

## Automated Email Sending Using Customized Features

1.Abhinay.S, 2.Aakash.J, 3.Aarthi.S, 4.Aashritha.S,  
5.Aashritha.D, 6.Aastha Kotriwar,  
7.Dr.D.Thiyagarajan 1,2,3,4,5,6 Student,7 Associate  
Professor

Artificial Intelligence & Machine Learning  
Department Of Computer Science And Engineering  
Malla Reddy University,Hyderabad,Telangana,India

### Abstract:

*An automated Email Sender is a Powerful tool that is used in new-generation marketing technology.*

*Python already has a module called Django, it creates a wrapper around the module. Email Automation is also been a strong platform and gives strong communication between sender and receiver. Python provides a mail-sending interface via the "smtplib" module. Django provides a couple of light wrappers over it. These wrappers are going to make sending emails extra quick and provide support for platforms that cannot use SMTP. The code major module is "django.core.mail" and we use a few functions and classes from that to send this email instead of sending one mail repeatedly. The mail is sent using the SMTP host and port or port numbers specified in the EMAIL\_HOST, and EMAIL\_PORT settings. The EMAIL\_PORT user and EMAIL\_HOST\_PASSWORD settings. And EMAIL\_USE\_TLS and EMAIL\_USE\_SSL settings control whether a secured connection is used.*

*In this project, we are using this email message class so all the class does is allow us to provide the subject line and the body of the email. The first thing we need to do is set up the email backend for example if we are using Gmail then smtp.gmail.com can also set up the port numbers and TLS. Email recipients and email host password to that email account we have imported Email messaging and settings so we can have access to all of those configurations that we made. We also use render to a string which is essentially going to grab our template and allow us to send it as a string so we can go into the email body. In this, we can include our "Subject", "Body", "Recipient" and their password so in the template we would write our message or body that links to the discord server while the receiver can view the message.*

**keywords:** Django;Automation;Autoresponder;Django templates;SMTP.

### INTRODUCTION

Automated email sending, also known as email automation, refers to the process of sending prewritten emails to a predetermined list of recipients at specific intervals or triggered by specific events. This approach to email marketing enables businesses to save time and

effort while providing personalized and timely communication to their customers.

Automated emails are typically created using specialized software or email marketing platforms that allow you to set up a series of email templates, configure rules for sending, and segment your email list based on various criteria such as demographics, behavior, or interests.

Examples of automated emails include welcome emails, abandoned cart reminders, order confirmations, product recommendations, and post-purchase follow-ups. By using automated emails, businesses can deliver relevant and targeted messages to their subscribers, improve customer engagement, and increase revenue. Automated emails can be a powerful tool for businesses of all sizes, enabling them to communicate more effectively with their customers and drive more sales. However, it's important to ensure that your email content is engaging, valuable, and relevant to your subscribers to avoid being marked as spam or causing them to unsubscribe.

### LITERATURE REVIEW

This literature review examines the topic of automated email sending, focusing on its applications, benefits, challenges, and best practices. The review explores various studies, articles, and research papers published on the subject, highlighting key findings, methodologies, and recommendations for effective implementation. The goal is to provide a comprehensive overview of the current state of automated email sending and its impact on communication efficiency and productivity. This literature review provides a comprehensive overview of the current knowledge and understanding of automated email sending. By examining various aspects such as applications, benefits, challenges, and best practices, it aims to assist practitioners and researchers in implementing effective email automation strategies and improving communication efficiency. Further research and exploration are encouraged to enhance our understanding of the evolving role of automated email sending in the digital age.

### PROBLEM STATEMENT

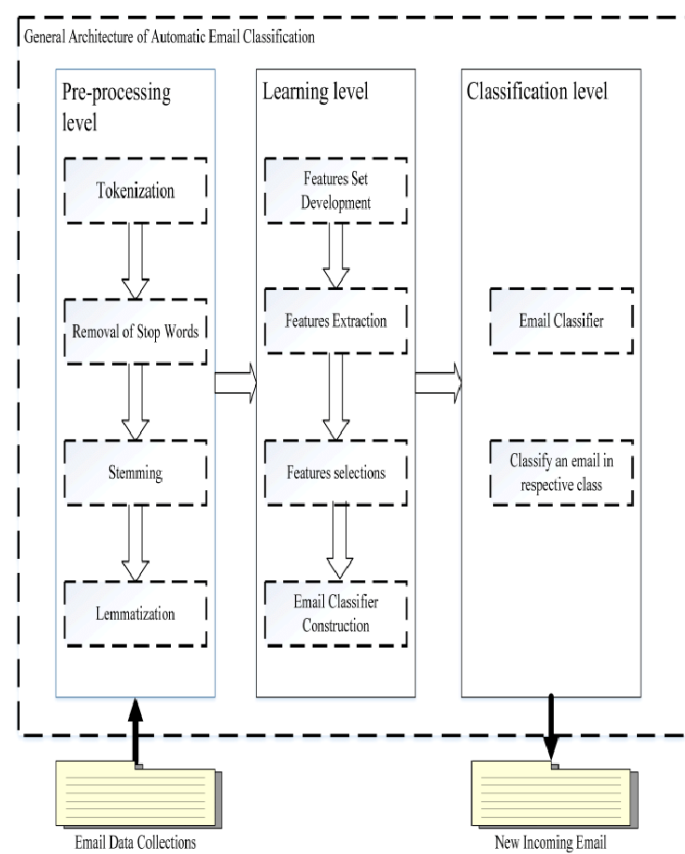
Automated email sending has become an integral part of communication strategies for organizations across

various industries. While it offers numerous benefits, such as time savings and improved customer engagement, there are several challenges and issues associated with its implementation. This problem statement aims to identify and address the key problems faced in the context of automated email sending.

