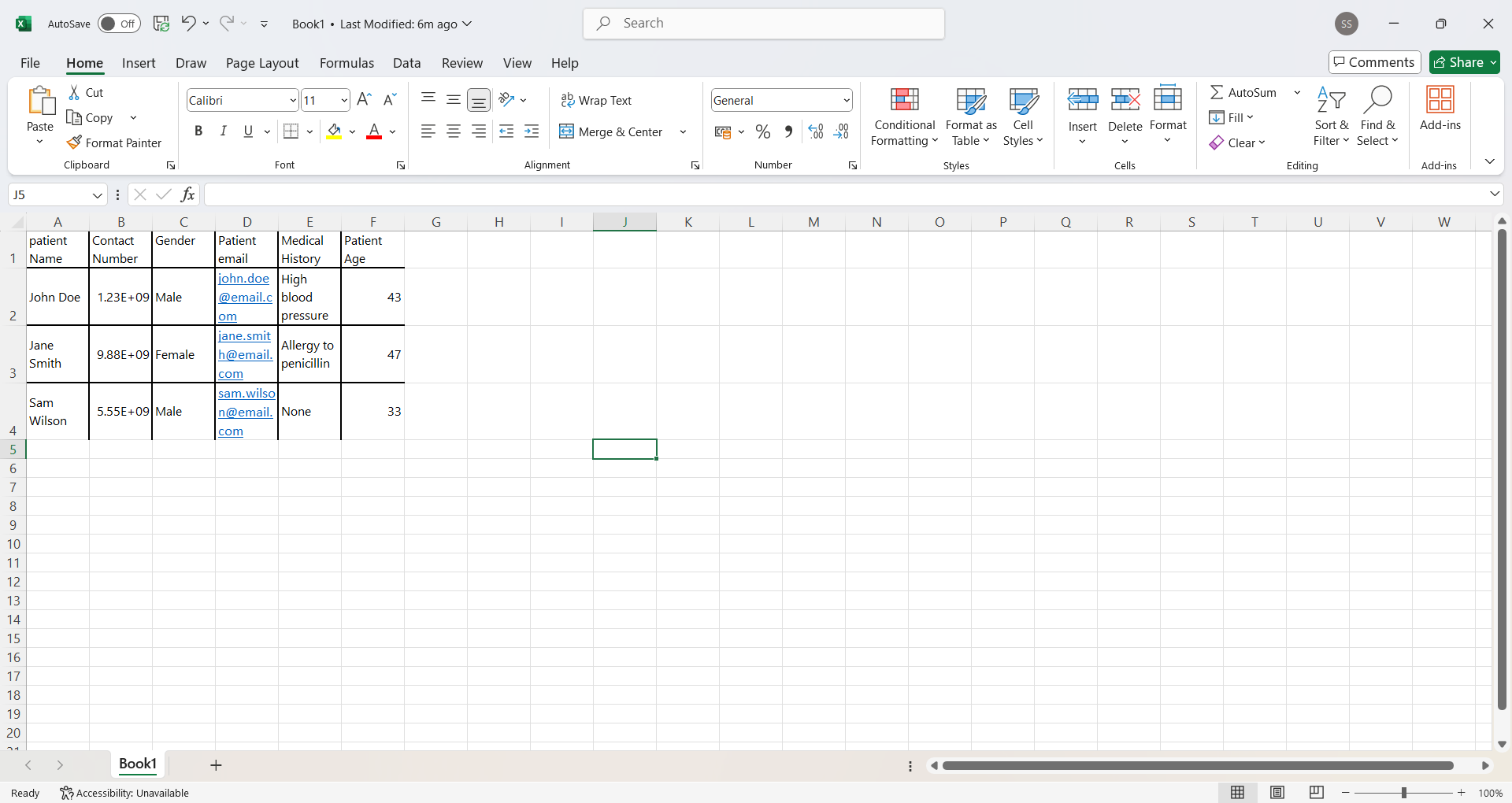
Phase 8: Data Management & Deployment:

