TaskFlow – A CLI Task Manager –

**Project Overview:**

TaskFlow is a command line interface (CLI) application designed to help users manage their daily tasks efficiently. It integrates with external APIs to provide a unique user experience.

**Key Features:**

1. Task Management:
   1. Add, delete, and view tasks.
   2. Sort tasks by priority: “low”, “medium”, “hard.”
   3. Mark tasks as “completed” and automatically remove them from the to-do list and send them to archive.
   4. At the ends of the day, choose to either delete unfinished tasks or carry them over to the next day.
2. Task Categorization:
   1. By default, you have [work, personal, urgent]
   2. Later you can add or delete categories.
3. Task archives
   1. Archive completed tasks for future reference.
   2. Restore tasks from the archive if needed.
4. API integration:
   1. Display a motivational quote each time the program runs.
   2. Show the current date and time using and API, which customizable formats.
5. Customization:
   1. Configure settings through JSON file.
   2. Customize date and time formats (e.g. DD/MM/YYYY or MM/DD/YY)
   3. Choose between a 12-hours or 24-hour clock.
   4. Personalize the color scheme of the text using hex values.

**Getting Started:**

* Language: Java (or whatever the language you want to create with. It doesn’t really matter.)
* Requirements: Basic understanding of programming, arrays, file handling, enough motivation to go through and complete this project.
* Tools: JDK (if, you choose to use java), JSON library, some other library that can make your code executable.

**Guidance:**

I will give you the bare minimum guidance. I want you personally to come up with all the things. Trust me, if you receive this project, that means that I trust you and believe you that you can do this. That being said, don’t hesitate to reach out to me, or look up online. DO NOT USE ANY AI APPLICATION, it will steal your opportunity to grow.

1. Think of the architecture, what kind of class, files, would you have? What kind of parent class can you have, so that it makes your life “less” miserable? What is the commonality (or redundant parts) among this project?
2. Start little, make the bare minimum, expand from there.
3. Use test-driven-development. Don’t know what it is? Well, look up what it is.
4. Focus on making the application functional first. Then add extra features like customization and api integration.

**Little bit of hint:**

So, I wanna take time to talk about commands that you’ll be making for this project.

Normally speaking, when we are making commands for CLI project, we follow these two rules:

1. If the command is just a letter, like “p” for priority, or “d” for due date, we use “-“to indicate that that is one letter command, so it’ll look like “-p” for priority, “-d” for due date, etc.
2. If it’s a one word, then we’ll use “- -“instead of “-“ so two of them (we don’t have space in between in commands, but I had to have it to make it obvious.) So, it’ll be “- - help” or “- - version” or stuff like that.

Here are the commands that I think can be implemented:

* --add
  + Adds a new task with an optional category, priority, and due date.
  + Example: --add “Buy groceries” -c “personal” -p “medium” -d “2024-08-30”
* --del
  + Sends task to archive by its ID. Use –force to delete the task completely.
  + Example: --del 345j –force
* --edit
  + Edits the details of an existing task.
  + Example: edit 345j -n “Buy groceries and medicine” -p “high”
* --view
  + View all task or filters them by category, priority, or completion status.
  + Example: --view -c “work”, --view -p “high”, --view -s “completed”
* --complete
  + Marks a task as completed.
  + Example: --complete 345j
* --endday
  + At the end of the day, prompts the user to delete or carry over incomplete tasks.
  + Example: --endday
* --cat
  + Category commands. Use -a to add, -e to edit, -v to list them.
  + Example: cat -a “school”, cat -e “urgent” -n “high priority”, cat -l
* --archive
  + Moves completed tasks to the archive, views archived tasks, or restores tasks from archive.
  + Example: --archive –add 345j, --archive -v, --archive –restore 345j
* --config
  + Commands for comigration. Use -v to view the config, -e to edit, etc.
  + Example: --config -v, --config -e -d “MM/DD/YY” -t “12-hour -c “#FF0000”
* --help
  + Displays a list of all available commands with their description.
  + Example: --help
* --exit
  + Exit the program.
  + Example: --exit

If you have any question, please do not hesitate to contact me.