

CUSTOMER PORTAL
Custom HTML splashpage Guide

CUSTOM HTML

Custom HTML splash page templates are intended for users who have a good knowledge of HTML, CSS and JavaScript and want to create a splash page design that cannot be achieved using the standard splash page template designer.

This guide will introduce you to some of the tools available to help you build your custom HTML splash page template and provide you with a practical step by step example of how this can be achieved. In this guide you will create a simple 'offline' custom HTML design for a coffee shop named 'Chicago Cafe'.

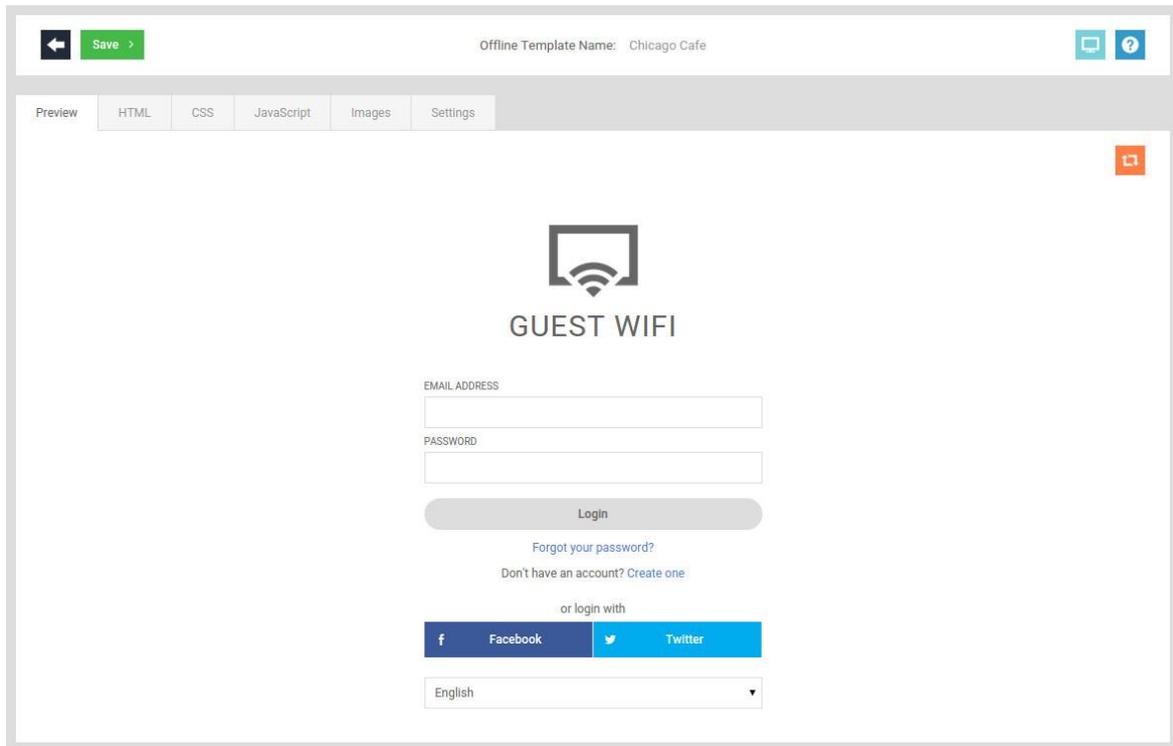


Create a new Custom HTML template

1. Whilst viewing your splash page template list, click on the 'Create new template' found at the top right of the page.
2. Enter the details for the new template in the window dialogue that appears and ensure that you choose the 'Custom HTML' option in the 'Type' dropdown.

