PYTHON-DJANGO PAYMENT INTEGRATION USING RAZORPAY

[1]Soham C. Adhyapak, [2]Vedant J. Deshpande, [3]Atharva A. Kulkarni, [4]Nikita R. Pawar, [5]Urmila C. Talikoti, [6]Prof. T. L. Patil

Student, Computer Science and Engineering Department, A.G. Patil Polytechnic Institute, Solapur[1,2,3,4,5],
Project Guide Computer Science and Engineering Department, A.G. Patil Polytechnic Institute, Solapur[6]

ABSTRACT

The main purpose was to demonstrate how we can add a payment gateway to any website using Razorpay. The department has recently completed a major effort in restructuring and refining its departmental goals, objectives and operations in preparation.

In order to streamline the process of encryption, decryption and how payment integration actually works, we have developed Web-based application in order to demonstrate the payment signature with proper security. This was developed and designed to present how an Online transaction works and how we can verify whether transaction is fraud or not.

The Payment gateway is integrated in Django website using two keys such as private key and public key which is generated at Razorpay side. An assistive chatbot interface is also provided for the users to know the information about this website and to solve the queries of the users. This chatbot is implemented using Diaglogflow and that model is being trained for real-time users.

Keywords: Payment Gateway, API Integration, end-to-end Transactions

INTRODUCTION

Initially, user need to create an account to access this website or to login with their account with proper credentials. If the user is new, then he needs to verify his account by the token link sent to their respective email.After that user will be provided the website once the mail is verified. On the website in order to demonstrate how actually online transactions works using third party interface, we have provided sample products with their information and price and depending on that price the transaction will be carried out.

After selecting the product and providing data to the Razorpay API, a unique key will be generated that will be the response of API in form of JSON. Another key will be generated at the backend and stored in database. The key generated and the key returned by the API will be compared for security purpose and based on that value further activity will be decided. After that payment methods will be provided for the user to complete the transaction. One Time Password will be sent to their respective mobile number for verification purpose.

As part of the online transaction, how actually the transaction is carried out, the demo of the online payment is implemented using Razorpay. Also, a navigation bar is also provided which includes the information about the project, members and the feedback option.

The Chatbot is also integrated with this website using Diaglogflow which solves the queries of real time users.The basic process includes handling user data, processing data and providing the resulted output are organized differently in different development process. The team is will be divided to work on each module of project. All the members of our team will play separate roles in each part of project assigned to them

LITERATURE SURVEY

Integrate Razorpay Standard Checkout with your website to start accepting online payments from your customers. Razorpay supports a various modes of payment methods such as net banking, credit and debit cards, wallets and UPI. Our Standard Checkout library provides all the essential features for integrating Razorpay Checkout with the client-side of your application. This is available only for web-based integrations.

For this purpose, a special framework of python "Django" is used. As Django follows the principle of "Model-View-Template" architecture, it is possible to demonstrate the three-layer architecture in payment gateway integration. The main focus is to use most of the open-source data efficiently. Razorpay is an open source and easy interface for processing online transactions. It provides standard checkouts for your website to start accepting online payments from customers. Razorpay supports various payments such as UPI, Net Banking, Credit and Debit cards, QR code.

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because at this point in time, not too many detailed designs of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis. Understand the different technologies involved in the proposed system before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the new system. Find out whether the organization currently possesses the required technologies. Is the required technology available with the organization?

The technical feasibility is frequently the most difficult area encountered at this stage. It is essential that the process of analysis and definition be conducted in parallel with an assessment to technical feasibility. It centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed system

PROPOSED SYSTEM ARCHITECTURE

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI'S at the top level have been categorized as

Administrative user interface & Operational or Generic user interface.

The system after careful analysis has been identified to be presented with the following modules:

The modules involved are:

1.Admin: In this module, Admin has all the access to the database, including the Checkouts, Profiles, Products. The admin panel is generated by registering the admin to the website and by creating a super user with their email and password.

2. Login: In this module, user can login into their account in order to access the main website. User cannot the payment website until he is logged into the website.

3.Register: In this module, before accessing the main website user needs to create and account if he/she is new to the website. The process of creating includes data to be entered such as name, password and email. Based on that email, an authentication token is sent to the user's email. Until that email is verified, the Boolean value in database of users (is_verified) is set to false.

