



**Shree Vallabh Sanskar Dham's
Smt. Shobhaben Pratapbhai Patel Day Boarding
School
Killa - Pardi**

**A Project Report
On**

**TRIVIA MASTERS E-QUIZ
(PYTHON)**

As a part of the Informatics Practices Course (065)

Submitted By:
Swayam P. Panda
Dharal P. Naik

Under the Guidance of:
Mr. Rahul Gupta

Registration No.

CERTIFICATE

This is to certify that the Project entitled **TRIVIA MASTERS** is a bona fide work done by Master Swayam P. Panda and Dharal P. Naik of class XII Science Session 2022-23 in partial fulfillment of CBSE's school Examination 2023 and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidate.

Sign of student

Sign of Teacher

Sign of Principal

Name: _____

Roll no: _____

Date: _____

Sign of External

ACKNOWLEDGEMENT

I undertook this Project work, as the part of my XII-Informatics Practices course. I had tried to apply my best of knowledge and experience, gained during the study and class work experience. However, developing software system is generally a quite complex and time-consuming process. It requires a systematic study, insight vision and professional approach during the design and development. Moreover, the developer always feels the need, the help and good wishes of the people near you, who have considerable experience and idea.

I would like to extend my sincere thanks and gratitude to my teacher Mr. Rahul Gupta. I am very much thankful to our Principal Mr. Devendra Singh for giving valuable time and moral support to develop this software.

I would like to take the opportunity to extend my sincere thanks and gratitude to my father **Mr. Debashish Panda**, and my mother **Mrs. Santoshi Mishra** for being a source of inspiration and providing time and freedom to develop this software project.

CONTENTS

<u>SR NO.</u>	<u>CONTENTS</u>	<u>PAGE NO.</u>
1	Introduction	
2	Objective and Scope of the Project	
3	Theoretical Background	
3.1	What is GUI?	
3.2	What is Python IDLE?	
3.3	What is tkinter?	
3.4	What is Customtkinter?	
4	Problem Definition and Analysis	
5	Main libraries used and why?	
6	Which modules(libraries) are used?	
6.1	List of modules	
7	User Manual	
7.1	How to install software	
8	Working with software	
8.1	Source Code	
8.2	Screenshots	
9	References	

1. INTRODUCTION

This software project, named **TRIVIA MASTERS**, is an online quiz and trivia game developed from scratch to enhance skills such as developing research skills, improving language skills and reasoning ability, developing the power of concentration and increasing creative abilities.

Note: This software project is fully based on Graphical User Interface (GUI).

TRIVIA MASTERS is designed to promote a fun way to study and, in the process, help improve one's General Knowledge. It improves and expands one's knowledge of things. It is to encourage players to look beyond their textual knowledge and establish a relationship between theory and application of the learnt concepts.

2. OBJECTIVE & SCOPE OF THE PROJECT

TRIVIA MASTERS improves IQ Research states that solving questions helps raise the IQ (intelligence quotient) level of the player. It forces the solver to think and reason.

It improves **cognitive ability** in students, it enhances their ability to visualize and understand the whole - part relationship.

The proposed software system is expected to do the following functionality-

- To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment.
- To provide visual and user-friendly interface to interact like a real E-quiz **TRIVIA MASTERS**.
- To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

There is a good scope for improving one's general knowledge. You can be one step ahead of others. You can be more confident on any topic and gain more **knowledge and** also able to convey the information to others as you already know about it. By these projects students will Acquire knowledge for developing many more projects like **TRIVIA MASTERS**.

This program improves methods of solving Quiz. The report concludes by evaluating the end application to analyze how good it met its objectives and the performance of **TRIVIA MASTERS e-quiz game**.

Despite of the best effort of the developer, the following limitations and functional boundaries are visible, which limits the scope of this application software:

1. This software fails to fetch random questions from the internet and works only on the 05 questions stored in the dictionary.
2. If the user does not enter his/her name at the beginning he/she would have to open the software again.
3. Cannot work without Wi-Fi/Internet connection.
4. Windows always open in minimized format.

3. THEORETICAL BACKGROUND

3.1 What is GUI?

Introduction and Concepts:

A graphics-based operating system interface that uses icons, menus and a mouse (to click on the icon or pull down the menus) to manage interaction with the system. Developed by Xerox, the GUI was popularized by the Apple Macintosh in the 1980s.

A GUI is considered to be more user-friendly than a text-based command-line interface, such as MS-DOS, or the shell of Unix-like operating systems.

Unlike a command-line operating system or CUI, like Unix or MS-DOS, GUI operating systems are easier to learn and use because commands do not need to be memorized. Additionally, users do not need to know any programming languages. Because of their ease of use and more modern appearance, GUI operating systems have come to dominate today's market.

What are the elements of a GUI?

To make a GUI as user-friendly as possible, there are different elements and objects that the user use to interact with the software. Below is a list of each of these with a brief description.

- **Button** - A graphical representation of a button that performs an action in a program when pressed
- **Dialog box** - A type of window that displays additional information, and asks a user for input.
- **Icon** - Small graphical representation of a program, feature, or file.
- **Menu** - List of commands or choices offered to the user through the menu bar.
- **Menu bar** - Thin, horizontal bar containing the labels of menus.
- **Ribbon** - Replacement for the file menu and toolbar that groups programs activities together.
- **Tab** - Clickable area at the top of a window that shows another page or area.
- **Toolbar** - Row of buttons, often near the top of an application window, that controls software functions.

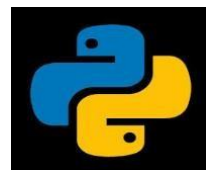
3.2) What is Python IDLE?

