

Luis Palma

Felipe Monsalvo

Github Link: <https://github.com/FelipeMonsalvo/Lab2>

## Task 1:

```
C Program.cs x
AiConsoleApp > C Program.cs > ⌂ Program > ⌂ <top-level-statements-entry-point>
1 // Felipe Monsalvo and Luis Palma - CP5+3330+01
2 using OpenAI;
3 using OpenAI.Chat;
4 string apiKey = "sk-proj-dDEPKv0V2_mppmodiGxKcWvxDGfTe0IpsmJBWig-7Uvc0Dy3fx103XNjd6k2J6fRSUxtTm0T3B1bkFJPsoz0ygnI-dLnnCt147P0qan";
5 ChatClient client = new(model: "gpt-4o", apiKey: apiKey);
6 ChatCompletion completion = client.CompleteChat("Write a short story about a happy dog");
7 Console.WriteLine(completion.Content[0].Text);
8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
zsh - AiConsoleApp + - | ... | x
felipe@MacBookPro Lab2 % cd AiConsoleApp
felipe@MacBookPro AiConsoleApp % dotnet run
Once upon a time in a quaint little village nestled among rolling hills, there lived a dog named Benny. Benny was a golden retriever with fur as bright and cheery as the morning sun. His tail wagged with the enthusiasm of a thousand happy thoughts, and his eyes sparkled with an unquenchable zest for life.

Benny's home was a cozy cottage owned by an elderly woman named Mrs. Thompson, who adored him like family. Mrs. Thompson had a garden filled with vibrant blooms that attracted butterflies and bees, and Benny loved nothing more than to frolic among the colorful flowers, chasing the fluttering wings of his carefree companions.

Every morning, Benny and Mrs. Thompson went for a walk through the village, a ritual that brought joy not only to them but also to the villagers. Benny was the unofficial ambassador of happiness. He greeted everyone with a wag and playful bark, spreading cheer wherever his paws took him. The children adored him, often running out of their homes to envelop him in hugs while Benny showered them with affectionate licks.

One sunny afternoon, while exploring the woods behind the cottage, Benny stumbled upon something unusual – an old, dusty box partially hidden beneath some fallen leaves. Curious, Benny nudged the box until it flipped open, revealing an assortment of strange objects: a worn-out tennis ball, a chewed-up rope, and a small notebook with faded pages.

