 A Project Report

On

**Automating Emails**

**Submitted by**

Abhishek Singh,2115000036

Khushi Singh,2115000536

Rishabh Saxena,2115000844

Siddhant Sahay,2115000992

Tanishka Sharma,2115001032

**Supervisor**

Dr. Pratik Roy

Assistant Professor,Dept. Of CEA

GLA UNIVersity

**Department of CEA**

**GLA University,Mathura**



**GLA University, Mathura - 281406**

Date:



**DECLARATION**

We Abhishek Singh,2115000036 ; Khushi Singh,2115000536 ; Rishabh Saxena,2115000844 ; Siddhant Sahay,2115000992 ; Tanishka Sharma,2115001032 hereby declare that the work presented in this project report entitled “AUTOMATING EMAILS” is an authentic record of our own work carried out under supervision of Dr. Pratik Roy

Name, Roll No. & Signature of Student 1 Name, Roll No. & Signature of Student 2

Name, Roll No. & Signature of Student 3 Name, Roll No. & Signature of Student 4

Name,Roll No. & Signature of Student



**CERTIFICATE**

This is to certify the project work entitled "**AUTOMATING EMAILS**" is carried out by ABHISHEK SINGH, KHUSHI SINGH, RISHABH SAXENA, SIDDHANT SAHAY, TANISHKA SHARMA the bonafide student of "**G.L.A UNIVERSITY ,MATHURA**". They have completed their project for partial fulfillment of the award bachelor degree in "**Computer Science Engineering**" during the 2021-2022. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for said degree.

Date: ……………………………………..

Place: Mathura

Name and Signature with Affiliation of Supervisor

(Dr. Pratik Roy)



**Contents:**

|  |  |  |
| --- | --- | --- |
| **Certificate &** | | **ii**  **iiiI** |
| **Declaration** |
| **Table of Contents** | | **iv** |
| 1. **Introduction, Motivation and Objective** | | **v-vi** |
| 1. **Project Description and Work done** | | **vii-xv** |
| 1. **Geotagged Images of Students at the place of work** | | **xvi** |
| 1. **Findings and Conclusion** | | **xvii** |
| **Bibliography/ References** | | **xviii** |



**Introduction, Motivation**

**and Objective**

**INTRODUCTION:**

As we can imagine, Python is the ideal companion to send any kind of Email automatically with or without our supervision.

The problem is that, although there is a simple package to handle all this, the guides we find on the web are often fragmented and incomplete.

I want to collect and show you once and for all the easiest methods to send Emails automatically with or without attachments .

While commercial email clients offer the convenience of a user-friendly GUI, they often lack the flexibility and customizability that many developers or content creators desire for their emailing needs.

Be it for sending a thank you email to a new subscriber on your social media channel or adding email functionality to your latest project, the ability to send automated messages using SMTP in Python is something that is bound to come in handy .



****

**MOTIVATION:**

Email is used for many different purposes, including contacting friends, communicating with professors and supervisors, requesting information, and applying for jobs, internships, and scholarships.

Depending on your purposes, the messages you send will differ in their formality, intended audience, and desired.

Business communication can be done easily without any security issues, and it is faster than any other forms of communication. So we have read and listen about emails a lot and we discuss that

can we send a email automatically and we decided to do this as our project .

And we choose python as our language to do this project because python is easy language . We can write a code in python easily .

****

****

**Objective:**

The objective of our project is how to send automated emails messages which involve delivering text messages, essential photos, and important files, among other things in Python.

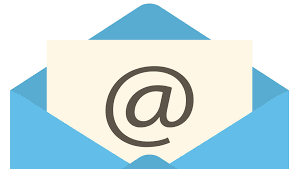
We are using two libraries for this purpose.These are email and smtplib, as well as Glob Module.

****

**Description and Work done **

**Email Module:**

The email package is a library for managing email messages. It is specifically not designed to do any sending of email messages to SMTP , NNTP, or other servers; those are functions of modules such as smtplib and nntplib .

The overall structure of the email package can 

be divided into three major components, plus a fourth component that controls the behaviour

of the other components.

**I.Central Component:**

The central component of the package is an “object model” that represents email messages.

The email object model is a tree structure of objects that all provide the Email Message API.

Email message provides the core functionality for setting and querying header fields, for

accessing message bodies, and for creating or modifying structured messages.

