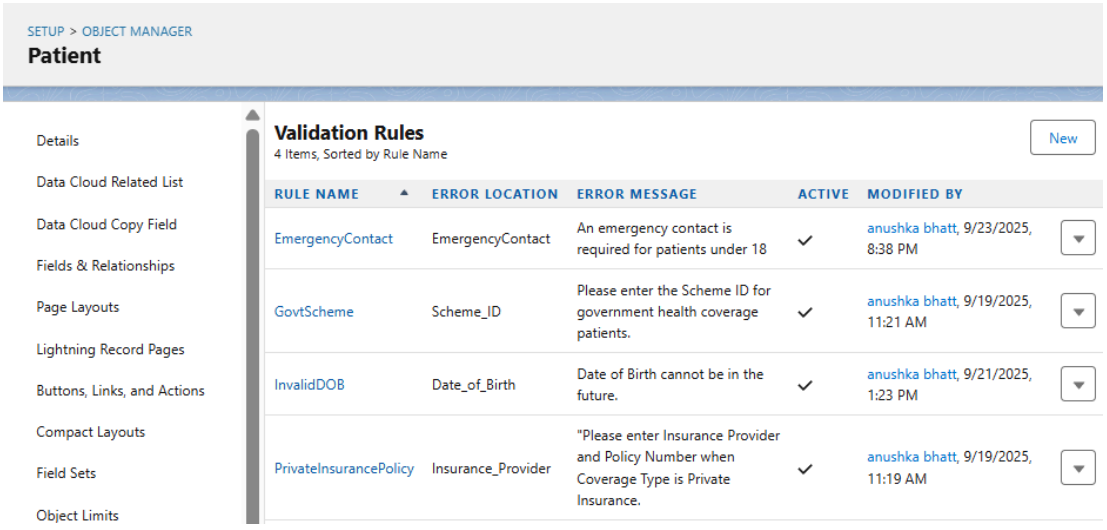
PHASE-4

Validation Rules

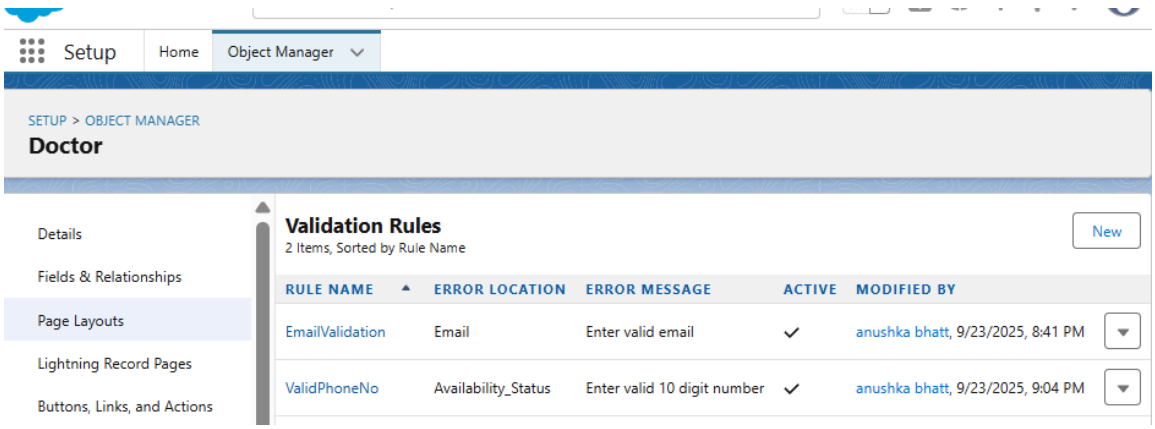
Object: Patient(Contact)

Name	Purpose	Formula
PrivateInsuranc ePolicy	If coverage_Type is private Insurance user needs to provide the Provider details and policy number	AND(ISPICKVAL(Coverage_Type__c, "Private Insuranc OR(ISBLANK(Insurance_Provider__c), ISBLANK(Insurance_Policy_Number__c)))
GovtScheme	If the patient's coverage_type is a government scheme like CGHS, ECHS, etc, provide schemeID	AND(OR(ISPICKVAL(Coverage_Type__c, "CGHS"), ISPICKVAL(Coverage_Type__c, "ECHS"), ISPICKVAL(Coverage_Type__c, "ESIC"), ISPICKVAL(Coverage_Type__c, "State Health Scheme")), ISBLANK(Scheme_ID__c))
InvalidDOB	Date of Birth can’t be in the future	Date_of_Birth >TODAY()
EmergencyCont act	If the patient is a minor Emergency Contact is a must	(TODAY() - Date_of_Birth__c) / 365 < 18 && ISBLANK(EmergencyContact__c)



