**Create a web Page of your choice with the following usecases.**

**Use Case -1**

**Objective:** Build a website based on the following user stories:

**User Story #1:** My tribute page should have an element with a corresponding id="main", which contains all other elements.

**User Story #2:** I should see an element with a corresponding id="title", which contains a string (i.e. text) that describes the subject of the tribute page (e.g. "Dr. Norman Borlaug").

**User Story #3:** I should see a div element with a corresponding id="img-div".

**User Story #4:** Within the img-div element, I should see an img element with a corresponding id="image".

**User Story #5:** Within the img-div element, I should see an element with a corresponding id="img-caption" that contains textual content describing the image shown in img-div.

**User Story #6:** I should see an element with a corresponding id="tribute-info", which contains textual content describing the subject of the tribute page.

**User Story #7:** I should see an a element with a corresponding id="tribute-link", which links to an outside site that contains additional information about the subject of the tribute page. HINT: You must give your element an attribute of target and set it to \_blank in order for your link to open in a new tab (i.e. target="\_blank").

**User Story #8:** The img element should responsively resize, relative to the width of its parent element, without exceeding its original size.

**User Story #9:** The img element should be centered within its parent element.

Hint: <https://codepen.io/freeCodeCamp/full/zNqgVx>

## Use Case -2 : Build a Survey Form

**User Story #1:** I can see a title with id="title" in H1 sized text.

**User Story #2:** I can see a short explanation with id="description" in P sized text.

**User Story #3:** I can see a form with id="survey-form".

**User Story #4:** Inside the form element, I am required to enter my name in a field with id="name".

**User Story #5:** Inside the form element, I am required to enter an email in a field with id="email".

**User Story #6:** If I enter an email that is not formatted correctly, I will see an HTML5 validation error.

**User Story #7:** Inside the form, I can enter a number in a field with id="number".

**User Story #8:** If I enter non-numbers in the number input, I will see an HTML5 validation error.

**User Story #9:** If I enter numbers outside the range of the number input, which are defined by the min and max attributes, I will see an HTML5 validation error.

**User Story #10:** For the name, email, and number input fields inside the form I can see corresponding labels that describe the purpose of each field with the following ids: id="name-label", id="email-label", and id="number-label".

**User Story #11:** For the name, email, and number input fields, I can see placeholder text that gives me a description or instructions for each field.

**User Story #12:** Inside the form element, I can select an option from a dropdown that has a corresponding id="dropdown".

**User Story #13:** Inside the form element, I can select a field from one or more groups of radio buttons. Each group should be grouped using the name attribute.

**User Story #14:** Inside the form element, I can select several fields from a series of checkboxes, each of which must have a value attribute.

**User Story #15:** Inside the form element, I am presented with a textarea at the end for additional comments.

**User Story #16:** Inside the form element, I am presented with a button with id="submit" to submit all my inputs.

Hint: <https://codepen.io/freeCodeCamp/full/VPaoNP>.

**Use Case -3: Product Landing Page**

**User Story #1:** My product landing page should have a header element with a corresponding id="header".

**User Story #2:** I can see an image within the header element with a corresponding id="header-img". A company logo would make a good image here.

**User Story #3:** Within the #header element I can see a nav element with a corresponding id="nav-bar".

**User Story #4:** I can see at least three clickable elements inside the nav element, each with the class nav-link.

**User Story #5:** When I click a .nav-link button in the nav element, I am taken to the corresponding section of the landing page.

**User Story #6:** I can watch an embedded product video with id="video".

**User Story #7:** My landing page has a form element with a corresponding id="form".

**User Story #8:** Within the form, there is an input field with id="email" where I can enter an email address.

**User Story #9:** The #email input field should have placeholder text to let the user know what the field is for.

**User Story #10:** The #email input field uses HTML5 validation to confirm that the entered text is an email address.

**User Story #11:** Within the form, there is a submit input with a corresponding id="submit".

**User Story #12:** When I click the #submit element, the email is submitted to a static page (use this mock URL: <https://www.freecodecamp.com/email-submit>) that confirms the email address was entered and that it posted successfully.

**User Story #13:** The navbar should always be at the top of the viewport.

**User Story #14:** My product landing page should have at least one media query.

**User Story #15:** My product landing page should utilize CSS flexbox at least once.

Hint: <https://codepen.io/freeCodeCamp/full/RKRbwL>

**Use Case -4: Technical Documentation Page**

**User Story #1:** I can see a main element with a corresponding id="main-doc", which contains the page's main content (technical documentation).

**User Story #2:** Within the #main-doc element, I can see several section elements, each with a class of main-section. There should be a minimum of 5.

**User Story #3:** The first element within each .main-section should be a header element which contains text that describes the topic of that section.

**User Story #4:** Each section element with the class of main-section should also have an id that corresponds with the text of each header contained within it. Any spaces should be replaced with underscores (e.g. The section that contains the header "JavaScript and Java" should have a corresponding id="JavaScript\_and\_Java").

**User Story #5:** The .main-section elements should contain at least 10 p elements total (not each).

**User Story #6:** The .main-section elements should contain at least 5 code elements total (not each).

**User Story #7:** The .main-section elements should contain at least 5 li items total (not each).

**User Story #8:** I can see a nav element with a corresponding id="navbar".

**User Story #9:** The navbar element should contain one header element which contains text that describes the topic of the technical documentation.

**User Story #10:** Additionally, the navbar should contain link (a) elements with the class of nav-link. There should be one for every element with the class main-section.

**User Story #11:** The header element in the navbar must come before any link (a) elements in the navbar.

**User Story #12:** Each element with the class of nav-link should contain text that corresponds to the header text within each section (e.g. if you have a "Hello world" section/header, your navbar should have an element which contains the text "Hello world").

**User Story #13:** When I click on a navbar element, the page should navigate to the corresponding section of the main-doc element (e.g. If I click on a nav-link element that contains the text "Hello world", the page navigates to a section element that has that id and contains the corresponding header.

**User Story #14:** On regular sized devices (laptops, desktops), the element with id="navbar" should be shown on the left side of the screen and should always be visible to the user.

**User Story #15:** My Technical Documentation page should use at least one media query.

Hint:

## Use Case -5: Personal Portfolio Webpage

**User Story #1:** My portfolio should have a welcome section with an id of welcome-section.

**User Story #2:** The welcome section should have an h1 element that contains text.

**User Story #3:** My portfolio should have a projects section with an id of projects.

**User Story #4:** The projects section should contain at least one element with a class of project-tile to hold a project.

**User Story #5:** The projects section should contain at least one link to a project.

**User Story #6:** My portfolio should have a navbar with an id of navbar.

**User Story #7:** The navbar should contain at least one link that I can click on to navigate to different sections of the page.

**User Story #8:** My portfolio should have a link with an id of profile-link, which opens my GitHub or FCC profile in a new tab.

**User Story #9:** My portfolio should have at least one media query.

**User Story #10:** The height of the welcome section should be equal to the height of the viewport.

**User Story #11:** The navbar should always be at the top of the viewport.

Hint: <https://codepen.io/freeCodeCamp/full/zNBOYG>