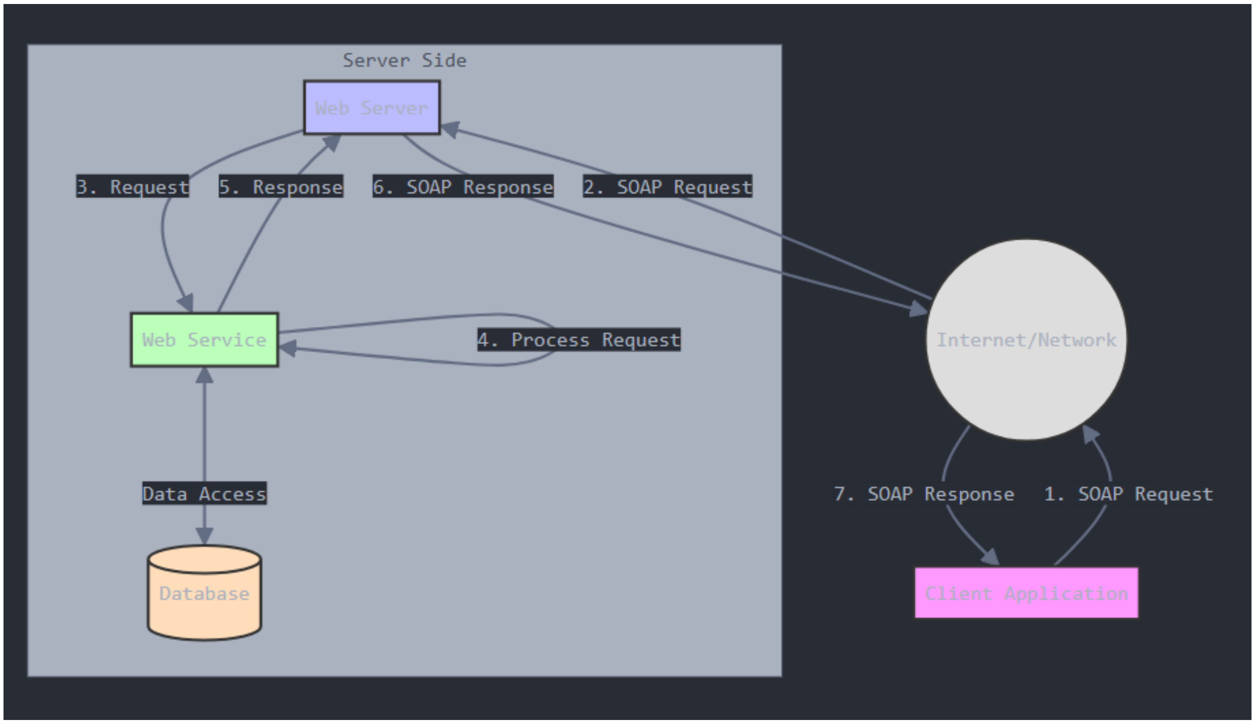
Question 1: Draw web services execution model?



Description: The Web Services Execution Model involves a client (such as a browser or web application) sending a request over HTTP using the SOAP protocol. This request is handled by the IIS web server, passed to the ASP.NET runtime, and processed by the web service file (.asmx). The web method executes and returns a SOAP response back to the client with the result.

Question_2:

CalculatorService.asmx

<%@ WebService Language="C#" CodeBehind="CalculatorService.asmx.cs" Class="MyWebService.CalculatorService" %>

CalculatorService.asmx.cs

using System;

using System.Web.Services;

```
namespace MyWebService
{
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    public class CalculatorService : WebService
    {
        [WebMethod]
        public string Add(double a, double b)
        {
            double result = a + b;
            return result.ToString();
        }

        [WebMethod]
        public string Subtract(double a, double b)
        {
            double result = a - b;
            return result.ToString();
        }

        [WebMethod]
        public string Multiply(double a, double b)
        {
            double result = a * b;
            return result.ToString();
        }

        [WebMethod]
        public string Divide(double a, double b)
        {
            if (b == 0)
                return "Cannot divide by zero.";
            double result = a / b;
            return result.ToString();
        }
    }
}
```

Description:

This web service, named **CalculatorService**, is a simple arithmetic service developed using ASP.NET (.asmx) that provides four basic mathematical operations: **Add**, **Subtract**, **Multiply**, and **Divide**. Each web method accepts two numerical parameters of type double and returns the result as a string. The service is designed to demonstrate the core concepts of building and exposing web services, following the architecture and implementation guidelines typically covered in foundational web service chapters of classic ASP.NET textbooks. The Divide method also includes basic error handling to manage division by zero, ensuring robustness. This service can be accessed and tested through a standard web browser or consumed by client applications such as .NET Windows Forms, JavaScript clients, or mobile apps.