Evaluation Project

GENERAL INSTRUCTIONS

Complete as much as possible in under eight (8) hours. You do not have to complete the project in one sitting – you may divide the eight hours as you wish. Keep record of the total time utilized to complete the project. You may use reference materials (Google, YouTube, channel9.msdn.com, etc.) and/or consult with college or friend for advice. However, the code must be your own, based on your own knowledge and understanding of the technologies.

STEP I – Database

1. If you don't have access to a MS SQL database, download and install the free version at MS SQL Express.
2. Create a database named "TPPR".
3. Run script "tpprDabase.sql" to create the tables. This will create four (4) tables: Courses, Students, StudentCourses and CoursePayments:
   a. ***Students*** stores the names of the students.
   b. ***Courses*** stores the names of the courses and their cost.
   c. ***StudentCourses*** stores the relation between the student and courses. In other words, it stores which students are enrolled in which courses.
   d. ***CoursePayments*** store the payments amounts done by specific students to specific courses.
4. Create a user and assign him as database owner to the TPPR database.

STEP II – Application requirements

1. If you don't have Visual Studio, download the free (VS Community) or evaluation (VS Professional) version at Visual Studio Home.
2. Create a new C# MVC or MVC Core project, name "TPPREvalPrj" in Visual Studio, with "Individual User Accounts".
3. Change the connection string in the Web.Config to point to the database created in Step I.
4. Do the Code First/Database Migrations as required.

5. Create a page that displays the Students list.
6. Create a page that displays the Courses list.
7. Create a page that displays a list of Students, which courses they are enrolled in, the course cost, how much have they paid and the amount owned. One row per student/course relation.

STEP III –Optional

You may choose which, if any, of the following optional steps to complete:

1. Integrate the provided free template (FreeTemplate.zip).
2. Create separate pages for each of the following forms/actions:
   a. Create new course
   b. Edit course
   c. Create new students
   d. Edit students
   e. Register a student on a course
   f. Enter a student’s course payment
3. When a student is created, redirect to the details page of the student create, as opposed to the students list page.
4. While user is logged in, when creating a new student, the ModfiedBy should be fill with the UserID (GUI).
5. Modify the sql tables so that, when a new row is entered, ModifiedOn is filled with the current date and time.
6. Create a new Web API that returns:
   a. Sum of cost of all course registrations (projected income base on registrations)
   b. Sum of all course payments
   c. Sum of all debts
7. Populate a JavaScript pie chart, from the template (chartjs.html), with the Web API data.
8. If you wish, you may also complete other tasks/improves of your own design.

REFERENCES

- https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started
- https://mva.microsoft.com/en-US/training-courses/8322?l=nKZwZ8Zy_3504984382
  https://mva.microsoft.com/en-US/training-courses/introduction-to-aspnet-core-with-visual-studio-2017-16841
- https://mva.microsoft.com/en-US/training-courses/introduction-to-aspnet-mvc-8322
- https://mva.microsoft.com/en-US/training-courses/implementing-entity-framework-with-mvc-8931
- https://mva.microsoft.com/en-US/training-courses/introduction-to-aspnet-core-with-visual-studio-2017-16841
- https://www.asp.net/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api
- https://www.asp.net/web-api/videos/getting-started/aspnet-web-api