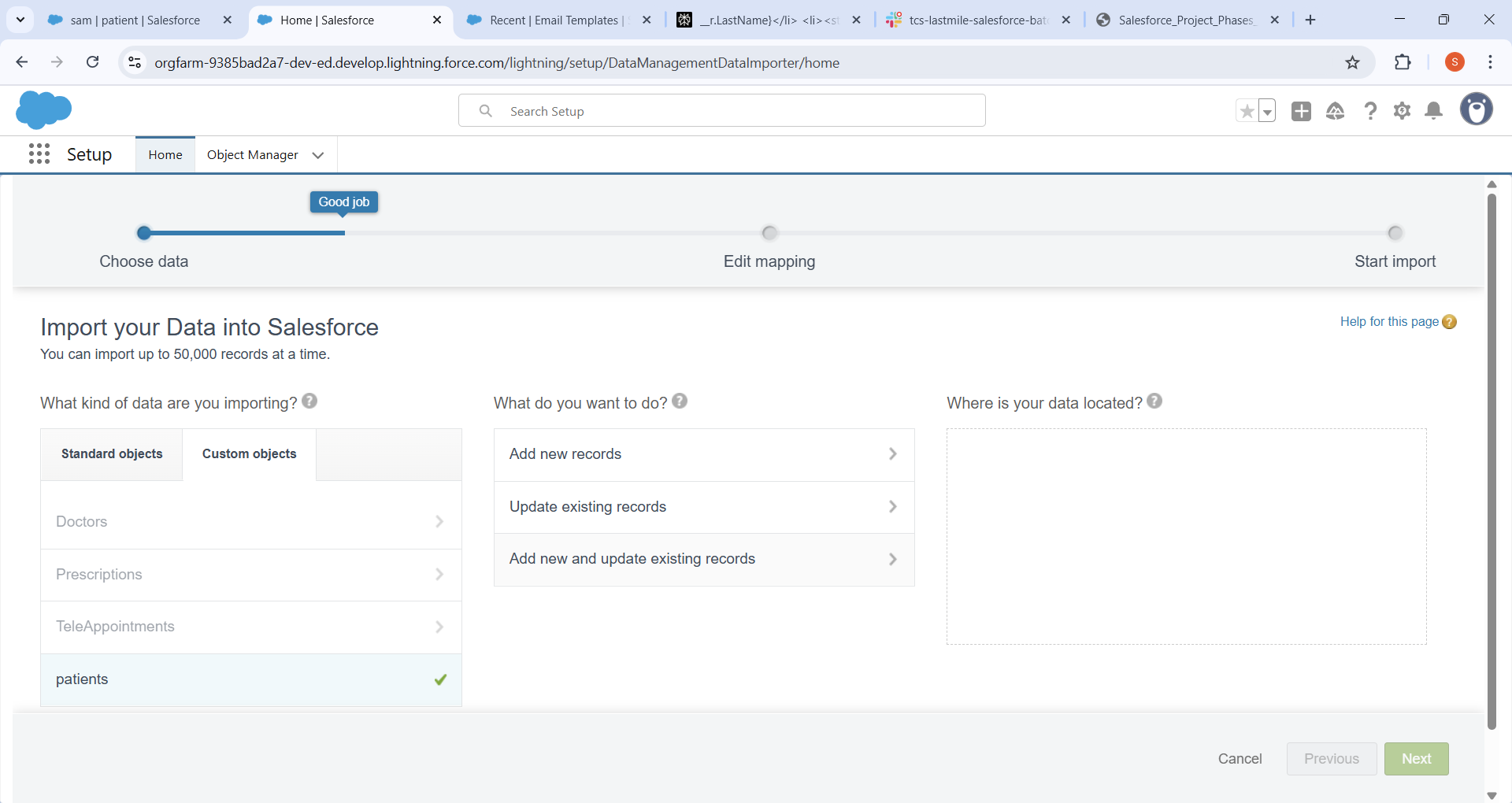
**1.Data Import Wizard:**

1. Created the Data of a Patient in the Word and save it in the .csv

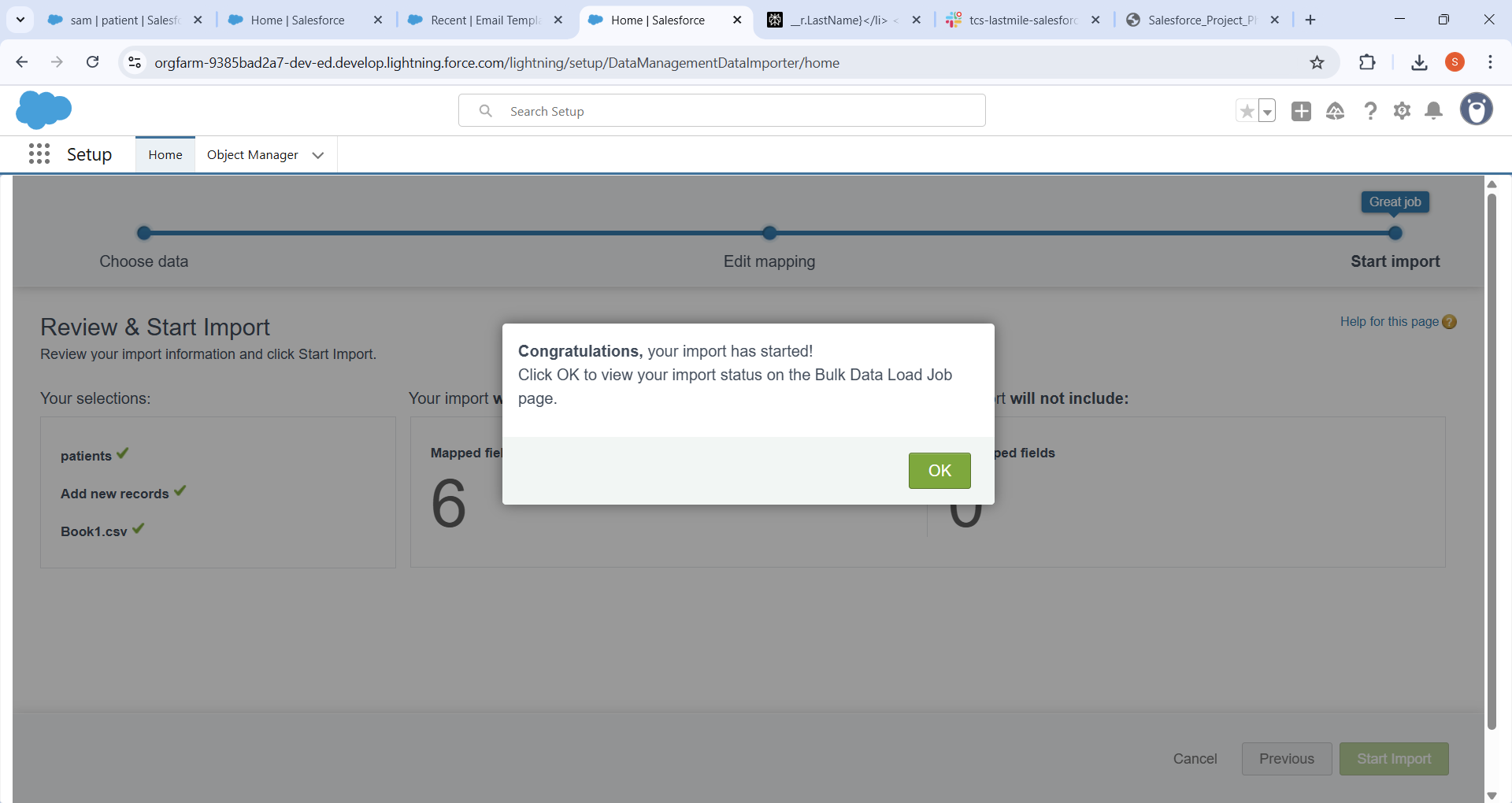
****

1. Create and Updating the setting in the Import Data

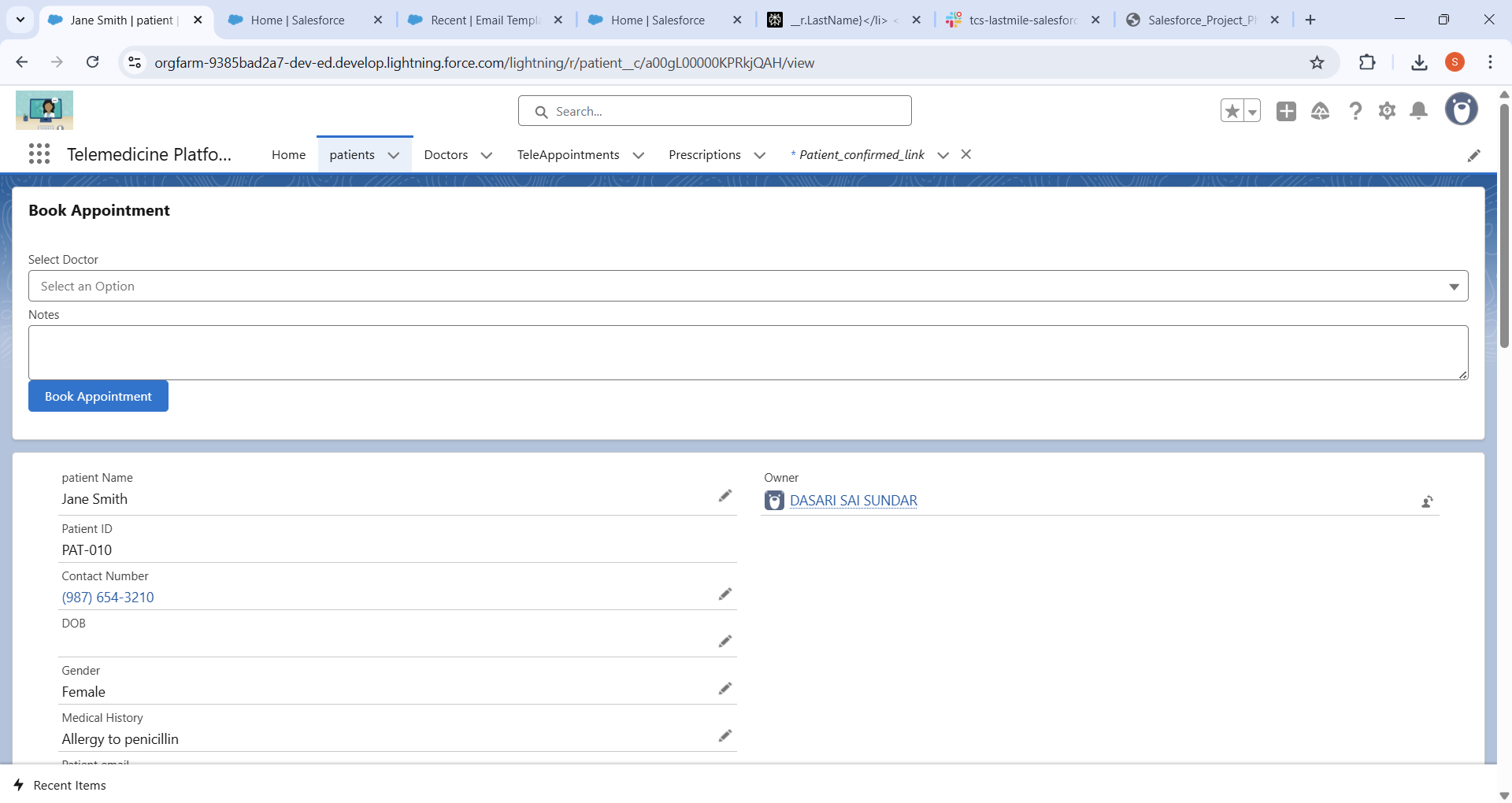
* With select the Patient as the source object🡪Locate our .csv