**II. Parser and Generator:**

The other two major components of the package are:

i)the parser , and

ii)the generator

**Parser:** The parser takes the serialized version of an email message (a stream of bytes) and converts it into a tree of Email Message objects.

**Generator:** The generator takes an Email Message and turns it back into a serialized byte stream.

(The parser and generator also handle streams of text characters, but this usage

is discouraged as it is too easy to end up with messages that are not valid in one

way or another.

**III. Control Component:**

The control component is the policy module. Every Email Message, every generator, and every parser has an associated policy object that controls its behaviour.

**Smtp MODule:**

Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers.

Python provides smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

Here is a simple syntax to create one SMTP object, which can later be used to send an e-mail −

import smtplib 

smtpObj = smtplib.SMTP( [host [, port [, local\_hostname]]] )

Here is the detail of the parameters −

• host − This is the host running your SMTP server. You can specify IP address of the host or a domain name like tutorialspoint.com. This is optional argument.

• port − If you are providing host argument, then you need to specify a

port, where SMTP server is listening. Usually this port would be 25. 

• local\_hostname − If your SMTP server is running on your local machine,

then you can specify just localhost as of this option.

An SMTP object has an instance method called sendmail, which is typically used to

do the work of mailing a message. It takes three parameters:

• The sender − A string with the address of the sender.

• The receivers − A list of strings, one for each recipient.

• The message − A message as a string formatted as specified in the various RFCs.

**Glob module:**

Glob is a general term used to define techniques to match specified patterns according to

rules related to Unix shell. Linux and Unix systems and shells also support glob and also provide function glob() in system libraries.

In Python, the glob module is used to retrieve **files/pathnames** matching a specified pattern. The pattern rules of glob follow standard Unix path expansion rules. It is also predicted that according to benchmarks it is faster than other methods to match pathnames in directories. With glob, we can also use wildcards apart from exact string search to make path retrieval more simple and convenient.

This module come built-in with python , so there is no need to install it externally.



**Step-by-step Implementation:**

Step 1: Import the following modules:

**Code:**

#IMPORT THE FOLLOWING MODULES

#IMPORT smtplib MODULE

import smtplib

#IMPORT EmailMessage MODULE

from email.message import EmailMessage

#IMPORT glob MODULE

import glob

Step 2: Write your message you want to Email.

I)Declare the variable ‘msg’ and assign EmailMessage Module .

**Code:**

#Write Your message you want to email

msg=EmailMessage()

II) Write subject of the Email.

Code:

#Subject of the Email

msg['Subject']="Birthday Invitiation"

III) Write name of the Sender.

Code:

#Sender of the Email

msg['From']="SQUAD Family"

IV) Enter the email of users to whom you want to send your email. #Person's Email Username you want to send email

msg['To']="siddhant.sahay\_cs21@gla.ac.in","rishabhsaxena577@gmail.com","abhijaat763@gmail.com","khushus7525@gmail.com","tanishkasharma225@gmail.com"

V) Enter the message you want to send in your email.

Code:

#Message to display on the email

msg.set\_content("Greetings From SQuad Family")

VI) Add the text file in your project if you want to add.

Code:

#Add the Text file to the email

with open("fg.txt") as myfile:

#READ THE FILE DATA IN TEXT FILE

data=myfile.read()

#CONNECTION TO THE EMAIL DATA

msg.set\_content(data)

VII) Add the attachment file to your email.

Code:

#Add the Attachment file to the Email

for files in glob.glob("\*.docx"):

with open(files,"rb") as f:

#READ THE FILE DATA IN ATTACHMENT DOCUMENT

file\_data=f.read()

#NAME OF THE FILE

file\_name=f.name

#ADD ATTACHMENT TO THE EMAIL

msg.add\_attachment(file\_data,maintype="application",subtype="docx",filename=file\_name)

VIII) Display the message if the email attachment is done.

Code:

#Display the message if the attachment is done

print("Atttachment Done!!!")

Step 3: Establish connection to the server of smtp.

Code:

#Connection to the server of smtp

#SERVER NAME

server= smtplib.SMTP\_SSL('smtp.gmail.com',465)

#LOGIN IN THE SERVER

server.login("sahaysidd1503@gmail.com","siddhu@1503")

#SEND THE MESSAGE

server.send\_message(msg)

server.quit()

Step 4: Display the message if your email is send to above email usernames.