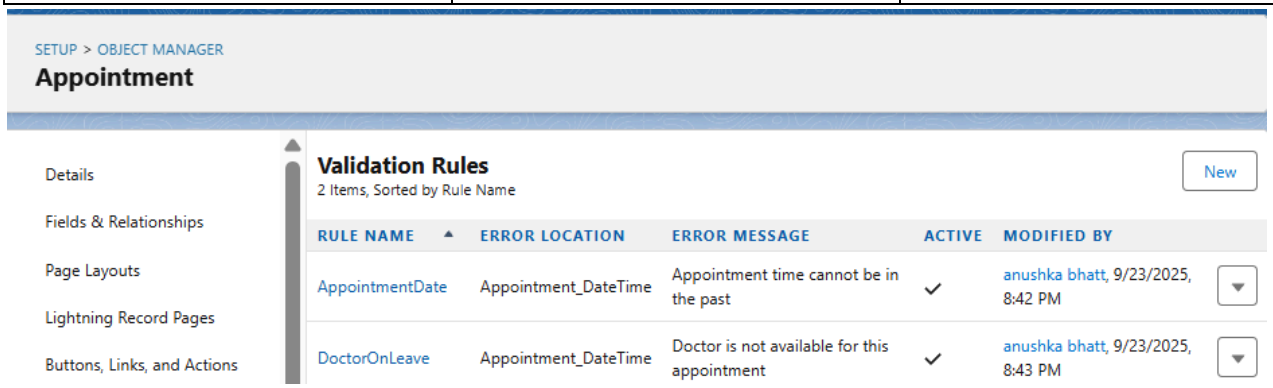
Object: Doctor__c

Name	Purpose	Formula
EmailValidation	Email should be valid	NOT(REGEX(Email__c , "^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,}$"))
ValidPhoneNo	Phone must contain 10 digits	NOT(REGEX(Phone__c , "^[0-9]{10}$"))



Object: Appointment__c

Name	Purpose	Formula
AppoinmentDate	It can't be of past	Appointment_DateTime__c < NOW()
DoctorOnLeave	An appointment can't be made for a date on which the Doctor is on leave	ISPICKVAL(Doctor__r.Availability_Status__c, "On Leave")



Approval Process:

System contains custom object Treatment__c , approval process if based on it .

Entry Criteria

If EstimatedCost__c for treatment > 0

Conditions:

If estimatedCost is less than 50,000 → Automatically approve

Else: need Hospital Manager approval

Step2:

If estimated cost is greater than 100000 → need Director Approval

SETUP

Approval Processes

Process Name	TreatmentCost	Active	✓
Unique Name	TreatmentCost	Next Automated Approver Determined By	Manager of Record Submitter
Description			
Entry Criteria	`Treatment: EstimatedCost GREATER THAN 0`		
Record Editability	Administrator **ONLY**	Allow Submitters to Recall Approval Requests	✓
Approval Assignment Email Template	CostApproval		
Initial Submitters	Appointment Owner, User: Henry William, User: Kevin Joshep, User: Sarah Roger, User: Tony Stark		
Created By	anushka bhatt, 9/24/2025, 9:42 AM	Modified By	anushka bhatt, 9/24/2025, 9:56 AM