3. Click on the '**Create New**' button to create the new template.

4. You will then be redirected to the custom HTML splash page template editor, where you can edit your new template.

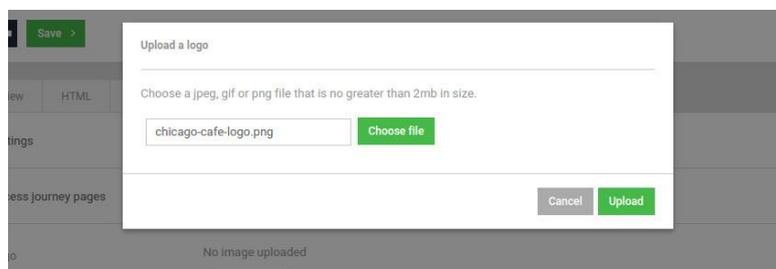


Your new custom HTML template

A responsive, HTML and CSS boilerplate template is applied to each new custom HTML template that you create. This is an example template that you can adapt to suit your needs or completely replace with your own custom HTML, CSS and JavaScript. As this boilerplate template already contains a responsive CSS design we are going to adapt it to create the 'Chicago Cafe' template.

Uploading a logo

1. Firstly, we will upload a logo image that we can use in the custom template and that will also be used for the interstitial pages of the WiFi access journey (e.g. the Facebook 'Like' page, additional profile page, SMS authentication page etc). We will also set a background colour that will be used to style the background of headers and buttons on these pages.
2. Click on the **'Settings'** tab in the editor and you should see that there is a section named **'Access journey pages'**. Click on the **'Upload logo'** button to select and upload your logo image file.



3. Once you have successfully uploaded your logo, click on the 'Background colour' colour picker drop down and choose a colour.

Access journey pages

Logo



Upload logo

Background colour

#000000

Editor settings

Code editor theme

Font size

A color picker dialog box with a gradient background from black to red, a vertical rainbow color bar on the right, and a small black square in the top right corner. Below the color bar is a text input field containing "#000000". At the bottom are two buttons: "Cancel" (red) and "Choose" (green).

4. Click on the 'Save' template dropdown and choose 'Save template' to save these changes.

Upload custom images

1. Next, we will upload the remaining images that will be used in our template. These images are used to set the background for the HTML <body> element and to create custom social media authentication buttons.
2. Click on the **'Images'** tab in the editor and click on the **'Upload files'** button and upload the background and social media image files.
3. Once they have been successfully uploaded the images will appear in the list and will be ready for use.

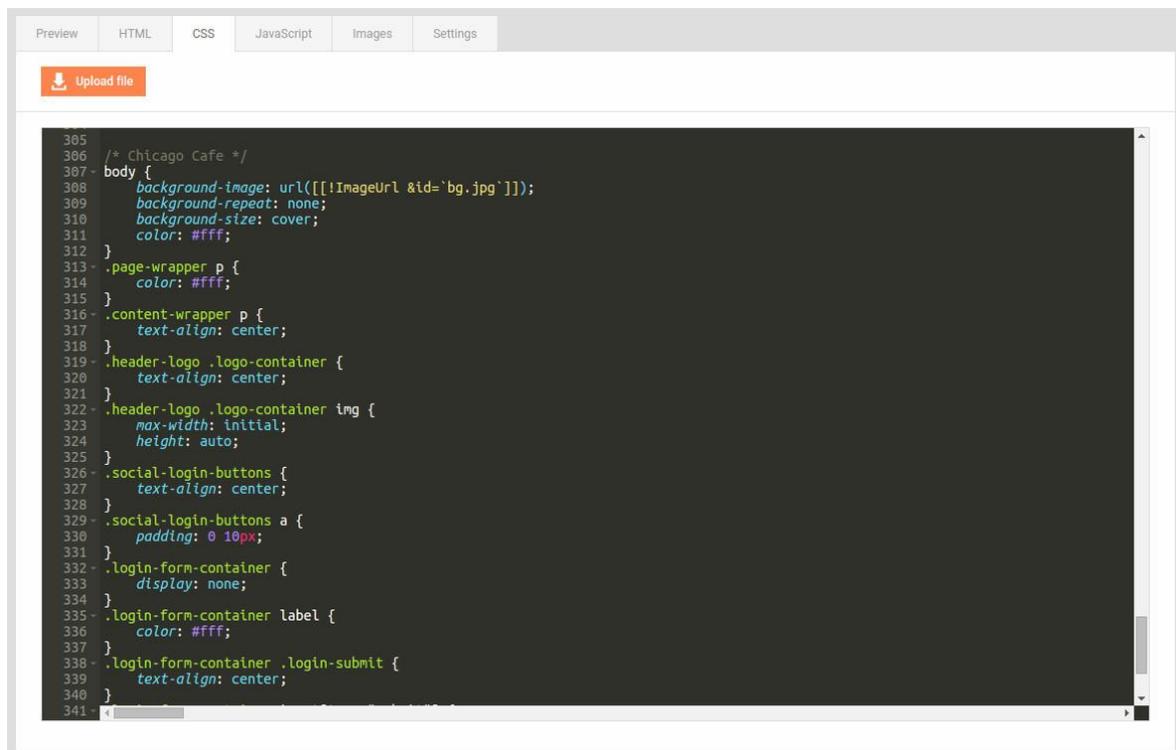
The screenshot shows the 'Images' tab of an editor interface. At the top, there is a 'Save' button and the text 'Offline Template Name: Chicago Cafe'. Below the tabs (Preview, HTML, CSS, JavaScript, Images, Settings), there is an 'Upload files' button. A table lists the uploaded images with their names, snippet codes, sizes, and dimensions. Each row has a three-dot menu icon on the right.

