Customerlabs Assessment
ASSESSMENT PART 1

1. General Understanding:

● What is the primary purpose of the Customer Labs JavaScript API documented at the given link?

As per the above documentation, the primary purpose of the Customer Labs javascript API is to help a website tell/report/send data to CustomerLabs what Users are up to - what pages they're checking, buttons they're clicking, and so on. Once CustomerLabs gets this information, it organizes it neatly, creating a clear picture of each user. Then, it shares this structured information with all the other tools like email, advertising tools, etc.

It tracks the user's actions by
1. Pageview,
2. Click tracking,
3. Form Submissions, etc..,

These events are tracked by sending data to the Customer labs by javascript APIs and there is separate API calls for every event made by the user.

For example, for form submissions, the function params:
_cl.trackSubmit(eventName, properties)

● Summarize the key features and functionalities provided by the Customer Labs JavaScript API.

The key features are tracking user activities/ events such as,

1. Pageview
2. Click tracking
3. Form Submissions.
4. Default e-commerce events.

1. Pageview

Pageview events collect all information about pages that customers visit, and the default pageview events get triggered when the customer labs JavaScript is enabled.
These events are automatically shared with all the places you've allowed. If you decide you don't want to share them with a particular place, you can switch off the option for that event in the "Setup Event Workflow" settings of that place.
Function Params:
_cl.pageview(eventName, properties)

2. Click Tracking:

It helps us to track the users's actions from the webpage's links and button click events.
Function Params:
_cl.trackClick(eventName, properties)

3. Form Submissions

Tracking the User actions from the login/login and contact form submissions
Function Params:
_cl.trackSubmit(eventName, properties)

4. Default E-commerce :

There are various e-commerce events that you can track and send to CustomerLabs for better tracking and analysis.

● Explain the importance of website event tracking in the context of Customer Labs.

Importance of Website Event Tracking:
Website event tracking helps to send user event data to CustomerLabs, unifying and syncing their data across multiple platforms. This enables CustomerLabs to segment users based on their behaviors and events, which provides valuable information for better decision-making.

2. Initialization and Setup:

● Describe the steps involved in initializing the Customer LabsJavaScript API on a website.

Steps involved in the integration of customer labs CDP with Shopify :

Shopify (or) Shopify Plus is an e-commerce platform that allows users to set up an online store. By connecting your Shopify store with CustomerLabs CDP, you can start tracking the events (user-behavioral activity) of your website visitors without writing any code.

1. Log in to the Customer Labs account.
2. On the Home page → Under "Connect your domain" → Enter your website URL → Click Save.

3. Copy the Tracking code
4. Login to your Shopify account → Click Online store
5. Under Themes, go to Actions → Edit code
6. Under theme.liquid, search for "</head>" and paste the CustomerLabs code in the header section and Save.
7. Create a custom pixel in Shopify by navigating to Settings → Customer events → Add a custom pixel
8. Give the Pixel name as Customerlabs → Add pixel

● What parameters are required when initializing the Customer Labs JavaScript API, and what do they signify?

function(p1, p2, p3,p4,p5,p6, mid){

//

code

}

P1→event name→String

P2→even attributes→Object

p3→products→Array

p4→identify→boolean

p5→user external_id→String

p6→Group external_id→String

3. Event Tracking:

● How does the Customer Labs JavaScript API facilitate event tracking on a website?

As per the documentation, Customer Labs JavaScript API facilitates event tracking on a website by
1. Pageview
2. Click tracking
3. Form Submissions.

● Provide an example of tracking a custom event using the Customer Labs JavaScript API.

```
// Assume the following variables contain the user information

var userId = "12";
var userEmail = "steffy@gmail.com";
var userName = "Steffy";
var userPhone = "9876543210";

// Customer Labs JavaScript API code for tracking the custom events
```

```
customerlabs.track('user_registered', {

  user_id: userId,
  user_email: userEmail,
  user_name: userName,
  user_phone: userPhone

});
```

variables (userId, userEmail, userName, and userPhone) containing the relevant user information, and it uses the customerlabs.track function to send the custom event named 'user_registered' along with the user details to Customer Labs.

4. Custom Properties:

● Explain the concept of custom properties in the context of event tracking with the Customer Labs JavaScript API.

In Simple words**,** customProperties lets you track all event & user-related information. Custom properties provide additional information that you can attach to events or users to offer more context and details. They allow you to track specific data points relevant to your business or application, including user-specific details or parameters related to an event.

```
{ "customProperties": {
   "page_url": {
     "t": "string",
     "v": window.location.href
   },    "clicked_from": {
     "t": "string",
     "v": "header section"    },
} }
```

● How can you include custom properties when tracking events?

We can create CustomProperties by creating an object with "customProperties" as key and the properties that are needed to add as values. For example,

In form submission event, the API call would be ,

```
var cl_form_submission = setInterval(function() {
    if (((window.CLabsgbVar || {}).generalProps || {}).uid) {
        var eventName = "contact form submission";
        var properties = {
            "customProperties": {
                "first_name": {
                    "t": "string",
                    "v": "arjun"
                },
                "phone": {
                    "t": "string",
                    "v": "+9198394343"
                },
                "email": {
                    "t": "string",
                    "v": "arjun@clabs.co"
                },
                "form_submitted_from": {
                    "t": "string",
                    "v": "customerlabs.co/contact_us"
                }
            }
        }
        _cl.trackClick(eventName, properties)
        clearInterval(cl_form_submission)
    }
}, 1000);
```

As we can see the above function has a property variable which stores the values of "customProperties".

Custom properties:

```
{
  "customProperties": { /* reserved keyword */
    "page_url": { // property name
      "t":"string", // property type
      "v": window.location.href  // property value
    },
    "clicked_from": {
      "t":"string",
      "v": "header section"
    },
    }
}
```

5. User Identification:

● Describe the methods available for identifying users with the Customer Labs JavaScript API.

Identifying a user

An anonymous User ID is assigned to every new visitor. All events associated with this ID until the user reveals personal data. When the user provides personal data, old anonymous sessions are merged into that user.

Map user information to a cookie

To link user information with a cookie, set up a "Create user" event before tracking activities like sign-ups or purchases. This ensures the system maps user traits and external IDs to a unique anonymous user ID, commonly using identifiers like email or phone number.

Update user event

In case, of multiple forms where you collect the same user data for the second time, set up an “Update user” event, so that additional traits are appended to that particular user.

Assigning user traits

While setting up the create or update user event, the form fields that give more information about the user must be added as user traits. In the above form, fields such as name, email, phone, and company should be added as user traits.

User External IDs

Apart from fetching standard user identifiers such as email & phone number, the system also supports the collection of External User IDs such as google_analytics_client_ID, Facebook_fbp, Facebook_fbc

● What is the purpose of user identification, and how does it contribute to analytics?

Purpose of identifying the user:

The main purpose of user identification is to identify every user event data and organize/combine the data under a user's behavior. This enables the customization of user experiences based on individual preferences and behaviors. Significantly, it is mainly helpful in analytics of the user's behaviors according to the user data which is identified by User-Identification.