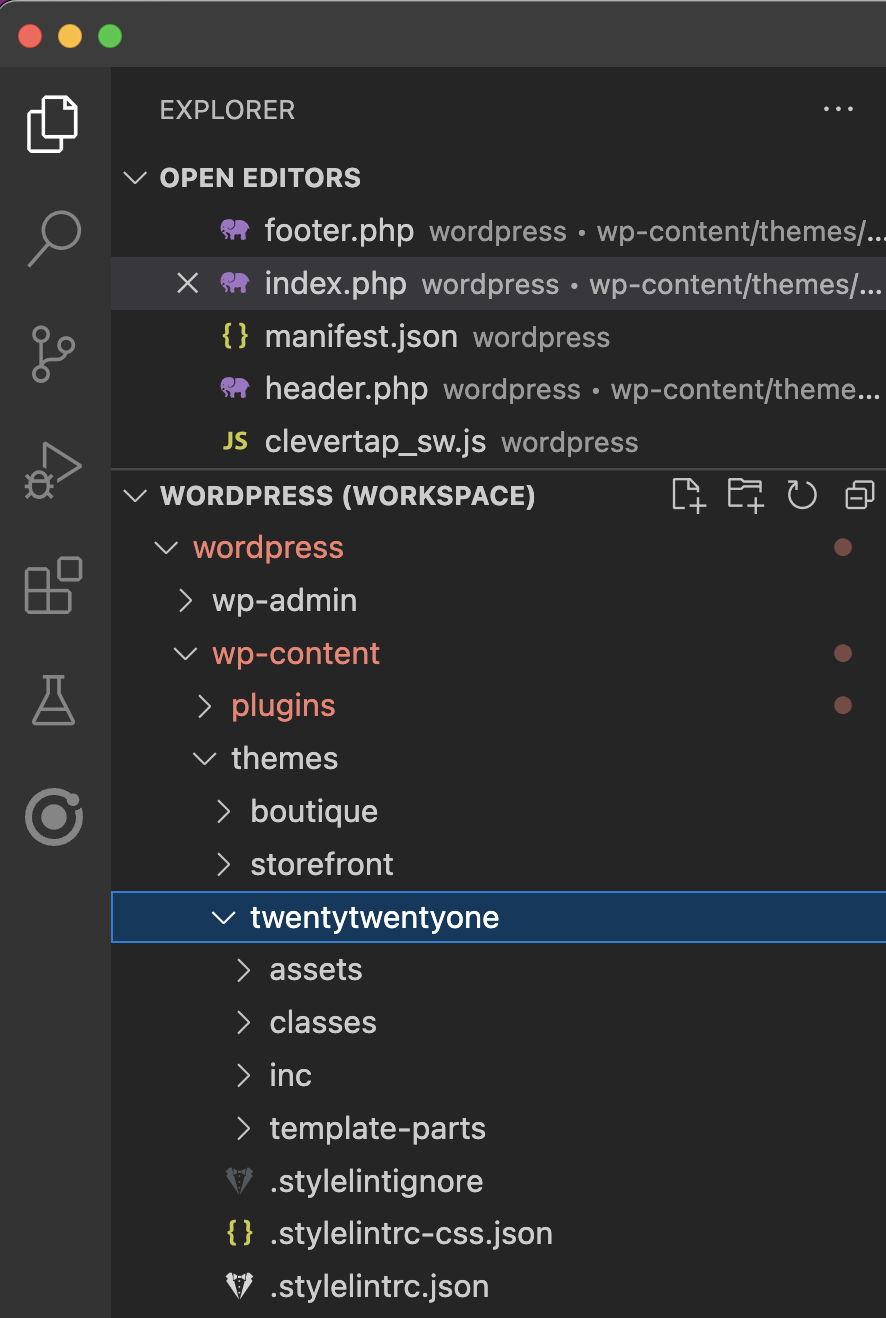
<Things can be implemented by referring to doc>

* SDK

First, you need to install a theme and if you are going to use Woocommerce you need the plugin

Open the project in Visual Studio code or any other developer platform.

