

Project

Shahin Badhuru Nisha M

shahinbadhurnisha@gmail.com

1. Election 2024

```
import smtplib import
datetime

import mysql.connector

mydb=mysql.connector.connect(
    host="localhost", user="root",
    password="12345",
    database="election_2024"
)

def vote():
    print("***election 2024***")
    candidates=["aki","pm","azh"] print("press
    1-> aki")
    print("press 2-> pm")
    print("press 3-> azh") try:
        user_input=input("enter the number you want to vote for:") user=int(user_input)
        if user==1: candidate_voted="aki"
        elif user==2:
            candidate_voted="pm" elif
        user==3:
            candidate_voted="azh"
```

```

else:

    print("please type 1 , 2 , 3 only!") return

email=input("enter your email-id to receive voting details:") x=datetime.datetime.now()

date_time=x.strftime("%d/%B/%Y  %H:/%M:/%S")

mycursor=mydb.cursor()

sql="insert into votes(user_email, candidate_voted, vote_time) values (%s, %s, now())"

mycursor.execute(sql,(email,candidate_voted)) sql="update votes set

vote_count = vote_count +1 where
candidate_voted=%s"

mycursor.execute(sql,(candidate_voted,)) mydb.commit()

send_email(email,candidate_voted)

sql="select candidate_voted, count(*) as vote_count from votes group by candidate_voted"

mycursor.execute(sql)

results=mycursor.fetchall() print("current
vote counts:") for row in results:

    print(f"{row[0]}:{row[1]}") mycursor.close()

mydb.close() except

ValueError:

    print("dont use the name , please type numbers only...") except

mysql.connector.Error as err:

    print(f"database error: {err}")

def send_email(recipient_email,candidate_voted):

    sender_email="sender_email" password="sender_password"

    message=f"you voted for {candidate_voted} in election 2024. this is a confirmation email for your
vote."

    try:

```

```
with smtplib.SMTP('smtp.gmail.com',587) as s: s.starttls()
```

```
s.login("sender_email","sender_password")
```

```
s.sendmail(sender_email,recipient_email,message)
```

```
print("email sent successfully") except:
```

```
print("failed.")
```

```
vote()
```

output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
***election 2024***
press 1-> aki
press 2-> pm
press 3-> azh
enter the number you want to vote for:2
enter your email-id to receive voting details:raheemasyed7@gmail.com
email sent successfully
current vote counts:
pm:7
azh:3
aki:1
PS E:\besant py> & C:/Users/ashim/AppData/Local/Programs/Python/Python312/python.exe "e:/besant py/voting.py"
***election 2024***
press 1-> aki
press 2-> pm
press 3-> azh
enter the number you want to vote for:1
enter your email-id to receive voting details:shahinbadhurnisha@gmail.com
email sent successfully
current vote counts:
pm:7
azh:3
aki:2
PS E:\besant py> & C:/Users/ashim/AppData/Local/Programs/Python/Python312/python.exe "e:/besant py/voting.py"
***election 2024***
press 1-> aki
press 2-> pm
press 3-> azh
enter the number you want to vote for:2
enter your email-id to receive voting details:mahuthummohamed48@gmail.com
email sent successfully
current vote counts:
pm:8
azh:3
aki:2
PS E:\besant py> █
```

The screenshot shows a database management interface. On the left, a 'SCHEMAS' pane lists databases: election_2024, movie_bookings, sakila, sys, and world. The 'election_2024' database is expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'movie_bookings' database is also expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'world' database is selected. The main pane shows a query: `SELECT * FROM election_2024.votes;`. The 'Result Grid' displays 13 rows of data. The columns are: id, user_email, candidate_voted, vote_time, and vote_count. The data is as follows:

id	user_email	candidate_voted	vote_time	vote_count
1	mahuthummohamed48@gmail.com	pm	2024-06-29 21:04:30	8
2	ashimazabeen03@gmail.com	pm	2024-06-29 21:05:04	7
3	induummu40@gmail.com	azh	2024-06-29 21:05:32	3
4	ibrahimashik754@gmail.com	pm	2024-06-29 21:07:05	6
5	shahinbadhurnisha@gmail.com	aki	2024-06-29 21:08:26	2
6	mafriithahamedhussain@gmail.com	pm	2024-06-29 21:09:31	5
7	mashimazabeen@gmail.com	azh	2024-06-29 21:10:58	2
8	mfowsiya551@gmail.com	pm	2024-06-29 21:14:11	4
9	ibrahim12@gmail.com	azh	2024-06-29 21:15:44	1
10	moufcamahuthum@gmail.com	pm	2024-06-29 21:20:52	3
11	raheemasyed7@gmail.com	pm	2024-07-04 21:38:44	2
12	shahinbadhurnisha@gmail.com	aki	2024-07-04 21:41:12	1
13	mahuthummohamed48@gmail.com	pm	2024-07-04 21:42:30	1
HULL	HULL	HULL	HULL	HULL

2. Movie ticket booking

```
import mysql.connector

import smtplib

import datetime

mydb=mysql.connector.connect(

host="localhost",

user = "root",

password = "12345",

database= "movie_bookings"

)

def book_movie(movie_choice, email):

    cursor=mydb.cursor()
```

```

try:
    movies = {1: "conjuring", 2: "end_game", 3: "sita_ramam"}
    if movie_choice in movies:
        table_name = movies[movie_choice]
        booking_date=datetime.datetime.now()
        cursor.execute(f"INSERT INTO {table_name} (email, booking_date) VALUES (%s, %s)", (email,
booking_date))
        mydb.commit()
        print(f"Movie booking successful! Your email is {email}.")
        send_confirmation_email(movie_choice, email)
    else:
        print("Invalid movie choice.")
except:
    print("Error booking movie")
finally:
    cursor.close()

```

