



A CRM Application to Manage the Services offered by an Institution

1. Project Overview

Develop a centralized Customer Relationship Management (CRM) application in Salesforce to manage and optimize the institution's service offerings, customer interactions, and internal workflows. The application will enhance service delivery, improve customer satisfaction, and streamline operational efficiency.

2. Objectives

The objective of this project is to develop a centralized CRM application in Salesforce to streamline the management of services offered by the institution, enhance customer interactions, and improve operational efficiency. By creating a 360-degree view of customers, automating service workflows, and enabling data-driven insights through real-time analytics, the system will empower the institution to deliver personalized, timely, and high-quality services. Additionally, it will provide a self-service portal for customers, integrate multi-channel communication, and ensure compliance with data security and regulatory standards. The ultimate goal is to enhance customer satisfaction, optimize resource utilization, and scale operations seamlessly to meet future demands.

3. Salesforce Key Features and Concepts Utilized

This Salesforce CRM project leverages a wide range of key features and concepts to streamline service management and enhance customer interactions. **Service Cloud** is utilized for efficient case management, Omni-Channel support, and a unified service console for agents. **Salesforce Flow** enables automation of workflows, including approvals, service request processing, and guided processes using screen flows. The **Experience Cloud** provides a self-service portal and

knowledge base, empowering customers to access services independently and find answers to common queries. Through **Customer 360**, the system consolidates customer data across accounts, contacts, and cases, ensuring a comprehensive view for personalized service delivery.

Analytics and Reporting tools, including real-time dashboards and custom reports, offer actionable insights into performance metrics and trends. The modern **Lightning Experience** ensures an intuitive interface for users, with custom functionality enabled by Lightning Web Components (LWC). **Salesforce Scheduler** streamlines appointment and service scheduling, while SLA tracking and escalation rules ensure timely resolution of customer issues. **Einstein AI** adds predictive analytics and sentiment analysis to proactively address customer needs and satisfaction levels.

The project integrates seamlessly with external systems through APIs and leverages **AppExchange solutions** for advanced functionalities like document management and surveys. Robust security measures, including role-based access and data encryption, ensure compliance with privacy regulations. For enhanced engagement, tools like **Marketing Cloud** or **Pardot** can be incorporated to manage personalized campaigns and automated customer journeys. Collaboration tools such as **Chatter** and the **Salesforce Mobile App** further enhance communication and accessibility, making the system versatile and scalable for future growth.

4. Detailed Steps to Solution Design:

Step 1: Create Objects from Spreadsheet

Create Objects from Spreadsheets

- Use the "Create Object from Spreadsheet" feature to upload and map fields for Course, Consultant, and Student objects.

Establish Relationships

- Create lookup relationships between:
 - Appointment and Student
 - Appointment and Consultant.

Create Registration Object

- Build a custom object called Registration to link Student and Course data.

Add Lookup for Student Queries

- Add a lookup relationship between Student and Case to handle immigration and visa queries.

Update Case Object Fields

- Modify the Type field: Add Immigration and Visa Application.
- Update the Status field: Add Open and In-progress.

Build the Lightning App

- Use App Manager to create a new Lightning app called EduConsultPro. Add Home, Students, Courses, Consultants, Appointments, Registrations, and Cases as navigation items.

Assign Profiles

- Assign the System Administrator profile to the app and save the configuration.

object-creator.salesforce.com/load-data-wizard.shtml

Create a custom object from a spreadsheet

Define object and fields

Choose the data source, map fields and their types, and import field data.

CSV File Details

Encoding Format

Unicode (UTF8)

Values Separated By

Comma

Field Label Source

Enter manually

Detect from row

* Field Labels Row

1

Import 5 rows of Data?

No, skip import

Yes, import data

Record Name Field

Let Salesforce Create a Default F

Fields 5 of 5 to import

Hide mapped fields

IMPORT FILE FIELD NAME	Salesforce FIELD NAME	Salesforce FIELD TYPE	ADD TO LAYOUTS	FIELD PREVIEW
✓ Course Name	Course Name	Text	✓	IELTS
✓ Description	Description	Text	✓	Let's Learn IELTS
✓ Start Date	Start Date	Phone	✓	03-01-2024
✓ End Date	End Date	Date	✓	05/31/2024
✓ Instructor	Instructor	Text	✓	Sandeep


Back

Next

object-creator.salesforce.com/load-data-wizard.shtml

Create a custom object from a spreadsheet

Nice Work!



Now you can add your object to a Lightning app. You might need to refresh the object list to see it.

