Technical instructions

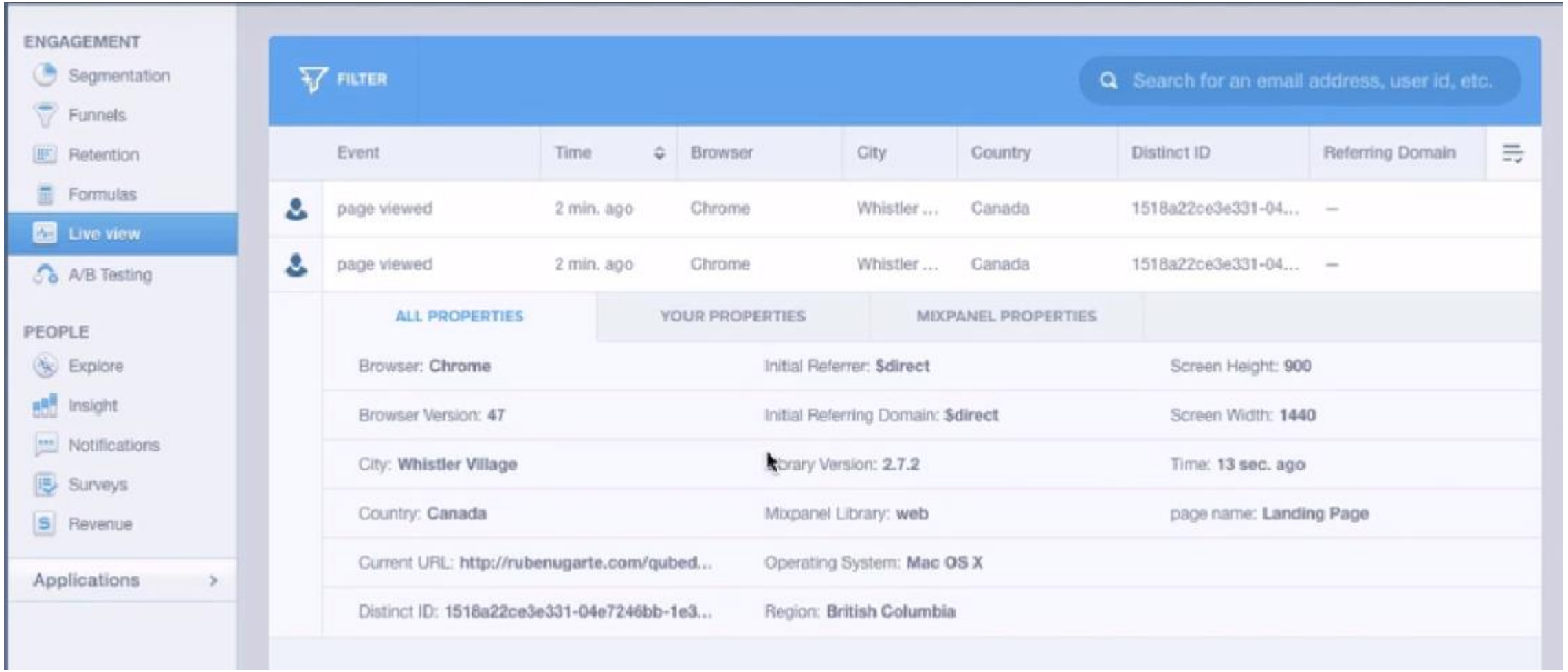
New daily Data Pipeline

Agenda

- Project goal
- Introduction
- Technical requirements
  - Data Wrangling and
  - Deliverables

Project goal

The goal of this project is to extract and provide daily information (events) from MixPanel platform and make available for the data science team. Basic sample below



