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

A Project Report
On

K.B.C. – Kaun Banega Crorepati
(PYTHON)

As a part of the Informatics Practices Course (065)

SUBMITTED BY	Under the Guidance of:
Jay J. Patel	Mr. Rahul Gupta
Harsh B. Parmar	

Registration No.

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

A Project Synopsis
For Term-I
On

K.B.C. – Kaun Banega Crorepati
(PYTHON)

As a part of the Informatics Practices Course (065)

SUBMITTED BY
Jay J. Patel
Harsh B. Parmar

Under the Guidance of:
Mr. Rahul Gupta

Registration No.

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

A Project Synopsis
For Term-II
On

K.B.C. – Kaun Banega Crorepati
(PYTHON)

As a part of the Informatics Practices Course (065)

SUBMITTED BY
Jay J. Patel
Harsh B. Parmar

Under the Guidance of:
Mr. Rahul Gupta

Registration No.

CERTIFICATE

This is to certify that the Project entitled K.B.C. – Kaun Banega Crorepati is a bona fide work done by Master Jay J. Patel and Harsh B. Parmar of class XII Science Session 2021-22 in partial fulfillment of CBSE's school Examination 2022 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 **Shri Jagdishbhai Thakorbhai Patel,** and my mother **Mrs. Sangitaben Jagdishbhai Patel** for being a source of inspiration and providing time and freedom to develop this software project.

CONTENTS

1. INTRODUCTION

This software project is developed to automate the functionalities of a K.B.C. – Kaun Banega Crorepati (Game). K.B.C. GAME involves enhancing 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).

K.B.C. is design to promote a fun way to study and in the process, help improve once General Knowledge. K.B.C. improve or expanding once knowledge of things. K.B.C. is to encourage players to look beyond their textual knowledge and establish a relationship between theory and application of the learnt concepts.

When players are expected to grasp the objects and fit them into proper. This improves program building abilities using Graphic as K.B.C. is graphically oriented program.

2. OBJECTIVE & SCOPE OF THE PROJECT

K.B.C. – Kaun Banega Crorepati Improve IQ Research states that solving Questions helps raise the IQ (intelligence quotient) level of the player. K.B.C. force the solver to think and reason.

K.B.C. improve **cognitive ability** in students, K.B.C. GAMES enhance 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 as per **K.B.C. Show**.
- To provide graphical and user-friendly interface to interact like a real K.B.C. Game show.
- 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 K.B.C..

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 **Kaun Banega Crorepati Simulation 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 40 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. If a audio questions comes, then we cannot hear the timer sound but the timer still goes on and after the audio question, the audio continues to play until it gets over.

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.

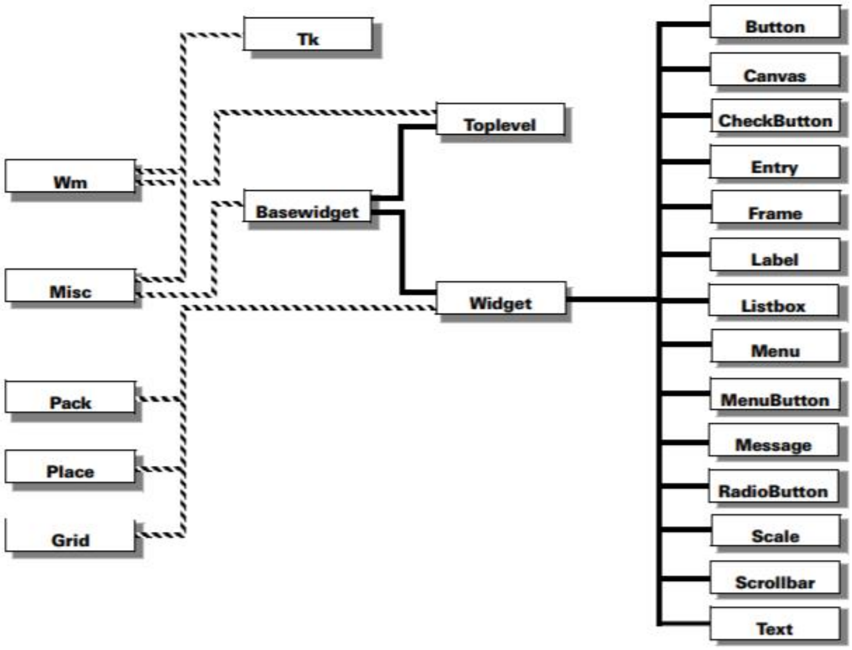
- Python is a cross-platform language. It works equally on different OS platforms like Windows, Linux, Mac OSX etc. Hence Python applications can be easily ported across OS platforms.

- 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”.

3.3 What is pygame and pyttsx3?

3.3.1 About Pygame

Pygame is a free-to-use and open-source set of Python Modules. And as the name suggests, it can be used to build games. You can code the games and then use specific commands to change it into an executable file that you can share with your friends to show them the work you have been doing. It includes computer graphics and sound libraries designed to be used with the Python programming language.

By default, Python doesn't come with PyGame as an in-built Library. So we have to install it using the command prompt. Open a command prompt and type the following command:

```
pip install pygame
```

If you already have PyGame installed, use the following command to check the version:

```
pip show pygame
```

Note: Only the mixer (pygame.mixer) sub-module is used for this software

3.3.1 About pyttsx3

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech.

The pyttsx3 module supports two voices first is female and the second is male which is provided by “sapi5” for windows.

It supports three TTS engines :

- *sapi5* – SAPI5 on Windows
- *nsss* – NSSpeechSynthesizer on Mac OS X
- *espeak* – eSpeak on every other platform

To install the pyttsx3 module, first of all, you have to open the terminal and write

```
pip install pyttsx3
```

Note: Only the simple use of engine() and setProperty() attributes are used for making the functionality of PhoneAFriend lifeline in this software

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 endeavour 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 behaviour 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 the K.B.C. Game Show, 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 the real quiz show and the lifelines should work accordingly.
- To provide the randomness of questions so the user curiosity is being maintained.
- To provide the appropriate tunes for the events such as for displaying the next questions, losing the game, winning the game etc.

5. SYSTEM IMPLEMENTATION

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

- ✓ Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz   1.19 GHz
- ✓ 1.19 GHz having 4 GB RAM and other required devices.

The **Software's** used:

- ✓ Microsoft Windows 10 Pro as Operating System
- ✓ Python 3.8.3 as programming language
- ✓ JetBrains PyCharm as an IDE for making GUI
- ✓ Visual Studio Code as an expert IDE

6. WHICH MODULES(LIBRARIES) ARE USED?

6.1 Modules List

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 tkinter.ttk.ProgressBar & tkinter.messagebox

Tkinter has many sub-module from which we used the ttk and messagebox. Also from ttk we used specifically ProgressBar.

The main purpose to use ttk.ProgressBar library is to create the polling functionality in the lifeline named Audience Poll.

The main purpose the use messagebox library is to show the pop-up window when an error comes or to show an information.

6.1.4. pygame & pyttsx3

In this software, pygame.mixer is the only sub-module from pygame library used to add voices and audio in a program/GUI.

In this software, pyytsx3 is used to convert the answer into the speed(AI audio) for making the game better and attractive.

6.1.5. random

Random module is used for random select of a sequence. We have used this module for the randomness of questions and their respective options.

