Journal of Applied Information Science
10 (2) 2022, 26-30
http://www.publishingindia.com/jais

Dynamic E-Certificate Designing with Automatic Mailing System using Python and SQLite3

Abhay Verma[1*] and Bharat Sharma[2]
[1]Department of Computer Science & Engineering, Vivekananda Institute of Technology, Jaipur, Rajasthan, India.
Email: verma.abhay2420@gmail.com
[2]Department of Computer Science & Engineering, Vivekananda Institute of Technology, Jaipur, Rajasthan, India.
Email: bhavyabharat10121999@gmail.com
*Corresponding Author

Abstract: In the scenario where things are getting online, including education, the number of online courses and even online degrees is increasing day-by-day. After the COVID-19 outbreak, most of the seminars, workshops, and trainings have switched over to the virtual platform, and the organizations are providing the virtual certificates, or e-certificates, which save a lot of effort for both the learner and the organization. Most organizations face the issue of designing different certificates for different requirements and events, which can sometimes be harder to design on demand. Currently, the system available is capable of inserting the name of the candidate into the pre-designed certificate. Also, certificates need to be designed first on some designing software, and then a script is used to insert names at the desired location. The aim of the project is to reduce the institute's overhead of designing certificates for various different events. Our project aims to solve the issue by providing a number of choices for the institute to generate certificates as per their needs. The built-in material for different kinds of certificates, which include certificates of participation, certificates of appreciation, certificates of attendance, etc., eases the institute's overhead for designing multiple certificates. The dynamic approach enables the certificate to be designed dynamically whenever needed. Furthermore, the certificates will be mailed to the respective candidates, providing feedback on every mail sent by the system. It involves very little human effort and is reliable due to the feature of automatic mailing and keeping a backup of original certificates.

Keywords: Automated E-Mailing, E-Certificate, Python, SQLite3.

I. Introduction

In the digital scenario where e-certificates are the demand of every institution, it becomes difficult to design and create certificates for a vast number of participants. Currently, the most prominent application available provides a way to enter the participant name and the certificate ID. The rest of the details are to be designed and pre-coded into the certificate template, including the authority names, designations, signatures, event names, etc. and what is the most time consuming thing is designing different certificates for different events. In fact, at least the content has to be changed every time for every new event, along with the authority signatures. The proposed application is a web-based application that provides independence to the user from time changing certificate template requirements. The web app provides an interface where the user manually enters a few details and the outcome is the final generated e-certificates of all the attendees/participants, irrespective of the number of participants. Further, the certificates are mailed to the participants via the e-mail service, which has been automated, and the system provides feedback of every email sent and a list of email ids where transmission fails.

II. Literature Review

In order to create a solution for educational institutes with the changing needs of e-certificates and design them for every occasion, we designed an application that allows users to:

- Select the type of certificate needed for the event, i.e., certificate of participation, certificate of appreciation, certificate of achievement, certificate of completion, etc.
- Pass an excel file that strictly follows the format specifications, which includes the name, email id, institute name, and rank of candidate.
- Insert the names, post and signature (in the form of an image file) of the authorities, which are limited to 3 members.
- Generate and store every e-certificate on the database in binary format, which can be retrieved at any time and on a local device as well.
- Automatically email the e-certificates to the candidates and provide feedback on every email sent, i.e., whether the email was successfully sent or not.

In this project, the web application is designed using React, which is an open-source JavaScript library. The React framework is built with the intention of being used for building cross-platform native apps [1]. We have used the SQLite3 module for the database support, which provides an SQL interface compliant with DB-API 2.0. SQLite3 is a light-weight, small-sized database engine that does not need to be separately installed as it is shipped with the Python programming language [2].

III. Libraries/Tools

Table I: Tools and Libraries Used in Project

Name of Libraries/ Tools	Description
PIL (Pillow)	Python Imaging Library (PIL) enables image processing capabilities to the Python interpreter.
Pandas	Python based data analysis and manipulation tool.
Smtplib	Python library to send emails using SMTP protocol.
Ssl	Python library that provides Transport Layer Security encryption and peer authentication.
React	JavaScript framework for building dynamic web applications.
SQLite3	Provides an SQL interface compliant with the DB-API 2.0.
CGI (Common Gateway Interface)	An interface between Web client and the web server that runs CGI script/ program.