Open the theme folder as the image



Now go to the header.php file and add the latest version of the Web SDK

<script type="text/javascript">

var clevertap = {event:[], profile:[], account:[], region: 'YOUR-REGION EX -> US1', onUserLogin:[], notifications:[], privacy:[]};

// replace with the CLEVERTAP\_ACCOUNT\_ID with the actual ACCOUNT ID value from your Dashboard -> Settings page

clevertap.account.push({"id": "YOUR-CLEVER-ACCOUNT-ID"});//

clevertap.privacy.push({optOut: false}); //set the flag to true, if the user of the device opts out of sharing their data

clevertap.privacy.push({useIP: true}); //set the flag to true, if the user agrees to share their IP data

(function () {

var wzrk = document.createElement('script');

wzrk.type = 'text/javascript';

wzrk.async = true;

wzrk.src = ('https:' == document.location.protocol ? 'https://d2r1yp2w7bby2u.cloudfront.net' : 'http://static.clevertap.com') + '/js/clevertap.min.js';

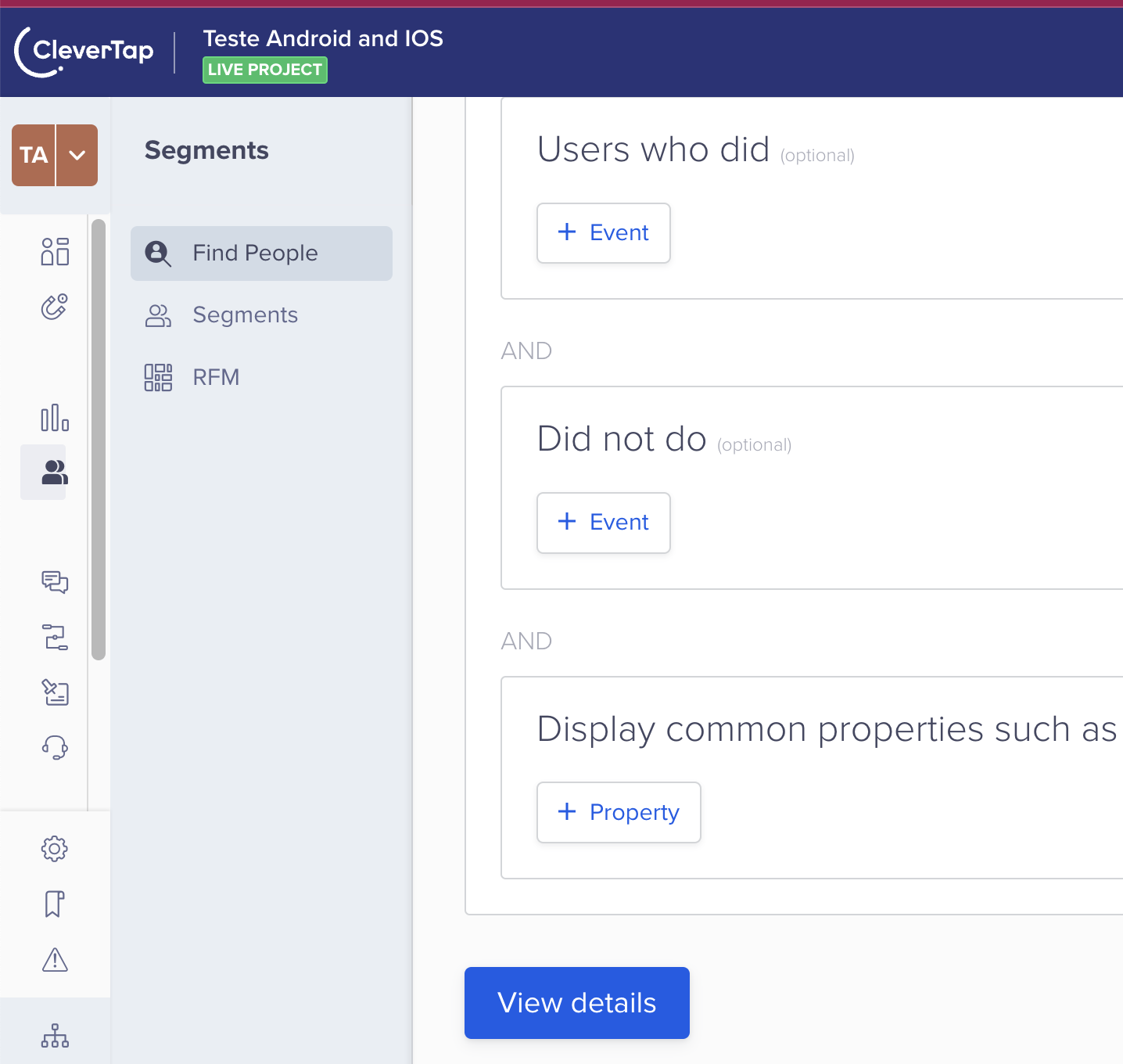
var s = document.getElementsByTagName('script')[0];

s.parentNode.insertBefore(wzrk, s);

