LEGAL OBLIGATION	ACTIONS REQUIRED	SOURCE
E.g. Privacy		
Add rows as necessary		

Task 2. Prepare site environment: folder structure, naming convention

1.4 Create an appropriate folder structure. All folders must contain relevant files and each folder should not exceed 1 MB.

1.5 The index.html file must be placed in the root folder

1.6 Naming convention: All filenames must be descriptive of the content, lowercase, and no spaces are allowed in the names of files and objects as specified in the style guide.

Task 3. Responsive page layout design considerations

1.7 The page layout design must conform to these requirements:

a) A responsive five (5) page website (to suit multiple devices)

b) A hamburger menu that is responsive

c) Animated :hover state for menu items

d) Format and appearance as per style guide

e) Web Form page:

- A minimum of seven form fields
- Four different input types
- Attributes with input restrictions: pattern, maxlength, max, min and title for error message
- Every input field needs a label and error div
- Element(s) for errors in form input empty by default
- A ‘help’ function on mouse hover (link)
- Use of a scripting language to add interactivity is allowed but not mandatory

1.8 Create a wireframe or skeletal outline of the page layout, that is, a diagram or sketch that identifies each section of the page, with elements positioning and dimensions. Name all sections in the layout diagram. It is expected that the 5 pages will have a common basic layout but small differences are allowed for example, for the web form page. Create a **separate wireframe diagram** for any page that partially modifies the layout (*this includes the form page which should detail the form requirements given above*).

These diagrams will be used as the blueprint to create the HTML pages.

You need two (2) versions of the page layouts (wireframes) to suit the layout of at least two (2) devices, for example a PC/laptop screen and a mobile phone screen.

1.9 Access the style guide provided and **document** the typography, iconography and colour scheme for each elements/section outlined in the wireframe diagram. *You can use the style guide as an example of how to document this information.*

> **Note**
>
> This information will form the basis for the styling of your site so you should refer to this as you are developing your website

Task 4. Create HTML 5 page layout sections

1.10 HTML5 has been selected as the most appropriate markup language for the task. Create the pages as per your wireframe structures (*you should include images within your content as appropriate*) and making sure that you:

a) Use semantic elements only for the layout structure

b) Use ID and CLASS elements as required when no semantic element exists for the function

c) Create the menu with *HREF* elements that target the appropriate pages

d) Ensure that pages are correctly linked

e) Correct indentation of HTML code

PART 2 – Creating the external CSS file to format the site layout and appearance

2.1 Create a single external CSS file and save it in the CSS folder. This file will provide the common formatting features for the five (5) pages.

2.2 Create the necessary CSS rules to produce the required format and appearance:

a) Flex and/or Grid model and @media queries to achieve responsiveness

b) Layout should be in flow. No position: absolute *(position: fixed mobile view allowed)*

c) Layout style on user-defined classes

d) Common CSS for site pages – external CSS

e) One common menu using list items li elements *(included in every page via CSS or code)*

f) One common footer *(included in every page via CSS or code)*

g) Demonstrated use of:

- two (2) types of CSS combinator; and
- three (3) pseudo-classes

h) Text styles consistently applied for headings, paragraph, and figcaption as per style guide

i) Form CSS:

- Use grid layout and @media query to remove one of the input fields on mobile devices

- Use @media query rules to remove the form labels in mobile view and set the form width to 100%
- Use four (4) form-specific pseudo-classes to aide in user input functionality
- Use ::before or ::after pseudo-element to denote fields with required attribute
- Ensure the user is aware of erroneous input as they are inputting text into your form
- Use an absolutely positioned icon inside each relatively positioned text ::after
- Create a ‘help’ link that overlays an absolute or fixed position video element (A popup window) when the user hovers over the link. *Note this will be functional on desktop version only*.

Ensure that the layout is consistent with the wireframes prepared/refined in Task 3 -1.8.

PART 3 – Test, debug and validate web page

3.1 Test the webpage design and functionality to confirm that it has been developed according to the client’s requirements and the layout design. You should add the relevant client and design requirements (including your styling requirements from Task 1.9) to the ***UptownIT Functionality Test Report Template*** (*available for download in the Course Hub*) and complete this document.

Tip

For Task 3.1 you should review the client/site requirements:

design requirements, wireframe layouts, styling requirements (refer to your style guide), page layout etc

and add these to the test report and use these as the basis for your testing to ensure you have met the requirements.

3.2 Use browser developer tools to debug page design and functionality as required until it meets the client’s requirements. Check with your manager (teacher/assessor) to confirm requirements, as required. You should add the relevant functionality tests to the ***UptownIT Functionality Test Report Template*** and complete this document.

Tip

For Task 3.2 you should review the functional requirements for the site:

section links, page display, validation, browser/device and accessibility

and add these to the test report and use these as the basis for your testing.