**Tasks Breakdown**

**Task 1: Solution Design**

* **Description**: Design the solution architecture and document it.
* **Deliverables**:
  + Architecture diagram in **draw.io**.
  + Documentation of the architecture, including components and their interactions.

**Task 2: Frontend (Angular 17)**

**Subtasks:**

1. **Setup Angular Project**
   * Bootstrap an Angular 17 project using ng new.
   * Configure routing for modularity.
   * Install needed libraries.
2. **Create Components**
   * **Code Editor Component**:
     + Allows users to write and edit C# code and save it.
   * **Code List Component**:
     + Displays the list of saved Codes (retrieved from the backend).
   * **Execution Result Component**:
     + Displays the output of executed functions.
3. **Integrate APIs**
   * Use Angular's HttpClient to communicate with the backend APIs for:
     + Fetching and saving source files.
     + Sending execution requests.
4. **Implement State Management**
   * Use Angular's BehaviorSubject to manage the state of selected functions.
5. **Build and Deploy**
   * Build the Angular app for production (will do if have time).
   * Configure the Dockerfile for hosting the Angular app in a container(will do if have time).

**Task 3: Backend (ASP.NET 8)**

**Subtasks:**

1. **Setup ASP.NET 8 Project**
   * Create an ASP.NET 8 project.
   * Install required NuGet packages:
     + Microsoft.CodeAnalysis.CSharp (Roslyn Compiler for dynamic execution).
     + EntityFrameworkCore (EF Core for database interactions).
2. **Database Schema**
   * Define the schema for:
     + **SourceFiles**: Contains the C# source code.
3. **API Endpoints**
   * **Source File Management**:
     + POST /api/CodeEditor/snippts: Save a new C# source file.
     + GET /api/CodeEditor/snippt: List all source files.
   * **Function Execution**:
     + POST /api/CodeEditor/execute: Execute a code.
4. **Dynamic Code Execution**
   * Implement dynamic compilation and execution using the **Roslyn Compiler**.
5. **Build and Deploy**
   * Create a Dockerfile for the .NET backend.

**Task 4: Database**

**Subtasks:**

1. **Setup Database**
   * Use SQL as the database.
   * Define tables for SourceFiles, Functions.
2. **Migrations**
   * Use EF Core migrations to generate and apply the database schema.
3. **Dockerize Database**
   * Write a docker-compose.yml file to spin up the PostgreSQL container.

**Task 5: Integration**

**Subtasks:**

1. **Frontend-Backend Integration**
   * Ensure the Angular app communicates with the .NET backend over REST APIs.