Level 2: Intermediate

Task 3: Create a console application for basic CRUD operations on a list of tasks.

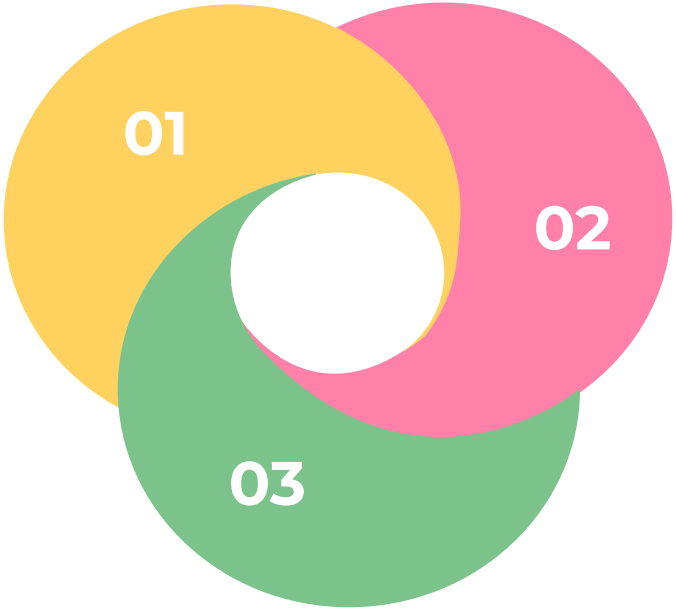
By Shradha Pujari

Presentation on Task Management

An overview of a console application for basic CRUD operations on a list of tasks

Steps:

Define a Task class with necessary attributes.

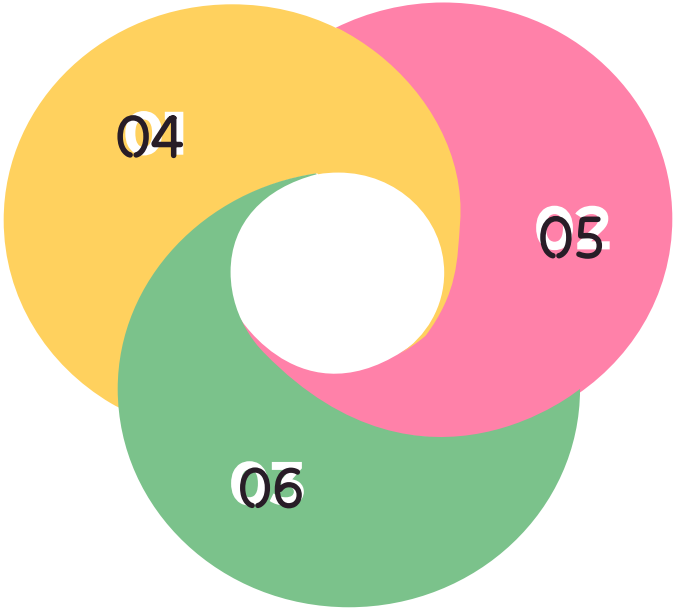


Implement functionality to create a new task

Develop a method to read and display tasks.

Steps

Allow users to update task details.



Provide an option to delete tasks.

Test the application with various scenarios

Defining the Task Class

Task Class: The foundation of our application.

Attributes:

- **title (String):** A concise description of the task.
- **description (String):** Optional, provides additional details about the task.
- **completed (Boolean):** Indicates whether the task is completed (True) or pending (False).

Code:

Python

```python
class Task:
  def __init__(self, title, description):
    self.title = title
    self.description = description
    self.done = False  # Flag to mark completed tasks

  def mark_done(self):
    self.done = True

  def __str__(self):
    status = "Done" if self.done else "Pending"
    return f"- {self.title} ({status}) - \n{self.description}\n"
```

Output

Initial Menu:

```
Task Manager
1. Create Task
2. View Tasks
3. Update Task
4. Delete Task
5. Mark Task Done
6. Exit

Enter your choice (1-6): 1
```

Create Task

01 Adding task to the list

02 Functionality to create a new task

03 Input validation

Code

Python

```python
def create_task():
  title = input("Enter task title: ")
  description = input("Enter task description (optional): ")
  return Task(title, description)
```

Output

```
Enter task title: Grocery Shopping
Enter task description (optional): Buy milk, bread, and eggs
Task created successfully!

Task Manager
1. Create Task
2. View Tasks
3. Update Task
4. Delete Task
5. Mark Task Done
6. Exit

Enter your choice (1-6): 2
```

Read Task

01 Viewing individual task details

02 Functionality to display all tasks

03 Sorting and filtering options

Code

Python

```python
def display_tasks(tasks):
  if not tasks:
    print("There are no tasks currently.")
  else:
    print("\nYour Tasks:")
    for task in tasks:
      print(task)
```

Output

```
Your Tasks:
- Grocery Shopping (Pending) -
Buy milk, bread, and eggs

Task Manager
1. Create Task
2. View Tasks
3. Update Task
4. Delete Task
5. Mark Task Done
6. Exit

Enter your choice (1-6): 3
```

Update Task

- Functionality to update task details
- Editing task properties
- Validating updated task

