ISSN: 2584-0495 Vol. 2, Issue 5, pp. 866-870

International Journal of Microsystems and IoT

ISSN: (Online) Journal homepage: https://www.ijmit.org

Development of an e-platform for Result Generation and auto-e-mailing

Vaishali Dhare, Shakshi Likhia

Cite as Dhare, V., & Likhia, S. (2024). Development of an e-platform for Result Generation and auto-e-mailing. International Journal of Microsystems and IoT, 2(5), 866-870. https://doi.org/10.5281/zenodo.13132377

Published online: 15 May 2024.

Submit your article to this journal:

Article views:

View related articles:

View Crossmark data:

DOI: https://doi.org/10.5281/zenodo.13132377

International Journal of Microsystems and IoT
Vol. 2, Issue 5, pp. 866-870; DOI: https://doi.org/10.5281/zenodo.13132377

Development of an e-platform for Result Generation and auto-e-mailing

Vaishali Dhare, Shakshi Likhia

Department of Electronics and Communication Engineering, Institute of Technology Nirma University, Ahmedabad, India

ABSTRACT
Automation minimizes the efforts in terms of time and manpower costs. An automatic marksheet generation platform must be available to generate the final marksheet from course-wise component entries in the Universities. In this paper, the e-platform for automatic marksheet generation using Python is developed. This platform provides a Graphical User Interface to the administrator and students. This platform also sends the final marksheet to the respective students. So, a complete marksheet generation e-platform is developed in this paper. This e-platform can be available as an open source so that anyone can use it freely.

KEYWORDS
auto-e-mailing, e-platform, marksheet, Python, SMTP

1. INTRODUCTION

Nowadays Universities are conducting many exams like assignments, regular class tests and quizzes, etc. Manual generation of results considering all these components is time-consuming. Also, errors cannot be neglected in these manual calculations. Generating results is a crucial task in every University, involving the meticulous retrieval and calculation of marks according to the specified syllabus. Every one of them has a different approach to keeping track of student profiles and evaluations, frequently using record books to capture student data. However, this method has drawbacks, including prolonged search times for student profiles, the risk of misplacing record books, and chances of entering wrong information from large sets of data. Additionally, considering the complexities involved in processing massive volumes of data, the manual method might not be able to provide the accuracy needed for accurate student assessments and rankings. As a result, the requirement for increased precision highlights the importance of putting in place an automated system that may lighten the load on universities and related staff members while also hastening the student distribution of results.

Commercial software is available for automatic marksheet generation. In this paper, an e-platform is developed using Python that generates the marksheet automatically from a database. This platform not only generates the marksheet but also sends the marksheet to the registered mail id. Also, this platform can be made open-source.

In [1], the website is developed that enables to upload of each subject's results, and then their marks sheet will be automatically generated. The marksheet generation is presented in [2]. It only generates the marksheet. Auto-mailing option is not available in [2]. In [3], the auto marksheet generation for school students is presented. The auto-generated document platform for different applications is presented in [4-6]. Some online marksheet generator tools are available in [7,8] but the personal University database is to be shared with them. Al So, these platforms do not handle huge databases and auto-mailing is absent from these platforms. Microsoft Excel [9] is the best available software for calculation but it is commercial and requires programming.

Therefore, we have developed the GUI-based marksheet generation using Python. This e-platform generates the marksheet from the student's database supplied through the ".csv" file at any time for any semester. This module simplifies reading and writing ".csv" files in Python, making it easy to work with tabular data.

It also calculates the cumulative grades as and when required. University administrators can generate the results and send them to the concerned students to their respective registered email IDs through the SMTP server [10].

Following are the objectives,

- Auto generation of marksheet
- GUI based development
- Auto e-mail sending

The paper is organized as follows: Section II describes the background of the backend. The implementation of the proposed e-platform is presented in Section III.

The results are discussed in Section IV. The paper is concluded in Section V.

2. BACKGROUND

Nowadays Python has gained popularity in almost every domain [11]. It is an interpreter-based, object-oriented, high-level programming language. Python offers a variety of libraries that can be used for programming and GUI-based development with ease. The library is a pre-written code modules that provide classes, utilities, and reusable functions for particular applications [12-15].

In this paper, the following Python libraries are used:

1. Pandas: This library is useful for file reading and writing, and file Input/Output purposes.
2. Tkinter: This is the most useful library for making the backend things as the front end. This enables the programmer to create an interactive Graphical User Interface (GUI) of a developed backend program.
3. Email.mime: It **is** used for creating and parsing email messages according to the Multipurpose Internet Mail Extensions (MIME) standard, including creating and formatting email messages, adding attachments, specifying recipients, and setting message headers.
4. Smtplib: This is a Python library that provides a way to send emails using the Simple Mail Transfer Protocol (SMTP).

