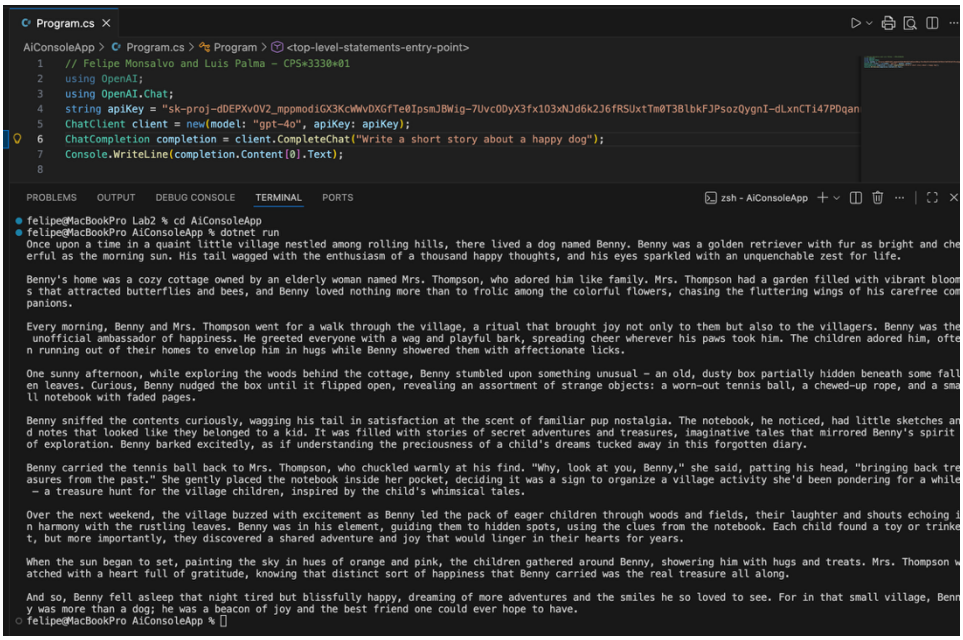
Luis Palma

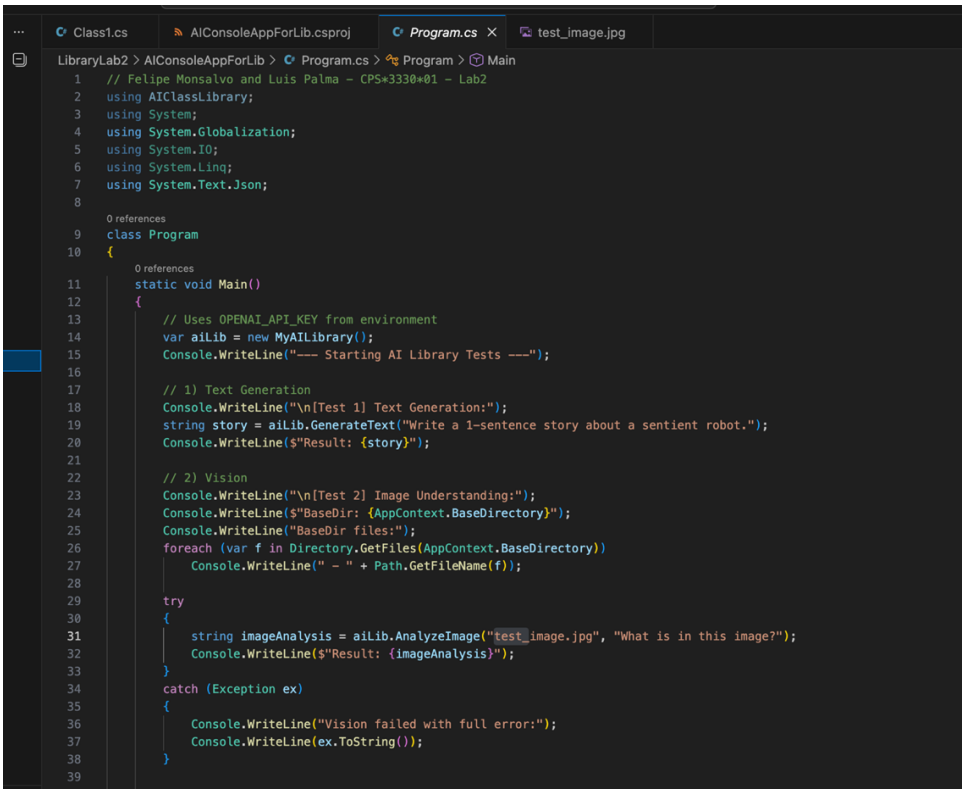