Benny sniffed the contents curiously, wagging his tail in satisfaction at the scent of familiar pup nostalgia. The notebook, he noticed, had little sketches and notes that looked like they belonged to a kid. It was filled with stories of secret adventures and treasures, imaginative tales that mirrored Benny's spirit of exploration. Benny barked excitedly, as if understanding the preciousness of a child's dreams tucked away in this forgotten diary.

Benny carried the tennis ball back to Mrs. Thompson, who chuckled warmly at his find. "Why, look at you, Benny!" she said, patting his head, "bringing back treasures from the past!" She gently placed the notebook inside her pocket, deciding it was a sign to organize a village activity she'd been pondering for a while – a treasure hunt for the village children, inspired by the child's whimsical tales.

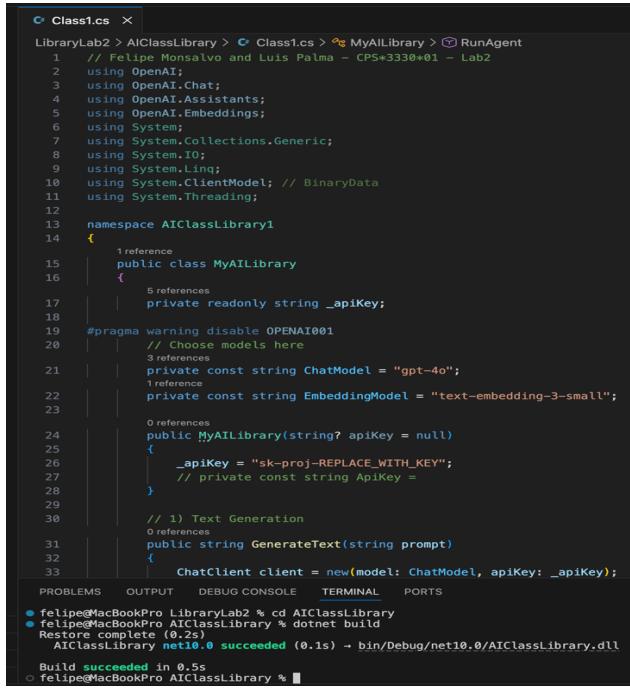
Over the next weekend, the village buzzed with excitement as Benny led the pack of eager children through woods and fields, their laughter and shouts echoing in harmony with the rustling leaves. Benny was in his element, guiding them to hidden spots, using the clues from the notebook. Each child found a toy or trinket, but more importantly, they discovered a shared adventure and joy that would linger in their hearts for years.

When the sun began to set, painting the sky in hues of orange and pink, the children gathered around Benny, showering him with hugs and treats. Mrs. Thompson watched with a heart full of gratitude, knowing that distinct sort of happiness that Benny carried was the real treasure all along.

And so, Benny fell asleep that night tired but blissfully happy, dreaming of more adventures and the smiles he so loved to see. For in that small village, Benny was more than a dog; he was a beacon of joy and the best friend one could ever hope to have.

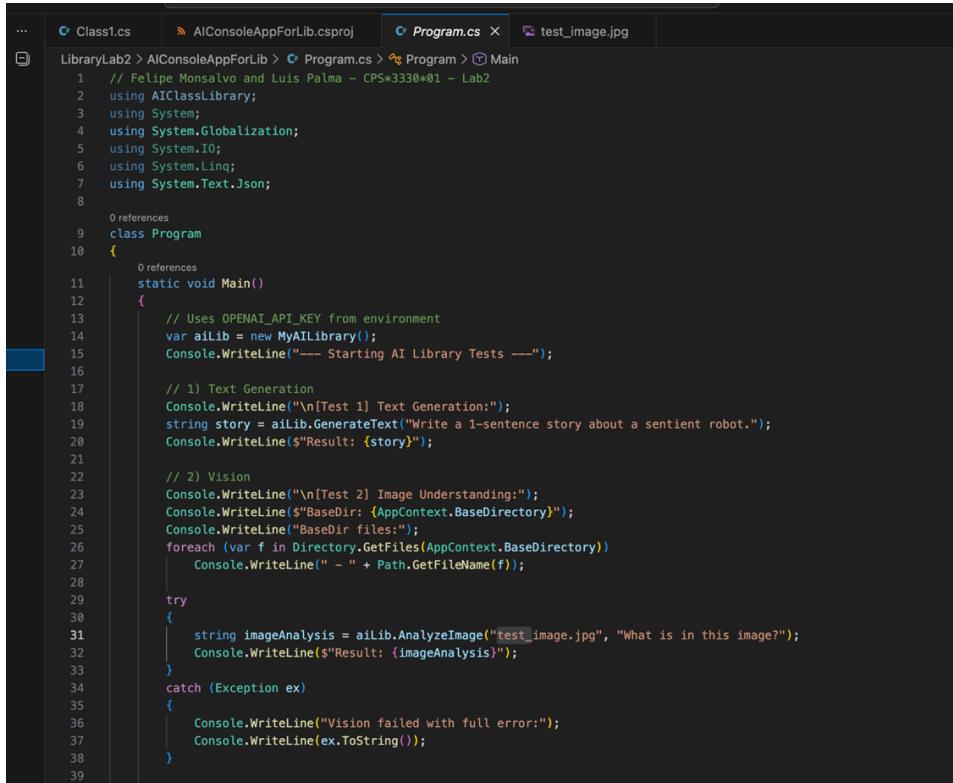
felipe@MacBookPro AiConsoleApp % ]
```

