

AUTOMATIC CERTIFICATE GENERATION SYSTEM

A Project Report

Submitted in Partial Fulfillment of the Requirement for the Award of the
Degree of

BACHELOR OF TECHNOLOGY (COMPUTER SCIENCE AND ENGINEERING)

To



**Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY
LUCKNOW**

Submitted by

SHAMBHAVI SRIVASTAVA
University Roll No.1816210069

SEEMA VERMA
University Roll No. 1816210065

SHREYA AGRAWAL
University Roll No. 1816210079

SHIKHA SINGH
University Roll No. 1901620109010

Under the Supervision of

MS. ANIMA SRIVASTAVA

Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SHAMBHUNATH INSTITUTE OF ENGINEERING AND
TECHNOLOGY, PRAYAGRAJ**

JUNE 2022

CANDIDATE'S DECLARATION

We, **SHAMBHAVI SRIVASTAVA** (1816210069), **SEEMA VERMA** (1816210065), **SHREYA AGRAWAL**(1816210079), **SHIKHA SINGH**(1901620109010), students of B.Tech of Computer Science and Engineering hereby declared that we own the full responsibility for the information, results etc provided in this project titled “**AUTOMATIC CERTIFICATE GENERATION SYSTEM**” submitted to Dr. A.P. J Abdul Kalam Technical University, Lucknow for award of B.Tech (Computer Science and Engineering) degree. We have taken care in all respect to honour the intellectual property right and have acknowledged the contributions of others for using them in this academic purpose. We further declared that in case of any violation of intellectual property right or copyright, we as the candidate would be fully responsible for the same. Our supervisor and institute should not be held for full or partial violation of copy right if found at any stage of our degree.

Date:

Place:

SHAMBHAVI SRIVASTAVA

(1816210069)

SEEMA VERMA

(1816210065)

SHREYA AGRAWAL

(1816210079)

SHIKHA SINGH

(1901620109010)



SHAMBHUNATH INSTITUTE OF ENGINEERING & TECHNOLOGY

CERTIFICATE

This is to certify that the project titled “**AUTOMATIC CERTIFICATE GENERATION SYSTEM**” is the bonafide work carried out by **Shambhavi Srivastava (1816210069), Seema Verma (1816210065), Shreya Agrawal (1816210079) , Shikha Singh(1901620109010)** student of B.Tech(C.S.E.) from Shambhunath Institute of Engineering & Technology, Jhalwa, Allahabad, during the academic year 2021-2022, in partial fulfillment of the requirements for the award of the degree of Bachelorette of Technology in Computer Science and Engineering. The project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

I hereby certify their work as Excellent/Good/Satisfactory.

COMMITTEE ON FINAL EXAMINATION FOR EVALUATION OF PROJECT

EXTERNAL EXAMINER.....

INTERNAL EXAMINER.....

PROJECT GUIDE.....

HEAD OF DEPARTMENT.....

CERTIFICATE

This is to certify that the project work entitled “**AUTOMATIC CERTIFICATE GENERATION SYSTEM**”, submitted by **Shambhavi Srivastava (1816210069)**, **Seema Verma (1816210065)**, **Shreya Agrawal (1816210079)** , **Shikha Singh(1901620109010)** to the Dr. A.P.J. Abdul Kalam Technical University, Lucknow, for the partial fulfilment of the requirement for the award of **Bachelor of Technology (Computer Science and Engineering)** degree, is a record of student’s own study carried out under my supervision and guidance.

This project has not been submitted to any other university or institution for the award of any other degree.

SUPERVISOR

(Mrs. Anima Srivastava)

Assistant Professor

Department of CSE

SIET, Prayagraj

ACKNOWLEDGMENT

We, **Shambhavi Srivastava (1816210069)**, **Seema Verma (1816210065)**, **Shreya Agrawal (1816210079)**, **Shikha Singh(1901620109010)**, are grateful to the management of Shambhunath Institute of Engineering & Technology for providing us an opportunity to undertake our major project in its prestigious college. First and foremost, we would like to thank our mentor **Ms. Anima Srivastava** who guided us in doing this project. She provided us with valuable advice and helped us in difficult periods. Her motivation and help contributed tremendously to the successful completion of the project. Besides, we would like to thank all the teachers who helped us by giving us advice and providing the equipment which we needed. . We must make special mention of our **H.O.D. Dr. Radha Raman Chandan**, for providing us a platform to complete our project successfully. Also, we would like to thank each of us for their support. Without that support we couldn't have succeeded in completing this project. At last, but not the least, we would like to thank everyone who helped and motivated us to work on this project.

ABSTRACT

Nowadays several numbers of certificate generation systems have been developed to solve the bit of difficulties faced in the manual system. These systems are functioning based on the many predefined templates & their formats by the system developer. However, systems allow end-users to define templates or their formats as per our requirements. This research work allow an end user to define certificate template & template format without the use of XML knowledge by clicking few buttons & typing from the system GUI, verifying the certificate and generating one or more certificate(s) simultaneously in an sequence manner.

Keywords: Certificate, Certificate Generation, Certificate Template, Certificate Verification, Participant, Template Format.