3. IMPLEMENTATION

The proposed e-platform consists of two parts. The first part is used for fetching the student data from a database available in a ".csv" file for various grade calculations. It could be exam-wise, semester-wise, or cumulative year-wise. We have provided GUI to see the grades and report generation.

In the second part, the generated marksheet in the specific format is automatically sent to the respective student. The e-platform automatically sends an email notification to the respective student, offering a convenient and efficient means of delivering academic updates.

Algorithm 1 shows the pseudo-code for implementation of e-platform.

```
Algorithm 1 e-platform (database)
import libraries
read data.csv
fetch data ( )
Set up GUI layout by creating a tkinter window, entry fields for roll number, name, and grades, and, labels for subject codes, credits, and quality points.
enter roll number
display ( ) function
return (CGPI)
for (i=0; i<10; i++)
send_email( )
Enter the text that is to be sent with the attachments using the MIMEMultipart function to define parts of an email.
Assign a subject and auto-generated 16-digit password to the variable.
Create buttons for fetching data, displaying results, and sending emails.
Use SMTP to send email with provided credentials.
Handle exceptions if the email fails to send.
Run Tkinter main event loop
Success
```

Firstly, libraries like **Tkinter** for GUI, **smtplib** for sending emails, **email. mime** for constructing email messages and, **CSV** for handling CSV files is imported.

Then set the title and geometry for the new tkinter window created. Entry Widgets (**e1 to e8**) are generated which are used for user input. Each widget represents an input field for a specific data item (roll number, name, and grades). Organize these entry fields in the window using a grid layout. Each widget is placed in a specific row and column of the GUI window using this **grid()** method.

Function **read_csv_file** (csv_file) reads data from a CSV file and returns it as a dictionary. The CSV file contains all information of a student such as registration number/ID, semester, roll number, name, and grades. Then to retrieve that data we use the function **fetch_data()**. It is triggered when the user clicks the "Fetch Data" button. It retrieves the roll number entered by the user (**e1.get()**). If the roll number exists in the dataset (data), it fetches the corresponding name and grades. Furthermore, it updates the Entry widgets (e2 to e6) with the fetched data.

To compute the total quality points and SGPA (Standard Grade Point Average) for a set of subjects based on grades entered in a GUI we use the **display()** function. It initializes a global variable **total_sum** to track cumulative quality points. Within a loop, it retrieves grades for each subject, converts them to their respective grade values, and multiplies them by the subject's credit. Subject totals are displayed in the GUI, and their values are summed up to compute **total_sum**. Finally, it updates the GUI with the cumulative quality points and SGPA. This function effectively calculates and presents academic performance metrics within the GUI interface when the user clicks on the “Result” button.

Labels are used to display text (e.g., headings, subject codes, Sr No, etc). While buttons trigger actions like fetching data and displaying results. In this system, we link each button to its respective functionality (fetch data, display results, and send email). We give each of these buttons a specific color to be easily differentiable and create a good visual.

Calculations of SGPA:

Semester Grade Point Average (SGPA) is the summation of the product of Credit Points and Grade Points divided by the summation of Credits for all Courses. Here, we have used a universal grading table method as follows:

Table. 1 Grading Table

Grade	Marks	Grade points	Remarks
A+	90 and above	10	Outstanding
A	80 – 90	9	Excellent
B+	70 – 80	8	Ver Good
B	60 – 70	7	Good
C+	50 – 60	6	Fair
C	40 – 50	5	Pass
IF	Less than 40	0	Fail

Therefore, the SGPA is calculated using eq. 1.

$$\text{SGPA} = \frac{\sum CG}{\sum C} \qquad \text{(eq. 1.)}$$

Where C is the credit point and G is the grade point for that respective course.

This part of the system automates the calculation of marks and SGPA. Now let's look at the second part of automating emails to the respective student.

To send an email, we begin by importing essential libraries, such as the Simple Mail Transfer Protocol (**SMTP**) which provides a convenient way to send emails using Python. Mime is also one of its libraries used as a **MIMEText** class from **email.mime.text** and **MIMEMultipart** class from **email.mime.multipart** both to construct email messages with various content types, including plain text and multipart messages.

