Contents

[Planning Analysis Sheet 3](#_Toc153012191)

[1. Design of Website 3](#_Toc153012192)

[Website Link 3](#_Toc153012193)

[Website Title 3](#_Toc153012194)

[Website Goal: 3](#_Toc153012195)

[Wireframe: 3](#_Toc153012196)

[Website Structure: 4](#_Toc153012197)

[Source of images: 4](#_Toc153012198)

[Styles Sheet: 4](#_Toc153012199)

[Home Page 4](#_Toc153012200)

[Header 5](#_Toc153012201)

[Navigation 5](#_Toc153012202)

[HomePage 5](#_Toc153012203)

[HTML 5](#_Toc153012204)

[CSS 5](#_Toc153012205)

[Javascript 5](#_Toc153012206)

[Herd Management 5](#_Toc153012207)

[HTML 5](#_Toc153012208)

[CSS 6](#_Toc153012209)

[Javascript 6](#_Toc153012210)

[AgriIoT 6](#_Toc153012211)

[HTML 6](#_Toc153012212)

[CSS 6](#_Toc153012213)

[Javascript 6](#_Toc153012214)

[Slurry 6](#_Toc153012215)

[HTML 7](#_Toc153012216)

[CSS 7](#_Toc153012217)

[Javascript 7](#_Toc153012218)

[Winter Fodder Management 7](#_Toc153012219)

[HTML 7](#_Toc153012220)

[CSS 7](#_Toc153012221)

[Javascript 7](#_Toc153012222)

[Javascript- Event Handlers 7](#_Toc153012223)

[Javascript- Functions 7](#_Toc153012224)

[Time Management 7](#_Toc153012225)

[HTML 7](#_Toc153012226)

[CSS 7](#_Toc153012227)

[Javascript 7](#_Toc153012228)

[About Us Page 8](#_Toc153012229)

[HTML 8](#_Toc153012230)

[CSS 8](#_Toc153012231)

[Javascript 8](#_Toc153012232)

[Contact Us Page 8](#_Toc153012233)

[Membership Page 8](#_Toc153012234)

[Register Animal 8](#_Toc153012235)

[Styles.css 8](#_Toc153012236)

[General Notes on Website Design: 9](#_Toc153012237)

[2. Testing & Check of Web Accessibility & Performance 9](#_Toc153012238)

[Testing with Desktop 9](#_Toc153012239)

[Testing with Mobile 9](#_Toc153012240)

[Testing with Tablet 10](#_Toc153012241)

[Web Accessibility 10](#_Toc153012242)

[SilkTide 10](#_Toc153012243)

[Page Speed Insights 10](#_Toc153012244)

[3. Search Engine Optimisation 12](#_Toc153012245)

[Screaming Frog 12](#_Toc153012246)

[HTML Validation 13](#_Toc153012247)

[CSS Validation 13](#_Toc153012248)

[Google Search Central 13](#_Toc153012249)

[References 15](#_Toc153012250)

[Agri References 15](#_Toc153012251)

# Planning Analysis Sheet

The Project was split into four phases.

1. Identifying the gaps in current Smart Agri Technology today
2. Design of Website for Farmers that can address some of these challenges
3. Testing of Website on different Browers and Devices
4. Search Engine Optimisation (SEO)

# Design of Website

## Website Link

The website is accessible at <https://ivcos.github.io/DairyCowManager/AboutUs.html>

## Website Title

FARMAI – Farmers in Control of Their Data

## Website Goal:

The goal of this website is to provide a Smart Agriculture Management System that is based purely on data. The goal of the website is to highlight to famers the capabilities of the technology our fictious company FarmAI support. Tools are also available for the farmer on the Website that allow them;

* to calculate winter fodder requirements (WinterFodderManagement\_grid.html).
* slurry tank capacity requirement (SlurryManagement.html) and slurry capacity required.
* register animal birth, new animal, AI and sale of animals (HerdManagement.html)

I have attempted to keep the website as simple as I have had feedback that some farmers do have not have good IT skills. So, I attempted to

* Make the Website easy to navigate.
* Make the Website mobile friendly as data would be entered from the phone in most case
* Allow the data to be entered easily for the tools and Forms available on the Website

## Wireframe:

The wireframe is a sketch or blueprint of the site, it shows the structure of the basic page including the elements.