****

1. Successfully saved the Import function

****

1. Test1-Success The data Fetched to the Telemedical platform to the Patient

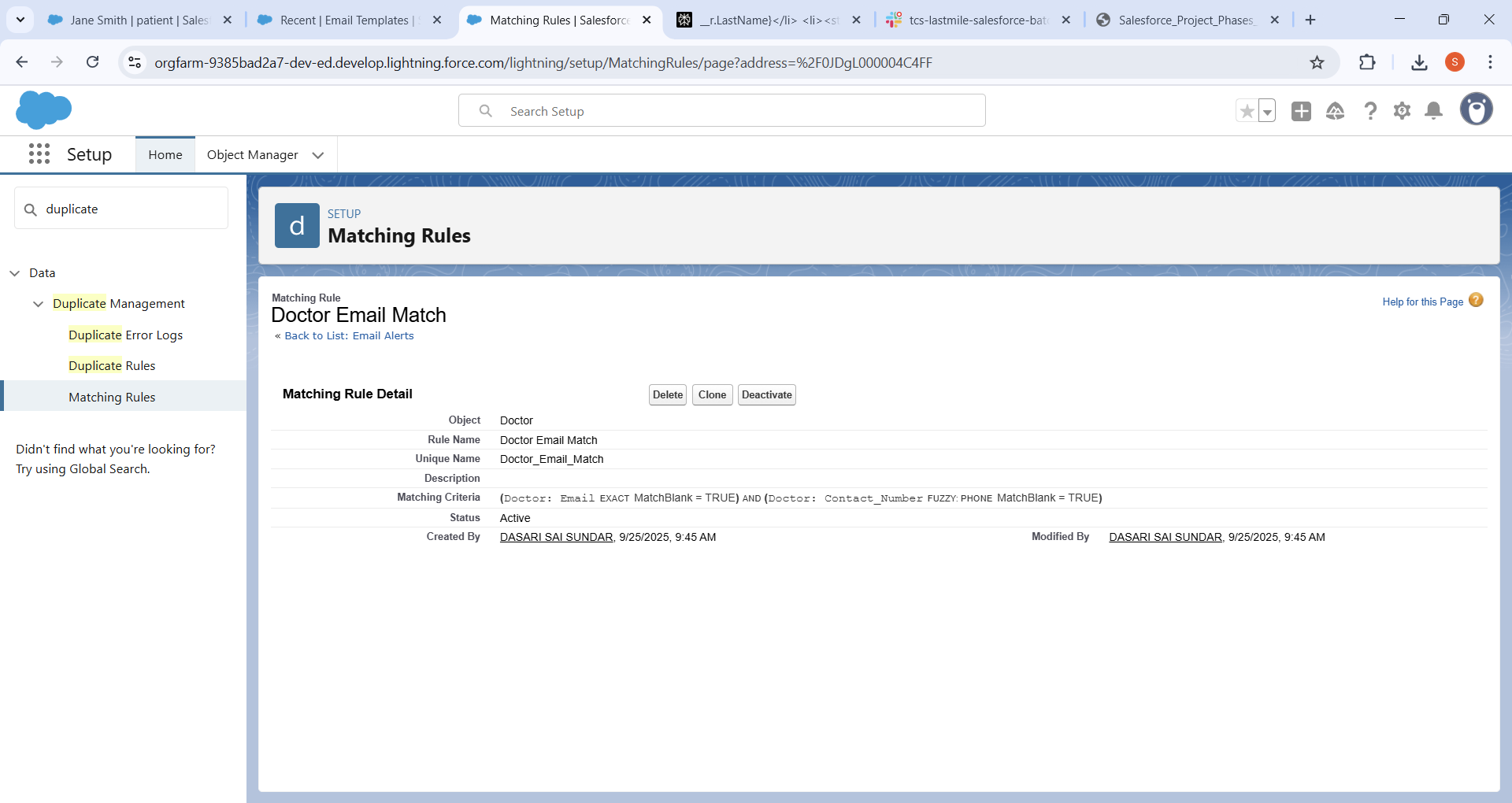


* Used Salesforce Data Import Wizard to upload patient information like names, contact numbers, date of birth, gender, email, medical history, and patient age from a well-prepared CSV file.
* Mapped CSV columns to corresponding Salesforce Patient fields to ensure correct data placement.
* Imported data as new patient records while preventing duplicates using matching and duplicate rules based on unique patient identifiers such as email and contact number.
* Validated the imported data by reviewing patient records for completeness and accuracy within Salesforce.
* Leveraged the imported patient data in appointment scheduling, teleconsultation workflows, and reporting modules to enable efficient telemedicine operations.

