DECISION REVIEW SYSTEM

PROJECT REPORT
OF MINI PROJECT

BACHELOR OF COMPUTER APPLICATION (BCA)

SUBMITTED BY

DINESH SINGH

BCA 5th Semester

Batch Year - 2020-2023

Enrollment No. - U2046035

PROJECT GUIDE: - Mr. ANAND DURGA SINGH

Centre of Computer Applications & Training

Institute of Professional Studies

University of Allahabad, Prayagraj

Uttar Pradesh

<u>ACKNOWLEDGEMENT</u>

This project report is based on **'DECISION REVIEW SYSTEM'**. I have taken all my efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Mr. ANAND DURGA SINGH** sir for his guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing the project. I would also like to express my gratitude towards my parents & members of **Institute of Professional Studies (IPS), University of Allahabad** for their kind cooperation and encouragement which helped me in completion of this project.

My thanks and appreciation also goes to my Friends in developing the project and people who have willingly helped me out with their abilities.

DINESH SINGH

BCA, 5th Semester

Enrollment No. - U204603

<u>CERTIFICATE</u>

It is here by certified that the project work entitled DECISION REVIEW SYSTEM using PYTHON is a bonafide work carried out by DINESH SINGH Enrollment NO: U2046035, Bachelor of Computer Education in Centre of Computer Education, Institute Of Professional Studies, University of Allahabad during the year 2020-2023.

It is certified that all the correction/suggestion indicated for Mini Project have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work allotted for the said degree.

Name of supervisor: <u>Mr. ANAND DURGA SINGH</u>

<u>DECLARATION</u>

I, **DINESH SINGH,** solemnly declare that the project report **DECISION REVIEW SYSTEM** is based on my own work carried out during the course of our study under the supervision of **Mr. ANAND DURGA SINGH.**

I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

1. The work contained in the report is original and has been done by me under the general supervision of my supervisor.

2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or Abroad.

3. We have followed the guidelines provided by the university in writing the report.

4. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and given their details in the references.

DINESH SINGH

BCA, 5th Semester

Enrollment No. – U2046035

TABLE OF CONTENT

INTRODUCTION

1.1 Objective

The main objective of the project is to built a Decision Review System (DRS) which helps on-field umpires to take decision whether the player is out or not out using Python.

1.2 Project Description

- Many times in some circumstances on-field Umpire are unable to decide whether the player is out or not out because the situation is too close, here Decision Review System(DRS) helps.

- Without Decision Review System (DRS) Umpire may take wrong decision which leads to crucial effect on the result of the match.

- Decision Review System(DRS) often called Third Umpire Review System is a technology which helps to take decision by revisiting the situation we want for.

- In this Project we use Python to develop the Decision Review System(DRS) which helps on-field umpires to take above crucial decision.

MOTIVATION

There have been quite a few instances when people felt the umpire giving out wrong judgment and the inclusion of DRS could be a means to solve this issue. DRS is a technology based system solely intended for the purpose of reviewing controversial decisions made by the on-field umpires, mostly on whether or not a batsman has been dismissed. There are some that are completely in favor of the system while the others call it a mere gimmick.

Fair decisions: Unfair and controversial decisions arising out of the inobservant behavior of some umpires at some point of time can cost a match or an entire series to a team that plays with all its might. When you are safe but sent out, it does discourage players and cause anger and anxiety in viewers. Including DRS will make cricket fairer. What more could we ask for?

Lessen unnecessary aggression: A bowler or a batsman who is unsatisfied with the decision of the umpire is usually seen to be aggressive and losing all his calm in the game. There rises sheer enmity and hatred between teams. This could instantly be done away with if DRS is given a prominent place and either of the bowler and batsman can ask for a review when he isn't satisfied.

Making use of technology: When you have a better way of reviewing decisions and it can be used to lessen or perhaps remove the chances of erroneous decisions, there is no reason why we shouldn't use it to the advantage of players. If it can be determined that the technology is capable of accurate judgment, we definitely should be using it.

An extra help: Going upstairs for each time there could be a no ball is sort of impractical and hence an extra added help from the technology that could clarify after each ball can be beneficiary for the on-field umpire. Each time there is a no ball when there is an appeal for out, the umpire is left alone to decide which gets difficult at times.

REQUIREMENT & SOFTWARE ANALYSIS

The process of deciding on the requirement of a software system, which determines the responsibilities of a system, is called requirement analysis. Requirement analysis is a software engineering task that bridges the gap between system level requirements engineering and software design.

Requirement engineering activities result in the specification of software's operational characteristics, indicate the software's interface with other system elements and establish constraints that the software must meet. The following section presents the detailed requirement analysis of our project

3.1   SOFTWARE REQUIREMENT

- **Operating System :**   Window/Linux/Mac
- **Language Used :**   Python
- **Additional Software :**   Photoshop
- **IDE :**   VsCode/Pycharm/IDLE

3.2   HARDWARE REQUIREMENT

- **CPU :**   Dual Core or Higher
- **RAM :**   2 GB or Higher

<u>SOFTWARE ANALYSIS</u>

Software analysis and design includes all activities, which help the transformation of requirement specification into implementation. Requirement specifications specify all functional and non-functional expectations from the software. These requirement specifications come in the shape of human readable and understandable documents, to which a computer has nothing to do.