The simplest way to select the random name from the list of 4 names is:

```
import random as rd
l = [“Aanad”, “Chennai”, “Delhi”, “Bandra”]
choice = rd.choice(l)
print(choice)
```

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
```

Note: In this project only the use .loc and reading csv file functions are used. By default the object which is used in reading the csv file is a **DataFrame Object.**

Simple Code to read a csv file:

```
import pandas as pd
df = pd.read_csv("filename.csv")
print(df)
```

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

```
from tkinter import *                     #for tkinter
from tkinter.ttk import PorgressBar       #for tkinter.ttk.ProgressBar
import random as rd                       #for random
import tkinter.messagebox as tmsg         #for tkinter.messagebox
from PIL import Image, IamgeTk            #for PIL
import pyttsx3                            #for pyttsx3
from pygame import mixer                  #for pygame.mixer
import pandas as pd                       #for pandas
```

6.2 Event Binding

The most important part to make a better GUI is to learn the event binding process. We can bind mouse clicks as well as the keyboard keys in a GUI which helps to command the required functions. The common syntax of binding a widget is:

```
Widget.bind(<event>, function)
```

The most used key bindings in tkinter with their uses are:

Event key	**Key binded**
Button-1	Single left click
Button-3	Single right click
Return	Enter key
ButtonRelease	Release mouse button
Enter	Mouse cursor entering in the widget
Double-Button-1	Double left click
Double-Button-3	Double right click
Leave	Mouse cursor leaving the widget

7. USER MANUAL

7.1 How to install Software:

- **Basic Hardware Requirement-**

  - ✓ Intel Atom® processor or Intel® Core™ i3 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 3.8 with the required libraries:
    - Tkinter
    - Pygame
    - Pyttsx3
    - Random
    - PIL(Python Image Library)
    - Tkinter.messagebox & tkinter.ttk.ProgressBar

You can easily download this game as we install the setup of other programs. Since it is converted into executable file, it will run on the computer irrespective of python and it's library installed or not.

Steps for installing:

1) Open the setup file
2) Select the directory where you want to install this game.
3) Make a new folder and give the name as per your wish
4) Select the folder which is made by you recently and click on Next button.
5) Your game has been installed successfully.

8. WORKING WITH SOFTWARE

8.1 Source Code(Game):

```
from tkinter import *
from pygame import mixer
from tkinter.ttk import Progressbar
import tkinter.messagebox as tmsg
from PIL import Image, ImageTk
import pyttsx3
import random as rd
import pandas as pd

mixer.init()

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)

main_questions = {"From which state does the Prime Minister of India
belongs to?":["Rajasthan", "Gujarat", "Maharashtra", "Uttar Pradesh"],
"The smallest particle in this whole world is ____":["Electron",
"Neutron", "Proton", "Quarks"], "Which state is the economical capital
of India?":["Gujarat", "Mumbai", "New Delhi", "Kerela"],
                  "Which acid is present in our stomach which helps
for digestion?":["Hydrocholoric acid", "Sulphuric acid", "Formic
acid", "Nitric acid"], "Complete the series: 3,-6,12,-24,__":["46",
"44", "48", "50"], "Which was the first spacecraft to successfully
land on moon?":["Explorer 33", "Luna 9", "Pioneer 0", "Ranger 4"],
                  "Which character owe Lord Rama in the whole
Ramayana?":["Vibhishan", "Hanuman", "Laxman", "Ravan"], "How many
countries does Reliance exports its products to?":["98", "108", "127",
"132"], "Which movie got the Oscar Award for best visual effect in
2020?":["Avenger's Endgame", "1917", "Parasite", "Joker"],
                  "Name the first month of Indian
calendar.":["Shravan", "Fagan", "Chaitra", "Vaishakha"], "The leaves
of which of the trees are shortest in the length?":["Coconut",
"Mango", "Date", "Papaya"], "In which year Python Programming language
was developed?":["1988", "1989", "1990", "1991"],
                  "On which day do we celebrate International Yoga
Day":["May 21", "June 21", "July 21", "August 21"], "Where was first
computer virus created":["Russia", "England", "Pakistan",
"California"], "White lung cancer is caused by ____":["Asbetos",
"Silica", "Textiles", "Paper"], "Who was the first Indian cricketer to
score a century in ODI?":["Sachin Tendulkar", "Kapil Dev", "Rahul
Dravid", "Sourav Gangulay"],
                 "Which part of the body does not grow from birth to
death?":["Retina", "Cornea", "Tongue", "Heart"], "Identify the TMKOC
character by the audio":["Bapuji", "Jethalal", "Bhide", "Sodhi"], "Who
among the following wrote Sanskrit grammar?":["Kalidasa", "Charak",
"Panini", "Aryabhatt"],
                 "The metal whose salts are sensitive to light
is?":["Zinc", "Copper", "Silver", "Aluminium"], "Which peninsular
river is least seasonal in flow?":["Narmada", "Krishna", "Godavari",
"Cauvery"], "Fastest shorthand writer was":["Dr. G. D. Bist", "J.M.
Tagore", "Khudada Khan", "J.R.D. Tata"],
                 "Epsom (England) is the place associated
with":["Shooting", "Polo", "Snooker", "Horse Racing"], "During World
War II, when did Germany attack France?":["1940", "1941", "1942",
"1943"], "Which is the religion for which the Fire temple is the place
of worship?":["Hinduism", "Jainism", "Zoroastrianism", "Buddhism"],
                 "The United Nations Organization has its
Headquarters at":["Bali", "Hague", "New York, USA", "Washington DC"],
"Curie the a unit of which of the following?":["Luminescence",
"Radioactivity", "Pressure", "Mass"], "Which Indian state had the
first woman Chief Minister":["Uttar Pradesh", "Madhya Pradesh",
"Sikkim", "Himachal Pradesh"],
                 "Maximum number of Famines attacked India
during__?":["1750-1800", "1800-1850", "1850-1900", "1900-1950"],
"Garvi Gujarat Bhavan, which is in news recently, is located in which
city?":["Chandigarh", "New Delhi", "Gandhinagar","Ahmedabad"],
"Aishwarya Rai was crowned Miss World in which year?":["1994", "1995",
"1996", "1993"],
                 "Which bollywood film has highest number of
songs?":["Neel Kamal", "Indra Sabha", "Alam Ara", "Kishan Kanhaiya"],
"When was first train started in India?":["1851", "1852", "1853",
"1854"], "What is the width of broad gauge railway line in India?":["5
feet 3 inches", "5 feet 6 inches", "4 feet 1 inches", "5 feet 4
inches"],
                 "Of the ancient six philosophical systems of ancient
India, which is the oldest one?":["Yoga", "Vedanta", "Nyaya",
"Samkhya"], "Finance bill of Indian Government is presented
in":["Upper House", "Middle House", "Lower House", "Raw House"],
                 "The world's smallest country is ____":["Moscow",
"Mexico", "Vatican City", "Bishop Rock"], "The National aquatic animal
is ____":["Dolphin", "Crocodile", "Fish", "Whale"], "World Tourism Day
is celebrated on":["September 12", "September 29", "September 27",
"September 25"],
                  "The super computer ‘PARAM’ was developed
by":["TATA", "IIT-Kharagpur", "IIT-Kanpur", "C-DAC"]}

