**Prescreening interview plan and requirements for candidates to .Net Lab (Kyiv 2020)**

**Interview plan**

The interview with a candidate to Lab takes 1 hour by following schedule:

5 – 7 min – general questions (education, experience etc.)

15 – 20 min – project presentation and questions related with the project.

30 – 40 min – questions on C#, DB and JavaScript.

5 min – questions from the candidate.

**Project description and requirements**

Students receive individual task, for example File Storage System, Forum, Auction, Blog etc., in the middle of the training and must implement by the end the following items:

Backend part - 3-layer architecture with low coupled DAL, BLL, PL (at least 3 projects). Layers’ content: DAL – EF code first, Repo&UOW. BLL – several services with applying DI and IoC, mapping, authentication based on Identity, PL – RESTful web api.

Simple frontend part – Angular application.

Unit tests (optional)

Links to students’ projects on GitHub

**Requirements to candidate technical skills**

The candidate to .Net Lab is a person who attended 3-month .Net external training (often case) and is able to develop project with described requirements. Candidate practical skill and knowledge level may be is not enough to propose the candidate on junior position on production project but plenty to start education in .Net Lab to improve skills during next 2-3 months of internal training to become .Net junior in epam.

The candidate to .Net Lab 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

*Design and architecture*

* Patterns
* SOLID
* 3-layer architecture
* MVC

*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

*HTML, JS*

* HTML, CSS – basics
* JS – basics
* Angular – basics

*DB*

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