Project Documentation

JSON Grid Viewer

(Web Application)

(Version 1.0.0)

Project Guide

- **Mr. Dinesh Badgujar**
- **Mr. Nilesh Babar**

Prepared by

- **Abhishek Tiwari**
- **Aditi Khandelwal**
- **Ashish Kumar**

DBOI Global Services Limited

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Intended Audience 3

1.3 Overview 3

1.3.1 JSON 3

1.3.2 What does JSON look Like ? 3

1.3.2 Excel 3

1.3.3 What does Excel look Like ? 3

1.3.4 CSV 4

1.3.5 Hadoop 4

1.3.6 HDFS 4

1.3.7 SQL 4

1.4 Product Scope 4

2. Overall Description 5

2.1 Application Perspective 5

2.2 User Flow for Web 6

2.3 Application Functionality 8

2.3.1 HomePage 8

2.3.2 Options Page 8

2.3.2.1 JSON File 8

2.3.2.2 JSON URL 10

2.3.2.3 JSON Input 11

2.3.3. Customisation Page 12

2.3.4 Preview Page 13

2.3.5 Save files to Hadoop 14

2.3.6 Create Edit and Download JSON 15

3. Tech Stack Used 16

4. Installation Instructions 16

4.1 Install Node 16

4.2 Install Python 17

4.2 Install Hadoop 17

5. Future Scope 19

1. Introduction

1.1 Purpose

The JSON format is often used for serializing and transmitting structured data over a network connection. It is used primarily to transmit data between a server and web application, serving as an alternative to XML. The Readability of Json files has always been an area of concern.
This project aims to develop an Application which converts the Json File into Excel / CSV format which would be readable to the users. The Data is then stored into Hadoop/SQL servers. The Application provides the user an interface to perform SQL based queries on the dataset. This would help the users to understand the data better and improve the performance.

1.2 Intended Audience

This SRS is intended for technical as well as Business users. It can be used by the project managers, developers, testers and documentation writers.
This SRS aims to provide a brief overview of the JSON grid Viewer which will help to convert the Json File into Excel / CSV and save the data to the desired data servers such as SQL /Hadoop.

1.3 Overview

1.3.1 JSON: JSON, or JavaScript Object Notation, is a minimal, readable format for structuring data. It is commonly used for transmitting data in web applications (e.g., sending some data from the server to the client, so it can be displayed on a web page, or vice versa).

1.3.2 What does JSON look like?

A JSON object is a key-value data format that is typically rendered in curly braces. ... Key-value pairs have a colon between them as in "key" : "value" . Each key-value pair is separated by a comma, so the middle of a JSON looks like this: "key" : "value", "key" : "value", "key": "value" .

1.3.3 EXCEL: Excel is a spreadsheet which features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications.

1.3.4 What does Excel look like?

An Excel is the format where the data is stored in cells inside the rows and columns in a tabular manner. Sorting, Searching, performing arithmetic operations are all the stuff which can be performed easily on Excel.

1.3.5 CSV: CSV is a simple file format used to store tabular data, such as a spreadsheet or database. CSV stands for "comma-separated values". Its data fields are most often separated, or delimited, by a comma.

1.3.6 Hadoop: Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs. Hadoop is not a type of database, but rather a software ecosystem that allows for massively parallel computing.

1.3.7 HDFS: HDFS is a distributed file system that handles large data sets running on commodity hardware. It is used to scale a single Apache Hadoop cluster to hundreds (and even thousands) of nodes.

1.3.8 SQL: SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.

1.4 Product Scope

Objective:

Business User: To convert the JSON data into excel and CSV format so that it is easy to read , understand and analyze the data.

Technical User: Prefer storage of data in data servers like SQL/Hadoop so that easy analysis can be done using queries.

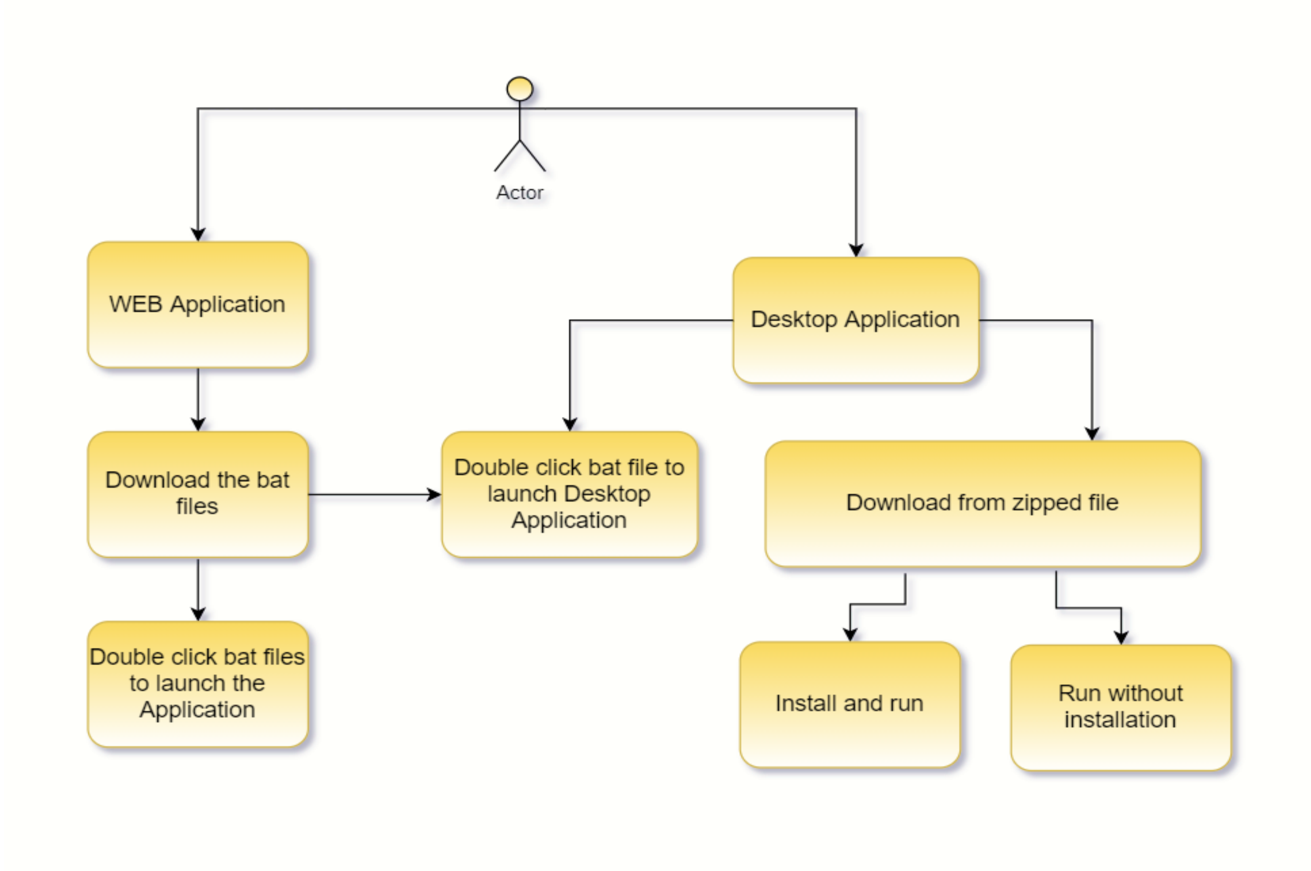
Benefits:

- Improved and easy readability of JSON file by converting it to Excel/CSV format.
- Can upload from any source such as File, URL, Raw Json.
- Store the data into Hadoop.
- Use the converted Excel/CSV for other uses.
- Data analysis using SQL Queries.
- Perform an excel-like operation on the converted data without even downloading.
- Check the validity of the JSON file, correct it and download it.

2. Overall Description

2.1 Application Perspective

The application focuses on converting the Json files into Excel/CSV files and also uploading them on Hadoop. Once the JSON file is ready the user can view the Converted files and can perform operations similar to that of Excel. The user can also perform SQL queries on the data set.


