Interview Test 

You will be provided with a solution containing four projects for a small sample mobile application:

* InterviewApp.Library: Defines some common classes for data models and access.
* InterviewApp: The cross-platform code that is shared between operating systems.
* InterviewApp.iOS: The platform-specific code for iOS devices.
* InterviewApp.Android: The platform-specific code for Android devices.

The app consists of two main pages: the About page and the Browse page. The About page shows some static information on the UI, provides a link to a website and has a label to indicate which OS the app is running on (iOS or Android). The Browse page displays data from a local SQLite store and allows the user to add items to it. The Item data is accessed via the IDataStore interface, of which SqliteDataStore is an implementation of. The SqliteDataStore service provides an abstraction layer over an Entity Framework Core DbContext.

You will need to complete the following tasks, and submit your resulting solution and some documentation of your process to us. The documentation just needs to be a .txt file or code comments that describe your thought process and the steps you took to complete the tasks. Tasks 1 and 2 are meant to test your debugging ability, and Tasks 3 and 4 are meant to test your overall coding ability and code style.

# Task 1: Build

Before you can run the project, you will need to fix a few build errors. Since you cannot continue with the rest of the tasks without building the app, you can contact us for the solution if you are stuck on this step.

Don’t worry if you only have access to one of the platforms, iOS development does require a Mac as a build host, so it’s likely you’ll only be able to work on the Android version.

# Task 2: Fix

After running the project, there will be two run-time errors that cause the app to crash. The first error is simple and will crash the app as soon as it is started, the second is a bit more complex and only causes a crash when adding a new item to the Browse page.

Since you cannot continue with the rest of the tasks without running the app, you can contact us for the solution if you are stuck on this step.

# Task 3: Improve

Once you have the app working and have successfully used every feature, make some changes to improve the Add Item feature. This could include improvements to the UI design, the user experience, or the technical aspect.

# Task 4: Implement

Finally, add at least one new feature to the app. You have complete control about what to implement, the only two requirements are:

* It should not require us to prepare anything when running your code, you should ensure we can download your code, and run it.
* Don’t just find a code snippet online and paste it into the app, put some effort into coming up with your own feature. You can use any third-party library or code (that you have permission to use), but make sure you adhere to licenses and also please attribute any work that is not your own. (A comment next to the code or usage of the library is fine).