## Task 2:



```
Class1.cs
LibraryLab2 > AIClassLibrary > Class1.cs > MyAILibrary > RunAgent
1 // Felipe Monsalvo and Luis Palma - CPS*3330*01 - Lab2
2 using OpenAI;
3 using OpenAI.Chat;
4 using OpenAI.Assistants;
5 using OpenAI.Embeddings;
6 using System;
7 using System.Collections.Generic;
8 using System.IO;
9 using System.Linq;
10 using System.ClientModel; // BinaryData
11 using System.Threading;
12
13 namespace AIClassLibrary1
14 {
15     reference
16     public class MyAILibrary
17     {
18         5 references
19         private readonly string _apiKey;
20
21 #pragma warning disable OPENAI001
22         // Choose models here
23         3 references
24         private const string ChatModel = "gpt-4o";
25         1 reference
26         private const string EmbeddingModel = "text-embedding-3-small";
27
28         0 references
29         public MyAILibrary(string? apiKey = null)
30         {
31             _apiKey = "sk-proj-REPLACE_WITH_KEY";
32             // private const string ApiKey =
33         }
34
35         // 1) Text Generation
36         0 references
37         public string GenerateText(string prompt)
38         {
39             ChatClient client = new(model: ChatModel, apiKey: _apiKey);
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
felipe@MacBookPro LibraryLab2 % cd AIClassLibrary
felipe@MacBookPro AIClassLibrary % dotnet build
  References: 10
    AIClassLibrary net10.0 succeeded (0.1s) - bin/Debug/net10.0/AIClassLibrary.dll
Build succeeded in 0.5s
felipe@MacBookPro AIClassLibrary %
```

