WEBSITE EVALUATION REPORT

UX Development Project – Part II

Written by Lotachi Ezechukwu

Student ID: 21158920

Email: Lotachi.ezechukwu@mail.bcu.ac.uk

Couse: MSc User Experience Design

Course code: CMP7216

The Brief: Appraise & Evaluate 3 different applications / websites

WEBSITE 1: MOZILLA FIREFOX

URL: https://www.mozilla.org/en-US/firefox/new/

The purpose is to provide information and resources about the Firefox web browser, which is an open-source browser. Users can download the latest version of Firefox, learn about its features and capabilities, troubleshoot any issues they may be experiencing, and get support.

The website is built using HTML, CSS, and JavaScript. It does not utilize any external frameworks, however, it makes use of its own JavaScript libraries to manage its behavior based on certain conditions such as changes in device accessing the website, as demonstrated in the figure below.

```
<!--[if !IE]><!-->
  <script type="text/javascript" src="https://www.mozilla.org/media/js/lib.212d7e62f37e.js" charset="utf-8"></script>
  <script type="text/javascript" src="https://www.mozilla.org/media/js/ui.e5168a14d6d4.js" charset="utf-8"></script>
  <script type="text/javascript" src="https://www.mozilla.org/media/js/fxa.3465945dcb1c.js" charset="utf-8"></script>
  <script type="text/javascript" src="https://www.mozilla.org/media/js/data.309416ba6525.js" charset="utf-8"></script>
<!--<![endif]-->
```

Interactions:

1. **Navigation bar:** The download button on the website bounces when the page loads, to catch the visitors' attention and direct them to the main goal of the page.
   The bar also includes navigation links that lead to different sections of the website and other useful resources. When the cursor hovers over each link, it displays a drop-down menu.
2. **Download button.**
   The website has download buttons that allow users to download the Firefox browser for installation. This functionality was achieved by utilizing the HTML anchor tag, when clicked, the button sends the device information to the download server which then determines the appropriate download link for the user's device.

```
▼<div class="c-intro-download">
  ▼<div id="download-button-thanks" class="mzp-c-button-download-contai
   ner c-button-download-thanks">
     <!--[if !IE]><!-->
     <a href="/en-US/firefox/download/thanks/" class="download-link mzp
     -c-button mzp-t-product mzp-t-xl" data-direct-link="https://downlo
     ad.mozilla.org/?product=firefox-stub&os=win&lang=en-US&attribution
     _code=c291cmNlPShub3Qgc2V0KSZtZWRpdW09KGRpcmVjdCkmY2FtcGFpZ249KG5v
     dCBzZXQpJmNvbnR1bnQ9KG5vdCBzZXQpJmV4cGVyaW1lbnQ9KG5vdCBzZXQpJnZhcm
     lhdGlvbj0obm90IHN1dCkmdWE9Y2hyb211JnZpc2l0X2lkPTUwMTg0MjE0Ni4xNjcx
     Mjg4MTk1&attribution_sig=84b7443bce4dea71306bcab442d6ff8f360e25afd
     510e275a2ce832bb9f0325e" data-link-type="download" data-download-
     location="primary cta" data-download-os="Desktop" data-display-
     name="Windows 32-bit" data-download-version="win"> Download
     Firefox </a> == $0
```

3. **CSS Transitions:** As a user scrolls down the homepage, the website displays a sticky note that slides in on the bottom left corner of the screen. The sticky note contains a download button as well as a link to the "Firefox Privacy Notice." To achieve this interaction, the division tag was styled using CSS classes that defined the animation behavior.

```
.mzp-c-sticky-promo.mzp-a-slide-in {                    firefox_des…1bdba.css:1
  animation-name: mzp-a-slide-in-right;
}
.is-firefox .promo-firefox, .mzp-c-sticky-promo,        firefox_des…1bdba.css:1
.promo-products {
  display: none;
}
@media (min-width: 480px)
.mzp-c-sticky-promo {                                   firefox_des…1bdba.css:1
  margin: ▸ 16px auto;
}
.mzp-c-sticky-promo {                                   firefox_des…1bdba.css:1
  animation: ▸ .6s ease .3s both;
  background-color: #fff;
  border-radius: ▸ 4px;
  bottom: 0;
  box-shadow: 0 8px 12px 1px rgb(29 17 51 / 4%),
     0 3px 16px 2px rgb(9 32 77 / 12%),
     0 5px 10px -3px rgb(29 17 51 / 12%);
  box-sizing: border-box;
  left: auto;
  margin: ▸ 16px;
  opacity: 0;
  padding: ▸ 24px;
  position: fixed;
  right: 16px;
  width: 304px;
  z-index: 10;
}
```