correct_dict = {"From which state does the Prime Minister of India
belongs to?":"Gujarat", "The smallest particle in this whole world is
____":"Quarks", "Which state is the economical capital of
India?":"Mumbai", "Which acid is present in our stomach which helps
for digestion?":"Hydrocholoric acid", "Complete the series: 3,-6,12,-
24,__":"48",
                "Which was the first spacecraft to successfully land
on moon?":"Luna 9", "Which character owe Lord Rama in the whole
Ramayana?":"Hanuman", "How many countries does Reliance exports its
products to?":"108",
                "Which movie got the Oscar Award for best visual
effect in 2020?":"1917", "Name the first month of Indian
calendar.":"Chaitra", "The leaves of which of the trees are shortest
in the length?":"Mango",
                "In which year Python Programming language was
developed?":"1989", "On which day do we celebrate International Yoga
Day":"June 21", "Where was first computer virus created":"Pakistan",
                "White lung cancer is caused by ____":"Textiles", "Who
was the first Indian cricketer to score a century in ODI?":"Kapil
Dev", "Which part of the body does not grow from birth to
death?":"Cornea", "Identify the TMKOC character by the
audio":"Bapuji",
                "Who among the following wrote Sanskrit
grammar?":"Panini", "The metal whose salts are sensitive to light
is?":"Aluminium", "Which peninsular river is least seasonal in
flow?":"Godavari", "Fastest shorthand writer was":"Dr. G. D. Bist",
                "Epsom (England) is the place associated with":"Horse
Racing", "During World War II, when did Germany attack
France?":"1940", "Which is the religion for which the Fire temple is
the place of worship?":"Zoroastrianism", "The United Nations
Organization has its Headquarters at":"New York, USA",
                "Curie the a unit of which of the
following?":"Radioactivity", "Which Indian state had the first woman
Chief Minister":"Uttar Pradesh", "Maximum number of Famines attacked
India during__?":"1850-1900", "Garvi Gujarat Bhavan, which is in news
recently, is located in which city?":"New Delhi",
                "Aishwarya Rai was crowned Miss World in which
year?":"1994", "Which bollywood film has highest number of
songs?":"Indra Sabha", "When was first train started in
India?":"1853", "What is the width of broad gauge railway line in
India?":"5 feet 6 inches",
                "Of the ancient six philosophical systems of ancient
India, which is the oldest one?":"Samkhya", "Finance bill of Indian
Government is presented in":"Lower House", "The world's smallest
country is ____":"Vatican City", "The National aquatic animal is
____":"Dolphin",
                "World Tourism Day is celebrated on":"September 27",
"The super computer ‘PARAM’ was developed by":"C-DAC"}
question = []
game_questions = []

first_option = []
second_option = []
third_option = []
fourth_option = []
correct = []

pool_list = ["Rs. 5,000", "Rs. 10,000", "Rs. 20,000", "Rs. 40,000",
"Rs. 80,000", "Rs. 1,60,000", "Rs. 3,20,000", "Rs. 6,40,000", "Rs.
12,50,000", "Rs. 25 Lakh", "Rs. 50 Lakh", "Rs. 1 Crore", "Rs. 3
Crore", "Rs. 5 Crore", "Rs. 7 Crore"]
timer_list = []
contestant_name = None

def start_game(event):
    global contestant_name
    contestant_name = namevalue.get()

    if contestant_name == None or contestant_name == "":
        tmsg.showerror("KBC - Kaun Banega Crorepati", "Please enter
your name....")
        root.destroy()

    else:
        name.destroy()
```

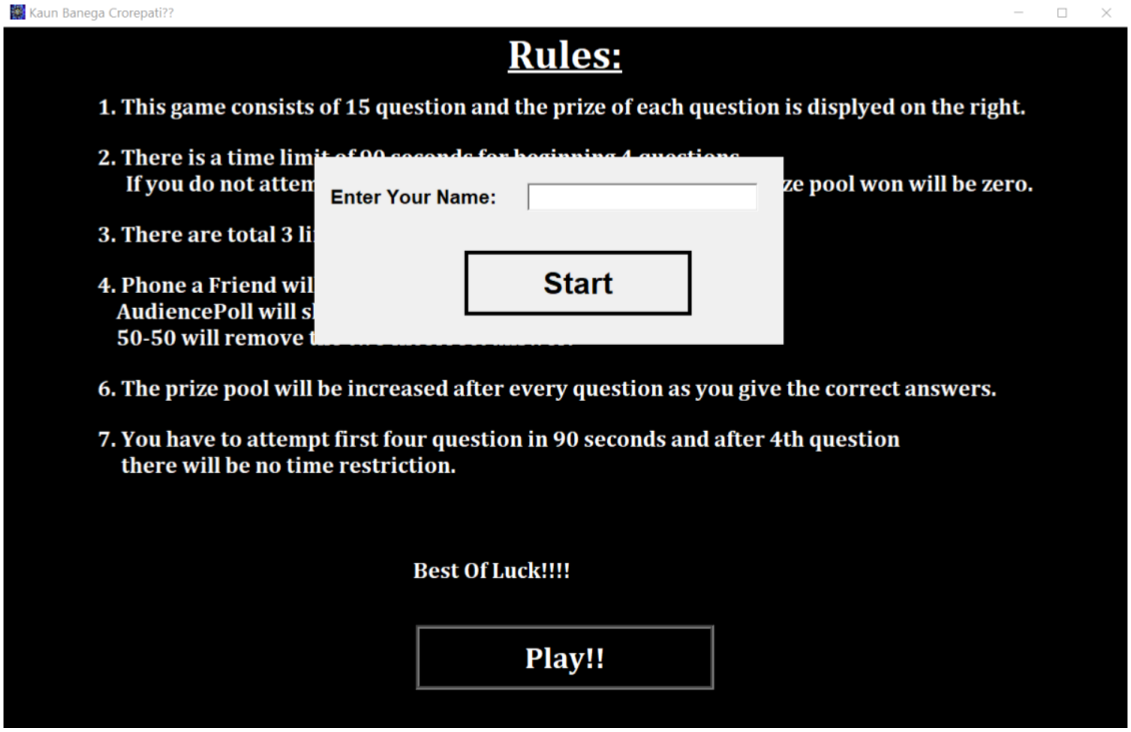
```
root = Tk()
root.title("Kaun Banega Crorepati")
root.geometry("1200x750+100+30")
root.wm_iconbitmap("resources\\kbc icon.ico")
root.config(bg="black")

name = Toplevel()
name.wm_iconbitmap("resources\\kbc icon.ico")
name.geometry("500x200+440+200")
name.overrideredirect(True)

name_label = Label(name, text="Enter Your Name:", font="arial 16
bold", relief=FLAT, bd=0)
name_label.grid(row=0, column=0, pady=28, padx=16)

namevalue = StringVar()
ask_name = Entry(name, width=20, textvariable=namevalue, font="arial
16 bold", bd=2, relief=SUNKEN)
ask_name.bind('<Return>', start_game)
ask_name.grid(row=0, column=1, pady=28, padx=16)


play_button = Button(name, text="Start", font="arial 24 bold",
width=12, relief=SOLID, bd=4, cursor="hand2")
play_button.bind('<Button-1>', start_game)
play_button.place(x=160, y=100)

class RulesWindow:
    mixer.music.load("resources\\kbc.mp3")
    mixer.music.play(-1)

    def __init__(self, root):
        self.window = Frame(root, bg="black")
        self.window.pack()
        self.rules()

    def rules(self):
        df = pd.read_csv("questions.csv")
        rule_label = Label(self.window, text="Rules:",
font=("cambria", 32, "bold", "underline"), bg="black", fg="white",
                           relief=FLAT, justify=CENTER)
        rule_label.pack(fill=X, side=TOP)

        rule1 = Label(self.window,
                      text=f"{df.iat[0, 0]}",
                      font=("cambria", 18, "bold"), bg="black",
fg="white", relief=FLAT, justify=LEFT)
        rule2 = Label(self.window, text=f"{df.iat[1, 0]}\n
{df.iat[2, 0]}",
                      font=("cambria", 18, "bold"), bg="black",
fg="white", relief=FLAT, justify=LEFT)
        rule3 = Label(self.window, text=f"{df.iat[3, 0]}, {df.iat[4,
0]}, {df.iat[5, 0]}",
                      font=("cambria", 18, "bold"), bg="black",
fg="white", relief=FLAT, justify=LEFT)
        rule4 = Label(self.window,
                      text=f"{df.iat[6, 0]}\n    {df.iat[7, 0]}\n
{df.iat[8, 0]}",
                      font=("cambria", 18, "bold"), bg="black",
fg="white", relief=FLAT, justify=LEFT)
        rule5 = Label(self.window,
                      text=f"{df.iat[9, 0]}",
                      font=("cambria", 18, "bold"), bg="black",
fg="white", relief=FLAT, justify=LEFT)
        rule6 = Label(self.window, text=f"{df.iat[10, 0]}\n
{df.iat[11, 0]}\n\n\n\n\t\t\t{df.iat[12, 0]}",
                      font=("cambria", 18, "bold"), bg="black",
fg="white", relief=FLAT, justify=LEFT)

