**Summary**

The task manager application allows users to add, view, and manage tasks with a simple user interface. The application is designed to work seamlessly on both desktop and mobile screens. Here's a breakdown of its key features and structure:

* Add Task: Users can input a task title and description. Upon clicking "Add Task," the task is added to a list with a default status of "In Progress."
* Task List: The added tasks are displayed below the input form. Each task has its own card, showing the task title, description, and status.
* Responsive Layout: The layout adapts to different screen sizes, ensuring a consistent experience across mobile and desktop devices.
* Styling: The page has a simple yet modern design with background and task cards styled for easy readability and user interaction.

**Core logic**

State Management:

* The app uses React's useState hook to manage the state of tasks and the new task form's inputs (newTitle and newDescription).
* When a task is added, the task data (including title, description, and status) is stored in the tasks state array.

Add Task Functionality:

* The handleAddTask function is triggered when the user clicks the "Add Task" button.
* It checks if both newTitle and newDescription are filled. If valid, it:
  + Adds the task to the tasks state array with a unique id based on Date.now().
  + Resets the input fields to empty strings (newTitle and newDescription).

Displaying Tasks:

* The tasks are mapped over using the tasks.map() function and passed as props to the TaskCard component.
* Each TaskCard is responsible for displaying the task information (title, description, and status).

Styling:

* The page is styled using CSS with a focus on creating a responsive and user-friendly layout.
* The background and task cards are styled to improve user interaction, with a clean, minimal look.
* The page and input elements are designed to be mobile-responsive and centered on all screen sizes.

Responsive Design:

* The use of relative units (such as percentages and rem) and Flexbox ensures that the layout adapts to various screen sizes, including mobile devices.
* An optional media query can be added to further optimize the app for mobile screens.





