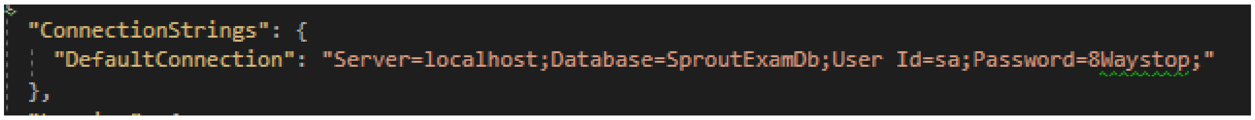
Requirements

1. Visual Studio 2019 with .net 5 Installed
2. SQL Server/ SQL Express 2016 or up
3. Updated node js
4. Visual Studio Code for front end editing
5. SproutExamDb.bak File (to be given by Exam Coordinator). Filename: SproutExamDB11052021.bak
6. Sprout Exam WebApp Solution (to be given by Exam Coordinator). Filename: Sprout.Exam.WebApp.Zip

Setting up the project

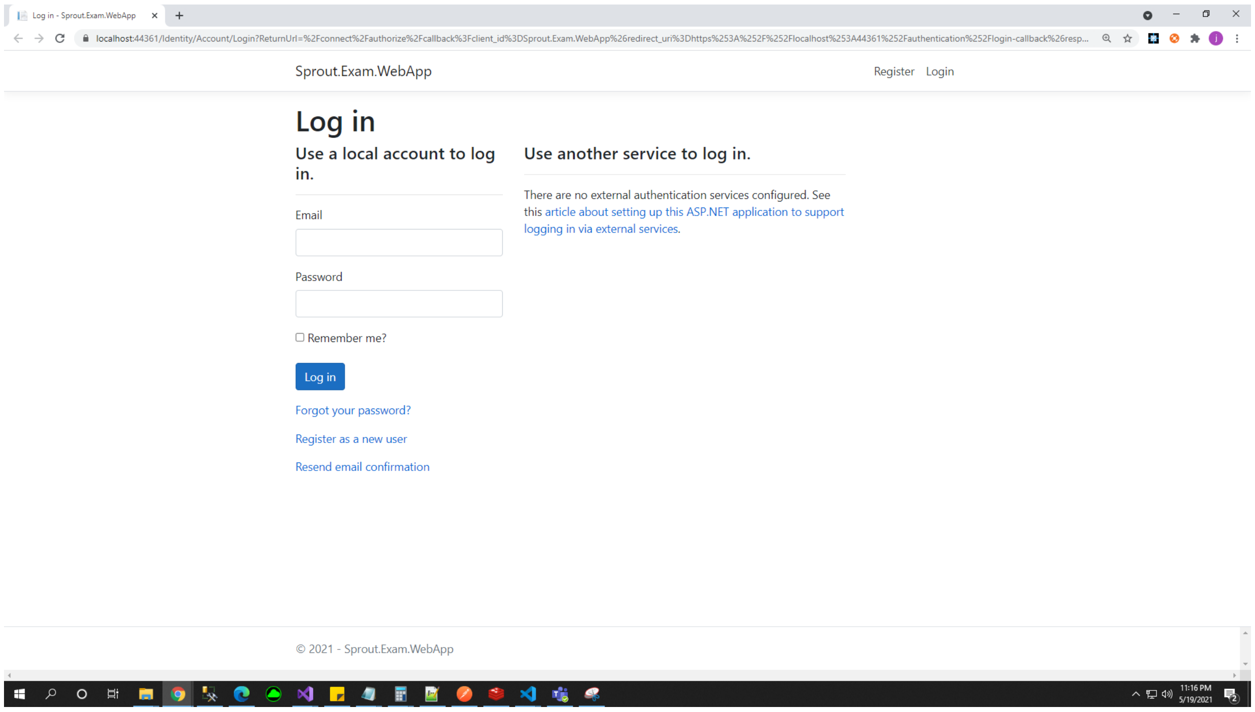
1. Restore the given database in your local sql server using SproutExamDb.bak File.

2. Change the value of the default connection depending on your setup and local environment.



```
"ConnectionStrings": {
  "DefaultConnection": "Server=localhost;Database=SproutExamDb;User Id=sa;Password=8Waystop;"
},
```

3. Run the project and you should see the login form.



Log in - Sprout.Exam.WebApp

localhost:44361/Identity/Account/Login?ReturnUrl=%2Fconnect%2Fauthorize%2Fcallback%3Fclient_id%3DSprout.Exam.WebApp%26redirect_uri%3Dhttps%253A%252F%252Flocalhost%253A44361%252Fauthentication%252Flogin-callback%26resp...

Sprout.Exam.WebApp

Register Login

Log in

Use a local account to log in.

Email

Password

Remember me?

Log in

Forgot your password?

Register as a new user

Resend email confirmation

Use another service to log in.