**2.** **Duplicate Rules:**

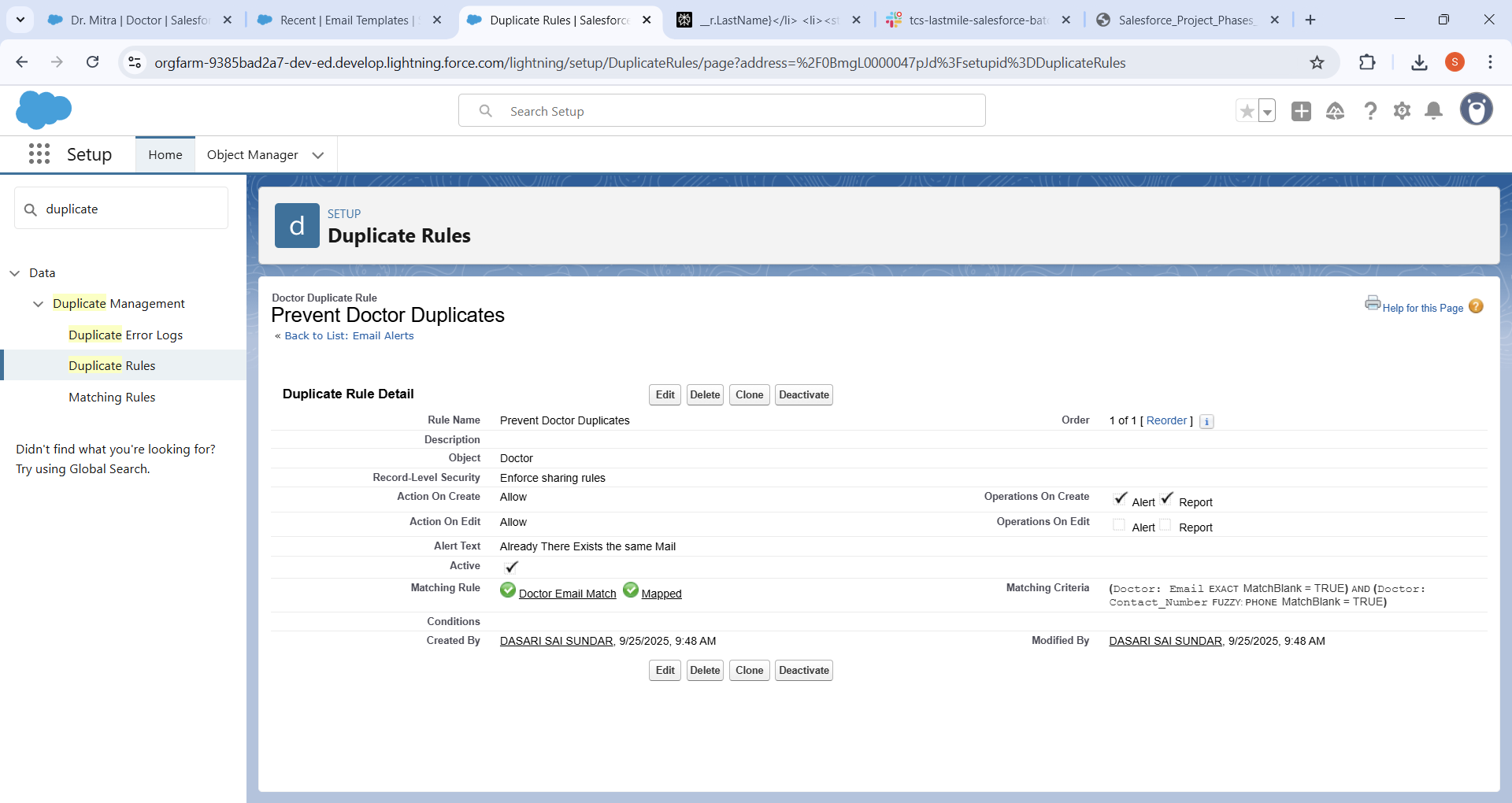
1. Matching Rules:

Created the Matching rule with Email.

****

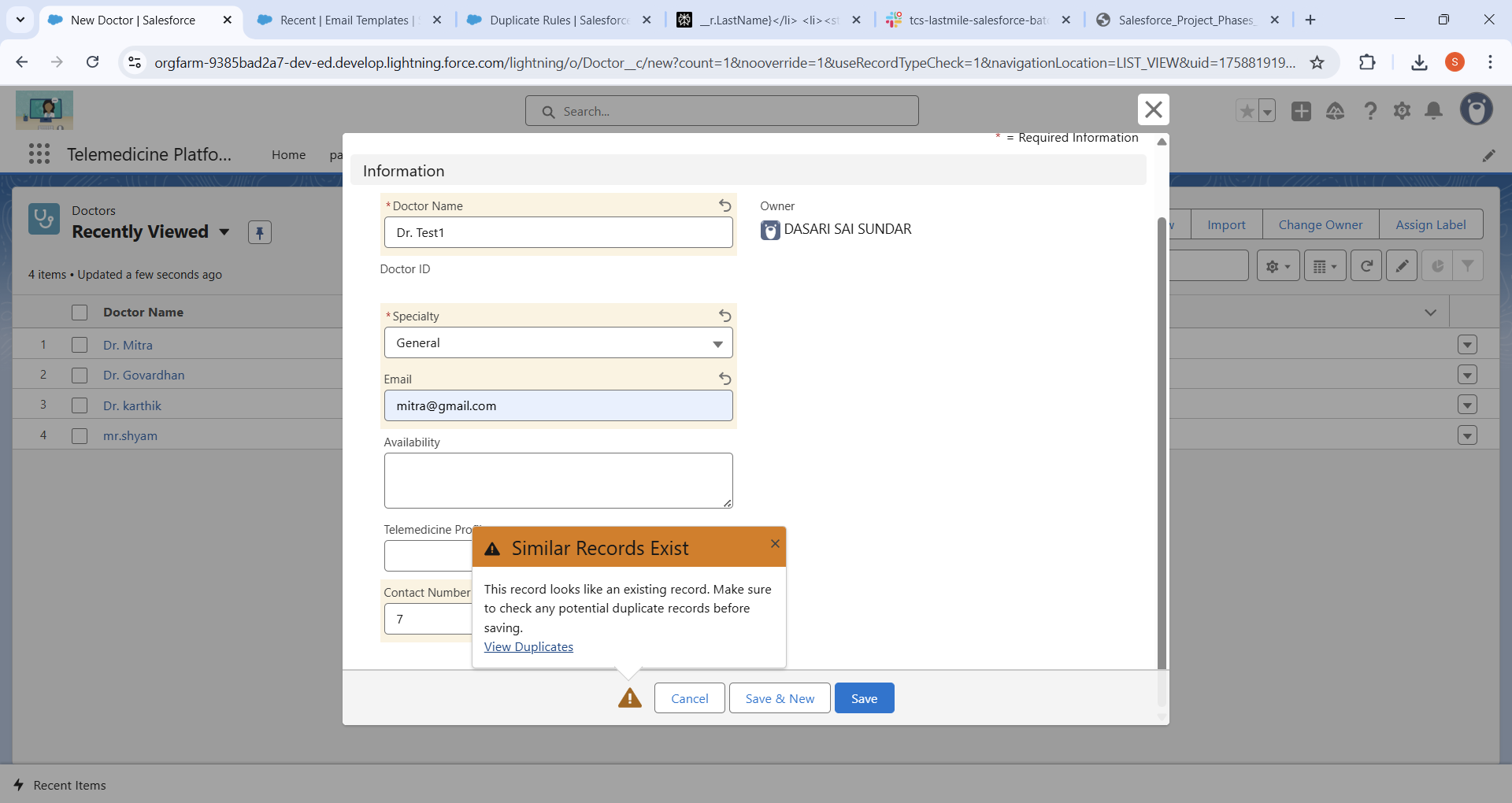
1. Duplicates Rules:

Created the duplicate rule with the above Matching rule.

