**WEB APPLICATION TO UPLOAD CSV FILE**

**CONTENTS:**

* The Problem
* Steps
* Requirements
* Code Explanation
* Full code
* Conclusion

**THE PROBLEM:**

Create a web application with an option a csv file of 1 lakh or more records,where the application should read the records ,process it,and push to the database table.

**STEPS:**

* Create file upload form
* Upload the CSV using Flask
* Parse CSV file data
* Connect to the database
* Insert rows into a specific table in the database

**STACK:**

**Flask;**

A Python Minimal (Micro) web frameworks.

**MySQL;**

If Windows use WAMP or XAMPP,

If Linux use XAMPP ,

If macOS use XAMPP or MAMP

**Pandas;**

Pandas is an open-source python library for data analytics and manipulation. It's widely used among data scientists and data engineers.

**mysql.connector;**

The library that we will use to connect to the MySQL database and insert the data.

**CODE EXPLANATION:**

**Install Flask;**

# install Python directly with pip.

pip install Flask

**Install XAMPP;**

#The XAMPP package will include MySQL database and PHPMyAdmin to manage our MySQL databases.

/#Install the required libraries (Pandas, sql.connector)

pip install pandas

pip install sql.connector

#A sample CSV file, I will be using addresses.csv from here.

#Now let's test if Flask works by starting to create our main file main.py

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route("/")

def hello():

     return "Flask CSV Filer Uploader and Parser"

if (\_\_name\_\_ == "\_\_main\_\_"):

     app.run(port = 5000)

#We can run the file by $ python main.py then heading tohttp://[127.0.0.1:5000/](http://127.0.0.1:5000/).

#If everything goes as planned, you will get the rendered message.

#Next step is to create the HTML page which will be responsible for uploading CSVfiles.

**1.Create file upload form:**

#Let's create templates directory and our HTML upload form in it.

<!doctype html>

<html>

  <head>

    <title>FLASK CSV File Upload</title>

  </head>

  <body>

    <h1>Upload your CSV file</h1>

    <form method="POST" action="" enctype="multipart/form-data">

      <p><input type="file" name="file"></p>

      <p><input type="submit" value="Submit"></p>

    </form>

  </body>

</html>

**2.Upload the files to the server:**

#add the reasonable function in main.py

from flask import Flask, render\_template, request, redirect, url\_for

import os

from os.path import join, dirname, realpath

app = Flask(\_\_name\_\_)

# enable debugging mode

app.config["DEBUG"] = True

# Upload folder

UPLOAD\_FOLDER = 'static/files'

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

# Root URL

@app.route('/')

def index():

     # Set The upload HTML template '\templates\index.html'

    return render\_template('index.html')

# Get the uploaded files

@app.route("/", methods=['POST'])

def uploadFiles():

      # get the uploaded file

      uploaded\_file = request.files['file']

      if uploaded\_file.filename != '':

           file\_path = os.path.join(app.config['UPLOAD\_FOLDER'], uploaded\_file.filename)

          # set the file path

           uploaded\_file.save(file\_path)

          # save the file

      return redirect(url\_for('index'))

if (\_\_name\_\_ == "\_\_main\_\_"):

     app.run(port = 5000)

#As everything is set, test the file upload now at <http://127.0.0.1:5000/>

#see the uploaded files at /static/files/

**3.Parse CSV file:**

#To parse CSV file data into rows,use pandas the data analysis manipulation library.

#Import the library

import pandas as pd

#Parse CSV function

def parseCSV(filePath):

      # CVS Column Names

      col\_names = ['first\_name','last\_name','address', 'street', 'state' , 'zip']

      # Use Pandas to parse the CSV file

      csvData = pd.read\_csv(filePath,names=col\_names, header=None)

      # Loop through the Rows

      for i,row in csvData.iterrows():

             print(i,row['first\_name'],row['last\_name'],row['address'],row['street'],row['state'],row['zip'],)

Don't forget to add parseCSV(file\_path) function soon after the file is saved in uploadFiles

**4. Connect to the database (XAMPP/ MySQL):**

#Add import mysql.connector

#Connect to the MySQL Server

import mysql.connector

mydb = mysql.connector.connect(

  host="localhost",

  user="root",

  password="",

  database="databaseName"

)

mycursor = mydb.cursor()

mycursor.execute("SHOW DATABASES")

# List All Databases

for x in mycursor:

  print(x)

**5.Insert rows into the database**

def parseCSV(filePath):

      # CVS Column Names

      col\_names = ['first\_name','last\_name','address', 'street', 'state' , 'zip']

      # Use Pandas to parse the CSV file

      csvData = pd.read\_csv(filePath,names=col\_names, header=None)

      # Loop through the Rows

      for i,row in csvData.iterrows():

             sql = "INSERT INTO addresses (first\_name, last\_name, address, street, state, zip) VALUES (%s, %s, %s, %s, %s, %s)"

             value = (row['first\_name'],row['last\_name'],row['address'],row['street'],row['state'],str(row['zip']))

             mycursor.execute(sql, value, if\_exists='append')

             mydb.commit()

             print(i,row['first\_name'],row['last\_name'],row['address'],row['street'],row['state'],row['zip'])

#check the new data entries at the database.

**FULL CODE:**

#main.py

from flask import Flask, render\_template, request, redirect, url\_for

import os

from os.path import join, dirname, realpath

import pandas as pd

import mysql.connector

app = Flask(\_\_name\_\_)

# enable debugging mode

app.config["DEBUG"] = True

# Upload folder

UPLOAD\_FOLDER = 'static/files'

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

# Database

mydb = mysql.connector.connect(

  host="localhost",

  user="root",

  password="",

  database="csvdata"

)

mycursor = mydb.cursor()

mycursor.execute("SHOW DATABASES")

# View All Database

for x in mycursor:

  print(x)

# Root URL

@app.route('/')

def index():

     # Set The upload HTML template '\templates\index.html'

    return render\_template('index.html')

# Get the uploaded files

@app.route("/", methods=['POST'])

def uploadFiles():

      # get the uploaded file

      uploaded\_file = request.files['file']

      if uploaded\_file.filename != '':

           file\_path = os.path.join(app.config['UPLOAD\_FOLDER'], uploaded\_file.filename)

          # set the file path

           uploaded\_file.save(file\_path)

           parseCSV(file\_path)

          # save the file

      return redirect(url\_for('index'))

def parseCSV(filePath):

      # CVS Column Names

      col\_names = ['first\_name','last\_name','address', 'street', 'state' , 'zip']

      # Use Pandas to parse the CSV file

      csvData = pd.read\_csv(filePath,names=col\_names, header=None)

      # Loop through the Rows

      for i,row in csvData.iterrows():

             sql = "INSERT INTO addresses (first\_name, last\_name, address, street, state, zip) VALUES (%s, %s, %s, %s, %s, %s)"

             value = (row['first\_name'],row['last\_name'],row['address'],row['street'],row['state'],str(row['zip']))

             mycursor.execute(sql, value, if\_exists='append')

             mydb.commit()

             print(i,row['first\_name'],row['last\_name'],row['address'],row['street'],row['state'],row['zip'])

if (\_\_name\_\_ == "\_\_main\_\_"):

     app.run(port = 5000)

**CONCLUSION:**

Thus a web application is created with an option a csv file of 1 lakh or more records,where the application should read the records ,process it,and push to the database table.