

Curd.cs

```
1  using System;
2  using System.Collections.Generic;
3
4  class Program
5  {
6      static List<Task> tasks = new List<Task>();
7
8      static void Main(string[] args)
9      {
10         bool exit = false;
11
12         do
13         {
14             Console.WriteLine("Task List Application\n");
15             Console.WriteLine("1. Create a task");
16             Console.WriteLine("2. Read tasks");
17             Console.WriteLine("3. Update a task");
18             Console.WriteLine("4. Delete a task");
19             Console.WriteLine("5. Exit\n");
20
21             Console.Write("Enter your choice: ");
22             int choice = Convert.ToInt32(Console.ReadLine());
23
24             switch (choice)
25             {
26                 case 1:
27                     CreateTask();
28                     break;
29                 case 2:
30                     ReadTasks();
31                     break;
32                 case 3:
33                     UpdateTask();
34                     break;
35                 case 4:
36                     DeleteTask();
37                     break;
38                 case 5:
39                     exit = true;
40                     break;
41                 default:
42                     Console.WriteLine("Invalid choice. Please try again.");
43                     break;
44             }
45
46             Console.WriteLine("\nPress any key to continue...");
47             Console.ReadKey();
48             Console.Clear();
49
50         } while (!exit);
51     }
52
53     static void CreateTask()
54     {
55         Console.WriteLine("Create a Task\n");
56
57         Console.Write("Enter task title: ");
```

```
58     string title = Console.ReadLine();
59
60     tasks.Add(new Task(title));
61     Console.WriteLine("Task created successfully!");
62 }
63
64 static void ReadTasks()
65 {
66     Console.WriteLine("Task List\n");
67
68     if (tasks.Count == 0)
69     {
70         Console.WriteLine("No tasks found!");
71     }
72     else
73     {
74         foreach (var task in tasks)
75         {
76             Console.WriteLine($"Title: {task.Title}");
77         }
78     }
79 }
80
81 static void UpdateTask()
82 {
83     Console.WriteLine("Update a Task\n");
84
85     if (tasks.Count == 0)
86     {
87         Console.WriteLine("No tasks found!");
88         return;
89     }
90
91     Console.Write("Enter the title of the task to update: ");
92     string title = Console.ReadLine();
93
94     Task taskToUpdate = tasks.Find(t => t.Title == title);
95
96     if (taskToUpdate == null)
97     {
98         Console.WriteLine("Task not found!");
99         return;
100     }
101
102     Console.Write("Enter new title for the task: ");
103     string newTitle = Console.ReadLine();
104
105     taskToUpdate.Title = newTitle;
106     Console.WriteLine("Task updated successfully!");
107 }
108
109 static void DeleteTask()
110 {
111     Console.WriteLine("Delete a Task\n");
112
113     if (tasks.Count == 0)
114     {
115         Console.WriteLine("No tasks found!");
116         return;
117     }
```

```
118
119     Console.Write("Enter the title of the task to delete: ");
120     string title = Console.ReadLine();
121
122     Task taskToDelete = tasks.Find(t => t.Title == title);
123
124     if (taskToDelete == null)
125     {
126         Console.WriteLine("Task not found!");
127         return;
128     }
129
130     tasks.Remove(taskToDelete);
131     Console.WriteLine("Task deleted successfully!");
132 }
133 }
134
135 class Task
136 {
137     public string Title { get; set; }
138
139     public Task(string title)
140     {
141         Title = title;
142     }
143 }
144
```