| Image | Snippet code | Size | Dimensions |
|-----------------------|---|---------|-----------------|
| twitter-btn.png | [[!ImageUrl &id="twitter-btn.png"]] | 2.4 kB | 46px x 46px |
| instagram-btn.png | [[!ImageUrl &id="instagram-btn.png"]] | 2.1 kB | 46px x 46px |
| facebook-btn.png | [[!ImageUrl &id="facebook-btn.png"]] | 1.9 kB | 46px x 46px |
| bg.jpg | [[!ImageUrl &id="bg.jpg"]] | 72.1 kB | 1920px x 1280px |
| chicago-cafe-logo.png | [[!ImageUrl &id="chicago-cafe-logo.png"]] | 3.6 kB | 316px x 82px |

Custom CSS

Custom HTML templates allow you to define your own CSS classes and selectors via the 'CSS' tab. You can manually enter styles in the editor within the tab or upload an entire CSS file that you may have developed offline.

1. Click on the 'CSS' tab in the editor
2. Click on the 'Upload file' button located in the upper left corner of the tab
3. Click on the 'Choose file' button and locate the 'styles.css' file
4. Click on the 'Upload' button to upload the file
5. Once the file has uploaded successfully you should see that the content of the CSS code editor now



6. You can update and add to your custom CSS at any time - click on the 'Preview' tab to preview the changes.
7. You may also notice that the above screenshot contains a short code named `[[!ImageUrl]]` where you would normally expect to find a relative or absolute url for the `<body>` background image. This is a snippet code that we have added to the CSS to embed a url for a 'captive portal' image, in this case the uploaded background image. We'll be looking at embedding snippet codes in the next step.

First HTML changes

1. The first changes that we will make to the boilerplate template are to add the logo, remove some of the boilerplate text that is provided in the template and set the document page title.
2. Click on the '**HTML**' tab to edit the source HTML of your template and scroll down to the first `<section>` tag in the HTML. You will see there is an `img` tag with an `'src'` attribute containing a `[[!GuestWiFiLogo]]` system snippet code.

```

18
19 - <section class="header-logo">
20 -   <div class="content-wrapper">
21 -     <div class="logo-container">
22 -       
23 -       <p>Guest WiFi</p>
24 -     </div>
25 -   </div>
26 - </section>
27

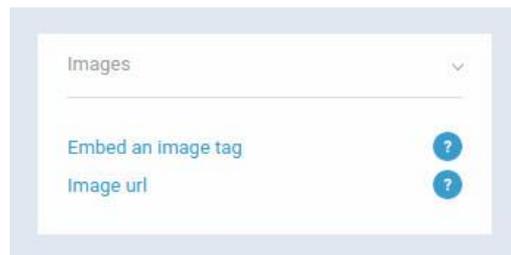
```

3. This snippet embeds a url to the 'Guest WiFi' logo used by the boilerplate template.

What are snippets?