Actor
WEB Application
Desktop Application
Download the bat files
Double click bat file to launch Desktop Application
Download from zipped file
Double click bat files to launch the Application
Install and run
Run without installation

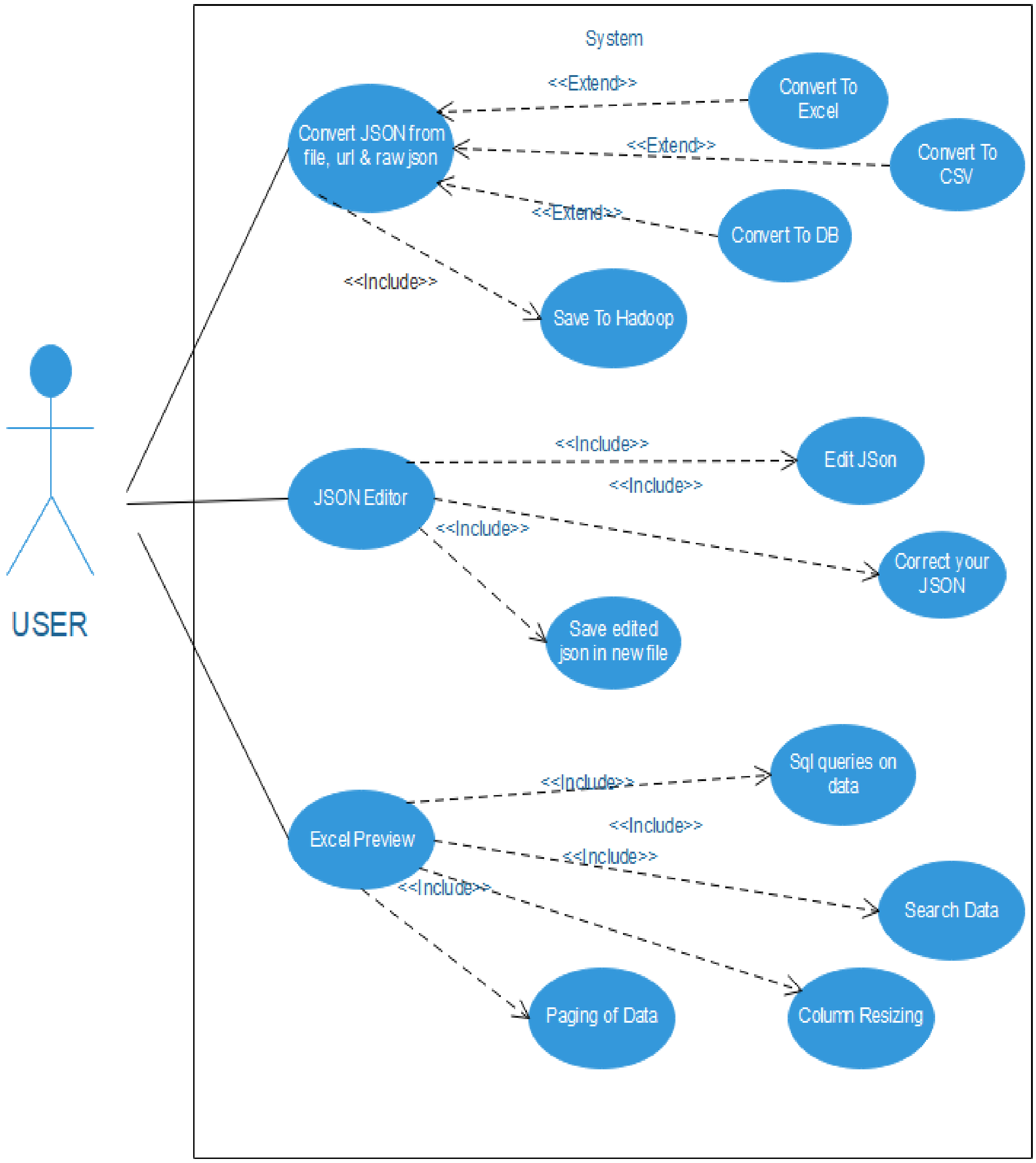
Run Backend

1. Open a new cmd window
2. cd *path to Json-Grid-View folder*
3. cd backend
4. python App.py

Run Frontend

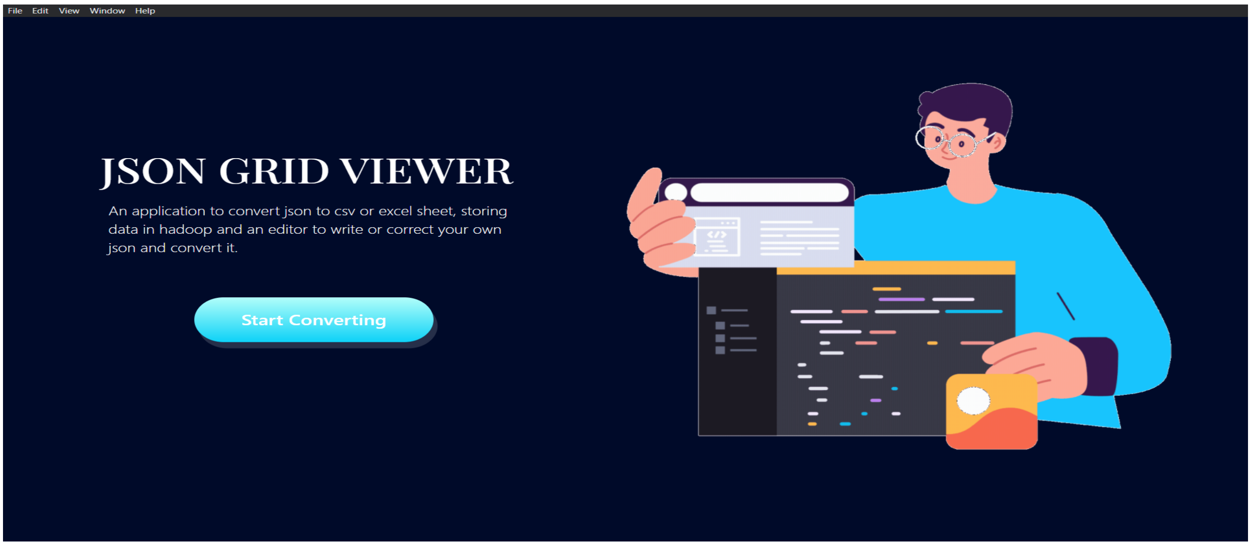
5. Open a new cmd window
6. cd *path to Json-Grid-View folder*
7. cd frontend
8. npm install
9. npm start

2.2 User flow for Web Application



2.3 Application Functionalities

2.3.1 Home Page

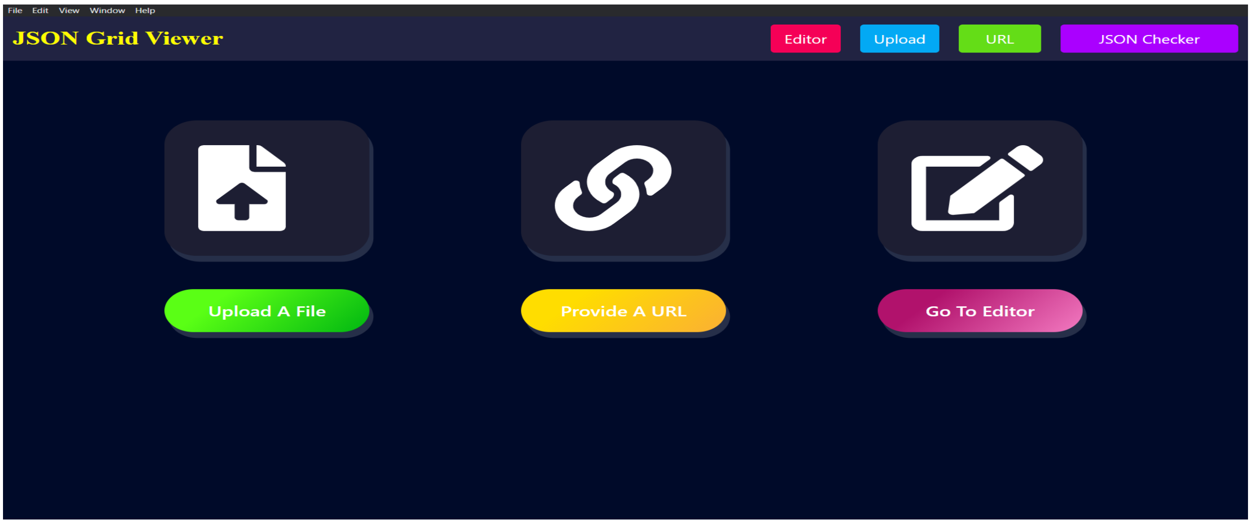


JSON GRID VIEWER

An application to convert json to csv or excel sheet, storing data in hadoop and an editor to write or correct your own json and convert it.

