TEMPO TIME CREDITS COOKIES POLICY

1. OUR USE OF COOKIES

1. Welcome to Tempo. This website, located at https://www.wearetempo.org (Site) is owned and operated by Tempo (we, us, our). Our Site may use cookies to distinguish you from other users. This helps us to provide you with a good experience when you browse the Site and also allows us to improve it. In this Cookies Policy you will find information on what cookies may be set when you visit any of our Site and the purposes for which we use those cookies.
For further detailed information about this Site, please see our Terms of Use and our Privacy Policy.

2. YOUR CONSENT TO USE COOKIES

- 1. Certain cookies we use are strictly necessary for the Site to work. For example, when you visit our Site for the first time, we present you with a pop-up message notifying you about our use of cookies. We had to use a cookie to remember that we have presented this notice to you (you can see details of this cookie in the Cookie Details section below).
- 2. In relation to all other types of cookies we use, by clicking the relevant button on the Cookie Banner or by continuing to use the Site, you are agreeing to our use of cookies in the manner described in this Cookies Policy. If you continue to use the Site, more cookies and other tracking technologies will be placed on your computer as described in this Cookies Policy in order to enhance your user experience whilst on the Site. Please see further section 4 below regarding the ways you can control or delete the cookies we use.

3. WHAT ARE COOKIES?

- 1. A cookie is a small data file that is placed on your browser or the hardware of your computer or other device to allow a website to recognise you as a user when you return to the website, either for the duration of your visit (using a 'session cookie') or for repeat visits (a 'persistent cookie'). Other similar files work in the same way and we use the word 'cookie' in this Cookies Policy to refer to all files that collect information in this way.
- 2. Cookies are an extremely common technology for remembering certain information about a visitor to a website. The vast majority of websites currently make use of cookies and they are commonly used for a wide range of tasks. We use cookies for the following purposes:
- Strictly necessary cookies. These are cookies that are required for the operation of our Site. They include, for example, cookies that enable you to log into secure areas of our Site.
- Analytical cookies. They allow us to recognise and count the number of visitors and
 to see how visitors move around our Site when they are using it. This helps us to
 improve the way our Site works, for example, by ensuring that users are finding what
 they are looking for easily.
- 3. This Cookies Policy gives you information on the specific cookies used on this Site.

4. HOW TO CONTROL AND DELETE COOKIES

- If you wish to restrict or block the cookies which are set by the Site (or, indeed, on any other website) you can do this through your browser settings. The 'Help' function within your browser should tell you how.
- Alternatively, you may wish to visit http://www.allaboutcookies.org/ which contains comprehensive information on how to do this on a wide variety of browsers.
 You will also find details on how to delete cookies from your computer, as well as

- more general information about cookies. Please note that, as these websites are not owned or operated by us, we are not responsible for any of the content on them.
- Please be aware that restricting cookies may mean that you will not be able to take full advantage of all the features or services available on the Site.

5. MAIN COOKIES USED BY US ON OUR SITE

 A list of the main cookies used by us on this Site together with details about what each is used for can be found in the Cookie Details section below.

6. THIRD PARTY COOKIES

- This Site uses third party cookies. More specifically, we use third party Analytics
 cookies to help us collect information that allows us to analyse web traffic to our Site.
 The information collected via the cookies, which is anonymous, is sent to a third
 party that operates analytics tools which we use to analyse the information collected
 and improve our Site.
- A list of the third party cookies used in connection with the Site can be found under the relevant section in the Cookie Details section below.

7. EMAIL TRACKING

- 1. Some of the emails we send to you may contain a 'web beacon pixel' (clear GIFs) or tracked links which allows us to identify when you have opened the email and to verify which links contained in the email you have accessed. We use this information to determine which parts of our emails are of most interest to you.
- 2. You can delete the pixel by deleting the email. If you do not wish to download the pixel to your computer or other device, you can ensure this by selecting to receive emails from us in plain text rather than HTML, or not opening pictures in your email.
- Alternatively you can unsubscribe from our mailing list by contacting us at hello@wearetempo.org

8. CHANGES

1. We may make changes to this Cookies Policy at any time by sending you an email with the modified terms or, where appropriate, by posting a copy of them on the Site. Any changes will take effect 7 days after the date of our email or the date on which we post the modified terms on the Site, whichever is the earlier.