Software analysis and design is the intermediate stage, which helps human readable requirements to be transformed into actual code.

In order to achieve the objectives of coming up with the DECISION REVIEW SYSTEM, the preliminary investigation about the workability of the software is necessary and need to be carried out first. This will equip me with the relevant materials and knowledge on how to carry out the implementation.

<u>TECHNOLOGIES USED</u>

PYTHON is a language used to built proposed project on Decision Review System. In this Project we have used GUI feature of Python called tkinter and various other modules as follows:-

- **Tkinter :** "Tkinter is a de-facto standard GUI package for Pythons (Graphical User Interface). It is on top of Tcl/Tk, a thin object-oriented layer. TKinter is not the only Python toolkit for GuiProgramming. However, it is the most commonly used one. Cameron Laird calls the annual decision to keep Tkinter "one of the minor traditions of the Python world." If you run python-m tkinter from the command prompt, it is important to open a window showing a simple Tk interface, letting you know that tkinter is correctly installed on your device, and also showing which version of Tcl/Tk is running''.

- **OPEN CV :** (Open-Source Computer Vision Library) is a library of programming functions aimed specifically at real-time computer vision. Originally designed by Intel, it was later sponsored by Willow Garage and then Itseez (later acquired by Intel). The framework is cross-platform and free-to-use under the open-source Apache 2 License. For real time operations beginning in 2011, OpenCV features GPU acceleration.

- **Pillow :** The Standard Python Library (abbreviated as PIL) (known in newer versions as Pillow) is an additional free and open-source Python programming language library that supports the opening, manipulation, and saving of several different image file formats.

- **Imultis :** With Open CV and both Python 2.7 and Python 3, it is much simpler to detect contours, a variety of high - quality custom such as transcription, rotation, reformatting, image segmentation, view of Matplotlib images, sorting of basic image processing functions.

SYSTEM DESIGN

1. WELCOME SCREEN

First welcome screen will be shown along with title of the project and creator name. We have lot of buttons just below the screen for browsing and controlling the flow of the clip.

2. BROWSING CLIP

Now we have "Browse" Button on the screen which is used to browse required clip from the file system.

3. VISITING CLIP

Then we have 4 Buttons (Forward (slow/Fast) and Backward (slow/Fast)) for controlling flow of video clip.

4. DECISION MAKING

After analyzing clip we have to take decision using 2 additional buttons (out/not out).

5. SPONSOR SCREEN

Now sponsor screen will be shown in this case we have "Third Umpire Review System".

6. DISPLAYING DECISION

At the End out/not out image will be shown as per the decision.

PROJECT ARCHITECTURE

Here we have graphically showing how this application will work in a basic way, which is shown below: -

```
┌─────────────────────┐
│   Welcome Screen    │
└─────────────────────┘
          │
          ▼
┌─────────────────────┐
│    Browsing Clip    │
└─────────────────────┘
          │
          ▼
┌─────────────────────┐
│    Visiting Clip    │
└─────────────────────┘
          │
          ▼
┌─────────────────────┐
│   Decision Making   │
└─────────────────────┘
          │
          ▼
┌─────────────────────┐
│   Sponsor Screen    │
└─────────────────────┘
          │
          ▼
┌─────────────────────┐
│ Decision(out / not out) │
└─────────────────────┘
```
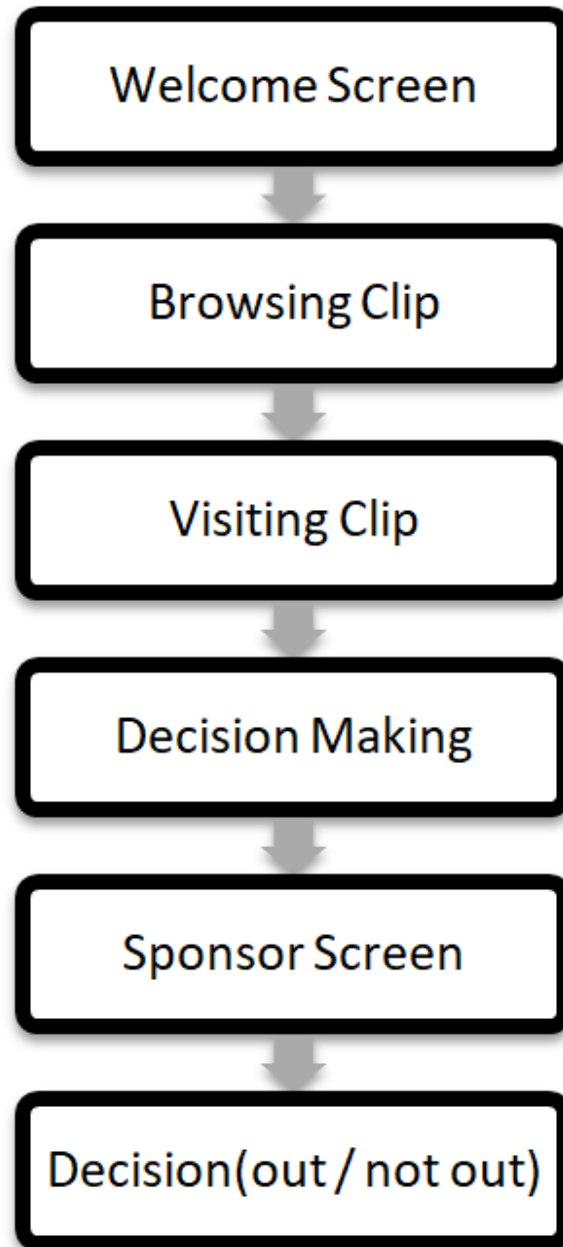
ACTIVITY DIAGRAM