Snippets are reusable pieces of short code that you can embed in your HTML and render HTML directly into your template output at run time. You can find the snippet library in the HTML tab which provides a variety of snippets to render forms and images. If you want to find out more about snippets before moving forward in this guide then there is a 'Help' button next to the 'Show snippets' option in the upper right of the HTML tab that will launch snippet help.

4. Let's replace the 'Guest WiFi' logo with your own logo you uploaded for 'Chicago Cafe' by using the `[[!ImageUrl]]` snippet.
5. Highlight the `[[!GuestWiFiLogo]]` snippet code text within the **'src'** attribute of the image tag (as seen in the image above)
6. Within the snippet library to the right of the HTML tab, click on the 'Images' section and then click on **'Image url'**.



7. In the modal dialogue, choose the image and click **'Embed'**
8. You should see that the 'src' attribute for the image is now a snippet code that references your uploaded logo image.

```

18
19 - <section class="header-logo">
20 -   <div class="content-wrapper">
21 -     <div class="logo-container">
22 -       
23 -       <p>Guest WiFi</p>
24 -     </div>
25 -   </div>
26 - </section>
27

```

9. You can add the `[[!ImageUrl]]` snippet to your HTML, CSS or JavaScript to embed a captive portal url to one of your template images. The **'id'** parameter is the **'Image'** name that is referenced in your uploaded 'Images' list.
10. Click on the **'Preview'** tab to see the change.

Remove the 'Guest WiFi' text

You'll see that the template currently says 'Guest WiFi' underneath the logo. We're going to remove this from our template simply by removing the relevant HTML tag in the HTML.

1. Within the **'HTML'** tab, scroll down to the next paragraph (`<p>`) tag after the logo `` tag.
2. Remove the entire line that contains `<p>Guest WiFi</p>`
3. Preview your changes in the **'Preview'** tab.

Change the document page title

Custom HTML templates allow you to set a custom page title for your document. Let's change this from 'Guest WiFi' to 'Chicago Cafe'.

1. Within the '**HTML**' tab, locate the <title> tag within the <head> tag.
2. Update the text to 'Chicago Cafe WiFi'.

```
7 </head>
8 <meta charset="utf-8">
9 <meta http-equiv="x-ua-compatible" content="ie=edge">
10 <title>Chicago Cafe WiFi</title>
11 <meta name="description" content="">
12 <meta name="viewport" content="width=device-width, initial-scale=1">
13 <link href="http://fonts.googleapis.com/css?family=Roboto:300,400,500,700" rel="stylesheet" type="text/css">
14 </head>
```

Add custom social media authentication buttons

Next, we are going to add some custom social media authentication buttons for our WiFi users to authenticate with. By default, the boilerplate design includes buttons for Facebook and Twitter so we're going to customise these and add an extra button for Instagram authentication.

1. Within the 'HTML' tab, locate the second <section> which has a class of 'login-form'.
2. Replace the entire <section> element with the following HTML:

```
<section class="login-form">
  <div class="content-wrapper">
    <p>Login to our WiFi with social media</p>
    <div class="social-login-buttons cf">
      <a href="#"></a>
      <a href="#"></a>
      <a href="#"></a>
    </div>
  </div>
</section>
```

3. If you followed the '[First HTML changes](#)' step then you should recognise the [[!ImageUrl]] snippet used to set the 'src' attribute for each of the tags in our new buttons. The only difference here is that we have added the snippet codes manually rather than using the snippets library to embed the image url codes.
4. Click on the '**Preview**' tab to preview your changes and you should see that the template now displays the 3 images as anchor links.



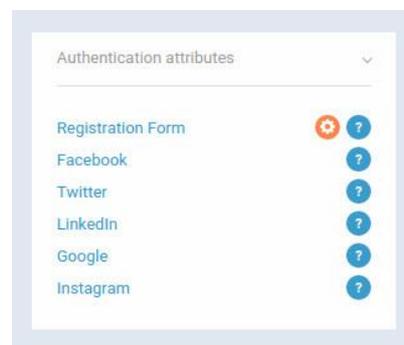
Use snippets to activate the social authentication buttons

Our new buttons are currently inactive so the next step is to ensure that they trigger the social media authentication process for the relevant network when our WiFi users click on them. This can be achieved by adding '**Authentication attributes**' to the anchor elements which will add a data attribute.