LIST OF ABBREVIATIONS

SYMBOLS	DESCRIPTION
ACG	Automatic Certificate Generation
XML	Extensible Markup Language
GUI	Graphical User Interface
CGPA	Cumulative Grade Point Average
PDF	Portable Document Format
SQL	Structured Query Language
CBGS	Credit Based Grading System
CSV	Comma-Separated Values

CONTENTS

1. CANDIDATE’S DECLARATION	i
2. CERTIFICATE	ii
3. ACNOWLEDGMENT	iii
4. ABSTRACT	iv
5. LIST OF ABBREVIATION	v
6. TABLE OF CONTENTS	vi
7. LIST OF FIGURES	vii
8. LIST OF TABLES	viii
9. CHAPTER 1: INTRODUCTION	1
10. CHAPTER 2: LITERATURE REVIEW	2-5
11. CHAPTER 3: PROPOSED SYSTEM	6-7
11. a. SYSTEM METHODOLOGY	7
11. b. SYSTEM ALGORITHM	8
12. CHAPTER 4: EXPERIMENTAL WORK	9-19
13. CHAPTER 5: RESULT & DISCUSSION	20
14. CHAPTER 6: COMPARISION	21
15. CHAPTER 7: CONCLUSION	22
16. REFERENCES	23

LIST OF FIGURES

Figure No.	Figure Description	Page No.
1.	System Architecture of ACG	7
1.1	Code for the login system	9
1.2	Code for the login system	10
1.3	Code for the login system	10-11
1.4	Outcome of the login system	12
2.	Code for admin dashboard	13
2.1	Code for the login system	14
2.2	Code for the login system	15
2.3	Code for the login system	16
2.4	Outcome of the admin dashboard for the home tab	17
3.	Code for the mailing certificate to the participants	18
3.1	Certificate template format	19

LIST OF TABLES

Table No.	Table Description	Page No.
1	Literature Review	2-5

CHAPTER 1

INTRODUCTION

In today's several organizations and many companies are performing their daily activities more efficiently using the software applications founded by use of computer science knowledge. The software developer tries to make it possible for people with illiterate computer knowledge to be able to operate computers by creating some application that help them via training to learn how to use them for a specific task. Application software is a computer program designed to carry out a specific task other than one related to the operation of the computer itself, typically to be used by end-users.

In this study, an application called Automatic Certificate Generation system (ACG) is developed based on the Graphical user Interface (GUI). This project focuses on certificate generation system by enabling multiple certificates generation simultaneously, & allow to end-users to define certificate template & template format as per their requirement. Certificate Generation System is homogeneous to document generation system. In this study, an application called Automatic Certificate Generation System (ACG) using Jupyter Notebook (anaconda3).Automating this job can easily save tons of time and manual work and thus also reducing the error rate.

Certificate Generation System is homogeneous to document generation system. Srushti [1] Developed a system for "generating marksheets of students" (results of students).

The system administrator has the privilege to input the obtain scores for students into the System . The system will calculate the CGPA based on the university grading system, "generate the mark sheet" for students respectively in "Portable Document Format" PDF.

According to ²Abraham [2], there is a need for certificate verification system to ease the process of verifying the certificate and reduce certificate forgery. Abraham developed a system that can be implemented as a standalone app or embedded in school official website depending on how the institute decides to use it.

¹ Srushti, A. S. (2014), Certificate Generation System.

² Abraham , Nwachukwu, K. C. (2015), Designing An Automatic Web-Based Certificate Verification System For Institutions.

CHAPTER 2

LITERATURE REVIEW

S.N	Paper Title	Authors	Year	Methods	Validation	Limitations
1.	Automated batch certificate generation & verification system	Ahmed dalhatuyusuf [3]	November 2017	This research work enables an end user to define certificate template & template format by using the GUI application.	The system admin store student record in a system & verifying certificate by providing matric number on certificate s.	<p>There is an issue of a template in the existing because the template is predefined by the system developer.</p> <p>Another drawback to the existing system is a unit generation of certificate.</p>
2.	Automated report generation	Dejan gjorgjevikj [4]	January 2011	This applicative solution for	The Architecture	

	system			generating reportsfrom	Solution Composed	
--	--------	--	--	---------------------------	----------------------	--

				<p>templates tries to deal efficiently with this issue effective architectural design to generate reports with their own criteria & design from their SQL databases.</p>	<p>of 3 main module: document parser, tag engine & tag generator. The first two module are the core of our application. The document parser module is designed to import a report template.</p>	
3.	Certificate generation system	Srushti A. shimpi [1]	February 2014	<p>This research is mainly based in the database technology & the credit-based grading system (CBGS).</p>	<p>Digital signature-based mark sheet generator: the outcome of sig. Process is digital sig. It provides authentic</p>	<p>Digital sig algorithm & protocol: do not inherently provide the certainty about the date & time at which</p>

					ty & integrity of data.	underlay ingdocumen t t was sig. Lack of user-friendly interface.
4.	Certificate generation system	Bharti chikonka r [7]	August 2020	This paper presents an e-certificate generation system by the use of GUI application For the generation of certificates in bulk.	For the validation a login &signup systemare available to authenticate the access to the users for the accounts.	

CHAPTER 3

PROPOSED SYSTEM

This work presents four main modules as mentioned below : -

- A. Template creation
- B. Certificate generation
- C. Certificate verification
- D. Upload module

A. Template Creation:

This is a system module that allows an end-user to define certificate template and template format in system GUI by clicking buttons and typing[3,7]. This module over comes an issue of the template in certificate generation system because the template is predefined by the system programmer [4]. Therefore, in a situation where there is need to modify the template or generate a certificate for a different purpose the different application needs to be developed for that purpose. However, the work of Dejan [4] allows an end-user to define template and template format using XML knowledge. In contrast to this work that does not require knowledge of XML.

B. Certificate Generation:

This module entails with adapting certificate template by adding participant(s) details on it. The record of participants can be fetched from CSV file. This module can be used to generate one or more certificate(s) concurrently, by selecting participant using a checkbox in the system. A unique number will be assigned to each certificate in this module.

C. Certificate Verification:

This module ensures the authenticity of a certificate. The module communicates with the system database to get a copy of a certificate from the database using the unique number assigned to certificates in the generation phase.

