KBC QUIZ GAME IN PYTHON

MINOR PROJECT REPORT

By

AVANI BADERIYA (RA2211003011780)
NEHA MAURYA (RA2211003011787)
MARUSHKA SINGHANIA (RA2211003011810)

Under the guidance of

MS.M HEMA

In partial fulfilment for the Course

of

21CSC203P – ADVANCED PROGRAMMING PRACTICE

in <**Computing Technologies**>

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC203P ADVANCED PROGRAMMING PRACTICE** entitled in **KBC Quiz Game** is the bonafide work of **AVANI BADERIYA (RA2211003011780), NEHA MAURYA (RA2211003011787), and MARUSHKA SINGHANIA (RA2211003011810)** who carried out the work under my supervision.

Department of Computing Technologies Kattankulathur 603 203 SRM IST

SIGNATURE

Ms. M. HEMA
Assistant Professor
Computing Technologies Department
SRM Institute of Science and Technology
Kattankulathur

SIGNATURE

Dr. M. PUSHPALATHA
Professor and Head
Computing Technologies Department
SRM Institute of Science and Technology
Kattankulathur

ABSTRACT

The project is about a quiz with carefully chosen questions for users. The quiz is short, focusing on a few questions for a clear experience. Users earn money for right answers, with the amount depending on the question's difficulty. This tiered setup adds challenge and excitement, encouraging users to try harder questions for more money. The limited question selection allows for careful curation, ensuring a varied and high-quality quiz for different interests. With different difficulty levels, the quiz stays interesting and gives users a sense of accomplishment as they progress. Overall, it's a simple yet rewarding intellectual quiz.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G , Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **Ms.M Hema , Assistant Professor , Computing Technologies** for her assistance, timely suggestion and guidance throughout the duration of this course project.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

CHAPTER NO	CONTENTS	PAGE NO
1	**INTRODUCTION**	**5**
	1.1 Motivation	**6**
	1.2 Objective	**6**
	1.3 Problem Statement	**6**
2	**DATABASE DESIGN**	**7**
3	**LITERATURE SURVEY**	**8**
4	**REQUIREMENT ANALYSIS**	**10**
5	**PROGRAM**	**12**
6	**RESULT/OUTPUT**	**21**
7	**CONCLUSION**	**22**
8	**REFERENCES**	**23**

1. INTRODUCTION

The project centers on a quiz featuring a curated set of questions for users to attempt. The quiz is designed with a limited number of questions, ensuring a focused and concise experience. Participants earn a specific amount of money for each correct answer, and the monetary reward varies based on the difficulty level of the question. This tiered structure adds an element of challenge and excitement, encouraging users to tackle more difficult questions for higher financial gains. The limited question pool facilitates meticulous curation, ensuring a diverse and high-quality set of challenges that cater to different interests and knowledge levels. With the incorporation of progressive difficulty levels, the quiz not only sustains user interest but also provides a sense of accomplishment as participants navigate through the various tiers, making the quiz a dynamic and rewarding intellectual pursuit.

1.1 Motivation

We want to make learning fun and rewarding by creating a quiz platform. Our motivation is to address the challenge of making education engaging and providing incentives for users. We aim to achieve this by offering a mix of interesting questions and monetary rewards for correct answers, making learning enjoyable and encouraging ongoing participation. Our goal is to create an educational tool that goes beyond traditional methods, making the journey of learning exciting and fulfilling for users.

