Major Project - Personal To-Do List Application

```
#code start here


import json

import os


# Task class representing a single task

class Task:

  def __init__(self, title, description, category):

    self.title = title

    self.description = description

    self.category = category

    self.completed = False


  def mark_completed(self):

    self.completed = True


  def to_dict(self):

    # Convert task object to dictionary for saving as JSON

    return {

      'title': self.title,

      'description': self.description,

      'category': self.category,

      'completed': self.completed

    }


  @staticmethod
```

```
    def from_dict(data):
        # Create a task object from a dictionary (used when loading tasks)
        task = Task(data['title'], data['description'], data['category'])
        task.completed = data['completed']
        return task

# Function to save tasks to a JSON file
def save_tasks(tasks, filename='tasks.json'):
    with open(filename, 'w') as f:
        json.dump([task.to_dict() for task in tasks], f, indent=4)
    print("Tasks have been saved successfully.")

# Function to load tasks from a JSON file
def load_tasks(filename='tasks.json'):
    if not os.path.exists(filename):
        return []
    with open(filename, 'r') as f:
        tasks_data = json.load(f)
        return [Task.from_dict(task) for task in tasks_data]

# Function to add a new task
def add_task(tasks):
    title = input("Enter task title: ").strip()
    description = input("Enter task description: ").strip()
    category = input("Enter task category (e.g., Work, Personal, Urgent): ").strip()
    task = Task(title, description, category)
    tasks.append(task)
    print(f"Task '{title}' added successfully.")

# Function to display all tasks
```

```
def view_tasks(tasks):
  if not tasks:
    print("No tasks to display.")
    return
  print("\nYour Tasks:")
  for idx, task in enumerate(tasks, 1):
    status = "✅" if task.completed else "❌"
    print(f"{idx}. [{status}] {task.title} - {task.category}")
    print(f"   Description: {task.description}")
  print()

# Function to mark a task as completed
def mark_task_completed(tasks):
  view_tasks(tasks)
  if not tasks:
    return
  try:
    choice = int(input("Enter the task number to mark as completed: "))
    if 1 <= choice <= len(tasks):
      tasks[choice - 1].mark_completed()
      print(f"Task '{tasks[choice - 1].title}' marked as completed.")
    else:
      print("Invalid task number.")
  except ValueError:
    print("Please enter a valid number.")

# Function to delete a task
def delete_task(tasks):
  view_tasks(tasks)
```

```
    if not tasks:
        return
    try:
        choice = int(input("Enter the task number to delete: "))
        if 1 <= choice <= len(tasks):
            removed_task = tasks.pop(choice - 1)
            print(f"Task '{removed_task.title}' deleted successfully.")
        else:
            print("Invalid task number.")
    except ValueError:
        print("Please enter a valid number.")

# Main function to display menu and interact with the user
def main():
    tasks = load_tasks()  # Load tasks from the JSON file when the app starts
    while True:
        print("\n--- Personal To-Do List ---")
        print("1. Add Task")
        print("2. View Tasks")
        print("3. Mark Task as Completed")
        print("4. Delete Task")
        print("5. Exit")
        choice = input("Choose an option (1-5): ").strip()

        if choice == '1':
            add_task(tasks)
        elif choice == '2':
            view_tasks(tasks)
        elif choice == '3':
            mark_task_completed(tasks)
        elif choice == '4':
            delete_task(tasks)
        elif choice == '5':
            save_tasks(tasks)  # Save tasks to the JSON file before exiting
            print("Exiting the application. Goodbye!")
            break
        else:
            print("Invalid choice. Please select a valid option.")


if __name__ == "__main__":
    main()
```