

DoctorQ with Payment Integration

Project Report Submitted in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Engineering *in* **Information and Technology**

Submitted by

Gautam Bansal: (Roll No. 19UITE9009)

Chirag Soni: (Roll No. 19UITE9007)

Under the Supervision of

Mentor - Dr. Alok Singh Gahlot

Assistant Professor

Guide - Abhisek Gour

Assistant Professor



Department of Computer Science and Engineering
Faculty of Engineering & Architecture
M.B.M University, Jodhpur

July, 2022

DoctorQ with Payment Integration

Project Report Submitted in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Engineering *in* **Information and Technology**

Submitted by

Gautam Bansal: (Roll No. 19UITE9009)

Chirag Soni: (Roll No. 19UITE9007)

Under the Supervision of

Mentor - Dr. Alok Singh Gahlot

Assistant Professor

Guide - Abhisek Gour

Assistant Professor



Department of Computer Science and Engineering
Faculty of Engineering & Architecture
M.B.M University, Jodhpur

July, 2022



Department of Computer Science & Engineering

M.B.M.University
Ratanada, Jodhpur, Rajasthan, India -342011

CERTIFICATE

This is to certify that the work contained in this report entitled **“Doctor Queue with Payment Integration”** is submitted by the group members Mr.. Gautam Bansal (Roll. No: 19UITE9009) and Mr. Chirag Soni (Roll No: 19UITE9007) to the Department of Computer Science, M.B.M University, Jodhpur, for the partial fulfillment of the requirements for the degree of **Bachelor of Engineering in Information Technology**.

They have carried out their work under my supervision. This work has not been submitted else-where for the award of any other degree or diploma.

The project work in our opinion, has reached the standard fulfilling of the requirements for the degree of Bachelor of Engineering in Information Technology in accordance with the regulations of the Institute.

Mentor - Dr. Alok Singh Gahlot Sir
Associate Professor
Guide - Abhisek Gour Sir
Assistant Professor

Dept. of Computer Science & Engg.
MBM University, Jodhpur

Prof. N.C. Barwar
(Head)
Dept. of Computer Science & Engg.

Acknowledgment

We, Chirag Soni and Gautam Bansal would sincerely like to thank Dr. Nemi Chand Barwar, Head of Department, Computer Science & Engineering, MBM University, Jodhpur for the support and availability of facilities by the department. I wish to express my deepest sense of gratitude to Assistant Prof. Abhishek Gaur, Associate Prof. Dr. Alok Singh Gahlot, for his able guidance and useful suggestions, that helped me in completing the project work, on time. His guidance, encouragement, suggestion, and constructive criticism have contributed immensely to the evolution of my ideas on the report.

Finally, yet most importantly, I would like to express my heartfelt thanks to my family, friends, and peers for their blessings, wishes, and support for the successful completion of this report.

Abstract

Life is becoming too busy to get medical appointments in person and to maintain proper health care. The main idea of this work is to provide ease and comfort to patients while taking appointments from doctors and it also resolves the problems that the patients have to face while making an appointment. The web application DoctorQ helps patients to place themselves in the queue digitally, without making physical efforts and it also provides facility to pay the fee digitally.

Contents

Chapter 1	1
Introduction	1
1.1 Problem Statement	1
1.2 Our Approach	2
Chapter 2	3
Technologies and Tools Used	3
2.1 Front-End	3
2.1.1 HTML	3
2.1.2 CSS	3
2.1.3 JavaScript	4
2.2 Flask	4
2.3 MySQL	5
2.4 VS Code	5
2.5 Github	6
2.6 PayTM Payment Gateway	7
2.6.1 Fundamentals of collecting payments with Paytm	7
2.6.2 Detailed Payment Flow Explained	7
Chapter 3	10
Design Interface	10
Chapter 4	14
Project Overview	14
4.1 System Specification Requirements	14
4.2 User/Admin manual	14
Chapter 5	16
Conclusion and Future work	16
References	17

List of Figures

1.1 Long queues even during the pandemic	1
2.6.1 PayTM Payment Flow Chart	9
3.1 Home Page	10
3.2 SignIn Page	10
3.3 SignUp Page	11
3.4 User DashBoard	11
3.5 Admin DashBoard	12
3.6 Payment Page	12
3.7 Payment Page 2	13
3.8 Payment Page after Pyament Failure	13

Chapter 1

Introduction

In the modern era of computerisation, we expect to resolve real world problems using computers and our project is an example of the same.

We live in a country with population greater than 100 crores. So, finding a lot of crowd for even the basic services is common. To manage the crowd the most common method is forming a queue. So our aim is to form a digital mechanism to handle such long queues and even providing payment facilities.