D. Upload Module:

This is a module that can be used to import information of participant into the system database (CSV file).

3. a. SYSTEM METHODOLOGY

3. b. SYSTEM ARCHITECTURE

The given diagram below shows the overall system architecture.

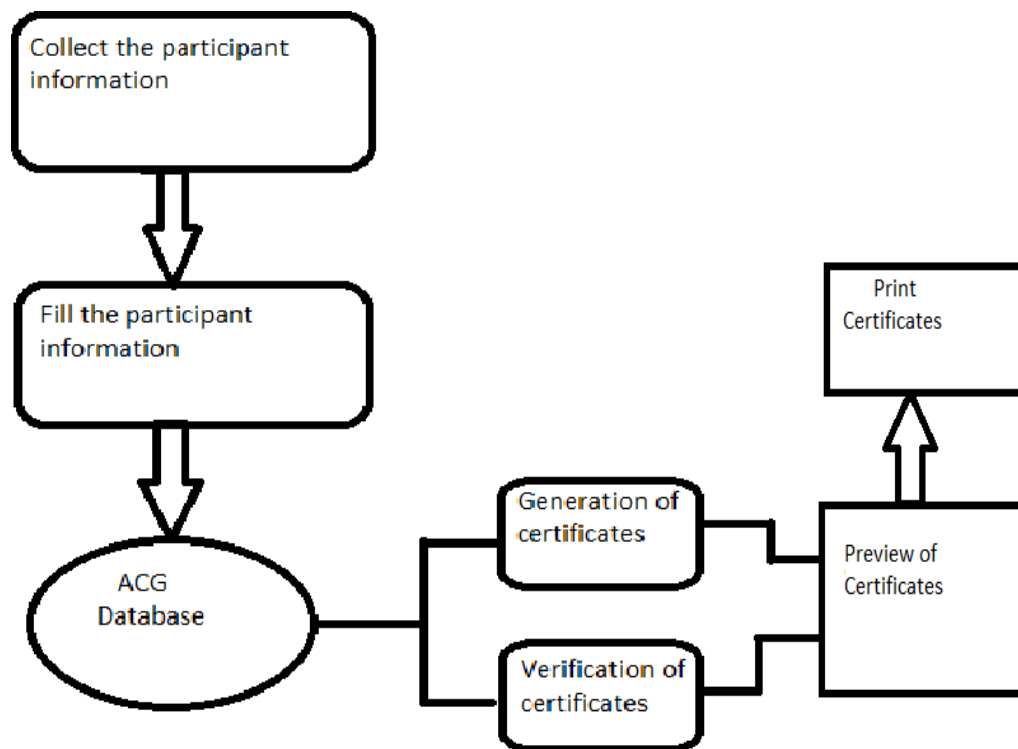


Figure1: System architecture of ACG

3. c. SYSTEM ALGORITHM

This is how the proposed system will work.

Step 1: Provide username and password to access the system.

Step 2: Check if the username and password are correct then further proceed if not then go back to step 1 else go to step 3.

Step 3: Access the system by getting the required information from the database, modifying data in the System.

Step 4: Then the system will generate certificates for all student in a sequence manner.

Step 5: Preview of the student certificate.

Step 6: Personnel/user can log out of the system.

CHAPTER 4

EXPERIMENTAL WORK

```
In [1]: from tkinter import*
from tkinter import messagebox
from PIL import Image
class LoginPage:
    def __init__(self,root):
        self.root=root
        self.root.title("Login System")
        #self.root.iconbitmap("image.ico")
        self.root.geometry("1199x700+100+50")
        self.root.resizable(False,False)
        #BG Image

        #self.bg=ImageTk.PhotoImage(file="C:\\Users\\Dell\\OneDrive\\Desktop\\Certificates\\login1.jpg")
        #self.bg_image=Label(self.root,image=self.bg).place(x=0,y=0,relwidth=1,relheight=1)
        self.root.configure(bg='blue')

        #Login Frame

        Frame_login=Frame(self.root,bg='white')
        Frame_login.place(x=150,y=150,height=340,width=500)

        title=Label(Frame_login,text='Login Here',font=('Impact',35,'bold'),fg='#d77337',bg='white').place(x=90,y=30)

        desc=Label(Frame_login,text='Admin',font=('Goudy old style',15,'bold'),fg='#d25d17',bg='white').place(x=90,y=100)
        lbl_user=Label(Frame_login,text='Username',font=('Goudy old style',15,'bold'),fg='gray',bg='white').place(x=90,y=140)
        self.txt_user=Entry(Frame_login,font=('times new roman',15),bg='lightgray')
        self.txt_user.place(x=90,y=170,width=350,height=35)

        lbl_pass=Label(Frame_login,text='Password',font=('Goudy old style',15,'bold'),fg='gray',bg='white').place(x=90,y=210)
        self.txt_pass=Entry(Frame_login,font=('times new roman',15),show='*',bg='lightgray')
        self.txt_pass.place(x=90,y=240,width=350,height=35)

        forget_btn=Button(Frame_login,text='Forgot Password?',cursor="hand2",command=self.forget_password,bg='white',fg='#d77337')
        Login_btn=Button(self.root,command=self.loginpage_function,cursor="hand2",text='Login',fg='white',bg='#d77337',font=('times new roman',15,'bold'))
        Register_btn=Button(Frame_login,text='Register New Account',cursor="hand2",command=self.my_window,bg='white',fg='#d77337',font=('times new roman',15,'bold'))
```