4. Feedback: In this module, User is provided with feedback based on the usage of the website by user.Feedback is provided in two forms either in star ratings or in form-based feedback.

5. Pay with Razorpay: This is the API Integration button which is displayed on when the user selects one of the donations and enters amount to be donated and when the API keys of Razorpay has established the connection with the project.

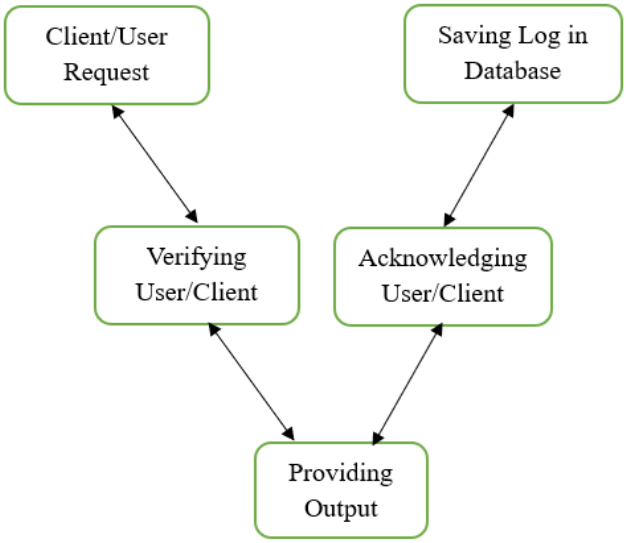
6. Chatbot: A real time chatbot is also developed using Dialog flow by google cloud. This chatbot is trained to solve all the queries of user starting from asking the name to explaining the website to the user. The bot is trained with the intents and the responses of all kinds.

