Round 1

About

This assignment is a part of evaluation process for candidates who applied for a role at our organization.

Data

You are being provided with the required source data in file 'customers.csv' & 'orders.csv'.

Background

The assignment focuses on development (design is provided, hence not required) middleware REST API(s) for a web application that can dynamically extract, transform, & load data from source to target data layer.

Problem

You need to implement from scratch in development stack (Angular.JS/ React.JS/ Ember.JS – Django REST Framework – MYSQL Database) as per design description below.

Task

Please refer attached *Data_Analyzer1.jpg* & *Data_Analyzer2.jpg* while going through below steps.
Demo video *(*http://bit.ly/2i95hoj*)* Enable subtitles on YouTube

Step 0 :

Create an instance in MYSQL database of your choice and load the data customers.csv in the MYSQL Database. (You can name database as 'busigence')

Step 1 :

1. Select Format in the Data_Analyzer1.jpg :

i) If you have selected the MYSQL Database option then UI should ask Username & Password and the IP address will be your local host (all these parameters you should have created in step 0). (Refer Data_Analyzer1.jpg: 1.Select Format)

ii) If you have selected the CSV option then a button must be created which will help in browsing the CSV file and loading into UI. (Refer Data_Analyzer2.jpg: 1.Select Format)

Step 2 :

2. Select Source :

i) If you have selected MYSQL Database then it should display all the databases (including 'busigence' in this case) in the MYSQL Database. If you click on the Database Name it should show Tables (in our case only customers.csv) in the respective database. If you click on the table it should show columns in the respective tables where you can select the fields you want to transform as well as the primary key. (Refer Data_Analyzer1.jpg : 2.Select Source)

ii) If you have selected CSV then it should show the list of CSV's that you have loaded (in our case only orders.csv). If you click on the CSV it should show the list of columns in the respective CSV you can select the fields you want to transform as well as the primary key. (Refer Data_Analyzer2.jpg : 2.Select Source) .

Step 3:
3. Visualizer :

i) D1 (from MYQL) & D2 (from CSV) (Refer Data_Analyzer1.jpg : 3. Visualizer) are the two dataframes that you have generated by selecting the fields from both sources (MYSQL Database & CSV) .

ii) After Generating D1 & D2 you have to apply join transformation (logic is attached in the assignment) in which you would have created a REST API which will trigger the join function that take's D1 & D2 and parameters (column name & jointype) as input which will be passed by the REST API to the external .py file (main.py) .

iii) After applying the join transformation you have to apply sort transformation(logic is attached in the assignment) on the data that has been generated after applying the join transformation in which you will create an REST API which will trigger the sort function that take's parameters (column name & sorting type) as input which will be passed by the REST API to the external .py file (main.py).

iv) After clicking "Run the Mapping" button both API's will submit the parameters to the external Main.py file and execute it .

v) The resultant dataframe after executing the Main.py(**df_d** in the code) is saved in the MYSQL Database with name **MYSQL_T .**

Step 4:
4. Preview :

i) User should be able to see the preview when he clicks on the preview button in Select Source Section (Refer Data_Analyzer1.jpg : 2.Select Source) .

Main.py :
Contains logic for Join and sort transformations .
Remember you have to convert the data loaded in frontend to pandas dataframe in order to execute main.py.

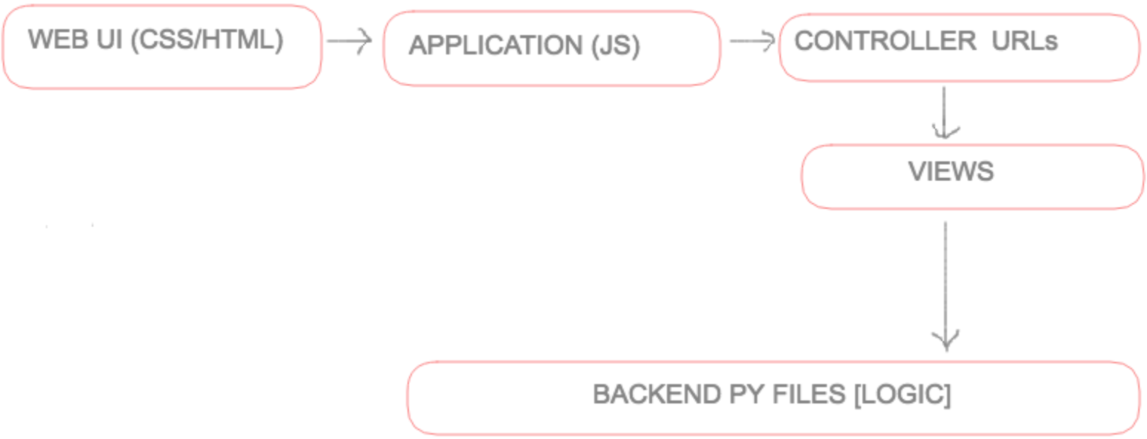
i) **Transform_join** function does the join transformation for which the API that you have developed will pass the necessary parameters .
df_a & df_b : Are the D1 & D2 that you have generated by selecting fields from NoSQL Database and CSV sources.
column_name : It is the column name on which you want to join the two tables

join_type : It is the join type which you want to perform for example : inner, outer, right, left etc.,

ii) **Transform_sort** function does the sort transformation for which the API that you have developed will pass the necessary parameters.
df : it is the data that has been generated after performing joining transformation(**df_c** in the code).
column_name : column name on which you want to perform the sorting operation.
sorting_type_value : it can be 0 or 1 .if it is 0 then ascending .if it is 1 then descending.

After executing the sort transformation data frame **'df_d'**(final output after joining and sorting) should be saved in the MYSQL database with name **MYSQL_T** .

Acceptance
You are required to develop REST APIs as per below architecture.



- Web UI can be made simple but working (we won't score you on this)
- Application can be developed on AngularJS/ReactJS/EmberJS/Normal JS/JQuery
- Controller shall propagate parameters from Application to Views. Don't show parameters in URL string for security reasons
- Don't hardcode either data properties (database, tables, fields etc.) or business logics in views.py. It should call functions from main.py
- Backend PY files are being provided (main.py) Only Main.py shall pull/push/transform data present in MYSQL in memory and return results

Submission
You are required to demo the application in your local system through screen sharing and/or it can be hosted on any free hosting portal (e.g. https://www.000webhost.com) and share us the link for web application developed

Evaluation
You shall be scored on the following:

1. Design
2. Functionality
3. Execution