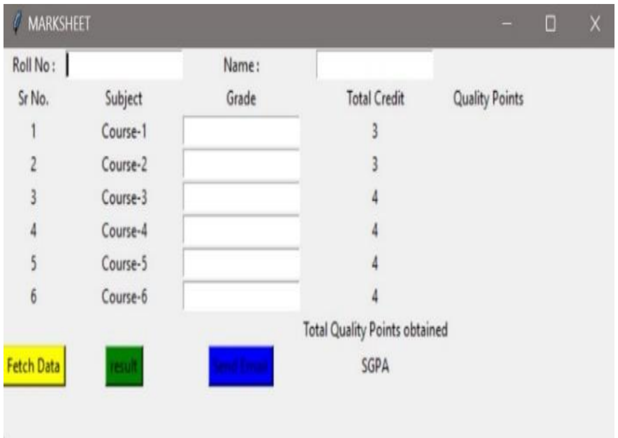
To establish a connection with the SMTP server, it's necessary to provide login credentials such as the SMTP port, SMTP server address, sender's email address, and an application-specific password for SMTP [17,18] email authentication. Here, we use **port 587** as it is the default port used for email submission to an SMTP (Simple Mail Transfer Protocol) server. It's specifically designated for email message submission by mail clients to mail servers.

Function **send_email()** is designed to send an email containing the mark sheet details of a student. Using the **MIMEMultipart()** object, it sets up the email message with the sender, recipient, and subject details, attaches the mark sheet content, and attempts to send the email using the Gmail SMTP server. Error handling is implemented to catch any exceptions that may occur during the email-sending process, printing an error message if the email cannot be sent. Finally, a tkinter button labeled "Send Email" is configured to trigger the **send_email()** function when clicked, allowing users to initiate the email-sending process from the GUI interface. The email is sent through the admin that is "shakshilikhia@gmail.com" in this system to all the students.

4. RESULTS AND DISCUSSION

The Python code is developed and runs using Visual Studio Code. The sample marksheet is generated by the platform step-by-step as shown in the resulting figures.

When we run this code, a GUI Marksheet pops up on our screen as shown in Fig. 1.



Roll No: Name:

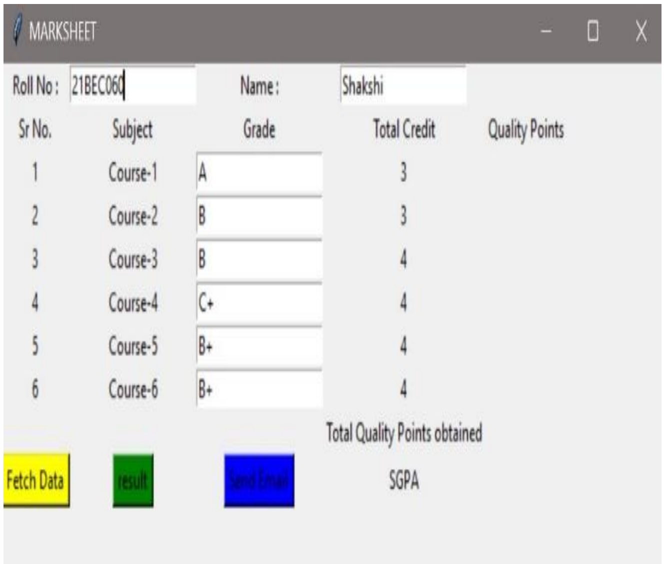
Sr No.	Subject	Grade	Total Credit	Quality Points
1	Course-1		3	
2	Course-2		3	
3	Course-3		4	
4	Course-4		4	
5	Course-5		4	
6	Course-6		4	

Total Quality Points obtained

SGPA

Fig. 1 Snapshot of GUI of a grade entry for a marksheet

It has been achieved with the use of Tkinter [19,20]. Then, to increase accuracy the code is developed in such a way that as the admin enters the roll number of a student, then by clicking on the "Fetch Data" button, this GUI interface fetches all other data of that respective student including their names and grades in all subjects as shown in Fig. 2. It automatically fills up the data from the database. And, in the case of entering the wrong roll number, the command prompt shows an error stating "No such roll number".