****

1. Test case1:

Tested with same email but found The Error Arises same record Exists

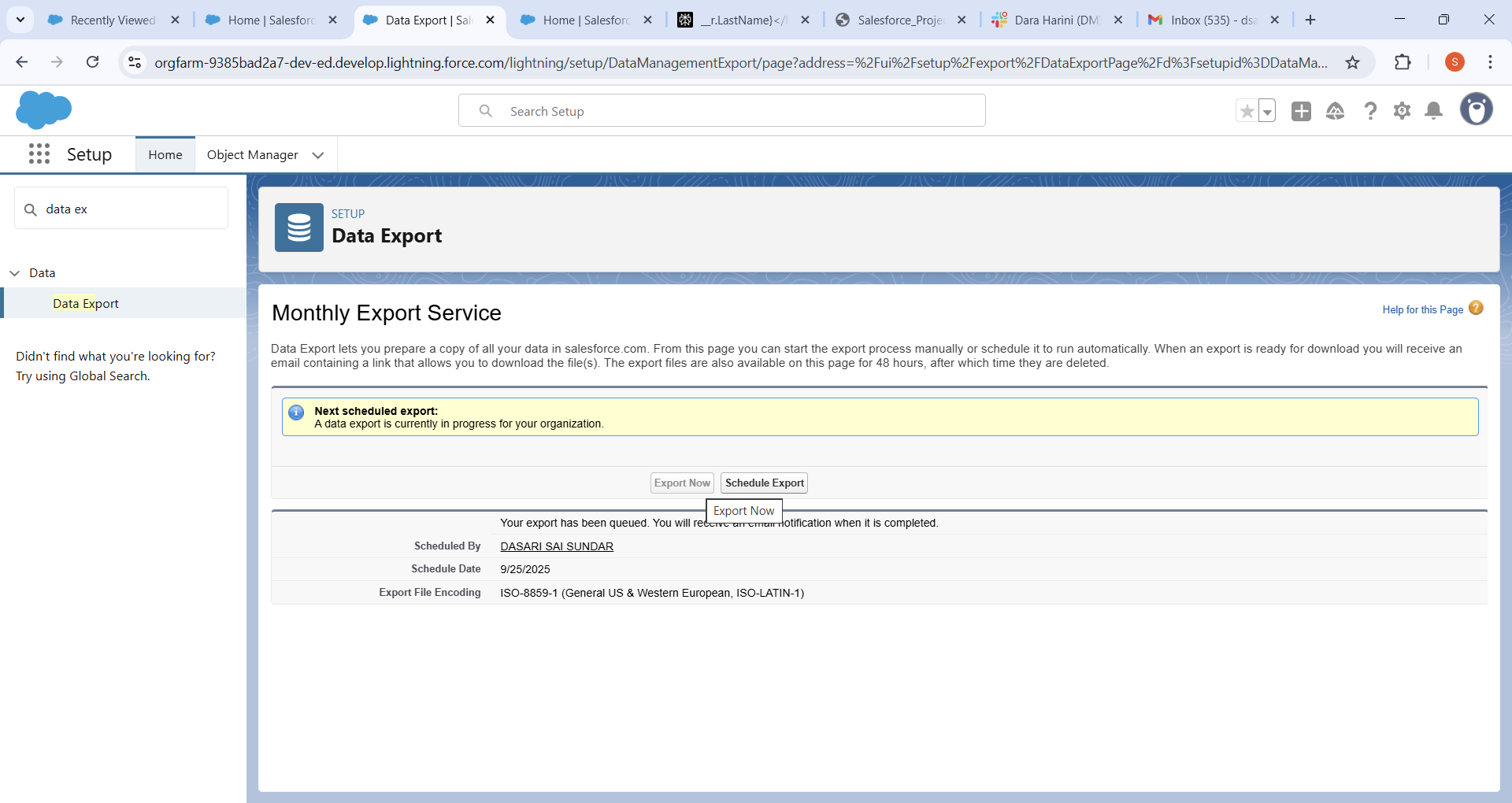


 Duplicate Management in Salesforce is vital to ensure patient and doctor data remains accurate and reliable.

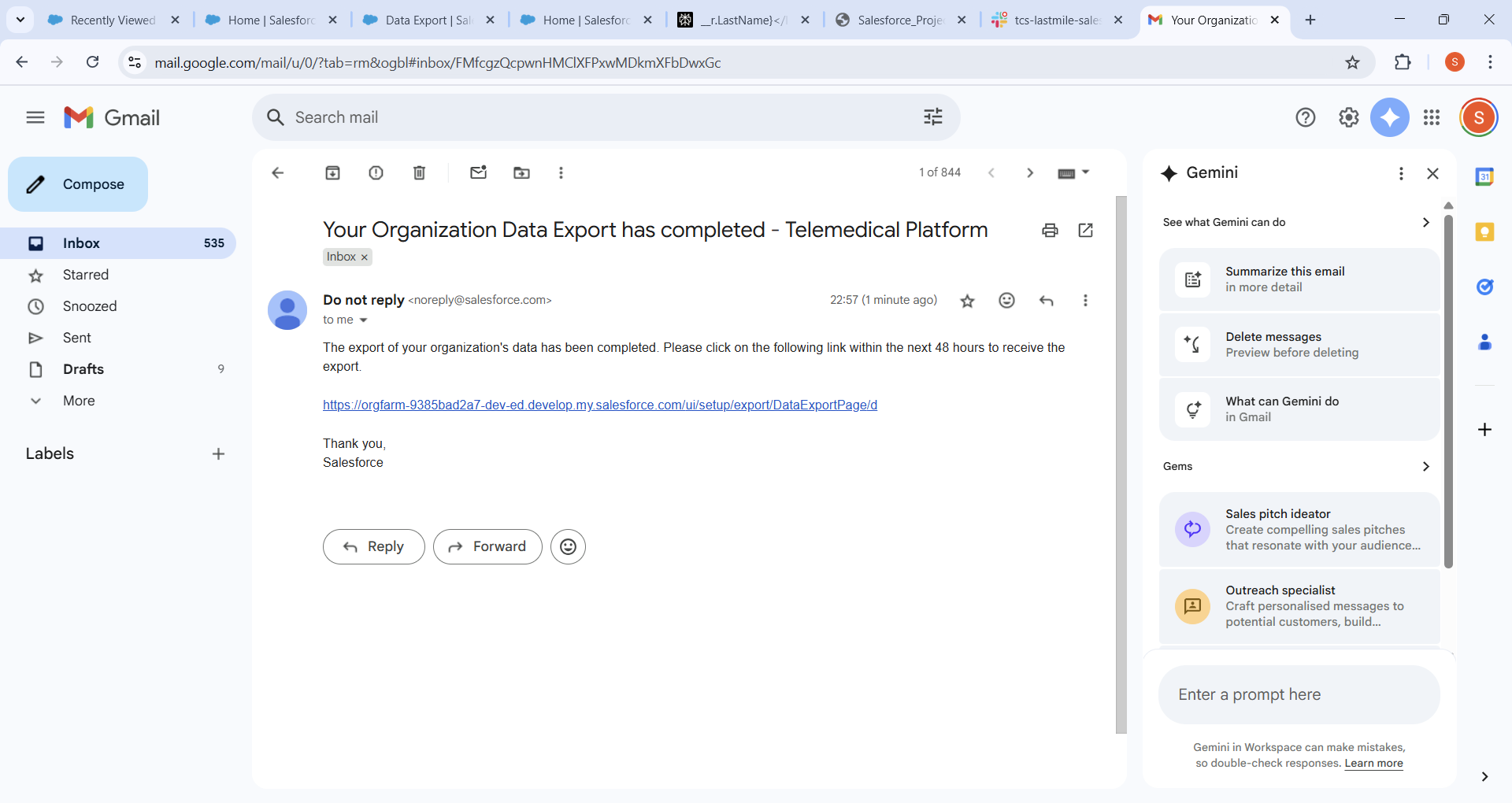
* Matching Rules can be defined to identify duplicates based on patient identifiers such as email, phone number, patient ID, as well as for doctors using license numbers or email addresses. This helps flag potential duplicate records when users enter or import data.
* Duplicate Rules enforce policies to either block duplicate patient or doctor record creation, alert users for manual review, or allow duplicates but log them for audit. This prevents issues like double bookings, duplicate billing, or inconsistent patient histories, ensuring smooth operation of telemedicine consultations and trustworthy data for healthcare decisions.