Fig. 1.1 - Long queues even during the pandemic

1.1 Problem Statement

India is a country with a vast population, which makes it tougher to access basic services like medical facilities, we can see long queues even at salons. So, waiting in long queues to get medical treatment isn't surprising. But, standing in queue can be frustrating sometimes, as leaving the queue can result in losing the position.

Problem Statement -: To make a mechanism that doesn't require any person to be physically present at the venue of the queue to mark their presence/position.

1.2 Our Approach

Our approach is to convert the physical queue into a digital one, where instead of a physical person their mobile will mark their digital presence. Firstly, we'll put QR codes outside the clinics or hospitals that have long queue problems. Now, any patient can simply scan the code which will make them land on a payment page, where they can pay the service fee and mark themselves into the queue. In this way, they won't need to stand in that queue and simply check their systems to see the number of patients ahead of them in the queue.

Chapter 2

Technologies and Tools Used

In this project, we have used vast number of technologies and tools. Let's know about them-:

2.1 Front-End

2.1.1 HTML

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within a tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-read.

2.1.2 CSS

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. It describes how a web page should look: it prescribes colors, fonts, spacing, and much more. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser.

while html uses tags, css uses rulesets. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

2.1.3 JavaScript

JavaScript is a lightweight, cross-platform, and interpreted compiled programming language which is also known as the scripting language for web pages. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments. Javascript is both imperative and declarative type of language .JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.

2.2 Flask

Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Pocco. Flask is based on the Werkzeug WSGI toolkit and the Jinja2 template engine.Both are Pocco projects.

WSGI

The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

Werkzeug

Werkzeug is a WSGI toolkit that implements requests, response objects, and utility functions. This enables a web frame to be built on it. The Flask framework uses Werkzeg as one of its bases.

jinja2

jinja2 is a popular template engine for Python.A web template system combines a template with a specific data source to render a dynamic web page.

Microframework

Flask is often referred to as a microframework. It is designed to keep the core of the application simple and scalable.

Instead of an abstraction layer for database support, Flask supports extensions to add such capabilities to the application.

2.3 MySQL

MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL).

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a place to hold vast amounts of information in a corporate network. In particular, a relational database is a digital store collecting data and organizing it according to the relational model. In this model, tables consist of rows and columns, and relationships between data elements all follow a strict logical structure. An RDBMS is simply the set of software tools used to actually implement, manage, and query such a database.

MySQL is integral to many of the most popular software stacks for building and maintaining everything from customer-facing web applications to powerful, data-driven B2B services. Its open-source nature, stability, and rich feature set, paired with ongoing development and support from Oracle, have meant that internet-critical organizations such as Facebook, Flickr, Twitter, Wikipedia, and YouTube all employ MySQL backends.

2.4 VS Code

Visual Studio Code (famously known as **VS Code**) is a free open source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times.

VS Code supports a wide array of programming languages from Java, C++, and Python to CSS, Go, and Dockerfile. Moreover, VS Code allows you to add on and even creating new extensions including code linters, debuggers, and cloud and web development support.

The VS Code user interface allows for a lot of interaction compared to other text editors.

2.5 Github

GitHub is a web-based version-control and collaboration platform for software developers. Microsoft, the biggest single contributor to GitHub, initiated an acquisition of GitHub for \$7.5 billion in June, 2018. GitHub, which is delivered through a software-as-a-service (SaaS) business model, was started in 2008 and was founded on Git, an open source code management system created by Linus Torvalds to make software builds faster.

Git is used to store the source code for a project and track the complete history of all changes to that code. It allows developers to collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers. GitHub allows developers to change, adapt and improve software from its public repositories for free, but it charges for private repositories, offering various paid plans. Each public or private repository contains all of a project's files, as well as each file's revision history. Repositories can have multiple collaborators and can be either public or private.

GitHub facilitates social coding by providing a web interface to the Git code repository and management tools for collaboration. GitHub can be thought of as a serious social networking site for software developers. Members can follow each other, rate each other's work, receive updates for specific projects and communicate publicly or privately.

Three important terms used by developers in GitHub are fork, pull request and merge. A *fork*, also known as a *branch*, is simply a repository that has been copied from one member's account to another member's account. Forks and branches allow a developer to make modifications without affecting the original code. If the developer would like to share the modifications, she can send a *pull request* to the owner of the original repository. If, after reviewing the modifications, the original owner would like to pull the modifications into the repository, she can accept the modifications and *merge* them with the original repository. Commits are, by default, all retained and interleaved onto the master project, or can be combined into a simpler merge via commit squashing.

2.6 PayTM Payment Gateway

Paytm Payment Gateway provides a secure, PCI-compliant way to accept Debit/Credit card, Net-Banking, UPI and Paytm wallet payments from customers.

