Further Practice 8-G

In this further practice, you will create additional web pages to interact with the `City` table. With the `Country` table web pages, you will also create a menu system that allows interaction between all the pages. In addition, web pages will be designed to edit the content of the `Country` and `City` tables.

I. Create city.php file

1. In the `c:\xampp\htdocs\project\` directory, create a file called `city.php` that contains the HTML title of "Cities of the World".
2. Create a PHP script that connects to the MySQL server that includes the connection information directly entered in to include the `world` database (versus using an external file to draw the information in).
3. Add to the script the ability to catch any errors in connecting to the MySQL server.
4. Add to the script a query against the `City` table in the `world` database that retrieves the country `ID`, `CountryCode`, `Name`, `Population`, and `District` where the `ID` is equal to the `ID` passed to the script via the `GET` method (`$_GET["ID"]`).
5. Add to the script the ability to catch any errors in the query execution.
6. Display the results of the query as follows:
`<h1>`*Name of the City*`</h1>`
`<p>Country Code:` *Country Code Link*`</p>(`*this link will open up the country web page – country.php?Code=CountryCode - to display the details of the country*`)`
`<p>Population:` *Size of the city population*`</p>`
`<p>District:` *Name of the District the city belongs to*`</p>`
`<p>Wikipedia Link</p>(`*this link will open up a Wikipedia web page - http://en.wikipedia.org/wiki/Name of City*`)`
7. Add to the script code that will empty (clean out) the result set.
8. Add to the script code that will disconnect from the MySQL server.
9. Close out and save the `city.php` file.
10. View the script using the following URL:
`http://localhost/project/city.php?ID=50`

II. Create cities.php file

11. In the `c:\xampp\htdocs\project\` directory create a file called `cities.php`.
12. Edit the `cities.php` file to create an HTML page that has a title of the "MySQL and PHP" and a header that states "Cities of the world".
13. Create a PHP script that connects to the MySQL server that includes the connection information directly entered in to include the `world` database (versus using an external file to draw the information in).
14. Add to the script the ability to catch any errors in connecting to the MySQL server.
15. Add to the script a query against the `City` table in the `world` database that retrieves the `ID` and `Name` for all the records.
16. Add to the script the ability to catch any errors in the query execution.
17. Display the results of the query by placing each record on its own line separating the columns by a dash (`-`). Create a link on the ID output that connects to a page called `cities.php` using the `GET` method to pass on the `ID (cities.php?ID=`*ID*).
18. Add to the script code that will empty (clean out) the result set.
19. Add to the script code that will disconnect from the MySQL server.
20. Close out and save the `cities.php` file.
21. View the script using the following URL:
`http://localhost/project/cities.php`

III. Create navigation.php file

22. In the `c:\xampp\htdocs\project\` directory create a file called `navigation.php`.
23. Edit the `navigation.php` file to create a script container that contains the following text:

```
<p>
<a href="countries.php">Countries</a><br/>
<a href="cities.php">Cities</a><br/>
</p>
```

24. Save and close the `navigation.php` file.

IV. Edit cities.php file

25. In the `c:\xampp\htdocs\project\` directory, edit the `cities.php` file. Add the highlighted text to the top of the PHP script:

```
<?php
include("navigation.php");
# Create Connection to MySQL server
...
```

This will place the navigation elements at the top of the Cities listing page.

26. Save and close the `cities.php` file.

V. Edit countries.php file

27. In the `c:\xampp\htdocs\project\` directory, edit the `countries.php` file. Add the highlighted text to the top of the PHP script:

```
<?php
include("navigation.php");
# Create Connection to MySQL server
...
```

This will place the navigation elements at the top of the Countries listing page.

28. Save and close the `countries.php` file.

VI. Edit city.php file

29. In the `c:\xampp\htdocs\project\` directory, edit the `city.php` file. Add the highlighted text to the top of the PHP script:

```
<?php
include("navigation.php");
# Create Connection to MySQL server
...
```

This will place the navigation elements at the top of the City details page.