7. Home: This module is in the Navbar, whenever clicked will redirect to the home page of the website.

8. About: This module provides all the details of the project.

9. Members: Provides all the necessary information of the team members.

10. Logout: Logouts the user from the main website and redirects to the login page.



Flow of the project

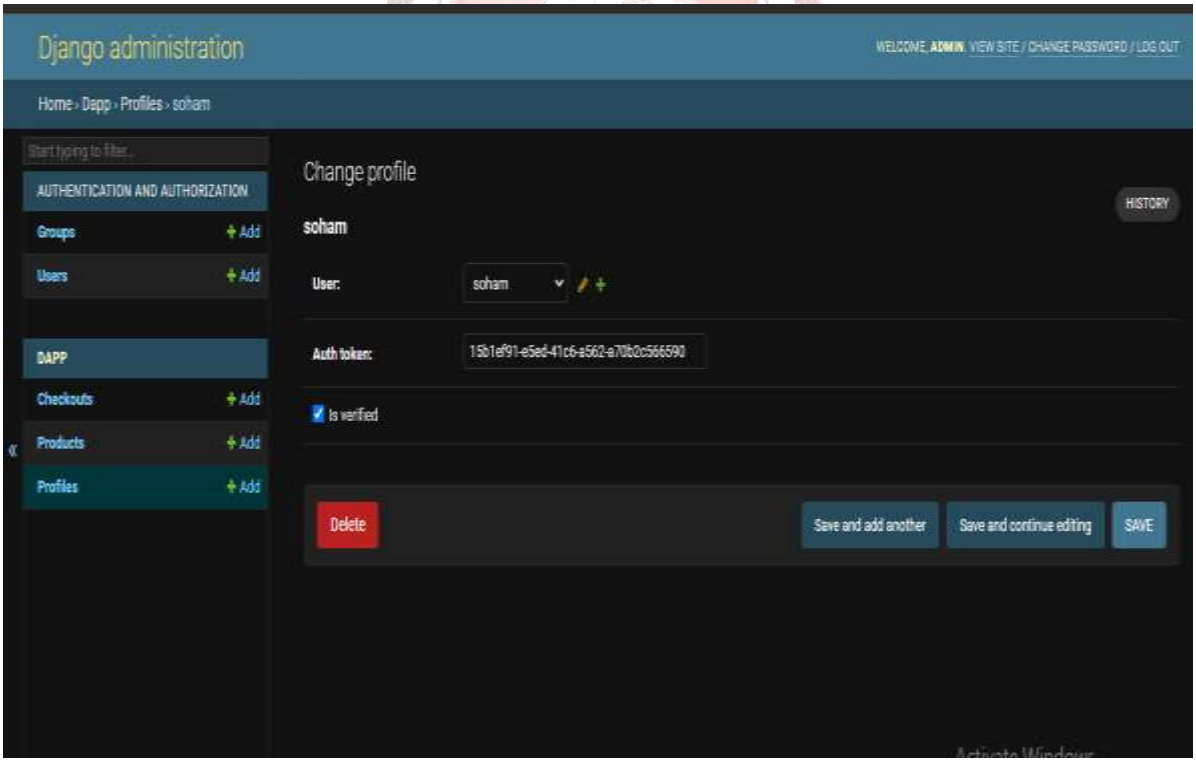
WORKING

This project is developed to ensure that at each step authentication is performed. The working is as follows:

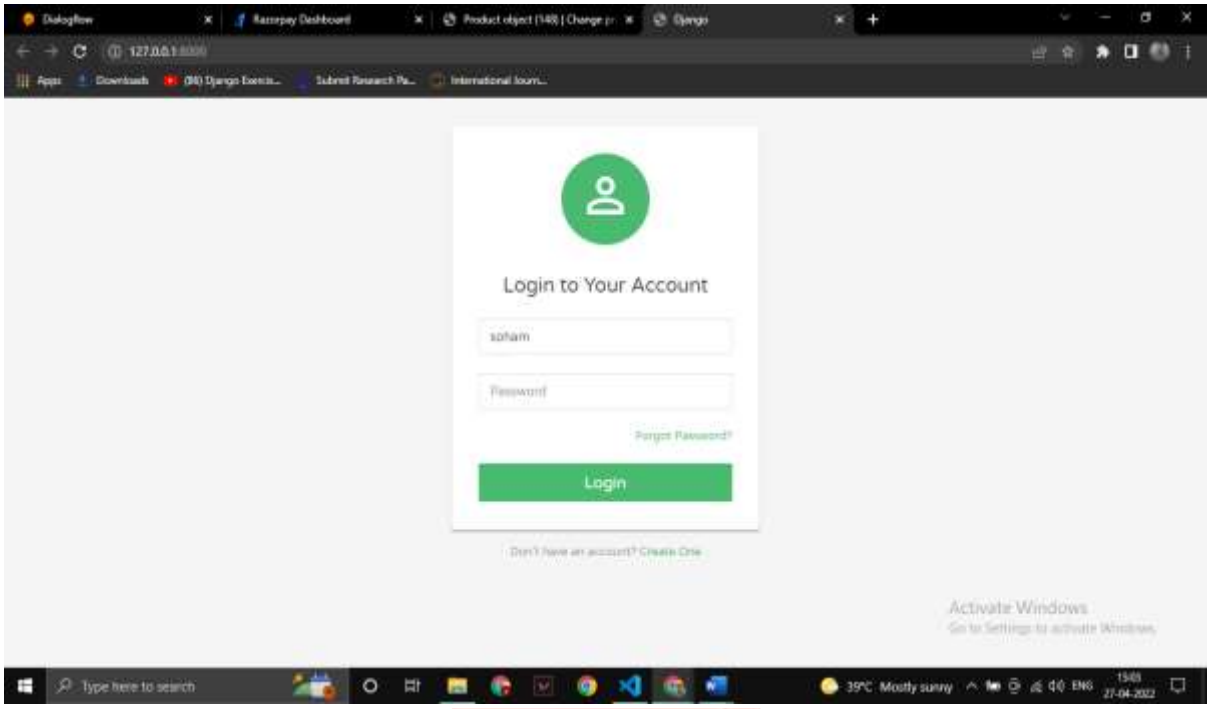
1. Takes the necessary information from the UI/Front End
2. Generates the unique payment ID for particular user.
3. Generates the Razor Pay client in order to process the transaction.
4. Creates an order and the information is passed for further transaction.
5. Provides the payment options
6. Compares the Razorpay id and id generated at backend.
7. Creates payment signature
8. If token is matched, payment is completed and directed to success page.
9. If token is not matched, payment is cancelled and if money is debited, it will be refunded in 2-4 business days.

