🌟 Perfect Project Plan – AI Scheduling Agent (LangChain + LangGraph)

Phase 1: Setup & Planning

- **Objective**: Build an AI agent that simulates a medical receptionist.
- **Frameworks**: LangChain + LangGraph for multi-agent orchestration.
- **Tools**:
  - UI → Streamlit or Gradio
  - Data → Pandas (CSV/Excel)
  - Email → SMTP / SendGrid
  - SMS → Twilio
  - Calendar → Excel (simulate Calendly)

Phase 2: Mock Data Creation

1. **Patients Database (patients.csv)**
   - Columns:
   - PatientID, Name, DOB, Email, Phone, DoctorPreference, PatientType, InsuranceCarrier, MemberID, GroupNumber
   - Generate ~50 synthetic records.
2. **Doctor Schedules (availability.xlsx)**
   - Columns:
   - DoctorName, Date, TimeSlot, Status (Available/Booked)
   - Populate with synthetic availability (e.g., 9–5 slots).
3. **Appointment Templates**
   - Intake form (PDF/DOC) to email after confirmation.

Phase 3: Multi-Agent Workflow Design (LangGraph)

Agents & Responsibilities

1. **Patient Interaction Agent**

    - Greets patient, collects **Name, DOB, Email, Phone, Doctor preference**.

2. **Patient Lookup Agent**
    - Reads patients.csv.
    - Checks if patient exists → classify as **new (60 mins)** or **returning (30 mins)**.
3. **Scheduling Agent**
    - Reads availability.xlsx.
    - Applies duration rule (60/30).
    - Finds free slot and reserves it.
4. **Insurance Agent**
    - Collects **InsuranceCarrier, MemberID, GroupNumber**.
    - Saves in patient record.
5. **Confirmation Agent**
    - Writes appointment into appointments.xlsx.
    - Sends **confirmation email + SMS**.
6. **Form Distribution Agent**
    - Emails intake form **after confirmation**.
7. **Reminder Agent**
    - Sends **3 reminders** (Email + SMS):
        - 1st: Basic reminder.
        - 2nd: Ask about form completion + confirmation.
        - 3rd: Final check → confirm/cancel with reason.
    - Updates appointment record.
8. **Admin Agent**
    - Generates daily report (appointments.xlsx) with:
        - Patient details, doctor, time, insurance, confirmation status, reminders sent.

Phase 4: Integration Requirements

- **Patient Data** → CSV (simulate EMR).
- **Calendar** → Excel (availability + updates).
- **Communication** → Email (SMTP/SendGrid) + SMS (Twilio).
- **Data Export** → Excel reports for admin review.

Phase 5: Conversation & Edge Cases

- Conversation Flow:
  - Greeting → Info collection → Lookup → Scheduling → Insurance → Confirmation → Forms → Reminders.
- Edge Cases:
  - Patient not found → add as new.
  - Doctor fully booked → suggest another slot/doctor.
  - Patient cancels → free slot in calendar + record reason.
  - Patient doesn't fill form → send reminder escalation.

Phase 6: UI Development

- Build a **chatbot interface** using Streamlit/Gradio.
- Show real-time booking flow with status messages:
  - *"You are a new patient, booking a 60-min slot with Dr. Kumar at 3 PM."*
- Include an admin view for reports.

Phase 7: Testing

- Test scenarios:
  - New patient with valid insurance.
  - Returning patient.
  - Patient canceling appointment.
  - Missing insurance details.
  - Multiple reminders & confirmation tracking.

Phase 8: Deliverables

1. **Technical Approach (PDF)** – Architecture, framework choice, integrations, challenges.
2. **Demo Video (3–5 mins)** – End-to-end chatbot flow with booking → confirmation → reminders.
3. **Code Package (ZIP)** – Python source code, requirements.txt, mock data, setup guide.

Phase 9: Success Metrics

- End-to-end booking flow works.
- Patient classification (new/returning) accurate.
- Correct slot allocation.
- Email/SMS successfully sent.
- Excel exports generated cleanly.
- Clean, modular codebase.

✅ This plan ensures you cover **all requirements** (integration, mock data, reminders, export) while demonstrating **multi-agent orchestration with LangChain + LangGraph**.

Do you want me to now create a **detailed architecture diagram (system-level)** showing how LangGraph agents connect to CSV, Excel, Email, and SMS systems?