Another animation on the webpage is the transformation of cards which zooms in when the page section is scrolled to focus. This is also done using the “animation” property in CSS.

```
@supports (animation-fill-mode:forwards)
.is-animated {                                    firefox_des…1bdba.css:1
   animation-duration: 1s;
   animation-fill-mode: forwards;
   animation-iteration-count: 1;
   animation-name: zoom;
}
```

WEBSITE 2: FIGMA

URL: https://www.figma.com/

The purpose is to provide information and resources for figma software, which is a cloud-based design tool that allows users to create, edit, and share interactive design files.

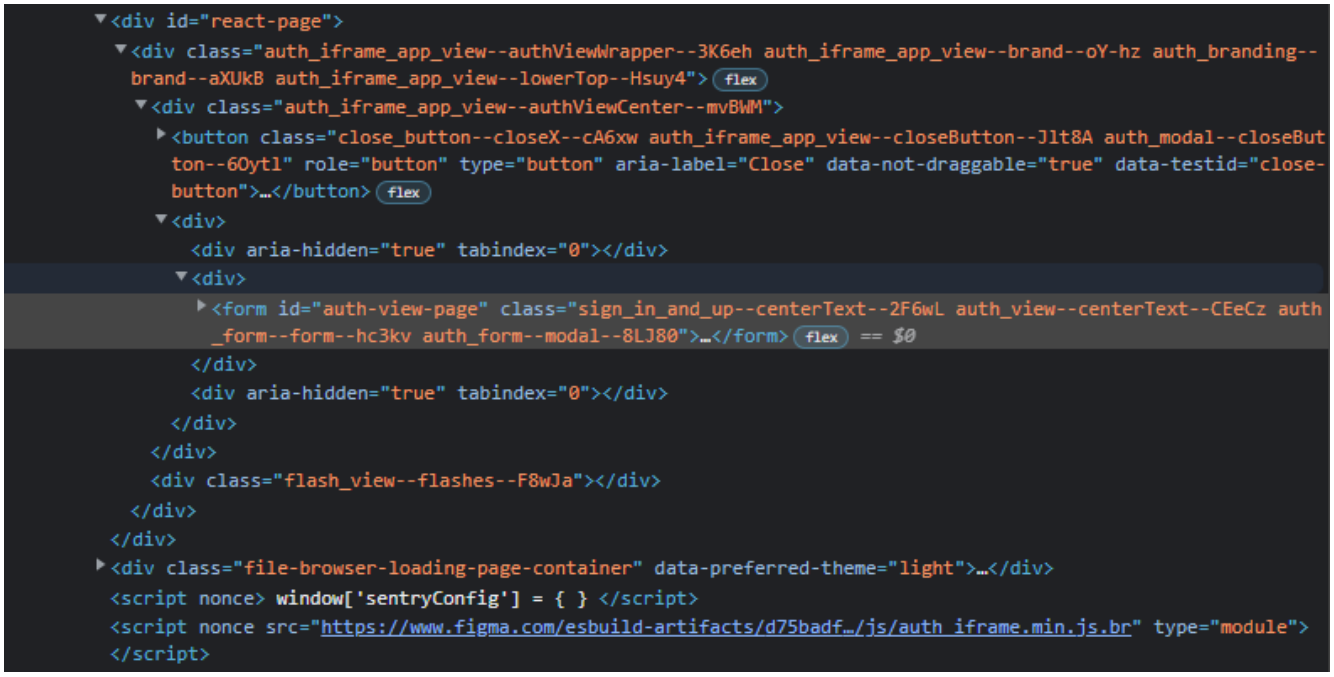
The website utilizes **React** JavaScript framework and **core.js** library to manage its appearance and behavior based on specific conditions.