2.6.1 Fundamentals of collecting payments with Paytm

- Your customer clicks the **Pay** button in your web/mobile application.
- The customer is shown a checkout form where she fills in her payment details and authorizes the payment.
- After completion of a transaction, Paytm posts the response (success or failed) on a Callback URL defined by you.
- As a recommended security measure, you validate each transaction response via a server-to-server (S2S) API call. Transaction revalidation protects from request/response tampering possible in-browser calls. This S2S call is not required for Paytm plugins and hosted e-commerce website integration solutions.
- Based on the response received, you display the order status to the customer.
- See a real-time summary of payments received and other insights in your dashboard.
- Receive payments collected from customers in your bank account on the next business day.

2.6.2 Detailed Payment Flow Explained

Transaction Creation

When a transaction request is received at Paytm's server, there are multiple validations carried out like valid source of the request, the structure of request, uniqueness of request etc. Once these validations are passed, a transaction is created.

Successful Transaction

The customer fills in basic payment details to authorize the payment. Once the authorization is successful, money is debited from the customer's account. This transaction is a successful transaction.

Failed Transaction

If the customer drops out from the payment process, or in the event of payment authorization failure, money is not deducted from the customer's account. This is marked as a failed transaction.

Pending Transaction

Sometimes Paytm doesn't receive real-time transaction status from the bank. This can be due to many reasons such as network issues, technical errors at customer's/bank's end etc. This is marked as a pending transaction. Refer to this for detailed handling of the pending transaction.

Settled Transaction

Payments received against successful transactions are credited into your bank account on T+1, where T is the date of a successful transaction. Once the payment is credited, the corresponding transaction is marked as Settled.

Refund Transaction

Sometimes there are use cases where you need to reverse payments for successful or settled transactions. The reversal transaction of a successful or settled payment is called a refund transaction.