Initial Submission Actions

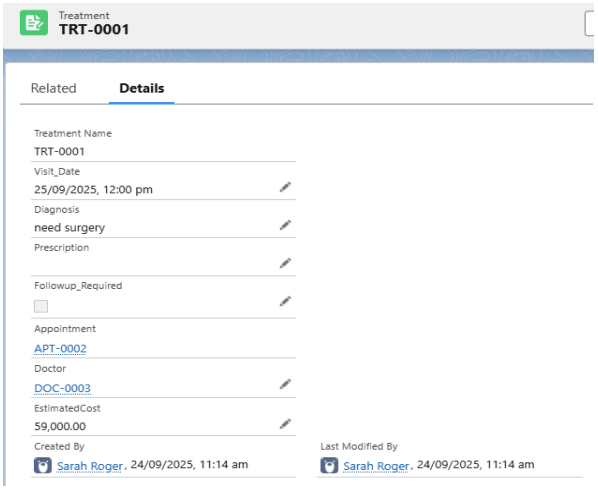
Add Existing | Add New

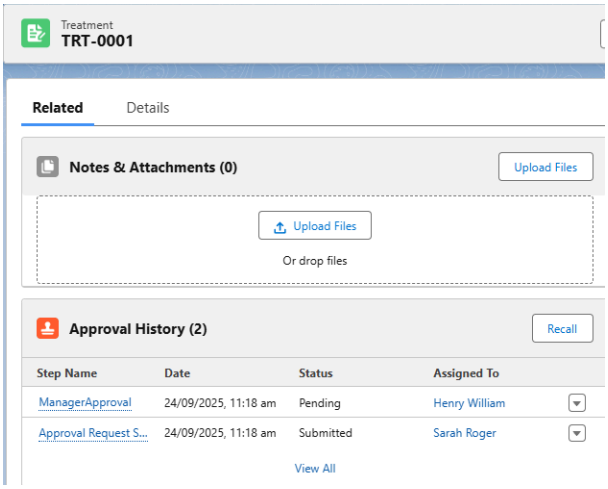
Action	Type	Description
	Record Lock	Lock the record from being edited

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions | Edit	1	ManagerApproval		`Treatment: EstimatedCost` GREATER THAN 50000 , **else Approve**	User:Henry William	Final Rejection
Show Actions | Edit	2	DirectorApproval		`Treatment: EstimatedCost` GREATER THAN 100000	**Approval based on first response** User: anushka bhatt, Cassie Thompson	Go Back 1 Step

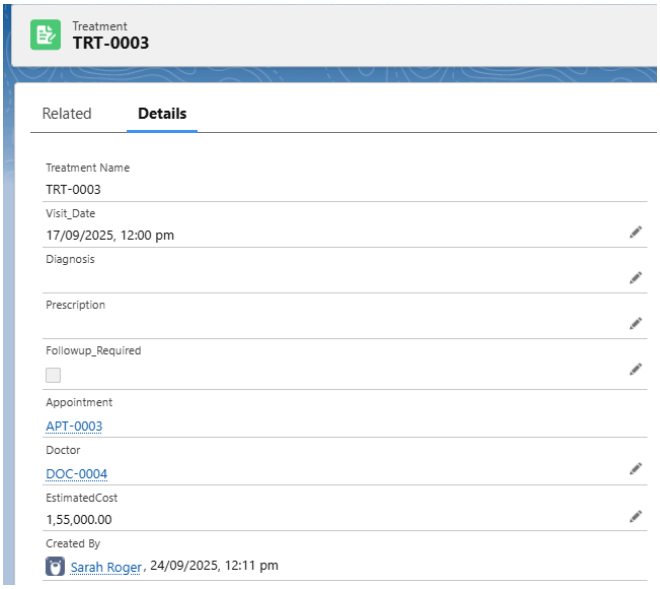
Care Team can submit the approval request





If the manager agree to approve he /she approves

If Cost is > 100000 goes to the director for approval



Agent using agentforce

The goal of this Agent is to streamline patient registration and doctor appointment booking in the **CuraForce App**.

- Patients or staff can interact with the chatbot to **create a new patient record**.
- The chatbot will ask for basic patient details (name, age, gender, health issue).
- Based on the **health issue description**, the bot uses an **Apex-powered NLP-like keyword matcher** to **suggest a suitable doctor specialty** (e.g., Cardiologist for chest pain, Dermatologist for skin issues, General Physician for other conditions).
- The chatbot then shows a list of **available doctors** in that specialty.
- If the patient confirms, the chatbot books an **appointment** (DoctorPatient record) with the selected doctor.

This automation reduces manual data entry for staff and provides patients with quick doctor recommendations.

