# Full Stack Development with MERN

### 1. Introduction

Project Title: Seamless Health Appointment Booking

Team Members:  
- Lella Manoj

### 2. Project Overview

Purpose:  
The Seamless Health Appointment Booking application aims to simplify and streamline the process of booking medical appointments. The platform provides an easy-to-use interface for patients to book appointments with doctors based on their specialization, and view previously booked appointments.

Features:  
- Patient form for appointment booking  
- Doctor selection by specialization  
- Appointment date selection  
- Appointment records stored in CSV or database  
- View booked appointments  
- List of available doctors

### 3. Architecture

Frontend:  
- Built using Streamlit for simplicity and rapid prototyping  
- Responsive UI with sidebar navigation

Backend:  
- Uses Python and Streamlit for logic and rendering  
- Appointments stored locally in a CSV file (can be extended to use MongoDB/Firebase)

Database:  
- NoSQL or CSV format (initial prototype uses appointments.csv)

### 4. Setup Instructions

Prerequisites:  
- Python 3.8+  
- pip (Python package installer)  
- Streamlit  
- Pandas  
- pyngrok (if deployed via Google Colab)

Installation:  
pip install streamlit pandas pyngrok

For Google Colab Deployment:  
- Set up ngrok token using pyngrok  
- Use Colab code cell to write and run the app

### 5. Folder Structure

project/  
├── app.py # Main Streamlit application  
├── appointments.csv # Stores booking data  
└── create\_app.py # Script to generate app.py dynamically

### 6. Running the Application

Locally:  
streamlit run app.py

In Google Colab:  
Use !streamlit run app.py & and expose using ngrok

### 7. API Documentation

No REST API in current version. All actions are UI-triggered in Streamlit.

### 8. Authentication

- Currently, authentication is not implemented.  
- Future versions may include login/signup with Google or email-based access.

### 9. User Interface

- Sidebar navigation for:  
 - Booking an appointment  
 - Viewing booked appointments  
 - Viewing doctors list  
- Forms for input and CSV-based persistence

### 10. Testing

- Manual testing of form submission and CSV output  
- Validation through user interaction

### 11. Screenshots or Demo

[Add screenshots here after deployment on Streamlit or Colab]

### 12. Known Issues

- No database integration (CSV only)  
- No authentication or admin dashboard  
- Limited validation on form inputs

### 13. Future Enhancements

- Integrate Firebase or MongoDB for real-time backend  
- Add user authentication (JWT or OAuth)  
- Enable doctor login and availability management  
- Add email/SMS confirmation for bookings  
- Admin panel for doctors and appointments

14. Code

# Step 1: Install required libraries

!pip install streamlit pyngrok pandas --quiet

# Step 2: Replace below with your valid authtoken

NGROK\_AUTH\_TOKEN = "2z4tNmBCYkX2yOYLsdrWPYd3irt\_7LFxKL4f7e1RcSuByQM4J"

# Step 3: Set up ngrok with token

from pyngrok import ngrok, conf

conf.get\_default().auth\_token = NGROK\_AUTH\_TOKEN

# Save the Streamlit app code

app\_code = '''

import streamlit as st

import pandas as pd

from datetime import datetime

st.set\_page\_config(page\_title="DocSpot - Book Your Doctor", layout="centered")

st.title("🩺 DocSpot - Seamless Health Appointment Booking")

# Doctor list

doctors = [

    {"Doctor ID": 1, "Name": "Dr. Asha Reddy", "Specialization": "Cardiologist"},

    {"Doctor ID": 2, "Name": "Dr. Kiran Kumar", "Specialization": "Dermatologist"},

    {"Doctor ID": 3, "Name": "Dr. Meena Joshi", "Specialization": "Neurologist"}

]

doctors\_df = pd.DataFrame(doctors)

# Load appointments

@st.cache\_data

def load\_appointments():

    try:

        return pd.read\_csv("appointments.csv")

    except:

        return pd.DataFrame(columns=["Patient Name", "Contact", "Doctor Name", "Date", "Timestamp"])

appointments = load\_appointments()

# Sidebar Navigation

menu = st.sidebar.radio("Navigation", ["🏠 Book Appointment", "📋 View Appointments", "📚 Doctors List"])

if menu == "🏠 Book Appointment":

    st.header("Book a New Appointment")

    with st.form("booking\_form"):

        name = st.text\_input("👤 Patient Name")

        contact = st.text\_input("📞 Contact Number")

        doc\_selected = st.selectbox("🧑‍⚕️ Select Doctor", doctors\_df["Name"])

        date = st.date\_input("📅 Select Appointment Date")

        submit = st.form\_submit\_button("✅ Book Appointment")

        if submit:

            new\_appointment = {

                "Patient Name": name,

                "Contact": contact,

                "Doctor Name": doc\_selected,

                "Date": str(date),

                "Timestamp": str(datetime.now())

            }

            appointments = pd.concat([appointments, pd.DataFrame([new\_appointment])], ignore\_index=True)

            appointments.to\_csv("appointments.csv", index=False)

            st.success("✅ Appointment Booked Successfully!")

elif menu == "📋 View Appointments":

    st.header("Your Appointments")

    if appointments.empty:

        st.warning("No appointments booked yet.")

    else:

        st.dataframe(appointments)

elif menu == "📚 Doctors List":

    st.header("Available Doctors")

    st.dataframe(doctors\_df)

'''

# Save app.py

with open("app.py", "w") as f:

    f.write(app\_code)

# Run Streamlit app in background

!streamlit run app.py &>/content/log.txt &

# Wait for app to spin up

import time

time.sleep(5)

# Open tunnel

public\_url = ngrok.connect(8501)

print("🌐 Public URL to access your Streamlit app:", public\_url)

15 . Output link

🌐 Public URL to access your Streamlit app: NgrokTunnel: "[https://6e9e-34-21-115-33.ngrok-free.app](https://6e9e-34-21-115-33.ngrok-free.app/)"