**Course Modules Based Assignment Model**

The Course Modules Based Assignment Model aims at generating an assignment based on the completion of modules by the user. This will be available for the user after he completes a module in their course and wants to test their skill.

The Code is elaborated in an extensive way as follows:

**Input**

The code takes input in the form of a json file (ModuleInput.json). This file must contain every feature the user wants to put in, in a detailed format:-

{

"\_id":{"$oid":"5ecea1b690cfa44a44f83139"},

"user\_id":"1",

"firstName":"Student",

“course\_name”:”frontend”,

“subsection”:”1”,

“difficulty”:”easy”,

“no\_of\_mcq”:”10”

}

Since all the user data has been already gathered in previous iterations and at the start of the courses they completed, there is no need to get access to more data. This is done to reduce redundancy.

This whole process is autonomous. The data taken has to be in the form of json. It should include the course\_name, Subsection, difficulty and the NoOfMCQ.

This program was initially made in IPython compiler and for converting to a .py file a lot of dependencies have to be installed.

This procedure can be accessed directly from the user and the need for a json file to transfer data can be avoided.

**Code**

* Initialy the json file is loaded on tto the model
* Data is taken from the json input of User\_ID,CourseName,SubSection,Difficulty,NoOfMCQs.
* A list with skillcodes is created for faster referencing to the dataset
* The getmcq(no\_of\_mcq,subcourse\_code,difficulty,id1) function is called which take in account the no\_of\_mcq,subcourse\_code,difficulty the user wants.
* Data is imported into a dataframe (df) through Dataframe Methods present in the pandas library. This dataframe is further processed and filtered on the basis of difficulty and the subcourse. It shuffles randomly by the sample method provided by pandas.
* QIDs of all the questions are fetched onto a dataframe. Finally, this dataframe is converted into json format.

**Output**

The final dataframe is then converted to json format and saved into a separate file with the name as 1cmb.json

Example output

{"QID":

{ "0":"1PYM","1":"1PYM",

"2":"21PYM","3":"12PYM",

"4":"21PYM","5":"21PYM",

"6":"1PYM","7":"21PYM",

"8":"21PYM","9":"21PYM"

}

}

\*Previous Json ouput had two coloums which not only adds significant overhead over data transfer but also is useless in cases of structured databases.