User needs to click the Start Converting button to go to the options Page for converting the data.

2.3.2 Options Page

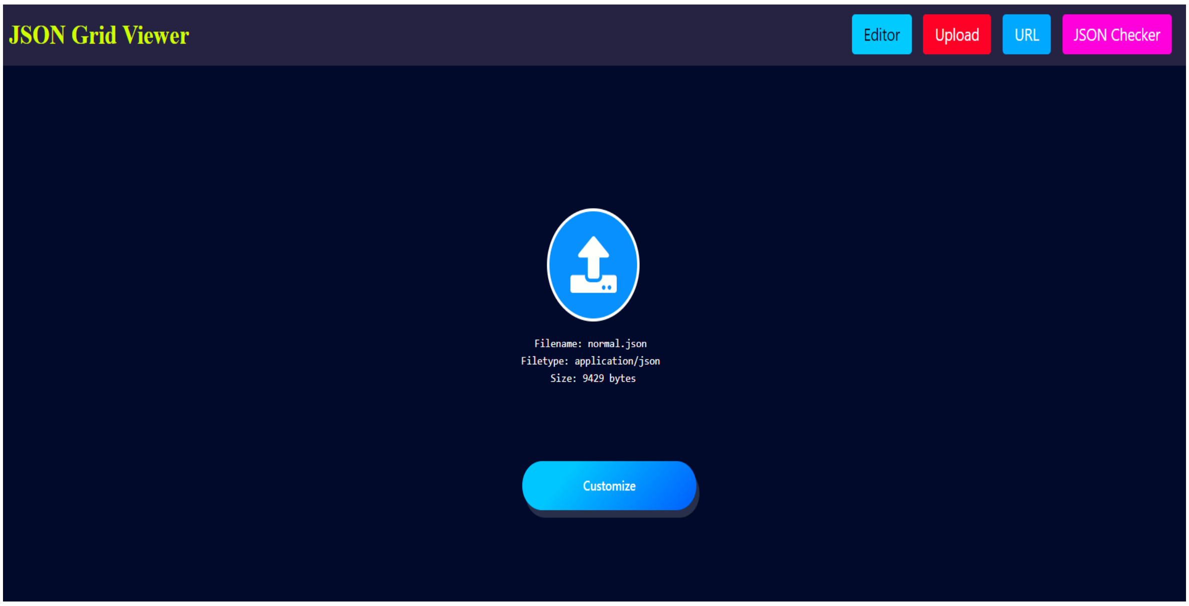


The User can input the data in any of the given below formats:

- JSON File
- URL containing JSON
- Raw JSON

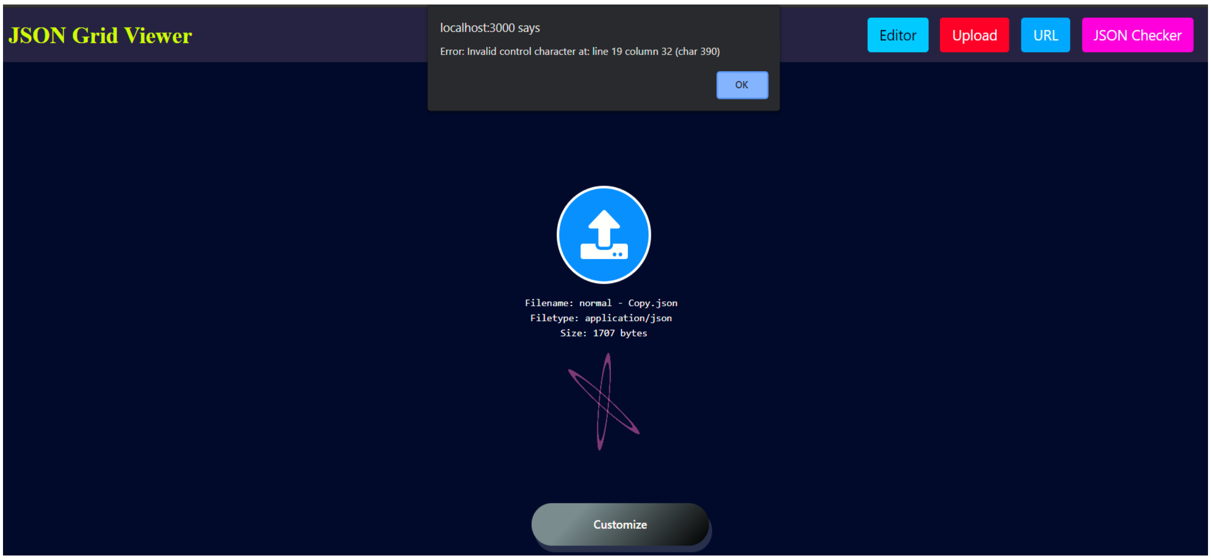
Users can also directly go to any page conversion page at any time using the buttons on the NavBar.

2.3.2.1 Upload the JSON File



```
Filename: normal.json
Filetype: application/json
Size: 9429 bytes
```

- ➢ The user can click the upload file button and choose the file to upload it.



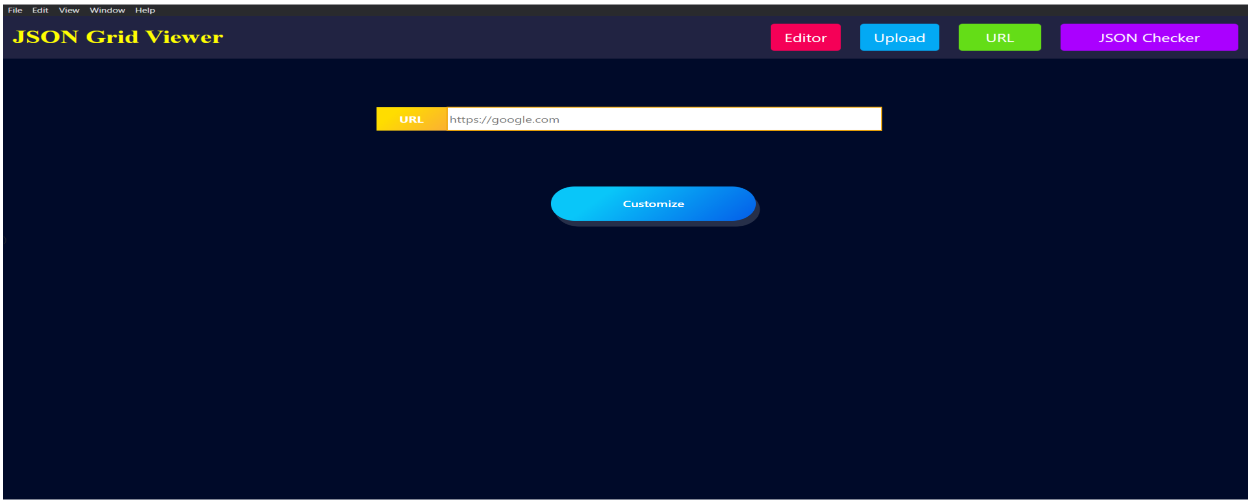
localhost:3000 says

Error: Invalid control character at: line 19 column 32 (char 390)

```
Filename: normal - Copy.json
Filetype: application/json
Size: 1707 bytes
```