Interactions:

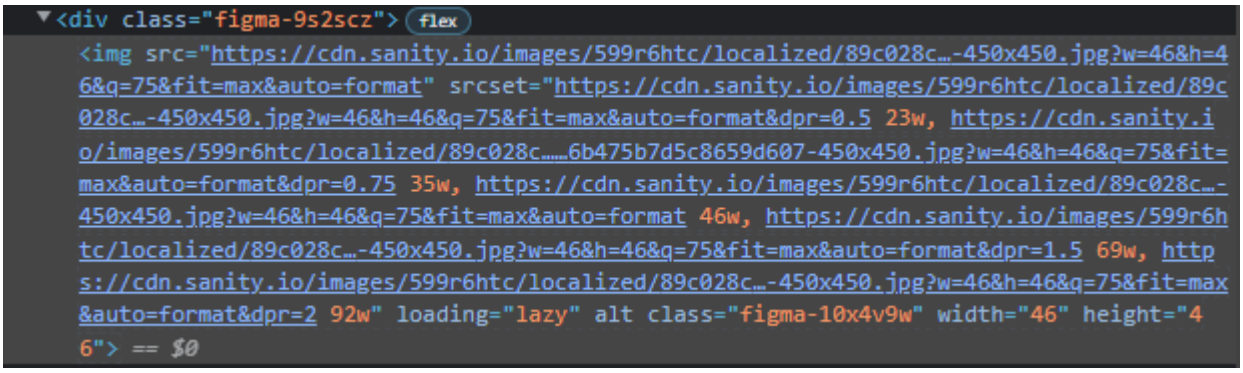
1. **Navigation:** The navigation menu, which includes links to other pages, adapts to different screen sizes. When the screen width is at least 1251 pixels, the menu items are displayed as anchor tags with a flex layout. When the screen width is no more than 1250 pixels, the menu items are condensed into a navigation icon. This was done with a combination of React state and props, CSS media queries, and JavaScript.

   The "**Get started**" link in the navigation bar uses core.js to detach itself from its parent element and maintain a fixed position when the screen scrolls to a specific point. This was done by using JavaScript to detect the scroll position of the page and to apply CSS styles to the anchor tag based on that scroll position.

2. **Account Management:** The website has interactions that allow users to create an account and log in. The form is built using react with validation directives.

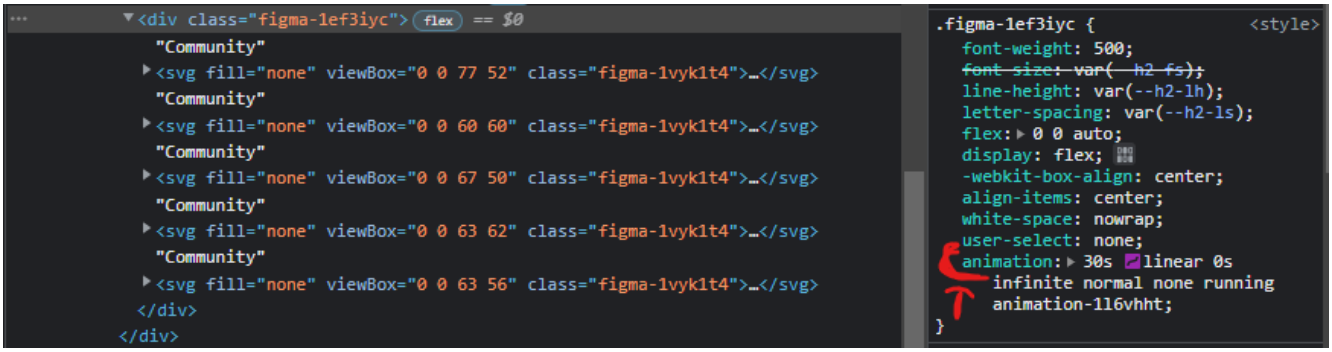


3. **Media Interaction:** The website utilizes the 'srcset' attribute for its images, enabling the browser to select the optimal image resolution based on device and network conditions. The lazy-loading attribute in the image tag allows the browser to only load images when they are in view, improving performance by avoiding loading images that are not currently visible. Query parameters are also applied to the images, such as **w, h, q,** and **fit,** which are used for modifying the image by dimensions, quality, or allotted space.



