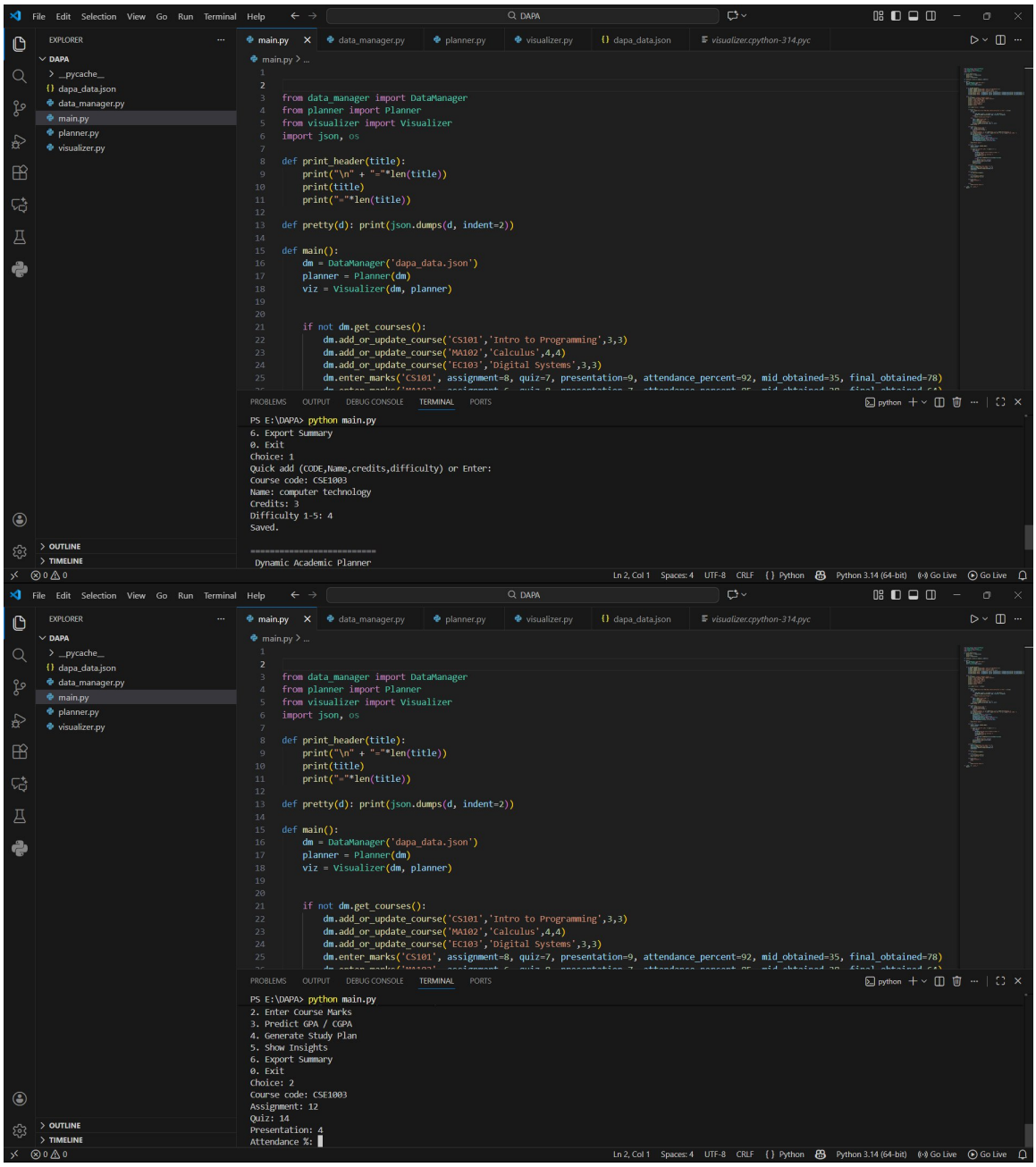
Testing was carried out with:

- Function-level validation
- Full-scenario runs (input → CGPA output)
- Invalid input tests