Felipe Monsalvo

Task 1:

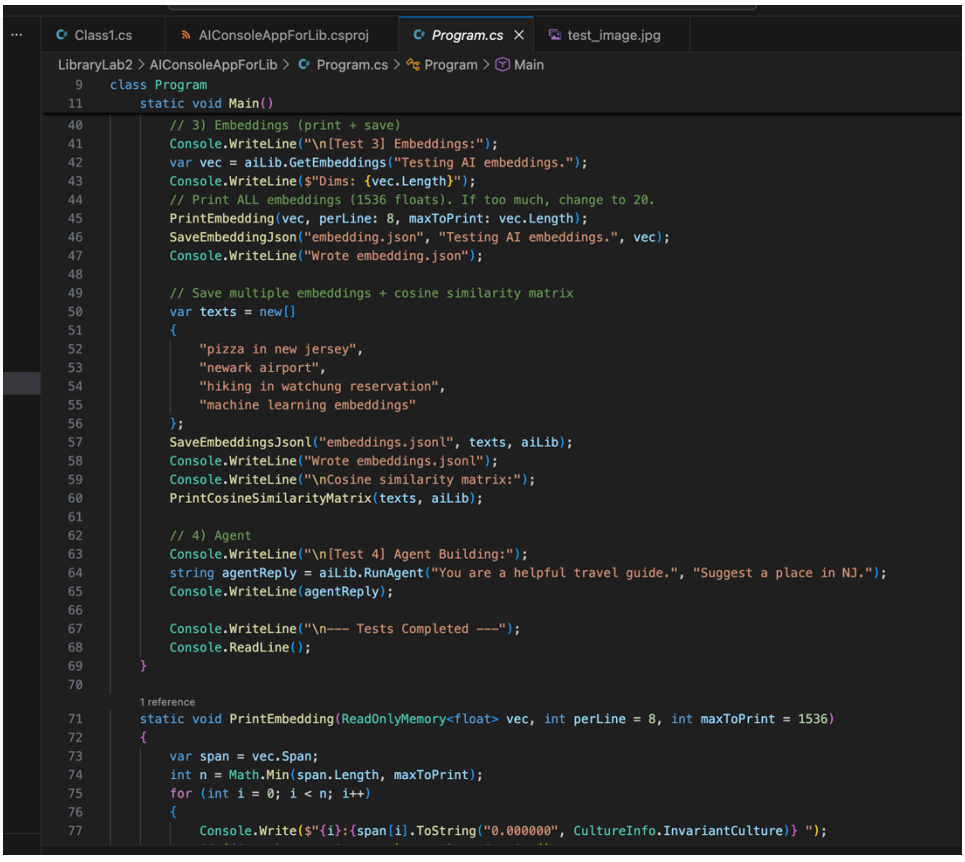


Task 2:

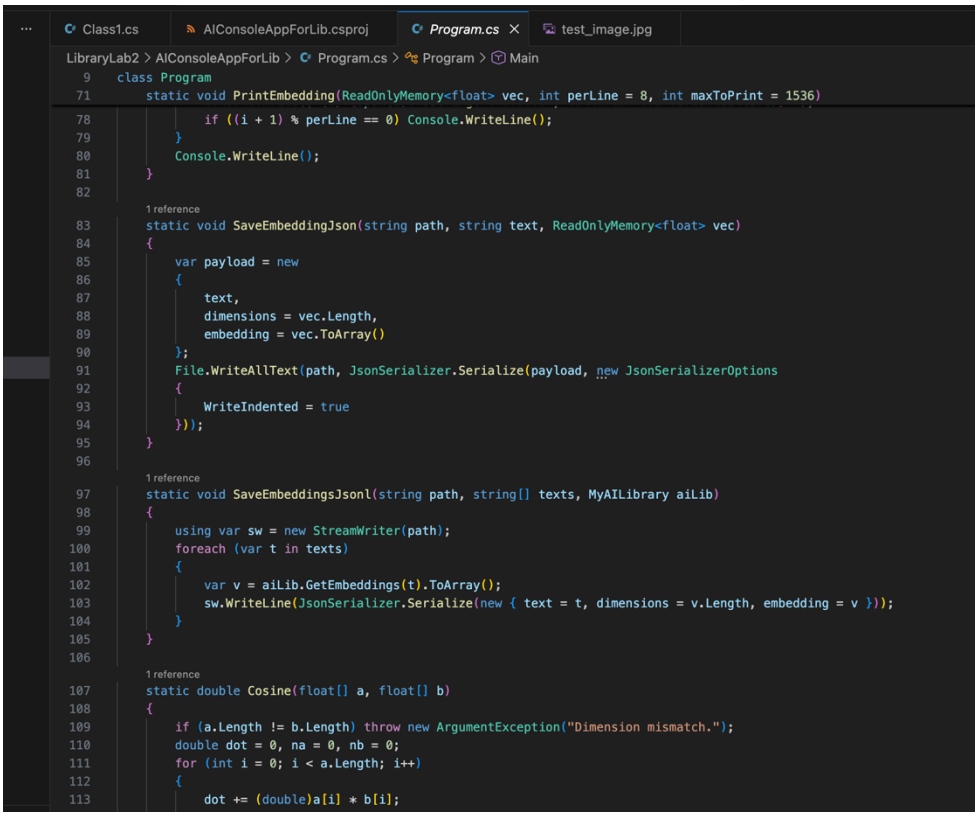
Task 3:



```
// Felipe Monsalvo and Luis Palma – CPS*3330*01 – Lab2
using AIClassLibrary;
using System;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Text.Json;

0 references
class Program
{
    0 references
    static void Main()
    {
        // Uses OPENAI_API_KEY from environment
        var aiLib = new MyAILibrary();
        Console.WriteLine("--- Starting AI Library Tests ---");

        // 1) Text Generation
        Console.WriteLine("\n[Test 1] Text Generation:");
        string story = aiLib.GenerateText("Write a 1-sentence story about a sentient robot.");
        Console.WriteLine($"Result: {story}");

        // 2) Vision
        Console.WriteLine("\n[Test 2] Image Understanding:");
        Console.WriteLine($"BaseDir: {AppContext.BaseDirectory}");
        Console.WriteLine("BaseDir files:");
        foreach (var f in Directory.GetFiles(AppContext.BaseDirectory))
            Console.WriteLine(" - " + Path.GetFileName(f));

        try
        {
            string imageAnalysis = aiLib.AnalyzeImage("test_image.jpg", "What is in this image?");
            Console.WriteLine($"Result: {imageAnalysis}");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Vision failed with full error:");
            Console.WriteLine(ex.ToString());
        }
```