Import Overview

Object Created

Course - Course (2)

Fields Detected

5

Fields Created

5

Rows Detected

5

Rows Imported

5

Import Another Object

Student | Introducing ChatGPT | OpenAI | ChatGPT | Case | Salesforce

flowcom-9b-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/Case/FieldsAndRelationships/Status/view

case

Setup | Home | Object Manager

Case

Details

Fields & Relationships

Case Page Layouts

Case Close Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Field Dependencies

No dependencies defined.

Validation Rules

No validation rules defined.

Case Status Picklist Values

Action	Values	API Name	Closed	Default	Chart Colors	Modified By
Edit Deactivate	New	New	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Assigned dynamically	Abhishek.K. 05/11/2024, 9:46 am
Edit Del Deactivate	Working	Working	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	Abhishek.K. 05/11/2024, 9:46 am
Edit Del Deactivate	Escalated	Escalated	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	Abhishek.K. 05/11/2024, 9:46 am
Edit Del Deactivate	Closed	Closed	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	Abhishek.K. 05/11/2024, 9:46 am
Edit Del Deactivate	Open	Open	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	Abhishek.K. 19/11/2024, 9:49 pm
Edit Del Deactivate	In-progress	In-progress	<input type="checkbox"/>	<input type="checkbox"/>	Assigned dynamically	Abhishek.K. 19/11/2024, 9:49 pm

Inactive Values

No inactive values defined.

Student | Object Manager | Salesforce | App Manager | Salesforce | Object creator

flowcom-9b-dev-ed.develop.lightning.force.com/lightning/setup/NavigationMenus/home

Search Setup

New Lightning App

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name: EduConsultPro

*Developer Name: EduConsultPro

Description: Enter a description...

App Branding

Image: Upload

Primary Color Hex Value: #0070C2

Org Theme Options

☐ Use the app's image and color instead of the org's custom theme.

Next

ID	App Name	App Description	Created Date	Theme	Status
11	Hive App	Hive App	05/11/2024, 10:12 am	Lightning	✓
12	Lightning Usage App	LightningInstrumentation	05/11/2024, 9:46 am	Lightning	✓
13	Marketing CRM Classic	Marketing	05/11/2024, 9:46 am	Classic	✓

Step 2: Create a Screen Flow for Student Admission Application process

1)Create a Screen Flow

- 1)Go to Setup > Flow Builder > New Flow > Screen Flow.
- 2)Add a Screen Element with label Student Info.
- 3)Create a record variable resource StudentRecordRes to display Student Object fields. Drag required fields to collect student data.

2)Create Student Record

- 1)Add a Create Element after the Student Info screen.
- 2)Label it Create Student Record, set "One Record to Create" and use StudentRecordRes to set record values.

3)Select Course

- 1)Add a Screen Element labeled Course Screen.
- 2)Add a Picklist Component labeled Select Course with choices: IELTS, GRE, GMAT, Duolingo, TOEFL.

4)Decision for Course Selection

- 1)Add a Decision Element labeled Selecting Course.
- 2)Create outcomes for each course (e.g., "Selected IELTS") with conditions:

Resource: Select_Course

Operator: Equals

Value: Corresponding course variable (e.g., IELTS).

5)Fetch and Register Course

Label it (e.g., Get IELTS Rec), set conditions:

Field: Course Name

Value: Select_Course.

Add a Create Element to create a Registration Record.

Object: Registration

6)Field Values:

Course_Name__c: {!Get_IELTS_Rec.Id}

Student_Name__c: {!StudentRecordRes.Id}

7)Send Email to Student

1)Add a Text Template Resource (e.g., StuRegistrationEmailTextTempBody) for the email body.

2)Add an Action Element labeled Send Email to Student, and set:

Body: {!StuRegistrationEmailTextTempBody}

Recipient Address List: {!StudentRecordRes.Email__c}

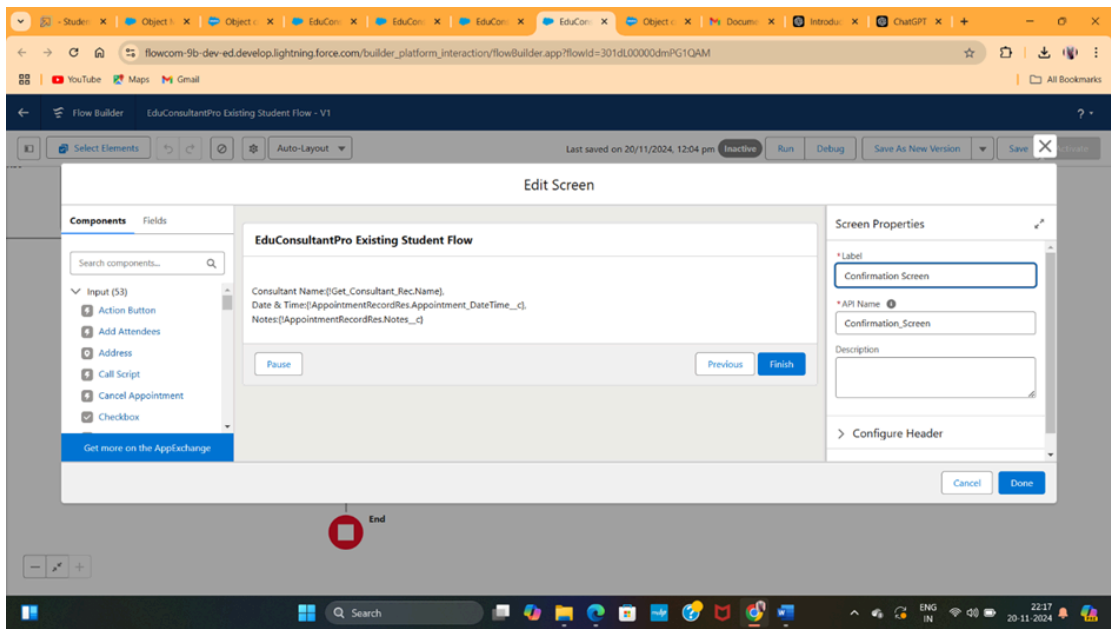
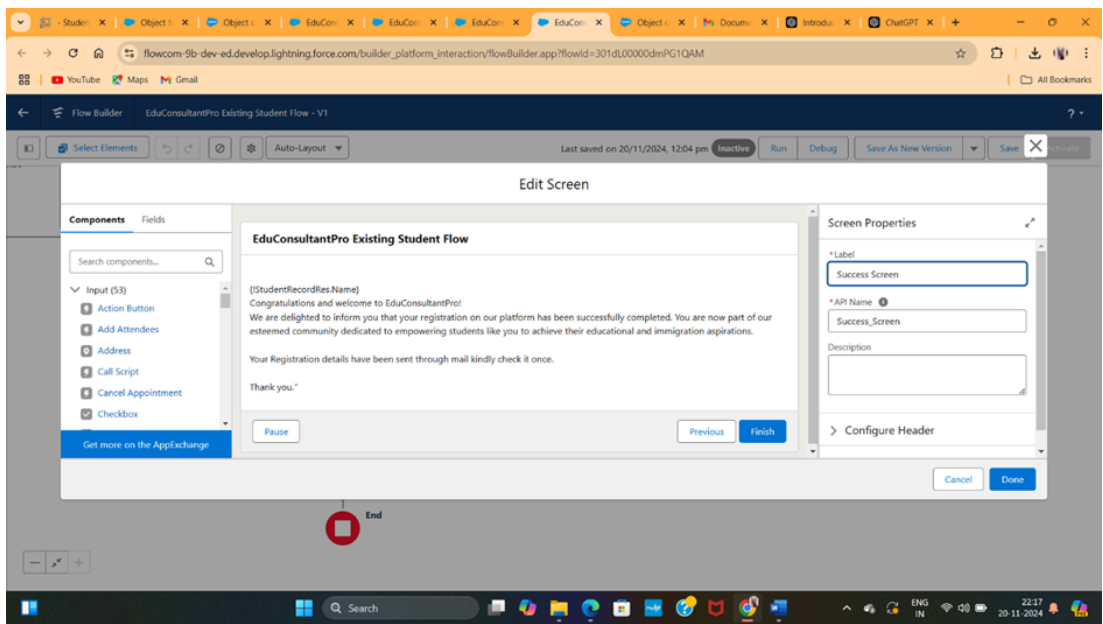
Subject: {!StuRegistrationEmailTextTempSub}

8)Success Screen

1)Add a Screen Element labeled Success Screen.

2)Drag a Display Text Component, label it SuccessMessage, and display a confirmation message.

3)This process completes a flow that collects student information, registers them for a course, and sends a confirmation email.



flowcom-9b-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301dL00000dmPG1QAM

Flow Builder: EduConsultantPro Existing Student Flow - V1

Last saved on 20/11/2024, 12:04 pm

Get GRE Rec Get Records

Create GRE Registration Rec Create Records

Get GMAT Rec Get Records

Create GMAT Registration Rec Create Records

Get Duolingo Rec Get Records

Create Duolingo Registration Rec Create Records

Send Email to Student Action

Confirmation Screen Screen

Send Email

*Label: Send Email to Student

*API Name: Send_Email_to_Student

Description:

Send Email emailSimple-emailSimple

Use values from earlier in the flow to set the inputs for the "Send Email" core action. To use its outputs later in the flow, store them in variables.