Python is a general-purpose high-level programming language. It is an open-source language, released under a GPL-compatible license. Python Software Foundation (PSF), a non-profit organization, holds the copyright of Python.

Guido Van Rossum conceived Python in the late 1980s. It was released in 1991 at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language. He named this language after a popular comedy show called 'Monty Python's Flying Circus' (and not after Python - the snake).

In the last few years, its popularity has increased immensely. According to stackoverflow.com's recent survey, Python is in the top ten Most Popular Technologies in 2018.

Official Web Site: <https://www.python.org>.



Python Features:

- Python is an interpreter-based language, which allows execution of one instruction at a time.
- Extensive basic data types are supported e.g. numbers (floating point, complex, and unlimited-length long integers), strings (both ASCII and Unicode), lists, and dictionaries.
- Variables can be strongly typed as well as dynamic typed.
- Supports object-oriented programming concepts such as class, inheritance, objects, module, namespace etc.
- Cleaner exception handling support.
- Supports automatic memory management.

Python Advantages

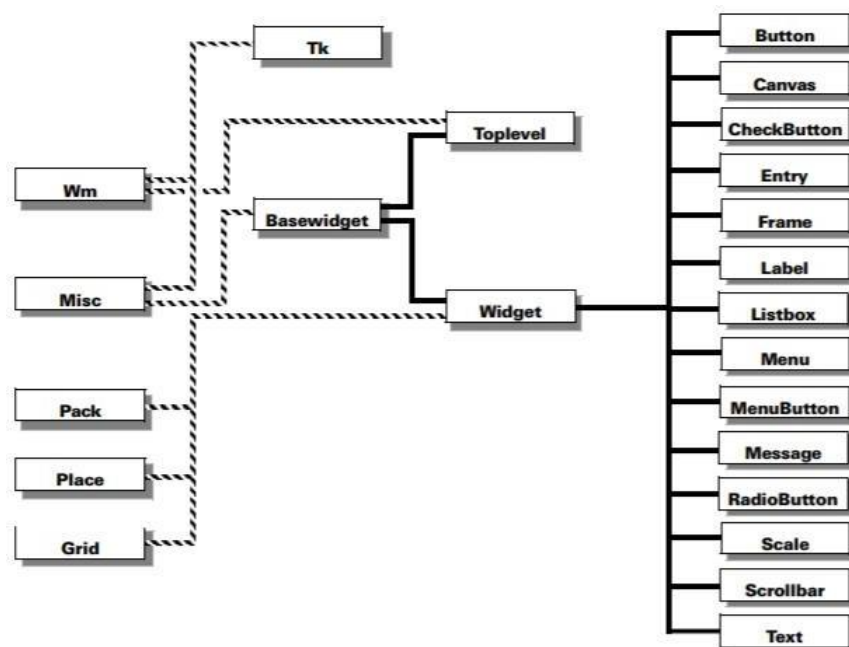
- Python provides enhanced readability.
- Python is free and distributed as open-source software. A large programming community is actively involved in the development and support of Python libraries for various applications such as web frameworks, mathematical computing and data science.

- Python is a cross-platform language. It works equally on different OS platforms like Windows, Linux, Mac OSX etc. Hence Python applications can be easily ported across OS platforms.
- It provides a very easy syntax for making a GUI and also helps to increase the code readability compared to other.

3.3) What is tkinter?

Tkinter is a pre-installed python library which is used to make GUI. Tkinter comes pre-installed with the Python installer binaries for Mac OS X and the Windows platform. So if you install Python from the official binaries for Mac OS X or Windows platform, you are good to go with Tkinter.

Here are some using classes and their inheritance which is used the most in making of the GUI using tkinter:



We can also understand the tkinter classes easily from the docstring of a function present the location of the main library folder where tkinter is stored and the file is named as “__init__.py”.

We have used CustomTkinter since it provides better functionalities customization as compared to the stock tkinter module.

3.4) What is Customtkinter?

CustomTkinter is a python UI-library based on Tkinter, which provides new, modern and fully customizable widgets. They are created and used like normal Tkinter widgets and can also be used in combination with normal Tkinter elements. The widgets and the window colors either adapt to the system appearance or the manually set mode ('light', 'dark'), and all CustomTkinter widgets and windows support HighDPI scaling (Windows, macOS). With CustomTkinter you'll get a consistent and modern look across all desktop platforms (Windows, macOS, Linux).

4. PROBLEM DEFINITION & ANALYSIS

The hardest part of building a software system is deciding precisely what to build. No other part of the conceptual work is so difficult as establishing the detailed technical requirement. Defining and applying good, complete requirements are hard to work, and success in this endeavor has eluded many of us. Yet, we continue to make progress.

Problem definition describes the *what* of a system, not *How*. The quality of a software product is only as good as the process that creates it. Problem definition is one of the most crucial steps in this creation process. Without defining a problem, developers do not know what to build, customers do not know what to expect, and there is no way to validate that the built system satisfies the requirement.

Problem definition and Analysis is the activity that encompasses learning about the problem to be solved, understanding the needs of customer and users, trying to find out who the user really is, and understanding all the constraints on the solution. It includes all activities related to the following:

- Identification and documentation of customer's or user's needs.

- Creation of a document that describes the external behavior and the association constraints that will satisfies those needs.
- Analysis and validation of the requirements documents to ensure consistency, completeness, and feasibility
- Evolution of needs.

After the analysis of Trivia Masters, the proposed system is expected to do the following-

- To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment for any new user.
- The GUI should look like an online quiz with various types of questions.
- A login and register system which would allow users to access the quiz more efficiently.
- To provide piece of mind features like a quit button, a back button to better utilize the features of the quiz.