Activity diagram provides a graphical representation of various activities that are carried out in an application. A clear overview of the DECISION REVIEW SYSTEM will be provided from the beginning till the end. The workflow of the activities and their dependencies with each other during the execution of the tasks and processes is shown in the activity diagram. The major functionalities of the site have been depicted using the diagram as shown below.
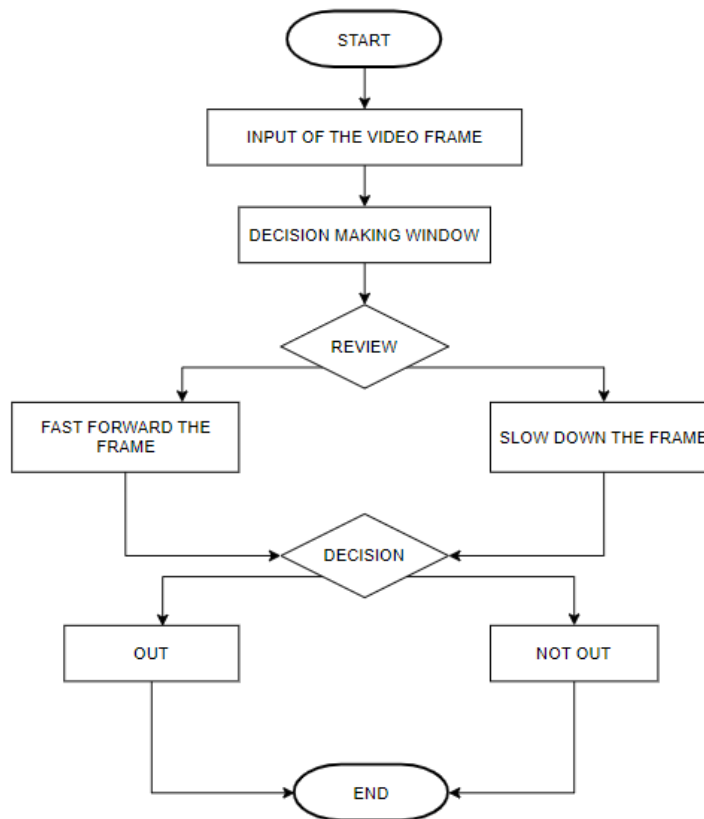
CODING

Importing Libraries

```
import tkinter
from tkinter import filedialog
import cv2
import PIL.Image, PIL.ImageTk
from functools import partial
import threading
import imutils
import time
import random
from tkinter.filedialog import askopenfile
```

Initializing Flag

```
flag = True
```

Browse Method

```
#Browsing Video Clip from File System
def browse_clip():
    """Method used to Browse Video Clip"""
    clip = filedialog.askopenfilename(type=("*.mp4"))
    global stream
    stream = cv2.VideoCapture(clip)
```

Play Method

```
#Creating play function
def play(speed):
    """Method used to control the flow of Video Clip"""
    global flag
    #playing the video clip in reverse/forward mode
    frame1 = stream.get(cv2.CAP_PROP_POS_FRAMES)
    stream.set(cv2.CAP_PROP_POS_FRAMES,frame1 + speed)

    grabbed,frame = stream.read()
```

```
    #Exit the window when video ends
    if not grabbed:
        exit()
    frame = imutils.resize(frame,width=SET_WIDTH,height=SET_HEIGHT)
    frame = PIL.ImageTk.PhotoImage(image = PIL.Image.fromarray(frame))
    canvas.image = frame
    canvas.create_image(0,0,image = frame,anchor = tkinter.NW)
    if flag:
        #Creating Blinking "Decision Pending" Text
        canvas.create_text(140,29, fill="Red",font = "Times 26 italic bold", text="Decision
Pending")
    flag = not flag
```