IV. Implementation

Web Application

The web application is created using the React framework and it provides users an interface to select the type of certificate, event name, certificate number, authority name, post, and e-certificate template. There is a separate section where users can retrieve the existing e-certificates from the database.

Common Gateway Interface (CGI)

It is a standard way for a web browser to pass a user's request to an application programme and receive back the output of the application to the user [3]. It is a mechanism that is included in the Hypertext Transfer Protocol that allows web browsers to submit forms and connect programmes over a web server. A typical architecture of a common gateway interface is shown below [4].

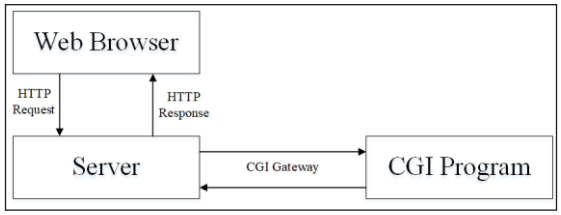

Fig. 1: Common Gateway Interface Architecture

We have used this technique in our application to link the backend scripts with the front end form and pass user inputs to the scripts, and receive errors (if any) generated in the backend. Also, the output of the script is fetched using the CGI mechanism and displayed on the application interface.

React

React is an open-source JavaScript library mostly used for building user interfaces for the web and mobile applications. It is efficient to update and render the components changed in the application by itself [1].

PIL (Pillow)

Python Imaging Library or Pillow is a Python library that adds image processing capabilities to the Python interpreter. It provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. The core image library is intended for fast access to data stored in basic pixel formats. It should provide a solid foundation for a general image processing tool [5].

This library is ideal for image archival and applications that need batch processing. Also, the library allows functionality that includes point operations, filtering with a set of built-in convolution kernels, colour space conversion, arbitrary affine transforms, global statistical analysis, etc [6]. It is another reason why we chose to use PIL over OpenCV.

Pandas

Pandas is a Python package that delivers fast, supple, and expressive data structures intended to make working with "*relational*" or "*labeled*" data both easy and intuitive [7].

Smtplib

The Smtplib module defines an SMTP client session object that can be used to send email to any machine online with an SMTP or ESMTP listener daemon [8]. It is a built-in module in Python, which enables users to not install it separately. These define the set of guidelines for the transmission of electronic

mail from one server to another over the internet [9]. This module enabled us to automate the mailing procedure just by providing the email ID and password of the organization's official email to the application. For security reasons, once the emails are sent, the application logs out from the email ID and forgets the password.

Ssl

This python-based module provides accessibility to Transport Layer Security (also called "Secure Sockets Layer") encryption and peer authentication ability for both client-side and server-side network sockets [10]. The module was an added security layer over the SMTP that enabled us to send the email securely with SSL/TLS encryption. Another major reason for using this module is that it uses the OpenSSL library, which is available on all modern Unix systems, Windows, MAC OS, and other platforms, as long as OpenSSL is installed on the platform [11].

Datetime

This python library provides numerous classes for dealing with dates and time in most simple and sophisticated way. It allows user to work with date and time of any zone. The different classes of the module provide several operative functions on date, time and time interval [12].

Re

Regular Expression (acronym of re) is the highly specialized language embedded within Python. Using this language, we specify the rules of possible strings to match in the application [13]. The matching set may contain English sentences, or e-mail addresses, or TeX commands, or any other string.

SQLite3

We have used the SQLite3 database as it can be operated over the local machine and does not require a separate installation and hosting [14]. It is easy to use, less complex, and best suited for small databases. It remains small in size even after storing image files in the database. SQLite is a C library that provides a frivolous disk-based database that doesn't need a separate server process and allows accessing the database using a non-standard variant of the SQL query language [15]. Some applications can use SQLite for internal data storage. It's also possible to exemplar an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

The complete application can be hosted over internet based hosting services, if there are multiple users in the same organization. Also, it can be hosted on the local device, including the database, if the need is not critical. This application supports all types of web browsers, and an active internet connection is required to access the application if hosted online via a hosting service or an intranet if hosted on a private network of the organization.