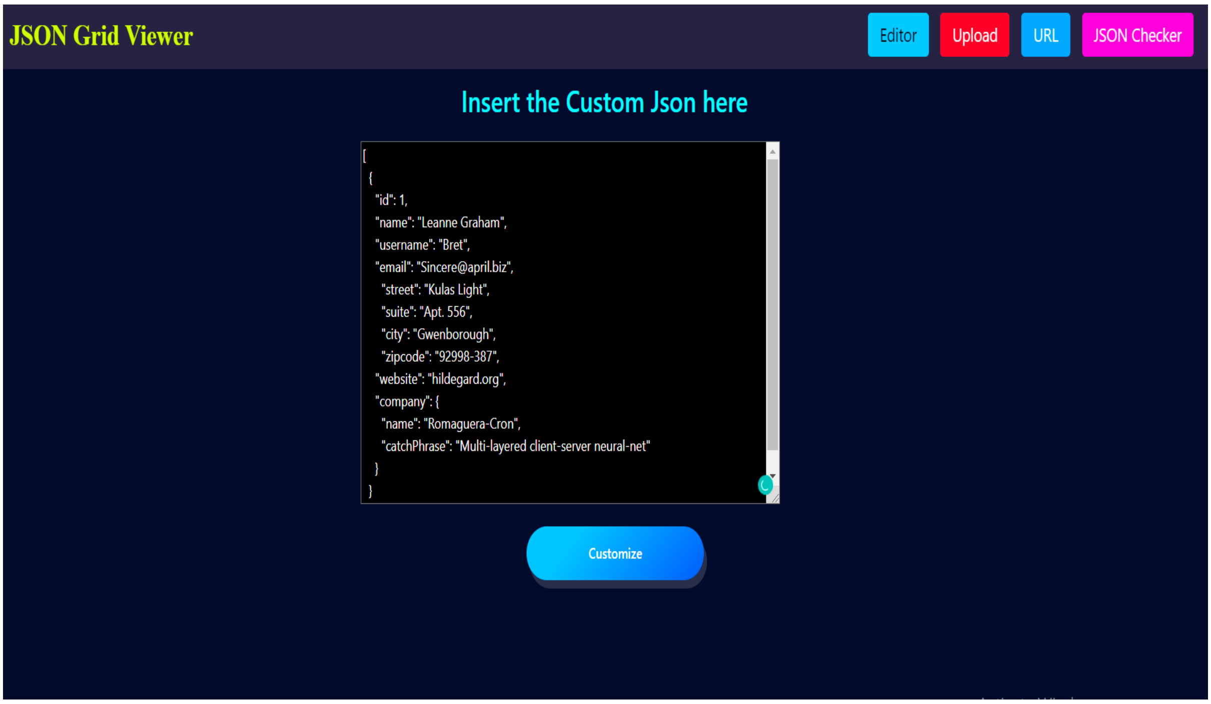
- ➢ If the file contains an invalid Json , an alert would pop up containing the line of error in the Json file.
- ➢ If the file is a valid JSON file, then after clicking the customize button, the user will be redirected to the customization page.

2.3.2.2. Insert URL containing JSON



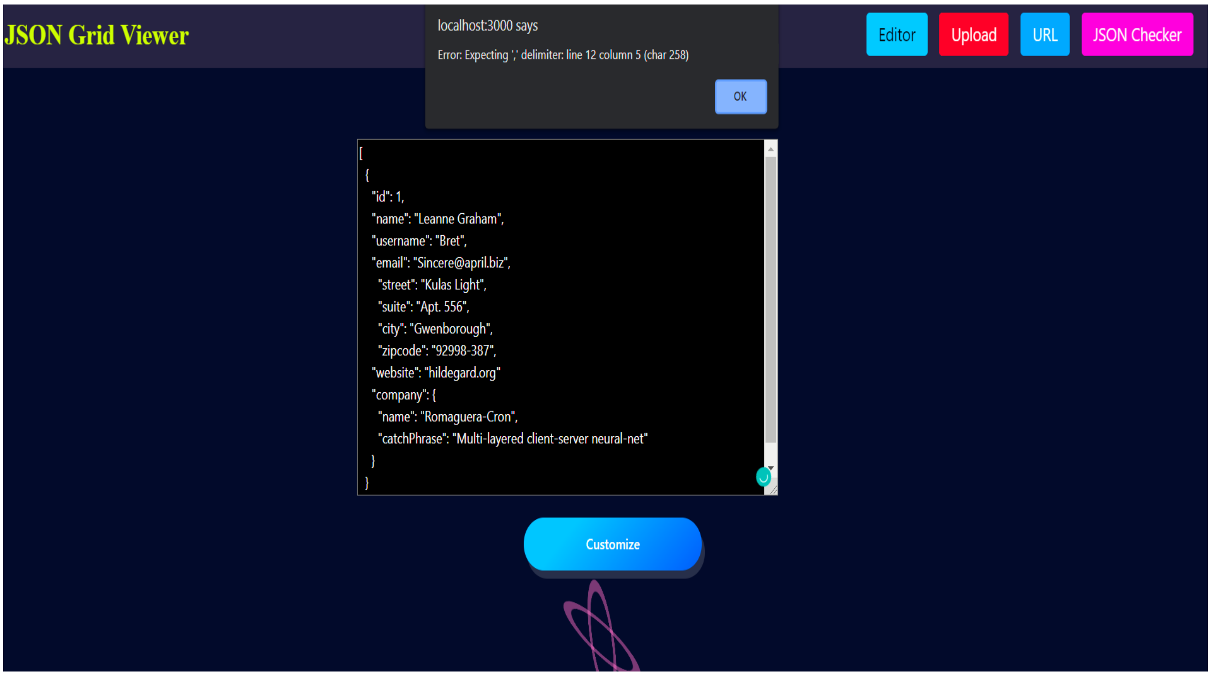
- ➢ The User can insert a URL/API containing JSON which is to be converted to Excel/CSV format.
- ➢ If the URL contains an invalid Json , an alert would pop up with an error message.
- ➢ If the URL contains a valid JSON file, then after clicking the customize button, the user will be redirected to the customization page.

2.3.2.3 Upload the JSON File



```
[
 {
  "id": 1,
  "name": "Leanne Graham",
  "username": "Bret",
  "email": "Sincere@april.biz",
   "street": "Kulas Light",
   "suite": "Apt. 556",
   "city": "Gwenborough",
   "zipcode": "92998-387",
  "website": "hildegard.org",
  "company": {
   "name": "Romaguera-Cron",
   "catchPhrase": "Multi-layered client-server neural-net"
  }
 }
```

- ➢ The users can input their own JSON file in the text editor provided.