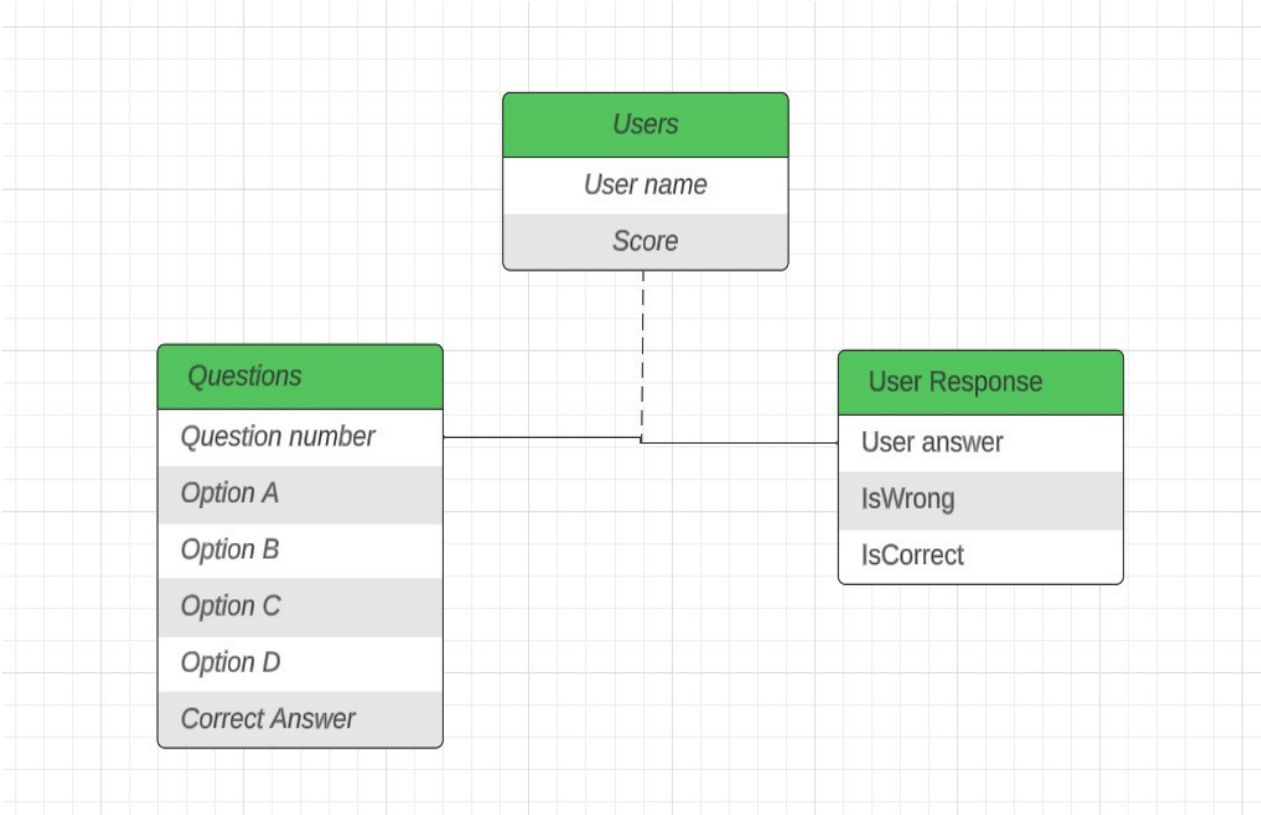
1.2 Objective

The project provides a fun platform for users to test their general knowledge. It employs increasing difficulty levels with each correct answer to help users assess their knowledge across various topics. The goal is to make learning enjoyable and encourage users to explore different subjects.

1.3 Problem Statement

In a world saturated with information, there is a pressing need for an engaging and concise educational platform. Traditional quizzes often lack the appeal to captivate a diverse audience. Our project aims to fill this gap by introducing an innovative quiz format that combines curated questions with monetary rewards for correct answers, providing users with a dynamic and rewarding intellectual experience.

2. DATABASE DESIGN



3. LITERATURE SURVEY

1) The use of serious games in education field is now attracting and needed more than ever with all the new challenges in our era. This paper focused on achieving a literature survey to identify the research studies showing significant achievements using serious games for educational goals. In this presented work we collected more than four hundred research paper from multiple scientific databases over the years to be processed using unsupervised topic modelling method with LDA (Latent Dirichlet Allocation) and Python. We first describe the main terms or concepts our research study address: serious games and education. Then we present the results right after we explain in detail the methodology we followed in our survey, and finally the conclusion.

2) Case-based reasoning(CBR) is a growing topic in machine learning. CBR is a system which contains 4Rs, retrieve, reuse, revise and retain. When a problem enters the system, all the cases which are similar to the problem are retrieved, then adaptation process is used and revised and then stored in the casebase. The observed problems of the system are lack of knowledge, adaptation is very difficult and different domain need different adaptation process for better result. In this paper, we get to know what is case-based reasoning and its adaptation methods. The literature survey tells that every domain may need different adaptation methods. We have used a few adaptation methods in python to a football match dataset and found which adaptation method gives the best accuracy. We have used myCBR for getting retrieved cases.