Roll No: 21BEC060 Name: Shakshi

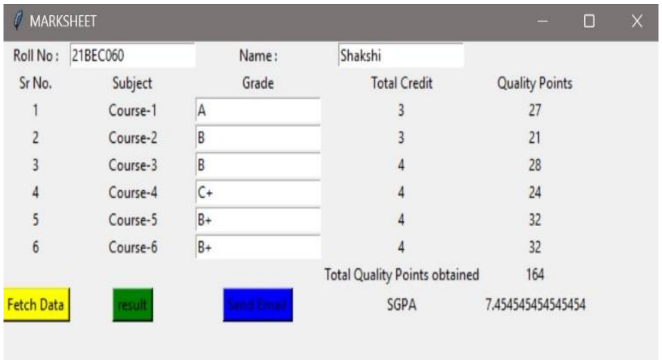
Sr No.	Subject	Grade	Total Credit	Quality Points
1	Course-1	A	3	
2	Course-2	B	3	
3	Course-3	B	4	
4	Course-4	C+	4	
5	Course-5	B+	4	
6	Course-6	B+	4	

Total Quality Points obtained

SGPA

Fig. 2 Fetching data button after the entered values

The interface has a button known as "Result". After the information is retrieved, we click on 'Result' which calculates the quality points and SGPA according to the **equation-1** and **Table-1** system, and then display the outcome as shown in Fig. 3.

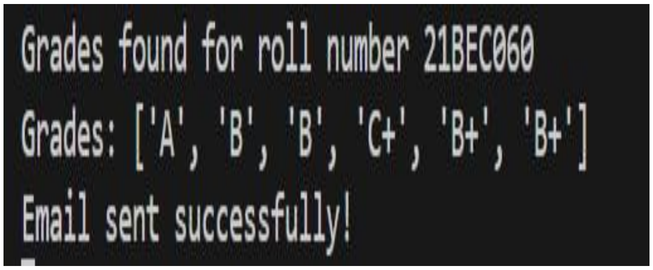


Roll No : 21BEC060 Name : Shakshi

Sr No.	Subject	Grade	Total Credit	Quality Points
1	Course-1	A	3	27
2	Course-2	B	3	21
3	Course-3	B	4	28
4	Course-4	C+	4	24
5	Course-5	B+	4	32
6	Course-6	B+	4	32
			Total Quality Points obtained	164
			SGPA	7.454545454545454

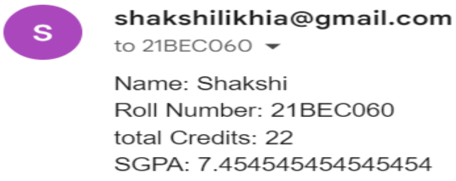
Fig. 3 Final result calculations

And, finally triggering the “Send Email” button after generating results, the mail is automatically sent to that student on their university mail with the subject of the mail as **MARKSHEET**. The command prompt shows an error if the email cannot be sent, or else confirms the mail as shown in Fig. 4. And, Fig. 5 displays the mail received by the student.



```
Grades found for roll number 21BEC060
Grades: ['A', 'B', 'B', 'C+', 'B+', 'B+']
Email sent successfully!
```

Fig. 4 Terminal-based visualization of a result



shakshilikhia@gmail.com
to 21BEC060

Name: Shakshi
Roll Number: 21BEC060
total Credits: 22
SGPA: 7.454545454545454

Fig. 5 Snapshot of received mail

5. CONCLUSION

In this paper, the e-platform for student's final result marksheet is developed using Python. This platform takes care of generating mark sheets automatically from the database. It is also automatically sending the generated marksheet to the concerned students. This also provides the GUI to the administrator and students. We have created a single platform to perform the marksheet generation and sending task. It can be made as an open source with the GUI.

Future scope: Security aspects are missing in this work. In future, the password protected options will be incorporated. Also, we will make it opensource with scalable version.