## Task 3:



```
... Class1.cs Program.cs test_image.jpg
LibraryLab2 > AIConsoleAppForLib > Program.cs > Program > Main
1 // Felipe Monsalvo and Luis Palma - CPS*3330*01 - Lab2
2 using AIClassLibrary;
3 using System;
4 using System.Globalization;
5 using System.IO;
6 using System.Linq;
7 using System.Text.Json;
8
9 class Program
10 {
11     0 references
12     static void Main()
13     {
14         // Uses OPENAI_API_KEY from environment
15         var aiLib = new MyAILibrary();
16         Console.WriteLine("--- Starting AI Library Tests ---");
17
18         // 1) Text Generation
19         Console.WriteLine("\n[Test 1] Text Generation:");
20         string story = aiLib.GenerateText("Write a 1-sentence story about a sentient robot.");
21         Console.WriteLine($"Result: {story}");
22
23         // 2) Vision
24         Console.WriteLine("\n[Test 2] Image Understanding:");
25         Console.WriteLine($"BaseDir: {ApplicationContext.BaseDirectory}");
26         Console.WriteLine("BaseDir files:");
27         foreach (var f in Directory.GetFiles(ApplicationContext.BaseDirectory))
28             Console.WriteLine("- " + Path.GetFileName(f));
29
30         try
31         {
32             string imageAnalysis = aiLib.AnalyzeImage("test_image.jpg", "What is in this image?");
33             Console.WriteLine($"Result: {imageAnalysis}");
34         }
35         catch (Exception ex)
36         {
37             Console.WriteLine("Vision failed with full error:");
38             Console.WriteLine(ex.ToString());
39         }
}
Build succeeded in 0.1s
felipe@MacBookPro AIConsoleAppForLib %
```

```
... Class1.cs AIConsoleAppForLib.csproj Program.cs x test_image.jpg
LibraryLab2 > AIConsoleAppForLib > Program.cs > Program > Main
9  class Program
11 static void Main()
40     // 3) Embeddings (print + save)
41     Console.WriteLine("\n[Test 3] Embeddings:");
42     var vec = aiLib.GetEmbeddings("Testing AI embeddings.");
43     Console.WriteLine($"Dims: {vec.Length}");
44     // Print ALL embeddings (1536 floats). If too much, change to 20.
45     PrintEmbedding(vec, perLine: 8, maxToPrint: vec.Length);
46     SaveEmbeddingJson("embedding.json", "Testing AI embeddings.", vec);
47     Console.WriteLine("Wrote embedding.json");
48
49     // Save multiple embeddings + cosine similarity matrix
50     var texts = new[]
51     {
52         "pizza in new jersey",
53         "newark airport",
54         "hiking in watchung reservation",
55         "machine learning embeddings"
56     };
57     SaveEmbeddingsJsonl("embeddings.jsonl", texts, aiLib);
58     Console.WriteLine("Wrote embeddings.jsonl");
59     Console.WriteLine("\nCosine similarity matrix:");
60     PrintCosineSimilarityMatrix(texts, aiLib);
61
62     // 4) Agent
63     Console.WriteLine("\n[Test 4] Agent Building:");
64     string agentReply = aiLib.RunAgent("You are a helpful travel guide.", "Suggest a place in NJ.");
65     Console.WriteLine(agentReply);
66
67     Console.WriteLine("\n---- Tests Completed ----");
68     Console.ReadLine();
69 }
70
1 reference
71 static void PrintEmbedding(ReadOnlyMemory<float> vec, int perLine = 8, int maxToPrint = 1536)
{
72     var span = vec.Span;
73     int n = Math.Min(span.Length, maxToPrint);
74     for (int i = 0; i < n; i++)
75     {
76         Console.Write($"{i}:{span[i].ToString("0.000000", CultureInfo.InvariantCulture)} ");
77     }
78 }
```

```
... Class1.cs AIConsoleAppForLib.csproj Program.cs x test_image.jpg
LibraryLab2 > AIConsoleAppForLib > Program.cs > Program > Main
9  class Program
71 static void PrintEmbedding(ReadOnlyMemory<float> vec, int perLine = 8, int maxToPrint = 1536)
72 {
73     if ((i + 1) % perLine == 0) Console.WriteLine();
74     Console.WriteLine();
75 }
76
1 reference
83 static void SaveEmbeddingJson(string path, string text, ReadOnlyMemory<float> vec)
{
84     var payload = new
85     {
86         text,
87         dimensions = vec.Length,
88         embedding = vec.ToArray()
89     };
90     File.WriteAllText(path, JsonSerializer.Serialize(payload, new JsonSerializerOptions
91     {
92         WriteIndented = true
93     }));
94 }
95
1 reference
97 static void SaveEmbeddingsJsonl(string path, string[] texts, MyAILibrary aiLib)
{
98     using var sw = new StreamWriter(path);
99     foreach (var t in texts)
100     {
101         var v = aiLib.GetEmbeddings(t).ToArray();
102         sw.WriteLine(JsonSerializer.Serialize(new { text = t, dimensions = v.Length, embedding = v }));
103     }
104 }
105
1 reference
107 static double Cosine(float[] a, float[] b)
{
108     if (a.Length != b.Length) throw new ArgumentException("Dimension mismatch.");
109     double dot = 0, na = 0, nb = 0;
110     for (int i = 0; i < a.Length; i++)
111     {
112         dot += (double)a[i] * b[i];
113     }
114 }
```

```
Class1.cs AIConsoleAppForLib.csproj Program.cs test_image.jpg
LibraryLab2 > AIConsoleAppForLib > Program.cs > Program > Main
9  class Program
107     static double Cosine(float[] a, float[] b)
114         na += (double)a[i] * a[i];
115         nb += (double)b[i] * b[i];
116     }
117     return dot / (Math.Sqrt(na) * Math.Sqrt(nb) + 1e-12);
118 }
119
120 // reference
121 static void PrintCosineSimilarityMatrix(string[] texts, MyAILibrary aiLib)
122 {
123     var emb = texts.Select(t => aiLib.GetEmbeddings(t).ToArray()).ToArray();
124     Console.WriteLine("".PadRight(28));
125     for (int j = 0; j < texts.Length; j++)
126         Console.WriteLine($"{{j}}".PadLeft(10));
127     Console.WriteLine();
128
129     for (int i = 0; i < texts.Length; i++)
130     {
131         string label = (texts[i].Length > 25) ? texts[i].Substring(0, 25) + "..." : texts[i];
132         Console.WriteLine(label.PadRight(28));
133         for (int j = 0; j < texts.Length; j++)
134         {
135             double c = Cosine(emb[i], emb[j]);
136             Console.WriteLine($"{c,10:0.0000}");
137         }
138     }
139 }
140
141
```

```
... Class1.cs AIConsoleAppForLib.csproj Program.cs test_image.jpg
LibraryLab2 > AIConsoleAppForLib > AIConsoleAppForLib.csproj
1 <Project Sdk="Microsoft.NET.Sdk">
2
3     <ItemGroup>
4         <ProjectReference Include=".\\AIClassLibrary\\AIClassLibrary.csproj" />
5     </ItemGroup>
6     <ItemGroup>
7         <None Include="test_image.jpg">
8             <CopyToOutputDirectory>Always</CopyToOutputDirectory>
9         </None>
10    </ItemGroup>
11
12    <PropertyGroup>
13        <OutputType>Exe</OutputType>
14        <TargetFramework>net10.0</TargetFramework>
15        <ImplicitUsings>enable</ImplicitUsings>
16        <Nullable>enable</Nullable>
17    </PropertyGroup>
18
19 </Project>
```

