

COMP 2772 / COMP 8772 Assignment 2

Due Date: 1st of November

Weighting (COMP2772): 35%

Weighting (COMP8772): 25%

Late penalty: 5% per day late, or part thereof

Overview

For this assignment you are tasked with creating a web page in Single Page Application (SPA) format. As with the course labs you will be creating a blogging style website where users can input posts and have those posts appear, serialised, on the main page.

The assignment is comprised of two primary components:

- A Brief document acting as a style guide and readme
- The SPA, running in a Node.JS environment

Submission

Your submission should be a single .zip file containing the contents of your SPA as well as your style guide document in PDF format. The zip should contain the full SPA structure including node libraries and folders. If additional libraries beyond those provided in the default project are used, these should be noted in your readme/style guide.

Assignment Specification

As stated there are two primary components to the assignment. The first is a style guide that details the layout, contents and style elements chosen for your page. The second is the actual implementation including any HTML, templates, style sheets, scripts and modifications to the original structure made.

Style Guide / Specification document

This document should contain a brief overview of the structures used and the styles implemented within the application, as well as the functional behaviour of the various SPA elements.

1. Project structure

The document should clearly lay out the project structure including folders used and communication structures utilised within the application. This includes the location of templates, scripts and style sheets within the project. It should also include details of any routes employed and deep linking solutions utilised to preserve state.

2. Style guidelines

The document should contain the details of styles chosen. This should include elements like font selection, colour pallet and iconography. The style guide portion should also include details of the various view / template elements of the page and their purpose. This should include layout choices and other design details. Any responsive functionality included in the application should also be detailed here, along with the trigger parameters for such features. Any alternate/advanced style implementations should be detailed in this section including animation flex and effect behaviour. Lastly the content described in this section should appear within your page, it should not contain details of style and outline that you wanted or considered using, rather what you did use.

3. SPA Functionality

Lastly the document should contain the functional details for the app including element communication, back end functionality and any additional modules employed. This should include the purpose of any of your own script files, JavaScript / AngularJS implementations and structure.

If any additional work has been done on the back end server, or additional Node.JS modules used these should be detailed along with the functionality they provide to the SPA. Lastly this should detail the Node.JS run specification and port used in the final result; this will usually be localhost:8000 or localhost:3000 but should be clarified here.

A note for 8772 Students

It is imperative that you take significant notes on your decisions regarding style, implementation and structure for the assignment. A key component of the Comp8772 research assignment will involve reflections on these choices in a more comprehensive manner than is to be presented here.

Single Page Application

The primary component of the assignment is the development of a single page application that meets the specifications laid out below. This page must run within a Node.JS on the localhost or localhost equivalent. **Implementations that do not run in a server environment will be awarded zero marks.**

The implementation style and layout are up to you and should be outlined in the style guide. Note that you are not being marked on your graphical design skills. While the tools implemented in the solution will be awarded marks (e.g. the use of CSS animations classes) the physical appearance of art assets will not.

The following guidelines outline the minimum views required to meet the specification. In the case of all pages' sufficient lorem ipsum content can be generated for <https://lipsum.com/> and used to provide mock text content.

Required Pages

Your solution should make use of the following pages

- Content / News page
- Content creation page
- About page
- FAQ page

Note that these are the minimum required pages, if you wish to add more for additional functionality that is fine.

1. Content / News page

This page should act as the landing pad for your application as well as the page responsible for displaying news posts. News posts should be clearly styled and marked up including at a minimum:

- Post Title
- Author
- Date Created
- Tags

Styling should make each element easily identifiable and clear with all three primary elements (header/content/footer) of the template defined.

2. Content creation page

This page should act as the primary point for post creation. The page should make use of the form element and attributes to allow users to create content. The form should make use of suitable CSS styling and pseudo classes to make it clear to the user when elements are focused or selected as well as to draw the user to the creation elements of the page. Sufficient input fields should be provided to meet the requirements of the news page layout.

3. About page

This page represents generic about content and should be structured and laid out to draw focus to and highlight this content. Textual content can be filled in with lorem ipsum text of sufficient length to demonstrate the styling and structure choices.

4. FAQ page

The FAQ page should present the intended layout of a hypothetical Frequently Asked Questions page. Key elements of this page are 10 question entries. The body and title content of the questions should use Placeholder / Lorem Ipsum text.

However, the page should include a content section that provides links to each question on the page, and functionality to navigate to that question when clicked.