        enter_button = Button(self.window, text="Play!!", width=16,
font=("cambria", 24, "bold"), bd=4, relief=GROOVE,
                              fg="white", bg="black",
activebackground="black", activeforeground="white",
                              cursor="hand2")
        enter_button.bind('<Button-1>', self.play)
        enter_button.pack(side=BOTTOM, pady=32)

        rule1.pack(pady=10, anchor=W, padx=48)
        rule2.pack(pady=10, anchor=W, padx=48)
        rule3.pack(pady=10, anchor=W, padx=48)
        rule4.pack(pady=10, anchor=W, padx=48)
        rule5.pack(pady=10, anchor=W, padx=48)
        rule6.pack(pady=10, anchor=W, padx=48)

    def play(self, event):
        if contestant_name == "" or contestant_name == None:
            tmsg.showerror("KBC - Kaun Banega Crorepati", "Please
enter your name....")
            root.destroy()

        else:
            self.window.destroy()
            mixer.music.stop()
            game = MainGame(root)

class MainGame:
    def __init__(self, root):

        question.clear()
        game_questions.clear()
        first_option.clear()
        second_option.clear()
        third_option.clear()
        fourth_option.clear()
        correct.clear()

        for que in main_questions:
            question.append(que)

        ask_question = rd.sample(range(len(question)), 15)

        for i in ask_question:
            game_questions.append(question[i])
            options = main_questions[question[i]]
            right_options = correct_dict[question[i]]

            option_index = rd.sample(range(4), 4)
            first_option.append(options[option_index[0]])
            second_option.append(options[option_index[1]])
            third_option.append(options[option_index[2]])
            fourth_option.append(options[option_index[3]])
            correct.append(right_options)

        for timer in range(0, 4):
            timer_list.append(game_questions[timer])

        leftframe = Frame(root, bg="black")
        leftframe.grid(row=0, column=0)

        rightframe = Frame(root, bg="black")
        rightframe.grid(row=0, column=1)

        lifelineframe = Frame(leftframe, bg="black")
        lifelineframe.grid(row=0, column=0, pady=16, padx=30)

        logoframe = Frame(leftframe, bg="black")
        logoframe.grid(row=1, column=0)

        questionframe = Frame(leftframe, bg="black")
        questionframe.grid(row=2, column=0)

        # Images:
        self.life50 = PhotoImage(file="resources\\50-50.png")
        self.audipoll = PhotoImage(file="resources\\audiencePole.png")
        self.phoneafriend =
PhotoImage(file="resources\\phoneafriend.png")
        self.mainlogo = PhotoImage(file="resources\\kbc_logo.png")
        layoutimage = PhotoImage(file="resources\\lay.png")
        self.amountimage =
ImageTk.PhotoImage(Image.open("resources\\question0.jpg"))
        self.cross50 = PhotoImage(file="resources\\50-50-X.png")
        self.crossaudi =
PhotoImage(file="resources\\audiencePoleX.png")
        self.crossphone =
PhotoImage(file="resources\\phoneafriendX.png")
        prize1 =
ImageTk.PhotoImage(Image.open("resources\\question1.jpg"))
        prize2 =
ImageTk.PhotoImage(Image.open("resources\\question2.jpg"))
        prize3 =
ImageTk.PhotoImage(Image.open("resources\\question3.jpg"))
```

```
        prize4 =
ImageTk.PhotoImage(Image.open("resources\\question4.jpg"))
        prize5 =
ImageTk.PhotoImage(Image.open("resources\\question5.jpg"))
        prize6 =
ImageTk.PhotoImage(Image.open("resources\\question6.jpg"))
        prize7 =
ImageTk.PhotoImage(Image.open("resources\\question7.jpg"))
        prize8 =
ImageTk.PhotoImage(Image.open("resources\\question8.jpg"))
        prize9 =
ImageTk.PhotoImage(Image.open("resources\\question9.jpg"))
        prize10 =
ImageTk.PhotoImage(Image.open("resources\\question10.jpg"))
        prize11 =
ImageTk.PhotoImage(Image.open("resources\\question11.jpg"))
        prize12 =
ImageTk.PhotoImage(Image.open("resources\\question12.jpg"))
        prize13 =
ImageTk.PhotoImage(Image.open("resources\\question13.jpg"))
        prize14 =
ImageTk.PhotoImage(Image.open("resources\\question14.jpg"))
        prize15 =
ImageTk.PhotoImage(Image.open("resources\\question15.jpg"))
        self.callimage = PhotoImage(file="resources\\phone.png")
        self.soundimage = PhotoImage(file="resources\\sound.png")
        self.timerimage =
ImageTk.PhotoImage(Image.open("resources\\clock.jpg"))

        self.prizelist = [prize1, prize2, prize3, prize4, prize5,
prize6, prize7, prize8, prize9, prize10, prize11, prize12,
                     prize13, prize14, prize15]

        # Adding picture:
        self.button50 = Button(lifelineframe, image=self.life50,
bg="black", activebackground="black", relief=FLAT, bd=0,
                          width=180, height=80, cursor="hand2",
command=self.lifeline50)
        self.button50.image = self.life50
        self.button50.grid(row=0, column=0)
        self.buttonpoll = Button(lifelineframe, image=self.audipoll,
bg="black", activebackground="black", relief=FLAT, bd=0,
                            width=180, height=80, cursor="hand2",
command=self.audiencepoll)
        self.buttonpoll.image = self.audipoll
        self.buttonpoll.grid(row=0, column=1)
        self.buttonfriend = Button(lifelineframe,
image=self.phoneafriend, bg="black", activebackground="black",
relief=FLAT,
                              bd=0, width=180, height=80,
cursor="hand2", command=self.phonelifeline)
        self.buttonfriend.image = self.phoneafriend
        self.buttonfriend.grid(row=0, column=2)

        logo_label = Label(logoframe, image=self.mainlogo, bg="black",
activebackground="black", relief=FLAT, bd=0,
                           width=320, height=320)
        logo_label.image = self.mainlogo
        logo_label.pack(padx=60)

        layout = Label(questionframe, image=layoutimage, bg="black")
        layout.image = layoutimage
        layout.pack(padx=100)

        self.amount = Label(rightframe, image=self.amountimage,
bg="black", width=300, height=700)
        self.amount.image = self.amountimage
        self.amount.pack(pady=24, padx=50)

        # Adding text areas:
        self.questionarea = Text(questionframe, font="arial 18 bold",
width=32, height=2, wrap="word", bg="black",
                            fg="white", relief=FLAT, bd=0)
        self.questionarea.place(x=180, y=10)
        self.questionarea.insert(END, game_questions[0])

        labelA = Label(questionframe, text="A.", font="arial 16 bold",
bg="black", fg="white")
        labelA.place(x=150, y=110)
        labelB = Label(questionframe, text="B.", font="arial 16 bold",
bg="black", fg="white")
```

```
        labelB.place(x=430, y=110)
        labelC = Label(questionframe, text="C.", font="arial 16 bold",
bg="black", fg="white")
        labelC.place(x=150, y=195)
        labelD = Label(questionframe, text="D.", font="arial 16 bold",
bg="black", fg="white")
        labelD.place(x=430, y=195)

        self.button1 = Button(questionframe, text=first_option[0],
font="arial 14 bold", bg="black", fg="white", relief=FLAT,
                         activebackground="black",
activeforeground="white", cursor="hand2", bd=0)
        self.button1.place(x=188, y=105)
        self.button2 = Button(questionframe, text=second_option[0],
font="arial 14 bold", bg="black", fg="white",
                         relief=FLAT, activebackground="black",
activeforeground="white", cursor="hand2", bd=0)
        self.button2.place(x=460, y=105)
        self.button3 = Button(questionframe, text=third_option[0],
font="arial 14 bold", bg="black", fg="white", relief=FLAT,
                         activebackground="black",
activeforeground="white", cursor="hand2", bd=0)
        self.button3.place(x=188, y=192)
        self.button4 = Button(questionframe, text=fourth_option[0],
font="arial 14 bold", bg="black", fg="white",
                         relief=FLAT, activebackground="black",
activeforeground="white", cursor="hand2", bd=0)
        self.button4.place(x=460, y=191)

        self.button1.bind('<Button-1>', self.select)
        self.button2.bind('<Button-1>', self.select)
        self.button3.bind('<Button-1>', self.select)
        self.button4.bind('<Button-1>', self.select)

        self.progressbarA = Progressbar(root, orient=VERTICAL,
length=120)
        self.progressbarB = Progressbar(root, orient=VERTICAL,
length=120)
        self.progressbarC = Progressbar(root, orient=VERTICAL,
length=120)
```