Set Input Values for the Selected Action

Add Threading Token to Body Not Included

Add Threading Token to Subject Not Included

flowcom-9b-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301dL00000dmPG1QAM

Flow Builder: EduConsultantPro Existing Student Flow - V1

Last saved on 20/11/2024, 12:04 pm

Selected IELTS

Get Rec Get Records

Get IELTS Rec Get Records

Create IELTS Registration Rec Create Records

Selected GRE

Get GRE Rec Get Records

Create GRE Registration Rec Create Records

Get Records

Get Records of This Object

*Object: Course - Course (2)

Filter Course - Course (2) Records

Condition Requirements

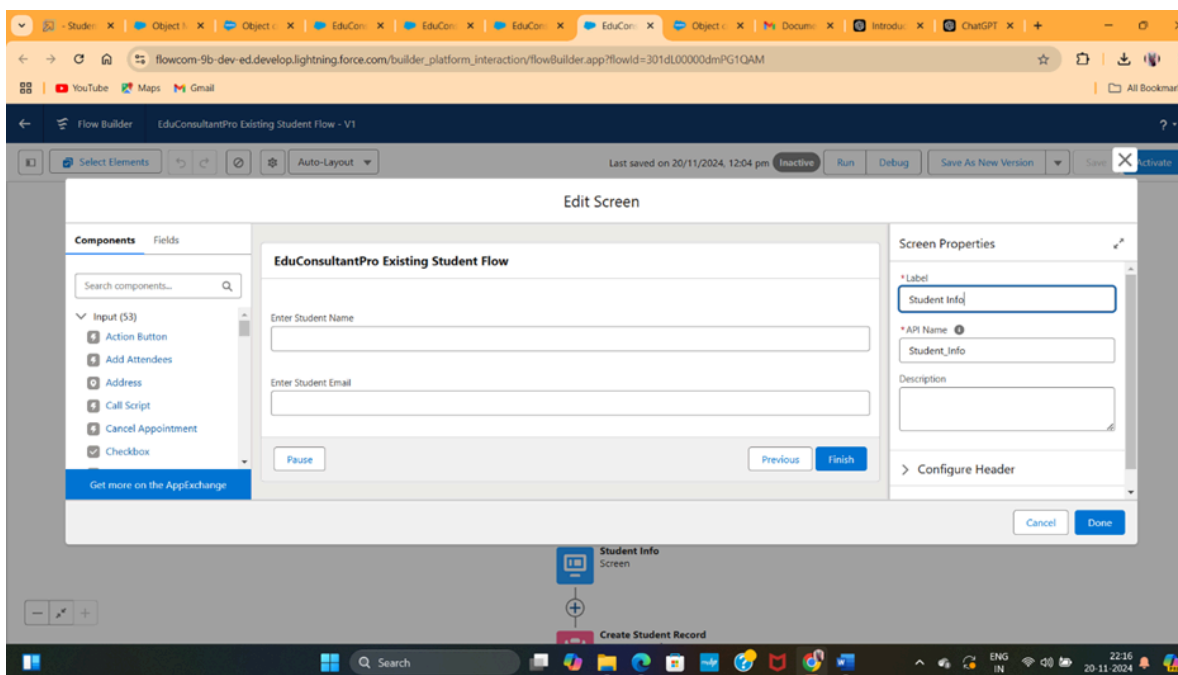
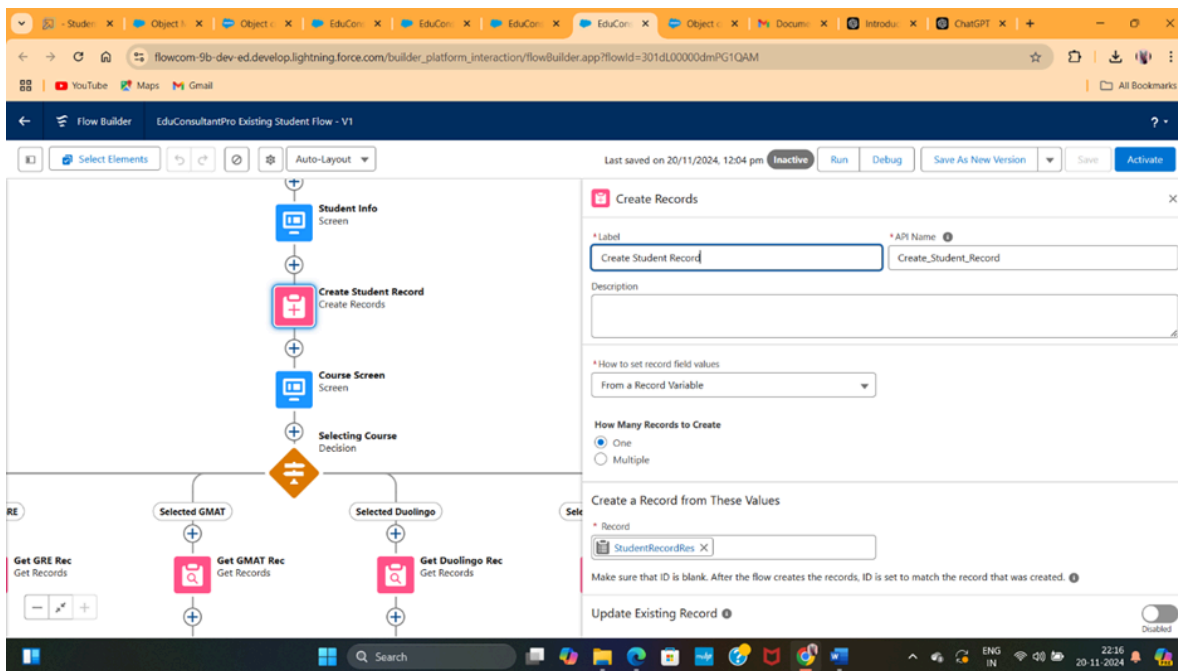
All Conditions Are Met (AND)

