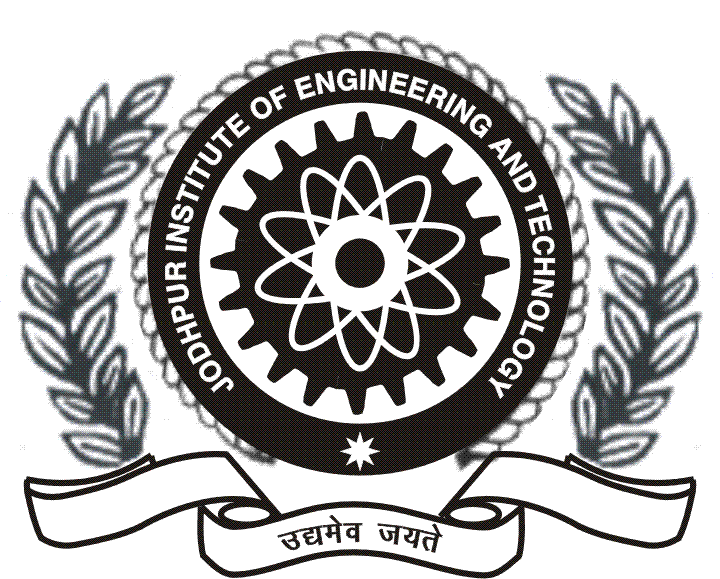
**A**

**GUI ( TKINTER) PROJECT REPORT**

**ON**

**QUIZ APPLICATION**

*In partial fulfillment of*

***B.Tech I yr* (Computer Science & Engg.)**

Submitted To: Submitted by:

**Surbhi Upadhyay Aman Kumar Sharma**

Assistant Professor Roll no – T21EJICS015

**Abhishek Kumar Singh** Roll no – T21EJICS006

# Acknowledgment

I would like to acknowledge the contributions of the following people without whose help and guidance this project would not have been completed.

I respectfully thank **Mrs. Surbhu Upadhyay Ma’am**, for providing me an opportunity to do this project work and giving me all support and guidance, which made me complete the project up to very extent.

I am also thankful to **Mrs. Mamta Garg Ma’am** HoD of Computer Science and Engineering Department, Jodhpur Institute of Engineering and Technology, for his/her constant encouragement, valuable suggestions and moral support and blessings.

Although it is not possible to name individually, I shall ever remain indebted to the faculty members of Jodhpur Institute of Engineering and Technology, for their persistent support and cooperation extended during his work.

This acknowledgement will remain incomplete if I fail to express our deep sense of obligation to my parents and God for their consistent blessings and

encouragement.

# Table of Contents

Acknowledgment

Table of Contents

1. Introduction 1

2. Technology used in Project 2

3. Details of Project

3.1 Functions/Modules Details

3.2 Flow Chart

3.3 Project Code

3.4 Project Screenshots

4.Applications

5. conclusion and Future Work

6. References

# 1. Introduction

.

The ‘MCQ Quiz Application’ project is developed to overcome the time consuming problem of manual system. Apart from that in current system, checking the answer sheets after taking test, waste the examiners time, so this application will check the correct answer and save the examiner time and carry the examination in an effective manner. The users which are use this system don’t need to high computing knowledge and also system will inform them while entering invalid data.

The aim of this project is to computerized the existing manual system and help the examiners to save their valuable time and important data. Apart from this, data which are exist in this system, will exist for long period of time and will be easy accessible. This project helps the examiners to manage their services in a good way and provide a better service to their users.

The objective of this project is to manage the details of students, examinations, marks, courses and papers in a good manner. The performance of the application will be fully control by administrator and administrator can guaranty any one to access. The project will reduce the manual process in managing examinations and all issues regarding that.

# 2. Technology used in Project

**Python - GUI Programming (Tkinter)**

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

1) Import the *Tkinter* module

2) Create the GUI application main window.

3) Add one or more of the above-mentioned widgets to the GUI application.

4) Enter the main event loop to take action against each event triggered by the user.

## Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

The widgets used in this project are presented below-

|  |  |
| --- | --- |
| **Sr.No.** | **Operator & Description** |
| 1 | [Button](https://www.tutorialspoint.com/python/tk_button.htm)  The Button widget is used to display buttons in your application. |
| 2 | [Radiobutton](https://www.tutorialspoint.com/python/tk_checkbutton.htm)  The Radiobuttion widget is used to display a number of options . The user can select one options at a time. |
| 3 | [Entry](https://www.tutorialspoint.com/python/tk_entry.htm)  The Entry widget is used to display a single-line text field for accepting values from a user. |
| 4 | [Label](https://www.tutorialspoint.com/python/tk_label.htm)  The Label widget is used to provide a single-line caption for other widgets. It can also contain images. |
| 5 | [Message](https://www.tutorialspoint.com/python/tk_message.htm)  The Message widget is used to display multiline text fields for accepting values from a user. |
| 6 | [Text](https://www.tutorialspoint.com/python/tk_text.htm)  The Text widget is used to display text in multiple lines. |
| 7 | [Toplevel](https://www.tutorialspoint.com/python/tk_toplevel.htm)  The Toplevel widget is used to provide a separate window container. |
| 8 | [LabelFrame](https://www.tutorialspoint.com/python/tk_labelframe.htm)  A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts. |
| 9 | [tkMessageBox](https://www.tutorialspoint.com/python/tk_messagebox.htm)  This module is used to display message boxes in your applications. |

# 3. Details of Project

In this section give the details of the work done during the training period. Following below details are required.

3.1 Flow Chart

## 

## 3.2 Functions/Modules Details

Tkinter:- It is the standard GUI library for python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Import the Tkinter module.

JSON:- Python has a built-in package called json, which can be used to work with JSON data.

Moreover the project includes the usage of user defined functions such as display\_result, check\_ans, next\_ques, etc.

**3.3 Project Code**

from lib2to3.pgen2.token import LEFTSHIFT

from tkinter import\*

from PIL import Image, ImageTk

top=Tk()

top.title("Quiz World")

top.geometry("1050x800")

top.configure(background="light yellow")

lbltitle=Label(top,text="Welcome to Quiz World",bg="light yellow",fg="Red",bd=20,relief=RIDGE,font=("times new roman",50,"bold"),padx=2,pady=6)

lbltitle.pack(side=TOP,fill=X)

DataFrameLeft=LabelFrame(top,text="Student's Details",bg="light yellow",fg="green",bd=20,relief=RIDGE,font=("times new roman",30,"bold"))

DataFrameLeft.place(x=220,y=140,width=650,height=300)

member2=Label(DataFrameLeft,text="Student's Name ",bg="light yellow",font=("times new roman",20,"bold"),padx=2,pady=6)

member2.grid(row=1,column=0,sticky=W)

member2=Entry(DataFrameLeft,font=("times new roman",20,""))

member2.grid(row=1,column=3 )

member3=Label(DataFrameLeft,text="Roll Number  ",bg="light yellow",font=("times new roman",20,"bold"),padx=2,pady=6)

member3.grid(row=2,column=0,sticky=W)

member3=Entry(DataFrameLeft,font=("times new roman",20,""))

member3.grid(row=2,column=3)

member4=Label(DataFrameLeft,text="Mobile Number ",bg="light yellow",font=("times new roman",20,"bold"),padx=2,pady=6)

member4.grid(row=3,column=0,sticky=W)

member4=Entry(DataFrameLeft,font=("times new roman",20,""))

member4.grid(row=3,column=3)

def Qu():

    from tkinter import messagebox as mb

    import json

    class Quiz:

        def \_\_init\_\_(self):

            self.q\_no=0

            self.display\_title()

            self.display\_question()

            self.opt\_selected=IntVar()

            self.opts=self.radio\_buttons()

            self.display\_options()

            self.buttons()

            self.data\_size=len(question)

            self.correct=0

        def display\_result(self):

            wrong\_count = self.data\_size - self.correct

            correct = f"Correct: {self.correct}"

            wrong = f"Wrong: {wrong\_count}"