Figure 1: code for the login system

```

Login_btn=Button(self.root,command=self.loginpage_function,cursor="hand2",text="Login",fg="white",bg="#d77337",font=(
Register_btn=Button(Frame_login,text="Register New Account",cursor="hand2",command=self.my_window,bg="white",fg="#d77337")

def loginpage_function(self):
    if self.txt_pass.get()==" or self.txt_user.get()=="":
        messagebox.showerror("Error","All fields are required",parent=self.root)
    elif self.txt_user.get()!="autocertigen" or self.txt_pass.get()!="auto@321":
        messagebox.showerror("Error","Invalid Username/Password",parent=self.root)
    else:
        messagebox.showinfo("Welcome",f"Welcome {self.txt_user.get()}\nYou have successfully login",parent=self.root)

def forget_password(self):
    self.root2=Tk()
    self.root2.title("Forget Password")
    self.root2.geometry("350x350+450+150")
    self.root2.config(bg="white")
    self.root2.resizable(False,False)

    t=Label(self.root2,text="Forget Password",font=("times new roman",20,"bold"),bg="white",fg="red").place(x=0,y=10,relwidth

    lbl_newpass=Label(self.root2,text='User Name',font=('Goudy old style',15,'bold'),fg='gray',bg='white').place(x=50,y=100)
    self.txt_newpass=Entry(self.root2,font=('times new roman',15),show='*',bg='lightgray')
    self.txt_newpass.place(x=50,y=130,width=250,height=35)

    lbl_confirmpass=Label(self.root2,text='New password',font=('Goudy old style',15,'bold'),fg='gray',bg='white').place(x=50,
    self.txt_confirmpass=Entry(self.root2,font=('times new roman',15),show='*',bg='lightgray')
    self.txt_confirmpass.place(x=50,y=210,width=250,height=35)

    Changepassword_btn=Button(self.root2,command=self.forget_password_function,cursor="hand2",text="Change Password",fg="whit

def forget_password_function(self):
    if self.txt_pass.get()==" or self.txt_user.get()=="":
        messagebox.showerror("Error","All fields are required",parent=self.root2)
    elif self.txt_user.get()=="autocertigen" or self.txt_pass.get()!="auto@321":
        messagebox.showinfo("Welcome",f"Welcome {self.txt_user.get()}\nYour password is change successfully",parent=self.root

```

Figure 1.1: code for the login system

```

    elif self.txt_user.get()!="autocertigen" or self.txt_pass.get()!="auto@321":
        messagebox.showinfo("Welcome",f"Welcome {self.txt_user.get()}\nYour password is change successfully",parent=self.root

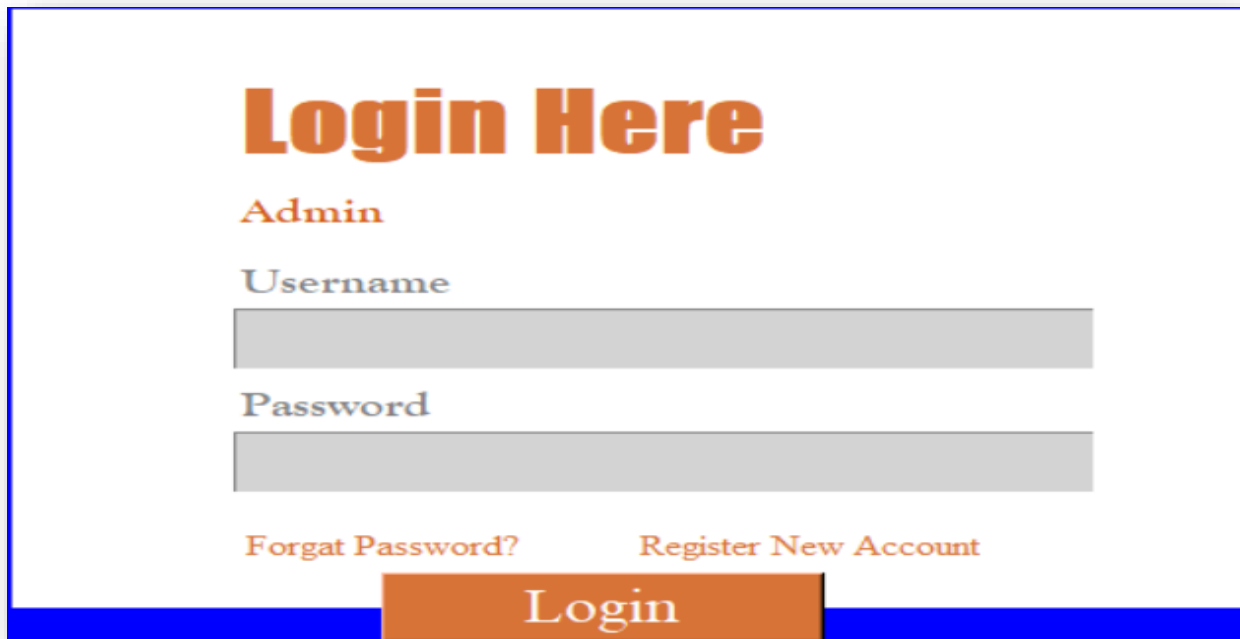
def my_window(self):
    self.root1=Tk()
    self.root1.title("Register")
    self.root1.geometry("500x500+450+150")
    self.root1.config(bg="white")
    self.root1.resizable(False,False)

    label_0 = Label(self.root1, text="Registration form",width=20,font=("bold", 20),bg='white')
    label_0.place(x=90,y=53)
    label_1 = Label(self.root1, text="FullName",width=20,font=("bold", 10),bg='white')
    label_1.place(x=80,y=130)
    entry_1 = Entry(self.root1)
    entry_1.place(x=240,y=130)
    label_2 = Label(self.root1, text="Email",width=20,font=("bold", 10),bg='white')
    label_2.place(x=68,y=180)
    entry_2 = Entry(self.root1)
    entry_2.place(x=240,y=180)
    label_3 = Label(self.root1, text="Gender",width=20,font=("bold", 10),bg='white')
    label_3.place(x=70,y=230)
    var = IntVar()
    Radiobutton(self.root1, text="Male",padx = 5, variable=var, value=1).place(x=235,y=230)
    Radiobutton(self.root1, text="Female",padx = 20, variable=var, value=2).place(x=290,y=230)
    label_4 = Label(self.root1, text="Password",width=20,font=("bold", 10),bg='white')
    label_4.place(x=70,y=280)
    entry_2 = Entry(self.root1)
    entry_2.place(x=240,y=280)
    Button(self.root1, text="Submit",width=20,bg='brown',fg='white').place(x=180,y=380)
    print("registration form seccussfully created...")

root=Tk()
obj=LoginPage(root)
root.mainloop()

