

Prerequisites

For this test you will need a version of Visual Studio that supports .NET Core 3.1 development. You can use Visual Studio 2019 Community Edition which can be downloaded from Microsoft at <https://visualstudio.microsoft.com/vs/>.

Check that you have the .NET Core 3.1 SDK installed by looking in add/remove programs. It is listed as "Microsoft .NET Core SDK 3.1.XXX (x64)". If not then download from Microsoft at <https://dotnet.microsoft.com/download/dotnet-core/3.1>

You will also need GIT to be able to commit to the repository. GIT can be downloaded from <https://git-scm.com/downloads> here if you don't have it.

Once installed you should then be able to open the solution file `Aris.ServerTest.sln` and you can begin the test.

Overview

The practical test should take around 90 minutes although there is no strict time limit. Please try and complete as many of the tasks as possible.

Please commit your code to the Git repository after each task has been completed.

This website is a mock casino that pulls a list of games onto the page via an API named Kore. You will need to read and understand some of the Kore API documentation to be able to complete some of the tasks below. The Kore API documentation can be found at <https://doc-kore7.aristx.net/>.

When you've finished, please send the repository back to Aris so we can see how you've done.

Tasks

Please follow the instructions below...

1. The solution will not build. Find and fix the build error which is preventing this.
2. When the application first runs, an application error is displayed. Please identify the cause of the error and modify the code to prevent it. Once the error has been fixed, the application should display a list of games.
3. The games are displayed in a table on the main page. Add a new column to the table titled "Free spins" Modify the application so that the "free_spins" attribute is displayed in this column for all rows where it is returned from the Kore API.
4. Modify the code so that the list of games is ordered by Category, then Platform, then Name.
5. Populate the category dropdown on the main page with a list of all possible categories. Implement filtering so that the when a category is chosen, only the games that match that category are displayed in the table.
6. Next to each game is an "Info" link. When one of these links is clicked, the application should display the rules and description for the game. This information can be in a pop-up or added to the page

somewhere that you think is appropriate. This should be done by using Javascript to call the Details action on the Home controller.

7. Implement the missing code in the application so that the login page works. You should use the `KoreClaimsService` to do this. You will need to obtain a player token (see [https://docs-kore7.aristx.net/#obtain-a-player-token](https://docs.kore7.aristx.net/#obtain-a-player-token)).

Before you start this, you will need to register a user:

- Run the application and click on the Register menu option
- Complete the form and ensure that account currency and country are set to United States (important!)

You will be able to use these credentials to log in when you've modified the code. If you use a genuine email address then you will be able to use the reset password feature. On successful completion of this task, some of the games in the table will be clickable. Clicking on these links will launch the games.

Please note that your account may be locked out if you use the wrong password 3 times in a row. You can use the reset password feature to unlock your account if this happens.