# D. student_marks_db_system

March 21, 2024

### 0.0.1 Simple Student Marks DB system using Python and MySQL

By Karthik Nair

[2]:
```
import mysql.connector
import tkinter as tk
from tkinter import *
from tkinter import Tk, Button, messagebox
from tkinter.filedialog import askopenfilename
import csv

def connect_to_mysql():
    try:
        return mysql.connector.connect(
            host="localhost",
            user="root",
            password=",428A3B2UDpY",
        )
    except mysql.connector.Error as err:
        messagebox.showerror("Error", f"Failed to connect to MySQL: {err}")
        return None

def create_database(cursor):
    try:
        cursor.execute("CREATE DATABASE IF NOT EXISTS students")
        messagebox.showinfo("Success", "Database 'students' created␣
 ↪successfully!")
    except mysql.connector.Error as err:
        messagebox.showerror("Error", f"Failed to create database: {err}")

def create_table(cursor):
    try:
        cursor.execute("CREATE TABLE IF NOT EXISTS student_credentials␣
 ↪(student_id INT PRIMARY KEY, name VARCHAR(255), password VARCHAR(255))")
        messagebox.showinfo("Success", "Table 'student_credentials' created␣
 ↪successfully!")
    except mysql.connector.Error as err:
        messagebox.showerror("Error", f"Failed to create table: {err}")
```

```
def push_to_db():
    filename = askopenfilename(title="Select CSV file", filetypes=[("CSV␣
 ↪files", "*.csv")])
    if filename:
        try:
            connection = connect_to_mysql()
            if connection:
                cursor = connection.cursor()
                create_database(cursor)
                connection.database = "students"
                create_table(cursor)
                with open(filename, newline='') as file:
                    reader = csv.DictReader(file)
                    for row in reader:
                        try:
                            cursor.execute("INSERT INTO student_credentials␣
 ↪(student_id, name, password) VALUES (%s, %s, %s)",
                                           (row['student_id'], row['name'],␣
 ↪row['password']))
                        except mysql.connector.Error as err:
                            messagebox.showerror("Error", f"Failed to insert␣
 ↪data: {err}")
                            break
                connection.commit()
                cursor.close()
                connection.close()
                messagebox.showinfo("Success", "Data inserted successfully!")
        except Exception as e:
            messagebox.showerror("Error", f"An error occurred: {e}")

root = tk.Tk()
root.title("Push Students' Credentials to DB")
root.geometry("400x100")

frame = tk.Frame(root)
frame.place(relx=0.5, rely=0.5, anchor="center")  # Placing the frame in the␣
 ↪center
# relx and rely are relative x and y coordinates of the center of the window

push_button = tk.Button(frame, text="Push Students' Credentials to DB",␣
 ↪command=push_to_db)
push_button.pack() # pack() is used to display the button in the frame

quit_button = tk.Button(frame, text="Quit", command=root.destroy)
quit_button.pack(pady=5)
```

```
root.mainloop()
```

[3]:
```
import pandas as pd
from tkinter import *
from tkinter import filedialog
from tkinter import messagebox
from tkinter import ttk
import os


def fill_zeros(filename, new_filename):
    try:
        df = pd.read_csv(filename, header=None)
        df.fillna(0, inplace=True)
        df.to_csv(new_filename, index=False, header=False)
        messagebox.showinfo("Success", "Operation completed successfully!")
    except Exception as e:
        messagebox.showerror("Failed", f"An error occurred: {str(e)}")


def choose_file(entry_widget):
    filename = filedialog.askopenfilename()
    entry_widget.delete(0, END)
    entry_widget.insert(0, filename)


def choose_folder(entry_widget):
    foldername = filedialog.askdirectory()
    entry_widget.delete(0, END)
    entry_widget.insert(0, foldername)


def main():
    root = Tk()
    root.title("Fill Zeros")
    root.resizable(False, False)

    frm = ttk.Frame(root, padding=10)
    frm.grid(row=0, column=0)

    Label(frm, text="Select marks file:").grid(row=0, column=0, sticky=W)
    filename_entry = Entry(frm)
    filename_entry.grid(row=0, column=1)
    Button(frm, text="Browse", command=lambda: choose_file(filename_entry)).
 ↪grid(row=0, column=2)
```

```
    Label(frm, text="Choose the output folder:").grid(row=1, column=0, sticky=W)
    folder_entry = Entry(frm)
    folder_entry.grid(row=1, column=1)
    Button(frm, text="Browse", command=lambda: choose_folder(folder_entry)).
 ↪grid(row=1, column=2)

    Label(frm, text="Enter updated file name:").grid(row=2, column=0, sticky=W)
    new_filename_entry = Entry(frm)
    new_filename_entry.grid(row=2, column=1)

    Button(frm, text="Fill Zeros", command=lambda: fill_zeros(filename_entry.
 ↪get(), os.path.join(folder_entry.get(), new_filename_entry.get()))).
 ↪grid(row=4, column=1)

    Button(frm, text="Quit", command=root.destroy).grid(row=5, column=1)

    root.mainloop()


if __name__ == "__main__":
    main()
```