```

Figure 1.2: code for the login system

A login system interface with a white background and a blue border. At the top, the text "Login Here" is displayed in a large, bold, orange font. Below this, the word "Admin" appears in a smaller, orange font. The interface includes two input fields: "Username" and "Password", both with light gray rectangular boxes. Below the "Password" field, there are two links: "Forgat Password?" and "Register New Account", both in orange text. At the bottom, there is a blue horizontal bar with a central orange button labeled "Login" in white text.

Login Here

Admin

Username

Password

[Forgat Password?](#) [Register New Account](#)

Login

Figure1.3 outcome of the login system

```

In [8]: from tkinter import *
from tkinter import ttk,font
from tkinter import filedialog as fd
from tkinter.filedialog import askopenfile
from tkinter import colorchooser
from PIL import Image,ImageTk
import os
root1=Tk()
#App Title
root1.title("CertiGen")
root1.geometry("1199x700+100+50")
#ttk.Label(root, text="Automatic Certificate Generation System").pack()
#Create Panedwindow
panedwindow=ttk.Panedwindow(root1, orient=HORIZONTAL) |
panedwindow.pack(fill=BOTH, expand=True)
#Create Frames
fram1=ttk.Frame(panedwindow,width=400,height=300,relief=SUNKEN)
fram2=ttk.Frame(panedwindow,width=400,height=400,relief=SUNKEN)
panedwindow.add(fram1)
panedwindow.add(fram2)

img1 = ImageTk.PhotoImage(Image.open("C:\\Users\\Dell\\OneDrive\\Desktop\\Certificates\\project\\CG.png"))
label = Label(fram1,image=img1)
label.place(x=0,y=0,relwidth=1,relheight=1)

TABS=ttk.Notebook(fram2)
TAB1=Frame(TABS)
TAB2=Frame(TABS)
TABS.add(TAB1,text="Home")
TABS.add(TAB2,text="Customize")
TAB3=Frame(TABS)
TAB4=Frame(TABS)
TABS.add(TAB3,text="Destination")
TABS.add(TAB4,text="Email")
TAB5=Frame(TABS)

```

Activate V
Go to Setting

Figure2: code for admin dashboard

```

def Open_File():
    file = fd.askopenfile(mode='r', initialdir="/", title="Select file", filetypes=(("txt files", "*.txt"), ("All files", "*.*")))
    if file:
        filepath = os.path.abspath(file.name)
        Label(TAB1, text="The File is located at : " + str(filepath), font=('Arial 11')).place(x=10, y=100)
        label=Label(TAB1, text='Choose Your Excel File', font=('Arial', 25, 'bold'), fg='black').place(x=10, y=10)
        ttk.Button(TAB1, text="Browse", command=Open_File()).place(x=510, y=50)
def Open_File():
    file = fd.askopenfile(mode='r', initialdir="/", title="Select file", filetypes=(("txt files", "*.txt"), ("All files", "*.*")))
    if file:
        filepath = os.path.abspath(file.name)
        Label(TAB1, text="The File is located at : " + str(filepath), font=('Arial 11')).place(x=10, y=350)
        label=Label(TAB1, text='Choose Your Blank Certificate', font=('Arial', 25, 'bold'), fg='black').place(x=10, y=210)
        ttk.Button(TAB1, text="Browse", command=Open_File()).place(x=510, y=290)
        #Next_btn=Button(tab1, cursor="hand2", text='Next', fg='black', font=('times new roman', 15)).place(x=510, y=470, width=100, height=25)
        title=Label(TAB2, text='Choose Columns ', font=('Arial', 20, 'bold'), fg='black').place(x=10, y=10)
        listbox=Listbox(TAB2, selectmode=SINGLE)
        listbox.insert(1, 'Name')
        listbox.insert(2, 'Role')
        listbox.insert(3, 'Branch')
        listbox.insert(4, 'College')
        listbox.insert(5, 'Email Id')
        listbox.place(x=10, y=50)
def my_window():
    top=Toplevel()
    top.title("Selet Region")
    top.geometry("800x700")
    c=Canvas(top, height=200, width=200)
    filename=ImageTk.PhotoImage(file="C:\\Users\\Dell\\OneDrive\\Desktop\\Certificates\\certi.png")
    lbl=Label(top, image=filename)
    lbl.place(x=0, y=0, relwidth=1, relheight=1)
    c.pack()
    top.resizable(False, False)
    #image_0 = Image.open("C:\\Users\\Dell\\OneDrive\\Desktop\\Certificates\\certi.PNG")
    bck_end=ImageTk.PhotoImage(image)
    lbl=Label(top, image=bck_end)
    lbl.place(x=0, y=0)
    Select_Regio btn=Button(TAB2, cursor="hand2", text="Select Region", command=my_window, fg='black', font=('times new roman', 15)).place

```

Activate Windows
Go to Settings to activate Windows.