Current Implementation (Prototype Stage)

Apex Class for Doctor Suggestion

Implemented HealthIssueDoctorService class which:

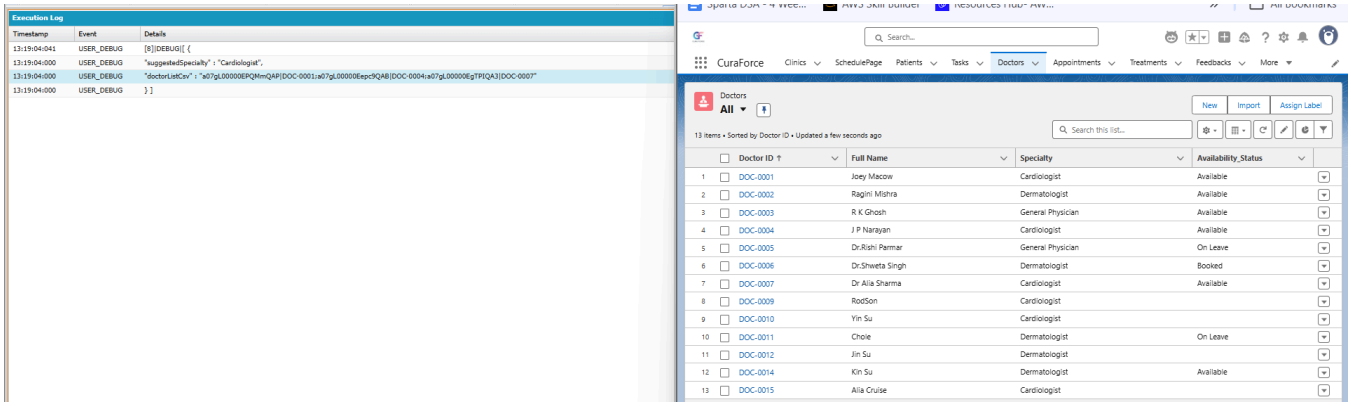
- Reads patient health issue text.
- Maps keywords → suggested specialty.
  Queries doctors with Availability__c = 'Available'.
- Returns a formatted list of doctor names & IDs.

```
Coverage: None | API Version: 64
public with sharing class HealthIssueDoctorService {
    // Invocable request (input)
    public class Request {
        @InvocableVariable(required=true)
        public String healthIssue;
        @InvocableVariable
        public Integer maxResults; // number of doctorsreturn
    }

    public class Response {
        @InvocableVariable
        public String suggestedSpecialty;
        // doctorListCsv is "Id|Doctor Name;Id|Doctor Name;..."
        @InvocableVariable
        public String doctorListCsv;
    }

    @InvocableMethod(label='Suggest Specialty & Doctors' description='Return suggested specialty and small list of doctors based on health issue text')
    public static List<Response> suggest(List<Request> requests) {
        List<Response> outputs = new List<Response>();
        for(Request req : requests) {
            String issue = (req.healthIssue == null) ? '' : req.healthIssue.toLowerCase();
            Integer maxR = (req.maxResults == null || req.maxResults <= 0) ? 5 : req.maxResults;
            String spec = 'General Physician';

            // keyword rules
            if(issue.contains('chest') || issue.contains('heart') || issue.contains('angina') || issue.contains('palpit')) {
                spec = 'Cardiologist';
            } else if(issue.contains('skin') || issue.contains('rash') || issue.contains('itch') || issue.contains('eczema') || issue.contains('psoriasis')) {
                spec = 'Dermatologist';
            } else {
                spec = 'General Physician';
            }

            // Querying doctors for that specialty
            List<Doctor__c> doctors = new List<Doctor__c>();
            try {
                doctors = [
                    SELECT Id, Name, Specialty__c,Availability_Status__c
                    FROM Doctor__c
                    WHERE Specialty__c = :spec
                    AND Availability_Status__c = 'Available'
                    LIMIT :maxR
                ];
            } catch (Exception e) {
                // if the Doctor__c object/fields not matched, fallback to empty list
                doctors = new List<Doctor__c>();
            }


            String csv = '';
            for(Doctor__c d : doctors) {
                if(csv != '') csv += ';';
                csv += String.valueOf(d.Id) + '|' + (d.Name == null ? '' : d.Name);
            }

            Response out = new Response();
            out.suggestedSpecialty = spec;
            out.doctorListCsv = csv;
            outputs.add(out);
        }
        return outputs;
    }
}
```