RESULT

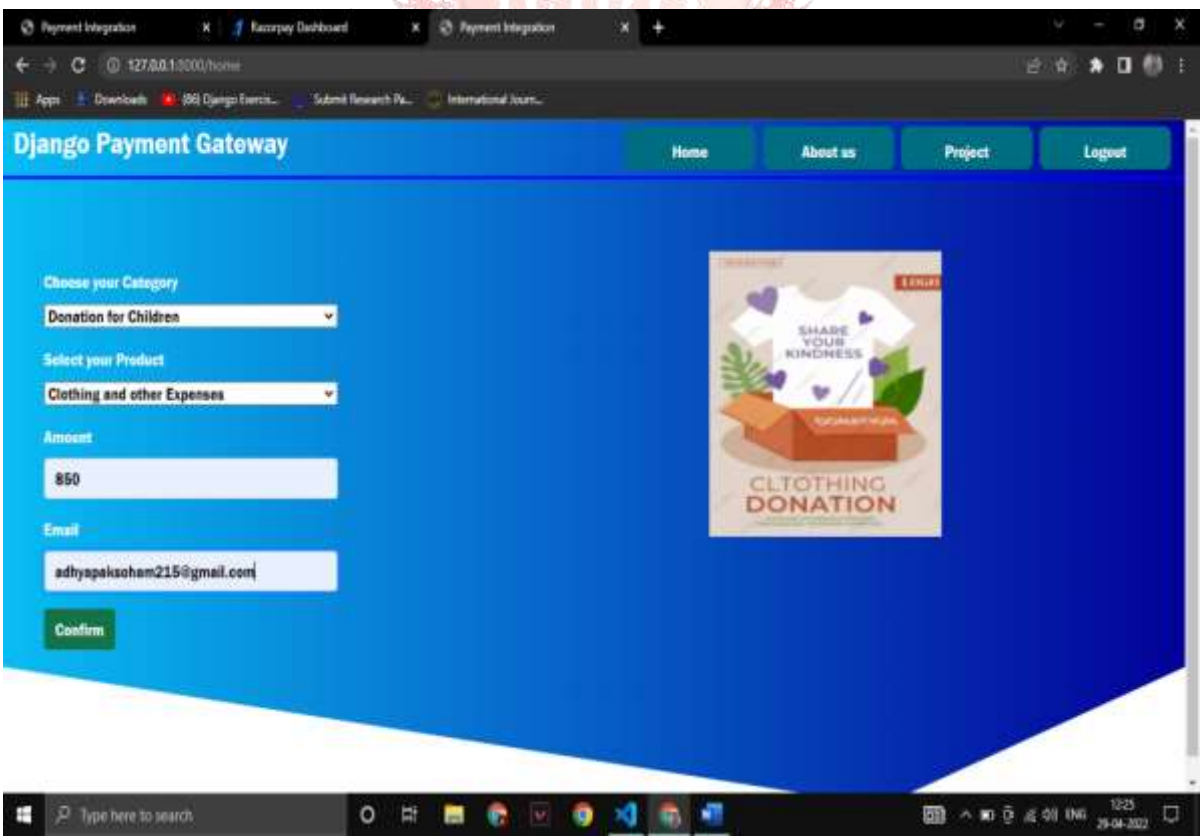
Whenever transaction is initialized, key is generated and stored in database, and when the transaction is completed successfully then a key is returned from the Razorpay Database Dashboard. When both keys are same then we can conclude that the transaction is completed successfully. Based on the status of transaction, the status is updated in database. Once the transaction is completed successfully, an acknowledgement receipt is sent to the user's email.



