International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 1409-1413 www.ijaem.net ISSN: 2395-5252

Online Examination System

Ajmal S Nirar[1], Anu Suresh[2], Anzila Rahim[3], Gautham M[4]
Department of Computer Science Engineering, UKF College of Engineering and Technology, Kollam , kerala,

Submitted: 25-06-2021 Revised: 06-07-2021 Accepted: 09-07-2021

ABSTRACT
With the development of society, different kinds of examination appear constantly. The way of traditional examination has been unable to meet the needs of the developing education information. The role was divided into function modules by analyzing users' requirement. An online examination system based on Web was designed, which adopted B/S mode, used the IDE as the coding tool, combined with the MySQL database and related technology. It realized the user login, security authentication, question bank management, test paper management, online examination, announcement, sourcing, check results.

I. INTRODUCTION

The examination plays an important role in social life, which is an important means of evaluating the ability of people. The traditional way of examination with paper has occupied faculty much time and energy [1]. Especially in the remote network teaching, this way did more and more not adapt to the demand of education information and modernization. Online examination from the original examination computer aided to online examination based on C/S, until current online examination based on Web, its technology and practical application are also constantly developing. And the remote education will play a significant role in the field of education in the future, which also promotes the further development of the online examination system technology with broad prospects.

II. THE USER REQUIREMENT ANALYSIS

The system's overall goal was concluded by analyzing the target audience of the then it built the model of system structure.

A. The examinee role analysis

The system users include administrators, examinees and teachers. The system sets different permissions and operation interface for users with different roles; the users enter into the different interface, and have different operation according to different permissions [2].

System's ultimate goal is providing online examination service for examinees; therefore, the examinee is the core of the system. After examinees enter into the system through the correct login information, their own information could be viewed and the password could be modified. The examinee can view the test information, which includes exam subjects, exam time, whether it can take the exam, etc. When the start time of the test has not started, the system will remind that the exam has not started; when the exam could be conducted, the examinee can start test by selecting subjects click "into the test".

B. The teacher role analysis

In order to ensure the correctness of the information, the teacher can view personal information and change the password after login system, and then arranges the relevant examination. In order to ensure that the test is normal, teachers need to choose the right subjects, start time, test time and other information.

C. The administrator role analysis

Administrator is the main of system, and the operation authority of system is the highest. That is to say, administrator can add, modify, and delete target users after entering into system, which could make sure that the examinees and teachers can login system accurately with the corresponding role and related operation. Additionally, administrator can also manage the examination and resources, achieve uploading file and announcement, which is convenient to share resource [3].

III. EXISTING SYSTEM

Traditional offline, pen-paper based exams involve a lot of logistics and manual work which is tedious and prone to human error at multiple stages. With the increasing number of schools, universities, educational institutes and obviously students, conducting exams has become a tedious task and a few times inefficient.

In recent years, numerous incidents of paper leaks have been reported. Paper leaking is a common malpractice that plagues examinations, from the Central Board of Secondary Education

(CBSE) to the Railway Recruitment Board (RRB) to State and Union Public Service Commission's (PSCs) have suffered from paper leaks multiple times [2].

Now there are many online examination system available but the major drawback of these system is multiple choice questions had become a common format of examination for such system but in case of descriptive type exam, a system to conduct those examination are yet to be developed. And also an effective way of evaluation is still to be developed in these existing systems.

IV. PROPOSED SYSTEM

Constant technological development leads to a rapid expansion of the assessment industry. Many of the coaching institutes and learning centers are developing their interest in conducting online tests through the online examination software for their students instead of pen and paper-based exam.

Our project put forward a complete online examination system for universities and schools. Through this system both multiple choice questions and descriptive questions are solved in complete online mode.