1. Navigate to the '**HTML**' tab and repeat the following step for each of the three social media buttons.
2. Place the cursor after the '**href**' attribute of the <a> tag and add a single '**space**'

```
<div class="social-login-buttons cf">
  <a href="#" ></a>
  <a href="#"></a>
  <a href="#"></a>
</div>
```

3. Locate and expand the '**Authentication attributes**' section in the snippets library found to the right of the HTML code editor



4. Click on the relevant '**Authentication attribute**' network link (e.g. '**Facebook**') to embed the snippet code at the cursor position

```
<div class="social-login-buttons cf">
  <a href="#" [[!FacebookAuthAttr]]></a>
  <a href="#"></a>
  <a href="#"></a>
</div>
```

5. Your HTML mark-up should now look like this after repeating this process for each social network button:

```
<section class="login-form">
  <div class="content-wrapper">
    <p>Login to our WiFi with social media</p>
    <div class="social-login-buttons cf">
      <a href="#" [[!FacebookAuthAttr]]></a>
      <a href="#" [[!TwitterAuthAttr]]></a>
      <a href="#" [[!InstagramAuthAttr]]></a>
    </div>
  </div>
</section>
```

6. If you now '**Preview**' your changes you should be able to click on each button and initiate the social media authentication process for each network.

'Authentication attribute' snippets

'**Authentication attribute**' snippets embed a data attribute on an element which the boilerplate custom template JavaScript then uses to initiate the WiFi authentication process. You can embed these snippets on any element within your custom HTML to initiate the WiFi authentication process for the relevant authentication method.

Add a registration form link and an embedded login form

We'll now add a link for our WiFi users to authenticate via the registration form and an embedded login form for registered users to log back in on return visits to our venue.

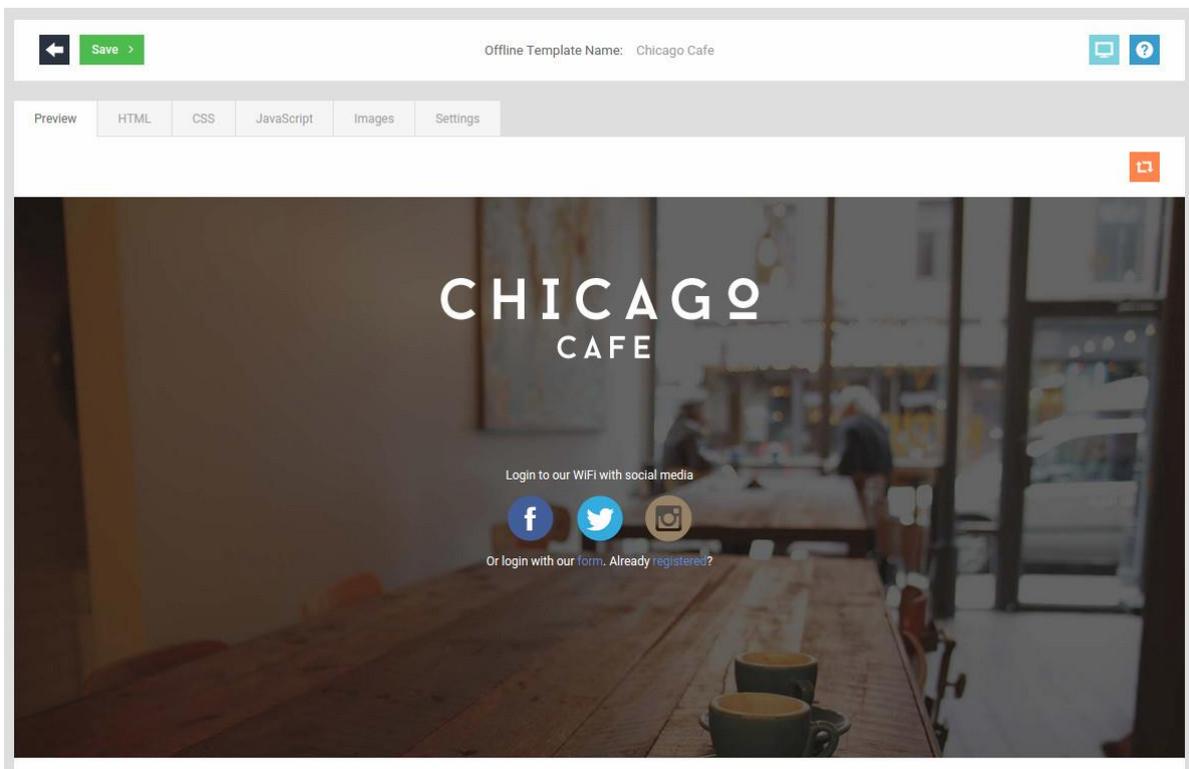
1. We will achieve this by adding a link to toggle the display of the login form, a snippet code of `[[!LoginForm]]` to embed the form and a link to initiate the authentication process via the registration form using the `[[!RegistrationFormAuthAttr]]` authentication attribute snippet.
2. Underneath the `<div>` containing the social media buttons add the following HTML markup:

```
<p>Or login with our <a href="#" [[!RegistrationFormAuthAttr]]>form</a>. Already <a id="btn-login" href="#">registered?</a></p>
<div class="login-form-container cf">
  [[!LoginForm]]
</div>
```