Field: Course_Name__c Operator: Equals Value: Select_Course X

Sort Course - Course (2) Records

Sort Order: Not Sorted

If you store only the first record, filter by a unique field, such as ID.



Step 3: Create a user with a Standard platform user profile.

Create New User:

- Go to Setup > Users > New User.
- Set Last Name: Consultant, License: Salesforce Platform, Profile: Standard Platform User.
- Fill mandatory fields and save.

Edit Approver Settings:

- In Setup > Users, click Edit next to your name.
- Under Approver Settings, set Manager to Consultant.

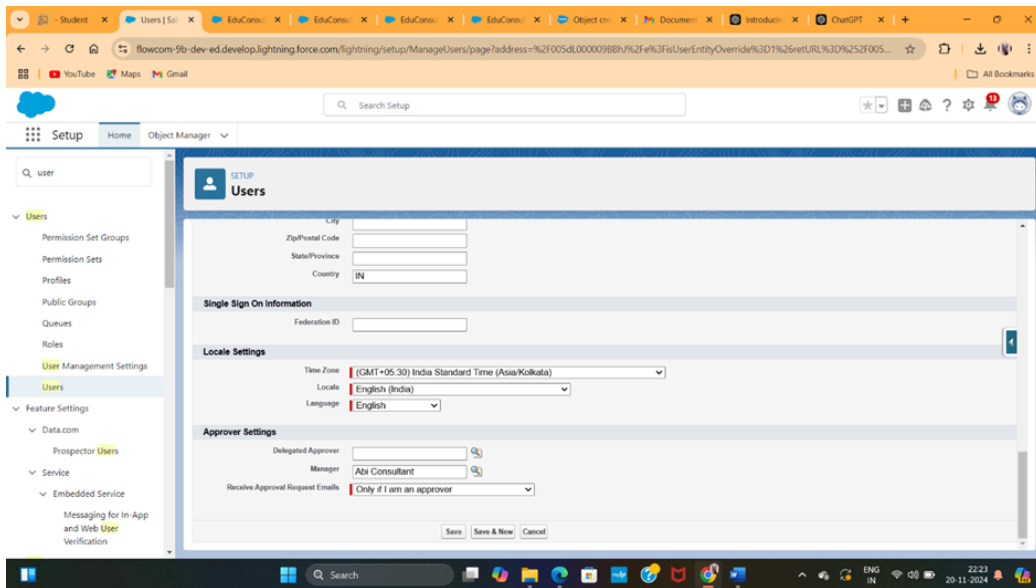
Save Changes:

- Click Save to update.

The screenshot displays the Salesforce Setup interface with the 'Users' section selected. The 'User Edit' form for 'Abivarsha K' is shown, with the following details:

General Information	
First Name	Abivarsha
Last Name	K
Alias	AK
Email	abivarsakamara031@gmail.com
Username	distribution@flow.com
Nickname	User173078018851929203
Title	
Company	NA
Department	
Division	

Role and Profile	
Role	SF Admin
User License	Salesforce
Profile	System Administrator
Active	<input checked="" type="checkbox"/>
Marketing User	<input checked="" type="checkbox"/>
Offline User	<input checked="" type="checkbox"/>
Knowledge User	<input type="checkbox"/>
Flow User	<input type="checkbox"/>
Service Cloud User	<input checked="" type="checkbox"/>
Site.com Contributor User	<input type="checkbox"/>
Site.com Publisher User	<input type="checkbox"/>
WDC User	<input type="checkbox"/>
Data.com User Type	None
Data.com Monthly Addition Limit	300



Step 4: Create an Approval Process for Property Object

1. Create Email Templates

1. Enable Lightning Email Templates:

- Go to Setup > Templates > Lightning Email Templates, and toggle it on.