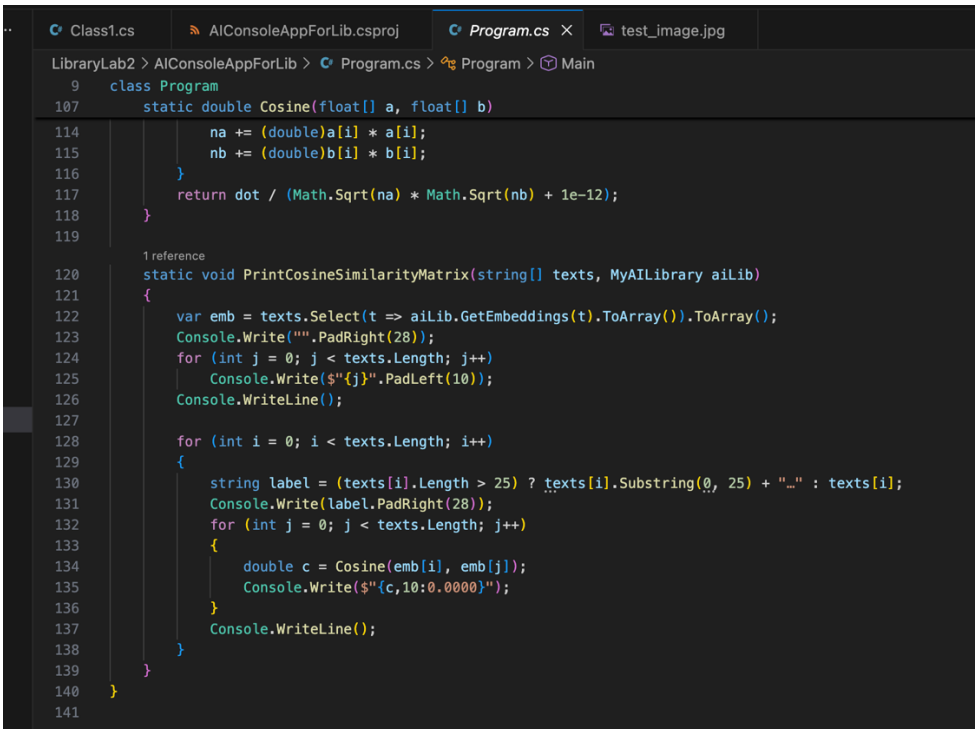
```
class Program
    static void Main()
        // 3) Embeddings (print + save)
        Console.WriteLine("\n[Test 3] Embeddings:");
        var vec = aiLib.GetEmbeddings("Testing AI embeddings.");
        Console.WriteLine($"Dims: {vec.Length}");
        // Print ALL embeddings (1536 floats). If too much, change to 20.
        PrintEmbedding(vec, perLine: 8, maxToPrint: vec.Length);
        SaveEmbeddingJson("embedding.json", "Testing AI embeddings.", vec);
        Console.WriteLine("Wrote embedding.json");

        // Save multiple embeddings + cosine similarity matrix
        var texts = new[]
        {
            "pizza in new jersey",
            "newark airport",
            "hiking in watchung reservation",
            "machine learning embeddings"
        };
        SaveEmbeddingsJsonl("embeddings.jsonl", texts, aiLib);
        Console.WriteLine("Wrote embeddings.jsonl");
        Console.WriteLine("\nCosine similarity matrix:");
        PrintCosineSimilarityMatrix(texts, aiLib);

        // 4) Agent
        Console.WriteLine("\n[Test 4] Agent Building:");
        string agentReply = aiLib.RunAgent("You are a helpful travel guide.", "Suggest a place in NJ.");
        Console.WriteLine(agentReply);

        Console.WriteLine("\n--- Tests Completed ---");
        Console.ReadLine();
    }

    1 reference
    static void PrintEmbedding(ReadOnlyMemory<float> vec, int perLine = 8, int maxToPrint = 1536)
    {
        var span = vec.Span;
        int n = Math.Min(span.Length, maxToPrint);
        for (int i = 0; i < n; i++)
        {
            Console.Write($"{i}:{span[i].ToString("0.000000", CultureInfo.InvariantCulture)} ");
```