Figure2.1 code for admin dashboard


```
Select Region_btn=Button(TAB2,cursor="hand2",text='Select Region',command=my_window,fg='black',font=('times new roman',15)).place(x=10,y=500,width=100,height=25)
#Back_btn=Button(tab2,cursor="hand2",text='Back',fg='black',font=('times new roman',15)).place(x=10,y=500,width=100,height=25)
my_font=font.families()
title=Label(TAB2,text='Configure Fonts',font=('Arial',20,'bold'),fg='black').place(x=270,y=10)
title=Label(TAB2,text='Font Family',font=('Arial',15),fg='black').place(x=270,y=50)
com=ttk.Combobox(TAB2,width=40,values=my_font)
com.current(12)
com.place(x=270,y=90)
title=Label(TAB2,text='Font Size',font=('Arial',15),fg='black').place(x=270,y=130)
listbox=Listbox(TAB2,selectmode=SINGLE)
listbox.insert(1,8)
listbox.insert(2,10)
listbox.insert(3,12)
listbox.insert(4,14)
listbox.insert(5,16)
listbox.insert(6,18)
listbox.insert(7,20)
listbox.insert(8,36)
listbox.insert(9,48)
listbox.place(x=270,y=160)
title=Label(TAB2,text='Font Color',font=('Arial',15),fg='black').place(x=270,y=330)
def click():
    color=colorchooser.askcolor()
    button=Button(TAB2,text='Choose Color',width=20,command=click)
    button.place(x=270,y=370)
#Next_btn=Button(tab2,cursor="hand2",text='Next',fg='black',font=('times new roman',15)).place(x=390,y=490,width=100,height=25)
label_0=Label(TAB3,text='Choose Your Destination Folder',font=('Arial',25,'bold'),fg='black').place(x=10,y=50)
def open_folder():
    if folder:
        folder = fd.askdirectory(initialdir=os.path.normpath("C://"))
        Label(TAB3, text="The File is located at : " + str(folder),font=('Arial 11')).place(x=50,y=150)
        ttk.Button(TAB3, text="Browse", command=open_folder).place(x=510,y=50)
label_1=Label(TAB4,text="Your Email Id:-",font=('Arial',14,'bold'),fg='black').place(x=50,y=50)
label_2=Label(TAB4,text="Your Password:-",font=('Arial',14,'bold'),fg='black').place(x=50,y=80)
label_3=Label(TAB4,text="Email Subject:-",font=('Arial',14,'bold'),fg='black').place(x=50,y=110)
label_4=Label(TAB4,text="Message:-",font=('Arial',14,'bold'),fg='black').place(x=50,y=140)
label_5=Label(TAB4,text="Attachment Name:-",font=('Arial',14,'bold'),fg='black').place(x=50,y=170)
label_6=Label(TAB4,text="Email Column:-",font=('Arial',14,'bold'),fg='black').place(x=50,y=200)
```

Activate Windows
Go to Settings to activate Windows.

Figure2.2 code for admin dashboard

```
label_4=Label(TAB4,text="Message:-",font=('Arial',14,'bold'),fg='black').place(x=50,y=140)
label_5=Label(TAB4,text="Attachment Name:-",font=('Arial',14,'bold'),fg='black').place(x=50,y=170)
label_6=Label(TAB4,text="Email Column:-",font=('Arial',14,'bold'),fg='black').place(x=50,y=200)
ent0=Entry(TAB4,width=50)
ent0.place(x=250,y=60)
ent1=Entry(TAB4,width=50)
ent1.place(x=250,y=90)
ent2=Entry(TAB4,width=50)
ent2.place(x=250,y=120)
ent3=Entry(TAB4,width=50)
ent3.place(x=250,y=150)
ent4=Entry(TAB4,width=50)
ent4.place(x=250,y=180)
listbox=Listbox(TAB4,selectmode=SINGLE)
listbox.insert(1,'Name')
listbox.insert(2,'Role')
listbox.insert(3,'Branch')
listbox.insert(4,'College')
listbox.insert(5,'Email Id')
listbox.place(x=250,y=210)
label=Label(TAB5,text="Done",font=('Arial',25,'bold'),fg='black').place(x=10,y=10)
label=Label(TAB5,text="All Certificates generated successfully",font=('Arial',10,'bold'),fg='black').place(x=10,y=60)
label=Label(TAB5,text="All email send successfully",font=('Arial',10,'bold'),fg='black').place(x=10,y=80)
button=Button(TAB5,cursor="hand2",text="Show In Folder",width=20)
button.place(x=10,y=250)
def Logout():
    root1.destroy
button=Button(TAB1,cursor="hand2",text="Logout",bg='blue',command=Logout,width=10)
button.place(x=700,y=5)
#alling Main()
root1.mainloop()
```