9. CONTACT US

 If you have any questions or concerns about cookies or your privacy when using this Site, please contact us as follows:

If you wish to write to us, please write to:

Tempo

Unit 2

58-62 Cowbridge Road West Cardiff

CF5 5BS

1. Our email address for data protection queries is hello@wearetempo.org alternatively you can use via the Site.

DATE PUBLISHED: 22.06.2020

Cookie Details

Cookies used on our Site

Cookie	Name	Type and Expiry	Category and Purpose
JSON Web Token	JWT	First party cookie Authentication cookie	Strictly necessary : This cookie is essential as it validates Site users' login credentials.
Banner cookie	JWT	First party cookie Persistent cookie: 6 months	Strictly necessary : This cookie is essential as we use it to remember whether you have acknowledged the cookie notice we present to you.

Cookie Details Third Party Cookies used on our Site

Cookie	Name	Category and Purpose	Further
			information
Google Analytics	_utma	First party cookie Session cookie	Analytical: These cookies enable
	_utmb		the Google

	_utmc		Analytics service
	_utmz		we use to function. Google Analytics helps us take and analyse visitor information such as browser usage and new visitor number, which assists us to improve our Site and your user experience.
Hotjar	_hjClosedSurveyInvites	Hotjar cookie that is set once a visitor interacts with a Survey invitation modal pop-up. It is used to ensure that the same invite does not reappear if it has already been shown.	365 days
	_hjDonePolls	Hotjar cookie that is set once a visitor completes a poll using the Feedback Poll widget. It is used to ensure that the same poll does not reappear if it has already been filled in.	365 days
	_hjMinimizedPolls	Hotjar cookie that is set once a visitor minimizes a Feedback Poll widget. It is used to ensure that the widget stays minimized when the visitor navigates through your site.	365 days
	_hjIncludedInSample	Hotjar cookie that is set to let Hotjar know whether that visitor is included in the sample which is used to generate Funnels.	365 days
	_hjShownFeedbackMessage	Hotjar cookie that is set when a visitor minimizes or completes Incoming Feedback. This is done so that the Incoming Feedback will load as minimized immediately if the visitor navigates to another page where it is set to show.	365 days