```
        self.progressbarD = Progressbar(root, orient=VERTICAL,
length=120)

        self.progressbarlabelA = Label(root, text="A", font="arial 18
bold", bg="black", fg="white")
        self.progressbarlabelB = Label(root, text="B", font="arial 18
bold", bg="black", fg="white")
        self.progressbarlabelC = Label(root, text="C", font="arial 18
bold", bg="black", fg="white")
        self.progressbarlabelD = Label(root, text="D", font="arial 18
bold", bg="black", fg="white")

        self.callbutton = Button(root, image=self.callimage,
relief=FLAT, bd=0, bg="black", activebackground="black",
                            cursor="hand2", command=self.calling)

        self.audiobutton = Button(root, image=self.soundimage,
relief=FLAT, bd=0, bg="black", activebackground="black",
                             cursor="hand2", command=self.sound)

        self.clock = Label(root, image=self.timerimage, bd=0,
relief=FLAT, activebackground="black", height=120)
        self.clock.image = self.timerimage
        self.clock.place(x=600, y=350)

        self.countdown_label = Label(root, font=("cambria", 24,
"bold"), width=2, bg="black", fg="white")
        self.countdown_label.place(x=664, y=388)
        self.countdown(90)
        mixer.music.load("resources\\timer 2.mp3")
        mixer.music.play(6)

    def countdown(self, count):
        # change text in label
        self.countdown_label["text"] = count

        if count > 0:
            # call countdown again after 1000ms (1s)
            root.after(1000, self.countdown, count - 1)

        try:
            if count == 0:
                try:
                    if self.questionarea.get(1.0, "end-1c") in
timer_list:
                        self.gameover()

                except Exception as e:
                    pass

        except Exception as e:
                    pass

    def lifeline50(self):
        self.button50.config(image=self.cross50, state=DISABLED)

        button_option = [self.button1["text"], self.button2["text"],
self.button3["text"], self.button4["text"]]
        option_list = correct_dict[self.questionarea.get(1.0, "end-
1c")]
        delete_list = []

        random50 = rd.sample(range(4), 4)
        for option in random50:
            if button_option[option] != option_list:
                delete_list.append(button_option[option])

        final_random = rd.sample(range(3), 2)

        for final in final_random:
            if delete_list[final] in button_option:
                number = (button_option.index(delete_list[final]))

                if self.button1["text"] == button_option[number]:
                    self.button1.config(text="")
                elif self.button2["text"] == button_option[number]:
                    self.button2.config(text="")
                elif self.button3["text"] == button_option[number]:
                    self.button3.config(text="")
                elif self.button4["text"] == button_option[number]:
                    self.button4.config(text="")
    def audiencepoll(self):
        self.buttonpoll.config(image=self.crossaudi, state=DISABLED)

        button_list = [self.button1["text"], self.button2["text"],
self.button3["text"], self.button4["text"]]
        option_list = correct_dict[self.questionarea.get(1.0, "end-
1c")]
        poll_list = []
        deduced_poll_list = []

        choice1 = rd.randrange(48, 98)
        choice2 = rd.randrange(48, 98)
        choice3 = rd.randrange(48, 98)
        choice4 = rd.randrange(48, 98)

        poll_list.append(choice1)
        poll_list.append(choice2)
        poll_list.append(choice3)
        poll_list.append(choice4)

        if self.button1["text"] == option_list:
            self.progressbarA.config(value=max(poll_list))
            poll_list.remove(max(poll_list))
            self.progressbarB.config(value=poll_list[0])
            self.progressbarC.config(value=poll_list[1])
            self.progressbarD.config(value=poll_list[2])

        elif self.button2["text"] == option_list:
            self.progressbarB.config(value=max(poll_list))
            poll_list.remove(max(poll_list))
            self.progressbarA.config(value=poll_list[0])
            self.progressbarC.config(value=poll_list[1])
            self.progressbarD.config(value=poll_list[2])

        elif self.button3["text"] == option_list:
            self.progressbarC.config(value=max(poll_list))
            poll_list.remove(max(poll_list))
            self.progressbarB.config(value=poll_list[0])
            self.progressbarA.config(value=poll_list[1])
            self.progressbarD.config(value=poll_list[2])
        elif self.button4["text"] == option_list:
            self.progressbarD.config(value=max(poll_list))
            poll_list.remove(max(poll_list))
            self.progressbarB.config(value=poll_list[0])
            self.progressbarC.config(value=poll_list[1])
            self.progressbarA.config(value=poll_list[2])

        self.progressbarA.place(x=550, y=180)
        self.progressbarB.place(x=590, y=180)
        self.progressbarC.place(x=630, y=180)
        self.progressbarD.place(x=670, y=180)

        self.progressbarlabelA.place(x=550, y=310)
        self.progressbarlabelB.place(x=590, y=310)
        self.progressbarlabelC.place(x=630, y=310)
        self.progressbarlabelD.place(x=670, y=310)

    def winner(self, name, pool):
        self.reward = Toplevel()
        self.reward.geometry("1120x510+120+100")
        self.reward.config(bg="#2935a3")
        self.reward.overrideredirect(True)

        self.reward_image =
ImageTk.PhotoImage(Image.open("resources\\reward.jpg"))

        label = Label(self.reward, image=self.reward_image)
        label.pack()

        name_label = Label(self.reward, text=name, font="arial 32
bold", relief=FLAT, fg="white", bg="#2f46ca")
        name_label.place(x=420, y=250)
        prize_label = Label(self.reward, text=pool, font="arial 32
bold", relief=FLAT, fg="white", bg="#2f46ca")
        prize_label.place(x=420, y=350)

        play_again = Button(self.reward, text="Play Again", width=18,
font="arial 28 bold", bd=0, fg="white", bg="#2935a3",
activebackground="#2935a3", activeforeground="white", cursor="hand2",
command=self.again)
        play_again.pack(side=LEFT, padx=90)
        exit_button = Button(self.reward, text="Exit Game", width=20,
font="arial 28 bold", bd=0, fg="white", bg="#2935a3",
activebackground="#2935a3", activeforeground="white", cursor="hand2",
command=root.destroy)
        exit_button.pack(side=RIGHT, padx=90)

        mixer.music.stop()
        mixer.music.load("resources\\win.mp3")
        mixer.music.play()

        label.mainloop()

    def again(self):
        self.reward.destroy()
        mixer.music.stop()
        play = MainGame(root)

    def again2(self):
        self.exit_window.destroy()
        mixer.music.stop()
        play = MainGame(root)

    def calling(self):
        for i in range(15):
            if self.questionarea.get(1.0, "end-1c") ==
game_questions[i]:
                engine.say(f"The correct answer is {correct[i]}")
                engine.runAndWait()

    def gameover(self):
        mixer.music.stop()
        mixer.music.load("resources\\game over.mp3")
        mixer.music.play()
        self.exit_window = Toplevel()
        self.exit_window.title("Kaun Banega Crorepati??")
        self.exit_window.geometry("500x500+200+100")
        self.exit_window.config(bg="black")