Pending Method

```
#Creating pending function
def pending(decision):
    """Method used to show decision pending image, sponsor image and display out/not out
image"""
    #Display decision pending image
    frame = cv2.cvtColor(cv2.imread("pending.png"),cv2.COLOR_BGR2RGB)
    frame = imutils.resize(frame,width=SET_WIDTH,height=SET_HEIGHT)
    frame = PIL.ImageTk.PhotoImage(image = PIL.Image.fromarray(frame))
    canvas.image = frame
    canvas.create_image(0,0,image=frame,anchor=tkinter.NW)

    #Wait for 1 second
    time.sleep(1)

    #Display sponsor image
    frame = cv2.cvtColor(cv2.imread("sponsor.png"), cv2.COLOR_BGR2RGB)
    frame = imutils.resize(frame, width=SET_WIDTH, height=SET_HEIGHT)
    frame = PIL.ImageTk.PhotoImage(image=PIL.Image.fromarray(frame))
    canvas.image = frame
    canvas.create_image(0, 0, image=frame, anchor=tkinter.NW)

    #Display 1.5 second
    time.sleep(1.5)

    #Display out/decision
    if decision == "out":
        decisionImg = "out.png"
    else:
        decisionImg = "not_out.png"
    frame = cv2.cvtColor(cv2.imread(decisionImg), cv2.COLOR_BGR2RGB)
```

```
    frame = imutils.resize(frame, width=SET_WIDTH, height=SET_HEIGHT)
    frame = PIL.ImageTk.PhotoImage(image=PIL.Image.fromarray(frame))
    canvas.image = frame
    canvas.create_image(0, 0, image=frame, anchor=tkinter.NW)
```

Out Method

```
#Creating out function
def out():
    """Method used for out decision"""
    thread = threading.Thread(target=pending,args=("out",))
    thread.daemon = 1
    thread.start()
```

Not Out Method

```
# Creating not out function
def not_out():
    """Method used for not out decision"""
    thread = threading.Thread(target=pending,args=("not_out",))
    thread.daemon = 1
    thread.start()
```

Initializing GUI Window

```
# Intializing window screen dimentions
SET_WIDTH = 650
SET_HEIGHT = 368

# Creating Window screen
window = tkinter.Tk()

#Changing Window background colour
window.configure(bg="Pink")

# Setting max and min size of the window
window.maxsize(SET_WIDTH,SET_HEIGHT+200)
window.minsize(SET_WIDTH,SET_HEIGHT+200)

# Assigning title to the window
window.title("Decision Review System(Mini Project BCA5)")

# Reading Welcome image
cv_img = cv2.cvtColor(cv2.imread("Welcome.png"),cv2.COLOR_BGR2RGB)
```

```
# Creating Canvas for Welcome image
canvas = tkinter.Canvas(window,width = SET_WIDTH, height = SET_HEIGHT)
photo = PIL.ImageTk.PhotoImage(image=PIL.Image.fromarray(cv_img))

# Packing Welcome image to window screen
image_on_canvas = canvas.create_image(0,0,ancho=tkinter.NW,image=photo)
canvas.pack()
```

Creating Buttons

```
#Buttons to Control Playback

#Creating Browse Button
btn =
tkinter.Button(window,bg="black",fg="white",text="Browse",width=50,command=browse_clip)
btn.pack()

#Creating Previous(fast) Button
btn =
tkinter.Button(window,bg="black",fg="white",text="<<Previous(fast)",width=50,command=part
ial(play,-25))
btn.pack()

#Creating Previous(slow) Button
btn =
tkinter.Button(window,bg="black",fg="white",text="<<Previous(slow)",width=50,command=pa
rtial(play,-2))
btn.pack()

#Creating Forward(fast) Button
btn =
tkinter.Button(window,bg="black",fg="white",text="Foword(fast)>>",width=50,command=parti
al(play,25))
btn.pack()

#Creating Forward(slow) Button
btn =
tkinter.Button(window,bg="black",fg="white",text="Foword(slow)>>",width=50,command=part
ial(play,2))
btn.pack()

#Creating Out Button
btn = tkinter.Button(window,bg="black",fg="white",text="Out!",width=50,command = out)
btn.pack()
```

```
#Creating Not Out Button
btn = tkinter.Button(window,bg="black",fg="white",text="Not Out!",width=50,command =
not_out)
btn.pack()
```