3) Game companies avoid sharing their game data with external researchers. Only a few research groups have been granted limited access to game data so far. The reluctance of these companies to make data publicly available limits the wide use and development of data mining techniques and artificial intelligence research specific to the game industry. In this paper, we developed and implemented an international competition on game data mining using commercial game log data from one of the major game companies in South Korea: NCSOFT. Our approach enabled researchers to develop and apply state-of-the-art data mining techniques to game log data by making the data open. For the competition, data were collected from Blade & Soul, an action role-playing game, from NCSOFT. The data comprised approximately 100 GB of game logs from 10 000 players. The main aim of the competition was to predict whether a player would churn and when the player would churn during two periods between which the business model was changed to a free-to-play model from a monthly subscription. The results of the competition revealed that highly ranked competitors used deep learning, tree boosting, and linear regression.

4) Within industry as well as academia, developing games that have wider impact on society has been of particular interest in the last decade. The increasing use of terms such as 'games with purpose', ' games for good' and 'serious games' has been mirrored by a flurry of activity in games research. Broader applications of games beyond entertainment are now well-understood and accepted, with universities and companies excelling in creating games to serve particular needs. However, it is not explicitly clear how undergraduates of game design and development courses can be directly involved in serious game creation. With most undergraduates inspired by commercial games development, and the games industry requiring that universities teach specific technical skills in their courses, balancing the research aspirations of academics with the educational requirements of an appropriate undergraduate course can be a difficult balancing act. In this paper, the authors present three case studies of games with purpose developed through collaboration between undergraduate students and academic staff. In all cases, the educational value of the projects for the students is considered in relation to the research value for the academics, who face increasing demands to develop research outcomes despite a necessity to provide a first-rate learning experience and nurture future game developers.

5) The vast majority of Germans are reluctant to invest parts of their savings in the capital market, even though this is generally considered best practice for financial and retirement planning. Market research shows that lacking knowledge and motivation are key reasons for this behavior. In this research project, we evaluate game design and game mechanics as a means to educating and motivating particularly the younger generations to follow widely accepted general financial planning advice. Our paper presents an overview of the research we conducted and discusses a prototype we built, playtested, and currently evaluate for field deployment.

6) The stability and performance of game servers are major issues in online games because the online game servers must be able to handle and verify a large number of connections. Test automation has been used in order to reduce testing time of online game servers. In this paper, we propose the VENUS II system, which supports blackbox testing and scenario-based testing as well as simple load testing. The massive virtual clients automatically generate packet loads to test the stability of game servers like the existing approaches. The main difference, however, is that we used the game grammar and game map to describe the game logics instead of using dummy game client code. So the test client code doesn't need to be re-written when a new game is to be tested. And the complex scenarios such as attack, party play and waypoint movement can be tested and moreover effectively because the scenarios can be flexibly constructed by combining protocols and actions already defined in game grammar. We have applied our tools on the ELMA online game application to verify the effectiveness of our method

4. REQUIREMENT ANALYSIS

Requirement analysis is a crucial phase in software development, including for a project like a KBC (Kaun Banega Crorepati) quiz game in Python. Here are some high-level requirements you might consider:

Functional Requirements:

1. User Registration and Authentication:
   - Users should be able to register with unique usernames.
   - Implement secure authentication mechanisms to protect user accounts.

2. Gameplay:
   - Questions should be presented to users in a quiz format.
   - Multiple-choice questions with one correct answer.
   - Questions should have varying difficulty levels.
   - Users should have a set time to answer each question.

3. Lifelines:
   - Users should have access to lifelines like "50:50," "Phone a Friend," and "Ask the Audience."

4. Scoring System:
   - Implement a scoring system based on the difficulty level of questions.
   - Users should accumulate points for correct answers.

5. Leaderboard:
   - Maintain a leaderboard to display high scores.
   - Implement a ranking system based on user scores.

6. Game Progress:
   - Save and resume game progress for users.
   - Allow users to restart or exit the game.

7. GUI (Graphical User Interface):
   - Design an intuitive and user-friendly interface.
   - Include buttons for lifelines, answer selection, and other interactions.

8. Sound and Visual Effects:
   - Include appropriate sound effects for correct and incorrect answers.
   - Visual effects for lifelines and other significant events.

9. Database Integration:
   - Store user profiles, questions, and game progress in a database.
   - Use a relational database to manage data efficiently.

Non-Functional Requirements:

1. Performance:
   - The game should be responsive and handle multiple users simultaneously.

2. Security:
   - Implement secure coding practices to prevent vulnerabilities.
   - Protect against common security threats.