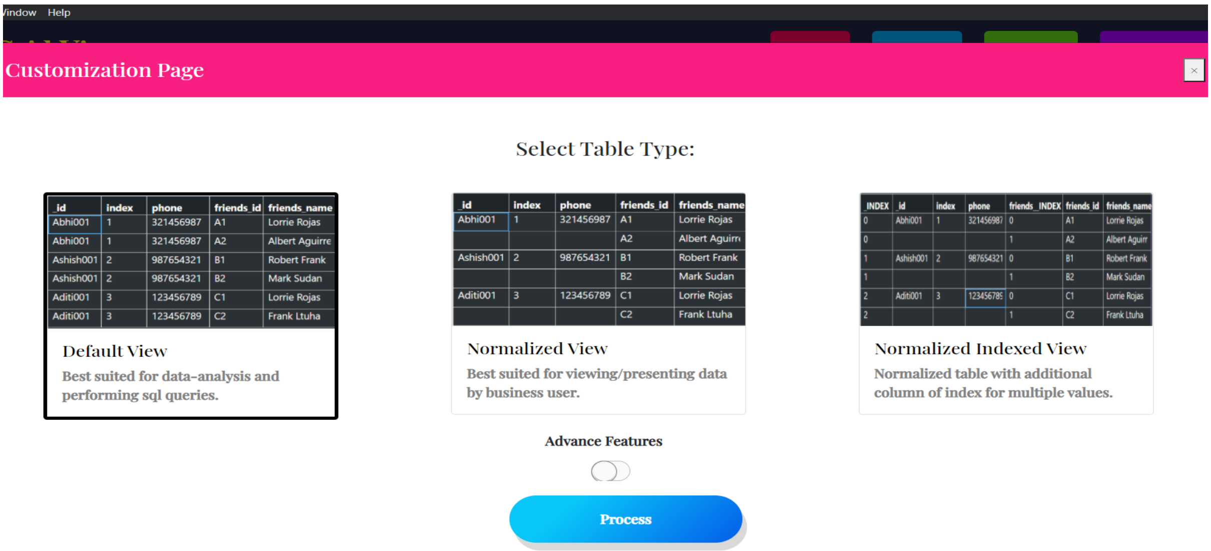
localhost:3000 says

Error: Expecting ',' delimiter: line 12 column 5 (char 258)

```
[
 {
  "id": 1,
  "name": "Leanne Graham",
  "username": "Bret",
  "email": "Sincere@april.biz",
    "street": "Kulas Light",
    "suite": "Apt. 556",
    "city": "Gwenborough",
    "zipcode": "92998-387",
  "website": "hildegard.org"
  "company": {
    "name": "Romaguera-Cron",
    "catchPhrase": "Multi-layered client-server neural-net"
  }
 }
```

- ➢ If the JSON is invalid , an alert would pop up with the line number and details about the error.
- ➢ If the JSON is valid , then after clicking the customize button, the user will be redirected to the customization page.

2.3.3 Customization Page



Select Table Type:

_id	index	phone	friends_id	friends_name
Abhi001	1	321456987	A1	Lorrie Rojas
Abhi001	1	321456987	A2	Albert Aguirre
Ashish001	2	987654321	B1	Robert Frank
Ashish001	2	987654321	B2	Mark Sudan
Aditi001	3	123456789	C1	Lorrie Rojas
Aditi001	3	123456789	C2	Frank Ltuha

Default View

Best suited for data-analysis and performing sql queries.

_id	index	phone	friends_id	friends_name
Abhi001	1	321456987	A1	Lorrie Rojas
			A2	Albert Aguirr
Ashish001	2	987654321	B1	Robert Frank
			B2	Mark Sudan
Aditi001	3	123456789	C1	Lorrie Rojas
			C2	Frank Ltuha

Normalized View

Best suited for viewing/presenting data by business user.

_INDEX	_id	index	phone	friends__INDEX	friends_id	friends_name
0	Abhi001	1	321456987	0	A1	Lorrie Rojas
0				1	A2	Albert Aguirr
1	Ashish001	2	987654321	0	B1	Robert Frank
1				1	B2	Mark Sudan
2	Aditi001	3	123456789	0	C1	Lorrie Rojas
2				1	C2	Frank Ltuha

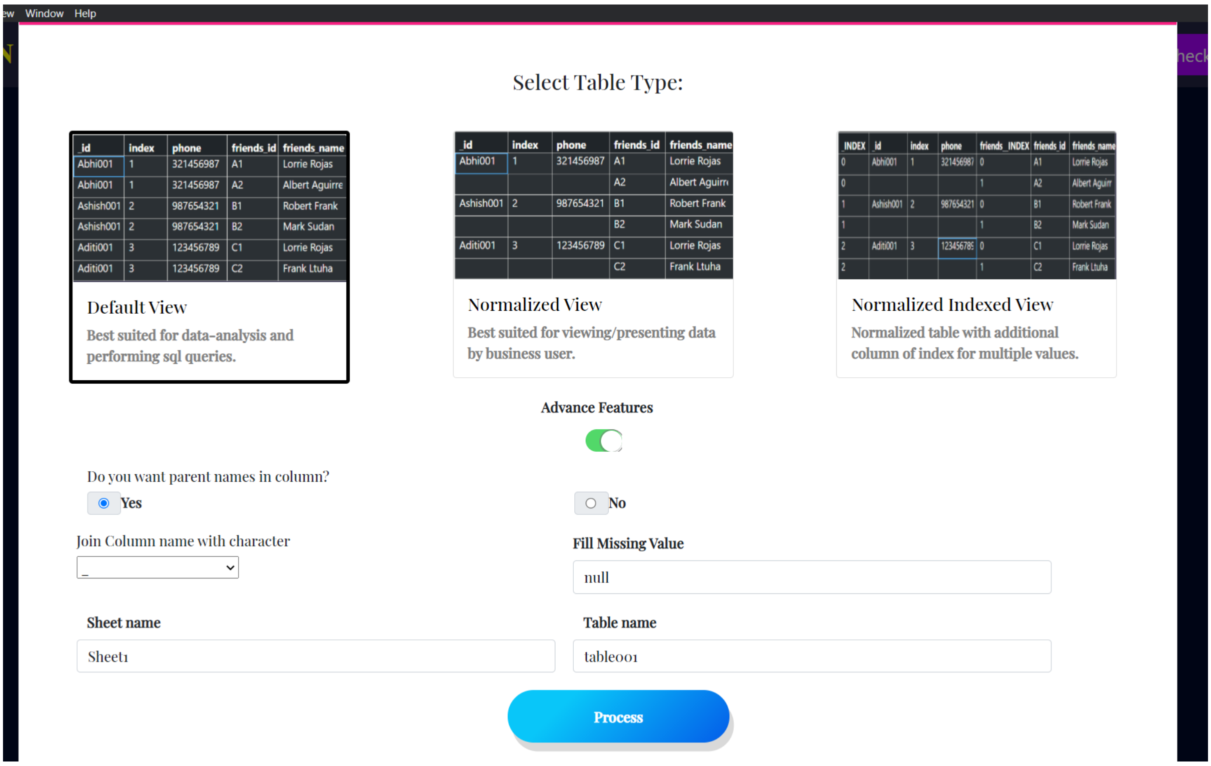
Normalized Indexed View

Normalized table with additional column of index for multiple values.

The customization page provides the user the flexibility to customize the tables before generating them and freely mend the table according to user-specific needs.

Customization page presents the user with **3 different types of views** for the same json, a brief description of the views are provided below

1. **Default view** : view containing no missing values, repeated data, and best for analysis or performing queries
2. **Normalized view** : view containing json-as-it-is, contains missing data, best for presentation and viewing in general
3. **Normalized indexed view** : adds extra Index columns in case records contain multiple records within them, helps to better locate the data, best for tasks similar to the Normalized view.



Select Table Type:

_id	index	phone	friends_id	friends_name
Abhi001	1	321456987	A1	Lorrie Rojas
Abhi001	1	321456987	A2	Albert Aguirre
Ashish001	2	987654321	B1	Robert Frank
Ashish001	2	987654321	B2	Mark Sudan
Aditi001	3	123456789	C1	Lorrie Rojas
Aditi001	3	123456789	C2	Frank Ltuha

Default View

Best suited for data-analysis and performing sql queries.

_id	index	phone	friends_id	friends_name
Abhi001	1	321456987	A1	Lorrie Rojas
			A2	Albert Aguirre
Ashish001	2	987654321	B1	Robert Frank
			B2	Mark Sudan
Aditi001	3	123456789	C1	Lorrie Rojas
			C2	Frank Ltuha

Normalized View

Best suited for viewing/presenting data by business user.

_INDEX	_id	index	phone	friends_INDEX	friends_id	friends_name
0	Abhi001	1	321456987	0	A1	Lorrie Rojas
0				1	A2	Albert Aguirr
1	Ashish001	2	987654321	0	B1	Robert Frank
1				1	B2	Mark Sudan
2	Aditi001	3	123456789	0	C1	Lorrie Rojas
2				1	C2	Frank Ltuha

Normalized Indexed View

Normalized table with additional column of index for multiple values.