when the customer first lands on a page with the Hotjar script. It is used to persist the Hotjar User ID, unique to that site on the browser. This ensures that behavior in subsequent visits to the same site will be attributed to the same user ID. _hjRecordingLastActivity This should be found in Session storage (as opposed to cookies). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it falls. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updatedhjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user into not in the sample. These attributes will only be saved if the	T	T	T
lands on a page with the Hotjar script. It is used to persist the Hotjar User ID, unique to that site on the browser. This ensures that behavior in subsequent visits to the same site will be attributed to the same user ID. _hjRecordingLastActivity _h	_hjid	Hotjar cookie that is set	365 days
Hetjar script. It is used to persist the Hotjar User ID, unique to that site on the browser. This ensures that behavior in subsequent visits to the same stee will be attributed to the same user ID. _hjRecordingLastActivity _hjRecordingLastActivity This should be found in Session storage (as opposed to cookles). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookle path we should use, instead of the page hostname. This is done so that cookles can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookle for different URL substring alternatives until it falls. After this check, the cookle is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updatedhjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user into not in the sample. These attributes will only be saved if the user into not in the sample. These attributes will only be saved if the user interacts with a		when the customer first	
persist the Hotjar User ID, unique to that site on the browser. This ensures that behavior in subsequent visits to the same site will be attributed to the same user ID. _hjRecordingLastActivity This should be found in Session storage (as opposed to cookies). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		lands on a page with the	
iD, unique to that site on the browser. This ensures that behavior in subsequent visits to the same user iD. _hjRecordingLastActivity This should be found in Session storage (as opposed to cookles). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookle path we should use, instead of the page hostname. This is done so that cookles can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookle for different URL substring alternatives until it fails. After this check, the cookle is removed. _hjUserAttributesHash User Attributes sent through the Hotjar identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookle stores User Attributes will only be saved if the user interacts with a		Hotjar script. It is used to	
the browser. This ensures that behavior in subsequent visits to the same site will be attributed to the same user ID. _hjRecordingLastActivity This should be found in Session storage (as opposed to cookies). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URI. Substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		persist the Hotjar User	
the browser. This ensures that behavior in subsequent visits to the same site will be attributed to the same user ID. _hjRecordingLastActivity This should be found in Session storage (as opposed to cookies). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URI. Substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		ID, unique to that site on	
ensures that behavior in subsequent visits to the same size will be attributed to the same user ID. _hjRecordingLastActivity This should be found in Session storage (as opposed to cookles). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookle path we should use, instead of the page hostname. This is done so that cookles can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookle for different URL substring alternatives until it fails. After this check, the cookle is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookle stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		I	
subsequent visits to the same site will be attributed to the same user ID. _hjRecordingLastActivity This should be found in Session storage (as opposed to cookies). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to other thehjTLDTest cookie for different URL substring alternatives until it falls. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user into in the sample. These attributes will only be saved if the user interacts with a			
same site will be attributed to the same user ID. _hjRecordingLastActivity This should be found in Session storage (as opposed to cookies). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it falls. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
attributed to the same user ID. _hjRecordingLastActivity This should be found in Session storage (as opposed to cookies). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it falls. After this check, the cookie is removed. _hjUserAttributesHash Light This cookie sont through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		•	
user ID. This should be found in Session storage (as opposed to cookies). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store thehjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash _hjUserAttributesHash _hjCachedUserAttributes _hjCachedUserAttributes This cookie stores User Attributes will only be saved if the user interacts with a			
hjRecordingLastActivity			
Session storage (as opposed to cookies). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API are cashed for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a	hiRecordingLastActivity		Session
opposed to cookies). This gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to ostore the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a	_njnecording_astActivity		30331011
gets updated when a visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
visitor recording starts and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
and when data is sent through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookle path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
through the WebSocket (the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookle path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it falls. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
(the visitor performs an action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
action that Hotjar records). _hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		_	
hjTLDTest			
_hjTLDTest When the Hotjar script executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		action that Hotjar	
executes we try to determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store thehjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		records).	
determine the most generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store thehjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a	_hjTLDTest	When the Hotjar script	Session
generic cookie path we should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store thehjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		executes we try to	
should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store thehjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		determine the most	
should use, instead of the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store thehjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		generic cookie path we	
the page hostname. This is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
is done so that cookies can be shared across subdomains (where applicable). To determine this, we try to store thehjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
can be shared across subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
subdomains (where applicable). To determine this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
applicable). To determine this, we try to store the hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash _hjUserAttributes Hash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
this, we try to store the _hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
hjTLDTest cookie for different URL substring alternatives until it fails. After this check, the cookie is removed. hjUserAttributesHash hjUserAttributesHash ldentify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. hjCachedUserAttributes hjCac			
different URL substring alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		-	
alternatives until it fails. After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
After this check, the cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		_	
cookie is removed. _hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
_hjUserAttributesHash User Attributes sent through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
through the Hotjar Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		cookie is removed.	
Identify API are cached for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a	_hjUserAttributesHash	User Attributes sent	Session
for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		through the Hotjar	
for the duration of the session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		Identify API are cached	
session in order to know when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
when an attribute has changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
changed and needs to be updated. _hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
_hjCachedUserAttributes			
_hjCachedUserAttributes This cookie stores User Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a		_	
Attributes which are sent through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a	hiCachedUserAttributes		Session
through the Hotjar Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a	joucheaosciAttibutes		30331011
Identify API, whenever the user is not in the sample. These attributes will only be saved if the user interacts with a			
the user is not in the sample. These attributes will only be saved if the user interacts with a			
sample. These attributes will only be saved if the user interacts with a		_	
will only be saved if the user interacts with a			
user interacts with a		I	
		-	
Γ Γ			
Hotjar Feedback tool.		Hotjar Feedback tool.	

_hjLocalStorageTest	This cookie is used to Under 100ms
	check if the Hotjar
	Tracking Script can use
	local storage. If it can, a
	value of 1 is set in this
	cookie. The data stored
	in_hjLocalStorageTest
	has no expiration time,
	but it is deleted almost
	immediately after it is
	created.
_hjIncludedInSample	Session cookie which is Session
	destroyed when user
	leaves the Hotjar site.