3. Scalability:
   - Design the system to handle an increasing number of users.
   - Optimize database queries for efficiency.

4. Compatibility:
   - Ensure compatibility with different operating systems (Windows, macOS, Linux).

5. Usability:
   - Conduct usability testing to ensure the game is easy to understand and navigate.

6. Reliability:
   - The system should be stable and minimize downtime.

7. Maintainability:
   - Develop code that is easy to maintain and update.
   - Include comments and documentation for future reference.

8. Testing:
   - Conduct thorough testing, including unit testing and system testing.

9. Legal and Ethical Considerations:
   - Ensure compliance with relevant laws and ethical standards.
   - Obtain necessary permissions for any copyrighted content.

10. Localization:
    - Consider the possibility of localization for different languages.

11. Backup and Recovery:
    - Implement regular data backup procedures and a recovery mechanism.

12. Logging:
    - Implement logging to track errors and user activities for troubleshooting.

Remember that these requirements may need further refinement based on the specific goals and constraints of your project. Additionally, iterative development and feedback from potential users can help refine and improve the requirements as the project progresses.

5. PROGRAM

```
#import random and time module
import random
import time
#print * 80 times
for i in range(80):
    print("*",end="")
    time.sleep(0)
print()
print("\t\t\t       Welcome to Kaun Banega Crorepati  ")
for i in range(80):
    print("*",end="")
    time.sleep(0)
print()
a=input("\tEnter Your Name - ")
for i in range(80):
    print("*",end="")
    time.sleep(0)
print()
print("\n\t\tOK ",a," Let's Start The Game")
time.sleep(1)
questions=["Who is The Prime Minister of India","In Which Country
Area 51 is Located","Which one is the largest Continent in the
world","What is the Latest Version of Windows Since
```

2019","Which One of These Is not a Software Company","How Many MB Makes 1 GB","Facebook Was Firstly Developed By","Founder of Apple is","___ is one of The Founder of Google","BIGG BOSS season 13 Starts in __ & ends in __","Apple's Laptop is Also Known as","First Apple Computer is Known as","Joystick is used For","_____ is used to Encrypt Drives in Computer"]