Advance Features

Do you want parent names in column?

● Yes ○ No

Join Column name with character: _

Fill Missing Value: null

Sheet name: Sheet1

Table name: table001

The user can further toggle the advanced options to inculcate user-specific needs and generate views especially tailor-made for them.

A brief summary of the advanced features is given below :

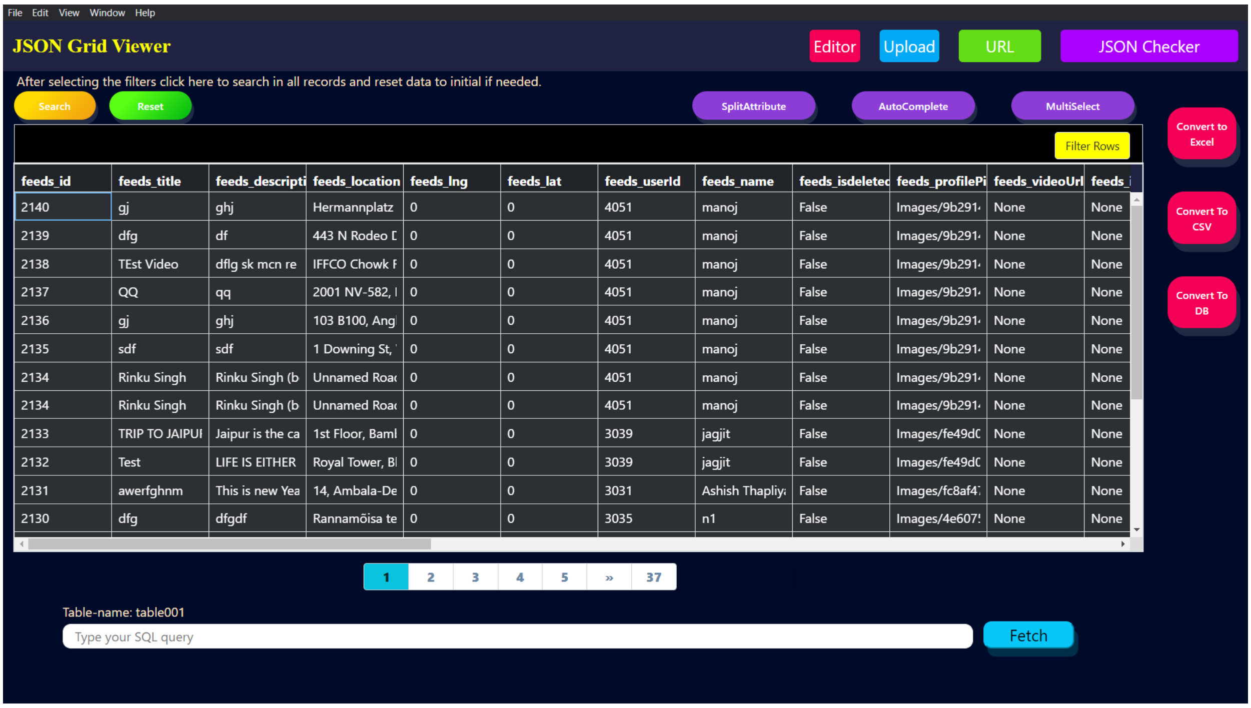
- **Do you want parent names in columns ?**
  Turning this to No could generate duplicate attribute names but provides shorter attribute names.
  If turned on, the attribute names contain the names of their parent joined to them using a joiner character (can be changed from Join Column name with character)

- **Join Column name with character :**
  Lets the user choose from a list of characters which would then be used to join parent-names to children-names for construction column-headers

- **Fill Missing Value :**
  Provides the user the freedom to input a string used for filling entries in view if json has missing data.

- **Sheet Name :**
  Let the user change the sheet name for the excel-file that can be generated later.
- **Table Name :**
  Name of the table for the SQL database, useful for performing sql queries on the preview page

2.3.4 Preview Page



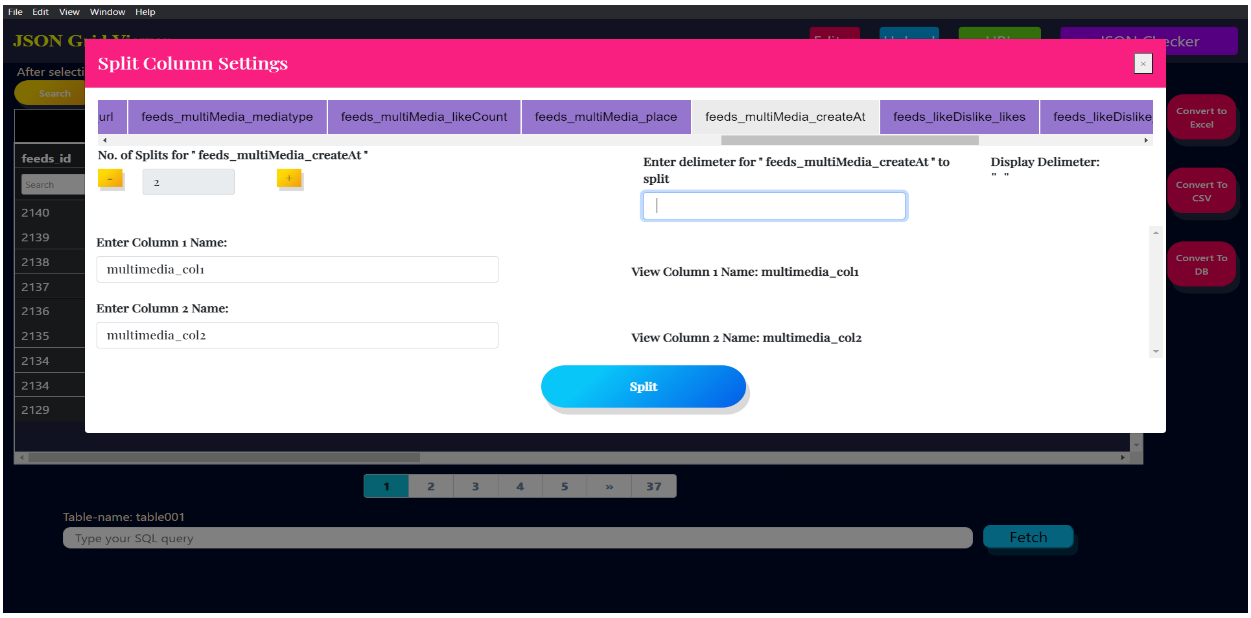
After selecting the filters click here to search in all records and reset data to initial if needed.

feeds_id	feeds_title	feeds_descripti	feeds_location	feeds_lng	feeds_lat	feeds_userId	feeds_name	feeds_isdeletec	feeds_profilePi	feeds_videoUrl	feeds_i
2140	gj	ghj	Hermannplatz	0	0	4051	manoj	False	Images/9b291·	None	None
2139	dfg	df	443 N Rodeo D	0	0	4051	manoj	False	Images/9b291·	None	None
2138	TEst Video	dflg sk mcn re	IFFCO Chowk F	0	0	4051	manoj	False	Images/9b291·	None	None
2137	QQ	qq	2001 NV-582, I	0	0	4051	manoj	False	Images/9b291·	None	None
2136	gj	ghj	103 B100, Ang	0	0	4051	manoj	False	Images/9b291·	None	None
2135	sdf	sdf	1 Downing St,	0	0	4051	manoj	False	Images/9b291·	None	None
2134	Rinku Singh	Rinku Singh (b	Unnamed Roac	0	0	4051	manoj	False	Images/9b291·	None	None
2134	Rinku Singh	Rinku Singh (b	Unnamed Roac	0	0	4051	manoj	False	Images/9b291·	None	None
2133	TRIP TO JAIPUI	Jaipur is the ca	1st Floor, Baml	0	0	3039	jagjit	False	Images/fe49dC	None	None
2132	Test	LIFE IS EITHER	Royal Tower, Bl	0	0	3039	jagjit	False	Images/fe49dC	None	None
2131	awerfghnm	This is new Yea	14, Ambala-De	0	0	3031	Ashish Thapliy	False	Images/fc8af47	None	None
2130	dfg	dfgdf	Rannamõisa te	0	0	3035	n1	False	Images/4e607!	None	None

Table-name: table001

Type your SQL query

- Excel Type functionalities are provided on this page. They are:
  - **Autocomplete Filter**: Automatic rendering of rows on the current page.
  - **MultiSelect Filter**: User can select values from the dropdown to perform analysis on data.
  - **Column Resizing** Property
- **Search Button** : Autocomplete Filter and MultiSelect Filter works only for the page selected. To apply these filters on all data records, user can use search button
- **Reset Button**: Users can use this button to reset preview data to original data without any filters.
- **Pagination** is added for fast processing and accessing of data records.
- Preview shown can be converted into excel file, csv file and db file.
- **SQL Queries** Users can also perform sql queries on the data records.



