



Programming Language

CLI

Filesystem

Beginner

Task Tracker

Build a CLI app to track your tasks and manage your to-do list.

Start building, submit solution and get feedback from the community.

Start Working

2

Submit Solution



5 upvotes



10 upvotes

Task tracker is a project used to track and manage your tasks. In this task, you will build a simple command line interface (CLI) to track what you need to do, what you have done, and what you are currently working on. This project will help you practice your programming skills, including working with the filesystem, handling user inputs, and building a simple CLI application.

Requirements

The application should run from the command line, accept user actions and inputs as arguments, and store the tasks in a JSON file. The user should be able to:

- Add, Update, and Delete tasks
- Mark a task as in progress or done
- List all tasks
- List all tasks that are done
- List all tasks that are not done
- List all tasks that are in progress

Here are some constraints to guide the implementation:

- You can use any programming language to build this project.
- Use positional arguments in command line to accept user inputs.
- Use a JSON file to store the tasks in the current directory.
- The JSON file should be created if it does not exist.
- Use the native file system module of your programming language to interact with the JSON file.
- Do not use any external libraries or frameworks to build this project.
- Ensure to handle errors and edge cases gracefully.

Example

The list of commands and their usage is given below:

```
# Adding a new task
task-cli add "Buy groceries"
# Output: Task added successfully (ID: 1)

# Updating and deleting tasks
task-cli update 1 "Buy groceries and cook dinner"
task-cli delete 1

# Marking a task as in progress or done
task-cli mark-in-progress 1
task-cli mark-done 1

# Listing all tasks
task-cli list

# Listing tasks by status
task-cli list done
task-cli list todo
task-cli list in-progress
```

Task Properties

Each task should have the following properties:

- `id`: A unique identifier for the task
- `description`: A short description of the task
- `status`: The status of the task (todo, in-progress, done)

- `createdAt`: The date and time when the task was created
- `updatedAt`: The date and time when the task was last updated

Make sure to add these properties to the JSON file when adding a new task and update them when updating a task.

Getting Started

Here are a few steps to help you get started with the Task Tracker CLI project:

Set Up Your Development Environment

- Choose a programming language you are comfortable with (e.g., Python, JavaScript, etc.).
- Ensure you have a code editor or IDE installed (e.g., VSCode, PyCharm).

Project Initialization

- Create a new project directory for your Task Tracker CLI.
- Initialize a version control system (e.g., Git) to manage your project.

Implementing Features

- Start by creating a basic CLI structure to handle user inputs.
- Implement each feature one by one, ensuring to test thoroughly before moving to the next e.g. implement adding task functionality first, listing next, then updating, marking as in progress, etc.

Testing and Debugging

- Test each feature individually to ensure they work as expected. Look at the JSON file to verify that the tasks are being stored correctly.
- Debug any issues that arise during development.


Finalizing the Project

- Ensure all features are implemented and tested.
- Clean up your code and add comments where necessary.

- Write a good readme file on how to use your Task Tracker CLI.

By the end of this project, you will have developed a practical tool that can help you or others manage tasks efficiently. This project lays a solid foundation for more advanced programming projects and real-world applications.

Happy coding!

 Found a mistake? [Help us improve.](#)

Join the Community

roadmap.sh is the [7th most starred project on GitHub](#) and is visited by hundreds of thousands of developers every month.

Rank 7th [out of 28M!](#)

309K

GitHub Stars

★ **Star us on GitHub**

Help us reach #1

+90k [every month](#)

+1.5M

Registered Users

 **Register yourself**

Commit to your growth

+2k [every month](#)

33K

Discord Members

 **Join on Discord**

Join the community

[Roadmaps](#) [Best Practices](#) [Guides](#) [Videos](#) [FAQs](#) [YouTube](#)

 roadmap.sh by @kamrify

Community created roadmaps, best practices, projects, articles, resources and journeys to help you choose your path and grow in your career.

© roadmap.sh · [Terms](#) · [Privacy](#) · [Advertise](#) ·   

THE NEW STACK

The top DevOps resource for Kubernetes, cloud-native computing, and large-scale development and deployment.

[DevOps](#) · [Kubernetes](#) · [Cloud-Native](#)