Dr. Sanjeev Netralaya
Online Booking

Prakash S Rathod

June 14, 2023

Abstract

This project report presents the implementation and customization of a token and appointment system. The system allows individuals to schedule appointments, tracks available slots per location, and assigns tokens and appointment times. The code provided in this project has been customized to incorporate age-based prioritization, doctor selection, and language preferences for doctors. This report provides a detailed description of the project, code structure, implementation details, compilation instructions, and the results obtained.

1 Introduction

The token and appointment system is designed to streamline the appointment scheduling process and ensure efficient allocation of resources. This report presents the implementation of the system, along with additional customizations to meet specific requirements. The goal of the project is to create a system that assigns tokens and appointment times based on various factors such as age, doctor availability, and language preferences.

2 Implementation Details

The implemented code is based on JavaScript and utilizes Google Apps Script for integration with Google Sheets and email functionalities. The code reads data from a Google Sheet containing applicant information, including names, emails, locations, doctors, and language preferences. It assigns tokens and appointment times based on the availability of slots per location, while considering the maximum number of appointments allowed per doctor.

3 Code Structure

The code structure consists of the following files:

- **project.js**: This file contains the main function `sendTokensAndAppointments`, which reads the applicant data from the Google Sheet and assigns tokens and appointment times. It also calls the `sendEmail` function to send confirmation emails to the applicants. This file contains the helper functions `calculateAppointmentTime` and `sendEmail`. The `calculateAppointmentTime` function calculates the appointment time based on the assigned token, and the `sendEmail` function sends confirmation emails to the applicants.

4 Compilation Instructions

To run the token and appointment system, follow these steps:

1. Open the Google form and fill the google form .
2. I will share the access for corresponding google sheets.
3. Open the associated Google Apps Script editor.
4. Copy and paste the code from the provided files (`project.js`) into the editor.
5. Save the code and run the `sendTokensAndAppointments` function.

Ensure that the necessary permissions are granted for accessing the Google Sheets and sending emails using Google Apps Script.

5 Results and Discussion

The implemented token and appointment system successfully assigns tokens and appointment times based on various factors such as availability of slots, doctor preferences. The system effectively handles duplicate entries and sends confirmation emails to the applicants. In one day total 30 slots are there. There were total 5 locations are there and 3 Doctors. Doctors were language specified and if anyone wants to according to their language status. Per day each doctor recieve 2 appointments according to each location.

References

[1] Spreadsheet.dev. (2023). *Send an Email for Every Row in a Google Sheet*. Retrieved from `https://spreadsheet.dev/send-an-email-for-every-row-in-a-google-sheet`

[2] w3schools.com. (2023). *JavaScript Tutorial*. Retrieved from `https://www.w3schools.com/js/`

[3] Garuba, O. (2023). *Google Apps Script: Automated Emails*. Retrieved from `https://www.codementor.io/@olatundegaruba/google-apps-script-automated-emails-m2m0ojq9v`