answer=["Narendra Modi","United States","Asia","Windows 10","Honda","1024","Mark Zuckenberg","Steve Jobs","Larry Page","2019 - 2020","Macbook","Mactonish","Playing Games","Bitlocker"]
wronganswers=[["Amit Shah","Aditya Nath Yogi","Azhar Ansari"],["India","Africa","Iraq"],["South Africa","North America","Europe"],["Windows 7","Windows 8","Windows 11"],["Oracle","Microsoft","Google"],["10024","1004","2024"],["Bill Gates","Larry Page","Azhar Ansari"],["Azhar Ansari","Charles Babbage","Sundar Pichai"],["Larry Hensberg","Sunder Pichai","Bill Gates"],["2020 - 2021","Not Starts Now","2018 - 2019"],["ThinBook","Notebook","ChromeBook"],["Apple v.1","Apple Computer","Appbook"],["Giving output command","Shutting down Computer","Log off

```
Computer"],["KeyGuard","Windows Secure","No Software like

this"]]
attempquestion=[]
count=1
amount=0
while True:
    while True:
        selectquestion=random.choice(questions)
        if selectquestion in attempquestion:
            pass
        elif selectquestion not in attempquestion:
            attempquestion.append(selectquestion)
            questionindex=questions.index(selectquestion)
            correctanswer=answer[questionindex]
            break
    optionslist=[]
    inoptionlist=[]
    optioncount=1


    while optioncount<4:
        optionselection=random.choice(wronganswers[questionindex])
```

```
    if optionselection in inoptionlist:
        pass
    elif optionselection not in inoptionlist:

        optionslist.append(optionselection)
        inoptionlist.append(optionselection)
        optioncount+=1
optionslist.append(correctanswer)
alreadydisplay=[]
optiontodisplay=[]
a1=True
while a1:
    a=random.choice(optionslist)
    if a in alreadydisplay:
        pass
    else:
        alreadydisplay.append(a)
        optiontodisplay.append(a)
        a1=not True
a1=True
while a1:
    b=random.choice(optionslist)
    if b in alreadydisplay:
        pass
```

```
    else:
        alreadydisplay.append(b)
        optiontodisplay.append(b)
        a1=not True
a1=True

while a1:
    c=random.choice(optionslist)
    if c in alreadydisplay:
        pass
    else:
        alreadydisplay.append(c)
        optiontodisplay.append(c)
        a1=not True
a1=True
while a1:
    d=random.choice(optionslist)
    if d in alreadydisplay:
        pass
    else:
        alreadydisplay.append(d)
        optiontodisplay.append(d)
        a1=not True
```

```
    right_answer=""
    if correctanswer==a:

        right_answer="a"
    elif correctanswer==b:
        right_answer="b"
    elif correctanswer==c:
        right_answer="c"



    elif correctanswer==d:
        right_answer="d"
    print("------------------------------------------------------------------------
------------------")
    print("\t\t\tAmount Win - ",amount)
    print("------------------------------------------------------------------------
------------------")
    time.sleep(1)

  print("\n\t\tQuestion ",count," on your Screen")
    print("------------------------------------------------------------------------
------------------")
    time.sleep(1)
    print("  |  Question - ",selectquestion)
    print("------------------------------------------------------------------------
```

```
------------------")
    time.sleep(1)



    print("\t|  A. ",a)
    time.sleep(1)
    print("\t|  B. ",b)
    time.sleep(1)
    print("\t|  C. ",c)
    time.sleep(1)
    print("\t|  D. ",d)
    useranswer=input("\t\tEnter Correct Option\t   or \t press Q to


quit.\n\t\t\t...").lower()
    if useranswer==right_answer:
        if count==1:
            amount=1000
        elif count==2:
            amount=2000
        elif count==3:
            amount=5000
        elif count==4:
            amount=10000
        elif count==5:
            amount=40000
```

```
elif count==6:
    amount=80000

elif count==7:

    amount=160000
elif count==8:
    amount=320000
elif count==9:
    amount=640000
elif count==10:
    amount=1500000
    print("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n")


    print("****************************")
    print("\t\t\t \\\\\\\\\\ Congratulations! //////////")
    print("\t\t\t|||||||||| You Won The Game ||||||||||")
    print("****************************")
    print("\n\n\t\t You Won Rs. ",amount)
    print()
    break
print("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n")
print("****************************")
print("\t\t\t \\\\\\\\\\ Congratulations! //////////")
```

```
            print("\t\t\t||||||||| Right Answer |||||||||")
            print("*****************************")
            count+=1
        elif useranswer=="q":
                print("\n\n\t\t You Won Rs. ",amount)


                break
        else:
            print("*****************************")
            print("\t\t\tWrong Answer")
            print("*****************************")
            print("\n\n\t\t \tYou Won Rs. ",amount)
            print("*****************************")
    break
```

6. OUTPUT

```
**********************************************************************
                Welcome to Kaun Banega Crorepati
**********************************************************************
       Enter Your Name - m
**********************************************************************

              OK  m  Let's Start The Game

              Question  1  on your Screen
----------------------------------------------------------------------
  |  Question -  Which One of These Is not a Software Company
----------------------------------------------------------------------
       |  A.  Oracle
       |  B.  Google
       |  C.  Honda
       |  D.  Microsoft

              Enter Correct Option or press Q to quit : c




          \\\\\ Congratulations! /////////
          ||||||||| Right Answer |||||||||


               You Won Rs.  1000
**************************************************************

              Question  2  on your Screen
----------------------------------------------------------------------
----------------------------------------------------------------------
  |  Question -  What is the Latest Version of Windows Since 2019
----------------------------------------------------------------------
       |  A.  Windows 11
       |  B.  Windows 8
       |  C.  Windows 7
       |  D.  Windows 10

              Enter Correct Option or press Q to quit : a
                      Wrong Answer!
                      You Won Rs.  1000
**********************************************************************
```

7. CONCLUSION

In conclusion, the quiz project offers an engaging and rewarding experience for users. Its limited-question approach ensures a focused and varied challenge, promoting a concise yet diverse quiz environment. Participants earn money based on question difficulty, introducing an exciting element that encourages tackling more challenging queries. The curated question pool reflects a commitment to quality and caters to a range of interests and knowledge levels. The inclusion of progressive difficulty levels not only sustains user interest but also fosters a sense of achievement as users progress through the tiers. Overall, the quiz project provides a dynamic and intellectually satisfying platform, making learning and testing one's knowledge an enjoyable journey.

8. REFERENCES

- https://realpython.com/python-quiz-application/
- https://www.crio.do/projects/android-java-quiz-application/
- https://www.geeksforgeeks.org/python-quiz-application-project/