**.NET Exercise:**  
Our preference is that you push your project to GitHub, so that we can clone your solution directly from there. Please ensure that cloning the repository, then running it directly (without any additional setup steps) will start the app properly. We consider this a reflection of how you work. If the application does not run, we will not be able to grade it.

Please get your solution in a week or so.

**Business Requirements**

* Develop REST endpoints to create, read, update and delete (CRUD) employees using .NET Core WebAPI
* An employee can have some of these properties - firstname, lastname, email address, age etc.
* The combination of first name, last name and email address should be unique.
* The application should start with some employees added.
* Provide API documentation (like Swagger) to interact with the REST endpoints.

**Technical Requirements**

* Use Entity Framework Core to interact with database
* The database table(s) should be created when the application is started. You can pick any of these approaches:
  + EF code first approach
  + Send the DB script to add the schemas
  + Use in-memory database
* Use Dependency Injection
* Add at least few unit tests

**Extra Credit Ideas**

* Add a REST endpoint that takes a search input then returns a list of employees whose first or last name matches
* Add one more table to save the employee's address. Include the address property in the CRUD Apis for employee
* Simulate latency in the API call