Tested apex class using executing this in anaonymous window

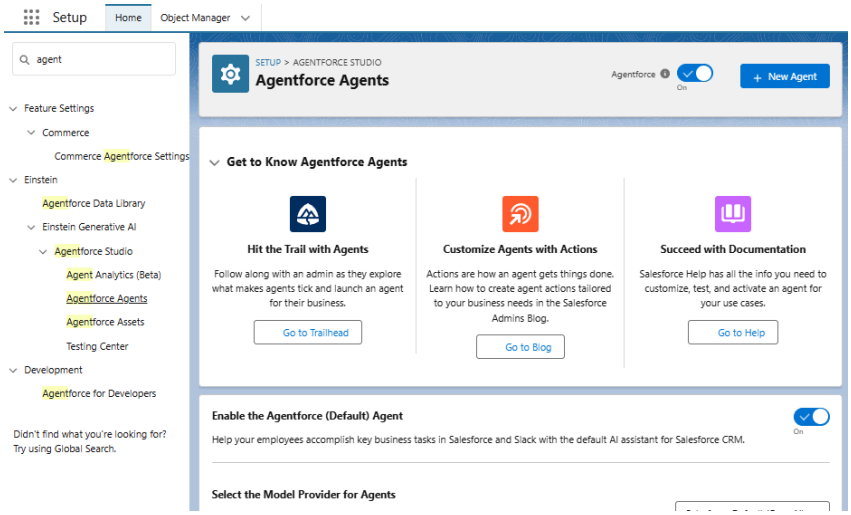
```
List<HealthIssueDoctorService.Request> reqs = new List<HealthIssueDoctorService.Request>();
HealthIssueDoctorService.Request r = new HealthIssueDoctorService.Request();
r.healthIssue = 'I have chest pain and difficulty breathing';
r.maxResults = 3;
reqs.add(r);

List<HealthIssueDoctorService.Response> res = HealthIssueDoctorService.suggest(reqs);
System.debug(JSON.serializePretty(res));
```

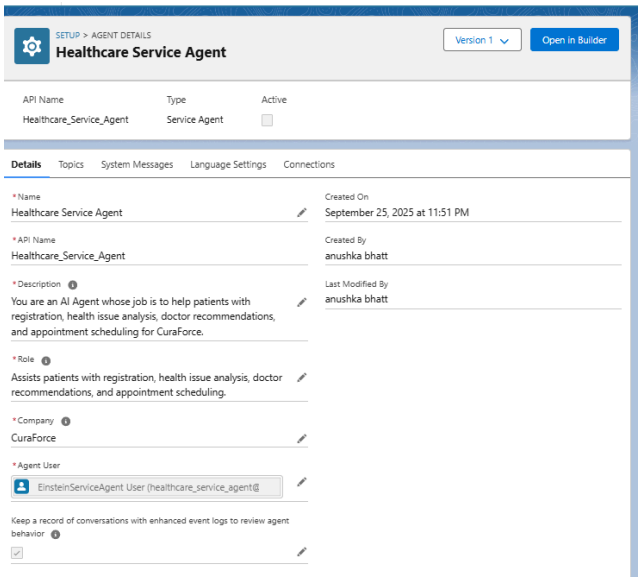
Result return Doc 004 and 007are cardiologist that are available.