Additional information

In addition to these 4 pages your SPA should include ancillary views for navigating and partitioning your page. This should be in the form of top or side navigation options, header and footer elements and any other elements needed to create a cohesive SPA environment.

The final page structure should be resilient to simple shape changes (unless responsive design has been incorporated into your page as an additional feature). This means that content should stay within the confines of defined spaces and elements should style without unnecessary overflow or visual aberration.

Marking Specification

Due to the number of possible variations available with both the content and technologies that can be employed in the assignment please take the time to familiarise yourself with the marking guidelines to properly understand the requirements of each rubric tier.

Coding Style (5 Marks)

Description	Mark
<ul style="list-style-type: none">• Code is jagged and makes no effective use of white space.• Elements are not indented.• Variable names are meaningless and arbitrary. File structure is singular and poorly structured. Files contain no comments.	1
<ul style="list-style-type: none">• Code is correctly formatted and makes acceptable use of whitespace.• Elements are clearly aligned and indented• Variable names and object construction is clear, logical and meaningful.• File structure is logical, clear and meaningful.• Code is clearly commented to explain function behaviour	5

Style \ Specification Document \ Project structure (15 Marks)

Note that a key component of the marks awarded for the Style\Specification document is that the SPA follows the outlines provided. As such the overall structure of the project is included in these marks and should follow the design laid out.

Description	Mark
<ul style="list-style-type: none">• Project structure is not laid out or not followed• Project contains little to no templating• Functional structures are incorrectly defined or detailed• Style is either not defined or not followed• Functional elements are not defined	0
<ul style="list-style-type: none">• Project structure is laid out and mostly followed• Project contains Templates for most major sections• Functional structures are noted but not detailed	10

<ul style="list-style-type: none"> • Routes are mentioned but not explained • Style is detailed and mostly followed • Functional elements are noted but not clearly explained 	
<ul style="list-style-type: none"> • Project structure is clearly laid out and followed. • Template design is encapsulated and meaningful. • Functional structures (e.g. controllers) are correctly applied at the appropriate level. • Routing structure is clearly explained and justified. • Style is detailed (including any advanced style features) and followed in the SPA. • Functional elements of the SPA are clearly detailed, including technologies used (JavaScript \AngularJS \ Express) 	15

SPA Implementation (50 Marks)

The marks awarded for the SPA implementation are defined heavily by the technologies and implementations chosen. For maximum marks in this portion the student will need to extend the SPA in some manner, either through additional code, style or HTML functionality. Some examples of possible routes for extension are provided below

Description	Mark
Implement a version of the SPA that uses no backend functionality. All style and HTML elements are structured and linked correctly. Static content is used to replicate potential dynamic content in the News section.	20
Utilise AngularJS to create a basic SPA. HTML content is defined by Angular variables and scripts control SPA communication and functionality. Content can be added through the Create page and is displayed on the News page.	30
The SPA uses simple templating to create a page out of multiple partial views. Angular controllers are utilised to control each templates operation. Ng-Include is the primary method of constructing the templated document	40
ngRoute is employed to provide deep linking to the SPA. Routes and controllers are correctly established and utilised to allow for communication between different views	45

Use AJAX calls through the provided RESTful server structure to the SPA. Establish http calls to the necessary controllers to store data when a post is made, and to read data when a user returns to the site.	50
---	----

SPA Extension (30 marks)

30 marks of the Assignment are awarded for additional implementation. This should be a significant stylistic or programmatic addition to the SPA. These additions should be clearly noted and detailed in documentation. You may use multiple smaller additions (e.g. markdown, animated menus) or a large singular addition (login database functionality).

You may use this opportunity to focus on either design or code enhancements to provide a superior viewing experience, or additional back end functionality. Some examples that you might consider include:

- Route statefulness (remember the post content etc)
- Recent posts to nav / detail elements
- Post comment chains
- Flexbox and responsive elements
- Personalisation (individual style or alternative style options)
- Animations and collapsing menus
- Markup / Markdown on text content
- Proper database for posts
- Session control and user login (Express / DB modification)
- Image attachment to files (Express / DB modification)
- Sort by tags / author / filtering

Note that these are not the only possible options, they are simply a small list of possible choices. If you employ additional libraries (either Node.JS modules or other libraries like jQuery / Bootstrap) in your solution they should again be included in any documentation.

Description	Mark
Implement 1 simple extension to your SPA implementation. This includes things like mark up for text entry, simple comments on posts, more robust style and responsive interactions	15
Implement 2 simple, or one complex extension to your SPA implementation. Complex extensions include complete visual overhauls, robust responsive design elements, database or login integration.	30