2. Create Folder for Templates:

- Use the App Launcher, search for Email Templates, and create a folder.

3. Create Email Templates:

- Inside the folder, create three email templates:
 - Submission Template: Enter the provided HTML content and save.
 - Approval Template: Use similar steps.
 - Rejection Template: Use similar steps.

2. Configure Approval Process

1. Go to Approval Processes:

- Navigate to Setup > Approval Processes, and select Appointment under Manage Approval Processes For.

2. Define Approval Steps:

- Next Automated Approver Determined By: Select Manager.
- Record Editability Properties: Choose Administrators OR the currently assigned approver.
- Save the process.

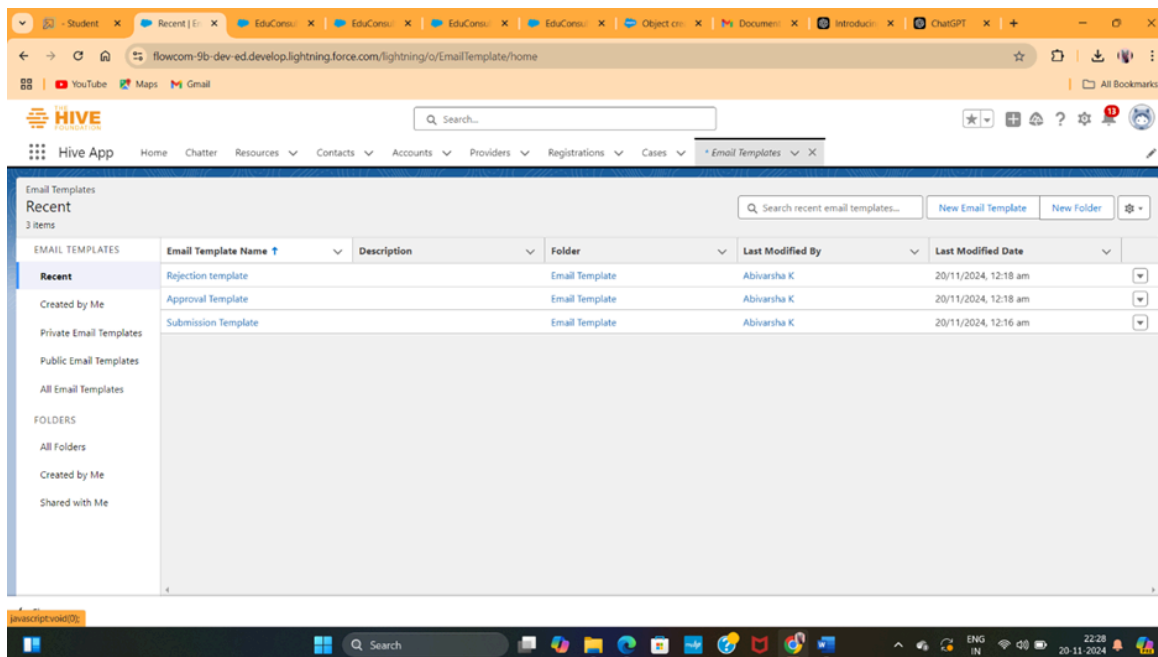
3. Add Initial Submission Actions:

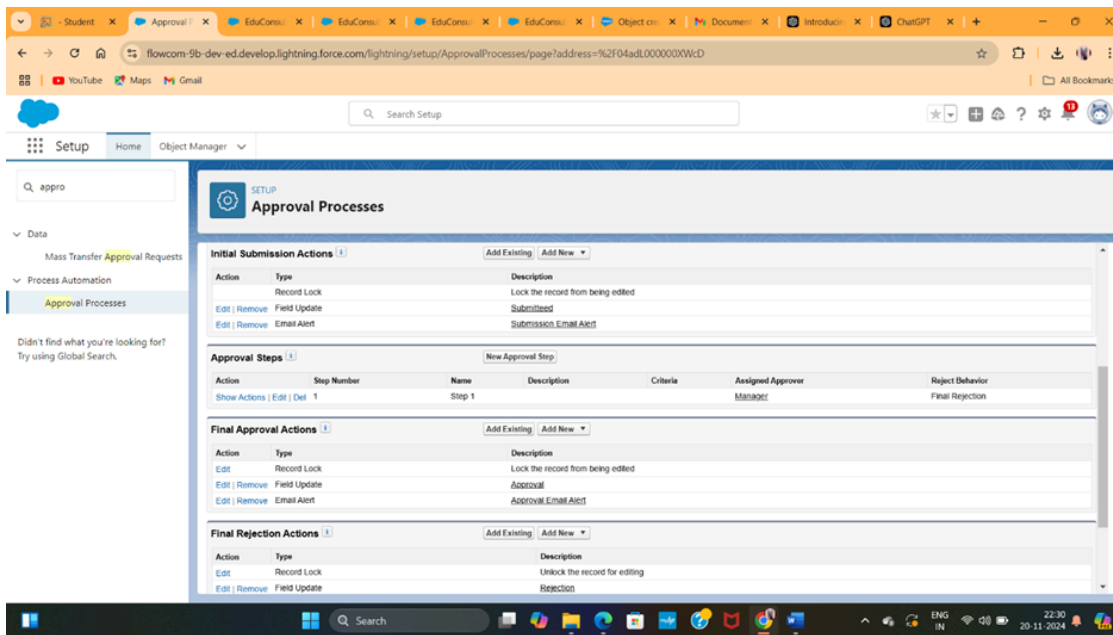
- Under Initial Submission Actions, create a Field Update:
 - Field Name: Submitted
 - Field to Update: Appointment: Status
 - Specific Value: Pending.

4. Create New Approval Process:

- Use the Jump Start Wizard to configure:
 - Process Name: Appointment Approval.
 - Select Approver: Choose Manager for automatic assignment.

This setup enables automated approvals for appointment records with the configured email notifications.





Step 5: Create a Record Triggered Flow

1)Go to Flows:

In Setup, search for Flows and click New Flow.

2)Select Flow Type:

Choose Record-Triggered Flow and click Create.

3)Set Object:

In the Configure Start window, select Object as Appointment.

4)Add Action Element:

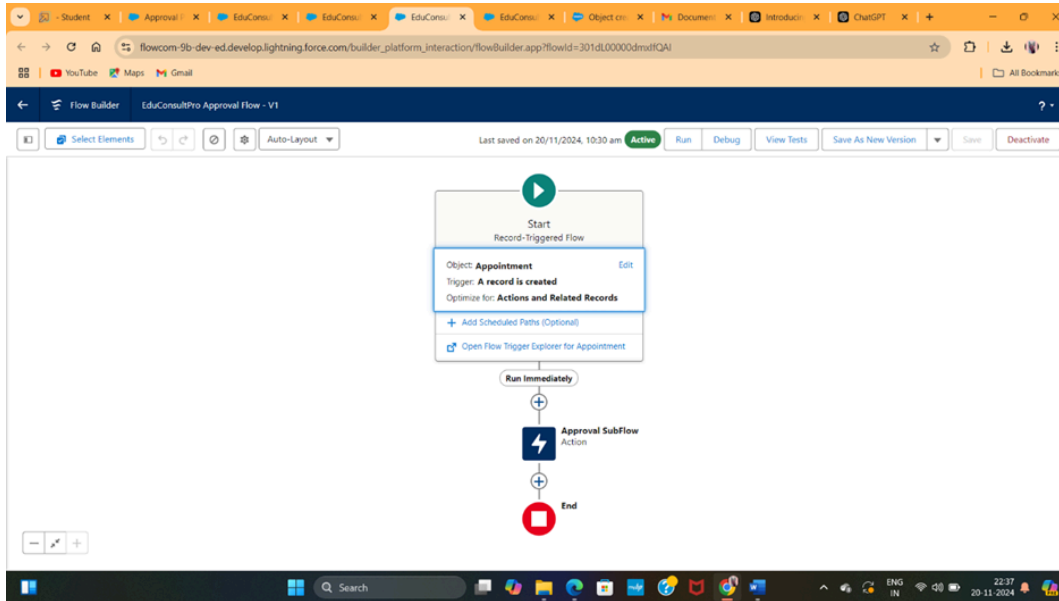
Drag an Action Element and select Submit for Approval.

Label it Approval SubFlow and set RecordId to `{!$Record.Id}`.

5) Save and Activate:

Save the flow as EduConsultPro Approval Flow and click Activate.

This setup will automatically submit appointments for approval.



Step 6: Create a Screen Flow for Existing Student to Book an Appointment

Student Info Screen:

- Add two text fields for **Student Name** and **Student Email**.

Fetch Student Record:

- Use **Get Record** to retrieve student details based on name and email.

