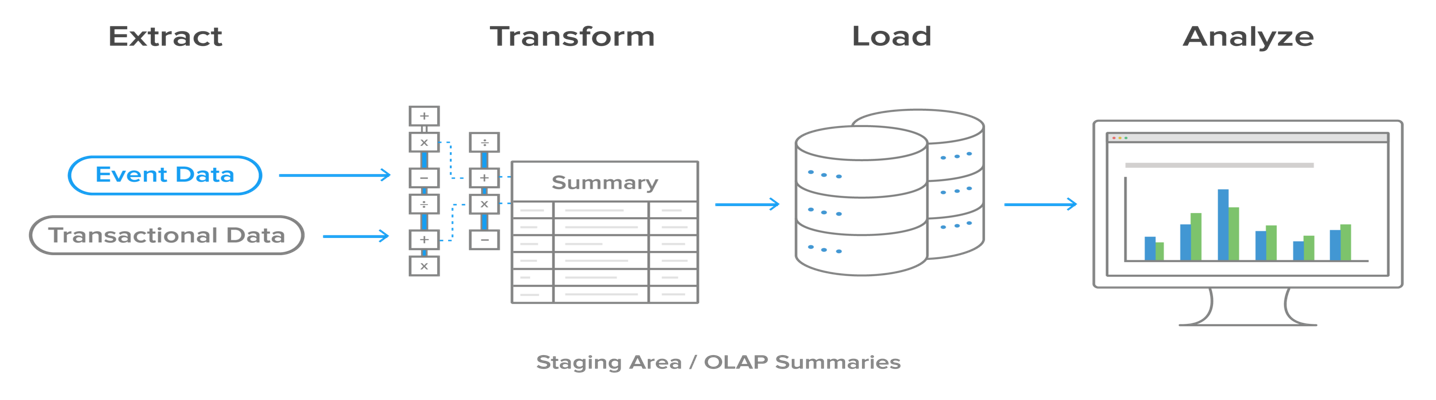
**ETL Project - Forex Data**

The scope of this project is to source Forex data from two varied data sources and perform ETL on it



***E***xtract

We went through numerous datasets on various public platforms such as Kaggle and Dataworld to gather suitable Data for this project. Kaggle provided us with daily historical Forex data for 22 countries. Since the idea was to source data from two or more sources we decided to fill in the blanks in the Kaggle dataset and make it current by using APIs.

* Sources for the Datasets
  1. Kaggle.com – Historical Forex Data from 2000-2019
  2. API – fixer.io – Current Forex Data from 2020 Jan 1st to present

***T***ransform

Steps taken to transform the data:

* Historical Forex Data – (2000-2019)
  1. Manually going through the data to look for discrepancies
  2. Used Pandas to read the csv files
  3. Deciding the columns needed and the columns to be discarded
  4. Dropping rows with Non-Decimal values (ND)
  5. Changing column names for better readability and understanding
  6. Dropping the index column
  7. Export the Dataframe to a csv file.
* Current Forex Data – (1/1/2020 – present)
  1. Created an account in fixer.io website.
  2. Access to API key was obtained once the user login was created. It is free to hit upto 10,000 per month.
  3. API Endpoints: There were five API end points through which we could access different kinds of data, all starting out with this base URL:

https://data.fixer.io/api/

Concatenate the unique access key to one of the end points as a query parameter.

https://data.fixer.io/api/latest?access\_key=<API-KEY>

* 1. Of the five end points, we used the “latest” endpoint and “convert” enpoint to retrieve the latest forex rates of all the 22 countries and convert them to USD standards.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