**3.Data Export:**

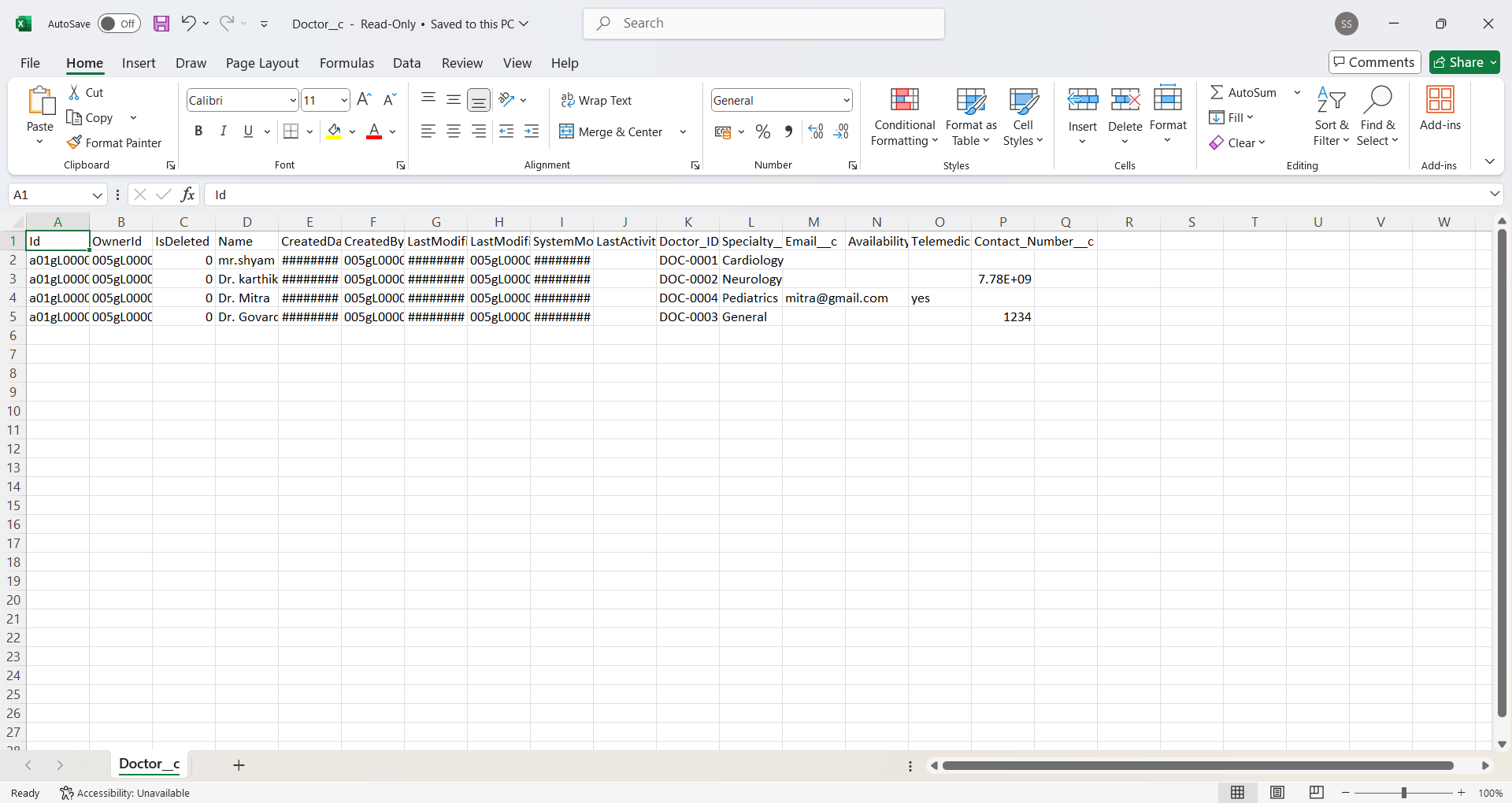
1. Add the object what to be export

****

1. Got a mail with download link

****

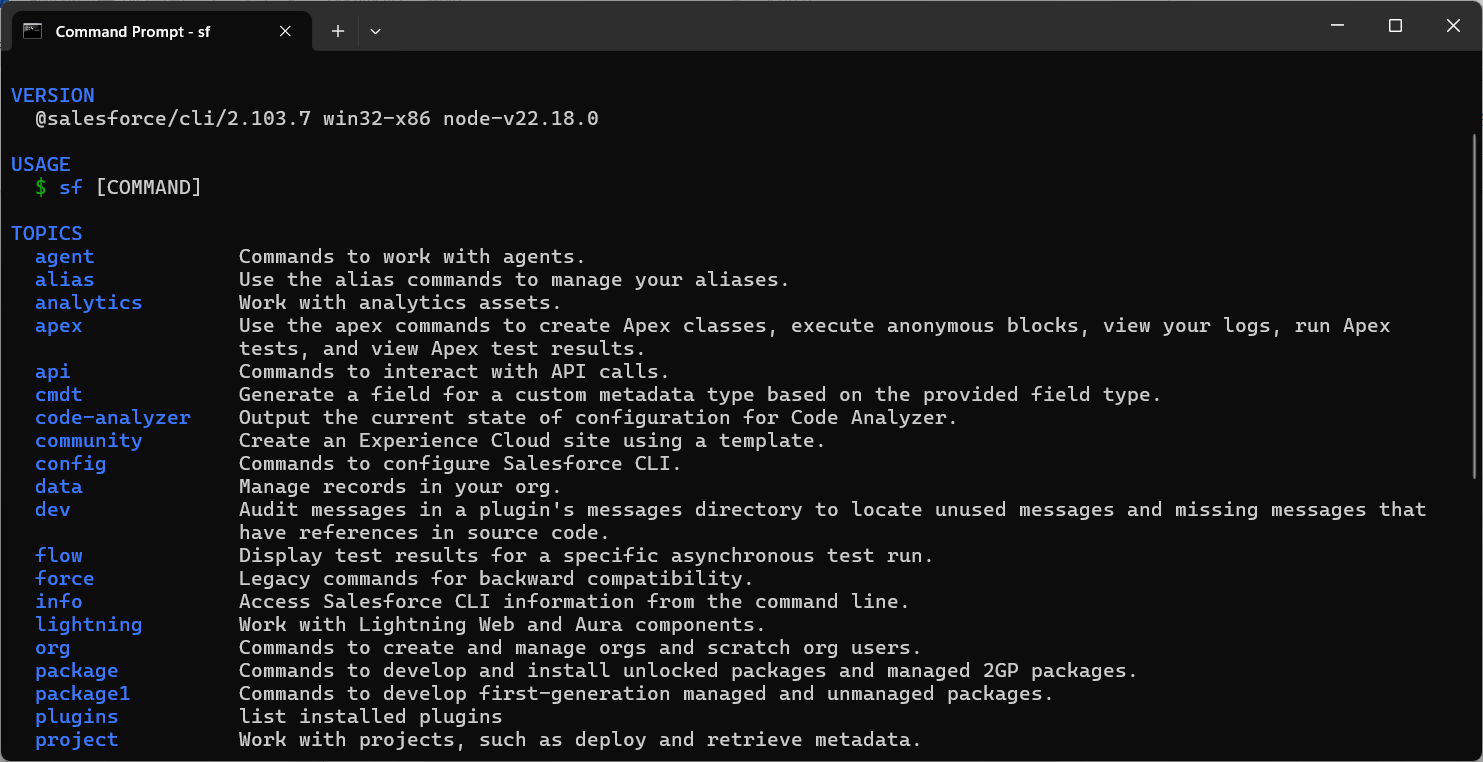
1. Downloaded the zip file and open the data in the excel