* Header
* Navigation
  + Primary Navigation changed to a burger bar on mobile,
* Content

In general, the website follows the Responsive Layout with media queries as outlined in page 38/50 of the Unit 7 slides in the First set of lectures on HTML and CSS. I have used Flexbox and CSS Grid as much as possible. The flex layout when used means that items can shrink and stretch inside their containers preventing wasted space and overload , making the layouts to fit a variety of view port sizes.

A screenshot of a wireframe

Description automatically generated

*Figure 1 Wireframe Layout of Website*

## Website Structure:

Figure 2 shows the basic structure of the FarmAI Site Layout of the planned layout for this site. However note that only some of the functionality is not implemented in the submission.

A diagram of a company

Description automatically generated

*Figure 2 FarmAI Site Diagram*

## Source of images:

* Photographs taken from an Irish farm and some stock inages from unsplash

## Styles Sheet:

* A single external style sheet was used as specified in the assignment

## Home Page

1. The Home page include an introduction and describes FarMAI features

# Header

The Header is made up the logo and the H1 heading.

# Navigation

The navigation is built as a list of links to provide semantic meaning. This is an unordered list. The **link**, **visited**, **hover**, and **active** are pseudo-classes used to define the styles of links based on their states are define for the main navigation bar. This allows different appearances for links depending on whether they are being hovered over, have been visited, or are active (clicked). All the links are defined as inline-block.

The navigation bar changes to a burger bar at 768px. ChatGpt3.5 was used to provide sample code on how a burger bar could be built. The code was modified to suit this project. Additions were made to the styles.css and a function was added to each .js file to make the navigation bar consistent across all pages.

# HomePage

The Homepage describes the capability of the services offered by FarmAI. In the near future, all services will be available on the App. Each of the main services are described on the home page. For now some of the services are not available with the Web version.

## HTML

Each of the main items are flex boxes. I had some problem get the image and paragraph lining up. The image in the flexbox was disappearing when the flexbox was set to column.

I found some material on the <https://stackoverflow.com/questions/69028147/flexbox-image-disappear-with-column>.

## CSS

Media Queries are applied to make the home page responsive. All images with the exception of the first one are removed from the view when on viewing the Home page on a mobile. The first image was included in the mobile view and all other images were hidden.

## Javascript

Javascript for the Home page is included in the index.js file.

# Herd Management

This webpage provides a number of options to the farmer for Herd Management.

## HTML

In the Herd Management page, there are a number of menu options presented to the user for, when clicked a corresponding form will be opened.;

1. Birth Registration- Register the birth of the a calf .
2. Animal Registration – A new animal bought into the farm.
3. AI Registration – Register Artificial insemination (with Calving date automatically calculated)
4. Animal Sale
5. Herd Health and EBI (This is a Roadmap feature)
6. Update Animal Information: lost Tag, animal health etc (This is a Roadmap feature)

I have used Flexbox feature predominantly in this page as I wanted the main top-level flexbox to change as the screen widths vary. The six flex items themselves are fixed but this works quite well as each of the flex items stack nicely in tablet and mobile view.

## CSS

Check if there are media queries ?

## Javascript

There are number of functions and eventlisteners defined in the RegisterCowandCalf.js files

* Function ***hideFlexItems()*** – Hides all the flex item options when the user has selected an option.
* Function dislayFlexItems – Displays all the flex items when the user has closed the form (error here with close form)
* Function ***getAIDate()-*** Auto populates the AI Form with the current Data and automatically updates the Calving Date text box by adding nine months and 7 days. [Note to Self 1]. Need to add javascript to auto update the calving date if the user manually changes the date in the form.
* Event Listeners are created for each of these six buttons/options on the HerdManagement.html. All forms are hidden initially and are revealed when the selects one of the options. Each individual form has several buttons for which there are also Event Listeners created.
* Event Listeners for each Button on each of the Forms.
* Function ***formatDate()*** The date is automatically populated with today’s date.
* Some of the fields have default data.
* Function ***validateAlphanumeric ().*** Validation to ensure that the user has entered valid data in the first three fields in the form. Only alphanumeric character are allowed to be entered. If the user enters anything else, then an error will be generated and displayed under the label where the erroneous value is entered.
* Functions for each of the Form submissions

# AgriIoT

## HTML

## CSS

## Javascript

# Slurry

This page provides;

* a tool that allows the farmer to calculate the slurry tank capacity on his farm.
* a tool to allows the farmer to calculate the estimated slurry that will be produced by his herd of animals.

