PatientFlow : Efficient Patient Appointment Management for Rural Clinics

Overview

PatientFlow is a lightweight desktop-based appointment booking and queue management system designed to streamline Outpatient Department (OPD) processes, especially for rural clinics with limited resources. The system enables patients to book appointments through a simple GUI while allowing doctors or receptionists to manage and monitor the appointment queue in real time.

Objectives

- Reduce long wait times in rural OPDs.
- Allow patients to pre-book 10-minute slots within preferred time brackets.
- Provide a live-updating dashboard for doctors to monitor, check in, or mark absentees.
- Ensure clean UI/UX with support for dark and light modes.
- Build a system ready for future cloud-based multi-device syncing.

Key Features

1. Patient-Side Application

- Built using **Tkinter** for a clean and responsive GUI.
- Allows users to enter:
- Full Name
- Age
- Gender
- Preferred Time Bracket (e.g., 10:00–11:00 AM)
- Automatically assigns the next available **10-minute** time slot within the selected bracket.
- Prevents double-booking by checking existing reservations from a shared JSON file (queue_data.json).
- Notifies patients of successful bookings or if a time bracket is full.
- Dynamic theming based on dark/light mode (Mac).

2. Doctor-Side Application

- Loads and displays the current queue from the same JSON file.
- Auto-sorts appointments based on time.
- Allows searching and filtering by patient name.
- Supports:
- **Mark Checked**
- **Mark Absent**
- **Reset Filters**
- Uses color-coding for status:
- Pending (Grey)
- Checked (Green)
- Absent (Red)
- **Auto-refresh** every 3 seconds to simulate real-time updates.

Technology Stack

- **Frontend & Logic:** Python 3, Tkinter
- **Local Data Store:** JSON (queue_data.json)
- **Platform Support:** Mac, Windows (Dark mode supported on Mac)

File Structure

```
PatientFlow/
│
├── patient_ui.py           # Patient-side GUI code
├── doctor_ui.py            # Doctor-side queue management code
├── queue_data.json         # Shared local data store for appointments
└── README.md               # Project overview and usage instructions
```

Future Enhancements

- **Firebase Integration:** Replace local JSON with real-time Firestore DB to support syncing across multiple devices.
- **Slot Customization:** Add the ability to configure slot duration and clinic timings.
- **Mobile App:** Flutter-based interface for broader accessibility.
- **Notifications:** Email or SMS reminders for upcoming appointments.

Conclusion

PatientFlow offers a practical and scalable foundation for digitising appointment workflows in under-resourced clinics. It enhances patient experience, reduces idle time for doctors, and lays the groundwork for a fully cloud-based healthcare system.