Class1.cs | AIConsoleAppForLib.csproj | Program.cs | 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)
78          if ((i + 1) % perLine == 0) Console.WriteLine();
79      }
80      Console.WriteLine();
81  }
82
        1 reference
83  static void SaveEmbeddingJson(string path, string text, ReadOnlyMemory<float> vec)
84  {
85      var payload = new
86      {
87          text,
88          dimensions = vec.Length,
89          embedding = vec.ToArray()
90      };
91      File.WriteAllText(path, JsonSerializer.Serialize(payload, new JsonSerializerOptions
92      {
93          WriteIndented = true
94      }));
95  }
96
        1 reference
97  static void SaveEmbeddingsJsonl(string path, string[] texts, MyAILibrary aiLib)
98  {
99      using var sw = new StreamWriter(path);
100     foreach (var t in texts)
101     {
102         var v = aiLib.GetEmbeddings(t).ToArray();
103         sw.WriteLine(JsonSerializer.Serialize(new { text = t, dimensions = v.Length, embedding = v }));
104     }
105 }
106
        1 reference
107 static double Cosine(float[] a, float[] b)
108 {
109     if (a.Length != b.Length) throw new ArgumentException("Dimension mismatch.");
110     double dot = 0, na = 0, nb = 0;
111     for (int i = 0; i < a.Length; i++)
112     {
113         dot += (double)a[i] * b[i];
```



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
        1 reference
120 static void PrintCosineSimilarityMatrix(string[] texts, MyAILibrary aiLib)
121 {
122     var emb = texts.Select(t => aiLib.GetEmbeddings(t).ToArray()).ToArray();
123     Console.Write("".PadRight(28));
124     for (int j = 0; j < texts.Length; j++)
125         Console.Write($"{j}".PadLeft(10));
126     Console.WriteLine();
127
128     for (int i = 0; i < texts.Length; i++)
129     {
130         string label = (texts[i].Length > 25) ? texts[i].Substring(0, 25) + "…" : texts[i];
131         Console.Write(label.PadRight(28));
132         for (int j = 0; j < texts.Length; j++)
133         {
134             double c = Cosine(emb[i], emb[j]);
135             Console.Write($"{c,10:0.0000}");
136         }
137         Console.WriteLine();
138     }
139 }
140 }
141
```

Class1.cs | AIConsoleAppForLib.csproj | Program.cs | test_image.jpg

LibraryLab2 > AIConsoleAppForLib > AIConsoleAppForLib.csproj

```
<Project Sdk="Microsoft.NET.Sdk">

  <ItemGroup>
    <ProjectReference Include="..\AIClassLibrary\AIClassLibrary.csproj" />
  </ItemGroup>
  <ItemGroup>
  <None Include="test_image.jpg">
    <CopyToOutputDirectory>Always</CopyToOutputDirectory>
  </None>
</ItemGroup>

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net10.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>

</Project>
```