            score = int(self.correct / self.data\_size \* 100)

            result = f"Score: {score}%"

            mb.showinfo("Result", f"{result}\n{correct}\n{wrong}")

        def check\_ans(self, q\_no):

            if self.opt\_selected.get() == answer[q\_no]:

                return True

        def next\_btn(self):

            if self.check\_ans(self.q\_no):

                self.correct += 1

            self.q\_no += 1

            if self.q\_no==self.data\_size:

                self.display\_result()

                gui.destroy()

            else:

                self.display\_question()

                self.display\_options()

        def buttons(self):

            next\_button = Button(gui, text="Next",command=self.next\_btn,

            width=10,bg="blue",fg="white",font=("ariel",16,"bold"))

            next\_button.place(x=350,y=380)

            quit\_button = Button(gui, text="Quit", command=gui.destroy,

            width=5,bg="black", fg="white",font=("ariel",16," bold"))

            quit\_button.place(x=700,y=50)

        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

        def display\_question(self):

            q\_no = Label(gui, text=question[self.q\_no], width=60,

            font=( 'ariel' ,16, 'bold' ), anchor= 'w' )

            q\_no.place(x=70, y=100)

        def display\_title(self):

            title = Label(gui, text="Welcome "+s,

            width=50, bg="yellow",fg="red", font=("ariel", 20, "bold"))

            title.place(x=0, y=2)

        def radio\_buttons(self):

            q\_list = []

            y\_pos = 150

            while len(q\_list) < 4:

                radio\_btn = Radiobutton(gui,text=" ",variable=self.opt\_selected,

                value = len(q\_list)+1,font = ("ariel",14))

                q\_list.append(radio\_btn)

                radio\_btn.place(x = 100, y = y\_pos)

                y\_pos += 40

            return q\_list

# Create a GUI Window

    gui=Toplevel(top)

    gui.geometry("800x450")

    gui.title("Quiz?")

    with open("data.json") as f:

        data=json.load(f)

    s=member2.get()

# set the question, options, and answer

    question = (data['question'])

    options = (data['options'])

    answer = (data[ 'answer'])

    quiz=Quiz()

    top.gestroy()

    gui.mainloop()

member5=Button(DataFrameLeft,text="Submit ",bg="light yellow",font=("times new roman",20,"bold"),padx=2,pady=6,command=Qu)

member5.grid(row=7,column=2)

DataFrameLeft=LabelFrame(top,text="Guided By :",bg="light yellow",fg="green",bd=20,relief=RIDGE,font=("times new roman",30,"bold"))

DataFrameLeft.place(x=10,y=460,width=500,height=300)

guid1=Label(DataFrameLeft,text="",bg="light yellow",font=("times new roman",20,"bold"),padx=2,pady=6)

guid1.grid(row=3,column=0,sticky=W)

name3=Label(DataFrameLeft,text="Surbhi Upadhya \n Assistant Professor ",bg="light yellow",font=("times new roman",25,"bold"),padx=2,pady=6)

name3.grid(row=1,column=1,sticky=W)

DataFrameLeft=LabelFrame(top,text="Guided To :",bg="light yellow",fg="green",bd=20,relief=RIDGE,font=("times new roman",30,"bold"))

DataFrameLeft.place(x=530,y=460,width=500,height=300)

name1=Label(DataFrameLeft,text="Abhishek Kumar Singh \nRoll no- T21EJICS006 \nBranch - CSE",bg="light yellow",font=("times new roman",20,"bold"),padx=2,pady=6)

name1.grid(row=12,column=4,sticky=W)

name2=Label(DataFrameLeft,text="Aman Sharma \nRoll no- T21EJICS015 \nBranch - CSE ",bg="light yellow",font=("times new roman",20,"bold"),padx=2,pady=6)

name2.grid(row=20,column=4,sticky=W)

load= Image.open("JGI.png")

render = ImageTk.PhotoImage(load)