V. SYSTEM DESIGN

a. Django architecture

For every website on the Internet, there are 3 main components or code partitions; Input Logic, Business Logic, and UI Logic.These partitions of code have specific tasks to achieve, the input logic is the dataset and how the data gets to organize in the database. It just takes that input and sends it to the database in the desired format. The Business logic is the main controller which handles the output from the server in the HTML or desired format. The UI Logic as the name suggests are the HTML, CSS and JavaScript pages.

When the traditional approach was used for programming all this code was written in a single file, i.e., every piece of code increases the webpage size, which is downloaded and rendered by the browser. This was not a big problem back in the time, the webpages were largely static and websites and didn't contain much multimedia and large coding. Also, this architecture poses difficulty for developers while testing and maintaining the project as everything is inside one file.

Now, time is changing and the websites are getting bigger and bigger while providing applications like cloud computing and online artificial intelligence training, online development environments and what not, these projects are all implemented using MVC architecture.

So, what is MVC? It is an acronym for Model View Controller. Don't worry we will learn every aspect of the MVC pattern and also relate it to Django.

MVC pattern is a Product Development Architecture. It solves the traditional approach's drawback of code in one file, i.e., that MVC architecture has different files for different aspects of our web application/ website.

The MVC pattern has three components, namely Model, View, and Controller.

This difference between components helps the developer to focus on one aspect of the web-app and therefore, better code for functionality with better testing, debugging and scalability.

b. Face Recognition with Python and Open CV

1. Face Recognition

In computer vision, one essential problem we are trying to figure out is to automatically detect objects in an image without human intervention. Face detection can be thought of as such a problem where we detect human faces in an image. There may be slight differences in the faces of humans but overall, it is safe to say that there are certain features that are associated with all the human faces. There are various face detection algorithms but Viola-Jones Algorithm is one of the oldest methods that is also used today and we will use the same later in the article.

Face detection is usually the first step towards many face-related technologies, such as face recognition or verification. However, face detection can have very useful applications. The most successful application of face detection would probably be photo taking. When you take a photo of your friends, the face detection algorithm built into your digital camera detects where the faces are and adjusts the focus accordingly.

2. Open CV

In the field of Artificial Intelligence, Computer Vision is one of the most interesting and Challenging tasks. Computer Vision acts like a bridge between Computer Software and visualizations around us. It allows computer software to understand and learn about the visualizations in the surroundings. For Example: Based on the color, shape and size determining the fruit. This task can be very easy for the human brain however in the Computer Vision pipeline, first we gather the data, then we perform the data processing activities and then we train and teach the model to understand how to distinguish between the fruits based on size, shape and color of fruit.

Currently, various packages are present to perform machine learning, deep learning and computer vision tasks. By far, computer vision is the best module for such complex activities. Open CV is an open-source library. It is supported by various programming languages such as R, Python. It runs on most of the platforms such as Windows, Linux and Mac OS.

c. Pattern Matching

Pattern matching is the act of checking a given sequence of tokens for the presence of the constituents of some pattern. In contrast to pattern recognition, the match usually has to be exact: "either it will or will not be a match." The patterns generally have the form of either sequences or tree structures. Uses of pattern matching include outputting the locations (if any) of a pattern within a token sequence, to output some component of the matched pattern, and to substitute the matching pattern with some other token sequence (i.e., search and replace).

Sequence patterns (e.g., a text string) are often described using regular expressions and matched using techniques such as backtracking [5].

Tree patterns are used in some programming languages as a general tool to process data based on its structure, e.g. C#, F#, Haskell, ML, Rust, Scala, Swift and the symbolic mathematics language Mathematics have special syntax for expressing tree patterns and a language construct for conditional execution and value retrieval based on it.

Often it is possible to give alternative patterns that are tried one by one, which yields a powerful conditional programming construct. Pattern matching sometimes includes support for guard.

VI. CODE DESIGN

i. Extracting Features From Face

First, you need to get a dataset or even create one of you own. Just make sure to arrange all images in folders with each folder containing images of just one person.

