

## Personal Programming Project Report

### Title: Study Habit Tracker

Provide a descriptive title for your project.

#### 1. (5pts) Honor Code and LLM Usage for this Report.

- I followed the Virginia Tech Honor Code and did not use any resources apart from the canvas assignment and online courses for the completion of this assignment.
- I used ChatGPT for limited debugging and testing support, such as understanding error messages and suggesting fixes or additional test cases.

#### 2. (15pts) Learning Objectives:

- By dividing the project into separate functions, I was able to achieve this goal. I used dictionaries and algorithms to total study time by course, and I decided to store data in CSV format. Coding was considerably simpler when the data structure was planned first.

#### 3. (15pts) Timeline:

- Outline how you spent time on your project. Break down the time into specific tasks or milestones. Here is an adjustable schedule to get you started. Actual Details should be 50-100 words each and should compare or reflect on differences from your proposal.

Time	Task	Expected Details from Proposal	Actual Details
Hour 1-2	Research and gather resources	Review Python notes, research CSV and command-line examples, create GitHub repo	Reviewed Python documentation for csv and datetime modules. Created repo with README. Sketched data structure on paper. It took slightly longer but was valuable.
Hour 3-4	Design the project structure and	Design menu system, choose CSV format,	Created menu with 4 options. Made study_tracker.py and functions.py to keep code

Time	Task	Expected Details from Proposal	Actual Details
	plan	write first Python file	organized. First commit with basic structure.
Hour 5-6	Start coding the basic functionalities	Implement MVP: menu, add sessions, save to file, load on startup	Built core features: menu loop, add session with csv module. Tested persistence across restarts. Made 4-5 commits. It went smoothly due to good planning.
Hour 7-8	Test and debug the initial version	Test with different courses, implement stats, handle edge cases	Added calculating stats using dictionaries. Fixed bugs with empty files and invalid input. Added validation for positive integers and date format. Made 3-4 commits for fixes.
Hour 9-10	Refine and add advanced features	None	Implemented Target Product: display stats and view sessions. Converted minutes to hours/minutes format.

#### 4. (55pts) Final Product Description:

Include your proposed MVP, Target, and Reach versions.

- i. Minimum Viable Product (MVP): Simple command-line program to log study sessions with date, course, and minutes. Displays all sessions and saves them to a file.
- ii. Target Product: Includes statistics showing total study time and time per course for better study allocation feedback.
- iii. Reach Version: Filter by course, show recent sessions, display hours format, track consecutive study days.
- iv. (20pts) Description of final product including target audience, user story, problem statement, key features, technical details and technologies used. (100 – 150 words)

My Study Session Tracker helps students keep track of their study time between classes. It resolves the issue of not knowing how study time is allocated. Adding

sessions, viewing every session, and presenting statistics with breakdowns by course and overall are among the features. With structured output that converted minutes to hours, I reached the Target Product level. Uses a standard interface, datetime for validation, and Python's csv package. Between runs, data is in study\_sessions.csv.

- v. (20pts) Provide a YouTube link to your video demonstration (1–2 minutes, narrated). **Important Note:** Do not upload your video file directly. Instead, upload your video to YouTube and include the video link clearly here in your report. The level of difficulty and detail of the project should be reasonable for 10 hours of work with LLM support. The project should not be something an LLM can solve without significant effort by the developer. (Be sure to have someone else test that your link is working.)

<https://www.youtube.com/watch?v=2dkn4ZPIQ3Q>

- vi. (15pts) Any input files, coding files, and test files should be uploaded. Provide a list here of file names and purposes, or any links to live sites or artifacts. Remember code should also be commented. A README file should be created and uploaded so that we have the option to follow your instructions to run your project.

- **code/**
  - study\_tracker.py - Main program file containing the menu loop and main function
  - functions.py - Helper functions for menu options
- **data/**
  - study\_sessions.csv - Data file storing all logged study sessions
  - sample\_data.csv - Example data for testing
- **tests/**
  - test\_basic.py - Basic functionality tests
  - test\_cases.txt - Documentation of manual test cases performed
- **docs/**
  - screenshots/ - Screenshots of the program running
  - design\_notes.txt - Initial design sketches and planning notes
- **report/**
  - Personal\_Programming\_Project\_Report.pdf
- **README.md** - Comprehensive documentation including setup, usage instructions, and video link

### **Project Repository & Code Submission Details:**

I have invited the assigned GTA to the repository as a Maintainer as we had done during week 6 classwork activity. Assuming the “collaborator” role was for those using GitHub since that option was not available on GitLab.

Repository should be well-organized, documented, and easy to navigate. At a minimum, include the following structure:

- **code/** – All source code files for your project (organized by component or module if applicable).
- **data/** – Any input files, datasets, or configuration files used by your program.
- **tests/** – Test scripts or files demonstrating how your code was verified.
- **docs/** – Supporting materials such as screenshots, reports, or documentation.
- **report/** This final report document.
- **README.md** – A detailed file describing:
  - Project overview and purpose
  - Video link of your project
  - Installation and setup instructions
  - How to run the program and reproduce results
  - Technologies or libraries used
  - Author(s) and contribution summary

#### **Required:**

- Maintain a logical directory structure, do not store all files at the root level.
- Include comments in your code to explain logic and design decisions.
- Keep your repository **private** until grades are released, then you may make it public.

Share access with the following personnel (Add them as collaborators):

<b>GTA Name</b>	<b>Section</b>	<b>Professor</b>
Mona Moghadampanah	83484	P. Sullivan
Yue Shen	83485	O. Emebo
Abdullah Al Noman	83486	O. Emebo

Suraj Vishwanath	83487	P. Sullivan
Juno Bartsch	91578	S. Nizamani

5. **(10pts) Consultation and Use of LLMs:**

Each student must create a unique project but is allowed to consult with other people and use Large Language Models (LLMs). Describe how you incorporated these resources into your project:

- **Consultation Description:**

Did not get any feedback from online communities but showcased project to my peers and for testing purposes. Went to office hours to consult whether my idea and the scope of the project.

- **Use of LLMs:**

I used ChatGPT for limited debugging and testing support, such as understanding error messages and suggesting fixes or additional test cases.