**Unit 13 Lab – Building a Web Application with JavaScript**

**Exercise 1 – Public Holidays using Fetch – part 1**

Take the code for exercise-5-class-generic-table-from-generic-data of unit-12 (JavaScript Advanced) and allow the user to select the public holiday information for the following countries; Ireland, France and Spain and the following years: 2024, 2025 and 2026

This exercise primarily affects the JavaScript code but you will need to add some classes to style the dropdown lists

1. Create a new folder called “unit 13” containing a folder called exercise-1-class-public-holidays-table
2. Copy either your own solution code for exercise-5 of unit-12 or the solution code as provided
3. Rename the class to “PublicHolidaysDataTable” to reflect the purpose of the class
4. Rename the “loadProductData” method to “loadData”
5. Update the class properties to the following;
   1. // properties
   2. #dataUrl = "https://date.nager.at/api/v3/PublicHolidays/"; // base URL to REST API
   3. #country = "IE"; // default country
   4. #year = "2024"; // default year
   5. #title; // title text for component
   6. #data; // data object returned via fetch()
   7. #componentRoot; // DOM node for component
   8. #columnNames; // an array of column names to display
6. In the main.js file, add two select controls (country and year) with the specified values to the render() method
7. Add a handleEvent() method to the class and
   1. extract the dropdown name and value for the event object and update the country and year properties
   2. call the “loadData” method
8. In the class constructor add an event listener for the “change” event for the class
9. Using the documentation for the public holiday REST API (<https://date.nager.at/swagger/index.html>)
10. Test your code

**Exercise 2 – Public Holidays using Fetch – part 2**

Building on exercise 1 replace the hardcoded list of values for the two dropdown lists with dynamic lists

1. Create a new folder called exercise-2-class-public-holidays-table
2. Copy either your own solution code for exercise-1 or the solution code as provided to the new folder
3. The api documentation (<https://date.nager.at/swagger/index.html>) outlines an URL which returns the list of available countries
4. Add two new private properties to the class:
   1. #dropdownCountries // an array of the available countries as returned by the api
   2. #dropdownYears // an array of the years available
5. Add a new method called “loadCountries” method which fetches the list of available countries and saves the returned array to the #dropdownCountries property
6. Add a new method called “loadYears” method which returns an array of 10 years from the current year forward, i.e. [2024, 2025, …. 2033]
7. Add new renderCountries() and renderYears() methods which returns the respective html for each
8. Update the “render” method to replace the hard-coded values in the two dropdown lists with the values returned by the “renderCountries” and “renderYears” methods respectively
9. Ensure that the currently selected value for both dropdowns is displayed, i.e. set the “selected” attribute for the corresponding option element
10. Test your code

**Exercise 3 – Public Holidays using Fetch – part 3**

Building on exercise 2:

* move the code for the PublicHolidaysDataTable class to its own JavaScript file and import it into the main.js file
* move the defaults for the country, year, title and columnsNames properties to a configuration object which is passed into the PublicHolidaysDataTable constructor when the new instance of the class is created in the init function

1. Create a new folder called exercise-2-class-public-holidays-table
2. Copy the code from exercise-2-class-public-holidays-table
3. Create a new folder called “modules” in the “js” folder
4. Move the JS code for the PublicHolidaysDataTable class to it’s own file in the “modules” folder
5. Add an “export” statement to this new file
6. Replace the class code in the main.js file with an import statement from the PublicHolidaysDataTable class
7. Replace the individual properties for country, year, title and columnsNames properties with a single private property called config
8. In the constructor update the config property with the config object passed into the constructor
9. Update the class code to use the individual values in the config object
10. Test your code. It should function as before