5. SYSTEM IMPLEMENTATION

While developing the software, the **Hardware** used is:

- ✓ MacBook Air M1, 2020
- ✓ 8 GB RAM

The **Software's** used:

- ✓ Microsoft Windows 10 Pro as Operating System
- ✓ Python 3.11.0 as programming language
- ✓ Visual Studio Code as an IDE
- ✓ MySQL command line client for storing user information
- ✓ CSV file for storing questions

6. WHICH MODULES (LIBRARIES) ARE USED?

6.1 List of Modules

6.1.1. tkinter

Tkinter is the backbone of this software. It helps us to make the well framed GUI. The most important thing in the tkinter is to make the GUI work as per our need and for which event binding is mandatory to be known. I have used Object-Oriented Programming to make this software as I have combined more than one GUI in a single GUI. The simplest code for making a GUI from OOP(Object-Oriented Programming) is:

```
from tkinter import *

root = Tk()
root.title("My GUI")
root.geometry("300x100")

class GUI:
    def __init__(self, root):
        label = Label(root, text="Hello World")
        label.pack()

myobject = GUI(root)

root.mainloop()
```

6.1.2. PIL (Python Image Library)

This is not a pre-installed library in python. It is help to insert the image of different abbreviation i.e., .gif, .jpg, .jpeg, .png into a GUI. Tkinter by default support only the .png form of image for insertion.

To install the PIL library, open your command prompt and write:

```
pip install pillow
```

6.1.3 csv

The `csv` module implements classes to read and write tabular data in CSV format. It allows programmers to say, “write this data in the format preferred by Excel,” or “read data from this file which was generated by Excel,” without knowing the precise details of the CSV format used by Excel. Programmers can also describe the CSV formats understood by other applications or define their own special purpose CSV formats.

6.1.4. requests

The requests module allows you to send HTTP requests using Python.

The HTTP request returns a Response Object with all the response data (content, encoding, status, etc).

6.1.5. mysql.connector

MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249). It is written in pure Python and does not have any dependencies except for the Python Standard Library.

6.1.6. pandas

Python is a great language for doing data analysis, primarily because of the fantastic ecosystem of data-centric python packages. Pandas is one of those packages and makes importing and analyzing data much easier.

To install the **pandas** library, open your command prompt and write:

```
pip install pandas
```

To import all the required libraries, write the following code:

```
from cProfile import label
from fnmatch import fnmatch
from gzip import FNAME
from tkinter import LEFT, IntVar, Button, Label, PhotoImage, Radiobutton, Tk, Entry
from itertools import count, cycle
from tkinter import messagebox as mb
import csv
from unicodedata import name
from pandas import *
import numpy
from tkinter.messagebox import QUESTION
from re import A, I
import pandas as pd
import requests as r
from matplotlib.ft2font import BOLD
from cv2 import destroyAllWindows
from unittest import result
from tracemalloc import start
from tkinter.font import ITALIC
import tkinter as tk
from multiprocessing.sharedctypes import Value
from io import BytesIO
from gettext import install
from email.mime import image
from doctest import master
import tkinter.messagebox
import PIL
from turtle import width
import mysql.connector
from tkinter import *
import tkinter
import customtkinter
from tkinter import Label, PhotoImage
from PIL import Image, ImageTk
```

7. USER MANUAL

7.1 How to install Software:

- **Basic Hardware Requirement-**

- ✓ Intel or AMD processor
- ✓ 128 MB RAM and 4GB HDD space is desirable.
- ✓ Standard I/O devices like Keyboard and Mouse etc.

- **Basic Software Requirement-**

- ✓ Windows 7 OS or above is desirable.
- ✓ Python 3.10 or above with the required libraries:
 - Tkinter
 - CustomTkinter
 - PIL
 - MySQL Connector
 - CSV
 - Tkinter.messagebox

8. WORKING WITH SOFTWARE

8.1 Source Code:

```
# Original
from cProfile import label
from fnmatch import fnmatch
from gzip import FNAME
from this import d
from tkinter import LEFT, IntVar, Button, Label, PhotoImage, Radiobutton, Tk, Entry
from itertools import count, cycle
from tkinter import messagebox as mb
import csv
from unicodedata import name
from pandas import *
import numpy
from tkinter.messagebox import QUESTION
from re import A, I
import pandas as pd
import requests as r
from matplotlib.ft2font import BOLD
from cv2 import destroyAllWindows
from unittest import result
from tracemalloc import start
from tkinter.font import ITALIC
import tkinter as tk
from multiprocessing.sharedctypes import Value
from io import BytesIO
from gettext import install
from email.mime import image
from doctest import master
import tkinter.messagebox
import PIL
from turtle import width
import mysql.connector
from tkinter import *
import tkinter
import customtkinter
from tkinter import Label, PhotoImage
from PIL import Image, ImageTk

customtkinter.set_appearance_mode("Dark")
def loginPage(logdata):
```

```

global loginPage
login = customtkinter.CTk()
login.title('Quiz App Login')
login.geometry("800x700")

user_name = StringVar()
password = StringVar()

# Heading
heading = Label(login, text="Quiz App Login",
                fg="cyan", bg="#212325")
heading.config(font=('calibri 100 italic bold'))
heading.place(relx=0.1, rely=0.15)

# Image
image = Image.open('login_edited.png')
imgre = image.resize((600, 600))
img1 = ImageTk.PhotoImage(imgre)

img = customtkinter.CTkLabel(
    login, image=img1, fg='black', bg='black')
img.place(relx=0.6, rely=0.3)

#USER NAME
user_name = StringVar()
ulabel = customtkinter.CTkLabel(
    login, text="Username", fg='white', bg='black')
ulabel.place(relx=0.15, rely=0.5)
uname = customtkinter.CTkEntry(login, bg='black', fg='black',
                               textvariable=user_name)
uname.config(width=15)
uname.place(relx=0.28, rely=0.5)

# PASSWORD
plabel = customtkinter.CTkLabel(
    login, text="Password", fg='white', bg='black')
plabel.place(relx=0.15, rely=0.6)
pas = customtkinter.CTkEntry(login, bg='black', fg='black',
                              textvariable=password, show="*")
pas.config(width=15)
pas.place(relx=0.28, rely=0.6)

# QUIT BUTTON
quit_button = customtkinter.CTkButton(text="Quit", master=login,
                                       corner_radius=15, command=login.destroy)
quit_button.place(relx=0.1, rely=0.95, anchor=tkinter.CENTER)