Code:

#Diplay the message if email is sent

print("Email Sent!!!!")

****

**SOURCE CODE:**

#IMPORT THE FOLLOWING MODULES

#IMPORT smtplib MODULE

import smtplib

#IMPORT EmailMessage MODULE

from email.message import EmailMessage

#IMPORT glob MODULE

import glob

#Write Your message you want to email

msg=EmailMessage()

#Subject of the Email

msg['Subject']="Birthday Invitiation"

#Sender of the Email

msg['From']="SQUAD Family"

#Person's Email Username you want to send email

msg['To']="siddhant.sahay\_cs21@gla.ac.in","rishabhsaxena577@gmail.com","abhijaat763@gmail.com","khushus7525@gmail.com","tanishkasharma225@gmail.com"

#Message to display on the email

msg.set\_content("Greetings From SQUAD FAMILY")

#Add the Text file to the email

with open("fg.txt") as myfile:

#READ THE FILE DATA IN TEXT FILE

data=myfile.read()

#CONNECTION TO THE EMAIL DATA

msg.set\_content(data)

#Add the Attachment file to the Email

for files in glob.glob("\*.docx"):

with open(files,"rb") as f:

#READ THE FILE DATA IN ATTACHMENT DOCUMENT

file\_data=f.read()

#NAME OF THE FILE

file\_name=f.name

#ADD ATTACHMENT TO THE EMAIL

msg.add\_attachment(file\_data,maintype="application",subtype="docx",filename=file\_name)

#Display the message if the attachment is done 

print("Atttachment Done!!!")

#Connection to the server of smtp

#SERVER NAME

server= smtplib.SMTP\_SSL('smtp.gmail.com',465)

#LOGIN IN THE SERVER

server.login("sahaysidd1503@gmail.com","siddhu@1503")

#SEND THE MESSAGE

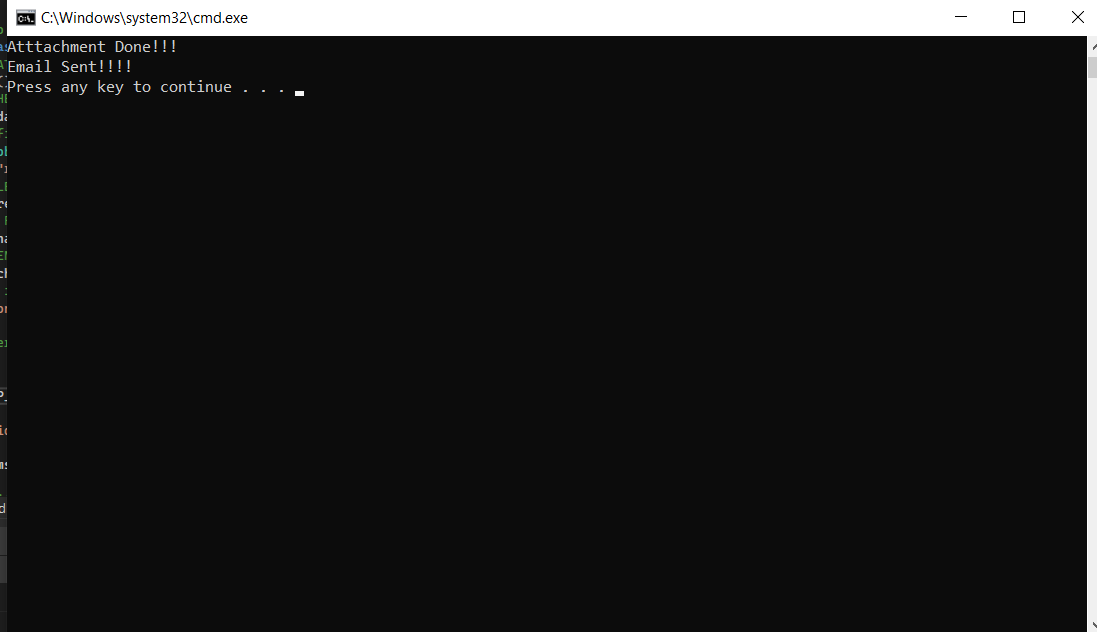
server.send\_message(msg)