Displaying Window

```
#displaying window
window.mainloop();
```

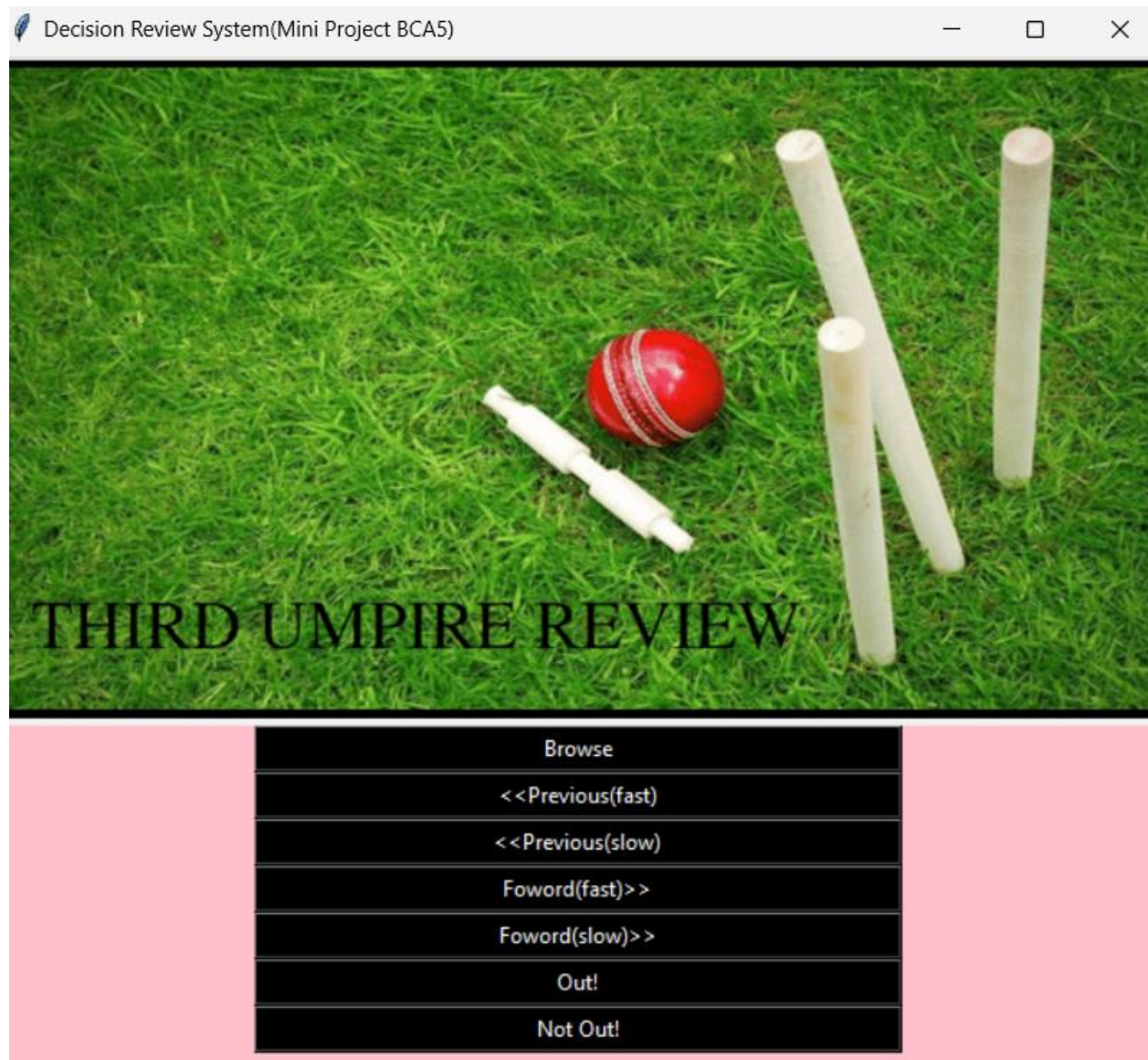# **SNAPSHOTS**