```



```

def check():
    global b
    for a, b, c in logdata:
        if b == uname.get() and c == pas.get():
            #print(logdata)
            login.destroy()
            ### Ignore this file. Use mainlogreg.py ###
            # import required classes from tkinter
            # and import messagebox as mb from tkinter

            customtkinter.set_appearance_mode("Dark")
            # For GIF

class ImageLabel(tk.Label):
    def load(self, im):
        if isinstance(im, str):
            im = Image.open(im)
            frames = []

            try:
                for i in count(1):
                    frames.append(ImageTk.PhotoImage(im.copy()))
                    im.seek(i)
            except EOFError:
                pass
            self.frames = cycle(frames)

            try:
                self.delay = im.info['duration']
            except:
                self.delay = 100

            if len(frames) == 1:
                self.config(image=next(self.frames))
            else:
                self.next_frame()

    def unload(self):
        self.config(image=None)
        self.frames = None

    def next_frame(self):
        if self.frames:
            self.config(image=next(self.frames))
            self.after(self.delay, self.next_frame)

class Quiz:
    def __init__(self):

```

```

self.title_label = Label(gui, text='Trivia Masters',
                        fg="cyan", bg="#212325")
self.title_label.config(font=('calibri 80 italic bold'))
self.title_label.place(relx=0.03, rely=0.05)
self.info_label = Label(
    gui, text="Press Continue to Start the Quiz:", bg="#212325")
self.info_label.config(font=('calibri 69 italic bold'))
self.info_label.place(relx=0.03, rely=0.25)

self.info_label1 = Label(gui, text="1. There are a total of 5 questions.\n2. The quiz is
    in the form of a MCQ\
\n3. There is no negative marking.\n 4. The quiz is based on current affairs.\n\n\n\n\
Press continue to start the quiz now.", fg="white", bg="#212325", justify=LEFT)
self.info_label1.config(font=('calibri 60 italic bold'))
self.info_label1.place(relx=0.03, rely=0.25)
self.lbl = ImageLabel(gui)
self.lbl.pack()
self.lbl.load('brainisback.png')
self.lbl.place(relx=0.7, rely=0.3)

self.buttons(place_next=False)
self.ent = self.entry_button()

def add_label(self, display_text, x, y):
    label = Label(gui, text=display_text, width=20,
                  font=('arial', 12, 'bold'), anchor='w')
    label.place(x=x, y=y)
    return label

def start(self):
    self.title_label.destroy()
    self.info_label.destroy()
    self.info_label1.destroy()
    self.lbl.destroy()
    self.info_label
    self.ent.destroy()
    self.q_no = 0
    self.img = None
    self.display_image()
    self.display_title()
    self.display_question()
    self.back_button()
    self.opt_selected = IntVar()
    self.opts = self.radio_buttons()
    self.display_options()
    self.next, self.quit = self.buttons()
    self.data_size = len(question)
    self.correct = 0
def show_msg(self, msg):

```

```

mb.showinfo(msg)

def display_result(self):

    wrong_count = self.data_size - self.correct
    sum = 0
    for value in result_ans.values():
        sum += value
    wrong_count = self.data_size - sum
    correct = f"{sum}"
    wrong = f"{wrong_count}"
    score = int(sum / self.data_size * 100)
    result = f"Score: {score}%"
    # mb.showinfo("Result", f"{result}\n{correct}\n{wrong}")
    gui.destroy()
    app = customtkinter.CTk()
    app.overridedirect(True)
    app.geometry(
        "{0}x{1}+0+0".format(app.winfo_screenwidth(), app.winfo_screenheight()))

    app.title("Result")

    # Name
    name_label = tkinter.Label(
        master=app, text="Congratulations", justify=tkinter.LEFT)
    name_label.config(font=('calibri 55 italic bold'),
        fg="white", bg="#222325")
    name_label.place(relx=0.05, rely=0.10)

    name_name = tkinter.Label(
        master=app, text=b, justify=tkinter.LEFT)
    name_name.config(font=('calibri 55 italic bold'),
        fg="white", bg="#222325")
    name_name.place(relx=0.226, rely=0.10)

    # Correct
    c_label = tkinter.Label(
        master=app, text="Number of correct answers is", justify=tkinter.LEFT)
    c_label.config(font=('calibri 55 italic bold'),
        fg="white", bg="#222325")
    c_label.place(relx=0.05, rely=0.30)

    c_label1 = tkinter.Label(
        master=app, text=correct, justify=tkinter.RIGHT)
    c_label1.config(font=('calibri 55 italic bold'),
        bg="#222325", fg="white")
    c_label1.place(relx=0.365, rely=0.30)
    #