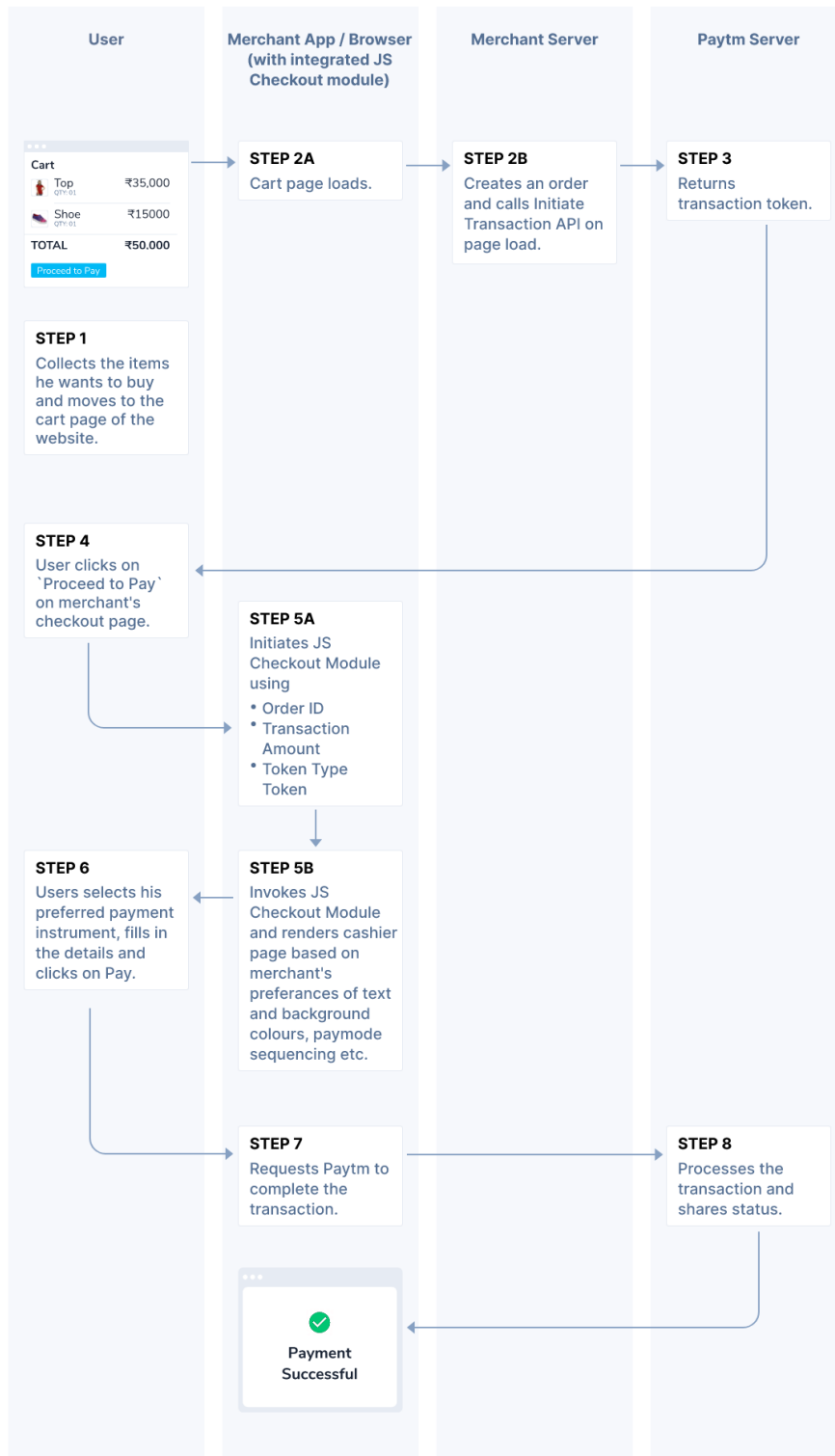


Fig. 2.6.1 - PayTM Payment Flow Chart

Chapter 3

Design Interface

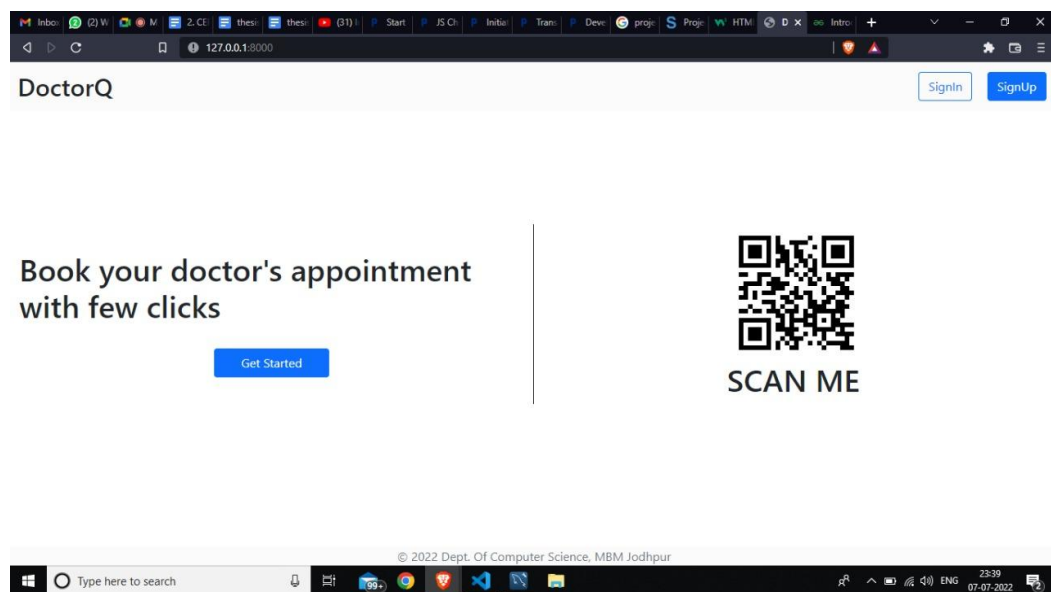


Fig 3.1 - Home Page

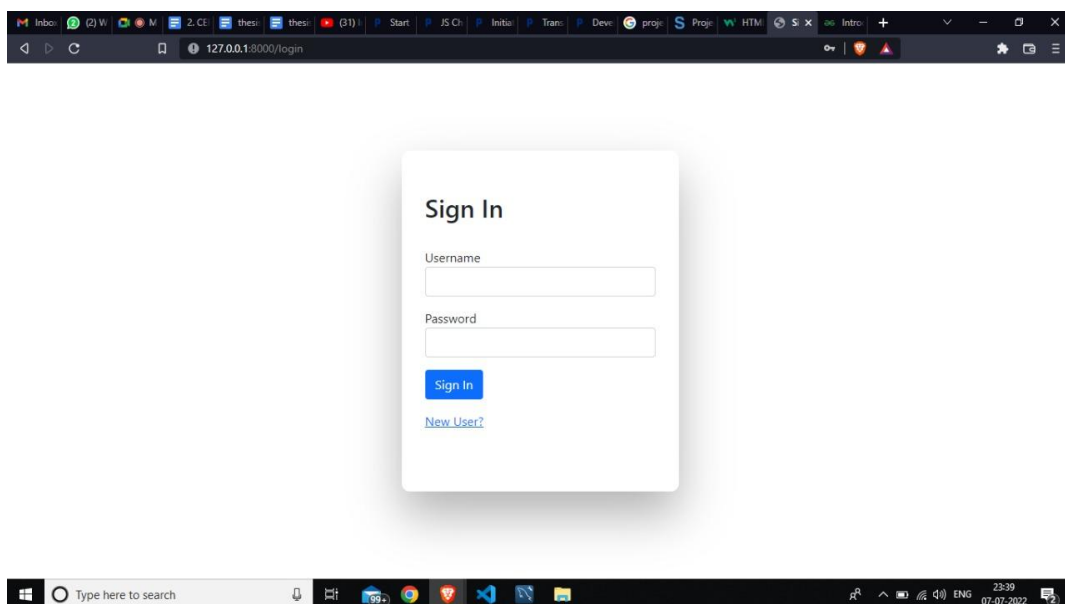


Fig 3.2 - SignIn page

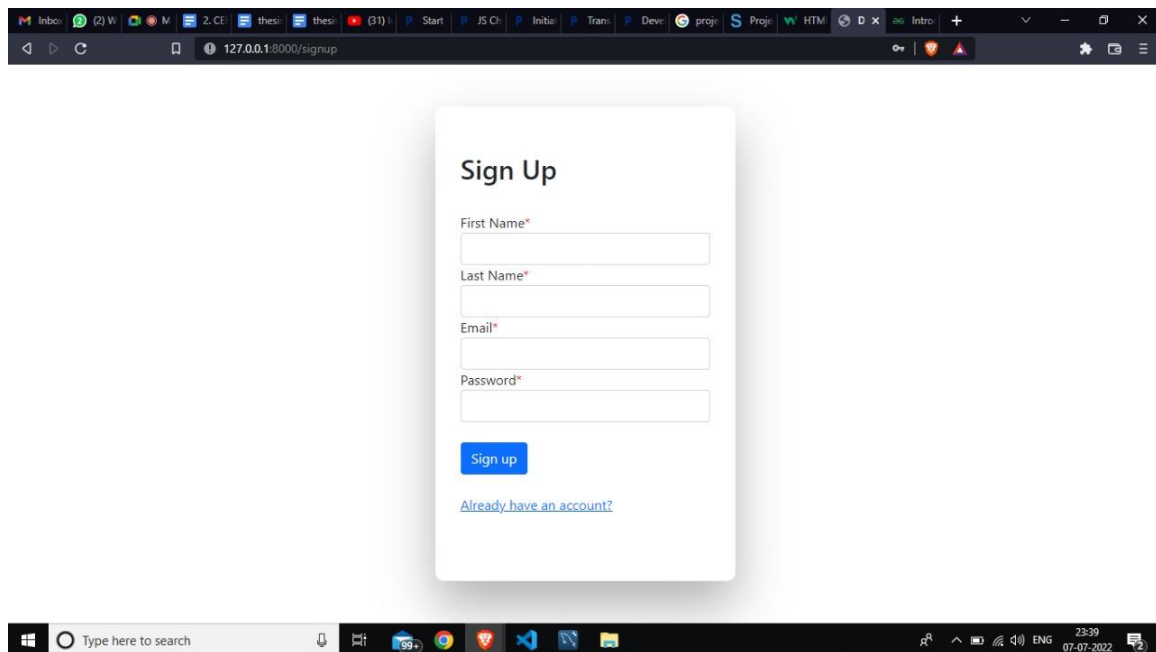


Fig 3.3 - SignUp page

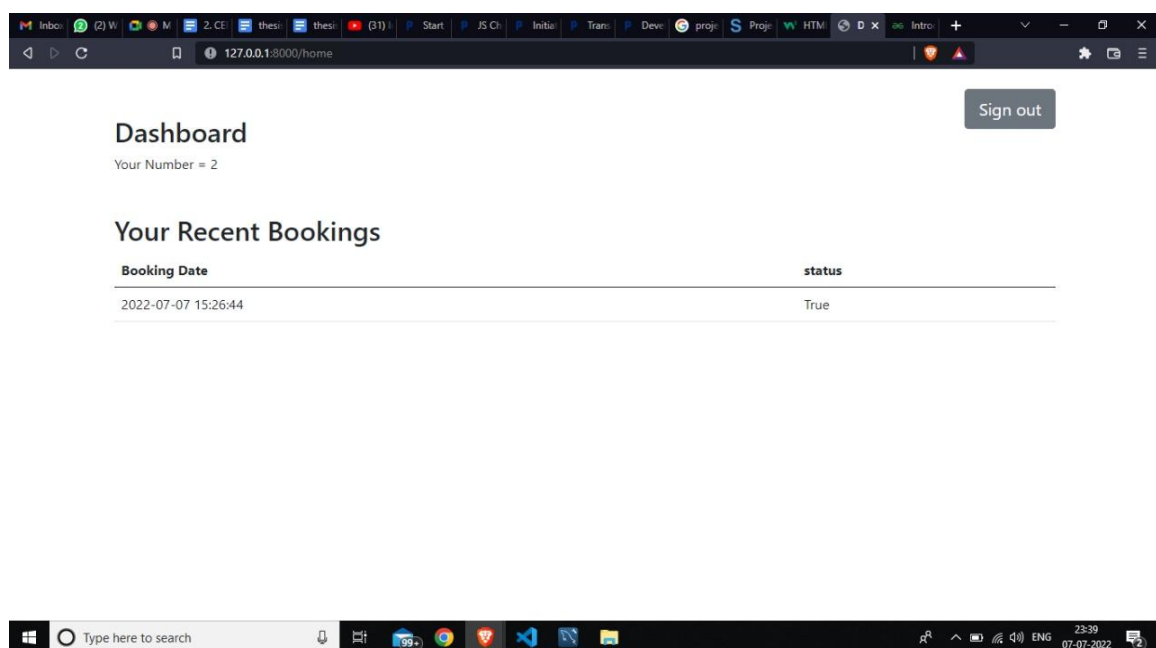


Fig 3.3 - User DashBoard

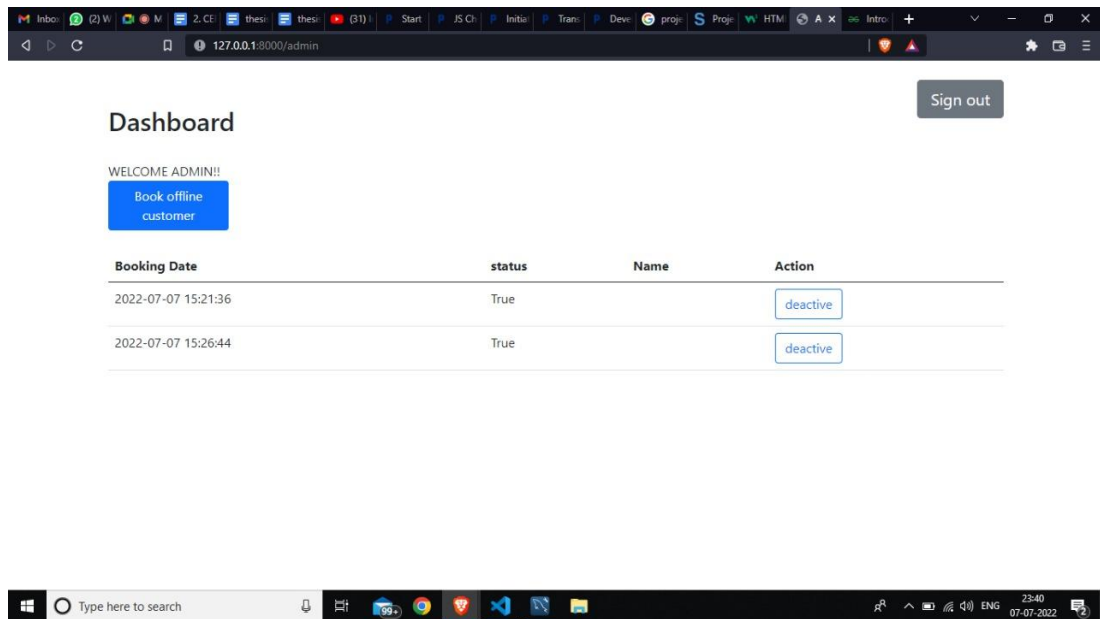


Fig 3.5 - Admin DashBoard

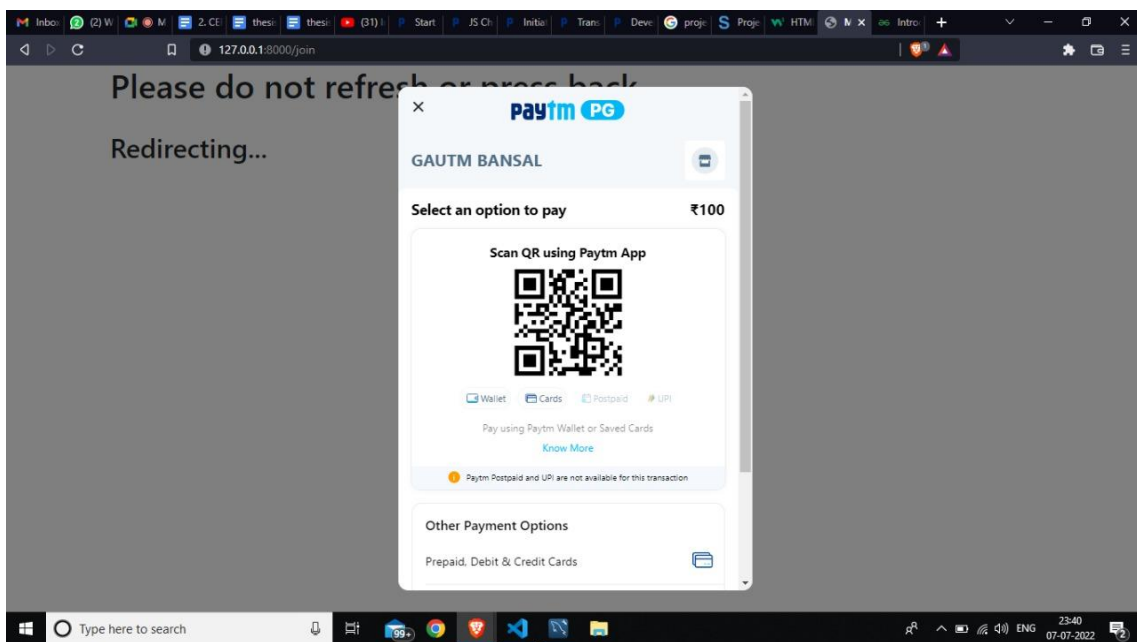


Fig 3.6 - Payment Page

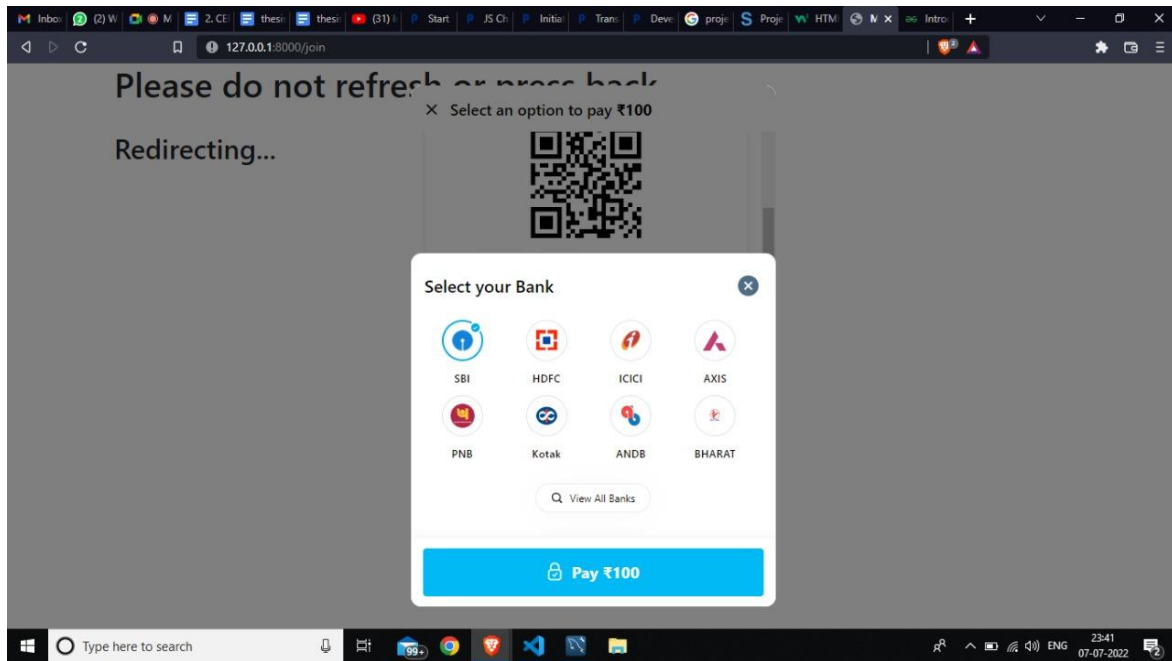


Fig 3.7 - Payment Page 2

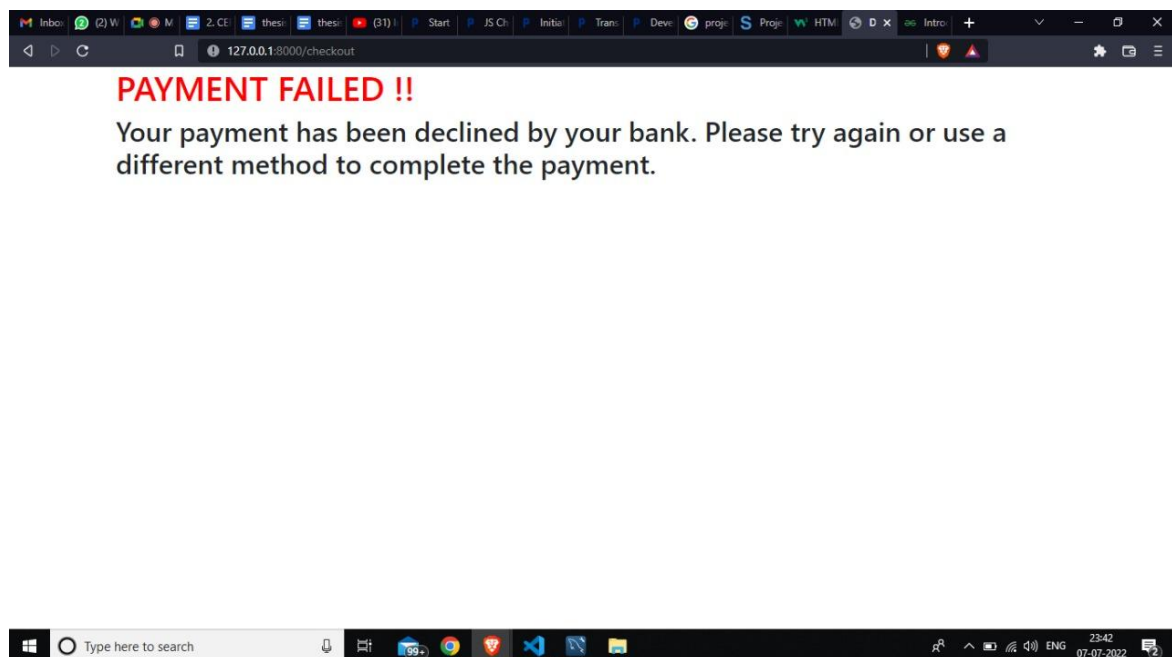


Fig 3.8 - Payment Page after Payment Failure

Chapter 4

Project Overview

Through this chapter, we want to provide an overview of our project like system requirements, user manuals etc.

4.1 System Specification Requirements

The following packages should be installed globally, as the superuser, for all users on the system to access.