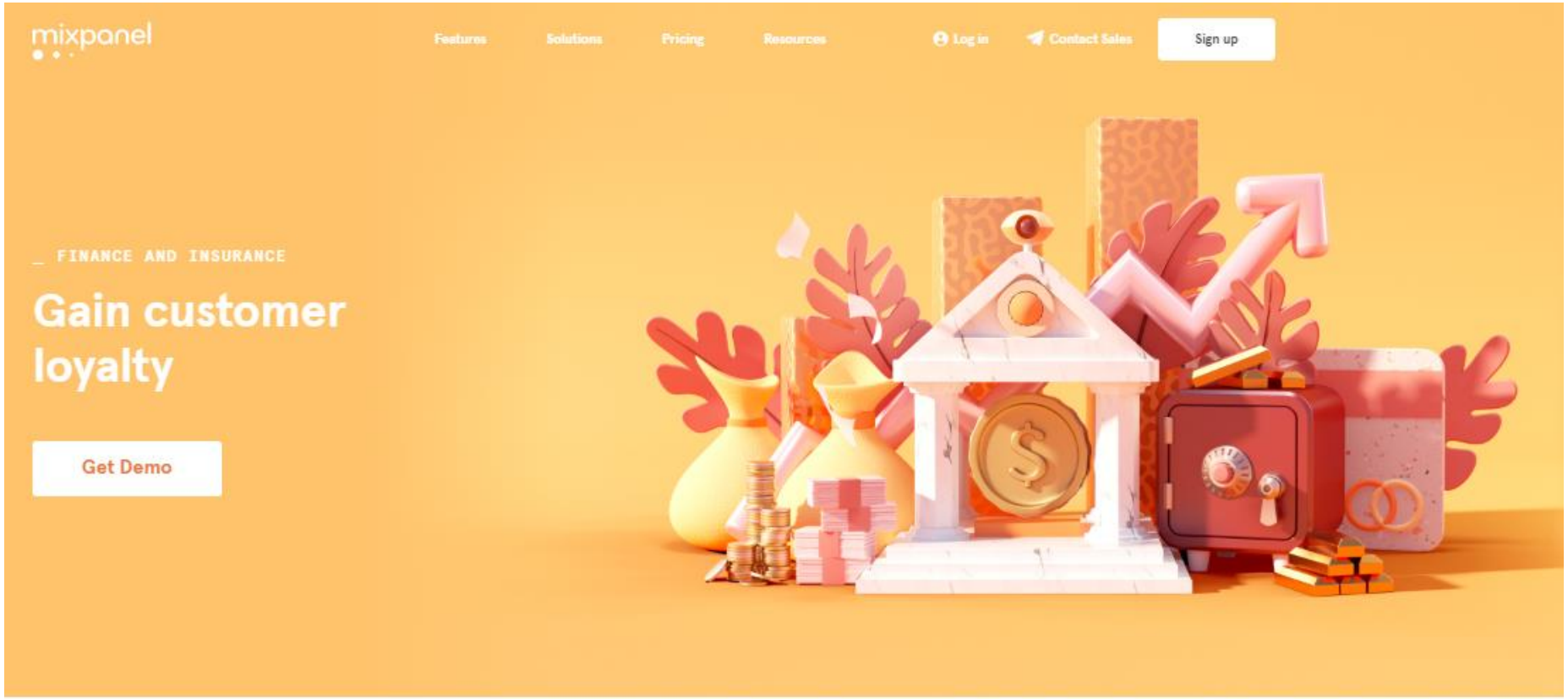
Introduction – MixPanel platform

Mixpanel is a business analytics service company.

It tracks user interactions with web and mobile applications and provides tools for targeted communication with them.

Its toolset contains in-app A/B tests and user survey forms.

Data collected is used to build custom reports and measure user engagement and retention.



Betterment experian. PayPal Lemonade

Technical requirement

The requirement is to implement an automated data pipeline to extract, clean, and merge data from company analytics platform

- The data is stored on MixPanel (https://www.mixpanel.com)
- Here you can find the MixPanel API documentation: https://developer.mixpanel.com/docs/implement-mixpanel
- This is the Secret Key for a MixPanel test database: SECRET KEY XXXX)9b03e4631ba22f687
- There is data in this test database from Oct 31 to Nov 04

You must architect and implement a data pipeline that connect to MixPanel daily, and pull the following events (Export API):

1. Company First Start
2. Conversation Started
3. Conversation Completed
4. A/B - Onboard - Voice
5. Subscription Confirmed

Data Wrangling

Complete the following data wrangling steps to prepare the data for our data scientist:

1. Download the events data. The API returns a JSON structure
2. Define a function to shorten each word of the event name to its first three characters, convert them to lowercase and remove spaces and special characters
    a. E.g. *Conversation Started* event must become ***consta***.
3. Convert the JSON properties to lowercase and remove white spaces
4. Rename the JSON properties (except for distinct_id and conversationid). The property name must be the concatenation of the original one with the shortened event name (step number 2).
    a. E.g. Column ***conversationindex*** from *Conversation Started* event must become ***conversationindexconsta***.
5. Merge all events in one data structure (Data Frame), matching on distinct_id and conversationid
6. Reduce the merged data structure to include only the following properties:
    a. **distinct_id, conversationid, app_versionconsta, conversationindexconsta, conversationstartedatconsta, valueabonbvoi, pathscompletedconcom, datelocaltimeyoufirsta, frequencysubcon**
7. Remove rows with **frequencysubcon** equal to "monthly"
8. Convert **valueabonbvoi** from a string variable to a numeric variable with values 1 and 2
9. Split the array from **pathscompletedconcom** to create new columns called path1, path2, path3, etc.
10. Calculate the number of hours between **datelocaltimeyoufirsta** and **conversationstartedatconsta** and create a new column called **hours_since_start**

Data Wrangling

Label / target information (num_convo) and additional feature (hours_since_last)

- Calculate the number of Conversations Started events that each user has in the dataset and create a new column called **num_convo**
- Calculate the amount of time in hours between Conversations Started events for each user and create a new columns called **hours_since_last**

Deliverables

Your deliverable should include:

1. The final output of the data pipeline: a CSV file containing the merged data, stored in a safe repository.
2. The code of the your project (you can use any programming language).
   You must publish everything in a GitHub repository.
3. A documentation about how to run the application.