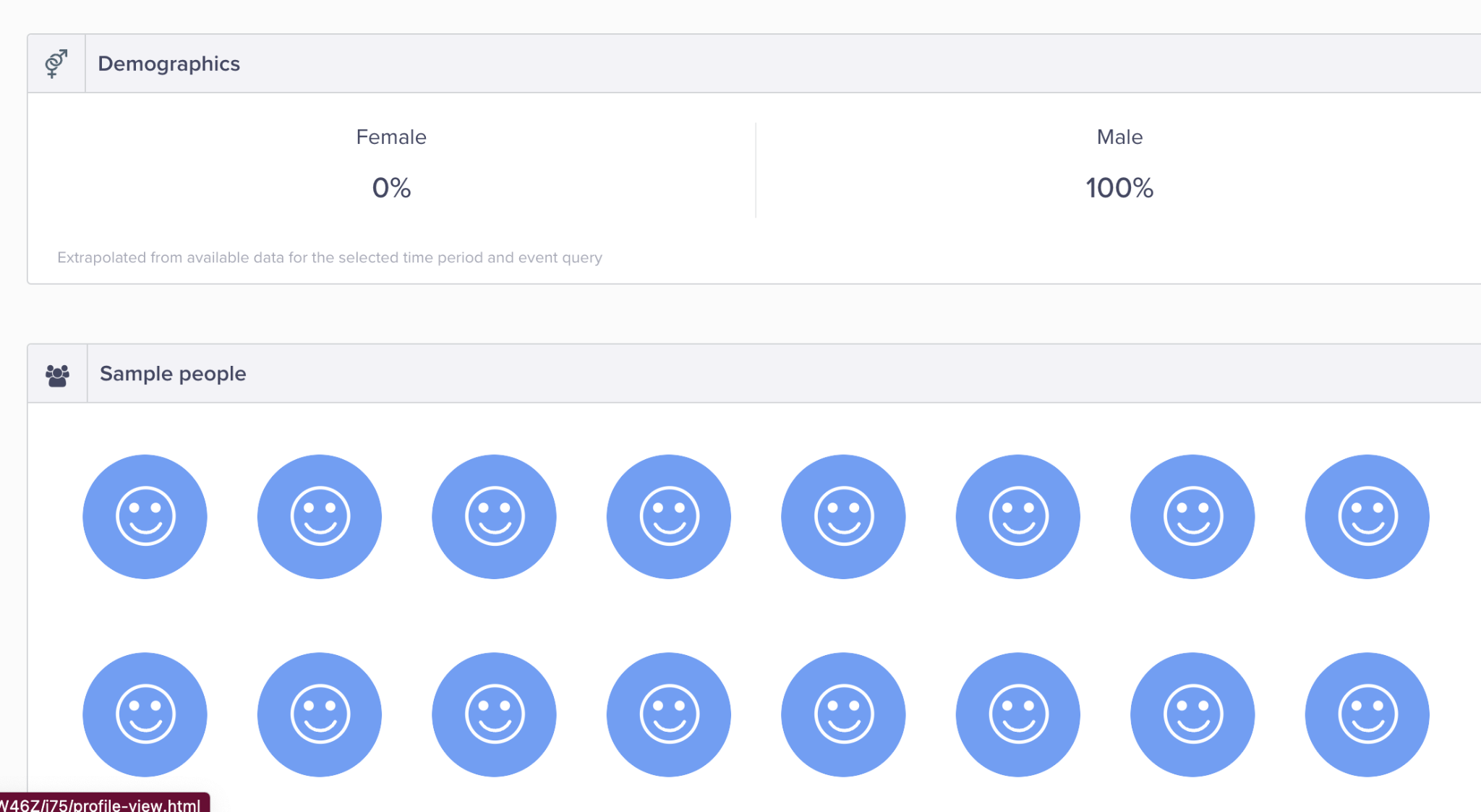
})();

</script>

Now go to the dashboard in -> segments -> find people -> don't add any criteria and click in the view details



Now you can see some anonymous users like the image:



Now your basic SDK integration is done.