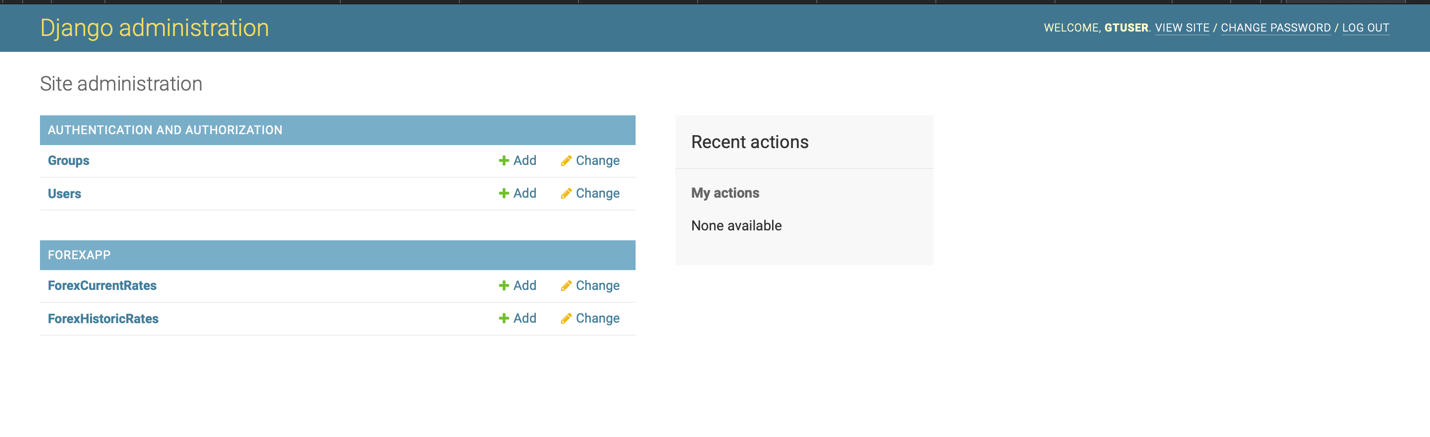
Description automatically generated

* 1. Integrated the current dataset with the historic dataset for or ForEx Data Analysis application.

***L***oad

## The last step was to transfer our data from the csv files to a database. Since our data was structured we decided to use MySQL. We used Django to create models and load the data. The reason behind this choice was to conduct data loading in SQL along with Web framework. By means of Django loading and the frameworks were both managed simultaneously.

* **Django Details**

****

## To run the Django PROJECT

* + 1. python3 manage.py runserver
    2. It opens the Django in its default port:
    3. Django Admin Super User: User: gtuser Password: rpython

## "postgresql" used in this project is from Google Cloud and the access is restricted from GeorgiaTech IP only.

## For Public Access Please update the settings.py for postgresql

* 1. DATABASES = { 'default': { 'ENGINE': 'django.db.backends.postgresql\_psycopg2', \* postgresql 'NAME': 'databasename', 'USER': 'user', 'PASSWORD': 'password', 'HOST': 'hostname or ip', 'PORT': 'port', } }

## Django Instructions

* + 1. Clone the repository URL <https://github.com/ishanku/ForexDataAnalysis>
    2. Go to the Folder "ForexDataAnalysis"
    3. conda activate ""
    4. conda install django