Python 2.7.x or 3.4 and above.

Python development libraries (i.e. header files for compiling C code)

pip

virtualenv

virtualenvwrapper

Operating system: Linux- Ubuntu 16.04 to 17.10, or Windows 7 to 10, with 2GB RAM (4GB preferable) 2. You have to install Python 3.6 and related packages

4.2 User/Admin manual

In this project, there are two interfaces user and admin. Here's a user guide -:

1. Log in to DoctorQ website, and sign up using email address and password.
2. After logging in, user will face a dashboard showing the number of queuers before him, and an option to place themselves in the queue.
3. After pressing the "Join" button, user will be redirected to the payment page to make advanced payment, to eliminate unethical users.
4. From that payment page, user needs to pay the appointment fee to get themselves registered in the queue.

5. After successful payment, user will be redirected to earlier dashboard, which then will show the queue status, and the number at which user is standing.

Admin manual -:

1. Login to the DoctorQ website, and login using email and password. (Currently the admin sign up button is not available).
2. After logging in, Admin will face admin dashboard that will have all the facilities to manage the queue.
3. Dashboard will have options to manually add an offline user, remove an existing user from queue.

Chapter 5

Conclusion and Future work

The proposed digital queue mechanism website is developed using HTML, CSS, JavaScript and Flask (python). The tasks involved in this work are divided into modules. The data is approached and shared by using API'S between the website. The proposed system is efficient and has friendly user interface. Addition of the admin module in the application makes it easy for the admin to handle all the patients. The admin would be able to use the web app for managing the details of the patients and the doctors. A payment or some