Split Column Settings

No. of Splits for " feeds_multiMedia_createAt " 2

Enter delimeter for " feeds_multiMedia_createAt " to split

Display Delimeter: " "

Enter Column 1 Name: multimedia_col1

View Column 1 Name: multimedia_col1

Enter Column 2 Name: multimedia_col2

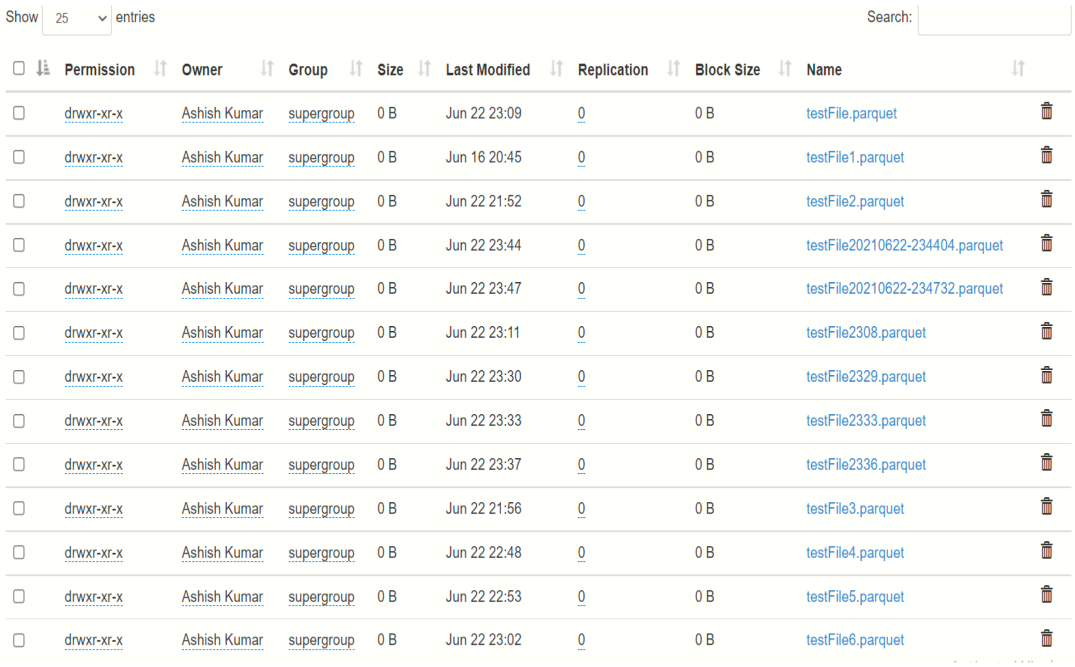
View Column 2 Name: multimedia_col2

Table-name: table001

Type your SQL query

➢ **Split Attribute:** User can choose the columns for which he/she wants to perform a split. After selecting the column, users can set the no. of splits and the delimiter with which they want to split the column. Users can also provide the names for the new columns formed.

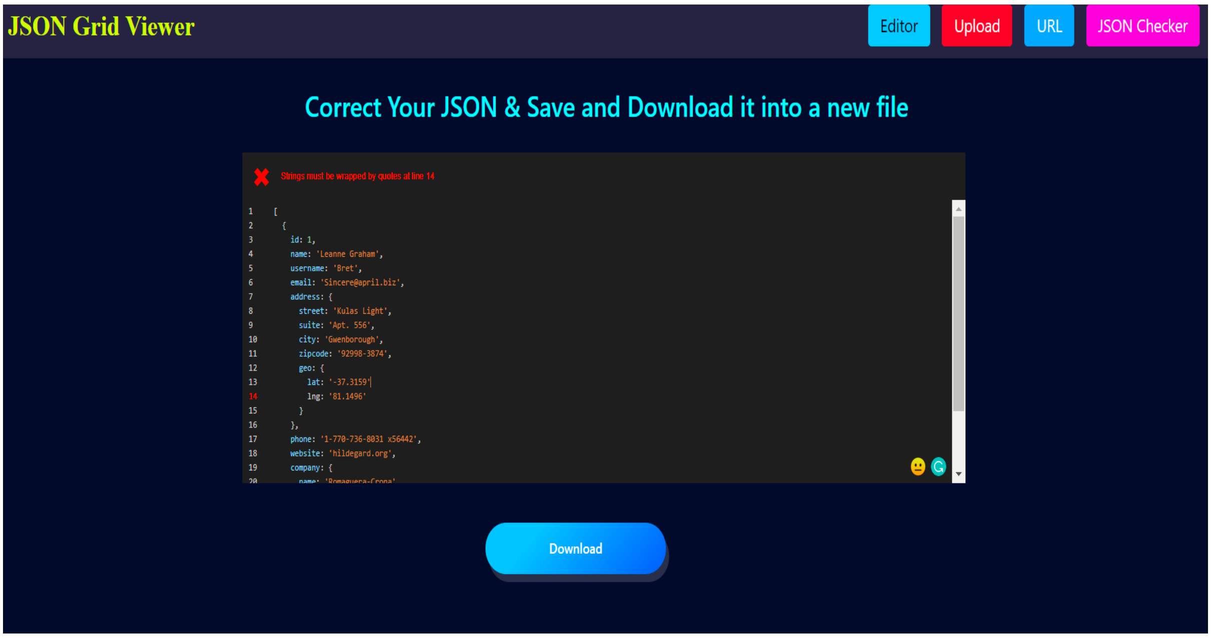
2.3.5 Saving files to HDFS



☐	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 23:09	0	0 B	testFile.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 16 20:45	0	0 B	testFile1.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 21:52	0	0 B	testFile2.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 23:44	0	0 B	testFile20210622-234404.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 23:47	0	0 B	testFile20210622-234732.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 23:11	0	0 B	testFile2308.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 23:30	0	0 B	testFile2329.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 23:33	0	0 B	testFile2333.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 23:37	0	0 B	testFile2336.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 21:56	0	0 B	testFile3.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 22:48	0	0 B	testFile4.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 22:53	0	0 B	testFile5.parquet	🗑
☐	drwxr-xr-x	Ashish Kumar	supergroup	0 B	Jun 22 23:02	0	0 B	testFile6.parquet	🗑

As soon as the user clicks the convert to DB button the data is saved into HDFS in parquet format and a db file is generated which can be downloaded.

2.3.6 Check Edit and Download the Custom JSON File



Correct Your JSON & Save and Download it into a new file

✖ Strings must be wrapped by quotes at line 14

```
1   [
2     {
3       id: 1,
4       name: 'Leanne Graham',
5       username: 'Bret',
6       email: 'Sincere@april.biz',
7       address: {
8         street: 'Kulas Light',
9         suite: 'Apt. 556',
10        city: 'Gwenborough',
11        zipcode: '92998-3874',
12        geo: {
13          lat: '-37.3159'
14          lng: '81.1496'
15        }
16      },
17      phone: '1-770-736-8031 x56442',
18      website: 'hildegard.org',
19      company: {
20        name: 'Romaguera-Crona'
```

The user can correct their invalid JSON using this JSON checker. If the JSON is correct then the user can download the JSON file. If the JSON is incorrect the user can correct it using the error details and line number provided by the editor.