* Events

You can add events at any page or files on the Wordpress

In the example below we add in the footer of the page to register the page visit

Remember that for Web SDK integration we don't have an app launch event so is recommended that you add an event to the main page to register for the Web App launch

In this case, we are going to edit the footer.php

<?php wp\_footer(); ?>

<script type="text/javascript">

clevertap.event.push("Main Page Visited", {

"Name": "Main Page Wordpress",

"Date": new Date()

});

* User properties

For testing, we can create a Static onUserLogin function remember to use this function to register and login into your application, and next, we are going to simulate the login HTML form in Javascript.

clevertap.onUserLogin.push({

"Site": {

"Name": "Bruno Hauck - Wordpress", // String

"Identity": 619085435, // String or number

"Email": "bruno@clevertap.com", // Email address of the user

"Phone": "+14155551234", // Phone (with the country code)

"Gender": "M", // Can be either M or F

"DOB": new Date(), // Date of Birth. Date object

// optional fields. controls whether the user will be sent email, push etc.

"MSG-email": false, // Disable email notifications

"MSG-push": true, // Enable push notifications

"MSG-sms": true, // Enable sms notifications

"MSG-whatsapp": true, // Enable WhatsApp notifications

"Team": "San Francisco 49",

}

})

* Push
* In-app and so on

<prerequisites>

* Clevertap account
* Event design sheet
* Firebase account
* Apple developer account
* Worpress and Woocomerce Instalation
* Theme and Template installed

<Identity Management> [refer to this doc for content](https://docs.google.com/document/d/1VrWThpzgHJ5Q8XXnRYNHzBo7f0afqd0YdfMNE4jA-iM/edit)

* What is identity
  + Example: Identity management is a very important part of all marketing as well as user management efforts. You need to uniquely identify users and avoid sending the same campaigns to the same user multiple times, and other such mistakes.
* Why is Identity Management Important?
* Choosing the right unique Identifier:

<Events design> [refer to this doc for content](https://docs.google.com/document/d/1VrWThpzgHJ5Q8XXnRYNHzBo7f0afqd0YdfMNE4jA-iM/edit)

* What are events
* What are user properties?
* How the events and user properties will be helpful
* Explain data types and why important
* Special event

<Integration>

* <Install sdk>
  + Steps to install SDK with screenshots
  + Where can find the account id and token with screenshots
  + Explain important dependencies like firebase, install referrer, exoplayer
  + Initialize Clevertap SDK - with screenshots
  + Activity life cycle code with screenshots and explanation of why it required
  + At the end display a video, on how to check if the data passing to the Clevertap dashboard
  + Webview code if required
* <Location tracking>. - don't have IT for WEB
  + Setlocation code steps with screenshots
  + Gdpr location code steps with screenshots
  + How to check if the GDPR location/setlocation is displayed correctly on the clevertap video
* <Event and user property integration>
  + Onuserlogin code, explanation with screenshots(Should contain all data types)

OnUserLogin function should be used to login into the application and register.

clevertap.onUserLogin.push({

"Site": {

"Name": "Bruno Hauck - Wordpress", // String

"Identity": 619085435, // String or number

"Email": "bruno@clevertap.com", // Email address of the user

"Phone": "+14155551234", // Phone (with the country code)

"Gender": "M", // Can be either M or F

"DOB": new Date(), // Date of Birth. Date object

// optional fields. controls whether the user will be sent email, push etc.

"MSG-email": false, // Disable email notifications

"MSG-push": true, // Enable push notifications

"MSG-sms": true, // Enable sms notifications

"MSG-whatsapp": true, // Enable WhatsApp notifications

"Team": "San Francisco 49",

"Score-value": 36,

}

})

* + Profile push code, explanation with screenshots(Should contain all data types)

The profiles push function only should be used to edit user info or add information to the profile after they already performed an onUserLogin function and the sessions are still valid.

clevertap.profile.push({

"Site": {

"Name": "Bruno Hauck - Wordpress", // String

"Identity": 619085435, // String or number

"Email": "bruno@clevertap.com", // Email address of the user

"Team": "San Francisco 49",

"Score-value": 36, //integer value

"Spent": 55.22,

"Premium-account": true,

"Register-date": new Date(),

}

})

* + Events code with event properties, explanation with screenshots.(Should contain all data types)

clevertap.event.push("Product Viewed", {

"Product name": "Casio Chronograph Watch",

"Category": "Mens Accessories",

"Price": 59.99,

"Date": new Date()

});

* + Charged event code with event properties, explanation with screenshots.(Should contain all data types)

clevertap.event.push("Charged", {

"Amount": 300,

"Payment mode": "Credit Card",

"Charged ID": 24052013,

"Items": [

{

"Category": "Books",

"Book name": "The Millionaire next door",

"Quantity": 1

},

{

"Category": "Books",

"Book name": "Achieving inner zen",

"Quantity": 1

},

{

"Category": "Books",

"Book name": "Chuck it, let's do it",

"Quantity": 5

}

]

