**LOW LEVEL DOCUMENTATION**

**MISSING COMPONENTS REPORT**

BHAVYA JS

1.creates DB connection.

client = MongoClient("localhost", port=27017)  
dbname = config.dbname  
collectionname=config.missingreportcn

2.Function to get reports and stores as list.

def getallreports():  
  
 *"""This function fetches masterinventory and crossreference  
 from DB and stores as list."""* colname\_master = config.materinventorycn  
 colname\_xref=config.crossreferencecn  
 col\_master = client[dbname][colname\_master]  
 col\_xref=client[dbname][colname\_xref]  
  
 masterinventory = list(col\_master.find({'type': {"$ne": "metadata"}},{"component\_name":1,"component\_type":1, "\_id":0}))  
 crossref = list(col\_xref.find({'type': {"$ne": "metadata"}},{"called\_name":1,"called\_type":1,"\_id":0}))  
  
 return Missingreport(masterinventory,crossref)

3. Function to store the ‘called name’ and ‘called type’ in separate set named master, stores ‘component name’ and ‘component type’ in a set named Xref. And compares the sets and adds it to list that are not found.

def Missingreport(masterinventory,crossref):  
  
 *"""This function compares masterinventory with crossreference  
 and finds the missing data and stores it in list. """* Missingreports=[]  
 master = set()  
 Xref = set()  
 for i in masterinventory:  
 a = i["component\_name"]  
 a = a.split(".")[0]  
 master.add(a + "," + i["component\_type"])  
 #print(master)  
 for j in crossref:  
 Xref.add(j["called\_name"] + "," + j["called\_type"])  
 #print(Xref)  
  
 for item in Xref:  
 #print(item)  
 if item not in master:  
 Missingreports.append(item)  
 #print(Missingreports)  
 #print(Missingreports)  
 return Missingreport\_json(Missingreports)

4. Function to create a json for missing reports.

def Missingreport\_json(Missingreports):  
 *"""This function stores all missing reports in a dictionary"""* MissingreportJson=[]  
 for item in Missingreports:  
 MissingreportDict={ }  
 called\_name = item.split(",")[0]  
 called\_type = item.split(",")[1]  
  
 MissingreportDict["component\_name"] = called\_name  
 MissingreportDict["component\_type"] = called\_type  
 #print(MissingreportDict)  
 MissingreportJson.append(MissingreportDict)  
 return MissingreportJson

5.Function to insert the created json to database.

def Missingreport\_json(Missingreports):  
 *"""This function stores all missing reports in a dictionary"""* MissingreportJson=[]  
 for item in Missingreports:  
 MissingreportDict={ }  
 called\_name = item.split(",")[0]  
 called\_type = item.split(",")[1]  
  
 MissingreportDict["component\_name"] = called\_name  
 MissingreportDict["component\_type"] = called\_type  
 #print(MissingreportDict)  
 MissingreportJson.append(MissingreportDict)  
 return MissingreportJson