Figure2.3 code for admin dashboard

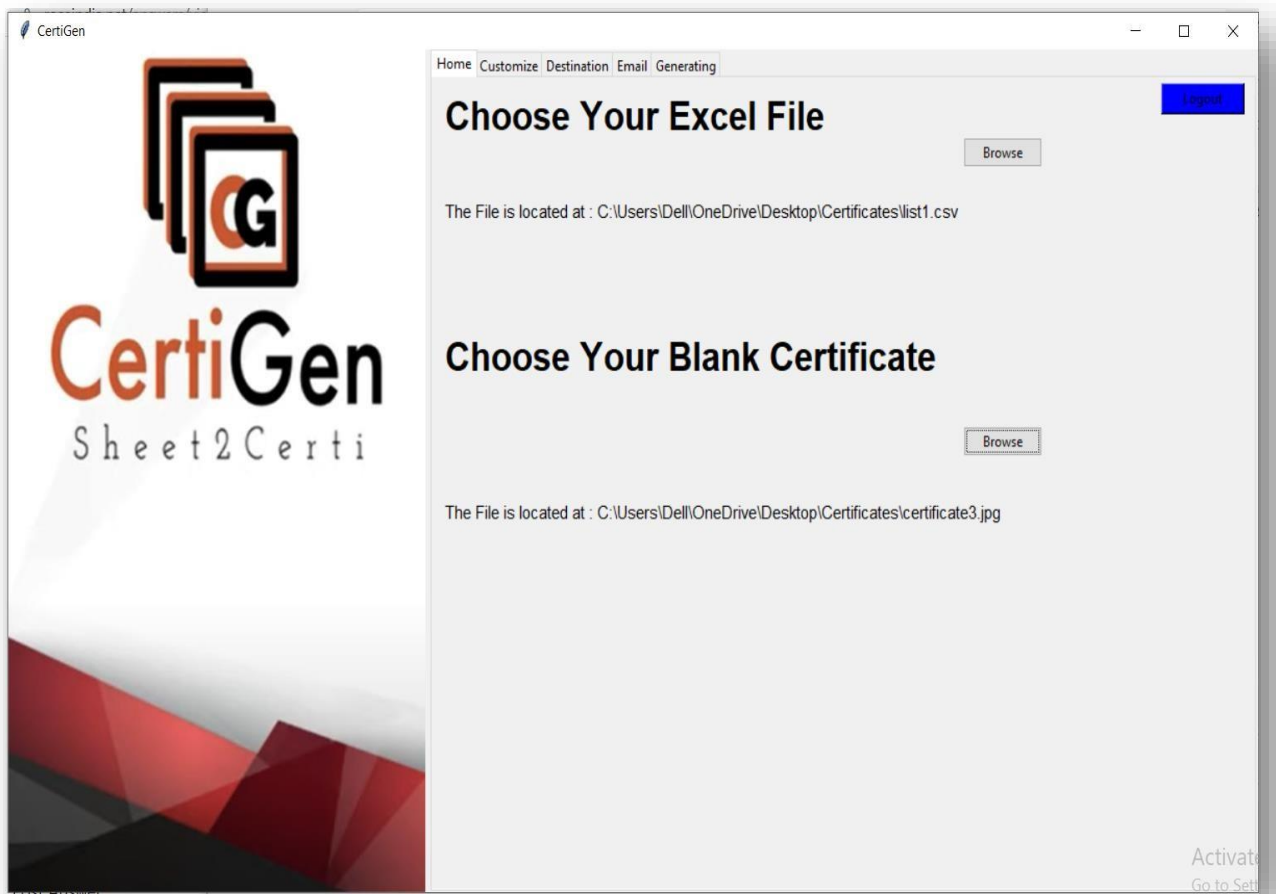


Figure2.4 outcome of the admin dashboard for the home tab

```
In [15]: import pandas as pd
import numpy as np
import PIL
import pip
import smtplib

In [17]: from PIL import Image, ImageDraw, ImageFont
import pandas as pd
import os
import smtplib
from email.message import EmailMessage
df = pd.read_csv('C:\\Users\\sriva\\Desktop\\Work\\list.csv')
font = ImageFont.truetype('arial.ttf', 22)
for index, j in df.iterrows():
    img = Image.open('C:\\Users\\sriva\\Desktop\\Work\\certificate1.jpg')
    draw = ImageDraw.Draw(img)
    draw.text(xy=(200,190), text='{}'.format(j['Name']), fill=(0,160,260), font=font)
    draw.text(xy=(230,265), text='{}'.format(j['Role']), fill=(0,0,320), font=font)
    draw.text(xy=(100,320), text='{}'.format(j['Signature']), fill=(102,0,51), font=font)
    draw.text(xy=(400,320), text='{}'.format(j['Date']), fill=(0,137,209), font=font)
    draw.text(xy=(470,400), text='{}'.format(j['Serial No.']), fill=(0,120,100), font=font)
    img.save('C:\\Users\\sriva\\Desktop\\Work\\sem/{}.jpg'.format(j['Name'], j['Role'], j['Signature'], j['Date'], j['Serial No.']))
    names = df['Name']
    emails = df['Mail']
    for i in range(len(emails)):
        name = names[i]
        email = emails[i]
        s = smtplib.SMTP('smtp.gmail.com', 587)
        s.starttls()
        s.login("techcse04@gmail.com", "csetech2021")
        s.sendmail("techcse04@gmail.com", email, 'subject : Mailing Certificate \n\n Dear candidate \n\n Your certificate is a')
        s.quit()
    print("Mail Sent successfully")

Mail Sent successfully
Mail Sent successfully
Mail Sent successfully
Mail Sent successfully
```

