

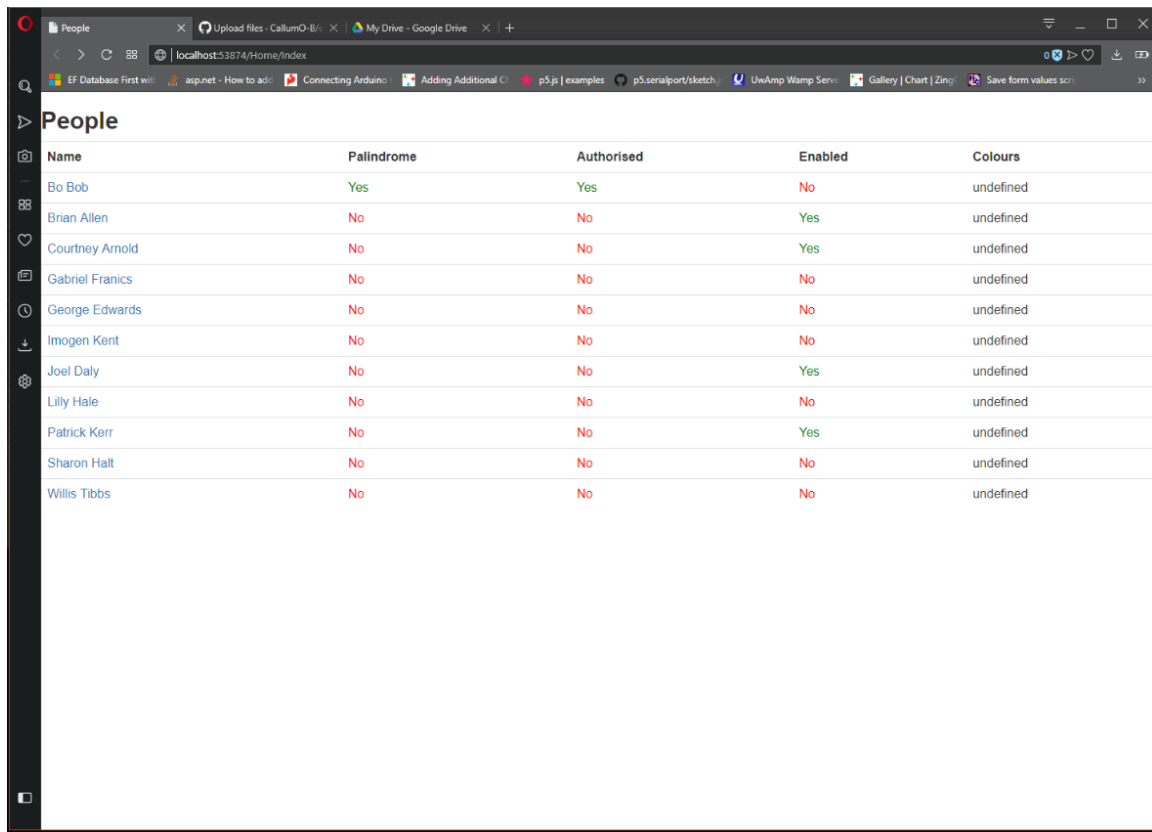
Interview Test

I have never created a single page application, therefore I had to do some research on single page applications to gain an understanding.

1. I have no experience with SQL Server. However, with some research and curiosity I restored the database:
 - I Opened the file in SQL Server.
 - Logged into the server using default settings.
 - The database was restored and contained the specified tables.
2. I had to do some research on ASP.NET MVC and how to open a blank project. I found a tutorial showing how to create an application using ASP.NET MCV which helped with some of the test. I achieved this task by:
 - Opening Visual Studio and creating a new project named TechTest1, after selecting ASP.NET Web Application (.NET Framework) within the C# tab. I also created another project named TechTest, this was used for testing so that I did not lose track or ruin the original project.
 - Empty template was selected and host to cloud option un-ticked. MVC folders option was ticked.
 - The tutorial I was watching advised Power Tools and Web Essentials to be installed, to help [with programming.
3. I had to do some further research on what a controller, home route and a blank view were. However, through curiosity I created a project that was not blank and found that a view contains the code that creates the user interface, a controller contains a set of functions within a class and the home route is the route for the home page of the application. Through research, I found how to create these.
 - An empty MVC 5 controller was added to the controllers folder, named HomeController. This contained an ActionResult named Index, therefore the view created was named Index to create a link between the view and controller.
 - A Home folder was added to the Views folder.
 - A view was then added to the Home folder, named Index. This added the view to the home route, creating the homepage. The view was created without a template and without a layout.