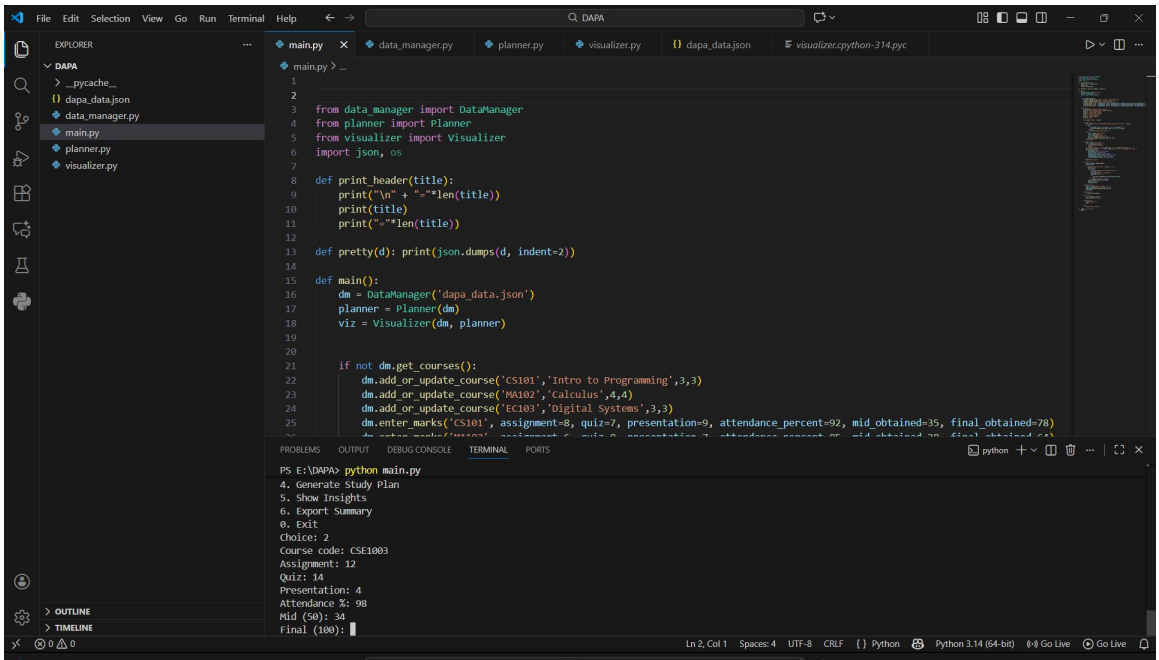


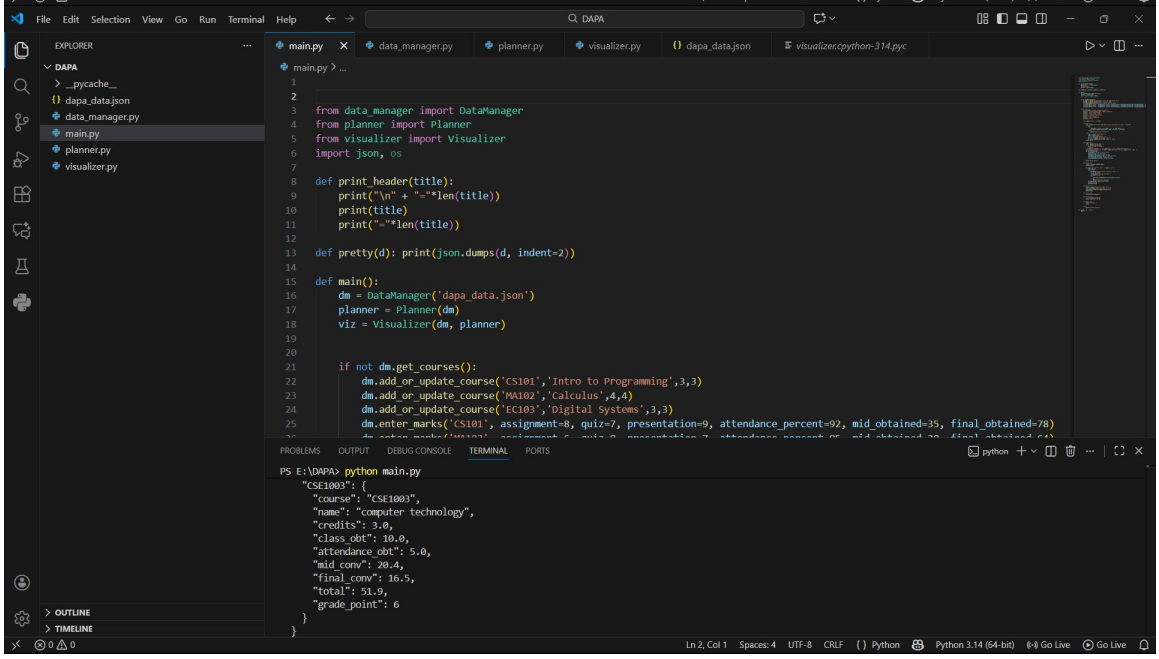

File Edit Selection View Go Run Terminal Help
DAPA
EXPLORER
DAPA
__pycache__
dapa_data.json
data_manager.py
main.py
planner.py
visualizer.py
main.py
data_manager.py
planner.py
visualizer.py
dapa_data.json
visualizer.cpython-314.pyc
main.py > ...
from data_manager import DataManager
from planner import Planner
from visualizer import Visualizer
import json, os
def print_header(title):
print("\n" + "="*len(title))
print(title)
print("="*len(title))
def pretty(d): print(json.dumps(d, indent=2))
def main():
dm = DataManager('dapa_data.json')
planner = Planner(dm)
viz = Visualizer(dm, planner)
if not dm.get_courses():
dm.add_or_update_course('CS101','Intro to Programming',3,3)
dm.add_or_update_course('MA102','Calculus',4,4)
dm.add_or_update_course('EC103','Digital Systems',3,3)
dm.enter_marks('CS101', assignment=8, quiz=7, presentation=9, attendance_percent=92, mid_obtained=35, final_obtained=78)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python
PS E:\DAPA> python main.py
Dynamic Academic Planner
1. Course Manager (Add / Update Course)
2. Enter Course Marks
3. Predict GPA / CGPA
4. Generate Study Plan
5. Show Insights
6. Export Summary
0. Exit
Choice:
OUTLINE
TIMELINE
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF Python Python 3.14 (64-bit) Go Live