});

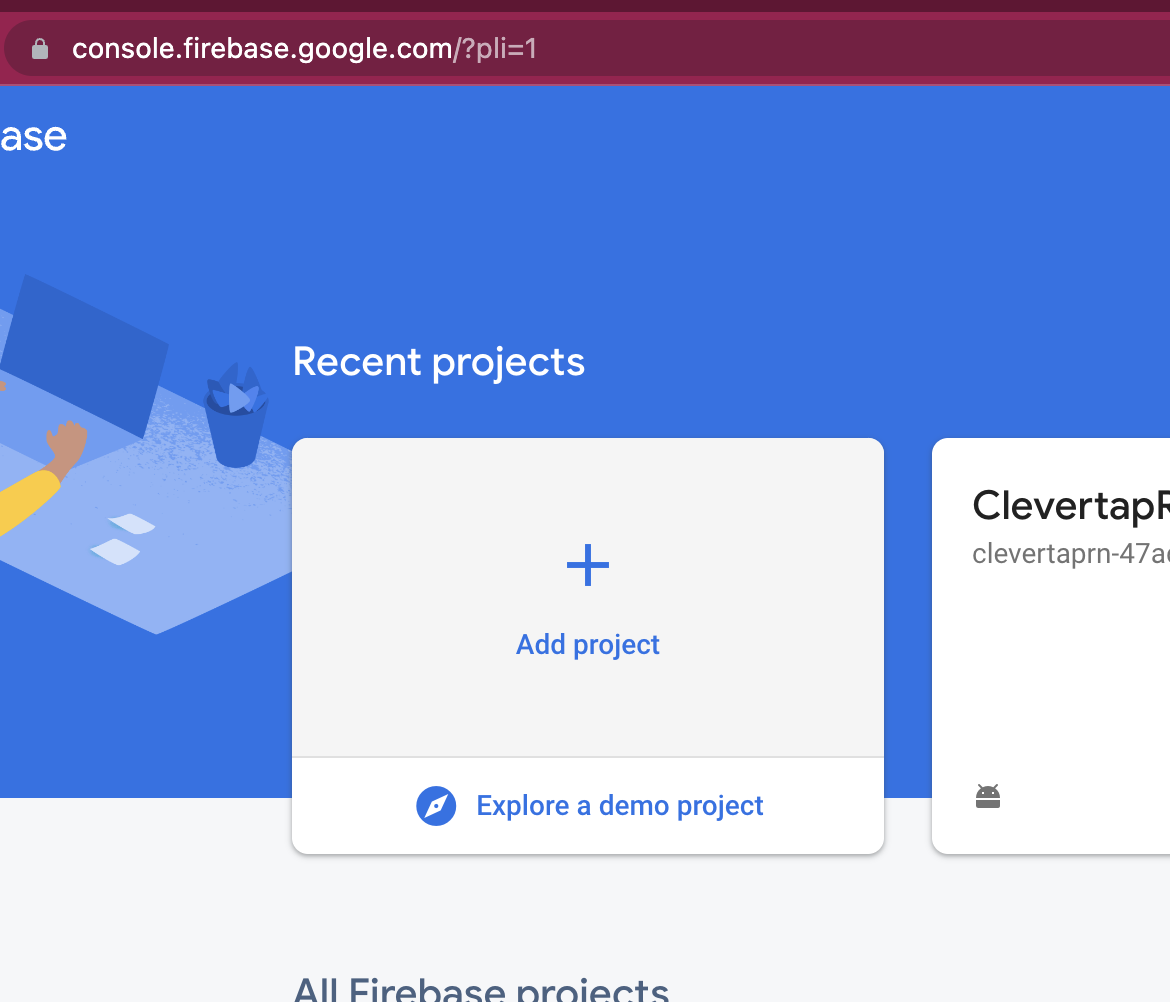
* + Video record to see how the events sending and how can they verify the data type and event properties value - need to be done
* <Push Notifications>(All with screenshots and explanation)
  + Channel ID code
  + FCM receiver
  + Ios code
  + Image notifications
  + Small icon code
  + Video of testing push notifications

