Implement API endpoints for an out-patient appointment system. Doctors practice on a weekly schedule. For simplicity, let's assume they practice on only one location, only on evenings, may consult X number of patients and always leave on Sundays. API implementation is expected for doctors listing, doctor detail page and appointment booking.

STEP:01

pip install Flask

STEP:02

Create a Python file (e.g. **application.py**) and set up your Flask app:

```
from flask import Flask

app = Flask(__name__)

if __name__ == "__main__":
    app.run(debug=True)
```

STEP:03

Create Dummy Data:

You'll need some dummy data to represent doctors, their schedules, and appointments. For simplicity, you can use a Python dictionary to store this data.

```
doctors = [
    {
        "id": 1,
        "name": "Dr. John Smith",
        "schedule": ["Monday", "Wednesday", "Friday"],
```

```
        "max_patients": 5,
    },
    {
        "id": 2,
        "name": "Dr. Jane Doe",
        "schedule": ["Tuesday", "Thursday"],
        "max_patients": 4,
    },
]

appointments = []

# Function to get available appointment slots for a doctor
def get_available_slots(doctor_id):
    # Implement logic to calculate available slots based on the doctor's schedule and existing appointments
    pass
```

STEP:04

Define API Endpoints:

Define the API endpoints for doctors listing, doctor detail page, and appointment booking using Flask's route decorators.

```
from flask import jsonify

@app.route("/doctors", methods=["GET"])
def get_doctors():
    # Return a list of all doctors
    return jsonify(doctors)

@app.route("/doctors/<int:doctor_id>", methods=["GET"])
def get_doctor(doctor_id):
    # Return details of a specific doctor
    # You can use doctor_id to filter the doctor from the 'doctors' list
```

```
    pass

@app.route("/appointments", methods=["POST"])
def book_appointment():
    # Implement logic to book an appointment
    pass
```

STEP:05

Implement Booking Logic

In the **book_appointment** function, you need to implement the logic for booking an appointment. This includes checking if the doctor is available on the selected date and time, and if there are available slots.

Error Handling and Validation

Implement error handling and validation for your API endpoints to handle cases where the user provides invalid data or the system encounters an error.

Testing and Documentation

Test your API endpoints thoroughly to ensure they work as expected. You can use tools like **curl** or Postman for testing. Additionally, create documentation for your API endpoints so that others can understand how to use them.