**Purpose:** To categorize a transaction history file by business type through user input and machine learning.

**Requirements:**

* User Input Requirements:
  + User must be able to upload a transaction history file in CSV format
    - Must be able to append to already loaded transactions, not replace.
  + Each user must have their own collection of transactions in mongoDB -> (Clarify)
  + User must be able to categorize the transactions by business type
  + Upon login, the user must be displayed all transactions they’ve uploaded so far
  + User’s transactions must be displayed in tabular format
  + User must be able to create new business types as they wish
  + Multiple transactions of identical business names are allowed, but only one categorization per UNIQUE business name must occur (the user should only be shown one (arbitrary) transaction of each business)
  + **Technical Requirements:**
    - Web-based application using:
      * Python (Flask)
      * Javascript (JS)
      * HTML5
      * MongoDB
* Machine Learning Requirements:
  + TO BE DEFINED

**Design:**

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Allows user to choose CSV file from local drive

Uploads CSV to mongoDB

Table

Description automatically generated

A dropdown menu used to categorize each transaction

Table created to display 10-20 entries at a time.

Button used to see previous entries

Button used to save categorizations of transactions (to mongoDB)

Button used to see next entries

Graphical user interface

Description automatically generated with medium confidence

Create new business type

Table

Description automatically generated

Enter new business type and click submit/ enter

Graphical user interface, application, table

Description automatically generated

Business type appears as category for all remaining transactions

Transaction classified as type ‘Restaurant’

**Logic:**

* On page load-up, JS scripts (both local->database\_connect.js and non-local->AJAX from googleAPI) are loaded in
* User (currently) uploads a CSV transaction file. On that event:
  + Frontend converts CSV file to JSON format
  + Call to backend is made with this JSON file as data packet
  + Backend connects to mongoDB database and adds all the transactions from the JSON into the collection
  + The backend queries and stores all the UNIQUE transactions from the collection
  + It then gives the frontend the first DISPLAY\_ENTRIES entries from the above grouping
* The frontend receives these entries and displays them in tabular format
  + For each transaction, a dropdown menu is newly rendered with the selections ‘Select’, ‘Add Category’, and any other categories the user may have added (the very first page that is displayed has only these 2 options)
* On the event the user selects ‘Add Category’:
  + A text box is displayed with the buttons ‘Cancel’ and ‘Submit’
  + User types a string, and this string is added as a selection to all dropdown menus (on all pages)
* On the event the user clicks next/prev:
  + A call to the backend is made requesting the previous/next DISPLAY\_ENTRIES entries
  + Backend creates a JSON of these DISPLAY\_ENTRIES entries and sends to the frontend for display
* On the event the user selects ‘Save Page’:
  + Frontend collects all transactions which have been categorized by the user and packs them into a JSON
  + This JSON file is sent to the backend
  + The backend finds ALL instances of each transaction and categorizes them with the corresponding value in the JSON

**Questions:**

* Should the user be shown all transactions from ALL upload files, only transactions that have not yet been categorized from ALL upload files, or all transactions from only the most recently uploaded CSV?
* Why should there be a distinction among users? (for machine learning purposes, we need as many transactions as possible)
* How will the machine learning aspect play into this? (does the model make a selection in the dropdown menu when it thinks it knows what the transaction is, or is the categorization made in mongoDB and the user doesn’t know about it at all??)

**TODO:**

* Separate collections for each user
* Prompt user for saving all changes before logging out????
* Should we automatically go to next page after user saves page?
* Query all unique documents which have not been categorized yet from mongoDB at the beginning and hold them in python. As frontend requests documents for display, read from there instead of querying again.
  + Right now, we query ALL documents from mongoDB and store them in the backend
* Prompt user to save changes before logging out/ deleting tab