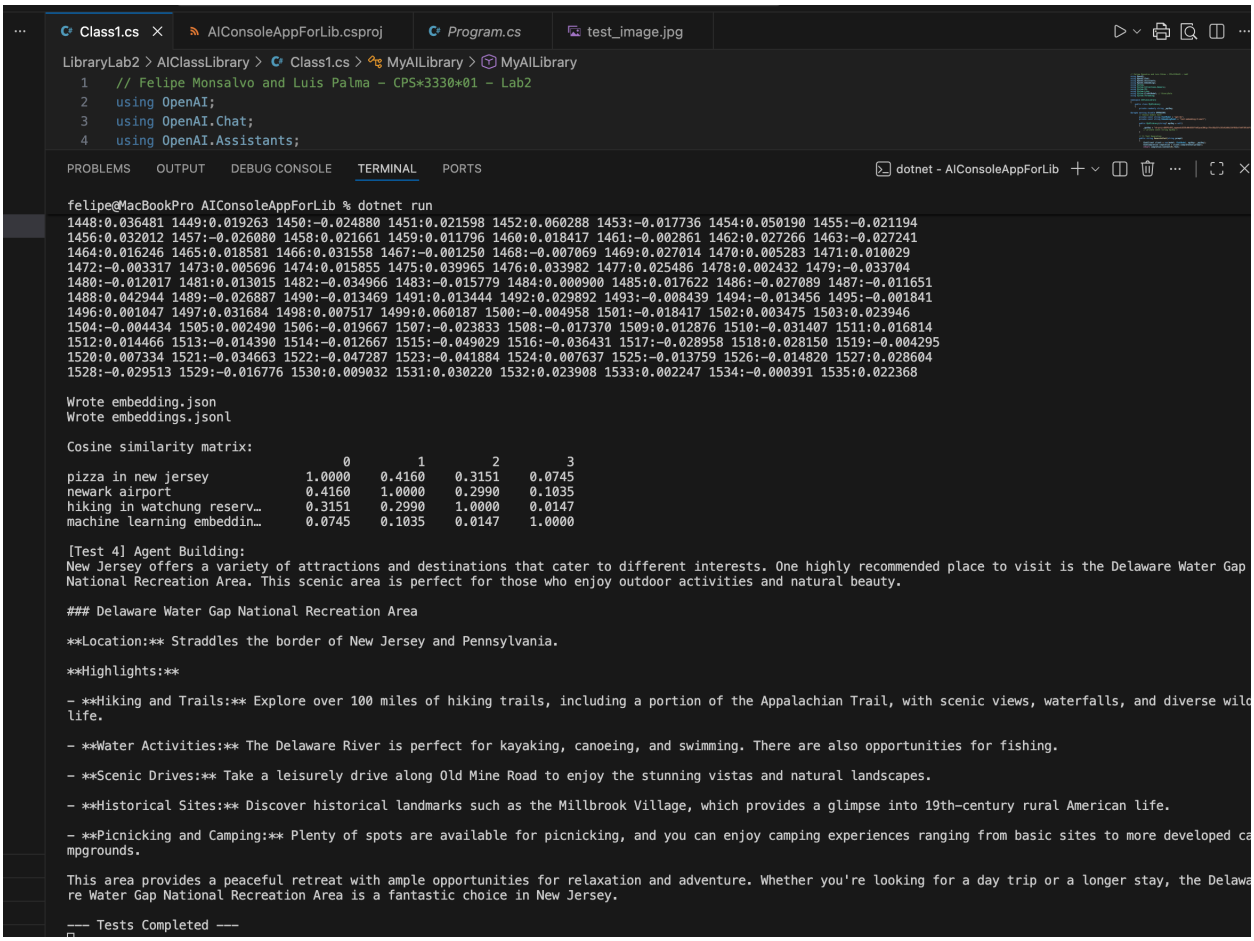
Class1.cs | AIConsoleAppForLib.csproj | Program.cs | test_image.jpg

LibraryLab2 > AIClassLibrary > Class1.cs > MyAILibrary > MyAILibrary

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

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS — dotnet - AIConsoleAppForLib

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

[Test 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 sto
ic humans weep.

[Test 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
[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 sce
ne 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 roc
ky outcrops.

