**The candidate must know:**

*.Net. C#*

* .Net. MSIL. Assemblies. Metadata. JIT-compiler. C# 6 and 7 syntaxis.
* Value and reference types. Boxing/unboxing.
* GC. Memory management. IDisposable
* OOP. Principles. Polymorphism, classes vs interfaces
* Collections. Generic. LINQ
* Delegates and events. Lambdas
* Exception handling
* Multithreading, tasks, asynchronous programming – basic concepts
* EF, ADO.NET
* Unit tests. TDD

*Design and architecture*

* GOF Patterns
* Patterns Repository and Unit of Work
* SOLID
* 3-layer or onion architectures
* MVC concept

*Web development*

* Web basic. HTTP. HTTP methods. HTTP request-response. HTTP processing pipeline
* REST. Web api. Controllers and methods. Web api method results.
* Models: validation and binding. Routing.
* Authentication and authorizing. Identity.
* Web api testing
* MVC (optional)

*HTML, JS*

* HTML, CSS – basics
* JS – basics
* Angular or other JS frameworks – optional

*DB*

* DB: tables, keys and relations, indexes
* SQL: simple queries, nested queries (subqueries), filtering, grouping, joining, union
* T-SQL: procedures, functions, triggers – fundamentals.

**The candidate can apply**:

IDE – VS or Rider etc,

Fiddler or Postman etc,

MS SQL Server and Management Studio etc,

Any Unit Test framework – Nunit or XUnit etc,

Swagger (optional),

Git

**Candidate experience**:

Commercial experience is not required.

Web pet projects experience is not required but desirable. Project requirements:

Backend part - 3-layer or onion architecture with low coupled DAL, BLL, PL (at least 3 projects).

Layers’ content:

* DAL – EF code first, migrations, Repo&UOW are applied
* BLL – several services with applying DI and IoC, mapping, authentication based on Identity
* PL – RESTful web api (or MVC)

Frontend part (optional) – html&CSS, JS or typescript