**Hangman Game**

A simple text-based hangman game that uses Python for the game logic and MySQL to store player data and high scores.

(In MySQL)

CREATE DATABASE hangman\_game;

USE hangman\_game;

CREATE TABLE players (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(255) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

high\_score INT DEFAULT 0

);

(In comm prompt)

pip install mysql-connector-python

(In python)

import mysql.connector

from mysql.connector import Error

import random

import getpass

# Database configuration

db\_config = {

'host': 'localhost',

'user': 'your\_username',

'password': 'your\_password',

'database': 'hangman\_game'

}

def create\_connection():

"""Create a database connection."""

try:

connection = mysql.connector.connect(\*\*db\_config)

if connection.is\_connected():

return connection

except Error as e:

print(f"Error: {e}")

return None

def close\_connection(connection):

"""Close the database connection."""

if connection.is\_connected():

connection.close()

def register(username, password):

"""Register a new player."""

connection = create\_connection()

if connection:

cursor = connection.cursor()

try:

cursor.execute("INSERT INTO players (username, password) VALUES (%s, %s)", (username, password))

connection.commit()

print("Registration successful.")

except Error as e:

print(f"Error: {e}")

finally:

close\_connection(connection)

def login(username, password):

"""Login an existing player."""

connection = create\_connection()

if connection:

cursor = connection.cursor()

cursor.execute("SELECT id, high\_score FROM players WHERE username = %s AND password = %s", (username, password))

result = cursor.fetchone()

close\_connection(connection)

if result:

return result[0], result[1]

return None, None

def update\_high\_score(player\_id, score):

"""Update the high score of a player."""

connection = create\_connection()

if connection:

cursor = connection.cursor()

cursor.execute("UPDATE players SET high\_score = %s WHERE id = %s", (score, player\_id))

connection.commit()

close\_connection(connection)

def play\_game(player\_id, high\_score):

"""Play the hangman game."""

words = ['python', 'mysql', 'database', 'programming', 'hangman']

word = random.choice(words)

guessed\_word = ['\_'] \* len(word)

guessed\_letters = set()

attempts = 6

print("\nWelcome to Hangman!")

print(" ".join(guessed\_word))

while attempts > 0 and '\_' in guessed\_word:

guess = input("Guess a letter: ").lower()

if guess in guessed\_letters:

print("You already guessed that letter. Try again.")

elif guess in word:

print("Good guess!")

for i, letter in enumerate(word):

if letter == guess:

guessed\_word[i] = guess

else:

print("Wrong guess.")

attempts -= 1

guessed\_letters.add(guess)

print(f"Guessed word: {' '.join(guessed\_word)}")

print(f"Remaining attempts: {attempts}")

if '\_' not in guessed\_word:

print("Congratulations! You guessed the word.")

score = attempts

if score > high\_score:

print("New high score!")

update\_high\_score(player\_id, score)

else:

print(f"Game over. The word was: {word}")

def print\_hangman(attempts):

if attempts == 7:

print(“\n+---+”)

print(“ |”)

print(“ |”)

print(“ |”)

print(“ ===”)

elif attempts == 6:

print(“\n+---+”)

print(“O |”)

print(“ |”)

print(“ |”)

print(“ ===”)

elif attempts == 5:

print(“\n+---+”)

print(“O |”)

print(“| |”)

print(“ |”)

print(“ ===”)

elif attempts == 4:

print(“\n+---+”)

print(“ O |”)

print(“/| |”)

print(“ |”)

print(“ ===”)

elif attempts == 3:

print(“\n+---+”)

print(“ O |”)

print(“/|\ |”)

print(“ |”)

print(“ ===”)

elif attempts == 2:

print(“\n+---+”)

print(“ O |”)

print(“/|\ |”)

print(“/ |”)

print(“ ===”)

elif attempts == 1:

print(“\n+---+”)

print(“ O |”)

print(“/|\ |”)

print(“/ \ |”)

print(“ ===”)

def main():

while True:

print("\nHangman Game")

print("1. Register")

print("2. Login")

print("3. Exit")

choice = input("Enter your choice: ")

if choice == '1':

username = input("Enter a username: ")

password = getpass.getpass("Enter a password: ")

register(username, password)

elif choice == '2':

username = input("Enter your username: ")

password = getpass.getpass("Enter your password: ")

player\_id, high\_score = login(username, password)

if player\_id:

print(f"Login successful. Your high score is {high\_score}.")

play\_game(player\_id, high\_score)

else:

print("Invalid username or password.")

elif choice == '3':

break

else:

print("Invalid choice. Please try again.")

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

main()

To run go comm. Prompt n type :-

(After choosing correct address (jisme file save h))

python hangman\_game.py

(n play the game in comm. Prompt only)

• EXPLAINATION

- Database Configuration: The `db\_config` dictionary contains the database connection details.

- Create Connection: The `create\_connection` function establishes a connection to the MySQL database.

- Close Connection: The `close\_connection` function closes the database connection.

- Register and Login: The `register` and `login` functions handle player registration and login.

- Update High Score: The `update\_high\_score` function updates the player's high score in the database.

- Play Game: The `play\_game` function contains the logic for the hangman game.

- Main Loop: The `main` function provides a command-line interface for interacting with the game.