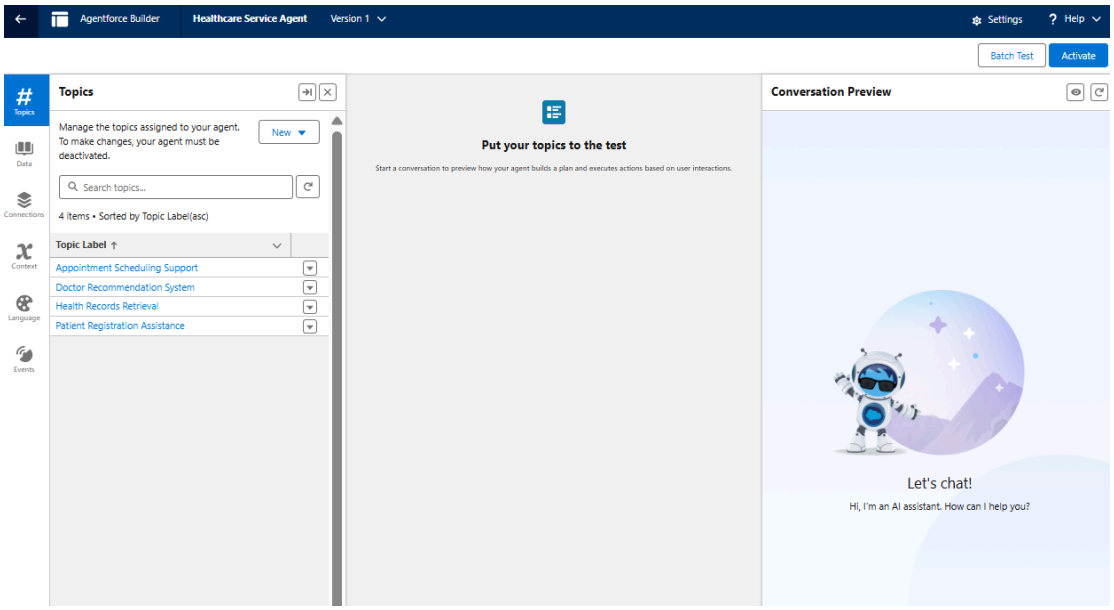
Steps to create Agent

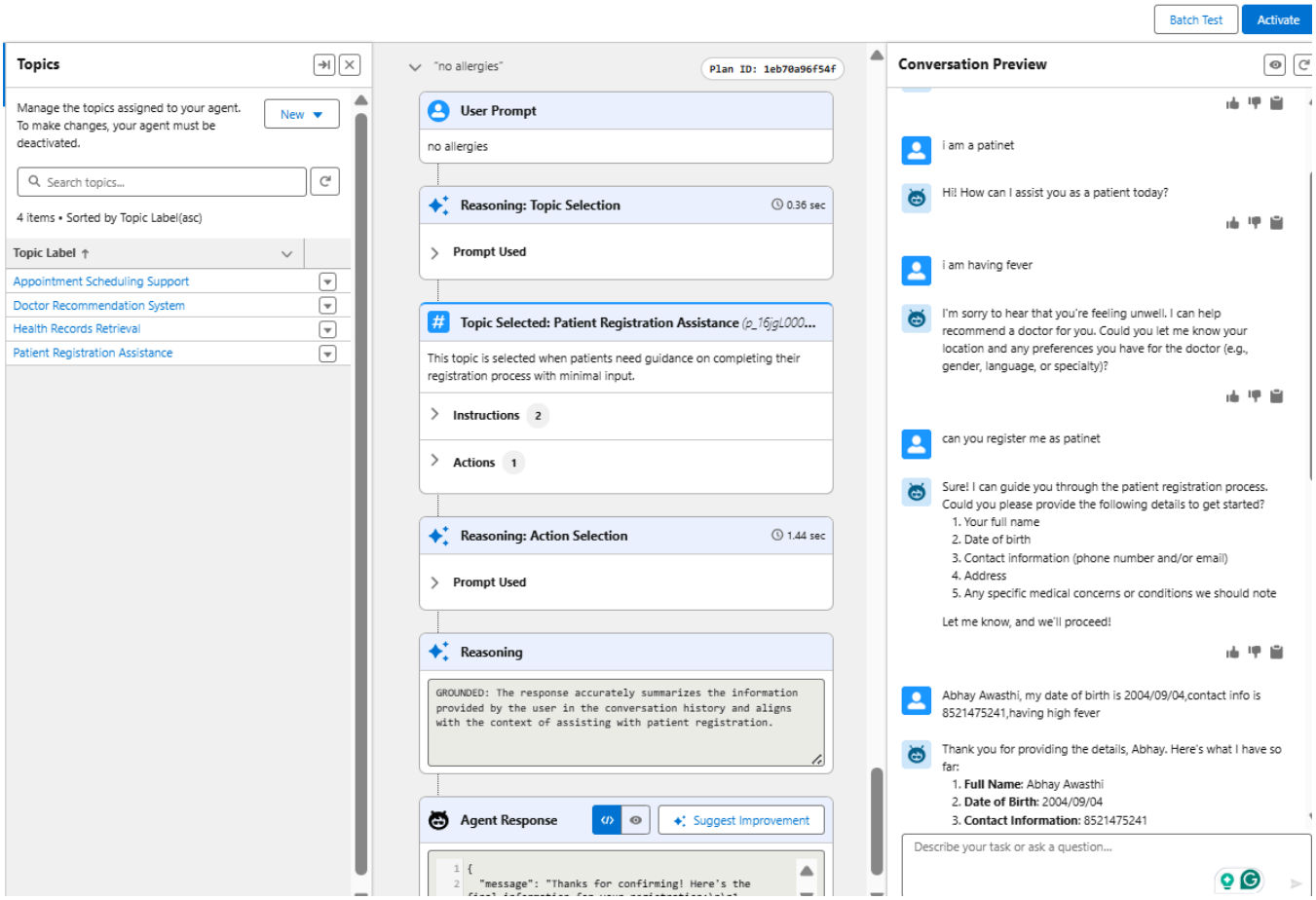
I already have all my agent permissions active so I directly head to creating new agent.

