ASP.NET Core MVC

**Top Tracker 400H And Online Judge Problem solves

1)Project SetUp

Ans:

1)Authentication configure,
2)Logger configure,
3)Autofac configure,
4)Dockerized kore project korte chaile docker file create kore buildable kore rakhte hbe.
5)Entity Framework use korle entity create korte hbe.
6)module Create korte hbe

2)Stored Procedure

Ans:A stored procedure is a prepared SQL code that we can save, so the code can be reused over and over again.
Following are the advantages of stored procedures:

1)Stored Procedure is good for writing more complex database queries.

2)They allow modular programming.(module e split kore develop)

3)They allow faster execution.(compiled thke,call korle ber ber run hoi na)

4)They can reduce network traffic.

5)They can be used as a security mechanism.(Stored Procedure is using parameterized query, by this way it avoids Sql-Injection attacks.)

3)Entity Framework

Ans : Entity Framework is an Object/Relational Mapping (ORM) framework that enables developers to work with relational database and deal with data as objects and properties.Using the Entity Framework, developers issue queries using LINQ, then retrieve and manipulate data as strongly typed objects using C# or VB.Net.

1.Difference between LINQ and SQL

S.No	LINQ (EF)	SQL (SP)
1	LINQ Stands for language integrated query.	SQL stands for Structured Query Language.
2	LINQ Statements are verified during compile time.	SQL statements can be used in a application gets verified for their syntaxes only in the run time.
3	To use LINQ we can depend upon our .Net Language syntaxes and also we can consume base class library functionalities.	To use SQL we need to be familiar with SQL syntaxes and also the pre-defined functions of SQL like MAX,MIN,LEN and SUBSTRING etc...
4	LINQ Statements can be debugged as they execute under framework environment. SQL .	As SQL statements execute on Database server debugging of the code is not possible.

4)Dependency Injection

Ans:

Dependency Injection is the design pattern that helps us to create applications which are loosely coupled(maintainability, testability and also re-usability).

A dependency is an object that our class needs to do the work, in other words, this object isn't created inside our class. The dependency is injected into a class that uses it, so the dependency is required to be external. The object instantiation (that is the dependency) must not be inside the class, but outside it.we can use constructor,setter-getter method,interface Implementation,service locator for dependency injection of a class.

5)Logger

Ans: Logging is keeping a record of all data input, processes, data output, and final results in a program. Logging serves numerous purposes including root cause analysis, bug analysis, and even performance reviews for the application.

Database e write : serilog.sinks.mssqlserver

Writes Serilog events to a set of text files, one per day : **serilog.sinks.rolling file**

6) Unit Test

Ans: Unit testing means testing individual modules of an application without any interaction with dependencies to confirm that the code is doing the right things.

Integration testing means checking if different modules are working fine when combined together as a group.

System Testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not.

Functional testing means testing a slice of functionality in the system (may interact with dependencies) to confirm that the code is doing the right things.

OneTimeSetUp --->sob gula test er shurute e ekber execute hobe.

OneTimeTearDown -->sob gula test er sesh e ekber execute hobe.

SetUp -->proti test er shurute execute hobe.

TearDown -->proti test er seshe execute hobe.

7)Docker
Ans: Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and deploy it as one package.
Docker command : All Command

Docker Compose:
1)docker-compose.yml file ,.env file --->Push on server(Docker hub)
2)cmd run kore
    docker-compose pull
    docker-compose up -d

Compose is a tool for defining and running **multi-container Docker applications**. With Compose, you use a **YAML file to configure your application's services**. Then, with a single command, you create and start all the services from your configuration. To learn more about all the features of Compose, see the list of features.

Compose works in all environments: production, staging, development, testing, as well as CI workflows. You can learn more about each case in Common Use Cases.

Using Compose is basically a three-step process:

1. Define your app's environment with a `Dockerfile` so it can be reproduced anywhere.
2. Define the services that make up your app in `docker-compose.yml` so they can be run together in an isolated environment.
3. Run `docker-compose up` and Compose starts and runs your entire app.

8)OOP/SOLID

Single Responsibility principle

In simple terms, a module or class should have a very small piece of responsibility in the entire application. Or as it states, a class/module should have not more than one reason to change.

Open/Closed Principle

The Open/closed Principle says "A module or class is open for extension and closed for modification".

Here "Open for extension" means, we need to design our module/class in such a way that the new functionality can be added only when new requirements are generated.When we want to extend functionality ,we can apply OCP by using inheritance,interface, abstract class, abstract methods and virtual methods."Closed for modification" means we have already developed a class and it has gone through unit testing. We should then not alter it until we find bugs.