3. Tech Stack Used:

- Python
- React JS
- PySpark
- SQLite
- Git

4. Installation Instructions

4.1 Installing Node

Step 1: Download Node.js Installer

In a web browser, navigate to https://nodejs.org/en/download/.
Click the Windows Installer button to download the latest default version. The Node.js installer includes the NPM package manager.

Step 2: Install Node.js and NPM from Browser

1. Once the installer finishes downloading, launch it.
Open the downloads link in your browser and click the file. Or, browse to the location where you have saved the file and double-click it to launch.
2. The system will ask if you want to run the software – click Run.

3. You will be welcomed to the Node.js Setup Wizard – click Next.
4. On the next screen, review the license agreement. Click Next if you agree to the terms and install the software.
5. The installer will prompt you for the installation location. Leave the default location, unless you have a specific need to install it somewhere else – then click next.
6. The wizard will let you select components to include or remove from the installation. Again, unless you have a specific need, accept the defaults by clicking Next.
7. Finally, click the Install button to run the installer. When it finishes, click Finish.

Step 3: Verify Installation
Open a command prompt (or PowerShell), and enter the following:
```
node -v
```
The system should display the Node.js version installed on your system.
```
npm -v
```
The system should display the npm version installed on your system.

4.2 Installing Python and Related Libraries

1. Open a browser window and navigate to the Python.org Downloads page for Windows.

2. Under the "Python Releases for Windows" heading, click the link for the Latest Python 3 Release - Python 3.8.4.

3. Scroll to the bottom and select either Windows x86-64 executable installer for 64-bit or Windows x86 executable installer for 32-bit.

4. Run the Python Installer once downloaded.

5. Please make sure to select the Install launcher for all users and Add Python 3.7 to PATH checkboxes.

6. Select Install Now – the recommended installation options.

7. Click on next and then wait for some times.It will get installed.

8. Open the Start menu and type "cmd."

9. Select the Command Prompt application.

10. Enter Python --ver to check the version of python installed

11. Enter pip -V in the console. (It would confirm if Pip and python were installed successfully or not)

12. Now open cmd again.

13. cd *path to extracted Json-Grid-View folder*

14. cd backend

15. pip install -r requirements.txt

4.3 Installing Hadoop

Prerequisites:

JAVA: You need to install the Java 8 package on your system.
HADOOP: You require the Hadoop 2.9.1 package.

Step 1. Download the hadoop 2.9.1 from the link provided below:
Hadoop Download Link:
https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-2.9.1/hadoop-2.9.1.tar.gz

Step 2. Create a folder path as below and copy the downloaded msi into this folder.
Path: 'C:/Hadoop/hadoop-2.9.1'

Step 3. Then download the windows compatible binaries from the git hub repo.
Link:- https://github.com/ParixitOdedara/Hadoop

Step 4. Extract the zip and copy all the files present under bin folder to C:\Hadoop\hadoop-

2.9.1\bin.

Replace the existing files as well.
Go to C:/Hadoop/hadoop-2.9.1 and create a folder 'data'.
Inside the 'data' folder create two folders 'datanode' and 'namenode'.

Step 5. Now setting up the Environment Variables for your Machine.
To set these variables, go to My Computer or This PC.
Right click --> Properties --> Advanced System settings --> Environment variables.
Click New to create a new environment variables.

Environment variables to be set:

HADOOP_HOME="C:\Hadoop\hadoop-2.9.1″
HADOOP_BIN="C:\Hadoop\hadoop-2.9.1\bin"
JAVA_HOME=<JDK installation location>"

Just to validate the above setting, open new cmd and check the output.
-- echo %HADOOP_HOME%

This should return "C:\Hadoop\hadoop-2.9.1".

-- echo %HADOOP_BIN%

This should return "C:\Hadoop\hadoop-2.9.1\bin".

To configure the hadoop on Windows10 we have to edit below mention files in the extracted location.

1. hadoop-env.cmd
2. core-site.xml
3. hdfs-site.xml
4. mapred-site.xml

Step 6. Edit hadoop-env.cmd

File location:- C:\Hadoop\hadoop-2.9.1\etc\hadoop\hadoop-env.cmd

Need to add:-

```
set HADOOP_PREFIX=%HADOOP_HOME%
set HADOOP_CONF_DIR=%HADOOP_PREFIX%\etc\hadoop
set YARN_CONF_DIR=%HADOOP_CONF_DIR%
set PATH=%PATH%;%HADOOP_PREFIX%\bin
```

Step 7. Edit core-site.xml

File Location:- C:\Hadoop\hadoop-2.9.1\etc\hadoop\core-site.xml

Need to add:-

(content within <configuration> </configuration> tags.)

```
 <configuration>
   <property>
     <name>fs.default.name</name>
     <value>hdfs://0.0.0.0:19000</value>
   </property>
</configuration>
```

Step 8. Edit hdfs-site.xml

File Location:- C:\Hadoop\hadoop-2.9.1\etc\hadoop\hdfs-site.xml.

Need to add:-

(below content within <configuration> </configuration> tags.)

```
 <configuration>
   <property>
         <name>dfs.replication</name>
         <value>1</value>
   </property>
   <property>
         <name>dfs.namenode.name.dir</name>
         <value>C:\Hadoop\hadoop-2.9.1\data\namenode</value>
   </property>
   <property>
         <name>dfs.datanode.data.dir</name>
         <value>C:\Hadoop\hadoop-2.9.1\data\datanode</value>
```

```
  </property>
</configuration>
```

Step 9. Edit mapred-site.xml

File location:- C:\Hadoop\hadoop-2.9.1\etc\hadoop\mapred-site.xml

Need to add:-

(below content within <configuration> </configuration> tags.
If you don’t see mapred-site.xml then open mapred-site.xml.template file
and rename it to mapred-site.xml)

```
 <configuration>
  <property>
        <name>mapreduce.job.user.name</name>
        <value>%USERNAME%</value>
  </property>
  <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
  </property>
  <property>
        <name>yarn.apps.stagingDir</name>
        <value>/user/%USERNAME%/staging</value>
  </property>
  <property>
        <name>mapreduce.jobtracker.address</name>
        <value>local</value>
  </property>
</configuration>
```

Step 10. Additional Configuration:-

Check if:

C:\Hadoop\hadoop-2.9.1\etc\hadoop\slaves file is present,
If that file is not available, create the file called slave and insert localhost.

Note:

One most common issue one can get is illegal character Exception.
This occurs when someone has a space in the name of their PC.
In this we need to open the hadoop-env.cmd and do the following changes.

File location:- C:\Hadoop\hadoop-2.9.1\etc\hadoop\hadoop-env.cmd

```
set HADOOP_IDENT_STRING="The name of your PC without Spacebar"
```

Step 11. Node formatting

To format the node, open the cmd and execute the below command:

```
--hadoop namenode -format
```

Step 12. To enable the hadoop open the CMD as Administrator and type below command.

-- start-all.cmd
It will open 4 new windows cmd terminals for 4 daemon processes, namely :
--namenode
--datanode
--nodemanager
--resourcemanager

-- To access Resource Manager go to http://localhost:8088 from your web browser.

-- To access Node Manager go to http://localhost:8042 from your web browser.

-- To access Name Node go to http://localhost:50070 from your web browser.

-- To access Data Node go to http://localhost:50075 from your web browser.

5. Future Scope

- Can be deployed over Openshift and can be used further in the future between teams.
- User Authentication can be done and it can be used at team level .
- Support different Operating Systems.
- Endpoints of this Application can be integrated with other Applications.