MRA Practical

Taxpayers System

Contents

[1. Language 2](#_Toc89209043)

[2. Packages used 2](#_Toc89209044)

[3. Step by step process 2](#_Toc89209045)

[4. Connectivity 3](#_Toc89209046)

The following steps were taken in coming up with the solution for tax payers system

# Language

The language that was used for this project for backend and from end is C# using ASP.Net Core framework. A combination of Html and JavaScript was used.

# Packages used

1. Newton.json
2. Bootbox
3. Datatables
4. Bootstrap

The project was created using Visual studio 2022 with dotnet 5.

# Step by step process

1. Open visual studio
2. Select new solution
3. Select asp.net core application
4. Enter the solution name
5. Click create
6. Add the necessary JavaScript and stylesheets files in the wwwroot folder for rendering pages
7. Create 2 partials called: \_sidenav for the side navigation bar and topnav from the top navigation bar. The sidenav must have 2 link (Home and Taxpayers)
8. Create an account controller where all the code for logging in and logging out will be stored
9. Create a taxpayers controller to render the tax payers page
10. Create a Taxpayer API controller to be calling the MRA Sand box API
11. Create the login page with a form with email and password. The form should be posting to action controller
12. In action controller, configure the HTTP Client and URL to connect to the MRA API
13. If the details supplied are correct, direct to dashboard/home page, if not, render the login page again
14. On the dashboard, there is a summary of all taxpayers in the system. The data comes from the Taxpayers controller which gets and count the total number of taxpayers available in the API
15. In Taxpayer account, configure HTTP Client and The URLs, create 4 functions, Get, Create, Update and Delete. Each serving its CRUD purpose respectively
16. The Get function should get all taxpayers in the system, create should create a taxpayer as per details submitted, Update function should edit a specific record and Delete should remove a record.
17. In all these, headers must be set with username which is kept in session after successful log in and the API for authentication
18. Only the form data that is correct is passed to the API and if an error occurs, it is shown to the user using Toastr notification
19. On the Taxpayers page, a table is displayed using Datatables plugin
20. The taxpayers’ data from the controller in put in a Viewbag for easy rendering on the page.
21. Inside the table, a loop is goes through the view back to display the data and two actions are placed on each record with its corresponding details. These actions are edit and delete
22. A button to add is placed on top on the table
23. Create a partial called \_form.cshtml and put the edit and add for there. To avoid code duplication, the same form is used for editing and its values are changed using JavaScript
24. The form should have all the details of Taxpayer table.
25. Create JavaScript functions in the index page of taxpayers and called Edit (which receives all the form data from table), delete (Which receives the TPIN from the table and submit which submits the data to the add or edit function on the API
26. The edit function must replace the edit form with values passed to the function after clicking on the edit button of a specific record and the modal should be shown
27. The delete button should show a conformation dialog box when clicked to confirm action to be executed by the user. Boot-box is used for this
28. The forms must be set with an onsubmit action to prevent the form from submitting normally but to use ajax instead for faster execution
29. In the submit JavaScript function, the form data must be serialized and passed to the function when it is received in the controller and handled according to the post method used
30. First start by checking if there are no modal state errors and proceed in none or show a notification to user if any
31. The logout button should be put at the top right corner and should route to account controller logout button where all the session created when logging in is removed immediately
32. To build the systems, press, control + shift + B or right click the solution and click build
33. To view on webpage, right click the project and select view on web
34. Once the logon page has loaded, enter the credentials, if correct, the user must be redirected to the Home Page
35. Click on taxpayers in the navigation tab, a taxpayers page must be shown
36. Click on add to add a tax payer, edit to update and or delete a taxpayer
37. The table on the page shows that taxpayers from database

# Connectivity

Make sure that you are connected to the internet as the API is hosted online and for you to access it, the computer must be online as well

Thank you