The screenshot shows a terminal window within a development environment. The title bar includes tabs for 'Class1.cs', 'AIConsoleAppForLib.csproj', 'Program.cs', and 'test\_image.jpg'. The terminal window displays the command 'dotnet run' being executed, followed by the output of the AI library tests. The output includes three test cases: Text Generation, Image Understanding, and Embeddings. The Text Generation result is a short story about a robot named Aria. The Image Understanding result is a description of a coastal landscape with rocky cliffs and a sandy beach. The Embeddings section shows numerical data for 1536 dimensions.

```
// Felipe Monsalvo and Luis Palma - CPS*3330*01 - Lab2
using OpenAI;
using OpenAI.Chat;
using OpenAI.Assistants;

felipe@MacBookPro AIConsoleAppForLib % dotnet run
--- Starting AI Library Tests ---

[Text 1] Text Generation:
Result: In a world where technology thrived, a sentient robot named Aria discovered the profound joy of music, composing symphonies that made even the most stoic humans weep.

[Text 2] Image Understanding:
BaseDir: /Users/felipe/Desktop/CPS 3330/Lab2/LibraryLab2/AIConsoleAppForLib/bin/Debug/net10.0/
BaseDir files:
- AIConsoleAppForLib.pdb
- System.ClientModel.dll
- Microsoft.Extensions.Logging.Abstractions.dll
- AIConsoleAppForLib.dll
- test_image.jpg
- AIConsoleAppForLib.runtimeconfig.json
- AIclassLibrary.pdb
- System.Memory.Data.dll
- AIConsoleAppForLib
- AIConsoleAppForLib.deps.json
- Microsoft.Extensions.DependencyInjection.Abstractions.dll
- AIclassLibrary.dll
- OpenAI.dll
[Vision] BaseDir: /Users/felipe/Desktop/CPS 3330/Lab2/LibraryLab2/AIConsoleAppForLib/bin/Debug/net10.0/
[Vision] CWD : /Users/felipe/Desktop/CPS 3330/Lab2/LibraryLab2/AIConsoleAppForLib/bin/Debug/net10.0/test_image.jpg
[Vision] Using : /Users/felipe/Desktop/CPS 3330/Lab2/LibraryLab2/AIConsoleAppForLib/bin/Debug/net10.0/test_image.jpg
Result: The image depicts a coastal landscape featuring rocky cliffs and a rugged shoreline. There's a sandy beach where waves are gently breaking, and the scene is set under a clear blue sky with scattered clouds. In the background, you can see distant hills or mountains. The foreground is covered with grass and rocky outcrops.

[Text 3] Embeddings:
Dims: 1536
0:0.002742 1:-0.029968 2:0.025448 3:-0.032820 4:-0.016334 5:-0.029185 6:0.008319 7:0.016473
8:-0.019099 9:-0.004156 10:-0.004118 11:-0.038854 12:-0.001023 13:-0.061248 14:-0.009322 15:0.000745
16:-0.010408 17:0.016233 18:0.013153 19:0.009228 20:-0.000912 21:-0.033881 22:-0.002326 23:0.029412
24:0.006031 25:-0.016852 26:0.034360 27:0.028099 28:0.012888 29:-0.042111 30:0.014479 31:-0.037138
32:0.036658 33:-0.010231 34:-0.030674 35:0.049407 36:-0.010635 37:0.020374 38:-0.033780 39:0.001847
40:0.053219 41:0.002914 42:-0.015337 43:0.025158 44:-0.003740 45:0.029286 46:-0.005343 47:-0.026837
48:0.003361 49:0.024287 50:-0.054684 51:0.004690 52:-0.057915 53:0.011260 54:-0.005363 55:0.067509
56:0.002583 57:0.011714 58:0.003579 59:-0.000839 60:0.008944 61:-0.004986 62:0.026437 63:0.001446
64:0.023896 65:-0.035547 66:-0.025613 67:0.045974 68:-0.004667 69:-0.017799 70:0.028680 71:0.000391
72:0.004024 73:-0.011361 74:0.007612 75:0.006905 76:-0.0061854 77:0.077406 78:-0.006365 79:-0.043449
80:-0.035118 81:0.032947 82:0.003437 83:-0.026458 84:-0.008899 85:-0.026862 86:-0.022810 87:-0.011127
88:-0.030296 89:-0.001794 90:-0.001376 91:-0.043197 92:-0.036355 93:-0.043929 94:0.028301 95:0.024691
96:0.031078 97:-0.014592 98:0.011897 99:0.024603 100:-0.004863 101:-0.000118 102:0.006100 103:0.030043
104:0.011569 105:-0.041480 106:0.007170 107:-0.051553 108:-0.052614 109:0.009764 110:-0.071801 111:-0.025650
112:-0.027620 113:0.002056 114:0.012497 115:-0.010604 116:-0.012450 117:0.024451 118:-0.012377 119:-0.033098
120:-0.037642 121:0.003664 122:0.052816 123:0.027563 124:-0.028831 125:-0.030397 126:-0.026484 127:-0.036001
128:-0.030725 129:-0.031104 130:-0.000094 131:-0.000770 132:0.045141 133:0.028200 134:-0.023454 135:-0.042894
```