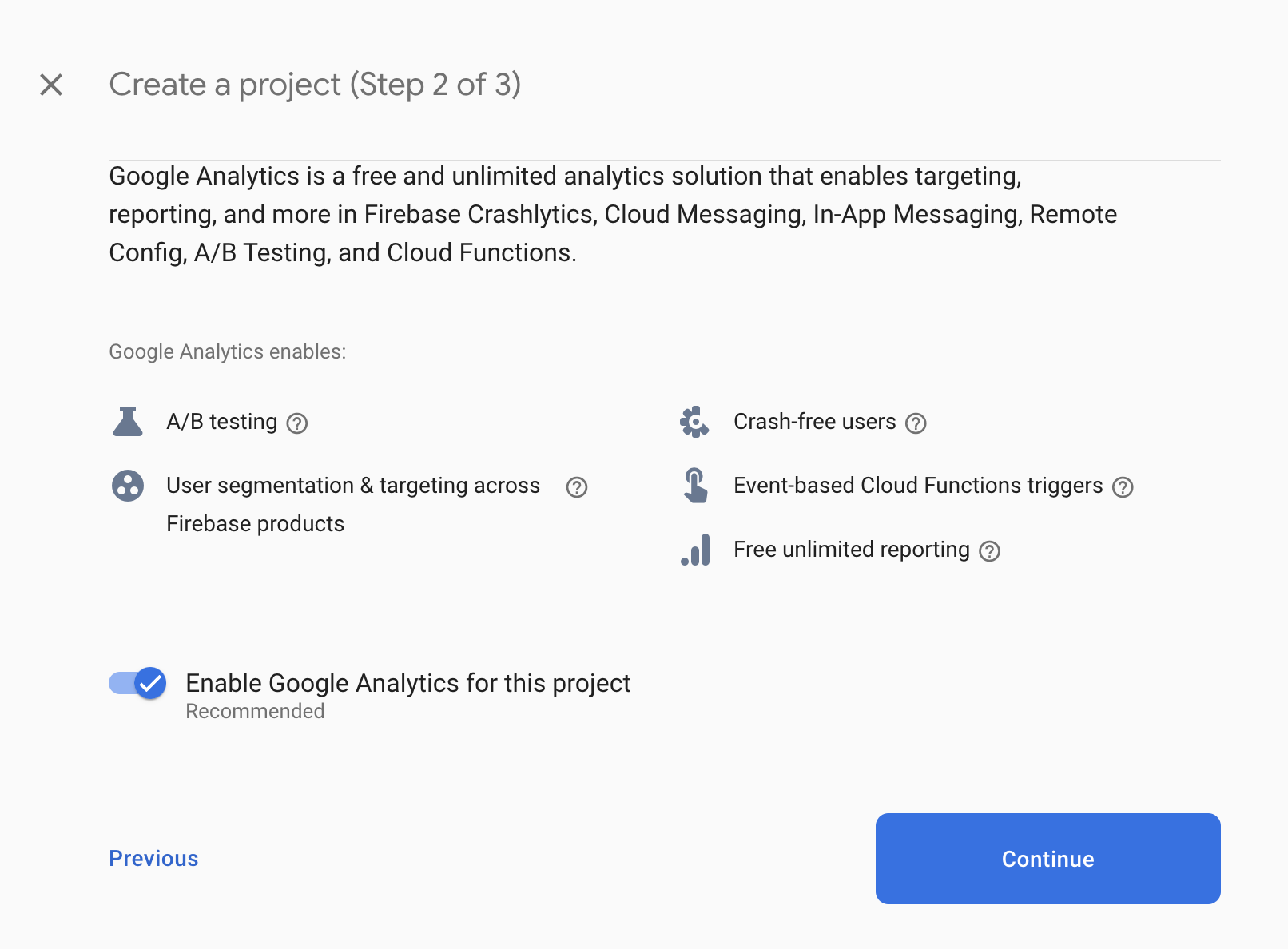
For push notifications, you need to create a project and add the web sdk firebase to your project