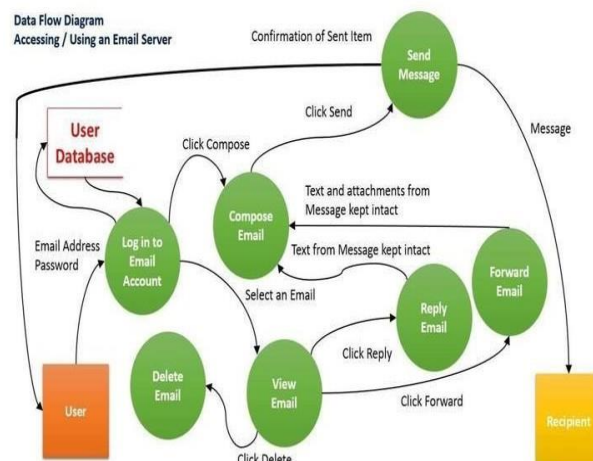
## METHODOLOGY

The methodology emphasizes goal setting, list segmentation, platform selection, content creation, workflow automation, testing, monitoring, and compliance. Continuous optimization and adaptation based on data analysis and recipient feedback are vital to maximizing the benefits of automated email sending.

## ARCHITECTURE OF AUTOMATED EMAIL SENDING



Email automation is the process of creating and sending emails to a predetermined list of recipients, based on specific triggers or conditions. It is an effective way to engage with your audience and provide them with relevant content or offers at the right time. Here is an introduction to designing an email automation system.



## EXPERIMENTAL RESULTS

### INPUT:

#### Contact

Name: Aashritha Dasari

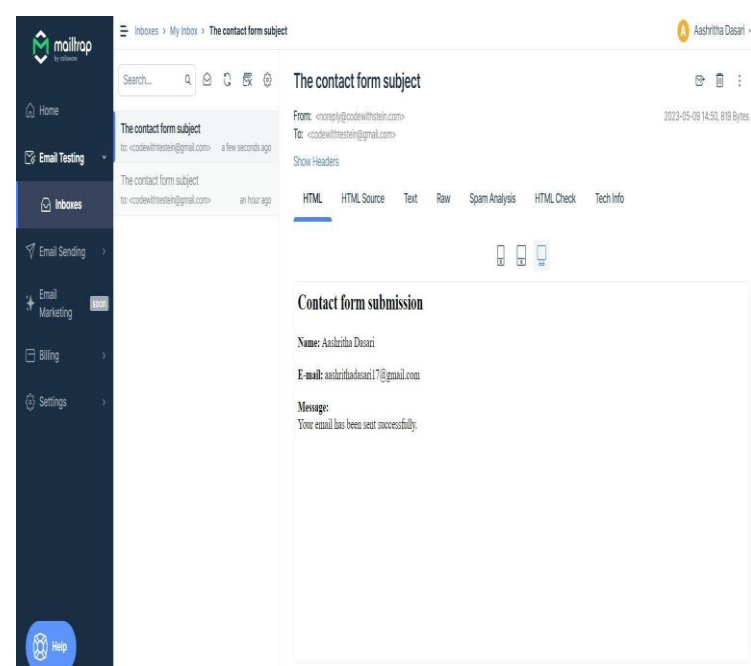
Email: aashrithadasari17@gmail.com

your email has been sent.

Content:

Submit

### OUTPUT:



## CONCLUSION

In this project a web application was built using the Django web framework and Python programming language. The application automates the process of sending emails with customized features based on business requirements. The application allows the user to create a list of addresses and their respective names. The application uses Django's built-in email sending functionality to send the emails. The application includes a contact form that allows users to submit their contact information and a message. When a user submits the form, an email is sent to the specified email address with the message and contact information. Overall, this project provides a useful tool for businesses to automate the process of sending customized emails to their target audience. It simplifies the process of sending emails and makes it more efficient and effective.

## FUTURE WORK

There are several possible future scope and improvements for this project. Adding more email customization options, the application can be extended to include more customization options such as adding attachments, customizing the email subject, or adding personalized content based on the recipient's information. Integrating with email marketing platforms, the application can be integrated with popular email marketing platforms like Mailchimp or Constant Contact, allowing businesses to manage their email campaigns more efficiently.

## REFERENCES

- [1] William S. Vincent, "Django for APIs: Build web APIs with Python and Django", 7 March 2018
- [2] Devin Knight, Brian Knight, Mitchell Pearson, Manuel Quintana, a top use case for MS Flow is to send custom email messages based on various trigger events.
- [3] Nigel George · 2016, If any of your Django views send email using Django's email functionality, Mastering Django: Core - Page 287
- [4] John Arnold, Ian Lurie, Marty Dickinson · 2009, Create a custom Thank You page for each individual product sold.