        image = PhotoImage(file="resources\\sad.png")
        logo_label = Label(self.exit_window, image=self.mainlogo,
bg="black", activebackground="black", relief=FLAT, bd=0,
                           width=320, height=320)
        logo_label.pack(padx=20)

        lose_label = Label(self.exit_window, text="You Lose!!",
font="arial 36 bold", bg="black", fg="white")
        lose_label.pack(pady=5)

        tryagain_button = Button(self.exit_window, text="Try Again",
font="arial 32 bold", bg="black", fg="white",
                                 relief=FLAT,
activebackground="black", activeforeground="white",
                                 cursor="hand2", bd=0,
command=self.again2)
        tryagain_button.pack(pady=10)

        sad1 = Label(self.exit_window, image=image, bg="black")
        sad1.place(x=20, y=325)
        sad2 = Label(self.exit_window, image=image, bg="black")
        sad2.place(x=400, y=320)

        self.exit_window.mainloop()

    def sound(self):
        mixer.music.stop()
        self.audiobutton.image = self.soundimage
        self.audiobutton.place(x=50, y=350)
        mixer.music.load("resources\\audio question.mp3")
        mixer.music.play()

    def phonelifeline(self):
        mixer.music.load("resources\\calling.mp3")
        mixer.music.play()
        self.buttonfriend.config(image=self.crossphone,
state=DISABLED)
        self.callbutton.image = self.callimage
        self.callbutton.place(x=50, y=250)

    def select(self, event):
        self.progressbarA.place_forget()
        self.progressbarB.place_forget()
        self.progressbarC.place_forget()
        self.progressbarD.place_forget()
        self.progressbarlabelA.place_forget()
        self.progressbarlabelB.place_forget()
        self.progressbarlabelC.place_forget()
        self.progressbarlabelD.place_forget()
        self.callbutton.place_forget()
        self.audiobutton.place_forget()

        button = event.widget.cget("text")

        for i in range(15):
            if button == correct[i]:
                if button == correct[14]:
                    mixer.music.stop()
                    self.winner(contestant_name, pool_list[14])

                mixer.Channel(0).play(mixer.Sound("resources\\next
question.mp3"))
                try:
                    if self.questionarea.get(1.0, "end-1c") == "From
which state does the Prime Minister of India belongs to?":
                        if button == "Mumbai":
                            if self.questionarea.get(1.0, "end-1c") in
timer_list:
                                self.gameover()
                            else:
                                pointer =
game_questions.index(self.questionarea.get(1.0, "end-1c"))
                                self.winner(contestant_name,
pool_list[pointer])

                    elif self.questionarea.get(1.0, "end-1c") ==
"Which state is the economical capital of India?":
                        if button == "Gujarat":
                            if self.questionarea.get(1.0, "end-1c") in
timer_list:
                                self.gameover()
                            else:
                                pointer =
game_questions.index(self.questionarea.get(1.0, "end-1c"))
                                self.winner(contestant_name,
pool_list[pointer])

                    elif
game_questions.index(self.questionarea.get(1.0, "end-1c")) == 4:
                        mixer.music.stop()
                        self.countdown_label.place_forget()
                        self.clock.destroy()

                    self.questionarea.delete(1.0, END)
                    self.questionarea.insert(END, game_questions[i +
1])

                    self.button1.config(text=first_option[i + 1])
                    self.button2.config(text=second_option[i + 1])
                    self.button3.config(text=third_option[i + 1])
                    self.button4.config(text=fourth_option[i + 1])
                    self.amount.config(image=self.prizelist[i])

                    if "helps for digestion" in
self.questionarea.get(1.0, "end-1c"):
                        if "Hydro" in self.button1["text"]:
                            self.button1.config(font="arial 13 bold")
                            self.button1.place(x=187, y=105)
                        elif "Hydro" in self.button2["text"]:
                            self.button2.config(font="arial 13 bold")
                        elif "Hydro" in self.button3["text"]:
                            self.button3.config(font="arial 13 bold")
                        elif "Hydro" in self.button4["text"]:
                            self.button4.config(font="arial 13 bold")
                            self.button4.place(x=460, y=191)

                    elif "TMKOC" in self.questionarea.get(1.0, "end-
1c"):
                        if "Bapuji" in self.button1["text"]:
                            self.button1.config(font="arial 14 bold")
                        elif "Bapuji" in self.button2["text"]:
                            self.button2.config(font="arial 14 bold")
                        elif "Bapuji" in self.button3["text"]:
                            self.button3.config(font="arial 14 bold")
                        elif "Bapuji" in self.button4["text"]:
                            self.button4.config(font="arial 14 bold")
                        mixer.music.stop()
                        self.sound()

                    elif "Oscar" in self.questionarea.get(1.0, "end-
1c"):
                        if "Endgame" in self.button1["text"]:
                            self.button1.config(font="arial 13 bold")
                            self.button1.place(x=187, y=105)
                        elif "Endgame" in self.button2["text"]:
                            self.button2.config(font="arial 13 bold")
                        elif "Endgame" in self.button3["text"]:
                            self.button3.config(font="arial 13 bold")
                        elif "Endgame" in self.button4["text"]:
                            self.button4.config(font="arial 13 bold")
                            self.button4.place(x=460, y=191)

                    else:
                        self.button1.config(font="arial 14 bold")
                        self.button2.config(font="arial 14 bold")
                        self.button3.config(font="arial 14 bold")
                        self.button4.config(font="arial 14 bold")
                        self.button1.place(x=188, y=105)
                        self.button4.place(x=460, y=191)

                except Exception as e:
                    pass

            elif button not in correct:
                try:
                    if self.questionarea.get(1.0, "end-1c") ==
game_questions[0]:
                        self.gameover()

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[1]:
                        self.gameover()
                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[2]:
                        self.gameover()

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[3]:
                        self.gameover()

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[4]:
                        self.winner(contestant_name, pool_list[3])

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[5]:
                        self.winner(contestant_name, pool_list[4])

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[6]:
                        self.winner(contestant_name, pool_list[5])

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[7]:
                        self.winner(contestant_name, pool_list[6])

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[8]:
                        self.winner(contestant_name, pool_list[7])

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[9]:
                        self.winner(contestant_name, pool_list[8])

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[10]:
                        self.winner(contestant_name, pool_list[9])

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[11]:
                        self.winner(contestant_name, pool_list[10])
                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[12]:
                        self.winner(contestant_name, pool_list[11])

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[13]:
                        self.winner(contestant_name, pool_list[12])

                    elif self.questionarea.get(1.0, "end-1c") ==
game_questions[14]:
                        self.winner(contestant_name, pool_list[13])

                except Exception as e:
                    pass

master = RulesWindow(root)