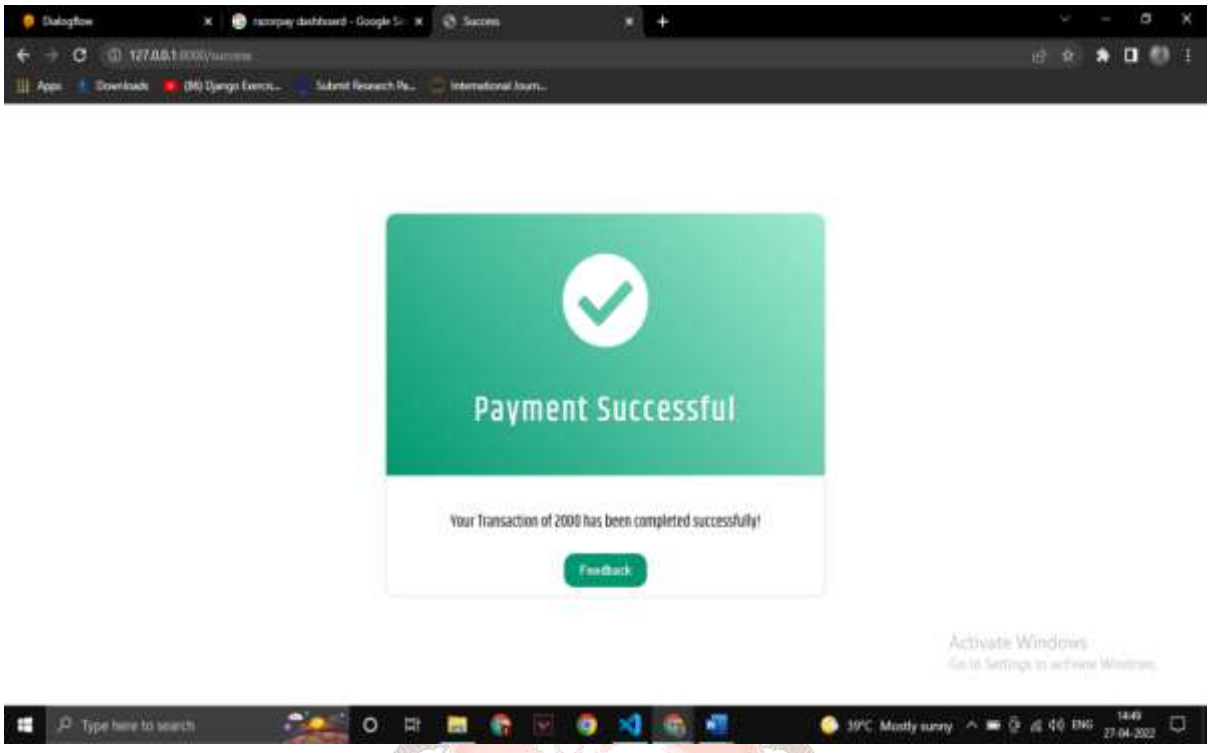
Database for profiles



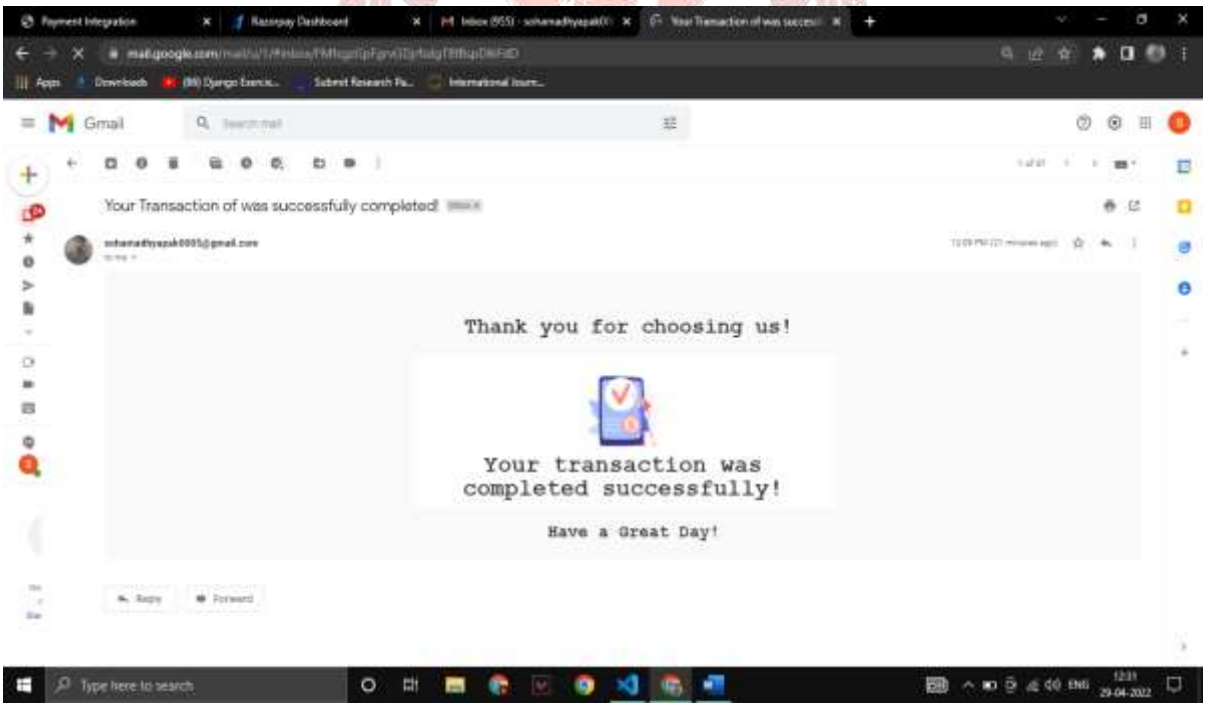
Login Page



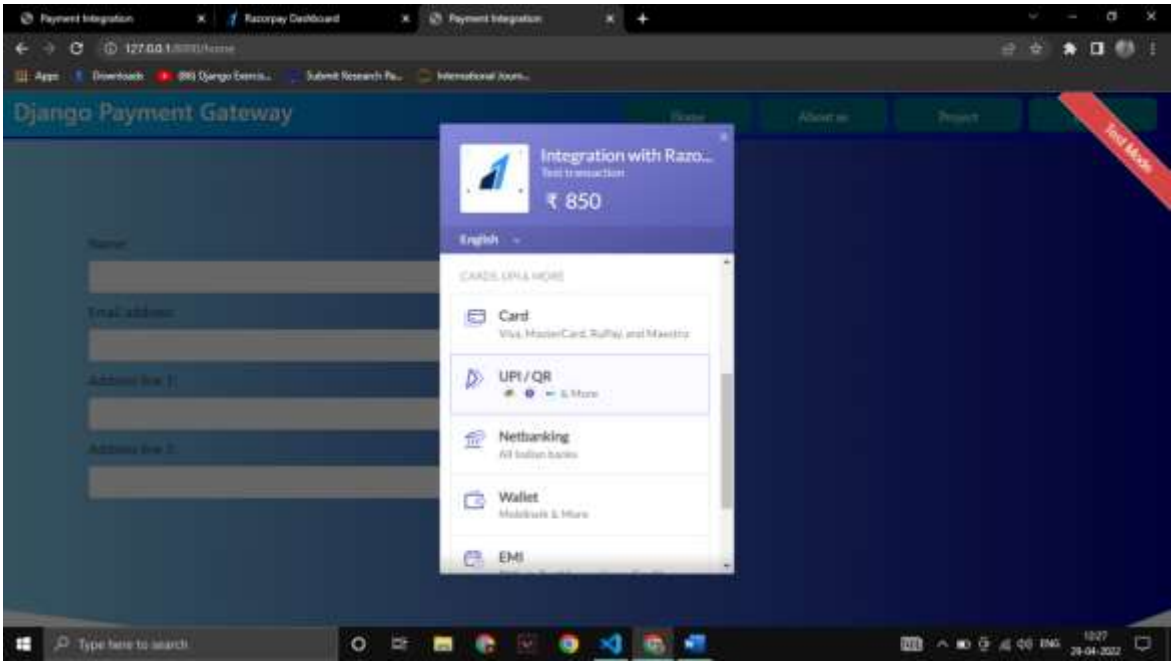
Homepage



Payment successful message



Acknowledgment message



Payment methods

CONCLUSION

1. Fully automated system
2. Centralized database maintenance
3. Easy access
4. This system provides a payment gateway for all types of payment which is end-to-end encrypted.

REFERENCES

[1] Django 1.2 Jesse Legg Packt Publishing.
[2] Django unleashed.
[3] www.razorpay.com
[4] www.docs.djangoproject.com
[5] www.realpython.com
[6] www.developer.mozilla.org