```

def send_confirmation_email(movie_choice, email):

```

```

    sender_email="sender_email"
    password="sender_password"
    movie_name = {1: "Conjuring", 2: "End_game", 3: "Sita Ramam"}[movie_choice]
    message= f"This email confirms your booking for the movie '{movie_name}'.\n"

```

```

try:
    with smtplib.SMTP('smtp.gmail.com',587) as s:
        s.starttls()
        s.login("sender_email", "sender_password")
        s.sendmail(sender_email, email, message)

```

```

        print("Confirmation email sent successfully.")
    except:
        print("Error sending email")

def main():

    movies = {1: "Conjuring", 2: "End_game", 3: "Sita Ramam"}
    print("Available movies:")
    for movie_id, movie_name in movies.items():
        print(f"{movie_id}. {movie_name}")

    while True:
        try:
            movie_choice = int(input("Enter your movie choice (1-3): "))
            if movie_choice in movies:
                break
            else:
                print("Invalid choice. Please enter a number between 1 and 3.")
        except ValueError:
            print("Invalid input. Please enter a number.")

    user_email = input("Enter your email address: ")

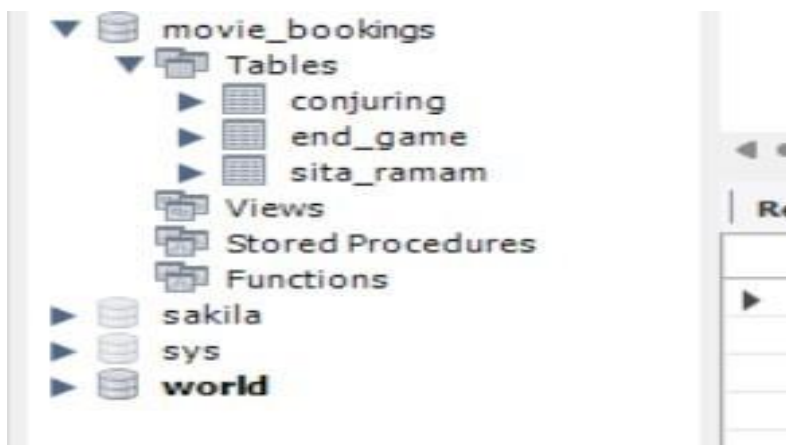
    book_movie(movie_choice, user_email)

if __name__ == "__main__":
    main()

```

Output:

```
PS E:\besant py> & C:/Users/ashim/AppData/Local/Programs/Python/Python312/python.exe "e:/besant py/ex7.py"
Available movies:
1. Conjuring
2. End_game
3. Sita Ramam
Enter your movie choice (1-3): 2
Enter your email address: ibrahimashik754@gmail.com
Movie booking successful! Your email is ibrahimashik754@gmail.com.
Confirmation email sent successfully.
PS E:\besant py> & C:/Users/ashim/AppData/Local/Programs/Python/Python312/python.exe "e:/besant py/ex7.py"
Available movies:
1. Conjuring
2. End_game
3. Sita Ramam
Enter your movie choice (1-3): 1
Enter your email address: raheemasyed7@gmail.com
Movie booking successful! Your email is raheemasyed7@gmail.com.
Confirmation email sent successfully.
PS E:\besant py> & C:/Users/ashim/AppData/Local/Programs/Python/Python312/python.exe "e:/besant py/ex7.py"
Available movies:
1. Conjuring
2. End_game
3. Sita Ramam
Enter your movie choice (1-3): 3
Enter your email address: mouficamahuthum@gmail.com
Movie booking successful! Your email is mouficamahuthum@gmail.com.
Confirmation email sent successfully.
PS E:\besant py> █
```



SCHEMAS

Filter objects

- election_2024
- movie_bookings
 - Tables
 - conjuring
- sakila
- sys
- world
 - Tables
 - Views
 - Stored Procedures
 - Functions

1 • `SELECT * FROM movie_bookings.conjuring;`

Limit to 1000 rows

Result Grid

email	booking_date
fouziyapm@gmail.com	2024-07-04 22:57:00
induammu40@gmail.com	2024-07-04 21:16:55
mafrithahamedhussain@gmail.com	2024-07-04 21:34:22
manhakathija07@gmail.com	2024-07-04 22:36:04
raheemasyed7@gmail.com	2024-07-04 21:36:50
NULL	NULL

SCHEMAS

Filter objects

- election_2024
- movie_bookings
 - Tables
 - conjuring
 - end_game
- sakila
- sys
- world
 - Tables
 - Views
 - Stored Procedures
 - Functions

1 • `SELECT * FROM movie_bookings.end_game;`

Result Grid

email	booking_date
fai fawwazmohamed@gmail.com	2024-07-04 22:41:43
ibrahimashik574@gmail.com	2024-07-04 21:31:48
ibrahimashik754@gmail.com	2024-07-04 21:36:02
mahuthummohamed48@gmail.com	2024-07-04 21:22:11
NULL	NULL

Filter objects

election_2024

movie_bookings

Tables

conjuring

end_game

sita_ramam

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

sakila

sys

world

Tables

Views

Stored Procedures

Functions

1 • `SELECT * FROM movie_bookings.sita_ramam;`

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Co

	email	booking_date
▶	mashimazabeen@gmail.com	2024-07-04 21:30:51
	mouficamahuthum@gmail.com	2024-07-04 21:37:22
	raheemasyed7@gmail.com	2024-07-04 22:35:05
	zabeen2003@gmail.com	2024-07-04 22:55:52
•	NULL	NULL