```

```

# Wrong
w_label = tkinter.Label(
    master=app, text="Number of wrong answers is", justify=tkinter.LEFT)
w_label.config(font=('calibri 55 italic bold'),
               bg="#222325", fg="white")
w_label.place(relx=0.05, rely=0.40)

w_label1 = tkinter.Label(
    master=app, text=wrong, justify=tkinter.RIGHT)
w_label1.config(font=('calibri 55 italic bold'),
                bg="#222325", fg="white")
w_label1.place(relx=0.365, rely=0.40)
#

# Score
s_label = tkinter.Label(
    master=app, text="Your total score is", justify=tkinter.LEFT)
s_label.config(font=('calibri 55 italic bold'),
               bg="#222325", fg="white")
s_label.place(relx=0.05, rely=0.50)

s_label1 = tkinter.Label(
    master=app, text=score, justify=tkinter.RIGHT)
s_label1.config(font=('calibri 55 italic bold'),
                bg="#222325", fg="white")
s_label1.place(relx=0.248, rely=0.50)
#

image = Image.open('thankyou.jpg')
imgre = image.resize((800, 800))

img1 = ImageTk.PhotoImage(imgre)

img = customtkinter.CTkLabel(
    app, image=img1, fg='black', bg='black')
img.place(relx=0.6, rely=0.24)

quit_button = customtkinter.CTkButton(
    text="Quit", master=app, corner_radius=15, command=app.destroy)

quit_button.place(relx=0.1, rely=0.95, anchor=tkinter.CENTER)

app.mainloop()

def check_ans(self, q_no):
    user_choice[q_no] = self.opt_selected.get()
    if self.opt_selected.get() == answer[q_no]:
        return True

def next_btn(self):

```

```

if self.q_no == len(question)-2:
    self.next['text'] = 'Submit'

if self.check_ans(self.q_no):
    self.correct += 1
    result_ans[self.q_no] = 1
else:
    result_ans[self.q_no] = 0
self.q_no += 1
if self.q_no == self.data_size:
    self.display_result()

else:
    self.display_question()
    self.display_image()
    self.display_options()

def add_textbox(self, x, y, placeholder="text"):
    textbox = Entry(text=placeholder)
    textbox.place(x=x, y=y)
    return textbox

def buttons(self, place_next=True):
    next_button = None
    if place_next:
        next_button = customtkinter.CTkButton(
            text="Next", master=gui, corner_radius=15, command=self.next_btn)

        next_button.place(relx=0.9, rely=0.95, anchor=tkinter.CENTER)

    quit_button = customtkinter.CTkButton(
        text="Quit", master=gui, corner_radius=15, command=gui.destroy)

    quit_button.place(relx=0.1, rely=0.95, anchor=tkinter.CENTER)
    return next_button, quit_button

def back_button(self):
    back_btn = customtkinter.CTkButton(
        text="Back", master=gui, corner_radius=15, command=self.go_back)
    back_btn.place(relx=0.7, rely=0.95, anchor=tkinter.CENTER)

def go_back(self):
    if self.q_no == 0:
        return
    self.q_no -= 1
    self.display_question()
    self.display_image()
    self.display_options()

```

```

def entry_button(self):
    entry_button = customtkinter.CTkButton(
        text="Continue", master=gui, command=self.start, corner_radius=15)
    entry_button.place(relx=0.9, rely=0.95, anchor=tkinter.CENTER)
    return entry_button

def display_options(self):
    val = 0

    self.opt_selected.set(0)
    for option in options[self.q_no]:
        self.opts[val]['text'] = option
        val += 1
    for count in range(1, 5):
        if user_choice[self.q_no] == count:
            self.opt_selected.set(count)

def display_question(self):

    q_no = Label(gui, text=question[self.q_no], width=100,
        font=('ariel', 40, 'bold'), fg='white', anchor='w', bg="#222325")

    q_no.place(x=10, y=100)

def display_image(self):
    url = imgs[self.q_no]

    u = r.get(url)
    raw_data = u._content
    u.close()

    im = Image.open(BytesIO(raw_data))
    size = (800, 800)
    im = im.resize(size)
    image = ImageTk.PhotoImage(im)
    self.img = image
    img = Label(image=image, height=800, width=800)
    img.place(x=1900, y=300)

def display_title(self):

    # The title to be shown
    title = Label(gui, text="Trivia Masters",
        width=90, bg="#0f0f61", fg="#64b9cc", font=("ariel", 40, "bold", ITALIC))

```

```

# place of the title
title.place(x=0, y=2)

def radio_buttons(self):

    q_list = []

    # position of the first option
    y_pos = 400

    # adding the options to the list
    while len(q_list) < 4:
        radio_btn = tkinter.Radiobutton(gui, text=" ", variable=self.opt_selected,

        value=len(q_list)+1, font=("ariel", 40, 'bold'), fg='#b3bdb9',bg="#222325")

        # adding the button to the list
        q_list.append(radio_btn)

        # placing the button
        radio_btn.place(x=75, y=y_pos)

# incrementing the y-axis position by 40
    y_pos += 150

    # return the radio buttons
    return q_list

gui = customtkinter.CTk()
# gui.geometry("800x600")
# gui.attributes('-fullscreen', True)
# gui.state('zoomed')
gui.overridedirect(True)
gui.geometry("{0}x{1}+0+0".format(gui.winfo_screenwidth(),
    gui.winfo_screenheight()))

gui.title("PhotoEvaluate Quiz")

with open('questions.csv', newline="") as f:
    reader = csv.reader(f)
    data = list(reader)
    options = []
    no_of_ques = 6 # no of total questions in quiz + 1
    for i in range(1, no_of_ques):
        opt = data[i][3:7]
        options.append(opt)