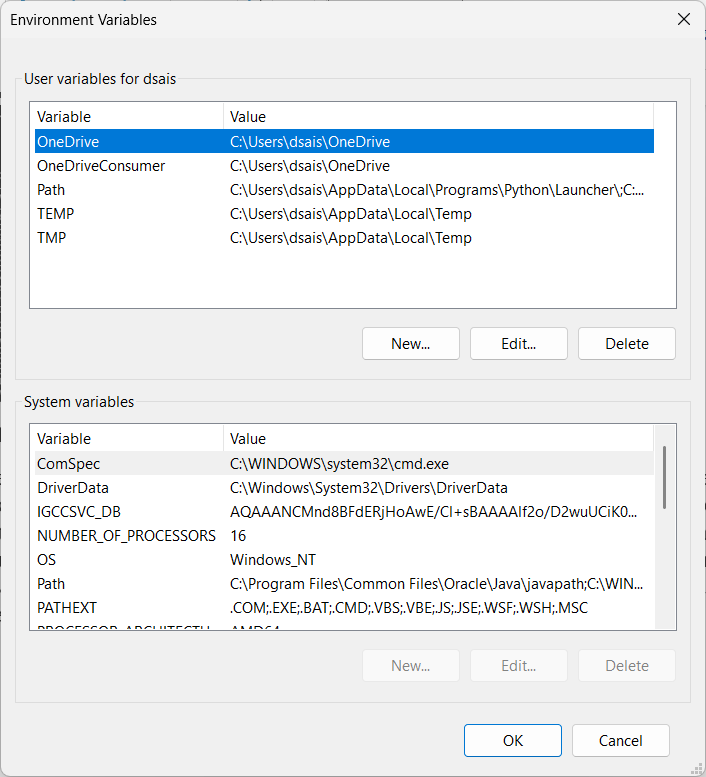
Salesforce Data Loader is a client application designed for bulk import, export, update, and deletion of Salesforce records using CSV files, suitable for processing large datasets efficiently. It supports an easy-to-use wizard interface, drag-and-drop field mapping, and works across Windows and Mac platforms, offering detailed success and error logs for effective data management.

**4.Deployment in vs code and SF dx:**

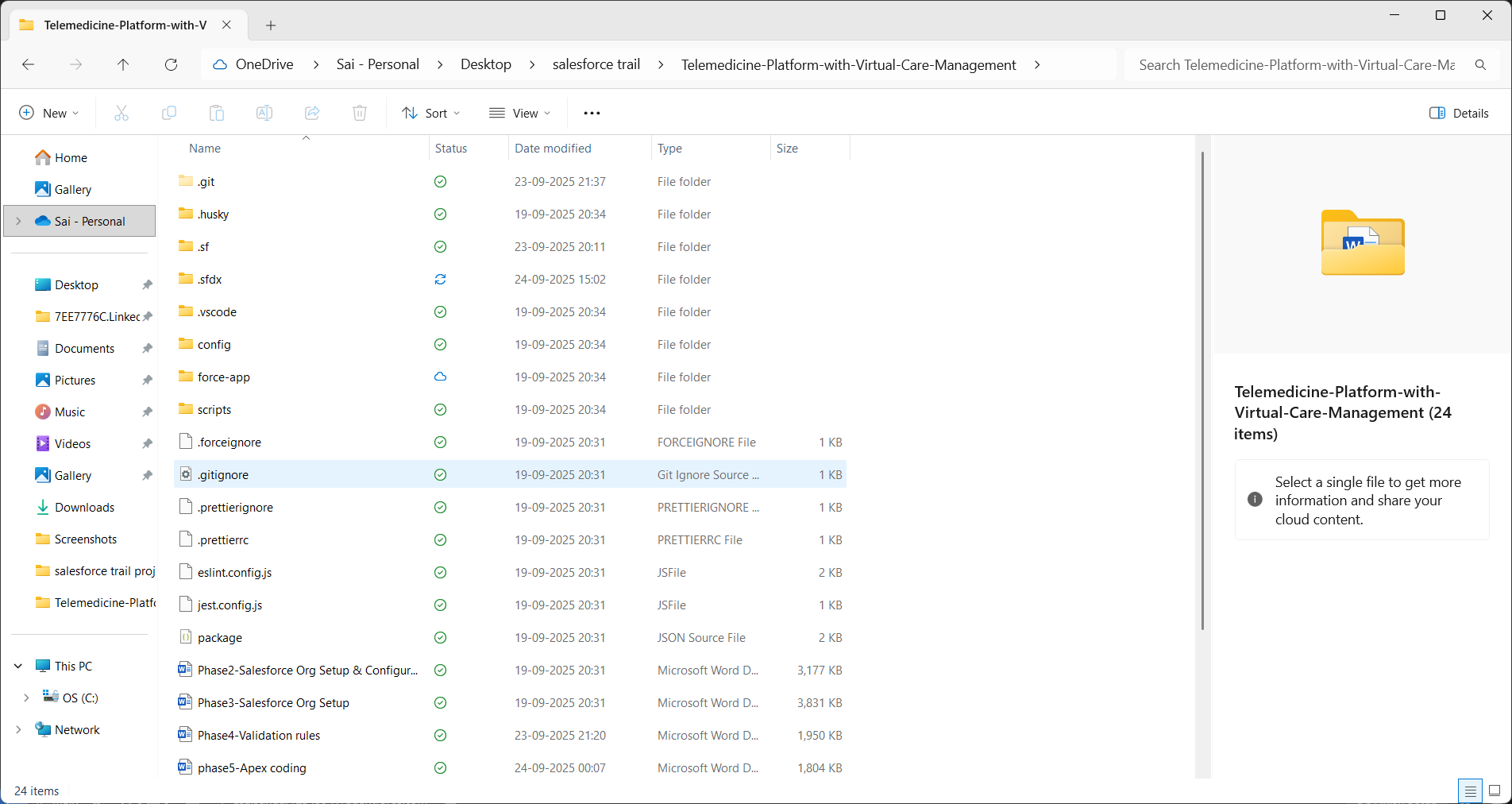
1. This is the sf command to check salesforce CLI is installed or not.



