main.cpp

Share Run

```cpp
#include <iostream>
#include <vector>
#include <string>

struct Task {
    std::string description;
    bool completed;

    Task(const std::string& desc) : description(desc), completed(false) {}
};


void displayMenu() {
    std::cout << "\n==== TO-DO LIST MENU ====\n";
    std::cout << "1. Add Task\n";
    std::cout << "2. View Tasks\n";
    std::cout << "3. Mark Task as Completed\n";
    std::cout << "4. Remove Task\n";
    std::cout << "5. Exit\n";
    std::cout << "Choose an option (1-5): ";
}


void addTask(std::vector<Task>& tasks) {
    std::string description;
    std::cin.ignore(); // clear buffer
    std::cout << "Enter the task description: ";
    std::getline(std::cin, description);
    tasks.emplace_back(description);
    std::cout << "Task added successfully.\n";
}


void viewTasks(const std::vector<Task>& tasks) {
```

```cpp
void viewTasks(const std::vector<Task>& tasks) {
    if (tasks.empty()) {
        std::cout << "No tasks in the list.\n";
        return;
    }

    std::cout << "\nYour Tasks:\n";
    for (size_t i = 0; i < tasks.size(); ++i) {
        std::cout << i + 1 << ". [" << (tasks[i].completed ? "✓" : " ") << "] "
                  << tasks[i].description << "\n";
    }
}


void markTaskCompleted(std::vector<Task>& tasks) {
    if (tasks.empty()) {
        std::cout << "No tasks to mark as completed.\n";
        return;
    }

    int index;
    viewTasks(tasks);
    std::cout << "Enter the task number to mark as completed: ";
    std::cin >> index;

    if (index < 1 || index > static_cast<int>(tasks.size())) {
        std::cout << "Invalid task number.\n";
    } else {
        tasks[index - 1].completed = true;
        std::cout << "Task marked as completed.\n";
    }
}

```

```
64      }
65  }
66
67
68  void removeTask(std::vector<Task>& tasks) {
69      if (tasks.empty()) {
70          std::cout << "No tasks to remove.\n";
71          return;
72      }
73
74      int index;
75      viewTasks(tasks);
76      std::cout << "Enter the task number to remove: ";
77      std::cin >> index;
78
79      if (index < 1 || index > static_cast<int>(tasks.size())) {
80          std::cout << "Invalid task number.\n";
81      } else {
82          tasks.erase(tasks.begin() + index - 1);
83          std::cout << "Task removed successfully.\n";
84      }
85  }
86
87
88  int main() {
89      std::vector<Task> tasks;
90      int choice;
91
92      do {
93          displayMenu();
94          std::cin >> choice;
95
96          switch (choice) {
97              case 1:
```

main.cpp

Share Run

```
87
88  int main() {
89      std::vector<Task> tasks;
90      int choice;
91
92      do {
93          displayMenu();
94          std::cin >> choice;
95
96          switch (choice) {
97              case 1:
98                  addTask(tasks);
99                  break;
100             case 2:
101                 viewTasks(tasks);
102                 break;
103             case 3:
104                 markTaskCompleted(tasks);
105                 break;
106             case 4:
107                 removeTask(tasks);
108                 break;
109             case 5:
110                 std::cout << "Exiting To-Do List Manager. Goodbye!\n";
111                 break;
112             default:
113                 std::cout << "Invalid choice. Please try again.\n";
114         }
115
116     } while (choice != 5);
117
118     return 0;
119 }
```

Output

Clear

```
==== TO-DO LIST MENU ====
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Remove Task
5. Exit
Choose an option (1-5): 1
Enter the task description: update
Task added successfully.

==== TO-DO LIST MENU ====
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Remove Task
5. Exit
Choose an option (1-5): 3

Your Tasks:
1. [ ] update
Enter the task number to mark as completed: 1
Task marked as completed.

==== TO-DO LIST MENU ====
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Remove Task
5. Exit
Choose an option (1-5): 5
5
Exiting To-Do List Manager. Goodbye!
```