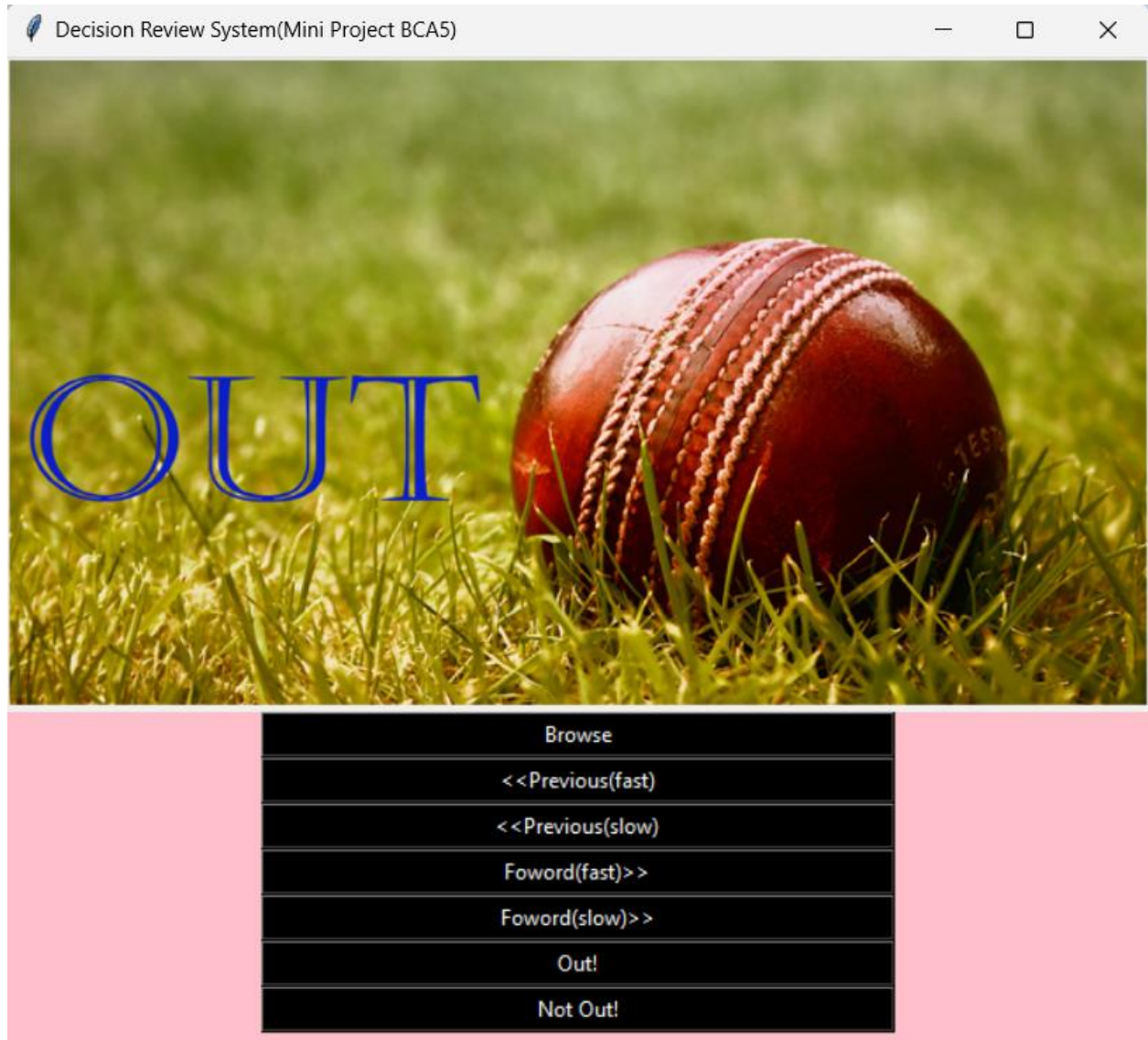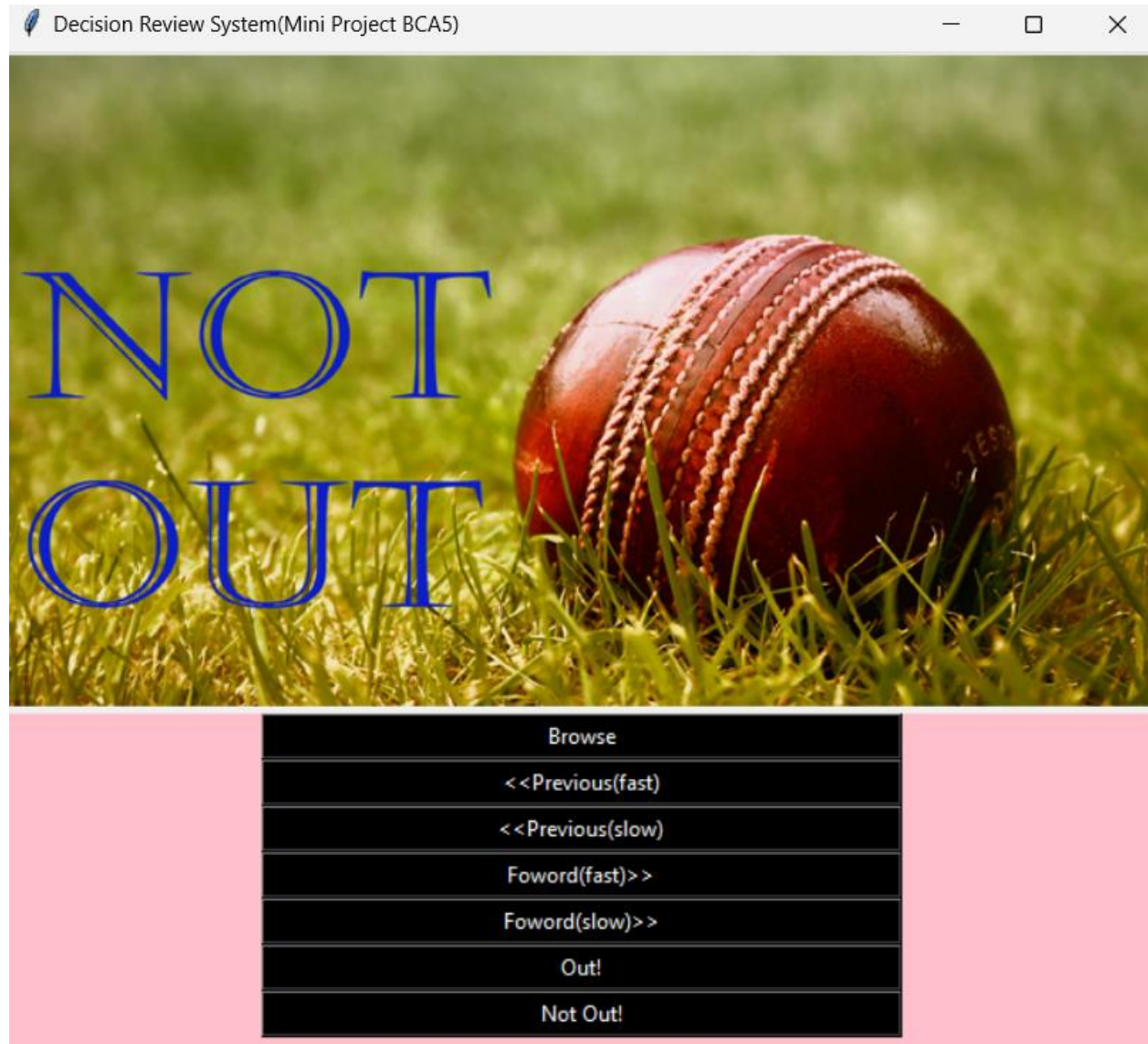**Welcome Screen**