3. The `<section>` should now look like this:

```
<section class="login-form">
  <div class="content-wrapper">
    <p>Login to our WiFi with social media</p>
    <div class="social-login-buttons cf">
      <a href="#" [[!FacebookAuthAttr]]>form</a>. Already <a id="btn-login" href="#">registered?</a></p>
    <div class="login-form-container cf">
      [[!LoginForm]]
    </div>
  </div>
</section>
```

4. If you **'Preview'** your template you should now see the additional form links underneath the social media buttons.



5. You'll notice that the registration form will display in an overlay when you click on the **'form'** text link. The JavaScript required to submit this form is already available in the template as part of the core boilerplate JavaScript and is embedded at run time.
6. Nothing happens when you click on the **'Already registered?'** link. In the next section we will add some custom JavaScript to toggle the display of our login form when this link is clicked by the user.

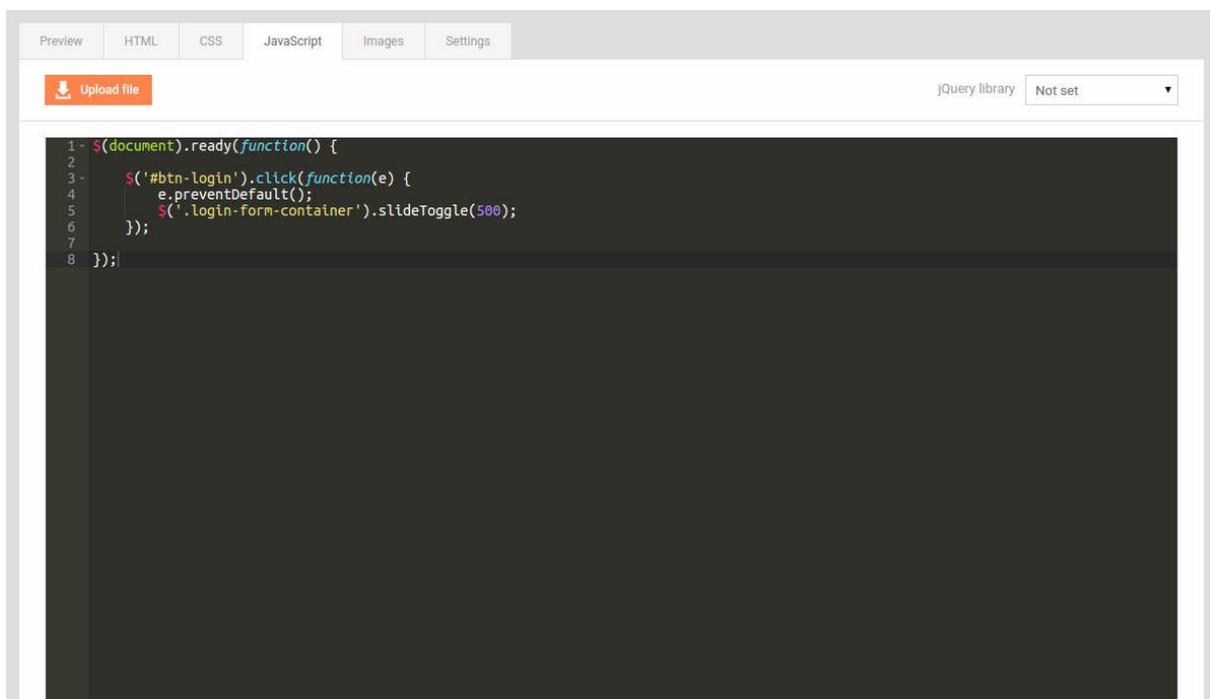
