Documentation

Contents

Introduction

The requirement of this project arises because currently each donation made to NetBSD electronically from PayPal and Stripe are handled manually using Request Tracker and tickets. The person handling the tickets needs to extract the donation details and enter them into the acknowledgement form and send it to the donor. Further the donor can decide if they want to be listed on the public list of donor and inform by a mail. This process involves manual intervention which may result in error and/or delay. Therefore, if an automated system can be made that handles all these tasks, it will speed up the process, reduce errors, and also save time and energy of the person who performed all these tasks.

The primary goal of this project is to develop a system that automates all these processes : acknowledgement of the donor, storage of donation details in the database(to be created) and updating the public list of donor on the official NetBSD website based on the response received form the donor through the online feedback form.

Tools Used

Python – used for developing the parser.

Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry. Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.

Flask – used for developing the feedback collection website.

Flask is a web application framework written in Python. Armin Ronacher develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine.

PostgreSQL – used for developing the database.

PostgreSQL can be pronounced as Post-Gress-Q-L. PostgreSQL is an object-relational database management system (ORDBMS), which was developed by the PostgreSQL Global Development Group and is popular because it is powerful and open source. PostgreSQL Global Development Group is a worldwide team of volunteers, and thus PostgreSQL is not controlled by any corporation or any other private entity. Thus making it open source and free to use.

Details:

Installation instructions and Usage

Requirements:

- Python 3
  - Modules
    - typing
    - sys
    - ssl
    - datetime
    - smtplib
    - configparser
    - psycopg2
    - dateutil
    - bs4
    - flask
- PostgreSQL
- Postfix
- HTML5
- CSS3

Instructions:

1. To start using the system, first make sure that all the required **packages and modules** are installed on your device.
2. Edit the **config.ini** file according to your required configurations.
3. Make sure that the **flask server**, **postfix mailing agent** and **postgresql server** are configured and running before you start using the system.
4. Compile the mails you want to process in a folder and run the **parser.py** program on each file of the folder to start the process.
5. You will receive the non-donation related mails in the mail address provided in config.ini and the rest (donation related mails) will be parsed and details of the donors will be stored in the database. Simultaneously the donor will also receive the acknowledgement mail (with feedback link) on the address extarcted by the parser from the donation mail.
6. Once the donors provide their feedback, details will be stored on the database.

The Code

***parser.py*:**

It is the program containing all the essential methods for parsing the mails and extracting the details from the donation mails, forwarding the non-donation related mails to the admin, sending the acknowledgment to the donors and storing the details extarcted form the mails in the database.

Classes: The following classes are created for the successful working of the program.

1. *HTMLFile*
   This class is used for parsing the HTML file using the module *BeautifulSoup*. It has two attributes:
   1. *base_html:* String file parsed from the HTML file provided to the program.
   2. span_list: List of all the span tags in *base_html*.

2. *ByPayPal*
   This class is used for processing the donations made by PayPal. It has an attribute *html_file,* which is an object of the HTMLFile class.

3. *ByStripe*:
   This class is used for processing the donations made by Stripe. It has an attribute *html_file,* which is an object of the HTMLFile class.

Methods: The methods of this program can be categorized as follows:

1. *ByPayPal* methods:

   1. *get_date_time*:
   2. *get_email*:
   3. *get_details*:
   4. *is_donation*:
   5. *analyse_file*:

2. *ByStripe* methods:

   1. *get_text_details*:
   2. *is_text_donation*:
   3. *analyse_text*:
   4. *get_html_details*:
   5. *is_html_donation*:
   6. *analyse_html*:
   7. *is_html*:
   8. *analyse_file*:

3. Database related methods:

   1. *connect_database*:
   2. *prepare_and_insert*:
   3. *commit_and_close*:
   4. *insert_record*:

4. Mail related methods:

    1. *reply*:
    2. *forward*:

Driver Code:

References