Invoice Helpmates

A Utiliy Calculator and Tip Calculator

Darshil Kapadia
Master in Computer Science (Fall2022)
Stevens Institute of Technology
dkapadi2@stevens.edu

***Abstract*— Build a CLI (Command Line Interface) and GUI (Graphical User Interface -- Web) version on two tasks. First task is the Utility calculator for two flat mates and another task is tip calculator. In CLI, user provide the input in terminal, functions generate the pdf from provide information to generate pdf and automated email is forwarded with pdf attached to the user. While in GUI version, user provide information to on the web and will have the result in results page. During these various libraries like Flask, fpdf, Filestack, wtforms, and many more. Both versions are deployed on public platform so that all the users could have access to the project. During whole implementations PyCharm IDE was used.**

Keywords—flask, automate_email, CSS, pdf_generate, wtforms, deploy.

I. Introduction

Project 'Invoice Helpmates' consists of two daily life tasks. This project is developed in two version (i.e.: CLI and GUI). This project can perform two different tasks. First task is Utility calculator. Another task is the tip calculator.

Utility Calculator returns the total amount paid by each of the two following roommates as per the number of days the stay at the following place. This is useful in daily task like electricity bill, gas bill and more. While on the other hand Tip Calculator is being useful while in restaurant. Tip Calculator can perform the calculation for the amount paid by each members considering the total bill amount and tip percentage set by user.

In this project, CLI version of the project is programmed to generate the pdf from scratch by the user's input and generated pdf will be automatically e-mail to the provided email id with the auto generated subject. This CLI version is deployed on the repl.it platform which give easy access to the other user no matter what OS version or systems they are on.

In this project, GUI version of the project is programmed to efficiently work on web. It consists of three pages: Homepage, Form Page (Utility and Tip) and Results page. Homepage give two choices: Utility Calculator and Tip Calculator. Flask is used during this process in programming. Various attributes of wtforms libraries like SpringField, SubmitField is used. Render Template of Flask library is used. Various CSS and HTML attributes were also integrated to give demonstration pf python and web script integration.

The following GUI project is also deployed on the public platform for the same reason the CLI version is deployed. Each version uses numerous different libraries which makes the project more efficient and readable. I tried to build almost all the components to be object-oriented which will able other programmers to understand the code.

II. Dataset

For CLI version, user enters the data through command line interface of their system i.e.: command prompt or terminal.

For GUI version, users enter the data in the text field through your web browser. And after clicking on the calculate button users will have results appeared on their devices screen.

III. This Project

About various libraries and elements used during implementation of the projects.

A. CLI and GUI Version

a. Fpdf:

PyFPDF is a library for PDF document generation under Python, ported from PHP. Its main features are easy-to-use, Many simple examples and scripts available in many languages, No external dependencies, or extensions. Compared with other PDF libraries, PyFPDF is simple, small, and versatile, with advanced capabilities and easy to learn, extend and maintain.

b. FileStack:

This is the official Python SDK for Filestack - API and content management system that makes it easy to add powerful file uploading and transformation capabilities to any web or mobile application.

c. Email:

The email package is a library for managing email messages. The overall structure of the email package can be divided into three major components, plus a fourth component that controls the behavior of the other components. The central component of the package is an "object model" that represents email messages. The application can use this API to ask questions about an existing email, to construct a new email, or to add or remove email subcomponents that themselves use the same object model interface.

d. Flask:

Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running.

e. Wtforms:

WTForms is a flexible forms validation and rendering library for Python web development. It can work with whatever web framework and template engine you choose. It supports data validation, CSRF protection, internationalization (I18N), and more. There are various community libraries that provide closer integration with popular frameworks.

IV. Future Scope

This project can also be able to do more required operations like Grocery Calculator and Salary Raise Calculator many more. Currently, I am working implementing these all functions into the iOS mobile application using SwiftUI. I am using the XCode platform for the following process.

References

[1] https://pyfpdf.readthedocs.io/en/latest/

[2] https://github.com/filestack/filestack-python

[3] https://docs.python.org/3.9/library/email.html

[4] https://www.fullstackpython.com/flask.html

[5] https://pypi.org/project/WTForms/

[6] https://www.udemy.com/user/adiune/

[7] https://www.udemy.com/user/shubham-sarda-6/

[8] https://www.w3schools.com/css/default.asp