If the capacity is greater than the expected volume, then the farmer has sufficient capacity for the winter period. If the expected volume of slurry exceeds capacity, it means that the farmer does not have enough slurry tank capacity.

## HTML

There are two CSS Grids setup for the Slurry Capacity Calculator and the Slurry Storage Required.

## CSS

## Javascript

The function calculate automatically returns the capacity of the Slurry Tank based on the data input by the user. The user is not allowed edit the Capacity value as it is set to read only.

# Winter Fodder Management

This page provides a tool to allow the farmer to input the numbers of cattle of different ages that on the farm and the tool automatically calculate the amount of winter fodder that he requires for the winter period. The tool works on all devices.

## HTML

Header – As Standard across all pages

Navigation- Standard across all pages

The calculator is a Grid (class= “grid-container”).

## CSS

The styling specific for WinterFodderManagement\_grid.html begins at line 746 in the styles.css.

## Javascript

The Javascript file is WinterFodderManagement.js. This file includes a number of event handlers and functions.

### Javascript- Event Handlers

When the HTML page has been loaded and parsed, the “DOMContentLoaded” event is triggered. The event handler function defined in WinterFodderManagement.js triggers two other event handlers to be setup; the eventlistener for the calculate silage button and the clear button. Once the DOM is loaded the program has now entered the event driven phase of the program

1. EventListener for button “Calculate Silage Required”
2. EventListener for button “Clear to reset values”

### Javascript- Functions

# Time Management

## HTML

## CSS

## Javascript

## About Us Page

1. The About Us Page describes who we are and the mission statement of the website.

## HTML

Basic HTML Page

## CSS

Small amount of HTML

## Javascript

The Javascript code for the carousel was developed with the assistance of ChatGPT3.5.

## Contact Us Page

Use a Flexbox for main

## Membership Page

## Register Animal

This page has the form that includes a Form to Register an animal at birth or add any animal. In the first project, I used the standard Form Layout. But this time, I have made the Form more responsive having read Chapter 19 (Learning Web Design, Robbins) by making each <li> item more responsive. I’ve used Flexbox to make the labels stack on top of their respective inputs when and fieldsets on narrow screens, so that there is no wasted space.

By default, browsers may use different fonts at different sizes for various input types. We would prefer that all inputs have the same font as the surrounding text on the page, you can force form elements to inherit font settings. (Pg. 548)

*/\* Corodinated Fonts across for all form elements. Pg. 548\*/*

button, input, select, textarea {

    font-family: inherit;

    font-size: 100%;

}

1. Consider an option where the main is the parent is a flexbox, form is a flexitem for this form like the [Design A Cool Responsive Registration Form Using HTML & CSS Flexbox - YouTube](https://www.youtube.com/watch?v=JpNr5aEIxSI) used body to be the parent item and the Form to be the Flex item. But I will have other elements so it would be better to have main as flexbox and Form as Flexitem.
2. The good thing about this is that building a Form using Flexbox means that w edon’t have to use media queries.
3. [CSS Flexbox Responsive Form Layout With Source Code (softauthor.com)](https://softauthor.com/css-flexbox-responsive-registration-form-with-source-code/)

## Styles.css

## General Notes on Website Design:

1. One style sheet for all Web pages as the look must be consistent across all HTML pages.
2. The Assignment sheet has stated that popular browsers including Chrome, Edge, Firefox must render each page correctly. Based on this, the submitted project may not render well for IE users. Only a small number of browsers support Internet Explorer, “1.5% of browser traffic in 2017 was using Internet Explorer versions 8 and earlier” (Robins, 2017)
3. For the sidebar, chose to use the ***<aside>*** HTML element as it is semantically appropriate for the content, which are supplementary links that are not directly related to the content but provides additional information. The content in the side bar has a semantic relationship with the main content. <div> would be used if the sidebar was for layout and styling.
4. The largest width for smart phones is 412px, so this is the default.
5. ***Width= device-width*** on all html pages. This tells the browser to set the width of the viewport to the width of the device screen. The initial scale is set to 1 (100%)
6. Used internal links to navigate in the home page. Add a link back to the top of the Home page to make navigation easier.
7. Used the information in <https://stackoverflow.com/questions/6885099/css-html-javascript-tricks-to-print-a-web-page-without-images> to prevent printing of the images to avoid unnecessary use of color ink. Also prevent the printing of the aside links as these links while related to soccer would provide any value in a printed-out form. Also stopped the printing the ***<nav>*** as advised in lecture notes. <figure> elements are also not printed.