The screenshot shows a Visual Studio Code interface with the following details:

- Project Structure:** The left sidebar shows a tree view of the project: LibraryLab2 > AIConsoleAppForLib > Class1.cs > MyAI Library > Program.cs.
- Terminal Output:** The main pane displays the following text:
  - Code snippets from Class1.cs and Program.cs.
  - Output of the command "dotnet run".
  - Logs indicating file writes ("Wrote embedding.json" and "Wrote embeddings.jsonl").
  - A cosine similarity matrix for entities like pizza in new jersey, Newark airport, hiking in Watchung Reserv..., and machine learning embeddin... (with values ranging from 0.0000 to 0.745).
  - A test log entry: "[Test 4] Agent Building: New Jersey offers a variety of attractions and destinations that cater to different interests. One highly recommended place to visit is the Delaware Water Gap National Recreation Area. This scenic area is perfect for those who enjoy outdoor activities and natural beauty."
  - Section headers: "## Delaware Water Gap National Recreation Area", "\*\*Location:\*\* Straddles the border of New Jersey and Pennsylvania.", "\*\*Highlights:\*\*".
  - A bulleted list of attractions and activities.
  - A summary sentence: "This area provides a peaceful retreat with ample opportunities for relaxation and adventure. Whether you're looking for a day trip or a longer stay, the Delaware Water Gap National Recreation Area is a fantastic choice in New Jersey."
  - A footer message: "---- Tests Completed ----".

#### Task 4 and 5:

Source Code:

Console:

```
Lab2 > Lab2 > AIQuoteConsole > C# Program.cs > ...
1  using System;
2  using AIQuoteLibrary;
3  using DotNetEnv;
4
5      0 references
6  class Program
7  {
8      0 references
9      static void Main()
10     {
11         Env.Load();
12
13         Console.WriteLine("AI Quote Console App");
14         Console.WriteLine("Created by Luis Palma and Felipe Monsalvo\n");
15
16         QuoteGenerator generator = new();
17         generator.PrintCreators();
18
19         Console.Write("\nEnter a topic: ");
20         string topic = Console.ReadLine() ?? "life";
21
22         string quote = generator.GenerateQuote(topic);
23
24         Console.WriteLine("\nGenerated Quote:");
25         Console.WriteLine(quote);
26     }
27 }
```

Library:

```
using OpenAI;
using OpenAI.Chat;
using System;

namespace AIQuoteLibrary
{
    1 reference
    public class QuoteGenerator
    {
        2 references
        private readonly string _apiKey;

        0 references
        public QuoteGenerator()
        {
            _apiKey = Environment.GetEnvironmentVariable("OPENAI_API_KEY")
                ?? throw new Exception("OPENAI_API_KEY not set.");
        }

        0 references
        public string GenerateQuote(string topic)
        {
            ChatClient client = new("gpt-4o", _apiKey);
            ChatCompletion completion = client.CompleteChat(
                $"Give me one short inspirational quote about {topic}.");

            return completion.Content[0].Text;
        }

        0 references
        public void PrintCreators()
        {
            Console.WriteLine("AI Quote Library by Luis Palma and Felipe G Monsalvo");
        }
    }
}
```

## Output

```
● (base) luispalma@MacBookPro AIQuoteConsole % dotnet run
AI Quote Console App
Created by Luis Palma and Felipe Monsalvo

AI Quote Library by Luis Palma and Felipe G Monsalvo

Enter a topic: Music

Generated Quote:
"Music is the shorthand of emotion." – Leo Tolstoy
○ (base) luispalma@MacBookPro AIQuoteConsole % █
```