Follow the steps

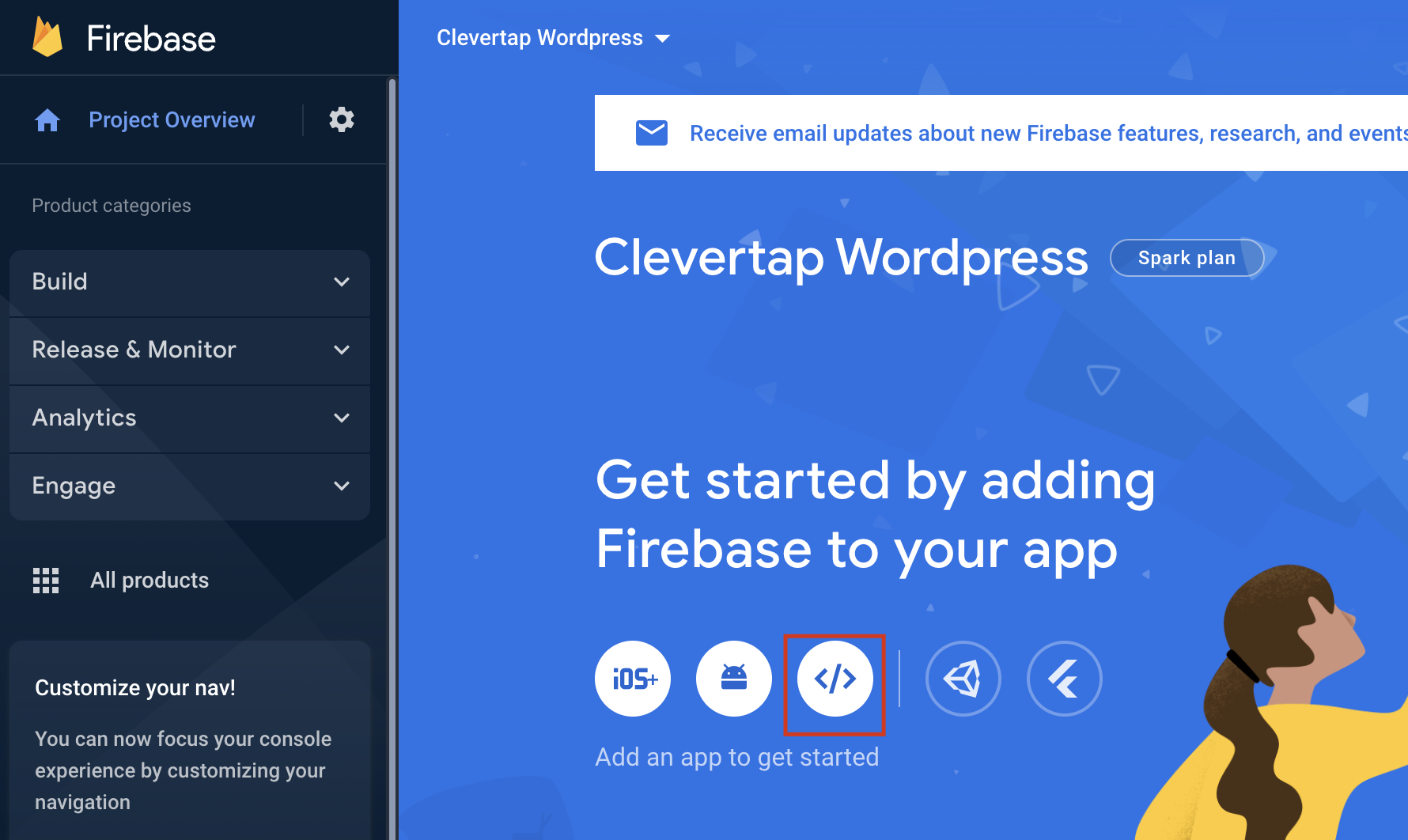
1 - Go to firebase console -> <https://console.firebase.google.com>