Add custom JavaScript

Your template will have JavaScript injected at run time to provide base functionality for the majority of snippet codes that are found in the snippet library (e.g. to submit boilerplate forms and handle errors and to initiate WiFi authentication for different authentication methods). However there maybe cases where you may want to add your own custom JavaScript code to add additional functionality or animation to your template.

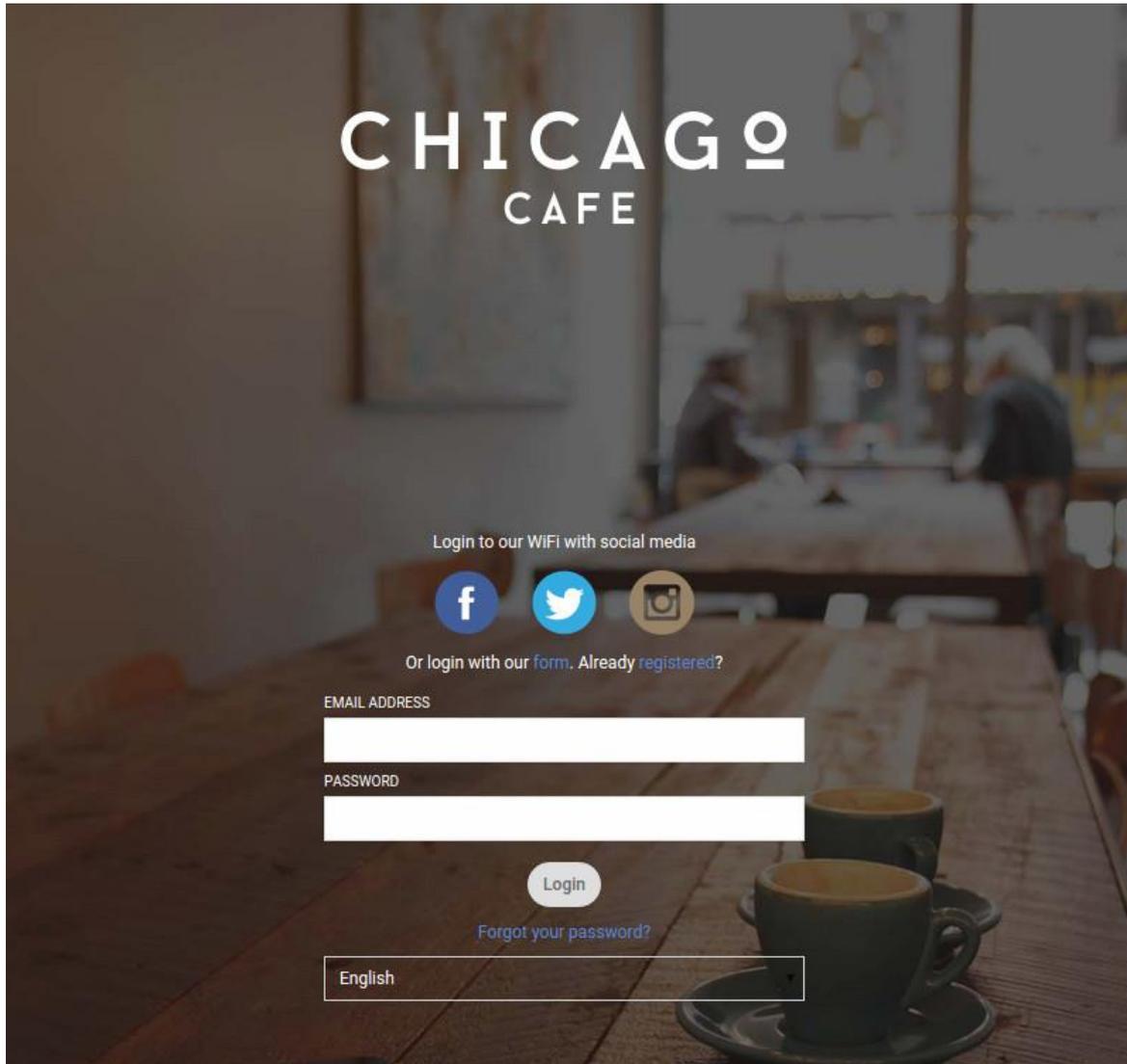
1. We're now going to add some custom JavaScript to toggle the display of our login form when our **'Already registered?'** link is clicked by the user. This JavaScript will toggle the visibility of the parent <div> element that contains the form.
2. Click on the **'JavaScript'** tab to edit the JavaScript for the template
3. Add the following jQuery code within the body of the document 'ready' event:

```
$('#btn-login').click(function(e) {  
    e.preventDefault();  
    $('.login-form-container').slideToggle(500);  
});
```

