EffortProject - Technical Overview
Description:
EffortProject is an ASP.NET Core Web API application that fetches and aggregates data from two external APIs:

A joke API

A posts API

It caches the data in memory for performance optimization and exposes a single endpoint to retrieve both datasets combined. The project also includes unit tests for core functionality.

Technologies Used:
ASP.NET Core Web API – Backend framework

HttpClientFactory – For making external API calls efficiently

Newtonsoft.Json – For JSON (de)serialization

IMemoryCache – For in-memory caching

XUnit – Unit testing framework

Moq – Mocking library for unit tests

Dependency Injection – For service and component decoupling

Architecture Overview:
Controllers

Handle API routes (e.g., BasicController)

Services

Business logic, coordinates memory and API services (JokeService, PostService)

Memory Services

Handle caching logic with IMemoryCache

API Services

Responsible for making HTTP requests to external APIs

Models

Represent data structures: Joke, Post, AggregatedData

Tests

Unit tests for services and controller using XUnit & Moq

Main Endpoint:
GET /api/FetchData
Returns combined jokes and posts as a single response:


```
{
  "jokeList": [
    { "type": "type", "setup": "setup", "punchline": "punchline" }
  ],
  "postList": [
    { "userId": 1, "title": "title", "body": "body" }
```

```
  ]
}
```

Features:

Fetches and merges data from two external APIs

Uses in-memory caching for 1 minute

Provides fallback mock data if an API fails

Fully unit tested with mocks for reliable testing

Clean, modular structure following separation of concerns