Next, save the dataset in a folder the same as you are going to make the file. Now here is the code:

```
from imutils import paths
import face_recognition
import pickle
import cv2
import os

#get paths of each file in folder named Images
#Images here contains my data(folders of various persons)
imagePaths = list(paths.list_images('Images'))
knownEncodings = []
knownNames = []
# loop over the image paths
for (i, imagePath) in enumerate(imagePaths):
    # extract the person name from the image path
    name = imagePath.split(os.path.sep)[-2]
    # load the input image and convert it from BGR (OpenCV ordering)
    # to dlib ordering (RGB)
    image = cv2.imread(imagePath)
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    #Use Face_recognition to locate faces
    boxes = face_recognition.face_locations(rgb,model='hog')
    # compute the facial embedding for the face
    encodings = face_recognition.face_encodings(rgb, boxes)
    # loop over the encodings
    for encoding in encodings:
        knownEncodings.append(encoding)
        knownNames.append(name)
#save emcodings along with their names in dictionary data
data = {"encodings": knownEncodings, "names": knownNames}
#use pickle to save data into a file for later use
f = open("face_enc", "wb")
```

```
f.write(pickle.dumps(data))
f.close()
```

ii. Face Recognition In Live webcam feed

```
import face_recognition
import imutils
import pickle
import time
import cv2
import os

#find path of xml file containing haarcascade file
cascPathface = os.path.dirname(
 cv2.__file__) + "/data/haarcascade_frontalface_alt2.xml"
# load the harcaascade in the cascade classifier
faceCascade = cv2.CascadeClassifier(cascPathface)
# load the known faces and embeddings saved in last file
data = pickle.loads(open('face_enc', "rb").read())

print("Streaming started")
video_capture = cv2.VideoCapture(0)
# loop over frames from the video file stream
while True:
   # grab the frame from the threaded video stream
   ret, frame = video_capture.read()
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
   faces = faceCascade.detectMultiScale(gray,
                                        scaleFactor=1.1,
                                        minNeighbors=5,
                                        minSize=(60, 60),
                                        flags=cv2.CASCADE_SCALE_IMAGE)
    # convert the input frame from BGR to RGB
   rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
   # the facial embeddings for face in input
   encodings = face_recognition.face_encodings(rgb)
   names = []
   # loop over the facial embeddings incase
   # we have multiple embeddings for multiple fcaes
   for encoding in encodings:
     #Compare encodings with encodings in data["encodings"]
     #Matches contain array with boolean values and True for theembeddings it matches closely
   #and False for rest
      matches = face_recognition.compare_faces(data["encodings"],
       encoding)
      #set name =inknown if no encoding matches
      name = "Unknown"
      # check to see if we have found a match
      if True in matches:
         #Find positions at which we get True and store them
         matchedIdxs = [i for (i, b) in enumerate(matches) if b]
         counts = {}
         # loop over the matched indexes and maintain a count for
         # each recognized face face
         for i in matchedIdxs:
            #Check the names at respective indexes we stored in matchedIdxs
 name = data["names"][i]
            #increase count for the name we got
            counts[name] = counts.get(name, 0) + 1
         #set name which has highest count
         name = max(counts, key=counts.get)


      # update the list of names
      names.append(name)
      # loop over the recognized faces
      for ((x, y, w, h), name) in zip(faces, names):
         # rescale the face coordinates
         # draw the predicted face name on the image
         cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
         cv2.putText(frame, name, (x, y), cv2.FONT_HERSHEY_SIMPLEX,
          0.75, (0, 255, 0), 2)
   cv2.imshow("Frame", frame)
   if cv2.waitKey(1) &amp; 0xFF == ord('q'):
      break
video_capture.release()
cv2.destroyAllWindows()
```

