**APPROACH:-**

* The PDFs are converted to JSON Object using Extract API and Parsed to JS Object from which different fields are filtered.
* The PDFs Given in Test Dataset have a fixed Format.
* The Alignment from left margin is fixed for every Text element. Also most of the elements are being extracted in the same order everytime.
* So, For the elements that are easily and accurately detected by the Extract API, The code just iterates over the parsed Object and keeps on separating the respective fields as they appear in it. Fields Like Company Name, Company Address, Company Description, Invoice No, Invoice Issue Date, Item Details are Filtered in this way.
* For Rest of the elements, That are sometimes not detected distinctly and properly, The code traverses the parsed Object and based on the left Bound, groups the Text Elements. Then separate respective fields from these groups of data created.

**CODE BREAKDOWN AND WORKFLOW:-**

* **app.js :-** This is the main file from where execution starts. It first sets up the SDK using the credentials and then traverses each PDF. For Every PDF, It sends the request to Extract API and then stores the result in an array. Once all the PDFs are extracted, the filtering of fields starts.
* **extractBillDetails.js :-** This Module contains the code for filtering the fields from the object array of all PDFs. It traverses over the respective object of each PDF and first filters out the fields based on the Left Bounds and after that, filters out the fields based on their order in the Object.
* **Create\_CSV :-** This folder contains the modules to create the CSV File. The column.js file contains the detail about the columns Title in the CSV File and the create\_CSV.js file uses csv-writer node module to set up the csv writer and contains a function to add record in the file.
* Credentials Folder contains the SDK credentials files. config folder is for managing the default SDK logging. zip\_Files is for zip files created by the extract API for every PDF. InvoicesData Folder is the folder provided in question and ExtractedData Folder contains the output CSV file.

🡪 Make Sure the CSV file is not open anywhere while executing the code, else

it will land to a Read Write OS Conflict and **Give Error.** Also, Make Sure to

**install node modules** before running the code.

**MUST READ KEY POINTS :-**

* The request to API is sent at intervals of **2.5sec** each to manage the API Limit of 25req/min.
* All the API requests are being sent asynchronously at their time without waiting for previous requests to be completed. This helps in using the API Server to its fullest to process requests **Parallely** and makes the extraction process **as Fast as possible Given the Rate Limit.**

**Without API Rate Limit**, all the requests could be sent at once and thus, the process would complete within **few seconds** only.

The logs don’t show default info messages and show only **the progress of the extraction process**.

* The Output CSV file contains **blank cells for Invoice Tax of Some PDFs**. This is because the **EXTRACT API is not detecting the tax field in those PDFs.**

For the **empty Invoice\_Desc fields in some PDFs, It is empty in those PDF also**. Also, the date is in DD/MM/YYYY format but excel recognizes date in MM/DD/YYYY format. So, Not all dates are recognized if the CSV File is viewed in Excel.

So, The Code gives an **accurate result** and has been verified several times.