root.mainloop()
```

8.2 Screenshots:



Rules:

1. This game consists of 15 question and the prize of each question is displyed on the right.

2. There is a time lim
If you do not attem ze pool won will be zero.

3. There are total 3 li

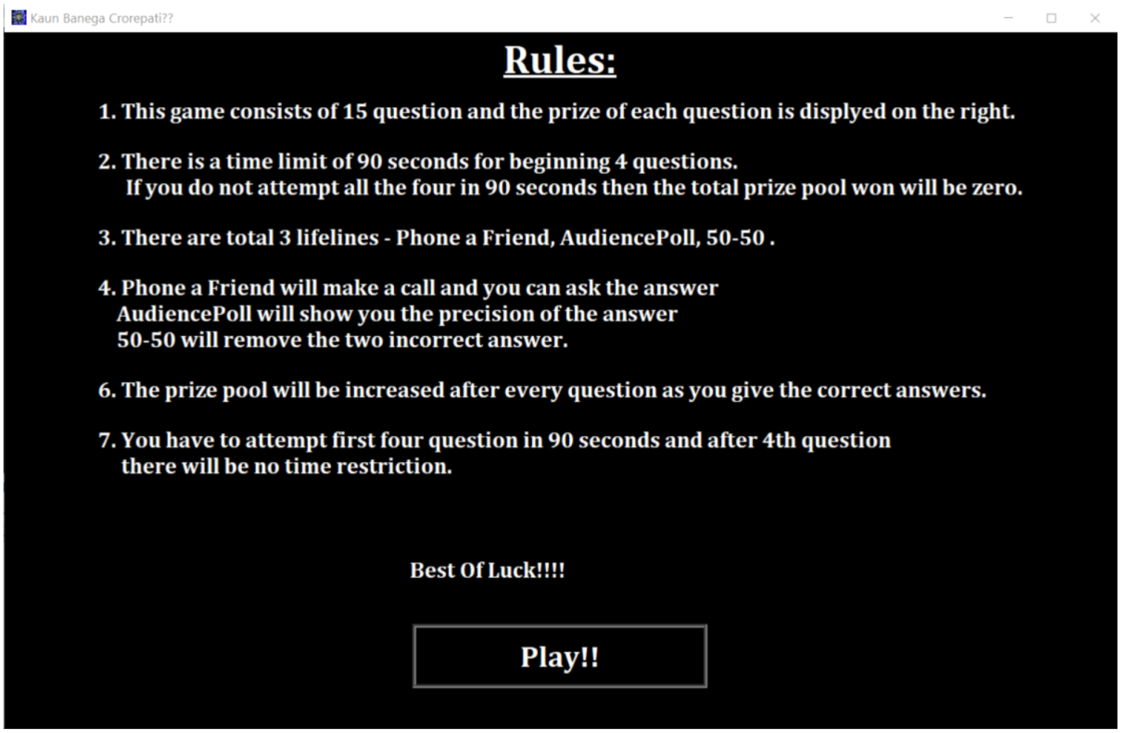
4. Phone a Friend wil
AudiencePoll will s
50-50 will remove t

Enter Your Name:

6. The prize pool will be increased after every question as you give the correct answers.

7. You have to attempt first four question in 90 seconds and after 4th question
there will be no time restriction.

Best Of Luck!!!!



Rules:

1. This game consists of 15 question and the prize of each question is displyed on the right.

2. There is a time limit of 90 seconds for beginning 4 questions.
If you do not attempt all the four in 90 seconds then the total prize pool won will be zero.

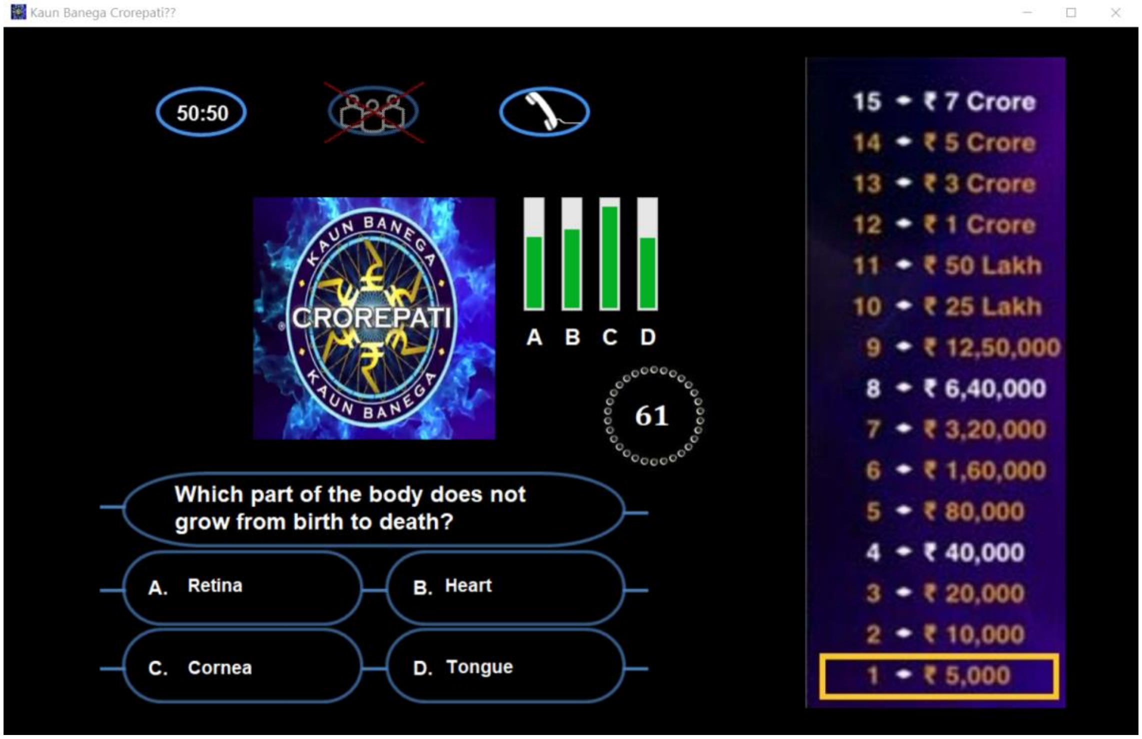
3. There are total 3 lifelines - Phone a Friend, AudiencePoll, 50-50 .

4. Phone a Friend will make a call and you can ask the answer
AudiencePoll will show you the precision of the answer
50-50 will remove the two incorrect answer.

6. The prize pool will be increased after every question as you give the correct answers.

7. You have to attempt first four question in 90 seconds and after 4th question