```

```
data = read_csv("questions.csv", encoding="ISO-8859-1")
```

```
question = data['question'].tolist()
```

```
answer = data['answer'].tolist()
```

```
imgs = data['imgs'].tolist()
```

```
result_ans = {}
```

```
user_choice = {item: 0 for item in range(len(question))}
```

```
quiz = Quiz()
```

```
gui.mainloop()
```

```
###
```

```
else:
```

```
    error = Label(
```

```
        login, text="Wrong Username or Password!", fg='blue', bg='grey')
```

```
    error.place(relx=0.37, rely=0.7)
```

```
#LOGIN BUTTON
```

```
log = customtkinter.CTkButton(
```

```
    login, text='Login', command=check, corner_radius=10)
```

```
log.place(relx=0.35, rely=0.7, anchor=tkinter.CENTER)
```

```
login.mainloop()
```

```
def signUpPage():
```

```
    fpage.destroy()
```

```
    global sup
```

```
    sup = customtkinter.CTk()
```

```
    sup.title('Quiz App')
```

```
    sup.geometry("800x700")
```

```
    fname = StringVar()
```

```
    uname = StringVar()
```

```
    passW = StringVar()
```

```
    heading = Label(sup, text="Quiz App Signup",  
                    fg="#fc034e", bg="#212325")
```

```
    heading.config(font=('calibri 100 italic bold'))
```

```
    heading.place(relx=0.1, rely=0.15)
```

```
    # Image
```

```
    image = Image.open('signup.png')
```

```
    imgre = image.resize((600, 600))
```

```
    img1 = ImageTk.PhotoImage(imgre)
```



```

img = customtkinter.CTkLabel(
    sup, image=img1, fg='black', bg='black')
img.place(relx=0.6, rely=0.3)

#full name
flabel = customtkinter.CTkLabel(
    sup, text="Full Name", fg='white', bg='black')
flabel.place(relx=0.15, rely=0.4)
fname1 = customtkinter.CTkEntry(sup, bg='black', fg='black',
                                textvariable=fname)
fname1.place(relx=0.28, rely=0.4)

```

```

#username
ulabel = customtkinter.CTkLabel(
    sup, text="Username", fg='white', bg='black')
ulabel.place(relx=0.15, rely=0.5)
user = customtkinter.CTkEntry(sup, bg='black', fg='black',
                              textvariable=uname)
user.place(relx=0.28, rely=0.5)

```

```

#password
plabel = customtkinter.CTkLabel(
    sup, text="Password", fg='white', bg='black')
plabel.place(relx=0.15, rely=0.6)
pas = customtkinter.CTkEntry(sup, bg='black', fg='black',
                              textvariable=passW, show="*")
pas.place(relx=0.28, rely=0.6)

```

```

def addUserToDataBase():
    fullname = fname.get()
    username = user.get()
    password = pas.get()

    if len(fname.get()) == 0 and len(user.get()) == 0 and len(pas.get()) == 0:
        error = Label(
            text="You haven't enter any field...Please Enter all the fields", fg='black', bg='white')
        error.place(relx=0.37, rely=0.7)

    elif len(fname.get()) == 0 or len(user.get()) == 0 or len(pas.get()) == 0:
        error = Label(text="Please Enter all the fields",
                      fg='black', bg='white')
        error.place(relx=0.37, rely=0.7)

    elif len(user.get()) == 0 and len(pas.get()) == 0:

```

```
error = Label(
    text="Username and password can't be empty", fg='black', bg='white')
error.place(relx=0.37, rely=0.7)
```

```
elif len(user.get()) == 0 and len(pas.get()) != 0:
    error = Label(text="Username can't be empty",
        fg='black', bg='white')
    error.place(relx=0.37, rely=0.7)
```

```
elif len(user.get()) != 0 and len(pas.get()) == 0:
    error = Label(text="Password can't be empty",
        fg='black', bg='white')
    error.place(relx=0.37, rely=0.7)
```

```
else:
```

```
    conn = mysql.connector.connect(
        host="localhost",
        user="root",
        password='root',
        database='quiz123'
    )
```

```
    create = conn.cursor()
    create.execute(
        'CREATE TABLE IF NOT EXISTS userSignUp(FULLNAME text, USERNAME
        text,PASSWORD text)')
    create.execute("INSERT INTO userSignUp VALUES (%s,%s,%s)",
        (fullname, username, password))
    conn.commit()
    create.execute('SELECT * FROM userSignUp')
    z = create.fetchall()
    #print(z)
    #L2.config(text="Username is "+z[0][0]+" \n Password is "+z[-1][1])
    conn.close()
    sup.destroy()
    loginPage(z)
```

```
def gotoLogin():
    sup.destroy()
    conn = mysql.connector.connect(host="localhost",
        user="root",
        password='root',
        database='quiz123'
    )
    create = conn.cursor()
    conn.commit()
    create.execute('SELECT * FROM userSignUp')
    z = create.fetchall()
    loginPage(z)
```

```

#signup BUTTON
sp = customtkinter.CTkButton(sup, text='SignUp', corner_radius=15,
                             command=addUserToDataBase, bg="black")
sp.place(relx=0.35, rely=0.7, anchor=tkinter.CENTER)

# QUIT BUTTON
quit_button = customtkinter.CTkButton(text="Quit", master=sup,
                                       corner_radius=15,
                                       command=sup.destroy)

quit_button.place(relx=0.1, rely=0.95, anchor=tkinter.CENTER)

#Already have a account
log = customtkinter.CTkButton(sup, text='Already have a Account?', corner_radius=15,
                              command=gotoLogin)
log.place(relx=0.35, rely=0.8, anchor=tkinter.CENTER)
sup.mainloop()

def start():
    global fpage
    fpage = tkinter.Tk()
    fpage.title("Trivia Masters")
    fpage.geometry("1100x700")
    fpage.config(background="#000000")
    img1 = PhotoImage(file="bg_returns.png")
    labelimage = Label(
        fpage,
        image=img1,
        background="#000000",
    )
    labelimage.pack(pady=(80, 40))

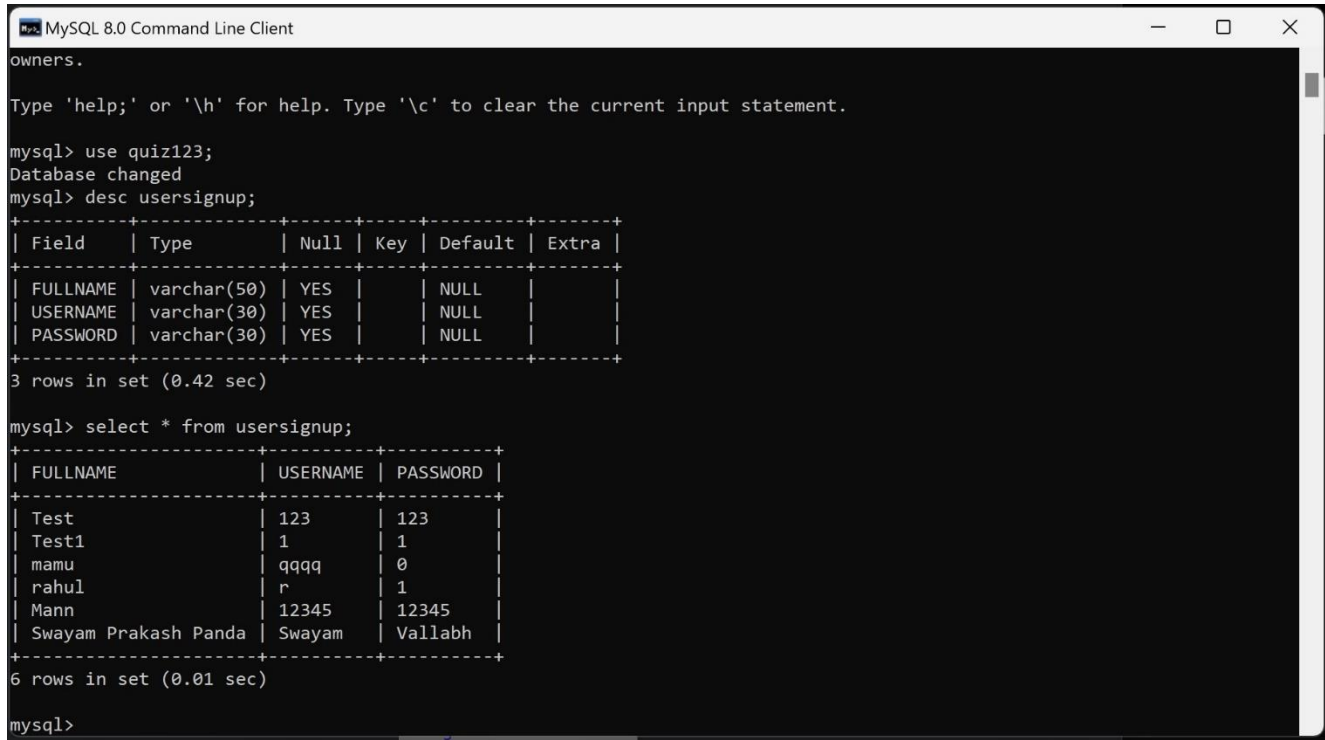
#Button
button = customtkinter.CTkButton(master=fpage,
                                  text="Start",
                                  command=signUpPage,
                                  corner_radius=0
                                  )
button.pack(pady=20)
button.place(relx=0.5, rely=0.8, anchor=tkinter.CENTER)

fpage.mainloop()

if __name__ == '__main__':
    start()