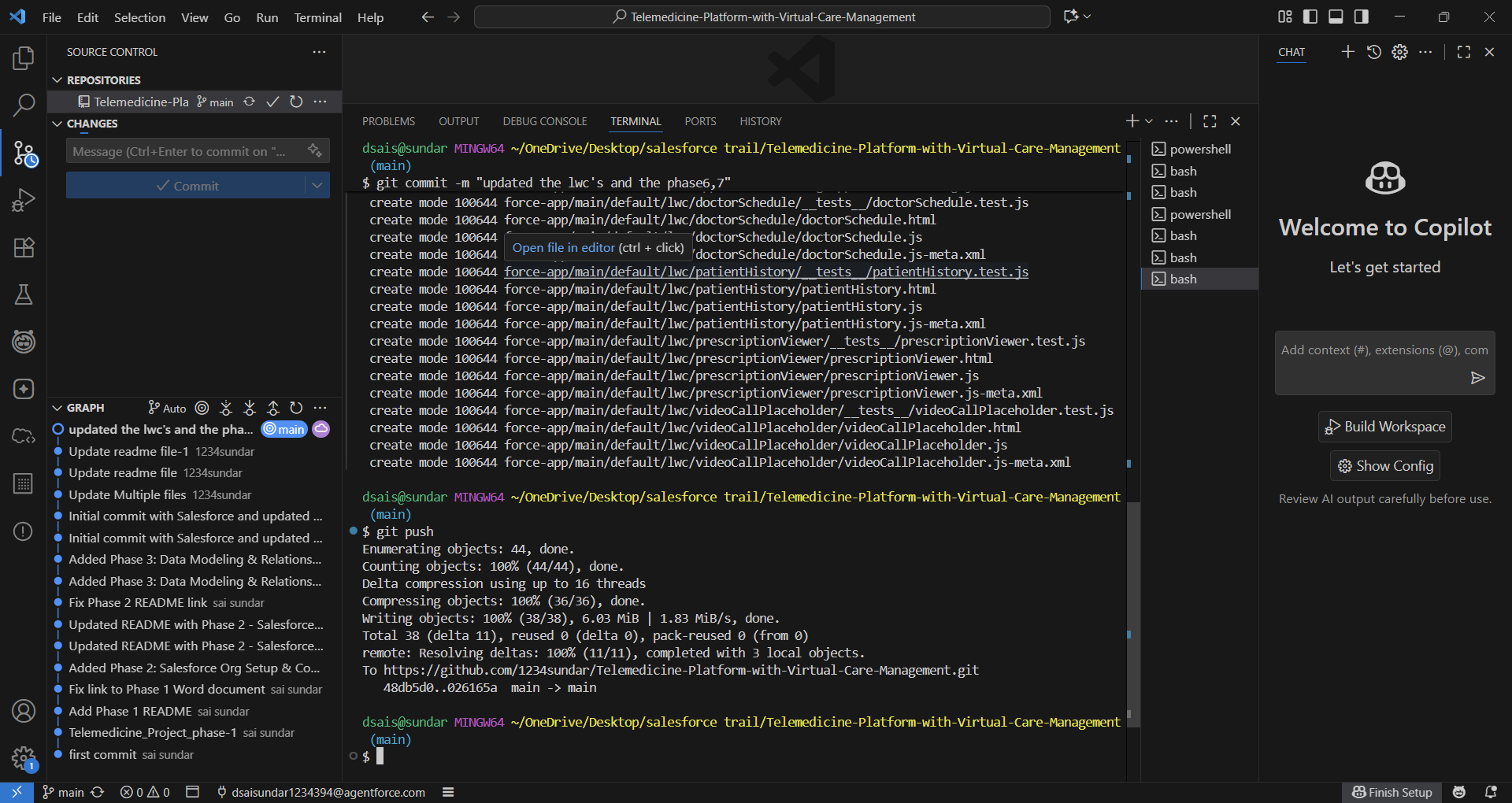
1. Check correct environment is set or not



1. **SFDX Folder**

****

1. **Successfully push the all lwc to the github repo using the commands**

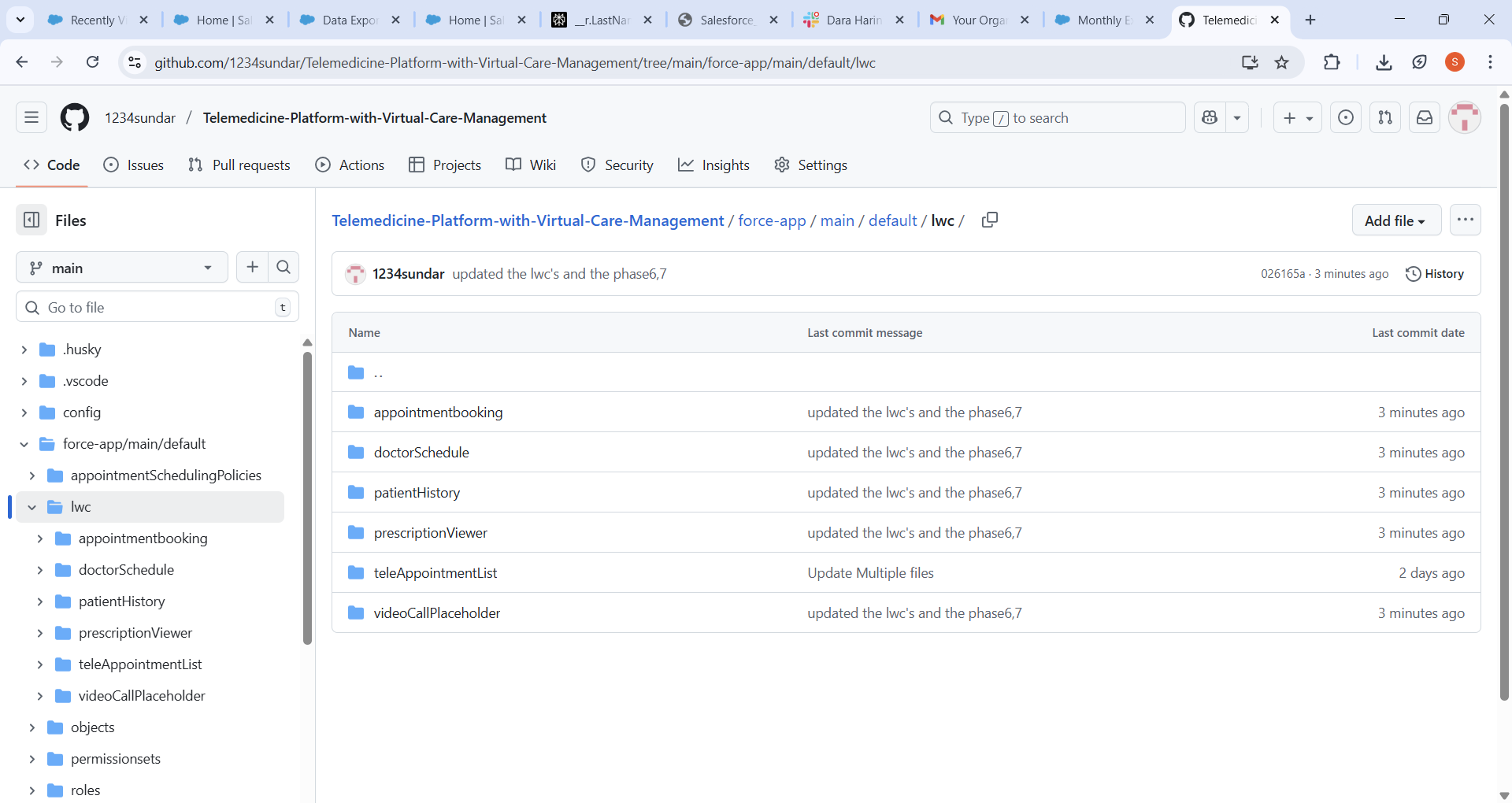


//git init

//git add .

//git commit -m “updated the github repo”

// git Push



For the deployment and retrival of the source is done by right click it and just tap deploy or retrive.

For your telemedicine project, Salesforce DX (SFDX) was used to efficiently manage and deploy custom objects like Patients and Appointments. Metadata and components were retrieved from your development org, enhanced locally in Visual Studio Code, and redeployed via SFDX commands. This method ensured controlled, versioned updates and smooth deployment of telemedicine features. Rigorous testing was performed to confirm workflows and UI elements operated as intended in your telemedicine system.