# Testing & Check of Web Accessibility & Performance

The project was tested on browsers and devices below.

## Testing with Desktop



*Figure 3 Browser Testing Results for Desktop*

## Testing with Mobile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | iPhone(portrait) | iPhone(landscape) | Android(Potrait) | Android(Landscape) | 3G Coverage (iPhone) |
| Page | Safari | | Chrome | | Safari |
| Index.html | Ok | Ok | Ok | Ok | Ok |
| About Us | OK | OK | OK | OK | OK |
| Contact Us | OK | OK | OK | OK | OK |
| Membership | OK | OK | OK | OK | OK |

*Figure 4 Browser Testing Results for Mobile*

## Testing with Tablet

|  |  |
| --- | --- |
| Ipad (potrait) | Ipad (landscape) |
| Safari | |
| Note 1 | Note 1 |
| Note 1 | Note 1 |
| Note 1 | Note 1 |
| Note 1 | Note 1 |

*Figure 5 Browser Testing Results for iPad*

Note:

1. There was one issue with the Safari on IPad running Safari where the header was being extended. The IPad was running on older versions of IOS, 12.5.7. But this was not happening with the Safari Browers using the “Toggle Device Toolbar.

## Web Accessibility

Ran the WAVE tool on website on all four pages. One Contrast Error across the four pages. It is the ‘Embed from Getty Images”. Investigating how to change this ‘grey’ font with Javascript.

## SilkTide

Ran the Silktide tool on the four pages, just focusing on Blindness. Overall the pages read like were meant to, so semantically the site is setup well.

## Page Speed Insights

Below are the result for from pagespeed.web.dev carried out on the 30th September.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Mobile** | | | |
| Page | Performance(%) | Accessibility(%) | Best Practices(%) | SEO(%) |
| index.html | 97 | 86 | 100 | 100 |
| About\_Us,html | 100 | 100 | 100 | 100 |
| Contact\_US.html | 100 | 100 | 100 | 100 |
| Memership.HTML | 100 | 100 | 100 | 100 |

*Figure 6 PageSpeed Results for Mobile. See Note 1 below.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Desktop** | | | |
| Page | Performance(%) | Accessibility(%) | Best Practices(%) | SEO(%) |
| index.html | 100 | 86 | 100 | 100 |
| About\_Us,html | 100 | 100 | 100 | 100 |
| Contact\_US.html | 100 | 100 | 100 | 100 |
| Memership.HTML | 100 | 100 | 100 | 100 |

*Figure 7 PageSpeed Results for Desktop. See Note 1 below.*

Note 1: The main page(index.html) is performing poorly as shown in the figure below.

# Search Engine Optimisation

## Screaming Frog

From lecture 11, important to have meta tags on each page. So used the name=Description” meta tag to describe each page in a meaningful way to increase chances of search engines finding the site.

A screenshot of a computer

Description automatically generated

*Figure 6 Screamingfrog*

A screenshot of a computer

Description automatically generated

*Figure7 Screamingfrog, Each html page has a meta description and is less than 150 characters.*

Added;

<link rel="canonical" href="https://ivcos.github.io/soccer\_fans\_project/membership.html">

ChatGPT

* **SEO**: Using **rel="canonical"** helps prevent SEO issues related to duplicate content and ensures that search engines index the right URL, which can improve search rankings.
* **User Experience**: It can also improve the user experience by ensuring that users are consistently directed to the preferred version of a page.

Added meta tag to provide a description of the page to improve SEO. I used a sentence in the meta name=”description” rather than individual key terms.