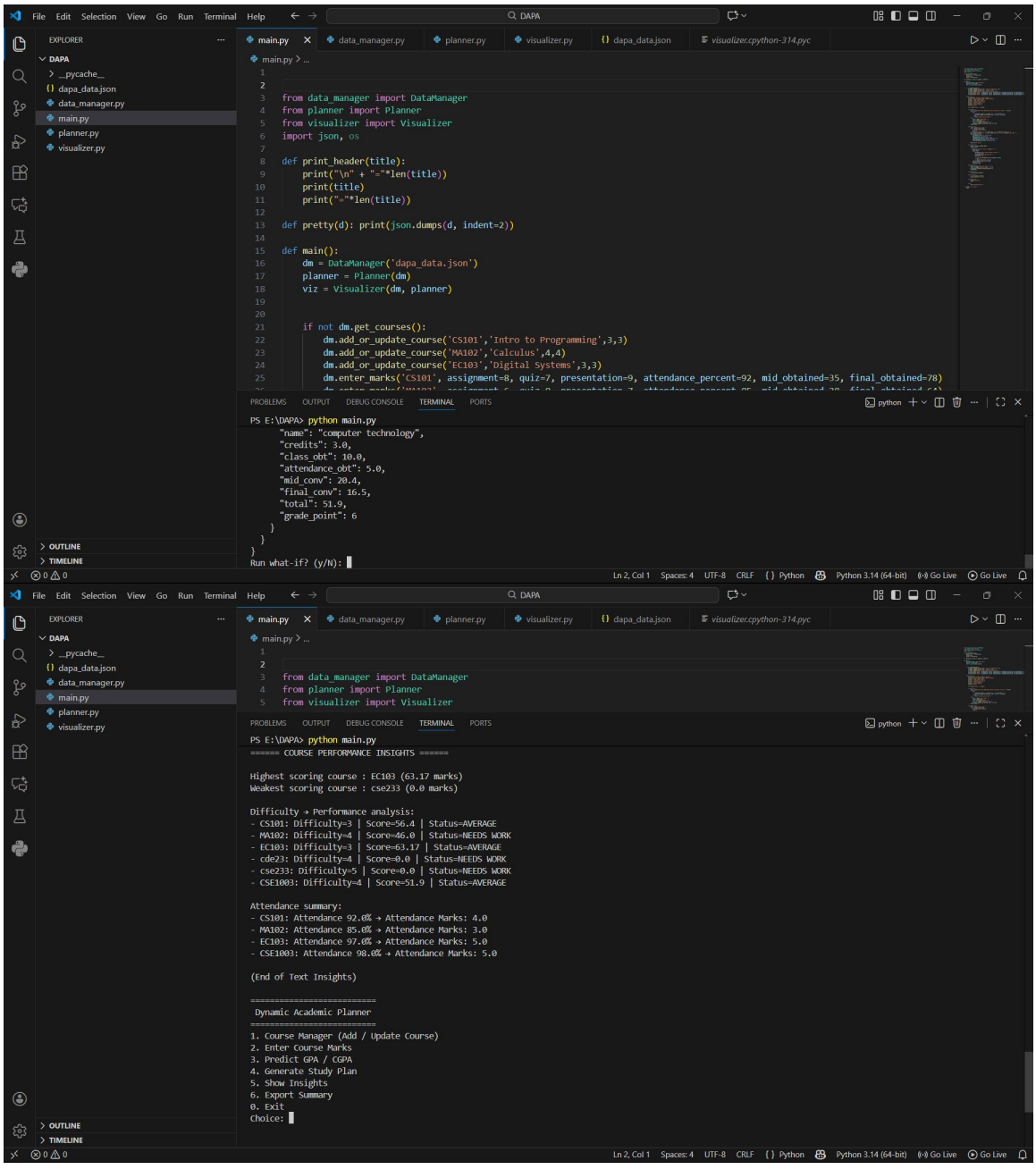
File Edit Selection View Go Run Terminal Help
DAPA
EXPLORER
DAPA
__pycache__
dapa_data.json
data_manager.py
main.py
planner.py
visualizer.py
main.py
data_manager.py
planner.py
visualizer.py
dapa_data.json
visualizer.cpython-314.pyc
main.py > ...
from data_manager import DataManager
from planner import Planner
from visualizer import Visualizer
import json, os
def print_header(title):
print("\n" + "="*len(title))
print(title)
print("="*len(title))
def pretty(d): print(json.dumps(d, indent=2))
def main():
dm = DataManager('dapa_data.json')
planner = Planner(dm)
viz = Visualizer(dm, planner)
if not dm.get_courses():
dm.add_or_update_course('CS101','Intro to Programming',3,3)
dm.add_or_update_course('MA102','Calculus',4,4)
dm.add_or_update_course('EC103','Digital Systems',3,3)
dm.enter_marks('CS101', assignment=8, quiz=7, presentation=9, attendance_percent=92, mid_obtained=35, final_obtained=78)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python
PS E:\DAPA> python main.py
2. Enter Course Marks
3. Predict GPA / CGPA
4. Generate Study Plan
5. Show Insights
6. Export Summary
0. Exit
Choice: 1
Quick add (CODE,Name,credits,difficulty) or Enter:
Course code: CSE1003
Name: computer technology
Credits: 3
Difficulty 1-5: 4
OUTLINE
TIMELINE
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF Python Python 3.14 (64-bit) Go Live