amount would be charged to the users/patients while making an appointment to avoid the unethical users. As many users only register themselves just for fun and has no concern by making an appointment.

Scope for Future Work -:

1. Making this DoctorQ website generalised to be used for other purposes and businesses too. Because we feel the need to resolve the queue issue at many other places too, like salon, restaurants, airports etc.
2. Since, currently the user interface of the system is not upto the standards of other fancy easy to use websites. Hence, work can be done to improve UI/UX design.
3. Currently the payment gateway only accpets the payments in one currence i.e. Rupee. Work can be done on gateway to be able to accept different currencies too.
4. Work can be done to make the queue managing algorithm more efficient.

References

- [1] "What is MySQL? Everything You Need to Know," *Talend - A Leader in Data Integration & Data Integrity*. [Online]. Available: <https://www.talend.com/resources/what-is-mysql/>. [Accessed: 07-Jul-2022].
- [2] "What is flask python," *Pythonbasics.org*. [Online]. Available: <https://pythonbasics.org/what-is-flask-python/>. [Accessed: 07-Jul-2022].
- [3] A. Z. Mustafeez, "What is Visual Studio Code?," *Educative: Interactive Courses for Software Developers*, 31-Jul-2020. [Online]. Available: <https://www.educative.io/answers/what-is-visual-studio-code>. [Accessed: 07-Jul-2022].
- [4] M. Courtemanche, "GitHub," *SearchITOperations*, 06-Jun-2018. [Online]. Available: <https://www.techtarget.com/searchitoperations/definition/GitHub>. [Accessed: 07-Jul-2022].
- [5] "Start building with Paytm," *Paytm.com*. [Online]. Available: <https://business.paytm.com/docs/getting-started>. [Accessed: 07-Jul-2022].
- [6] "Welcome to flask — flask documentation (2.1.X)," *Palletsprojects.com*. [Online]. Available: <https://flask.palletsprojects.com/>. [Accessed: 07-Jul-2022].