Figure 3: code for the mailing certificates to the participants

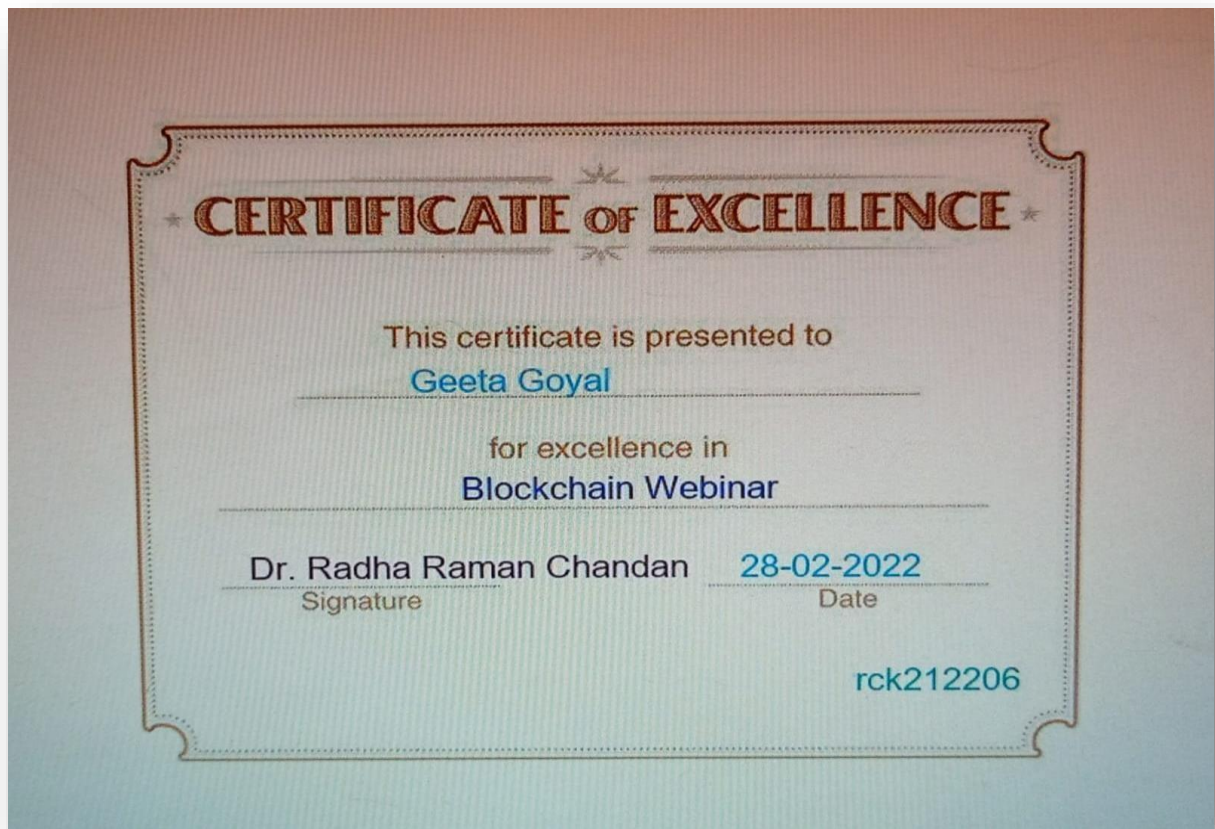


Figure3.1 this is the certificate template format

CHAPTER 5

RESULT AND DISCUSSION

System Test The following two levels of testing were conducted to ensure the proposed system is working perfectly and achieved the project objectives.

I. **Unit test:** at this level of test each module is tested independently to ensure is working.

II. **System test:** at this level, the entire system is tested to ensure all the system modules are integrated and are working perfectly.

The implementation phase of our project is still in progress. Thus far, we have successfully completed the implementation and work of the following factors:

- Login System where the admin can log in to the system.
- Admin Dashboard where admin can access the system.
- Backend of the certificate generation where certificates are generated.

The system was tested in generating and verifying the students' certificates of universities. The CSV file that contains the information of students. at this level, the entire system is tested to ensure all the system modules are working perfectly.

CHAPTER 6

COMPARISION

In this we try to compare our work with other one. As in SrushtiA. Shimpi [1] making a Certificate Generation System on that we seen that, this research is mainly based on database technology & the credit-based grading system. Which are not very user-friendly Interface. Because in this calculation of grades & credits for each course it is a very big task to perform. The outcome of signature Process is digital signature. It provide authenticity of certificates.

But in our project not the all these things. Our project is basically based on the GUI application which is more user- friendly Interface as we know. In our we are making Automatic Certificate Generation System & verify these certificates by providing ref. Id on it for the authenticity. And there is no limit generation of certificate(s). After the generation of certificates, we try to send all the certificates to the participant through the E-mail by clicking one button.

CHAPTER 7

CONCLUSION

The main purpose of this project is to develop a system that can be used in many institutions for generating and verifying the certificate. This project explored a new technique to address the issues of certificate generation system that uses predefined template and also, we try to resolve the problem of sending one certificate at only one time. Instead, we tried to send all the certificate to candidate through E-mail id at once. Dejan [4] work that requires an end-user to have a minor knowledge of XML to be able to define certificate template and certificate template format. but in this it is not required.

REFERENCES

URL: -

- 1.<https://dev.to/bhargavjoshi/automatic-certificate-generator>
- 2.<https://github.com/tanmaypardeshi/Certificate-Generator>
- 3.<https://pythonrepo.com/repo/GDSC-IIIT-Kalyani-WOCcertificate-generator-Hactoberfest2021-python-generalutilities>
- 4.https://www.researchgate.net/publication/225212534_ASGR_T_-_Automated_Report_Generation_System

Books And Research Paper: -

- 1. Srushti, A. S. (2014), Certificate GenerationSystem.
- 2.Abraham,Nwachukwu,K.C.(2015),Designing An Automatic Web-Based Certificate Verification System ForInstitutions
- 3. Ahmed Dalhatu Yusuf, Moussa Mahamat Boukar, Shahriar Shamiluulu, “Automated Batch Certificate Generation and Verification System” in IEEE, November2017.
- 4. Dejan, G. (2011), Applicative Solution for Generating Report Template – Automated Report GenerationSystem.
- 5. Dharani, R. P. (1998), Customized Document Generation System.
- 6. Kothari, C. R. (2004), Research Methodology Methods and Techniques.
- 7. Bharti Chikankar, Sidhant Jaiswal, “Certificate Generation System” in Engineering, Science and Management,August-2020