File Edit Selection View Go Run Terminal Help
DAPA
EXPLORER
DAPA
__pycache__
dapa_data.json
data_manager.py
main.py
planner.py
visualizer.py
main.py
data_manager.py
planner.py
visualizer.py
dapa_data.json
visualizer.cpython-314.pyc
main.py > ...
from data_manager import DataManager
from planner import Planner
from visualizer import Visualizer
import json, os
def print_header(title):
print("\n" + "="*len(title))
print(title)
print("="*len(title))
def pretty(d): print(json.dumps(d, indent=2))
def main():
dm = DataManager('dapa_data.json')
planner = Planner(dm)
viz = Visualizer(dm, planner)
if not dm.get_courses():
dm.add_or_update_course('CS101','Intro to Programming',3,3)
dm.add_or_update_course('MA102','Calculus',4,4)
dm.add_or_update_course('EC103','Digital Systems',3,3)
dm.enter_marks('CS101', assignment=8, quiz=7, presentation=9, attendance_percent=92, mid_obtained=35, final_obtained=78)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python
PS E:\DAPA> python main.py
6. Export Summary
0. Exit
Choice: 1
Quick add (CODE,Name,credits,difficulty) or Enter:
Course code: CSE1003
Name: computer technology
Credits: 3
Difficulty 1-5: 4
Saved.
Dynamic Academic Planner
OUTLINE
TIMELINE
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF { } Python Python 3.14 (64-bit) Go Live Go Live

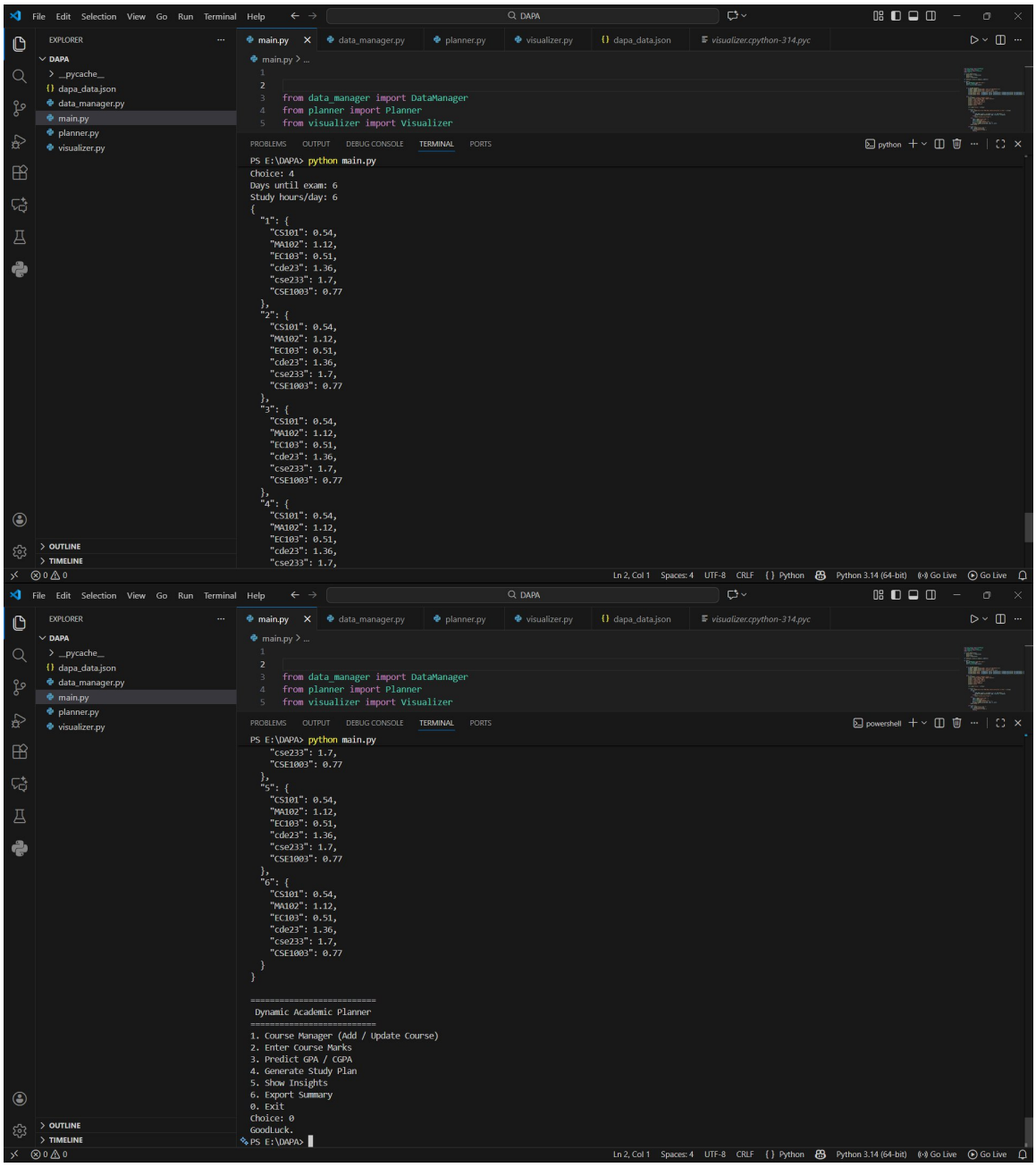