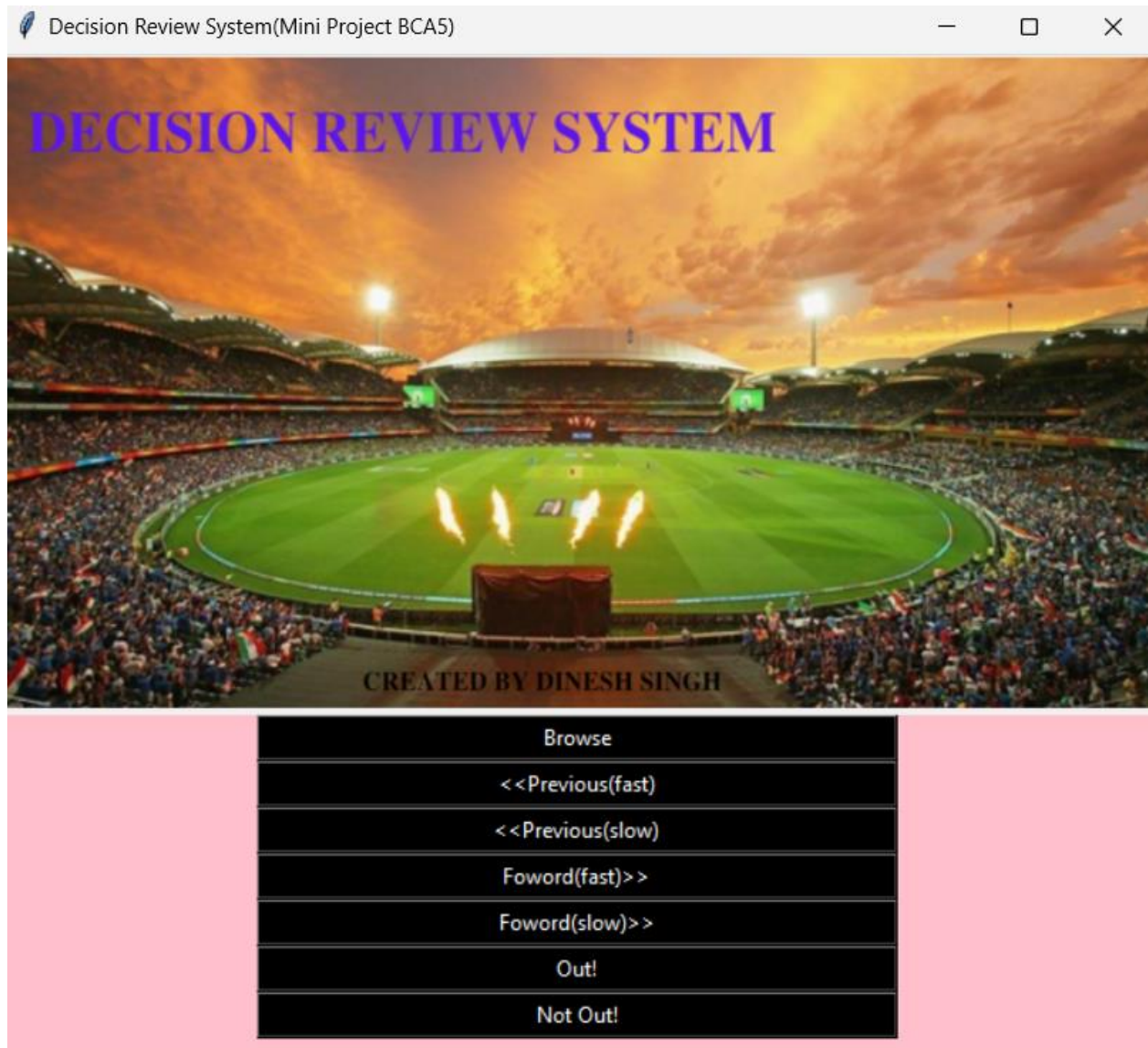
Clip Visit
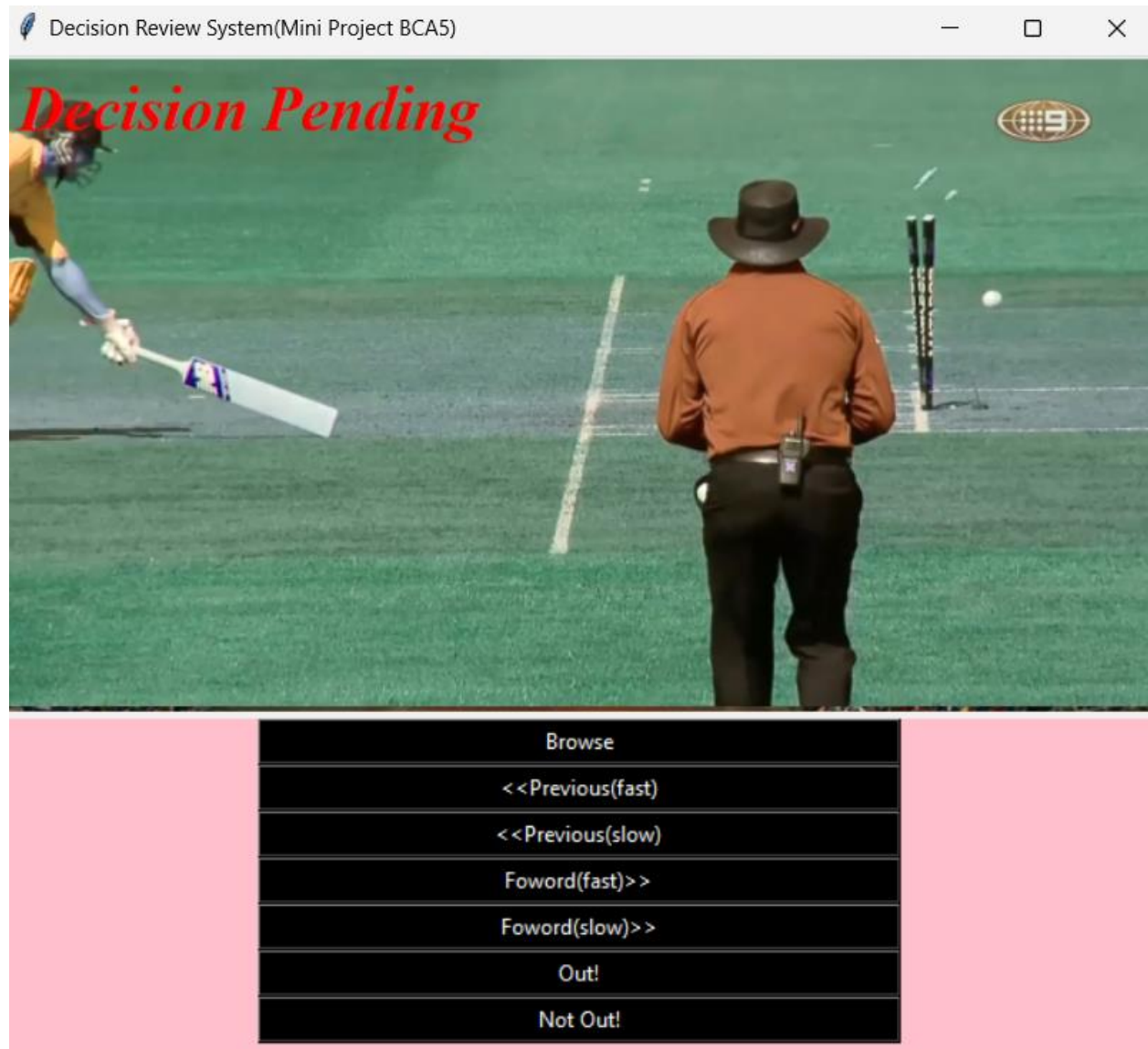
Decision Pending