V. Flowchart

The web application begins (over the internet/ intranet or local device) with two options for the user—generate e-certificate or retrieve e-certificate. In the generate e-certificate module, users need to deliver a set of inputs to the application along with the image file of the certificate template and signatures to be inserted. Furthermore, the application checks whether the inputs are valid and follow the necessities or not. If any discrepancy is found, the application warns the user about the error and terminates. If no discrepancy is found, it will display the template with the necessary details inserted. Once the user checks and confirms the template, the application starts its processing, generating certificates, storing them locally, making a database entry, and sending them over e-mail. If the mail is not sent successfully, it gives the user the list of mail IDs to which mail is not sent.

The retrieve certificate module asks user for the few inputs which includes name & e-mail ID of candidate and event name. If the entry is found, the e-certificate is downloaded over the local device. Otherwise, an error is displayed.

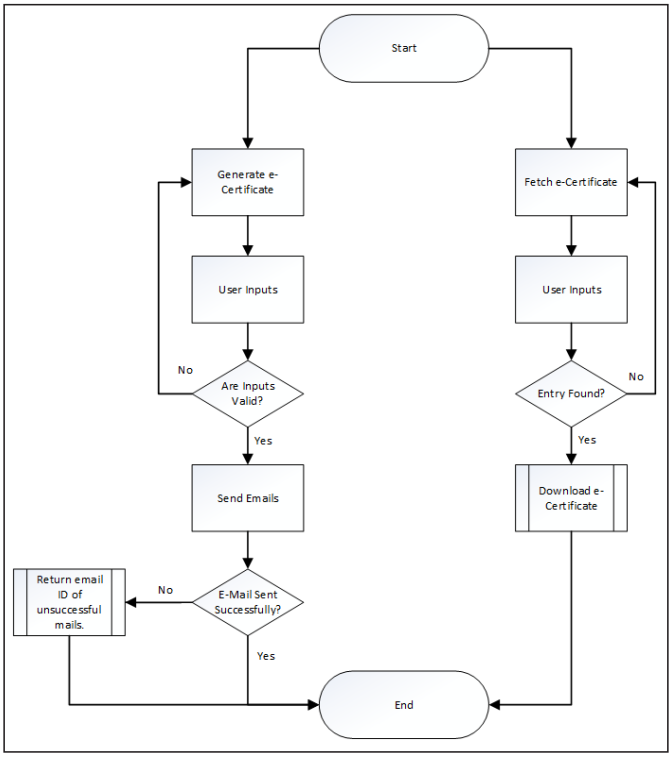

Fig. 2: Flowchart

VI. Intended Audience

This application has a wide range of targeting audiences, which are divided into primary and secondary audiences. Primary audience are the educational institutes, such as schools, colleges, universities, etc. where multiple events occur all over the year and require multiple types of certificates for the regular academics and on-demand certificates for special events. Secondary audience consists of all other public and private organizations that are dealing with public awareness, training, conducting online workshops, seminars, courses, etc. and requiring frequent design of e-certificates for different speakers and events.

VII. Quality Attributes

- *Usability:* The interface is user friendly and is very easy to use. There are direct input fields provided and an interactive input for file formats.
- *Reliability:* The application is highly reliable as it provides feedback at every stage of processing and prevents users from making mistakes. Even if some mistake is made, further processing is terminated along with an error message.
- *Maintainability:* The application does not require maintenance from the user side, which un-employs the user from the overhead task of updating and patching.
- *Adaptability:* The application is adaptable to all the operating systems, including Windows, Linux, and iOS.
- *Availability:* The application is available all the time, whether it is hosted in a cloud-based service or over the organization's network.
- *Performance Efficiency:* The application requires very few resources, which a mobile device or a normal computer system can easily fulfil.
- *Interoperability:* A centralised database allows interoperability of the application over the internet.