[4]:
```
import mysql.connector
from tkinter import Tk, Button, messagebox
from tkinter.filedialog import askopenfilename
import csv

def connect_to_mysql():
    try:
        return mysql.connector.connect(
            host="localhost",
            user="root",
            password=",428A3B2UDpY",
            database="students"
        )
    except mysql.connector.Error as err:
        messagebox.showerror("Error", f"Failed to connect to MySQL: {err}")
        return None

def create_table(cursor):
    try:
        cursor.execute("CREATE TABLE IF NOT EXISTS student_marks (student_id␣
 ↪INT PRIMARY KEY, marks1 INT, marks2 INT, marks3 INT, total_marks INT)")
        messagebox.showinfo("Success", "Table 'student_marks' created␣
 ↪successfully!")
    except mysql.connector.Error as err:
        messagebox.showerror("Error", f"Failed to create table: {err}")
```

```
def calculate_total_marks(row):
    marks1 = int(row['marks1']) if row['marks1'] else 0
    marks2 = int(row['marks2']) if row['marks2'] else 0
    marks3 = int(row['marks3']) if row['marks3'] else 0
    return marks1 + marks2 + marks3

def push_to_mysql():
    filename = askopenfilename(title="Select CSV file", filetypes=[("CSV␣
 ↪files", "*.csv")])
    if filename:
        try:
            connection = connect_to_mysql()
            if connection:
                cursor = connection.cursor()
                create_table(cursor)
                with open(filename, newline='') as file:
                    reader = csv.DictReader(file)
                    for row in reader:
                        try:
                            total_marks = calculate_total_marks(row)
                            cursor.execute("INSERT INTO student_marks␣
 ↪(student_id, marks1, marks2, marks3, total_marks) VALUES (%s, %s, %s, %s,␣
 ↪%s)",
                                           (row['student_id'], row['marks1'],␣
 ↪row['marks2'], row['marks3'], total_marks))
                        except mysql.connector.Error as err:
                            messagebox.showerror("Error", f"Failed to insert␣
 ↪data: {err}")
                            break
                connection.commit()
                cursor.close()
                connection.close()
                messagebox.showinfo("Success", "Data inserted successfully!")
        except Exception as e:
            messagebox.showerror("Error", f"An error occurred: {e}")

root = tk.Tk()
root.title("Push Data to MySQL")
root.geometry("400x100")

frame = tk.Frame(root)
frame.place(relx=0.5, rely=0.5, anchor="center")  # Placing the frame in the␣
 ↪center

push_button = tk.Button(frame, text="Push Data to MySQL", command=push_to_mysql)
push_button.pack()
```

```
quit_button = tk.Button(frame, text="Quit", command=root.destroy)
quit_button.pack(pady=5)

root.mainloop()
```

[5]:
```
import mysql.connector
from tkinter import Tk, Label, Entry, Button, messagebox

def connect_to_mysql():
    try:
        return mysql.connector.connect(
            host="localhost",
            user="root",
            password=",428A3B2UDpY",
            database="students"
        )
    except mysql.connector.Error as err:
        messagebox.showerror("Error", f"Failed to connect to MySQL: {err}")
        return None

def authenticate_student(student_id, password):
    try:
        connection = connect_to_mysql()
        if connection:
            cursor = connection.cursor()
            cursor.execute("SELECT * FROM student_credentials WHERE student_id␣
 ↪= %s AND password = %s", (student_id, password))
            student = cursor.fetchone()
            cursor.close()
            connection.close()
            return student
        else:
            return None
    except Exception as e:
        messagebox.showerror("Error", f"An error occurred: {e}")
        return None

def get_student_marks(student_id):
    try:
        connection = connect_to_mysql()
        if connection:
            cursor = connection.cursor()
            cursor.execute("SELECT * FROM student_marks WHERE student_id = %s",␣
 ↪(student_id,))
            student_marks = cursor.fetchone()
            cursor.close()
```

```
            connection.close()
            return student_marks
        else:
            return None
    except Exception as e:
        messagebox.showerror("Error", f"An error occurred: {e}")
        return None


def login():
    student_id = student_id_entry.get()
    password = password_entry.get()
    student = authenticate_student(student_id, password)
    if student:
        student_marks = get_student_marks(student_id)
        if student_marks:
            messagebox.showinfo("Success", f"Welcome {student[1]}!\nYour marks␣
 ↪are:\nMarks 1: {student_marks[1]}\nMarks 2: {student_marks[2]}\nMarks 3:␣
 ↪{student_marks[3]}\nTotal Marks: {student_marks[4]}")
        else:
            messagebox.showerror("Error", "Failed to retrieve student marks")
    else:
        messagebox.showerror("Error", "Invalid student ID or password")


root = tk.Tk()
root.title("Student Login")
root.geometry("300x150")

frame = tk.Frame(root)
frame.place(relx=0.5, rely=0.5, anchor="center")  # Placing the frame in the␣
 ↪center

student_id_label = Label(frame, text="Student ID:")
student_id_label.grid(row=0, column=0)
student_id_entry = Entry(frame)
student_id_entry.grid(row=0, column=1)

password_label = Label(frame, text="Password:")
password_label.grid(row=1, column=0)
password_entry = Entry(frame, show="*")
password_entry.grid(row=1, column=1)

login_button = Button(frame, text="Login", command=login)
login_button.grid(row=2, columnspan=2, pady=10)

quit_button = Button(frame, text="Quit", command=root.destroy)
```

```
quit_button.grid(row=3, columnspan=2, pady=5)

root.mainloop()
```