File Edit Selection View Go Run Terminal Help
DAPA
EXPLORER
DAPA
__pycache__
dapa_data.json
data_manager.py
main.py
planner.py
visualizer.py
main.py
data_manager.py
planner.py
visualizer.py
dapa_data.json
visualizer.cpython-314.pyc
main.py > ...
from data_manager import DataManager
from planner import Planner
from visualizer import Visualizer
import json, os
def print_header(title):
print("\n" + "="*len(title))
print(title)
print("="*len(title))
def pretty(d): print(json.dumps(d, indent=2))
def main():
dm = DataManager('dapa_data.json')
planner = Planner(dm)
viz = Visualizer(dm, planner)
if not dm.get_courses():
dm.add_or_update_course('CS101','Intro to Programming',3,3)
dm.add_or_update_course('MA102','Calculus',4,4)
dm.add_or_update_course('EC103','Digital Systems',3,3)
dm.enter_marks('CS101', assignment=8, quiz=7, presentation=9, attendance_percent=92, mid_obtained=35, final_obtained=78)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python
PS E:\DAPA> python main.py
2. Enter Course Marks
3. Predict GPA / CGPA
4. Generate Study Plan
5. Show Insights
6. Export Summary
0. Exit
Choice: 2
Course code: CSE1003
Assignment: 12
Quiz: 14
Presentation: 4
Attendance %:
OUTLINE
TIMELINE
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF { } Python Python 3.14 (64-bit) Go Live Go Live