4. Your custom JavaScript code should now look similar to this:



5. If you now **'Preview'** your template and click on the 'Already registered?' link you should see that the login form display now toggles.



JavaScript

Your custom HTML template includes the jQuery JavaScript library by default so you are free to use **'vanilla'** JavaScript or jQuery to add additional functionality to your design.

End of the guide

If you have followed all the steps in this guide then you should now have a functional custom HTML splash page template for 'Chicago Cafe' that you could apply to an Access Journey. Hopefully this guide has introduced you to some of the tools available to help you create your own custom HTML design and provided you with an understanding of how to use them.

Common questions and answers

Q- Can I make my own custom HTML buttons or elements for social media and registration form authentication (e.g. Facebook, LinkedIn, WiFi registration form etc)?

A - Yes, you can make your own custom HTML to provide links for your users to authenticate via all authentication methods. You can also use custom CSS to style these elements via the 'CSS' code editor and upload custom images. See step 7 of this guide which provides an example of this.

Q - Can I embed a registration form in my 'offline' template?

A - Yes, you can use the '**Authentication form**' snippets to embed a registration, login or password reset form in your '**offline**' template and use custom CSS to style them. Boilerplate JavaScript is added to the template at run time to provide base functionality for these snippets.

Q - Can I add a menu or special offers to my custom splash page template?

A - Yes, you can create your own HTML mark up and add your own CSS to create a menu, advertise special offers or for any other purpose.

Q- Can I link to resources that I currently host on my website (e.g. images, video, pdf documents etc)?

A - Yes and No. It is not currently possible to link to images or other resources within '**offline**' templates due to the WiFi user being within the '**captive portal**' at this stage of the WiFi authentication process. With '**offline**' templates it is possible to upload image files via the '**Images**' tab and then use the '**Image**' snippets to embed links to the uploaded images. With '**online**' templates it is possible to embed links to resources hosted on external domains such as your own website or other content delivery networks. At this point the user is '**online**' and outside of the '**captive portal**' so it is possible to deliver these resources to the user's browser.

Q - Can I make an animated, image slide show in my template?

A - Yes. You can create your own image slide show for a template by uploading your images, using the '**ImageUrl**' snippet to embed links to the uploaded images and then creating your own custom CSS and JavaScript to generate the slideshow.

Q - The standard splash page designer provides separate previews for 'mobile' and 'desktop' devices. Do custom HTML splash pages support each type of device?

A - Yes, but it is up to the template designer to implement a responsive design for both types of device. The boilerplate custom HTML features a responsive CSS design for both 'mobile' and 'desktop' devices that can be adapted to suit individual needs.

Q - When creating a language version of an Access Journey that uses a custom HTML splash page template, does the custom HTML splash page template get machine translated into the new language?

A - No. Custom HTML splash pages are not currently machine translated when a language version is created for an Access Journey.