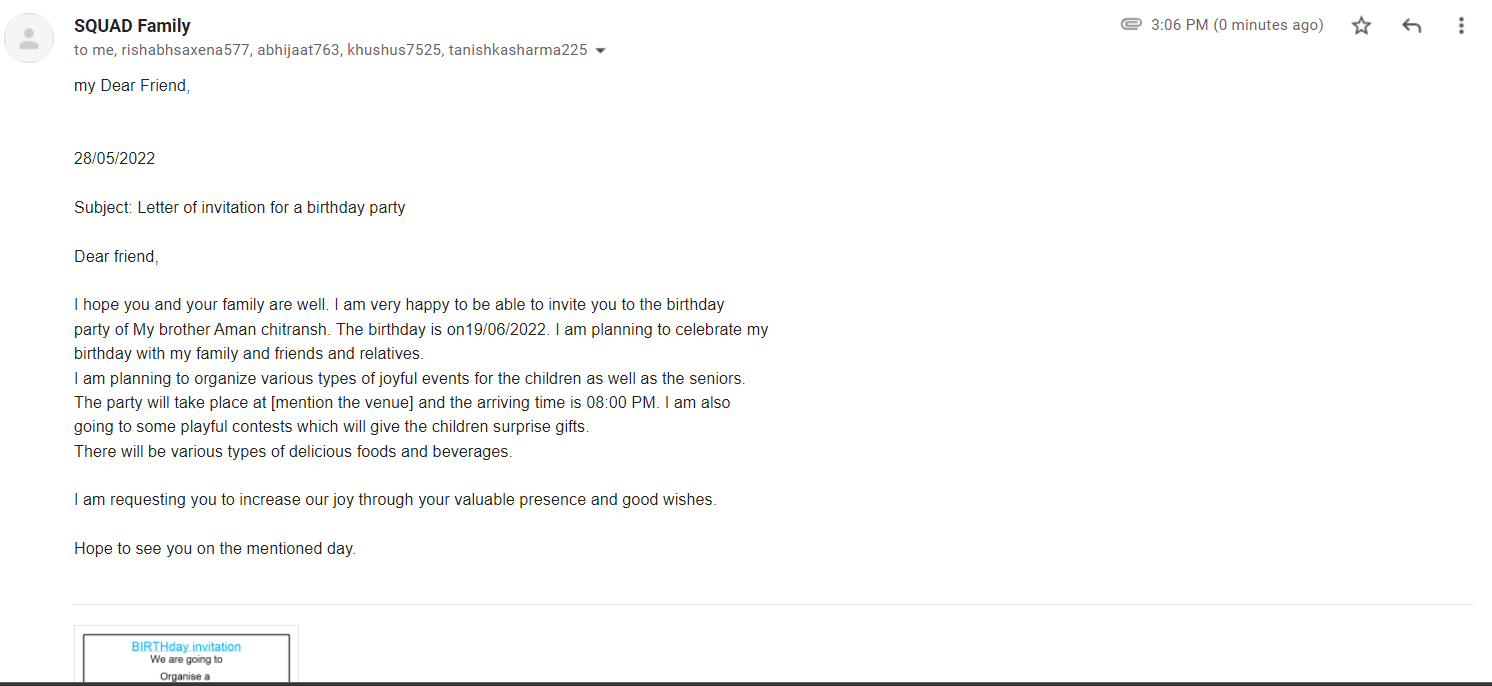
server.quit()

#Diplay the message if email is sent

print("Email Sent!!!!")

**Output:**

****

****

**Geotagged Images of Students at the**

**place of work**

**  **

** **

**Findings and Conclusion**

It was a wonderful and learning experience for us while working on this

project. This project took us through the various phases of project

development and gave me real insight into the world of automating emails.

The joy of work and the thrill involved while tackling various problems

and challenges gave me a feel of developer industry.

I enjoyed each and every bit of work I had put into this project.



**Bibliography/ References**

* [**www.geeksforgeeks.org**](http://www.geeksforgeeks.org)
* [**www.tutorialspoint.com**](http://www.tutorialspoint.com)
* [**https://www.salesmate.io/blog/email-automation/**](https://www.salesmate.io/blog/email-automation/)
* [**https://docs.python.org/3/library/smtplib.html**](https://docs.python.org/3/library/smtplib.html)