## HTML Validation

SEO is improved if there are no HTML errors. HTML validation completed on <https://validator.w3.org/> for all four files with no errors. All four HTML files passed on the 23/09/2023

## CSS Validation

SEO is improved if there are no CSS errors. CSS Validation passed on the 30/09/2023

## Google Search Central

1. Checked if site was on Google.
2. <https://developers.google.com/search/docs/crawling-indexing/sitemaps/overview>
3. Logged into Google Search Console and uploaded a Google.html file to Github to verify ownership verification of the link. I used the HTML verification method.
4. Sitemap successfully submitted.

A screenshot of a computer

Description automatically generated

*Figure 8 Message that Sitemap was submitted successfully.*

However, was getting an error, saying there was an error as I was loading HTML instead of XML. Generated an XML file using ScreamingFrog.

A screenshot of a computer

Description automatically generated

*Figure 9 ScreamingFrom generated sitemap xml file*

Below is the XML File generated.

<?xml version="1.0" encoding="utf-8"?><!--Generated by Screaming Frog SEO Spider 19.2-->

<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">

<url>

<loc>https://ivcos.github.io/soccer\_fans\_project/index.html</loc>

<lastmod>2023-10-01</lastmod>

<changefreq>daily</changefreq>

<priority>1.0</priority>

</url>

<url>

<loc>https://ivcos.github.io/soccer\_fans\_project/contact\_us.html</loc>

<lastmod>2023-10-01</lastmod>

<changefreq>daily</changefreq>

<priority>0.9</priority>

</url>

<url>

<loc>https://ivcos.github.io/soccer\_fans\_project/Duck&amp;DrakeClub.html</loc>

<lastmod>2023-10-01</lastmod>

<changefreq>daily</changefreq>

<priority>0.9</priority>

</url>

<url>

<loc>https://ivcos.github.io/soccer\_fans\_project/about.html</loc>

<lastmod>2023-10-01</lastmod>

<changefreq>daily</changefreq>

<priority>0.9</priority>

</url>

<url>

<loc>https://ivcos.github.io/soccer\_fans\_project/membership.html</loc>

<lastmod>2023-10-01</lastmod>

<changefreq>daily</changefreq>

<priority>0.9</priority>

</url>

</urlset>

Add the file to the project folder and committed to GitHub. Now the sitemap for my site had been successfully submitted.

A screenshot of a computer

Description automatically generated

*Figure 10 sitemap xml file successfully loaded.*

I will continue to monitor Google Search Console to see how my site is performing.

## References

Robbins, J. (2018). Learning Web Design. 5th ed. O’Reilly: CA95472.

## Agri References

Liver Fluke – The high levels of rainfall and mild temperature provide ideal conditions fpr flue to thrive in. ‘Dosing Plans are priority at housing begins”

[2019 - Assessing the Liver Fluke Risk on your Farm - Teagasc | Agriculture and Food Development Authority](https://www.teagasc.ie/publications/2019/assessing-the-liver-fluke-risk-on-your-farm.php)

Liver Fluke is caused by a flat leaf-like worm called Fasciola Hepatica. An intermediate host, the mud snail is also involved in the life cycle. Infection with liver fluke occurs when pasture that is contaminated with fluke is eaten by the animal. Once eaten fluke starts to feed and grow. It takes approximately twelve weeks for the flukes to grow to adult stage when they start to lay eggs. These eggs pass out in the faeces of the animal and when conditions are suitable (when temperatures go above 10 degrees) they hatch and use the mud snail to continue the life cycle. During this twelve week period the fluke are classified according to their stage of development:

[Liver fluke.pdf (afbini.gov.uk)](https://www.afbini.gov.uk/sites/afbini.gov.uk/files/publications/%5Bcurrent-domain%3Amachine-name%5D/Liver%20fluke.pdf)

Slurry Storage Capacity Calculator

[Slurry tank capacity - Teagasc | Agriculture and Food Development Authority](https://www.teagasc.ie/rural-economy/farm-management/farm-machinery/machinery-calibration/slurry-tank-capacity/)

[CSS Flexbox Responsive Form Layout With Source Code (softauthor.com)](https://softauthor.com/css-flexbox-responsive-registration-form-with-source-code/)

IFA gathering support for new data charter for farmers

Notes from HTML/CSS

1. Never use Ids, always use Class
2. Use the Hover over Button (Lecture 33, 16:57)

A screen shot of a computer

Description automatically generated

Herd Management Page