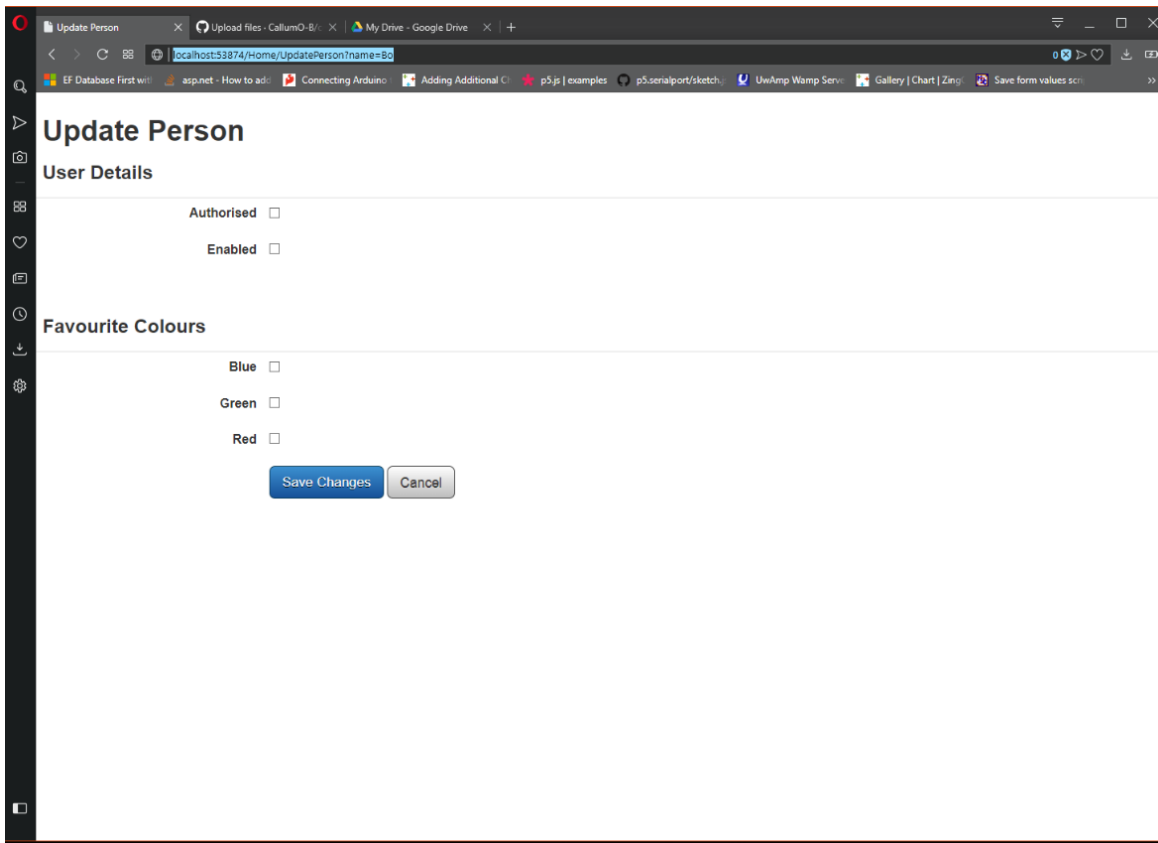
4. I had to do some research on Entity Framework EDMX and find out how it was able to point to a database.
 - An ADO.NET Entity Data Model named, TechTestModel was added to the project.
 - EF Designer from database was selected as this creates a model generated from the database and creates the edmx file as specified in the question.
 - A new connection to the database was made and the database was selected.
 - The connection was saved as TechTestEntities.
 - The tables were then selected, and the model namespace was named TechTestModel.
5. I had to do some research on where the JavaScript and HTML code had to be placed. Research had to be done on using AJAX, as I have only been taught the basic functionality. Using AJAX, HTML and C# together was a familiar task as I have done something similar in my dissertation where I had to get JavaScript to communicate with PHP and have them both work with HTML code. (Shown in figure 1).
 - A class for each database table was added to the Models folder. This created get and set for each database table field.
 - Using the Index.cshtml file, the JavaScript and HTML code was added.
 - A bootstrap CSS was used to provide the table design.
 - A jQuery CDN was used to provide jQuery functionality.
 - A div block was added containing the page heading and table.
 - A script block was added containing the JavaScript code.
 - An AJAX function was added, this created a table populated by data produced from JSON.
 - To convert the true and false values, from authorised and enabled data, to Yes and No values if and else statements were used. the Yes and No values were added to an array.
 - The first name and last name values were concatenated to form a single string and were added to a names array.
 - The names array was then sorted in alphabetically order. The authorise and enable arrays were reversed to match the values with the names. This was done to resemble the order depicted in the image.
 - A for each type function was added to iterate through the arrays adding the data to the table. To colour the yes and no text in the table, if and else statements were used. if a value in an array was "Yes" the value would be added to a corresponding variable and HTML font colour tags concatenated with it. For example, if a palindrome value was "Yes", the palindromeColour variable will be populated with a HTML font colour tag as green with the palindrome array variable concatenated with it.
 - For the colours column an issue was found and is still unresolved. From research, the FavouriteColours table was not added to the edmx file as its primary key is made up of two