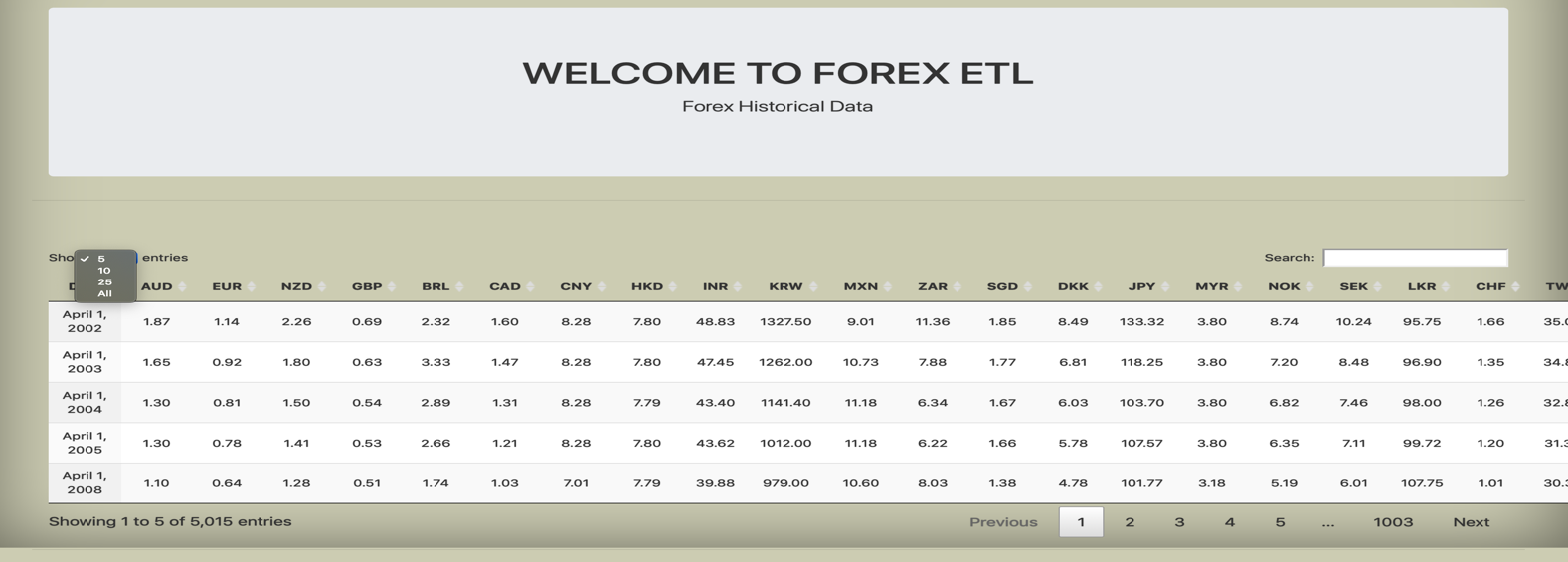
## To Create Models after updating settings.py

* + 1. python3 manage.py makemigrations
    2. python3 manage.py migrate
    3. python3 manage.py runserver
    4. This will launch Django in the port number 8000 visit <http://localhost:8000>

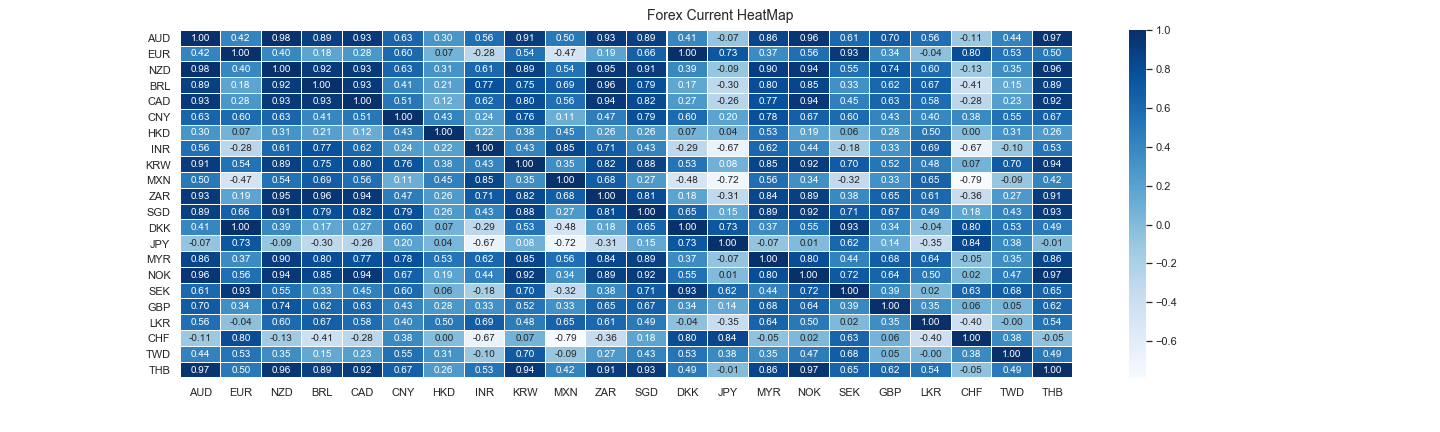
## Loading Details:

## Python file we used to load the data. - ForexDataAnalysis/forexapp/management/commands/**load\_forex\_data.py** /

## We created two tables – Forex Historical Data and Forex current Data. The tables were created using java script for better representation and making the table interactive for better usability.

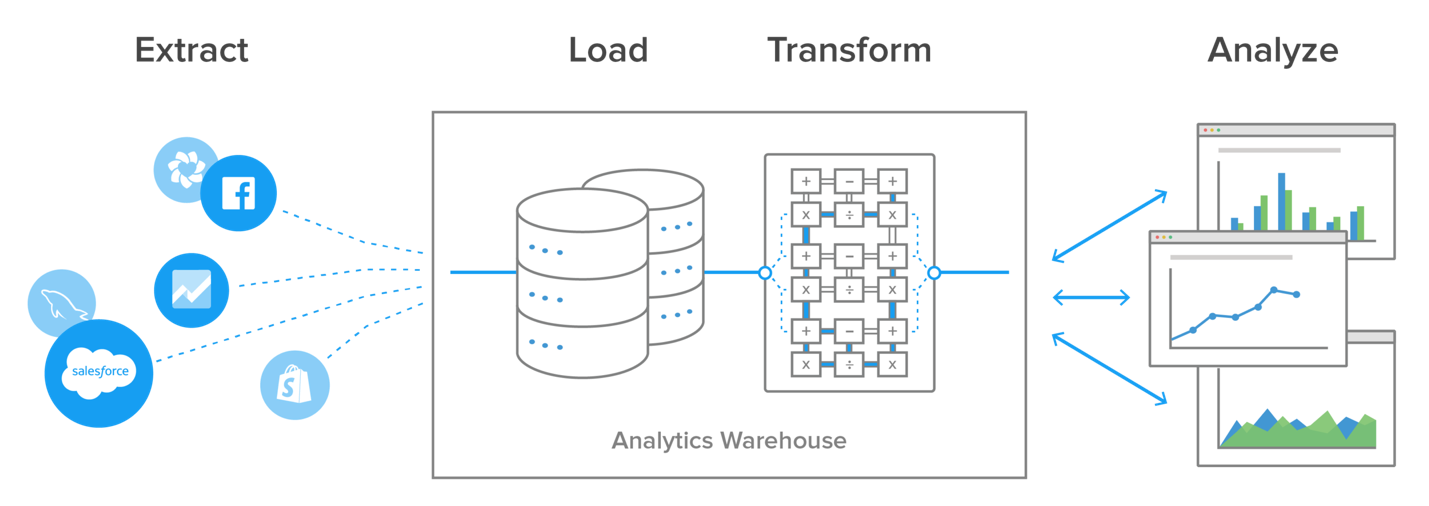


* 1. HeatMap to help with better data visualization



**Considerations**

Due to the time factor we could not completely expand on the potential of this project. Also based on the instructions we limited ourselves in certain areas which we want to research further eventually.



* Support for change data capture (CDC) – We would consider a better modeling of our data so as to support incremental data. That would allow us to update our analytics warehouse with new data without doing a complete reload.
* Use of APIs – We decided to use the API’s up until now in closed form for the project i.e. not giving access to the client. This was considering the cost factor involved in using APIs and to decrease the monitoring element.