**Create Application Database. This can be created 2 ways....**   
Scripts are to be found in the “Application DB Scripts” folder.

1. Auto create the complete DB
   1. Open SQL Server Management Studio and ensure that the “master” database is being used.
   2. Open the "Create Full Accredit Employee DB.sql" script
   3. Ensure that the MDF and LOG locations at the top of the script are changed accordingly, based on where the MDF and LOG files are stored for the SQL Server installation being used.
   4. Run the script
   5. Database will hopefully have been created.  
        
      OR…..
2. Manually create the DB
   1. Within SQL Server Management Studio, create a new database called "AccreditHRSite"
   2. Using the newly created AccreditHRSite database, run the following scripts to create each of the tables required.
      1. Create Employees Table.sql
      2. Create Departments Table.sql
      3. Create Statuses Table.sql

**Create Employee/Department & Statuses Data**

1. Use the newly created “AccreditHRSite”
2. Run the “Create Data.sql” script

Before running the application also please ensure the EmployeeAPI Project appsettings.json file is changed so that the ConnectionStrings setting points to the Server where the Accredit DB was created.

*"DefaultConnection": "****Server=DESKTOP-9T3DNTE****; Database=AccreditHRSite; Integrated Security=true; TrustServerCertificate=True"*

N.B. It hopefully won’t be a problem, but please be aware that the application uses “Integrated Security” to connect to the DB.

**Application Information**

I haven’t been well over the last couple of days unfortunately (some form of sickness bug leaving me feeling very drained) so I’ve not been able to spend as much time on this as I’d first hoped but I’ve tried to do my best under the circumstances. Saying that, I have still however spent quite a bit of time (over a number of days) to try and get this done. Due to this though, I started to implement the Unit Tests but ran out of time  
unfortunately. Not great I know, but I did try my utmost ☹.

**Other Application Information**

1. The solution has had to be created under ASP.Net Core 7 as my laptop/Visual Studio wasn’t playing ball with anything higher. Not entirely sure why, but I tried to resolve but it was just taking too long.
2. I was stuck between a rock and a hard place as to what technology to use as although in more recent times I have used Blazor, I have used third party components (Radzen Blazor) as discussed with Andrew, to make development a lot quicker and I know that this was not what was being looked for with this test and would’ve possibly taken me longer to implement. I decided to stick with MVC although this is something I last used some 6 months ago (with me developing in Blazor since then and being made redundant 2 months ago) so set it's own challenges with me “getting back into it”. Hopefully I’ve done a half decent job 😉?
3. If required, please set the solution startup projects to both the “EmployeeAPI” and the “EmployeeMVC”
4. If running solution in development environment, I have left the Swagger information on the API to be displayed still.
5. When creating a new employee I have defaulted the date of birth, to todays date -20 years. Reason for doing this was just so that the date picker was closer to an employees’ actual date of birth and so quicker to select if using the date picker.
6. I have applied a hard delete to the records although I would normally in other systems I have written, just carry out a soft deletes in case the record needs to be resurrected or is needed for auditing purposes.
7. The specification mentioned that the “Employee Number should be unique only when the status is Active”. Wasn’t too sure what to do about this as there is no actual “Active” status in the defined list.  
   Is “Active” deemed as an employee that does not have a status of “Disabled”? I ended up adding an “IsActive” flag to the Employee model and used this to carry out the required check. If the former was required, then it’s a simple change to check the Status field instead.
8. As per the specification, there is an EmployeeMVC “Appsettings” value (TotalResultsPerPage) that controls the Total number of rows shown on the Employee list page. This is defaulted to 5.
9. There are “IsActive” flags against both the Departments and Statuses tables. Setting any of the rows in these table to false, will prevent any new Employees being created from being able to select those values. Setting these to false will not affect any existing Employees.
10. I believe I have implemented everything that was required apart from completing the Unit Tests as I ran out of time as mentioned plus the Functional tests and API Endpoint authentication although the latter two, were “nice to haves” I believe with time allowing?

Hopefully the instructions suffice to get the DB installed and the application working. I hope the main core system on the whole is what you were after?

Martin Cluley