File Edit Selection View Go Run Terminal Help
DAPA
EXPLORER
DAPA
__pycache__
dapa_data.json
data_manager.py
main.py
planner.py
visualizer.py
main.py
data_manager.py
planner.py
visualizer.py
dapa_data.json
visualizer.cpython-314.pyc
main.py > ...
from data_manager import DataManager
from planner import Planner
from visualizer import Visualizer
import json, os
def print_header(title):
print("\n" + "="*len(title))
print(title)
print("="*len(title))
def pretty(d): print(json.dumps(d, indent=2))
def main():
dm = DataManager('dapa_data.json')
planner = Planner(dm)
viz = Visualizer(dm, planner)
if not dm.get_courses():
dm.add_or_update_course('CS101','Intro to Programming',3,3)
dm.add_or_update_course('MA102','Calculus',4,4)
dm.add_or_update_course('EC103','Digital Systems',3,3)
dm.enter_marks('CS101', assignment=8, quiz=7, presentation=9, attendance_percent=92, mid_obtained=35, final_obtained=78)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python
PS E:\DAPA> python main.py
4. Generate Study Plan
5. Show Insights
6. Export Summary
0. Exit
Choice: 2
Course code: CSE1003
Assignment: 12
Quiz: 14
Presentation: 4
Attendance %: 98
Mid (50): 34
Final (100):
OUTLINE
TIMELINE
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF { } Python Python 3.14 (64-bit) Go Live Go Live