```

8.1 Screenshots:



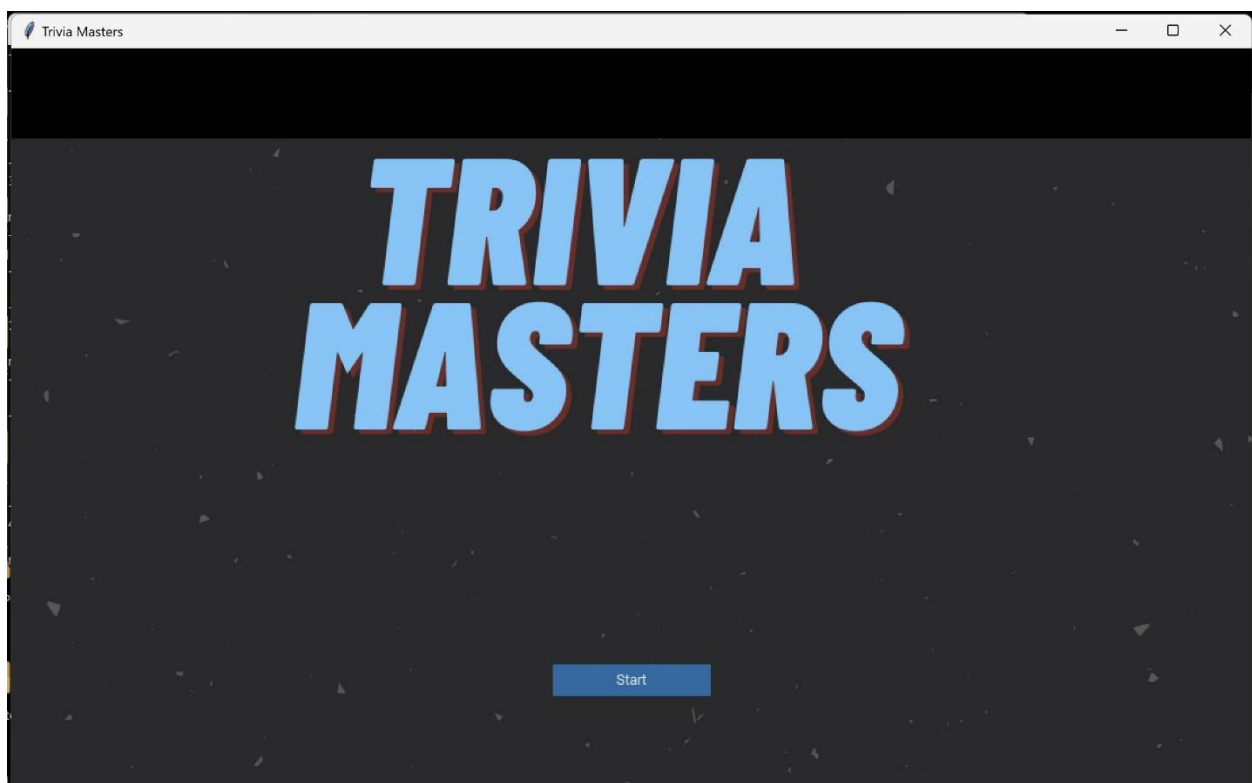
```
MySQL 8.0 Command Line Client
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use quiz123;
Database changed
mysql> desc usersignup;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| FULLNAME | varchar(50) | YES  |     | NULL    |       |
| USERNAME | varchar(30) | YES  |     | NULL    |       |
| PASSWORD | varchar(30) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.42 sec)

mysql> select * from usersignup;
+-----+-----+-----+
| FULLNAME | USERNAME | PASSWORD |
+-----+-----+-----+
| Test     | 123      | 123      |
| Test1    | 1        | 1        |
| mamu     | qqqq     | 0        |
| rahul    | r        | 1        |
| Mann     | 12345    | 12345    |
| Swayam Prakash Panda | Swayam   | Vallabh  |
+-----+-----+-----+
6 rows in set (0.01 sec)

mysql>
```



Quiz App

Quiz App Signup

Full Name


Username

Password

[SignUp](#)

[Already have a Account?](#)

[Quit](#)

A stylized graphic with the text "SIGN UP NOW!" in bold, pink, block letters. The text is enclosed in a pink rectangular frame with a jagged, speech-bubble-like bottom edge.

Quiz App Login


Quiz App Login

Username

Password

[Login](#)

[Quit](#)

A stylized graphic featuring a large white question mark inside a yellow, cloud-like shape with pink and orange accents.

Trivia Masters

1. There are a total of 5 questions.
2. The quiz is in the form of a MCQ
3. There is no negative marking.
4. The quiz is based on current affairs.



Press continue to start the quiz now.

Quit

Continue

Trivia Masters

What is the name of this historical structure which is located in Piazza del Duomo, Italy?

- Leaning Tower of pisa
- Milan Cathedral
- Ponte Vecchio
- Colosseum



Quit

Back

Next

questions									
Home Insert Draw Page Layout Formulas Data Review View Acrobat Tell me Share									
A1	question								
	A	B	C	D	E	F	G	H	
1	question	answer	imgs						
2	What is the name of this historical structure which is located in Piazza del Duomo, Italy?	1	https://cdn.pixabay.com/photo/2015/11/22/16/45/pisa-1056568_960_720.jpg	Leaning Tower	Milan Cathedral	Ponte Vecchio	Colosseum		
3	This famous statue located in New York City is referred to as:	4	https://cdn.pixabay.com/photo/2014/02/17/10/20/statue-of-liberty-267948_960_720.jpg	Spring Temple	Christ the Redeemer	Statue of Liberty			
4	The structure below located in the Middle East was built thousands of years ago, and still stands today.	3	https://cdn.pixabay.com/photo/2016/02/02/18/33/sphinx-1175828_960_720.jpg	Pharaoh's Tomb	Joseph's Tomb	Egyptian Pyramids	Mecca Temple		
5	The highest mountain in the world, in Nepal, standing at 8,848 meters.	4	https://cdn.pixabay.com/photo/2010/11/29/mount-everest-413_1280.jpg	Makalu	Nanga Parbat	Swiss Alps	Mount Everest		
6	This is the longest wall in the world and has a main-line length of 3,460 km	2	https://cdn.pixabay.com/photo/2017/02/01/14/14/great-wall-of-china-2030311_960_720.jpg	Kumbhalgarh	Great Wall of China	Amer Fort	Jai	Ancient Walls of Mesopotamia	

Congratulations Swayam

Number of correct answers is 5

Number of wrong answers is 0

Your total score is 100

Thank You!

Quit

9. BIBLIOGRAPHY

- ✓ Teachers Guidance
- ✓ SUMITA ARORA Textbook-Class XII
- ✓ PREETI ARORA Textbook-Class XII
- ✓ <https://stackoverflow.com/>
- ✓ <https://www.geeksforgeeks.org/>
- ✓ <https://www.codewithharry.com/>
- ✓ <https://www.google.co.in/>
- ✓ <https://youtube.com>