|  |  |  |  |
| --- | --- | --- | --- |
| A | Core Skills |  | |
|  | |
| 1 | **Programming, MCQ score, domain exercise all combined score** GOF design Pattern application, standard solution with details |  | |
| 2 | **Language fundamentals, Collections and Data Structure, basic Internals** Adv collection internals , Solid Principles |  | |
| 3 | **Generics,Lambda expr, Functional Interface,streams , predicate** Completable future, Parallel and Serial Stream, Lazy streams |  | |
| 4 | **Any one type GOF Design Patterns creational/structural/behavioural**) Awareness to more than one design pattern categories listed above |  | |
| 5 | **Rest principles,OAS consumer, versioning, pagination, Mocking** Contract 1st OAS, Best practices, Code generation |  | |
| 6 | **Cloud Deployment, basic cloud set up, Serverless and storage services** Provison with Terrform/ARM , Application architecture over cloud and adv svc |  | |
| This is Mandatory Section!Is the candidate good with coding skills to execute Simple to Medium Complexity code? | | |  |
| B | Advance Design & Development skills |  | |
|  | |
| 1 | **Database RDBMS/No SQL (SQL, Joins, Indexing) ORM** Entity Modeling, Aggregate functions, CDC |  | |
| 2 | **Multithreading: Deadloack,Fork & Join,Sync,TLP** Concurrent Collection, Lock Internals , Thread local |  | |
| 3 | **Messaging (pub-sub/queue, durability, tracing) , Schema, AMQP/IIOT/JMS, Kafka** Broker internals ,scaling, event based programming |  | |
| 4 | **Ent. Librabry, Unity Containers or caching** Distributed Caching, distributed transactions |  | |
| 5 | **Profiler, finalize, garbage collection basics** GC Dump Analysis,GC algos |  | |
| Is the candidate ready for Advance .NET implementations to execute with Simple to Medium Complexity? | | |  |
| C | Tooling & Framework |  | |
|  |  | |
| 1 | **Advance Topics: Grpc,Service mesh, Graph QL or Data Processing technologies (SSIS,spark, flink)** |  | |
| 2 | **Web (Sockets/Html/JS,UI etc.)(Full Stack?)** Adv React, Angular, design, SPA |  | |
| 3 | **TDD,Unit Testing, Functional Testing**, BDD,Tooling overall |  | |
| 4 | **Devops(CI, MSBuild, Jenkins, Quality plugins,Docker, kubernetes** Pipeline,Chef/Ansible, Branching Merging, Multi Geography |  | |
| 5 | **Log export and Monitoring(exporting and usage of log tools**. Configuration in any of  ELK, Grafana,COTS tools |  | |
| D | APIs, Cloud & Microservices |  | |
|  | |
| 1 | **MS Patterns for Orchestration Patterns** Choreography Patterns and Command Pattern |  | |
| 2 | **Bounded Context, Domain model(Fat Domain , Aneamic Domain**) , Transaction (Cancel, Rollback, compensating), Aggregates |  | |
| 3 | **CQRS Patterns (Database Model, API Impl and deployment** Event sourcing in Microservices, Choice of DB, Same DB strategy |  | |
| 4 | **API Security Basic Auth, Client Credential, JWT usage** Authcode Flow, Open IDC, IAM integration |  | |
| 5 | **API Gateway policy Implemenmtation security , traffic, Devportal** API Gateway Architecture (Commercial API Gateways) & Config |  | |
| Is the candidate ready for microservices  and Core .NET to execute with Simple to Medium Complexity?? | | |  |
| E | Solution & Architecture |  | |
|  | |
| 1 | **Logical views, sequence views , Application Layering , SCD** BCD, SCD,Technology and solution detailing, end to end view |  | |
| 2 | **NFR (Data at REST / Motion / In use)** Multi tenent , multi geography/Brand, Scale |  | |
| 3 | **Integration Patterns and Integrtation exp with any enterprise systems** Enterprise Service Bus e.g. Mule, Camel, public Cloud based |  | |
| 4 | **Production deployment checklist,operational steps , Canary release** Automation testing (SIT, Security, prod) |  | |
| 5 | **POC,  certifications, IC, execute Application end to end** Req gatethering, Estimate (Track level), Lead & Mentor |  | |

**Recently viewed in skillenza**

**Weather Service**

Develop, test and deploy a micro service to show the output of a city’s next 3 days high and low temperatures. If rain is predicted in next 3 days or temperature goes above 40 degree Celsius then mention ‘Carry umbrella’ or ‘Use sunscreen lotion’ respectively, in the output, for that day. The service should be accessible via web browser on internet and end user should be able to view results by changing city. The output should be presented in web browser using any one of JavaScript framework, HTML, JSON or plain text. And the service should be ready (in Git repository) to be released to production or live environment.

* API to fetch weather forecast data - [5 day weather forecast - OpenWeatherMap](https://openweathermap.org/forecast5)  
  If any other weather API is preferred that can also be used instead of this.

**Additional Instructions**

* NFRs  
  + Follow SOLID, 12 Factor and HATEOAS principles for design and implementation of the Micro Service
  + Any sensitive information used in the Micro Services such as API keys should be protected
  + Mention the design patterns used in the code
* Documentation  
  + Include the open-api spec to be part of the code. Should be able to view apis via swagger, that shows
  + Documentation to explain the purpose of the api along with Error codes that the clients must be aware of
  + Create a README.md file in the repository and explain the design and implementation approach
  + In the README, add a sequence diagram or flowchart created using [draw.io](https://www.draw.io/)
* Build & Deploy  
  + Build jobs and pipeline for your project(s); Pipeline scripts need to be part of the codebase;
  + Export the Jenkins job config., scripts as part of the project sources
  + Create a Docker image and publish service locally
  + The Micro Service should be deployed using a Docker container
  + Share the Docker image files as part of the code
  + Docker files (sources) need to be part of the codebase

Extend the solution to support offline mode with toggles; Ensures the service returns the relevant results as expected irrespective of the underlying API being available or not! (Using your own code/logic/data structures and without 3rd party libraries/DB)  
Demonstrate adding additional conditions, with the least code changes & deployment:

* In case of high winds (i.e.,) Wind > 10mph, then mention “It’s too windy, watch out!”
* In case of Thunderstorms, then mention “Don’t step out! A Storm is brewing!”

**Please uploads all sources as a zip file along with the executable uber jar with all dependencies in the field below**