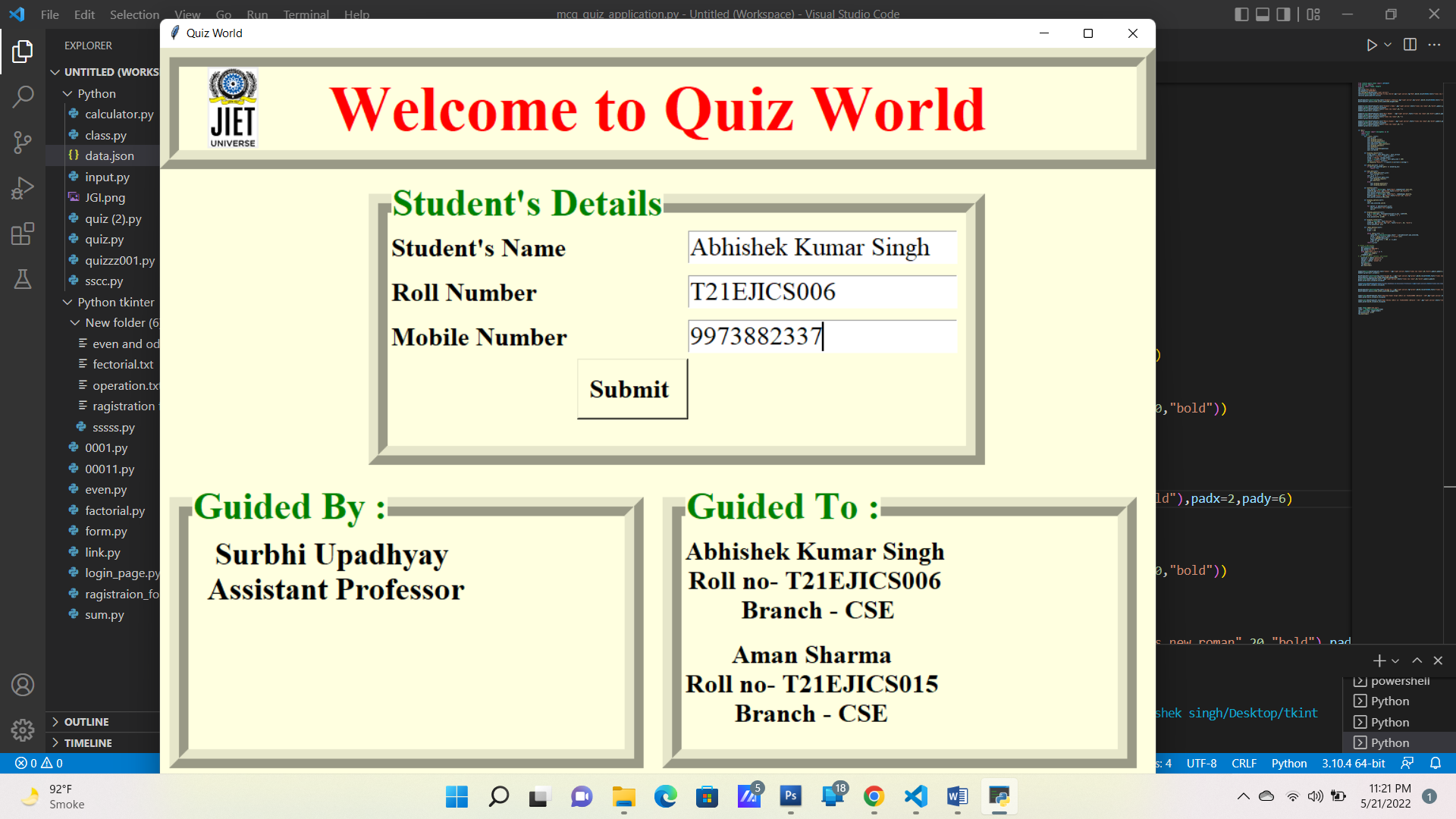
img = Label(top, image=render)

img.place(x=50, y=20)