2 - Create a new project or use an existing one





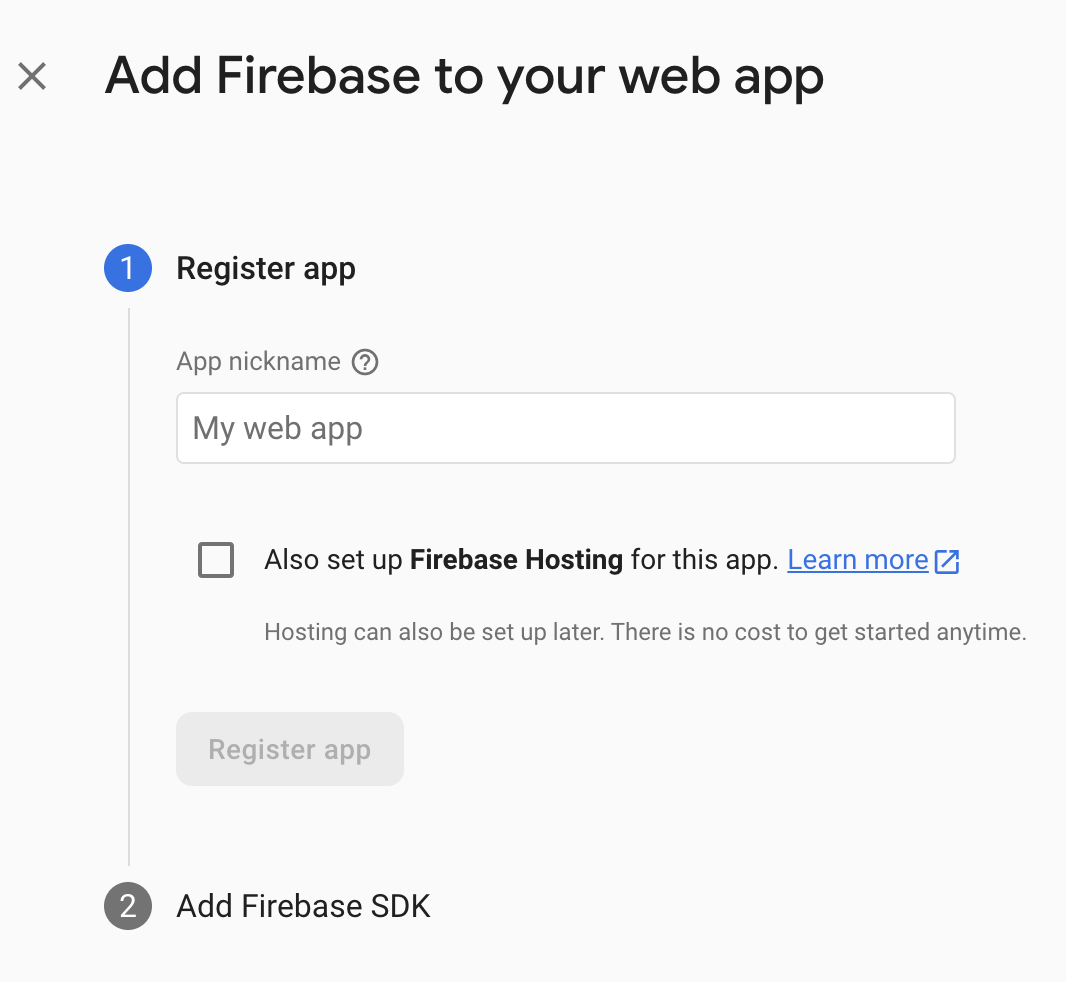
Now you have to add a Web Project



This is going to generate some scripts and you have to add them to your header project.

Remember if you have more than one theme you have to add the integration on all themes for Clevertap to work.

This action will generate the firebase credentials for the push notifications





Here is the code generated and add this to your header.

<script type="module">

// Import the functions you need from the SDKs you need

import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.1/firebase-app.js";

import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.17.1/firebase-analytics.js";

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration

// For Firebase JS SDK v7.20.0 and later, measurementId is optional

const firebaseConfig = {

apiKey: "AIzaSyAR5wo2VeQssCeKUD8AKIZCqLTc1DhHYQQ",

authDomain: "clevertap-wordpress.firebaseapp.com",

projectId: "clevertap-wordpress",

storageBucket: "clevertap-wordpress.appspot.com",

messagingSenderId: "537211652171",

appId: "1:537211652171:web:7cfe0100e32ecc241ce5ce",

measurementId: "G-0Q5GTP8R4C"

};

// Initialize Firebase

const app = initializeApp(firebaseConfig);

const analytics = getAnalytics(app);

</script>

* <In-app notifications> (All with screenshots and explanation) - don't have IT
  + In-app controls example
  + Video of testing in-app notifications
* <Naitve display>(All with screenshots and explanation) - don't have IT
  + Native display code with screenshots
  + Rendering example
  + Events code with examples
  + Video of testing native display
* <App inbox>(All with screenshots and explanation) - don't have IT
  + Clevertap ui code with screenshots
  + Rendering example
  + Video of testing app inbox
* <Web-Pop-Ups> (All with screenshots and explanation)

In the footer please add this code

<div class="btn">CLOSE</div>

<script>

var btn = document.querySelector('.btn');

var wrapper = window.parent.document.getElementById('wizParDiv0');

btn.addEventListener('click', closePopUp);

function closePopUp() { wrapper.remove(); }

var btn = document.querySelector('.btn');

var wrapper = window.parent.document.getElementById('wizParDiv2');

btn.addEventListener('click', closePopUp);

function closePopUp() { wrapper.remove(); }

var btn = document.querySelector('.btn');

var overlay = window.parent.document.getElementById('intentOpacityDiv');

var wrapper = window.parent.document.getElementById('intentPreview');

btn.addEventListener('click', closePopUp);

function closePopUp() { overlay.remove(); wrapper.remove(); }

* + Video of testing web pop notifications