Sponsor

Decision(Out)

Decision(Not Out)

<u>APPLICATIONS</u>

Diversity:

The diversified nature of the project intern helps in using the software in every other sport rather than bound to a single sport (cricket).

Small Scale Use:

The technology can be used at small as a district and inter school competitions.

Large Scale Use:

The technology can be used at high level tournaments like inter-state competitions.

Commercial Use:

The software can be used as a source of income, as in between the decisions, a small time-break has also been added to show commercial advertisements, which intake can be a good source for gaining profit even at a small level of inter-school tournaments.

<u>Future Scope</u>

There is always space for improvement. In this project specifically, live feed of the instance can be fed through the camera as an input stream for the software and thus decisions can be carried out in an instance, for the live audience. Ultimately saving the precious time. This project is only a toddler and there will always be room of improvement, making it more efficient to ease the human efforts and make one's work effortless.

<u>CONCLUSION</u>

Counting semantic structure, cinematography, framing, iconography, focus management and feedback, there's quite a lot it takes when creating a very efficient third umpire review system so everything we've learned here will help us create a simple and very efficient DRS system applicable in a wide range of outdoor sports.

There is always that instinct of patriotism in sports with fair decisions at the lesser level of sports tournaments to be played and this DRS can provide them a very cost effective and quality decision review system that will boost up the game spirit of each and every player.

Without any fancy equipment and with the help of just a computer system and a camera this system can run efficiently, thus making it compatible for all scale tournaments. This project is only an infant and there is always room of improvement and modifications making it more ready to go out and rock the gully cricket world. DRS technologies have been associated generally with very costly applications.

Today the core technologies have evolved and the cost of equipments are going up dramatically due to the integration and the increasing processing power. Certain decision review systems are not so cost effective and highly unreliable as they consume too much data and are time consuming.

<u>REFERENCES</u>

- www.github.com

- https://www.tutorialspoint.com/python

- https://docs.python.org/3/

MILESTONE

S.No.	ProjectActivity	Estimated Start Date	Estimated End Date
1.	Synopsis Submission	16/09/2022	16/09/2022
2.	Pre-Presentation	29/11/2022	29/11/2022
3.	Final-Presentation	14/12/2022	15/12/2022

MEETING WITH THE SUPERVISOR

Date of the meet	Mode	Comments by the Supervisor	Signature of the Supervisor