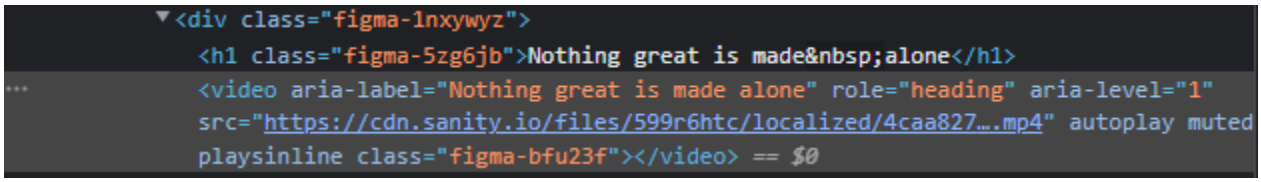
It's worth noting that these images are being served by a Content Delivery Network (CDN) called Sanity.io.

4. **CSS Animation:** The website employs CSS to create a slideshow effect for some of the SVG elements on the page.



Deficiencies:

1. **Media Interaction:** Although the website uses multimedia, such as videos or images, users can not interact with the videos either through play/pause functionality or loop functionality.



WEBSITE 3: PAYCIRCLE

URL: https://my.paycircle.co.uk/login

The purpose is to log into Paycircle, a cloud-based payroll platform for bureau payroll management.

The login page is built with AngularJS, Moment.js, core-js, jQuery, and uses the Google Font API.

Interactions:

1. The page uses the **form element** (form field) with directives for logic, validation and data posting via "post" method. The classes and attributes control behavior and appearance such as

- **form-style-a** is a custom class that is used to style the form in CSS.

- **ng-pristine, ng-scope, ng-invalid, ng-invalid-required** are AngularJS classes provided by the framework. They indicate the state of the form and its fields. *ng-pristine* means user has not modified the form, and *ng-invalid-required* means required fields are empty.
- **method="post"** sends form data to the server via HTTP POST.
- **name="loginForm"** sets the name of the form that can be referenced from the AngularJS controller.
- **ng-controller="LoginCtrl"** sets the name of the AngularJS controller that handles the form's logic.
- **ng-submit="login(credentials, loginForm)"**: invokes an AngularJS function "login" with two arguments "credentials" and "loginForm" when the form is submitted.
- **novalidate=""** disables built-in browser validation and allows AngularJS controller handle validation.

```
<form class="form-style-a ng-pristine ng-scope ng-invalid ng-invalid-required" method="post"
  name="loginForm" ng-controller="LoginCtrl" ng-submit="login(credentials, loginForm)" novalidate=""> event
  <form-header class="ng-isolate-scope"> ··· </form-header>
  <fieldset>
    <!--Username-->
    <div class="form-field"> ··· </div>
    <!--Password-->
    <div class="form-field"> ··· </div>
    <!--Login attempts limit reached or locked Account-->
    <!--ngIf: loginAttempts >= LOGIN_ATTEMPTS_LIMIT && !credentials.noCompaniesFound-->
    <!--Wrong Credentials-->
    <!--ngIf: loginAttempts < LOGIN_ATTEMPTS_LIMIT && credentials.loginFailed-->
    <!--Nothing to see, go home-->
    <!--ngIf: credentials.noCompaniesFound-->
    <!--Actions-->
    <div class="form-actions"> flex
      <button id="button-primary" class="button primary-action is-secondary is-fixed" data-page-state=""
      ng-class="{'success': state === 'saved'}" ng-disabled="disableButton || state === 'saving'"
      type="submit" button-theme="'is-secondary is-fixed'" disable-button="loginForm.$invalid" display-
      text="'Sign in'" state="loginForm.state" disabled="disabled"> ··· </button>
      <a class="s secondary" tabindex="-1" ui-sref="reset-password" href="/reset-password">
      Forgotten password?</a> event
    </div>
  </fieldset>
</form>
```

2. The HTML **button element** with the id "button-primary" is a submit button that when clicked will send the form data to the server for authentication. It also

has several AngularJS attributes within the element, including ng-class, ng-disabled, disable-button, display-text, and state.

- The ng-class attribute binds dynamic class based on form state.
- ng-disabled attribute disables the button when '**loginForm**' is invalid or saving.

3. The webpage also has a **link for forgot password**, that when clicked will redirect the user to a page to recover the password. '**ui-sref**' attribute with value "reset-password" is an AngularJS UI-Router state name. AngularJS UI-Router provides a way to navigate between different views and states in an AngularJS application.

4. The CSS media query above hides the element with class "right-pane" when screen width is less than or equal to 1024 pixels.