Liskov Substitution Principle

Substitutability is a principle in object-oriented programming and it states that, in a computer program, if S is a Subtype of T, then objects of type T may be replaced with objects of type S

Which means, Derived types must be completely substitutable for their base types

Interface Segregation Principle

Let us break down the above definition into two parts.

1. First, a class or module should not be forced to implement any method(s) of an interface they don't use.
2. Secondly, instead of creating a large interface or fat interface, create multiple smaller interfaces with the aim that the clients should only think about the methods that are of interest to them.

Dependency Inversion Principle

- High-level modules should not depend on low-level modules. Both should depend on abstractions.
  AND
- Abstractions should not depend on details. Details should depend on abstractions.

To simplify this we can state that while designing the interaction between a high-level module and a low-level one, the interaction should be thought of as an abstract interaction between them.

9)Design Pattern
Ans:

Singleton Design Pattern

This pattern ensures that the class has only one instance and provides a global point of access to it. The pattern ensures that only one object of a specific class is ever created. All further references to objects of the singleton class refer to the same underlying instance.

Builder Design Pattern

Builder pattern builds a complex object by using a step by step approach. Builder interface defines the steps to build the final object. This builder is independent of the objects creation process. A class that is known as Director, controls the object creation process.

Prototype design pattern

Prototype design pattern is used to create a duplicate object or clone of the current object to enhance performance. This pattern is used when the creation of an object is costly or complex.

For Example, An object is to be created after a costly database operation. We can cache the object, returns its clone on next request and update the database as and when needed thus reducing database calls.

10)Razor
Ans:
Razor is a markup syntax that lets we embed server-based code into web pages using C# and VB.Net. It is not a programming language. It is a server side markup language.

Tag Helpers are attached to HTML elements inside our Razor views and can help us write markup that is both cleaner and easier to read than the traditional HTML Helpers. On the other hand,HTML Helpers are invoked as methods that are mixed with HTML inside our Razor views.

Tag Helpers enable server-side code to participate in creating and rendering HTML elements in Razor files. For example, the built-in `LabelTagHelper` can target the HTML `<label>` element when the `LabelTagHelper` attributes are applied.

- using Microsoft.AspNetCore.Razor.Runtime.TagHelpers;
- using Microsoft.AspNetCore.Razor.TagHelpers;

An **HTML Helper** is just a method that returns a **HTML** string. The string can represent any type of content that we want. For example, we can use **HTML Helpers** to render standard **HTML** tags like **HTML** <input>, <button> and <img> tags etc.

11)Web API
Ans:
Web API is an application programming interface (API) that is used to enable communication or interaction with software components with each other.

REST API

REST means “Representational State Transfer” and API means Application Programming Interface.The application makes a request to the API and the API provides the required data from the database to the application in the form of JSON or Xml, and the application then displays that data to the user.

Fluent API in Entity Framework Core

Fluent API is an advanced way of specifying model configuration that covers everything that data annotations can do in addition to some more advanced configuration not possible with data annotations. Data annotations and the fluent API can be used together, but Code First gives precedence to Fluent API > data annotations > default conventions.To access Fluent API we need to override the OnModelCreating method in DbContext.

Entity Framework Core Fluent API configures the following aspects of a model:

1. **Model Configuration**: Configures an EF model to database mappings. Configures the default Schema, DB functions, additional data annotation attributes and entities to be excluded from mapping.
2. **Entity Configuration**: Configures entity to table and relationships mapping e.g. one-to-one, one-to-many, many-to-many relationships, table name,Index,PrimaryKey, AlternateKey etc.
3. **Property Configuration**: Configures property to column mapping e.g. column name,data type,default value, nullability, Foreignkey, concurrency column etc.

12)Version Control
13)AWS

14)Deployment
Ans:
IIS deployment:
1)publish the application
2)Create an Demo Website on IIS express
3)set the port on the Demo site.
4)Explore the Application file (copy-->paste)on Demo Site
Worker Service Deployment:
1)publish the application
2)Open vs developer command prompt and run
sc.exe create ServiceName binpath= folder\\filename.exe start= auto
Sc.exe delete ServiceName

15)JavaScript
Ans:
DOM

document.appendChild(*element*) **Add an HTML element**

16)DataTables
Ans:
DataTables can obtain the data that it is to display in the table body from a number of sources, including from an Ajax data source, using this initialisation parameter. As with other dynamic data sources, arrays or objects can be used for the data source for each row.

17)BootStrap
Ans: Grid System

18)Worker Service
Ans:
Event fire in worker service:
1)Start
2)execute
3)stop