Code

```
def update_task(tasks):
if not tasks:
print("There are no tasks to update.")
return display_tasks(tasks)
task_index = int(input("Enter the number of the task to update: ")) - 1
 if task_index < 0 or task_index >= len(tasks):
print("Invalid task number.")
 return task = tasks[task_index]
new_title = input("Enter a new title (or leave blank to keep current): ")
if new_title:
task.title = new_title new_description = input("Enter a new description (or leave blank to
keep current): ")
 if new_description:
task.description = new_description print("Task updated successfully!")
```

Output

Updating a Task:

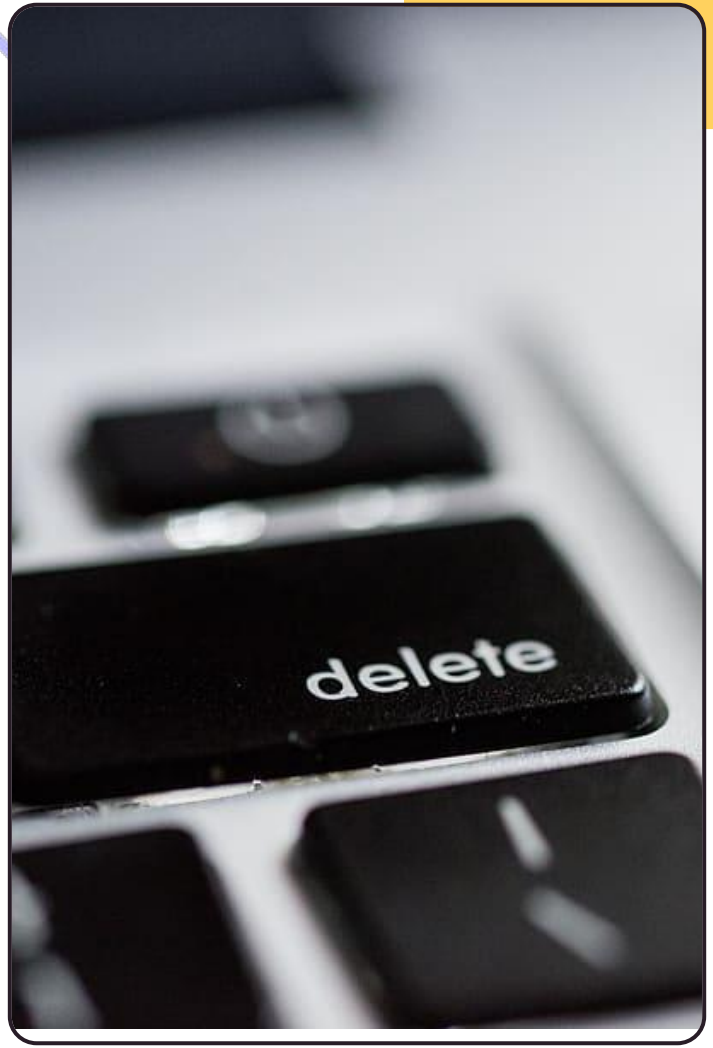
```
Your Tasks:
- Grocery Shopping (Pending) -
Buy milk, bread, and eggs

Enter the number of the task to update: 1

Task updated successfully!

Task Manager
1. Create Task
2. View Tasks
3. Update Task
4. Delete Task
5. Mark Task Done
6. Exit
```

Delete Task

01 Removing task from the list

02 Functionality to delete a task

03 Confirmation prompt



Code

```
def delete_task(tasks):
  if not tasks:
    print("There are no tasks to delete.")
    return

  display_tasks(tasks)
  task_index = int(input("Enter the number of the task to delete: ")) - 1

  if task_index < 0 or task_index >= len(tasks):
    print("Invalid task number.")
    return

  del tasks[task_index]
  print("Task deleted successfully!")
```

Output

```
Your Tasks:
- Grocery Shopping (Pending) -
Buy milk, bread, and eggs

Enter the number of the task to delete: 1

Task deleted successfully!

Task Manager
1. Create Task
2. View Tasks
3. Update Task
4. Delete Task
5. Mark Task Done
6. Exit
```

Testing the Application

Importance of Testing: Ensures the application functions as intended.

Test Scenarios:

- Adding new tasks with varying title lengths and descriptions.
- Reading and displaying the task list.
- Updating task details (title, description, completion status).
- Deleting tasks and confirming the deletion.
- Handling edge cases (empty titles, invalid user input).

Code

```
def main():
  tasks = []  # List to store tasks

  while True:
    print("\nTask Manager")
    print("1. Create Task")
    print("2. View Tasks")
    print("3. Update Task")
    print("4. Delete Task")
    print("5. Mark Task Done")
    print("6. Exit")

    choice = input("Enter your choice (1-6): ")

    if choice == '1':
      tasks.append(create_task())
      print("Task created successfully!")
```

Output

Viewing Tasks (after deleting the only task):

```
There are no tasks currently.

Task Manager
1. Create Task
2. View Tasks
3. Update Task
4. Delete Task
5. Mark Task Done
6. Exit

Enter your choice (1-6):
```

Thank you for your time! 😊