File Edit Selection View Go Run Terminal Help
DAPA
EXPLORER
DAPA
__pycache__
dapa_data.json
data_manager.py
main.py
planner.py
visualizer.py
main.py
data_manager.py
planner.py
visualizer.py
dapa_data.json
visualizer.cpython-314.pyc
main.py > ...
from data_manager import DataManager
from planner import Planner
from visualizer import Visualizer
import json, os
def print_header(title):
print("\n" + "="*len(title))
print(title)
print("="*len(title))
def pretty(d): print(json.dumps(d, indent=2))
def main():
dm = DataManager('dapa_data.json')
planner = Planner(dm)
viz = Visualizer(dm, planner)
if not dm.get_courses():
dm.add_or_update_course('CS101','Intro to Programming',3,3)
dm.add_or_update_course('MA102','Calculus',4,4)
dm.add_or_update_course('EC103','Digital Systems',3,3)
dm.enter_marks('CS101', assignment=8, quiz=7, presentation=9, attendance_percent=92, mid_obtained=35, final_obtained=78)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python
PS E:\DAPA> python main.py
"CSE1003": {
"course": "CSE1003",
"name": "computer technology",
"credits": 3.0,
"class_obt": 10.0,
"attendance_obt": 5.0,
"mid_conv": 20.4,
"final_conv": 16.5,
"total": 51.9,
"grade_point": 6
}
}
OUTLINE
TIMELINE
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF { } Python Python 3.14 (64-bit) Go Live Go Live


File Edit Selection View Go Run Terminal Help
DAPA
EXPLORER
DAPA
__pycache__
dapa_data.json
data_manager.py
main.py
planner.py
visualizer.py
main.py
data_manager.py
planner.py
visualizer.py
dapa_data.json
visualizer.cpython-314.pyc
main.py > ...
from data_manager import DataManager
from planner import Planner
from visualizer import Visualizer
import json, os
def print_header(title):
print("\n" + "="*len(title))
print(title)
print("="*len(title))
def pretty(d): print(json.dumps(d, indent=2))
def main():
dm = DataManager('dapa_data.json')
planner = Planner(dm)
viz = Visualizer(dm, planner)
if not dm.get_courses():
dm.add_or_update_course('CS101','Intro to Programming',3,3)
dm.add_or_update_course('MA102','Calculus',4,4)
dm.add_or_update_course('EC103','Digital Systems',3,3)
dm.enter_marks('CS101', assignment=8, quiz=7, presentation=9, attendance_percent=92, mid_obtained=35, final_obtained=78)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python
PS E:\DAPA> python main.py
"name": "computer technology",
"credits": 3.0,
"class_obt": 10.0,
"attendance_obt": 5.0,
"mid_conv": 20.4,
"final_conv": 16.5,
"total": 51.9,
"grade_point": 6
}
}
}
Run what-if? (y/N):
OUTLINE
TIMELINE
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF { } Python Python 3.14 (64-bit) Go Live Go Live

File Edit Selection View Go Run Terminal Help
DAPA
EXPLORER
DAPA
__pycache__
dapa_data.json
data_manager.py
main.py
planner.py
visualizer.py
main.py
data_manager.py
planner.py
visualizer.py
dapa_data.json
visualizer.cpython-314.pyc
main.py > ...
from data_manager import DataManager
from planner import Planner
from visualizer import Visualizer
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python
PS E:\DAPA> python main.py
====== COURSE PERFORMANCE INSIGHTS ======
Highest scoring course : EC103 (63.17 marks)
Weakest scoring course : cse233 (0.0 marks)
Difficulty → Performance analysis:
- CS101: Difficulty=3 | Score=56.4 | Status=AVERAGE
- MA102: Difficulty=4 | Score=46.0 | Status=NEEDS WORK
- EC103: Difficulty=3 | Score=63.17 | Status=AVERAGE
- cde23: Difficulty=4 | Score=0.0 | Status=NEEDS WORK
- cse233: Difficulty=5 | Score=0.0 | Status=NEEDS WORK
- CSE1003: Difficulty=4 | Score=51.9 | Status=AVERAGE
Attendance summary:
- CS101: Attendance 92.0% → Attendance Marks: 4.0
- MA102: Attendance 85.0% → Attendance Marks: 3.0
- EC103: Attendance 97.0% → Attendance Marks: 5.0
- CSE1003: Attendance 98.0% → Attendance Marks: 5.0
(End of Text Insights)
==========
Dynamic Academic Planner
==========
1. Course Manager (Add / Update Course)
2. Enter Course Marks
3. Predict GPA / CGPA
4. Generate Study Plan
5. Show Insights
6. Export Summary
0. Exit
Choice:
OUTLINE
TIMELINE
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF { } Python Python 3.14 (64-bit) Go Live Go Live