foreign keys, so the table was ignored. I have not yet found a simple solution for this. However, an idea of adding an auto-incrementing unique primary key to the table, could be made.



Name	Palindrome	Authorised	Enabled	Colours
Bo Bob	Yes	Yes	No	undefined
Brian Allen	No	No	Yes	undefined
Courtney Arnold	No	No	Yes	undefined
Gabriel Franics	No	No	No	undefined
George Edwards	No	No	No	undefined
Imogen Kent	No	No	No	undefined
Joel Daly	No	No	Yes	undefined
Lilly Hale	No	No	No	undefined
Patrick Kerr	No	No	Yes	undefined
Sharon Halt	No	No	No	undefined
Willis Tibbs	No	No	No	undefined

Figure 1 Home Page Index View



Update Person

User Details

Authorised ☐

Enabled ☐

Favourite Colours

Blue ☐

Green ☐

Red ☐

Figure 2 UpdatePerson View

After reading questions 6-8, these questions required a lot of research as I have basic experience with AJAX and JSON. However, question 9 was a palindrome question which I was able to do immediately. Therefore, I completed question 9 next.

6. **(QUESTION 9)** The original idea was to add a value from names array into another variable, remove the white space, make the value lowercase, and then compare characters. For example, for bobob the first and fifth character would be compared and then the next inner character. This would be done using for-loops. However, this was a long route and so another method was used.

- A value from names array was added to a variable named name.
- The name value, had whitespace removed and changed to lower case, this value was added to a variable named chars.
- A for loop was then used to iterate through each character.
- An if statement was then used, the condition would become true if a character did not match the opposite character. This made a variable palindromeCheck false. Otherwise, palindromeCheck was true.
- If and else statements were then used to populate a palindromes array with either yes or no values.

7. **(QUESTION 6 (shown in figure 2))**

- **Part 1:** adding a click event to an anchor link in the first column was added by concatenating HTML href tags around the names array. Within the tags an onclick function was added, the name value was passed to the function and added to the URL. This link would add the UpdatePerson view to the webpage. The function for this onclick event was not completed as I am unsure about the AJAX part of the question.
- A view was created named UpdatePerson. This would display the form depicted in the image. I am unsure if this is the correct method however, the page was successfully created and displayed after clicking the anchor link.
- A controller was added to HomeController which added the UpdatePerson View.
- In the UpdatePerson view, a bootstrap CSS was added for the table design, a jQuery CDN was added for jQuery functionality and an internal CSS was added to create the white button design. An internal CSS was used, as I could not get a connection to the external file.
- Within a div block, a table was created. This contains the check boxes, labels, headings, and buttons. The labels and corresponding check boxes were added to separate columns to replicate the alignment depicted in the image. The labels had to be adjusted vertically to match the height of their corresponding checkboxes.
- **Part 2:** AJAX call to the server as I am unsure why this must be made other than gaining the name of the selected person. However, I was unable to obtain the full name as the URL dropped the last name as the variable value contained a white space.

- An AJAX call to the database to get PersonIds was unsuccessful as, no data was returned. After research a solution for this has not yet been found.
- **Part 3:** although the form has been created I am unsure if this was the correct method, as I have no experience with single page applications.

8. (QUESTION 7)

- This task was achieved by adding a click event to the labels. The event called a function called selectFunction. Each label passed a string value to the function which identified which label was clicked, for example the red label would pass a red string value to the function. Within the selectFunction definition, 5 variables were added for each check box. The variables were assigned the id of a specific check box.
- If and else if statements were used to identify the check box label clicked and whether the checkbox is ticked. For example, if the authorise label was clicked, and the corresponding checkbox is unticked it will now become ticked.

9. (QUESTION 8)

- After researching the question, I am unsure how to complete this task. After research the idea I have is that AJAX will save the form data of which check boxes have been ticked and unticked and using the PersonId value, update that person's data in the database, when the save changes button is clicked. For the second part of the question, I am not sure how the form would be destroyed but a method of directing the user back to the Index view could be made. For example, with the cancel button once clicked the user is directed back to the Index view, by go back through the URL history.