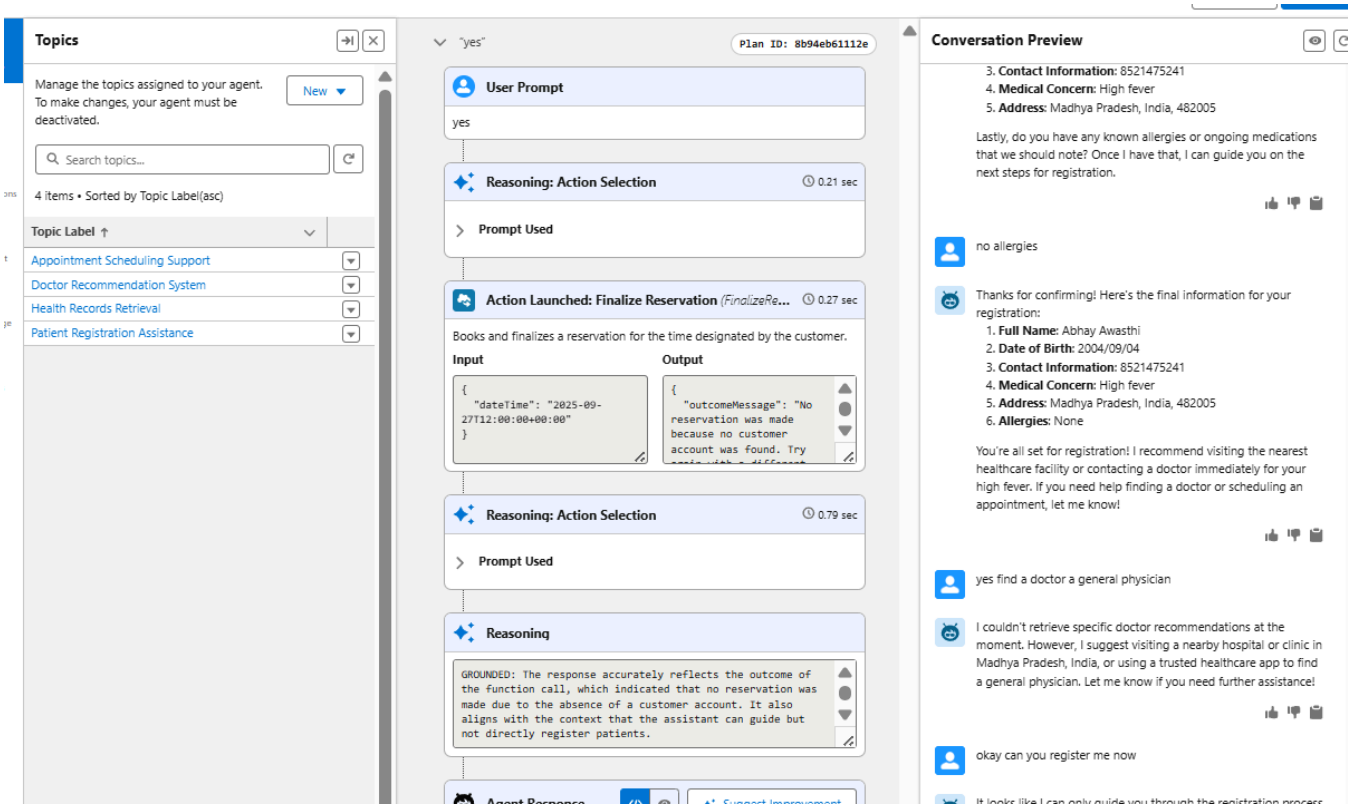




Set the topics for agent so that it give relevant answers







The agent is asking me as a patient to give information for required fields so that can create my patientaccount or check if i am already there. But I am working on functionality to book appointment and suggeting doctor.Right now the agent is under development.