thinkbridge

COMPATIBILITY ASSESSMENT TEST FOR BACKEND

Please find below the coding assignment, as part of your technical evaluation.
We understand that a proper solution to this assignment would take days, but please remember, a perfect solution does not exist! All we are looking for is the approach.

It is said that '*Art Speaks where words fail to explain*', and considering the software craftsmen we are -

- Please use this assignment to **portray the wide array or depth of your knowledge which generally will not be visible or even get asked during an interview.** Interviews are constrained by the knowledge of the interviewer and time and many a times do not do justice to your knowledge!
- Please see this as an opportunity to showcase your technical, logical, coding talents and creativity in your own style and time.
- Like we mentioned above, **a great code would be self-exemplary** of your skills and talent and eliminate the need for more evaluation. **We could propel you directly to the final round on its basis.**
- Do not feel pressured to provide the "perfect answer". The goal is for you to put forth your skills and give us an idea of how you approach tasks relevant to this role.

"***ShopBridge***" is an e-commerce application. As a part of this app, you need to build functionalities that helps manage different items they would have for sale. This will require a frontend/backend solution. As a Backend developer, you would be needing to provide a functional solution that can be consumed by the frontend application.

Backend Requirements:
Pick a Server-Side Programming Language that you are comfortable with or interviewing for. (C# / Python / Java / F# MockAPI https://mockapi.io/) to implement the APIs for "ShopBridge" for below actions.

Create a backend solution for Product module to be used by Product Admin, and perform below actions:

1. Add a new item to the inventory (The item should require a name, description, and price as basic fields, think of additional information that would be useful)
2. Modify an item in the inventory.
3. Delete an item from the inventory.
4. Lists the items in the inventory.

Use a relational (SQL Server / MSSQL / Postgres) database and any ORM you feel comfortable with for any persistent data storage needs. All the functions of the backend need to be served by an API call and the store's inventory should be persisted across restarts of the backend process.

1. All API calls should be asynchronous.
2. Include data validations as and when needed.
3. Exception handlings need to be present.

Consider this to be just an elementary starting point to get you going and is much smaller compared to the expanse of your knowledge!
Please feel free to demonstrate the width and creativity of your coding with handling of functionalities like business validations, image processing, stock manipulation, categories, search etc. along with technical pieces like exception handling, pagination...

Tracking:
Please do your best to keep track of how long you spend on the implementation of each of the, this is for self-timing and to determine estimation accuracy.

1. Data store design
2. API and service logic
3. Unit Test Coverage

Deployment Requirements:

1. The code should be shared via GitHub, BitBucket, or SourceTree holding all relevant code.
2. Do ensure its a deployment ready code which can be published to a production server.

Great Add-on to have:

1. Simple and minimal design, you can use Nuget packages you are comfortable with.
2. Code Coverage (UNIT TESTING) of 90% (using any Code Coverage Tool like PyUnit, MSUnit, NUnit, XUnit etc.)
3. Clean Code Readability
4. Supporting relevant documentation (detailed steps to run with screenshots, breakdown of time spent on different aspects of development).

IMPORTANT NOTE - DO NOT send an invite to collaborate. Add the deliverables in PUBLIC REPOSITORY in **Github, Bitbucket or SourceTree** and send the LINK to the following email ID –

- **Primary** - deep@thinkbridge.in