# CA3 - ASP.Net Core - Blazor WASM Project

## Introduction

For this project I created a Blazor WebAssembly Standalone application using Visual Studio 2022.

Creating this project gave me a simple working Blazor application. With a home page, a counter, and a page for the weather forecast.

## My Project

### Project Idea

This project will be a brewery web application. I want to get a list of breweries in Ireland and display them on the landing page. The list of breweries should be sortable by brewery name, and we should be able to filter breweries by county as well.

The data will be displayed in a QuickGrid with pagination to make it easier for the user to view the list.

### API

For this project I used the Open Brewery API which can be found at <https://www.openbrewerydb.org/>. This API contains a list of breweries in Ireland which is what I decided to use.

### Landing page – Brewery List

A screenshot of a computer

Description automatically generated

This is the landing page for the web app. The landing page, by default, displays breweries from all counties in Ireland, displays 10 items per page, and orders the data in ascending order according to the brewery name. The buttons at the end of the card can be used to navigate through the pages of data.

#### Filter By County

A screenshot of a computer

Description automatically generated

The breweries can be filtered by county. By default, it shows breweries from all counties but when a county is selected from the drop-down list, the header of the page changes and the items are filtered to only show breweries from the selected county.

#### Sort by Brewery Name

A screenshot of a computer

Description automatically generated🡪A screenshot of a computer

Description automatically generated

The data by default is order ascending according to brewery name. The sort order can be changed by clicking the column header for brewery name.

#### Pagination

A screenshot of a computer

Description automatically generated🡪A screenshot of a beer search

Description automatically generated

By default, the data is shown with 10 items per page. This can be changed using the drop-down list entitled “Items per page”.

### Brewery Detail Page

To view the detail page for each brewery, the “More details” button should be pressed from the brewery list page.

A screenshot of a phone

Description automatically generated

This button opens the brewery detail page for the selected brewery.

A screenshot of a bar

Description automatically generated

The brewery datil page shows the information for the brewery. The image at the top of the card is randomly selected from the /wwroot/images/ folder and will be different every time the detail page is opened. The brewery name is displayed and below there is some details listed for the bar including the website link. The last element in the card is an iframe that shows the brewery location in Google maps.

The link at the top right-hand corner brings the user back to the list of breweries.

A screenshot of a bar

Description automatically generated

### Testing

This project has been tested using selenium IDE. There are three tests in my test file:

1. That the URL opens the correct landing page.
2. When a county filter is applied, the header of the page changes.
3. That the content displayed on the page is correctly

All tests pass with flying colours and the test file is uploaded on to the github repo.

A screenshot of a computer

Description automatically generated

#### Viewing the website in different browsers

Chrome - Version 119.0.6045.160 (Official Build) (64-bit)

A screenshot of a computer

Description automatically generated

Microsoft Edge – Version 119.0.2151.72 (Official build) (64-bit)

A screenshot of a computer

Description automatically generated

### Deployment

This project is deployed on Azure and can be found at: <https://orange-stone-09d917710.4.azurestaticapps.net/>.

### Github

The code for this project can be found on my Github at: <https://github.com/rachelrring/CA3>.

## Conclusion

This web app has everything that was listed in the brief within reason:

* Sorting
* Paging
* Filtering

I’m proud of how it turned out!