30. Save and close the `city.php` file.

VII. Edit country.php file

31. In the `c:\xampp\htdocs\project\` directory, edit the `country.php` file. Add the highlighted text to the top of the PHP script:

```
<?php
include("navigation.php");
# Create Connection to MySQL server
...
```

This will place the navigation elements at the top of the City details page.

32. Save and close the `country.php` file.

VIII. Create city_edit.php file

33. In the `c:\xampp\htdocs\project\` directory, create a file called `city_edit.php` that contains the HTML title of "Edit City".

34. Include the `navigation.php` file to the top of the web output.

35. Create a PHP script that connects to the MySQL server that includes the connection information directly entered in to include the `world` database (versus using an external file to draw the information in).

36. Add to the script the ability to catch any errors in connecting to the MySQL server.

37. Add to the script a query against the `City` table in the `world` database that retrieves the country `ID`, `CountryCode`, `Name`, `Population`, and `District` where the `ID` is equal to the `ID` passed to the script via the `GET` method (`$_GET["ID"]`).

38. Add to the script the ability to catch any errors in the query execution.

39. Create a form that sends the data collected to a file called `city_save.php` via the `GET` method (to be created in later steps). The form should use the data collected to create the following form elements:

Name: *Name*
Population: *Population*
District: *District*

40. Pass on a hidden field with the form that contains the city ID value.

41. Add to the script code that will empty (clean out) the result set.
42. Add to the script code that will disconnect from the MySQL server.
43. Close out and save the `city_edit.php` file.

IX. Create city_save.php file

44. In the `c:\xampp\htdocs\project\` directory, create a file called `city_save.php` that contains the HTML title of "Saving changes to city".
45. Include the `navigation.php` file to the top of the web output.
46. Create a PHP script that connects to the MySQL server that includes the connection information directly entered in to include the `world` database (versus using an external file to draw the information in).
47. Add to the script the ability to catch any errors in connecting to the MySQL server.
48. Add to the script a `REPLACE` query against the `City` table in the `world` database that replaces the existing data in the table with the data submitted through the form (updating the existing record or inserting a new record depending on the city ID being updated).
49. Add to the script the ability to catch any errors in the query execution.
50. Add to the script a link back to the `city.php` file passing the city `ID` using the `GET` method.
51. Add to the script code that will disconnect from the MySQL server.
52. Close out and save the `city_save.php` file.

X. Test pages to edit city information

53. Test the last two files created by entering the following URL:
`http://localhost/project/city_edit.php?ID=500`
54. Modify the name of the city to read `Training Town` with a population of `2`.
55. Submit the changes.
56. Click on the Saved, Show City link to open the `city.php` source.
57. Are the changes made in step 53 visible?

XI. Edit city.php file

58. In the `c:\xampp\htdocs\project\` directory, edit the `city.php` file. Underneath the link to the Wikipedia, make two links. The first link will be called `Edit City` and will pass on the `ID` to the `city_edit.php` file (`city_edit.php?ID=`*ID*). The second link will be called `Delete City` and will pass on the `ID` to the `city_delete.php` (which will be made in steps to follow).
59. Save and close the `city.php` file.

XII. Create city_delete.php file

60. In the `c:\xampp\htdocs\project\` directory, create a file called `city_delete.php` that contains the HTML title of "Deleting city".
61. Include the `navigation.php` file to the top of the web output.
62. Create a PHP script that connects to the MySQL server that includes the connection information directly entered in to include the `world` database (versus using an external file to draw the information in).
63. Add to the script the ability to catch any errors in connecting to the MySQL server.
64. Add to the script a `DELETE` query against the `City` table in the `world` database that deletes the `ID` passed on from the `city.php` file.
65. Add to the script the ability to catch any errors in the query execution.
66. Add to the script a link to the `cities.php` file.
67. Add to the script code that will disconnect from the MySQL server.
68. Close out and save the `city_delete.php` file.
69. View the `cities.php` file. Locate the city with the `ID` of `542`. Select that link.
70. In the `city.php` file, the city of `Burgas` should be displayed. Select the `Delete City` link.
71. Choose the `Deleted, Show Cities` link. Locate the city `ID` just deleted. Is it visible in the list of cities?

XIII. Create and Edit Country Files

72. Using the city web pages as templates, create the ability to update and delete countries in the `world` database.