File Edit Selection View Go Run Terminal Help
DAPA
EXPLORER
DAPA
__pycache__
dapa_data.json
data_manager.py
main.py
planner.py
visualizer.py
main.py
data_manager.py
planner.py
visualizer.py
dapa_data.json
visualizer.cpython-314.pyc
from data_manager import DataManager
from planner import Planner
from visualizer import Visualizer
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\DAPA> python main.py
Choice: 4
Days until exam: 6
Study hours/day: 6
{
"1": {
"CS101": 0.54,
"MA102": 1.12,
"EC103": 0.51,
"cde23": 1.36,
"cse233": 1.7,
"CSE1003": 0.77
},
"2": {
"CS101": 0.54,
"MA102": 1.12,
"EC103": 0.51,
"cde23": 1.36,
"cse233": 1.7,
"CSE1003": 0.77
},
"3": {
"CS101": 0.54,
"MA102": 1.12,
"EC103": 0.51,
"cde23": 1.36,
"cse233": 1.7,
"CSE1003": 0.77
},
"4": {
"CS101": 0.54,
"MA102": 1.12,
"EC103": 0.51,
"cde23": 1.36,
"cse233": 1.7,
OUTLINE
TIMELINE
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF { } Python Python 3.14 (64-bit) Go Live Go Live

File Edit Selection View Go Run Terminal Help
DAPA
EXPLORER
DAPA
__pycache__
dapa_data.json
data_manager.py
main.py
planner.py
visualizer.py
main.py
data_manager.py
planner.py
visualizer.py
dapa_data.json
visualizer.cpython-314.pyc
from data_manager import DataManager
from planner import Planner
from visualizer import Visualizer
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\DAPA> python main.py
"cse233": 1.7,
"CSE1003": 0.77
},
"5": {
"CS101": 0.54,
"MA102": 1.12,
"EC103": 0.51,
"cde23": 1.36,
"cse233": 1.7,
"CSE1003": 0.77
},
"6": {
"CS101": 0.54,
"MA102": 1.12,
"EC103": 0.51,
"cde23": 1.36,
"cse233": 1.7,
"CSE1003": 0.77
}
}
==========
Dynamic Academic Planner
==========
1. Course Manager (Add / Update Course)
2. Enter Course Marks
3. Predict GPA / CGPA
4. Generate Study Plan
5. Show Insights
6. Export Summary
0. Exit
Choice: 0
GoodLuck.
PS E:\DAPA>
OUTLINE
TIMELINE
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF { } Python Python 3.14 (64-bit) Go Live Go Live

10. Conclusion

The Dynamic Academic Performance/Planner Assistant (DAPA) was created with a clear goal: to give students more control, awareness, and clarity over their academic journey. In a time when coursework, exams, and deadlines can feel overwhelming, DAPA acts as a personalized academic guide. It helps learners see where they stand and what steps they need to take next. During development, the focus was on making the system truly useful rather than overly complicated. The modular design, clean calculations, practical What-If simulator, and realistic study planner all aim to support real decision-making rather than just display numbers. By combining accuracy with simplicity, DAPA manages to turn academic data into meaningful insights. Beyond calculations, the project encourages students to think ahead. It helps them predict outcomes instead of just reacting to them. It supports intentional planning of study time rather than haphazard scheduling, and it organizes all course-related information in one place. With its mix of technical structure and student-friendly usability. In essence, DAPA is more than an academic tool. It represents a step toward creating a smarter, more informed, and more confident learning experience. As it develops, it has the potential to become a companion that supports students throughout their academic semesters, helping them perform better, plan better, and ultimately understand themselves better as learners.

12. Reference

Youtube videos and github examples