[Test 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:-0.000812 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 39:0.001047
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:-0.005368 55:0.067509
56:0.002583 57:0.011714 58:0.003576 59:-0.000839 60:0.008944 61:-0.004986 62:0.020437 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:-0.061854 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:-0.012459 117:0.024451 118:-0.012377 119:-0.033098
120:-0.037642 121:0.003664 122:0.052816 123:0.027569 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
```


Class1.cs
AIConsoleAppForLib.csproj
Program.cs
test_image.jpg
LibraryLab2 > AIClassLibrary > Class1.cs > MyAILibrary > MyAILibrary
1 // Felipe Monsalvo and Luis Palma – CPS*3330*01 – Lab2
2 using OpenAI;
3 using OpenAI.Chat;
4 using OpenAI.Assistants;
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
dotnet - AIConsoleAppForLib
felipe@MacBookPro AIConsoleAppForLib % dotnet run
1448:0.036481 1449:0.019263 1450:-0.024880 1451:0.021598 1452:0.060288 1453:-0.017736 1454:0.050190 1455:-0.021194
1456:0.032012 1457:-0.026080 1458:0.021661 1459:0.011796 1460:0.018417 1461:-0.002861 1462:0.027266 1463:-0.027241
1464:0.016246 1465:0.018581 1466:0.031558 1467:-0.001250 1468:-0.007069 1469:0.027014 1470:0.005283 1471:0.010029
1472:-0.003317 1473:0.005696 1474:0.015855 1475:0.039965 1476:0.033982 1477:0.025486 1478:0.002432 1479:-0.033704
1480:-0.012017 1481:0.013015 1482:-0.034966 1483:-0.015779 1484:0.000900 1485:0.017622 1486:-0.027089 1487:-0.011651
1488:0.042944 1489:-0.026887 1490:-0.013469 1491:0.013444 1492:0.029892 1493:-0.008439 1494:-0.013456 1495:-0.001841
1496:0.001047 1497:0.031684 1498:0.007517 1499:0.060187 1500:-0.004958 1501:-0.018417 1502:0.003475 1503:0.023946
1504:-0.004434 1505:0.002490 1506:-0.019667 1507:-0.023833 1508:-0.017370 1509:0.012876 1510:-0.031407 1511:0.016814
1512:0.014466 1513:-0.014390 1514:-0.012667 1515:-0.049029 1516:-0.036431 1517:-0.028958 1518:0.028150 1519:-0.004295
1520:0.007334 1521:-0.034663 1522:-0.047287 1523:-0.041884 1524:0.007637 1525:-0.013759 1526:-0.014820 1527:0.028604
1528:-0.029513 1529:-0.016776 1530:0.009032 1531:0.030220 1532:0.023908 1533:0.002247 1534:-0.000391 1535:0.022368
Wrote embedding.json
Wrote embeddings.jsonl
Cosine similarity matrix:
0 1 2 3
pizza in new jersey 1.0000 0.4160 0.3151 0.0745
newark airport 0.4160 1.0000 0.2990 0.1035
hiking in watchung reserv… 0.3151 0.2990 1.0000 0.0147
machine learning embeddin… 0.0745 0.1035 0.0147 1.0000
[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.
Delaware Water Gap National Recreation Area
Location: Straddles the border of New Jersey and Pennsylvania.
Highlights:
- **Hiking and Trails:** Explore over 100 miles of hiking trails, including a portion of the Appalachian Trail, with scenic views, waterfalls, and diverse wild
life.
- **Water Activities:** The Delaware River is perfect for kayaking, canoeing, and swimming. There are also opportunities for fishing.
- **Scenic Drives:** Take a leisurely drive along Old Mine Road to enjoy the stunning vistas and natural landscapes.
- **Historical Sites:** Discover historical landmarks such as the Millbrook Village, which provides a glimpse into 19th-century rural American life.
- **Picnicking and Camping:** Plenty of spots are available for picnicking, and you can enjoy camping experiences ranging from basic sites to more developed ca
mpgrounds.
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 Delawa
re Water Gap National Recreation Area is a fantastic choice in New Jersey.
--- Tests Completed ---

Task 4 and 5:

Source Code:

Console**:**

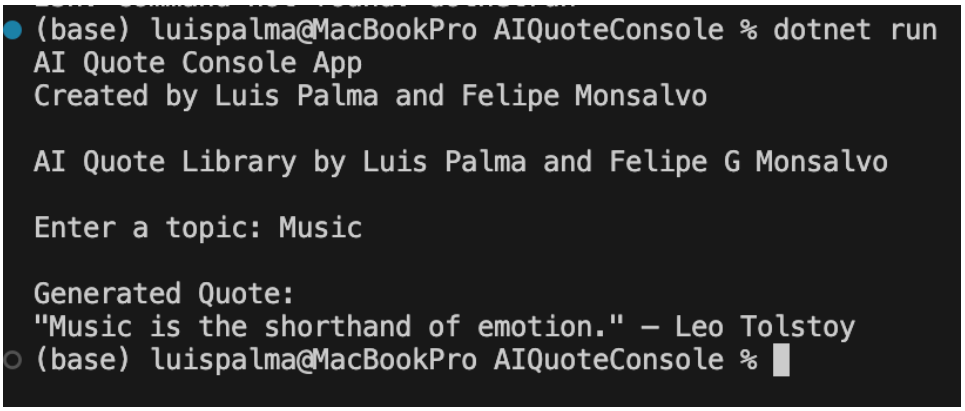
Lab2 > Lab2 > AIQuoteConsole > Program.cs > ...

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

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 %