REFERENCES

1. Hussain, Sada, Saeed, Anwar & Jan, Saeed Ullah. (2023). Online Marks Sheet Generation System, Thesis of SSRN, 1-23. http://dx.doi.org/10.2139/ssrn.4368974
2. Shimpi, Srushti, Mandare, Sanket, Sonawane, Tyagraj, Trivedi, Aman & Reddy, K. T. V. (2014). PDF Marksheet Generator. In Proceedings of Int. J. Info. Tech. Comp. Sci., 11(1), 36-41. https://doi.org/10.5815/ijitcs.2014.11.05
3. Añulika, Ezenma A., Bala, Emmanuel & Nyap, Choji D. (2014). Design and Implementation of a Result Processing System for Public Secondary Schools in Nigeria. International Journal of Computer and Information Technology, 3(1), 1-7.
4. Naik, Kavya N., Patil, Arnica R., Patil, Kinnari N., Sankhe, Vaidehi R., More, Shraddha S. & Lobo, Vivian Brian. (2023). A Python-Based Grade Converter Application. Second International Conference on Electronics and Renewable Systems (ICEARS), 180-184. https://doi.org/10.1109/icears56392.2023.10084961
5. Verma, Abhay & Sharma, Bharat. (2022). Dynamic E-Certificate Designing with Automatic Mailing System Using Python and SQLite3. Journal of Applied Information Science, 10(2), 54-60. https://doi.org/10.13140/RG.2.2.17907.20000
6. Agrawal, A., Parikh, P., Patel, D. & Dhare, V. (2023). Development of a Software Platform for E-Certificate Generation and Auto-Sending. 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE), Ghaziabad, India, 503-506. https://doi.org/10.1109/aece59614.2023.10428231
7. Projects Geek. (2017). Mark Sheet Generation Automation Project. https://projectsgeek.com/2017/06/mark-sheet-generation-automation-project.html
8. Edusys. College Gradebook Marksheet Generator. https://www.edusys.co/en-in/college-gradebook-marksheet-generator.html
9. Microsoft. https://www.microsoft.com/
10. SMTP Gmail. https://smtp.gmail.com
11. Python Software Foundation. https://www.python.org/
12. Moore, Alan D. (2021). Python GUI Programming with Tkinter: Design and Build Functional and User-Friendly GUI Applications. Packt Publishing Ltd.
13. Zumstein, Felix. (2021). Python for Excel. O'Reilly Media, Inc.
14. Jaime. (2018). Python Automation Cookbook: Explore the World of Automation Using Python Recipes That Will Enhance Your Skills. Packt Publishing Ltd.
15. Podrzaj, Primoz. (2019). A Brief Demonstration of Some Python GUI Libraries. In Proceedings of the 8th International Conference on Informatics and Applications ICIA 2019, 1-6. https://doi.org/10.22214/ijraset.2023.48848
16. Khanji, S., Jabir, R., Ahmad, L., Alfandi, O. & Said, H. (2016). Evaluation of Linux SMTP Server Security Aspects: A Case Study. 7th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 252-257. https://doi.org/10.1109/IACS.2016.7476120
17. Sureswaran, R., Bazar, H. A., Abouabdalla, O., Manasrah, A. M. & El-Taj, H. (2009). Active E-mail System SMTP Protocol Monitoring Algorithm. 2nd IEEE International Conference on Broadband Network & Multimedia Technology, Beijing, China, 257-260. https://doi.org/10.1109/ICBNMT.2009.5348490
18. Kasyanov, S. & Fadeeva, K. (2023). Online Educational Community as a Platform for Online Teaching of Python Programming Language to Schoolchildren. 3rd International Conference on Technology Enhanced Learning in Higher Education (TELE), Lipetsk, Russian Federation, 246-250. https://doi.org/10.1109/TELE58910.2023.10184372
19. Stubblebine, T. (2007). Regular Expression Pocket Reference: Regular Expressions for Perl, Ruby, PHP, Python, C, Java and .NET. O'Reilly Media, Inc.

Authors

Vaishali Dhare is working as Assistant Professor in Electronics and Communication Engineering Department at Institute of Technology, Nirma University, Ahmedabad, India. She obtained her BE degree in Electronics and Telecommunication Engineering in 2000 from North Maharashtra University, Maharashtra and MTech degree in VLSI Design in 2010 from Nirma University, Ahmedabad.

She obtained her PhD in Testing of Nanotechnology Devices

and Circuits in 2018 from Nirma University. She has 20+ years of teaching experience. She has guided 30+ MTech and BTech projects and various conducted seminar and workshops. She has published more than thirty-five papers in journals and conference proceedings. Currently she is guiding two research scholars. She has undertaken consultancy and research projects. Her areas of interest include Hardware Description Languages, Testing and Verification of VLSI Design, System Verilog for Verification, Quantum-dot Cellular Automata: Nanotechnology, Applied Algorithms for VLSI CAD and Reversible Logic. She is a member of IEEE, IST and VLSI Society of India.

Email: vaishali.dhare@nirmauni.ac.in

Shakshi Likhia is a student at Nirma University in the Electronics and Communications Engineering Department. She is pursuing her BTech degree in Electronics and Communications Engineering, 2021-2025 batch from the Institute of Technology, Nirma University, Ahmedabad, India. Her areas of interest include the Internet of Things (IoT), digital VLSI design, embedded system design, Artificial Intelligence & Machine Learning algorithms. She has designed and implemented a range finder system using ultrasonic sensors and an Arduino microcontroller. She worked on a project to develop a fake news detection system using machine learning algorithms. The goal was to differentiate between reliable news articles and misleading or fabricated content.

Email: 21bec060@nirmauni.ac.in