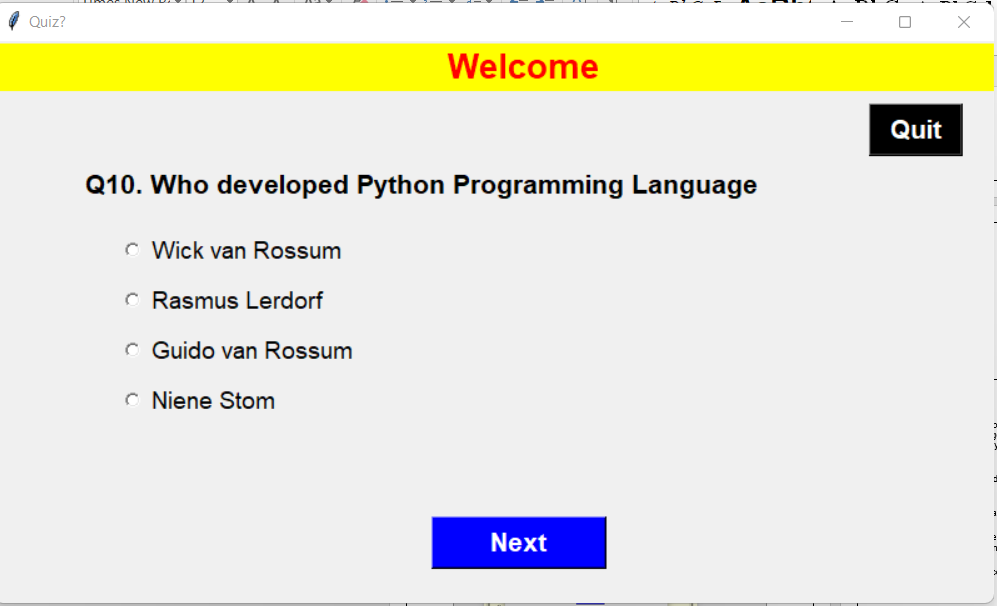
top.mainloop()

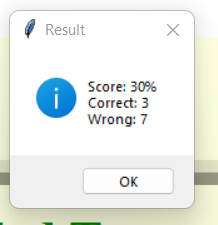
**# END OF THE PROGRAM**

## 3.4 Project Screenshots









# 4. Applications

Quizzes encourage pupil’s self awareness of progress and self assessment. By taking quizzes, pupil’s get instant feedback on their responses. And this can help them identify areas they need to develop themselves and highlight progress for them to be proud of.

This project can be used by educational institutions such as schools or colleges to evaluate the students.

Moreover companies can also use this for testing their future hiring needs.

It helps to manage the details of students, examinations, marks, courses and papers in a good manner.

The project will reduce the manual process in managing examinations and all issues regarding that.

# Future Work and Conclusion

The project can be improved b adding the following items:-

### 1.Randomizing questions

It is significant more easy to randomize your question with just one click than to do it all manually. Randomizing questions and even answers of those questions is not a lot a of work to do with online quizzes. Besides the advantage of time saving, it also helps preventing students from [cheating](https://www.onlinequizcreator.com/knowledge-center/oline-exam-center/how-to-prevent-cheating-on-an-online-exams-and-tests/item10616)

### 2.Set timer

The creator is able to [set a timer](https://www.onlinequizcreator.com/features/online-test-maker-with-timer/item10236) for the whole quiz or to set a timer per question. This is possible to do with written quizzes, but is very time consuming for the instructor. Plus it’s almost impossible to do with a huge amount of participants.

### 3.Better overview

### It’s possible to show one question at the time with online quizzes. People are not able to skip a question, because you will get a reminder that you can’t leave the answer blank.

Conclusion

To conclude this is a simple mcq quiz application which is developed for conduction of subjective examination. The marks of user will be calculated according to questions they attempt and will be displayed by the system to

the user as well as the administrator.

# References

**Links:-**

[www.wikipedia.org](http://www.wikipedia.org)

<https://www.geeksforgeeks.org/what-are-widgets-in-tkinter/>

**Books-**

Python Programming by Shatish Jain

**Teacher :-**

A special thanks to **Surbhi Ma’am and Mamata Ma’am** for their guidance

**Thank You**