**Miami HEAT**

**Software Engineer – Technical Assessment**

Please read carefully the entire assessment before starting the actual implementation.

The Miami HEAT’s Scouting team currently filling in a physical paper form to capture information about prospect players. The Product and Software Engineering teams propose a solution for Basketball Operation to capture this information using a modern and automated approach.

The Product team defined the rollout in two difference phases. Phase 1 will include a web app and phase 2 a mobile app. As part of the planning meeting, it was concluded that engineering team will focus its efforts to deliver a RESTful API that will be consumed by the Web App and the Mobile App.

The current Scouting report includes the following input fields:

* Scout Name
* Date Time Created
* General comment
* Defense Rate from 1 to 10
* Rebound Rate from 1 to 10
* Shooting Rate from 1 to 10
* Assist Rate from 1 to 10

Engineering concluded that the RESTful API will be implemented using .NET Core 3.1 with C# and Microsoft SQL Server. The scope of this assessment is:

1. **Set up a local development environment**
   1. Install SQL Server (You can choose either Developer / Express Edition / Azure SQL Server)
   2. Execute the TechnicalAssetmentDB.sql file provided in your SQL Server Instance
   3. Execute the TechnicalAssetmentData.sql file provided in your SQL Server Instance
   4. Create a new solution in Visual Studio
2. **RESTful API Implementation Basics**
   1. Basic Readers
      1. Create an endpoint that allow to search active players by name in a given season
   2. Scouting Report Command
      1. Create the necessary database structure to store Scouting Reports
      2. Create an endpoint that creates new scouting reports
   3. Scouting Report Readers
      1. Create an endpoint that retrieves the reports filtered by scout id and grouped by team
         1. This endpoint requires a nested structure including basic information about the team (Name, Nick name and Conference)  
            Ex:

[

{

"teamId":"Team A",

"nickName":"Nick name A",

"conferene":"West",

"players":[

{

"playerId":"1",

"playerName":"Player A",

"dob":"02/06/1990",

"reports":[

{

"scoutId":"1",

“createdDateTime”:” 2022-04-05 22:09:52.670”,

"comments":"Comments",

"defense":10,

"rebound":10,

"shooting":10

}

]

}

]

}

]

1. **RESTful API Implementation Nice to have**
   1. Basic Readers
      1. Create an endpoint that retrieves the list of leagues
      2. Create an endpoint that retrieves the list of teams filtered by league id
      3. Create an endpoint that retrieves the list of players filtered by team id and season
   2. Scouting Report Command
      1. Create an endpoint that updates the scouting report
      2. Create an endpoint that deletes the scouting report (soft delete)
   3. Scouting Report Readers
      1. Create an endpoint that retrieves the list of active scouts
2. **Make your code available for review in advance of the presentation (e.g., GitHub, Azure DevOps)**
3. **Go above and beyond**

This is not required you can either execute one, some or none of them:

* 1. Implement Unit Testing for your implementation
  2. Deploy your solution in Azure App Service using Azure DevOps
  3. Let your creativity fly implementing a front-end solution using any preferred framework (Angular / React (report api only) / Vue / Blazor / etc)

**Presentation:** You will present your project to appropriate members of the Miami HEAT using Microsoft Teams. You will explain your solution, identify any issues you encountered, and outline any enhancements that one might add later.

Notes:

* Feel free to take any architecture decisions for this implementation, but be prepared to elaborate about it
* You can use any IDE and/or tool you want, but we suggest using Microsoft Visual Studio, Microsoft SQL Server Management Studio and Postman
* Don’t worry about security for this particular project
* Questions about this project may be addressed to Enio Maiale at [enio.maiale@heat.com](mailto:enio.maiale@heat.com)