# 

# Final Document

# SWEN5236.21 Engineering Software I

# Product Name: FAQ and Cheat Sheets search widget

Dates:

Team Members:

Professor:

Table of Contents

[Project Overview 3](#_Toc66211545)

[Technologies Used 3](#_Toc66211546)

[Resources](#_Toc66211547) Used……………………………………………………………………………..3

[Feature](#_Toc66211548) of the Project……………………………………………………………………….3

[Tasks 4](#_Toc66211549)

[Feature 1: Creation of static pages and navigation to the corresponding page when hyperlink is clicked in the left menu 4](#_Toc66211550)

[Feature 2: Search widget implementation to navigate to the corresponding based on the text search 6](#_Toc66211551)

[Feature 3: Implementation of negative scenarios like the module does not present for the text entered and testing 8](#_Toc66211552)

[Supporting Concepts & Materials 9](#_Toc66211554)

[Setup Guide to Run the Project 11](#_Toc66211555)

[Future Scope 11](#_Toc66211556)

[References 11](#_Toc66211558)

# 

# Project Overview

FAQ and Cheat sheets search widget is a static C# website where we can search for a module to get the frequently asked questions for that module and some code blocks which can be useful for quick overview of the module. This is helpful for a quick overview of the module to go through whenever we want.

We can also go through individual module from the hyperlinks provided in the left menu of the application.

**Technologies Used**

* C#
* jQuery
* MVC
* thiscodeworks.com

**Resources Used**

* Visual Studio 2019
* Github

**Features of the Project**

**Table 1. User Story Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Requirements | Priority | Sprint Due |
| 1 | Creation of static pages and navigation to the corresponding page when hyperlink is clicked in the left menu | High |  |
| 2 | Search widget implementation to navigate to the corresponding based on the text search | High |  |
| 3 | Implementation of negative scenarios like the module does not present for the text entered and testing | Low |  |

**Feature 1: Creation of static pages and navigation to the corresponding page when hyperlink is clicked in the left menu**

* Create menu to the left of the page with the modules for which static pages will be created and redirected when clicked on the menu

Graphical user interface, application

Description automatically generated

*Image 1:* Left hyperlink menu

* Create static pages to navigate to the corresponding page which shows frequently asked questions and cheat sheets for the module clicked.

Graphical user interface, text, application, email

Description automatically generated

*Image 2:* FAQ page when functional programming hyperlink is clicked

* Project structure in visual studio

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

*Image 3:* Project structure

* + \_Layout.cshtml is partial view which holds the code to display the search text and button
  + SideMenu.cshtml is another partial view which is used in all static pages to display the left hyperlink menu
  + .cshtml files present in the home are the files which holds the static content of the pages
  + Utility.cs holds the implementation of the search widget logic which gives us the corresponding redirect page when we search with a particular text
  + HomeController.cs contains the logic to route to the corresponding page when clicked on hyperlink or through search

**Feature 1: Search widget implementation to navigate to the corresponding based on the text search**

* Create a textbox to enter the search text and button to search

Graphical user interface, text, application

Description automatically generated

*Image 4:* Search button and textbox

* Redirecting to corresponding page when we search with a text

Graphical user interface, text, application, email

Description automatically generated

*Image 4:* Navigating to Tuples page when searched with text tuples

Graphical user interface, text, application

Description automatically generated

*Image 5:* Navigating to Options page when searched with text options

* Redirecting to content page when partial search is matching. This feature is useful when there is some spelling mistake in the text entered by the user or if the user enters short text like tuple, tupl, tup, tuplle tupel, tuple for the tuples search

Graphical user interface, text, application, email

Description automatically generated

*Image 6:* Partial search for text tuple

Text

Description automatically generated

*Image 7:* Partial search for text higher

**Feature 3: Implementation of negative scenarios like the module does not present for the text entered and testing**

* There is a case where if there is no module with the text user searched then we will show error page which specifies that the module the user searched is not present so that the user can enter the valid search text to get results.

Graphical user interface, text, application

Description automatically generated

*Image 8:* Result when searched with text tule

Graphical user interface, text, application, email

Description automatically generated

*Image 9:* Result when searched with text delte

* Also, when the website is loaded, we have to display a default message to user

Graphical user interface, text, application, email

Description automatically generated

*Image 10:* Default text when the website is loaded

[**Supporting Concepts & Materials**](#_Toc66211554)

* We used thisCodeWorks.com website which is open source to add code blocks to the content pages. We can simply add code in the website then we can get the embed code and add this code in the content page which is very easy to use and just plug and play strategy.

Graphical user interface, text, application, email

Description automatically generated

*Image 11:* Creating code block

Graphical user interface, text, application

Description automatically generated

*Image 12:* Embed link which holds the embed link

Text

Description automatically generated

*Image 13:* Code that be embedded in our site

Graphical user interface, text

Description automatically generated

*Image 14:* After adding the embedded link it appears like this in our website

[**Setup Guide to Run the Project**](#_Toc66211555)

* Prerequisite
  + [Latest .NET Framework SDK]
  + <https://dotnet.microsoft.com/download/dotnet-framework/net48>
* Visual Studio
  + Open the .sln file with Visual Studio
  + Run the project (press `F5` key)

**Future Scope**

* Making the content dynamic to load from database or cloud
* Autocomplete for search textbox
* Single page application so that there won’t be redirects for each module transition

**References**

* <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>
* <https://www.thiscodeworks.com/>
* Functional Programming in C How to write better C code by Enrico Buonanno (z-lib.org)