iii. Face Recognition In Images

```
import face_recognition
import imutils
import pickle
import time
import cv2
import os

#find path of xml file containing haarcascade file
cascPathface = os.path.dirname(
 cv2.__file__) + "/data/haarcascade_frontalface_alt2.xml"
# load the harcaascade in the cascade classifier
faceCascade = cv2.CascadeClassifier(cascPathface)
# load the known faces and embeddings saved in last file
data = pickle.loads(open('face_enc', "rb").read())
```

```
#Find path to the image you want to detect face and pass it here
image = cv2.imread(Path-to-img)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
#convert image to Greyscale for haarcascade
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
faces = faceCascade.detectMultiScale(gray,
                                     scaleFactor=1.1,
                                     minNeighbors=5,
                                     minSize=(60, 60),
                                     flags=cv2.CASCADE_SCALE_IMAGE)
 # the facial embeddings for face in input
encodings = face_recognition.face_encodings(rgb)
names = []
# loop over the facial embeddings incase
# we have multiple embeddings for multiple fcaes
for encoding in encodings:
  #Compare encodings with encodings in data["encodings"]
  #Matches contain array with boolean values and True for the embeddings it matches closely
  #and False for rest
  matches = face_recognition.compare_faces(data["encodings"],
  encoding)
  #set name =inknown if no encoding matches
  name = "Unknown"
  # check to see if we have found a match
  if True in matches:
    #Find positions at which we get True and store them
    matchedIdxs = [i for (i, b) in enumerate(matches) if b]
    counts = {}
    # loop over the matched indexes and maintain a count for
    # each recognized face face
    for i in matchedIdxs:
      #Check the names at respective indexes we stored in matchedIdxs
      name = data["names"][i]
      #increase count for the name we got
      counts[name] = counts.get(name, 0) + 1
      #set name which has highest count
      name = max(counts, key=counts.get)
    # update the list of names
    names.append(name)
    # loop over the recognized faces
    for ((x, y, w, h), name) in zip(faces, names):
      # rescale the face coordinates
      # draw the predicted face name on the image
      cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
      cv2.putText(image, name, (x, y), cv2.FONT_HERSHEY_SIMPLEX,
       0.75, (0, 255, 0), 2)
  cv2.imshow("Frame", image)
 cv2.waitKey(0)
```

VII. CONCLUSION

Online live examination system is the way of improving examination activities fast and efficient. The most beautiful feature is to monitor the exam from control room and instantly get the total examination picture. The system is relatively simple to operate, and has a good interface. The system is added the completion and the realization of the automatic grading of short questions.

ACKNOWLEDGEMENT

This paper was supported by UKF College Of Engineering And Technology under APJ Abdul Kalam Technological University. In addition, the authors would like to thank the reviewers for their valuable comments and suggestions.

REFERENCE

[1]. Akinsanmi O., Agbaji, O.T. Ruth and M.B. Soroyewun (2010), “Development of an E-Assessment Platform for Nigerian Universities”, Research Journal Applied Sciences,Engineering and Technology 2(2): Page 170-175, ISSN: 2040-7467.

[2]. Magdi Z. Rashad, Mahmoud S. Kandil , Ahmed E. Hassan, and Mahmoud A. Zaher (2010), “An Arabic Web-Based Exam Management System”, International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 10 No: 01. Page 48-55.

[3]. Wkun. the design and implementation of online examination system based on .net [d]. shandong university, 2013.

[4]. Z. Yong-Sheng, F. Xiu-Mei and B. Ai-Qin, "The Research and Design of Online Examination System," 2015 7th International Conference on Information Technology in Medicine and Education (ITME), Huangshan, 2015, pp. 687-691, doi: 10.1109/ITME.2015.96.

[5]. https://www.google.com/url?sa=t&source=web&rct=j&url=http://ieeexplore.ieee.org/document/7429241&ved=2ahUKEwj1suHlOnsAhVI5nMBHYyMD3YQFjAAegQIARAB&usg=AOvVaw32Caj471VbqdFVO2hzp8UC