there will be no time restriction.

Best Of Luck!!!!



Which part of the body does not grow from birth to death?

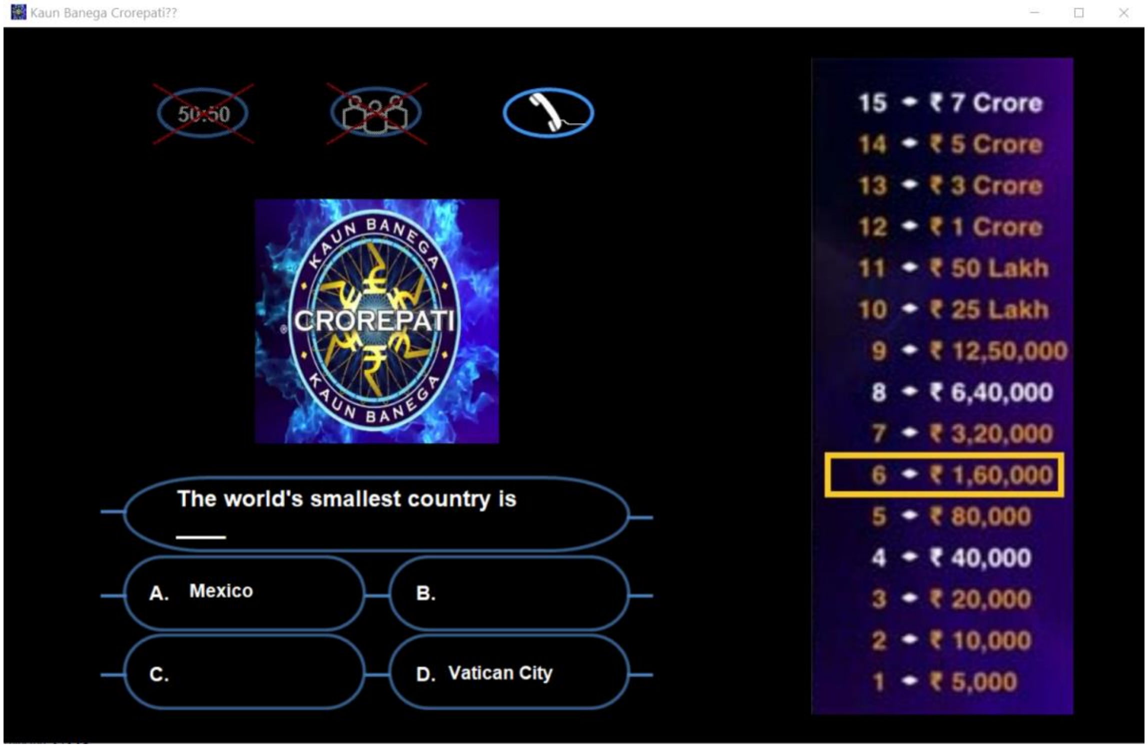
A. Retina

B. Heart

C. Cornea

D. Tongue

15 • ₹ 7 Crore
14 • ₹ 5 Crore
13 • ₹ 3 Crore
12 • ₹ 1 Crore
11 • ₹ 50 Lakh
10 • ₹ 25 Lakh
9 • ₹ 12,50,000
8 • ₹ 6,40,000
7 • ₹ 3,20,000
6 • ₹ 1,60,000
5 • ₹ 80,000
4 • ₹ 40,000
3 • ₹ 20,000
2 • ₹ 10,000
1 • ₹ 5,000



The world's smallest country is ____

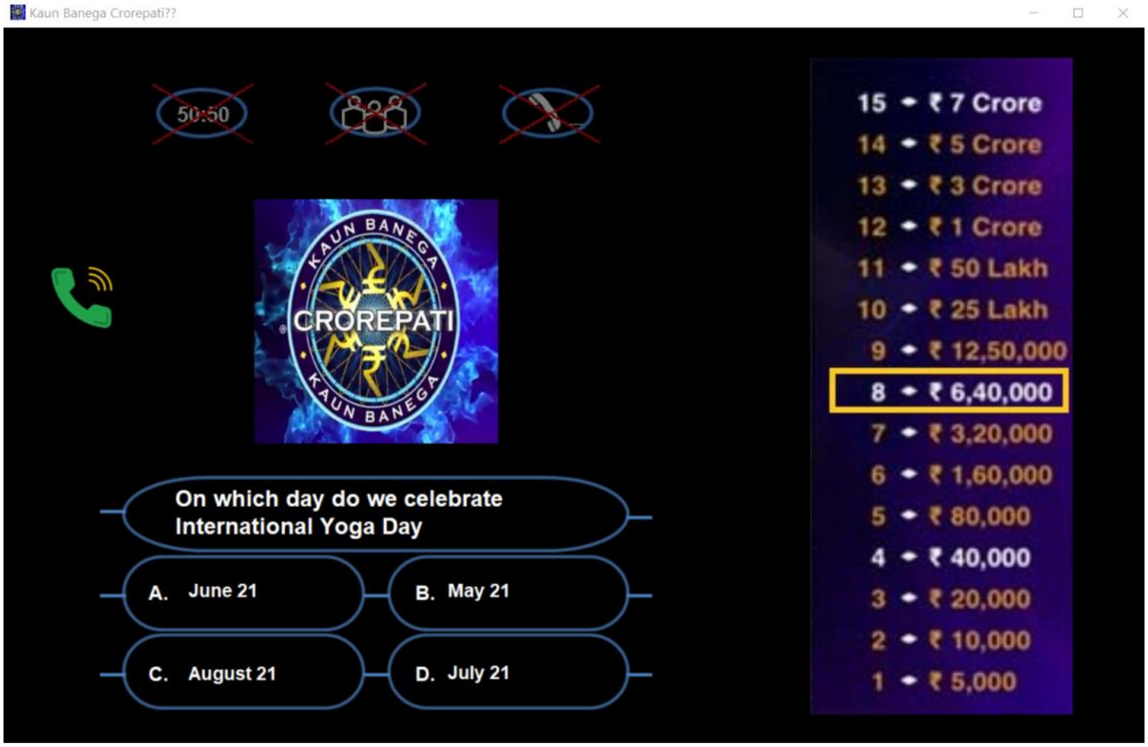
A. Mexico

B.

C.

D. Vatican City

15 • ₹ 7 Crore
14 • ₹ 5 Crore
13 • ₹ 3 Crore
12 • ₹ 1 Crore
11 • ₹ 50 Lakh
10 • ₹ 25 Lakh
9 • ₹ 12,50,000
8 • ₹ 6,40,000
7 • ₹ 3,20,000
6 • ₹ 1,60,000
5 • ₹ 80,000
4 • ₹ 40,000
3 • ₹ 20,000
2 • ₹ 10,000
1 • ₹ 5,000



On which day do we celebrate International Yoga Day

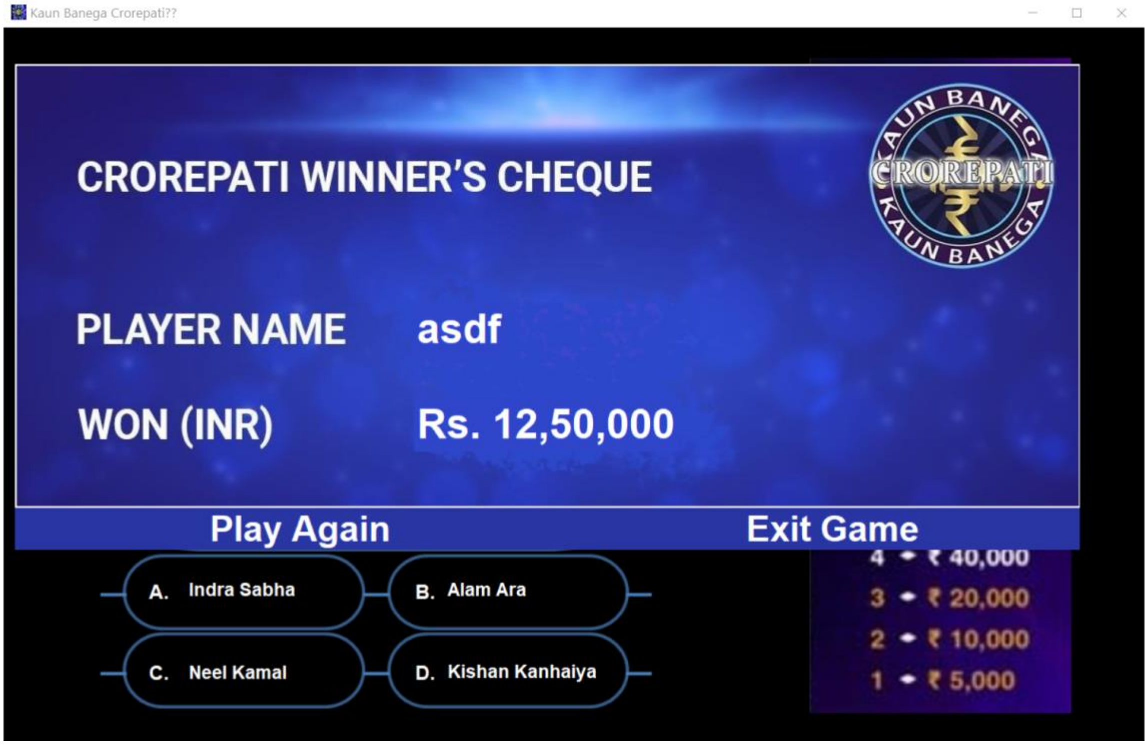
A. June 21

B. May 21

C. August 21

D. July 21

15	₹ 7 Crore
14	₹ 5 Crore
13	₹ 3 Crore
12	₹ 1 Crore
11	₹ 50 Lakh
10	₹ 25 Lakh
9	₹ 12,50,000
8	₹ 6,40,000
7	₹ 3,20,000
6	₹ 1,60,000
5	₹ 80,000
4	₹ 40,000
3	₹ 20,000
2	₹ 10,000
1	₹ 5,000



CROREPATI WINNER'S CHEQUE

PLAYER NAME asdf

WON (INR) Rs. 12,50,000

A. Indra Sabha

B. Alam Ara

C. Neel Kamal

D. Kishan Kanhaiya

4	₹ 40,000
3	₹ 20,000
2	₹ 10,000
1	₹ 5,000

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