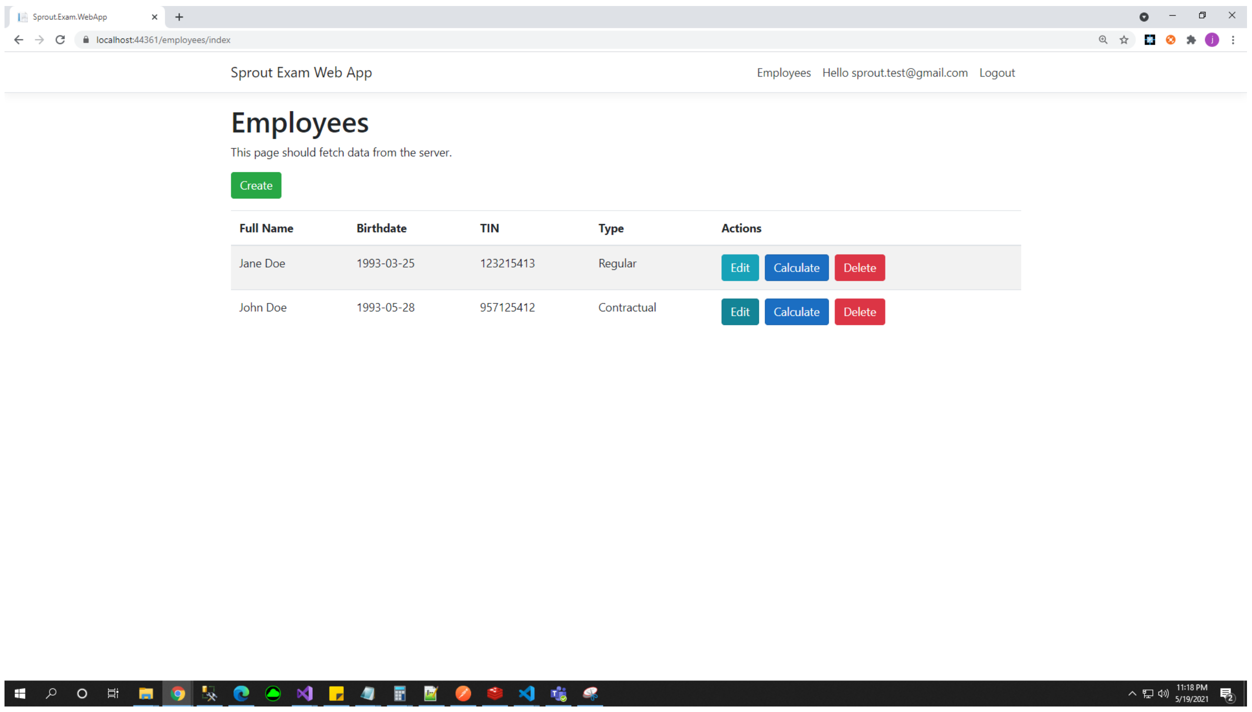
There are no external authentication services configured. See this article about setting up this ASP.NET application to support logging in via external services.

© 2021 - Sprout.Exam.WebApp

11:16 PM
5/19/2021

4. Login using the username: sprout.test@gmail.com and password: P@$$word6

5. If you can see the employee page then you are now ready to take the exam.



Sprout.Exam.WebApp

localhost:44361/employees/index

Sprout Exam Web App

Employees Hello sprout.test@gmail.com Logout

Employees

This page should fetch data from the server.

Create

Full Name	Birthdate	TIN	Type	Actions
Jane Doe	1993-03-25	123215413	Regular	Edit Calculate Delete
John Doe	1993-05-28	957125412	Contractual	Edit Calculate Delete

11:18 PM 5/19/2021

Background

Sprout Solutions automates all the administrative tasks around **HR** and **Payroll**. It's important to our clients that our applications run correctly and efficiently.

Assignment

- Your task is to create a web app that computes the salary of an employee.
  There are two types of employee:
  a. Regular employee
    - Per month salary
    - Absences will be deducted to the monthly salary (1 day deduction = monthly salary / 22)
    - Has 12% tax deduction
  b. Contractual employee
    - Per day salary
    - Salary is computed daily, and based on the number of days the employee reports to work
    - Has no tax deduction
- Your web app should be able to create a new employee and save it to the Database. Inputs are:

1. Name
2. Birthdate
3. TIN
4. Employee Type

- Your web app should also be able to delete and edit the employee details and make this reflect in the Database.
- The app should be able to compute the pay once "Calculate" is clicked.
  - If it's regular, you should be able to input the number of absences in days.
  - If it's contractual, you should be able to input the number of worked days.
  - Absent and worked days can have decimal places.
  - There should be a calculate button or whatever that will show the computed salary.
  - The answer should be rounded to 2 decimal places.
  - The answer should always show 2 decimal places (ex. 10,000.00).

Sample computation:

a. Regular Employee
- Has 20,000 basic monthly salary
- 1 day absent
- 12% tax
- = 20,000 - (20,000 / 22) - (20,000 * 0.12)
- = 16,690.91

b. Contractual employee
- Has 500 per day rate
- Reported to work for 15.5 days
- = 500 * 15.5
- = 7,750.00

What we're looking for

- **Working and running Application**. If the application doesn't run correctly, it doesn't matter how beautiful or efficient it is.
- **Clear**. It should be easy to read. The easier it is for someone else to come in and modify your application, the better.
- **Predictable**. Your application should be able to handle failures and exceptions.
- **Mandatory Requirement for Senior Developers but will be a plus for Mid Developer applicants:**
  - ➔ Proper Layering/Well-structured and Readable (use of naming conventions and standards)
  - ➔ Well-abstracted and should implement SOLID principles, design pattern (Factory Method Design Pattern) . Consider the possibility of a new employee type that will have a

different way of computing the salary (e.g., probationary, part-time).

- ➔ Unit Testing and Validation is mandatory
- ➔ Your web app should also contain a README with the answer to this question, explain your answer:
    - ◆ If we are going to deploy this on production, what do you think is the next improvement that you will prioritize next? This can be a feature, a tech debt, or an architectural design.

NOTE: It would be a plus if you could upload your code in the GitHub and share it to your interviewer. This way we could check your GitHub skills as well.