aat always active technologies

Technical Aptitude Assessment

While doing the assessment please keep in mind that we will score your submission based on the following criteria:

1. Object Orientated Programming Principles (Microsoft Standards)
2. Principals: SOLID, DRY, KISS, Anti Patterns, YAGNI
3. Naming Conventions and Coding Standards (Microsoft Standards)
4. Did your code compile out the box?
5. Did you submission run out the box?
6. Did we have to fix your bugs before we can review you code?
7. If there were nuances around your submission, did you document it?
8. Did you complete all the tasks in the request?

Environment: We will be running your submission on a vanilla install of Visual Studio 2022 Community Edition.

Tech Stack Requirements:

1. Framework: Microsoft .Net 8
2. Language: C#
3. Project Type: Blazor Server
4. Storage: SQLite
5. Data Access: EF Core/ Dapper/ SqlClient/Native
6. Source Control: Git
7. (Optional) Libraries for ease of use: MudBlazor or Bootstrap

Submission: You must provide a link to a git project that contains your solution.

Assessment Task Summary:

Create a simple single page application using MS Blazor that provides the following features.

1. Compute Data: - *see: Compute Data, Business Logic Requirements*
   a. Create a button on your page which starts data computation.
   b. When the data has been computed, Disable the button and enable the Save Data Button
2. Persist Data: - *see: Persistence, Save Data Business Requirements:*
   a. When clicked this button needs to save the computed data to **Storage**
   b. When the process is done disable the button and enable the following buttons
      i. Download Data as XML
      ii. Download Data as a Binary File

Compute Data, Business Logic Requirements:

1. Create a thread that randomly picks odd numbers. Add these numbers to a global/shared variable.
2. Create a second thread that calculates prime numbers in sequence. Negate these numbers and add them to the global/shared variable.
3. When the global/shared variable reaches 2,500,000 entries, create a third thread that only picks even numbers. Add these numbers to the global/shared variable.
4. Stop all the threads once the global/shared variable reaches 10,000,000. The global/shared variable must contain exactly 10,000,000 entries.
5. Sort the global list ascending
6. Count and display the total, odd and even numbers in the global/shared variable.
7. You will need to keep a handle on the list so that the user can save it

Persistence, Save Data Business Requirements:

1. Provide the SQLite DB with the following table model:
    - CREATE TABLE "Number" (
    - "Value" INTEGER NOT NULL,
    - "IsPrime" INTEGER NOT NULL DEFAULT 0,
    -);
2. Insert the previously generated global/shared list into this table as fast and as efficiently as possible

Download XML:

1. Retrieve all the records that you inserted into the Number Table above.
2. Serialize the records to XML
3. Return the .xml file for download

Download Binary:

1. Retrieve all the records that you inserted into the Number Table.
2. Serialize the record columns to a Binary format (.bin)
3. Return the .bin file for download