VIII. Performance Requirements

The application has some constraints that need to be carefully observed and the prescribed format and guidelines must be followed. Failing to achieve any of the performance requirements may lead to the failure of the application. The requirement that must be fulfilled for the proper working of the application includes:

- Only the allowed image file formats can be used, which includes JPEG and PNG for both certificate template and signature image files.
- It is strictly essential to use the prescribed excel file format.
- Require a valid email ID and password (including access from the mail server provider) from the organization for mailing the e-certificates to the participants.
- It requires an active internet connection to send e-mails.
- Only the administrator can download the e-certificate from the database.
- Applications cannot be installed over the smartphone. Only the cloud hosted version of the application can be accessed using a smartphone.

IX. Future Scope

This project aims to overcome the problem of designing new certificates for various different events by providing users with a facility to select the e-certificate of their choice and type. Currently, the application is limited to two types of e-certificates, i.e., certificate of participation and certificate of appreciation. We tend to add various other types of certificates. Other than this, we will be designing a drag-and-drop interface where the user can himself arrange the items dynamically, without needing extra software.

Finally, we will be working on encrypting the database for an added layer of security and creating an automatic backup schedule of the database- one on the local device and the other on the cloud server. Additionally, an administrator secured, encrypted logging database will be created, which will contain all the activity logs and only the user has permission to access them.

X. Result/Conclusion

We have developed a dynamic e-certificate designer and generator which can automatically send the e-certificates as email attachments, which tends to solve the institutes' problem of designing new e-certificates for different events and for different speakers. Since it is a web application, it can be hosted on internet servers and local devices both. The light weight application can run on both desktop and mobile devices that support web browsers. Only the cloud hosted version of the application can be accessed through a smartphone, but it can also be hosted on a desktop device locally. It is powered by the SQLite3 database and does not require separate installation, hardware, or extra resources to deal with.

References

[1] C. Gackenheimer, and A. Paul, *Introduction to React*, vol. 52, Apress, 2015.

[2] X. Liu, X. Fu, and G. Sun, "Recovery of deleted record for SQLite3 database," *2016 8th International Conference on Intelligent Human-Machine Systems and*

Cybernetics (IHMSC), 2016, pp. 183-187, doi: 10.1109/IHMSC.2016.173.

[3] G. Venkitachalam, and T. Chiueh, "High performance common gateway interface invocation," *Proceedings 1999 IEEE Workshop on Internet Applications (Cat. No.PR00197),* 1999, pp. 4-11, doi: 10.1109/WIAPP.1999.788011.

[4] J. Kolbitsch, "Fine-grained transclusions of multimedia documents in HTML," *Journal of Universal Computer Science*, vol. 11, no. 6, pp. 926-943, 2005.

[5] A. Clark, "Pillow (pil fork) documentation," *Readthedocs*, 2015. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf

[6] M. Driscoll, *Pillow: Image Processing with Python*. Independently Published, 2021.

[7] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. United States: O'Reilly Media, 2017.

[8] J. Goerzen, "Simple message transport protocol," in *Foundations of Python Network Programming*. Berkeley, CA: Apress, 2004. [Online]. Available: https://doi.org/10.1007/978-1-4302-0752-8_10

[9] M. F. Sarker, and S. Washington, *Learning Python Network Programming*. Packt Publishing Ltd., 2015.

[10] A. Bates, J. Pletcher, T. Nichols, B. Hollembaek, D. Tian, K. R. Butler, and A. Alkhelaifi, "Securing SSL certificate verification through dynamic linking," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, Nov. 2014, pp. 394-405.

[11] J. Viega, M. Messier, and P. Chandra, *Network Security with OpenSSL: Cryptography for Secure Communications*. O'Reilly Media, Inc., 2002.

[12] W. McGugan, "Exploring python," in *Beginning Game Development with Python and Pygame*: *From Novice to Professional*. Apress, 2007, pp. 19-40.

[13] T., Stubblebine, *Regular Expression Pocket Reference: Regular Expressions for Perl, Ruby, PHP, Python, C, Java and .NET.* O'Reilly Media, Inc., 2007.

[14] S. T. Bhosale, M. T. Patil, and M. P. Patil, "SQLite: Light database system," *Int. J. Comput. Sci. Mob. Comput*, vol. 44, no. 4, pp. 882-885, 2015.

[15] M. Owens, and G. Allen, *The Definitive Guide to SQLite*. Netherlands: Apress, 2010.