Display Student Details:

- Add a screen to show student info using display text.

- Add radio buttons for **Book an Appointment** or **Immigration Case**.

Decision Element:

- Create paths for **Appointment** and **Case**.

Appointment Path:

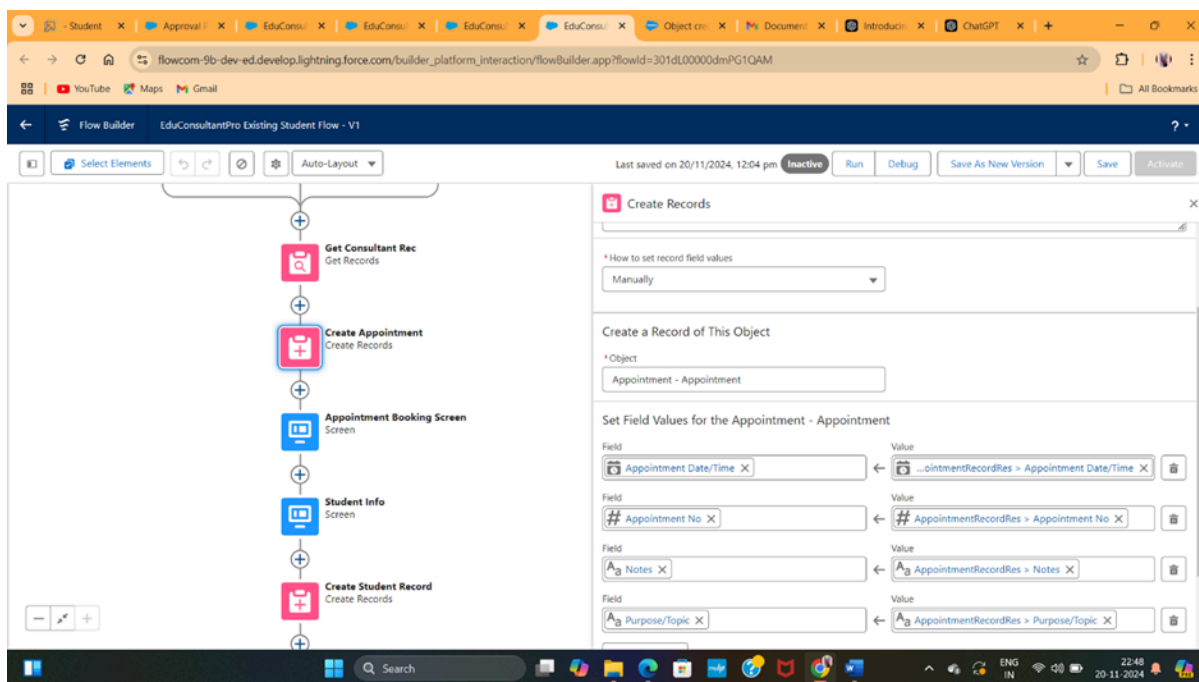
- Add an **Appointment Booking Screen**.
- Fetch consultant details using **Get Record**.
- Use **Create Element** to save appointment and display a confirmation screen.

Case Path:

- Add a **Subflow** to create a case.

Save and Activate:

- Save as **EduConsultantPro Existing Student Flow** and activate.



Flow Builder: EduConsultantPro Existing Student Flow - V1

Last saved on 20/11/2024, 12:04 pm

Flow Diagram:

```
graph TD; Start([Start]) --> CreateSubflow[Create Student Case Subflow]; CreateSubflow --> DisplayScreen[Display Student Details Screen]; DisplayScreen --> Decision{Appointment or Case Decision}; Decision -- Appointment --> AppointmentPath[Appointment]; Decision -- Case --> CasePath[Case]; Decision -- Default Outcome --> DefaultOutcomePath[Default Outcome]; AppointmentPath --> GetRecords[Get Consultant Rec Get Records]; CasePath --> GetRecords; DefaultOutcomePath --> GetRecords; GetRecords --> CreateAppointment[Create Appointment Create Records];
```

Decision Configuration:

- *Label: Appointment or Case
- *API Name: Appointment_Lor_Case
- Description:
- Outcomes: For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.
- OUTCOME ORDER: +
- OUTCOME DETAILS: Delete Outcome
- Appointment:
 - *Label: Appointment
 - *Outcome API Name: Appointment
 - Condition Requirements to Execute Outcome: All Conditions Are Met (AND)
 - Resource: A... Details > How may I help you X
 - Operator: Equals
 - Value: A... Book an Appointment X
 - + Add Condition
- Case:
- Default Outcome:

Flow Builder: EduConsultantPro Existing Student Flow - V1

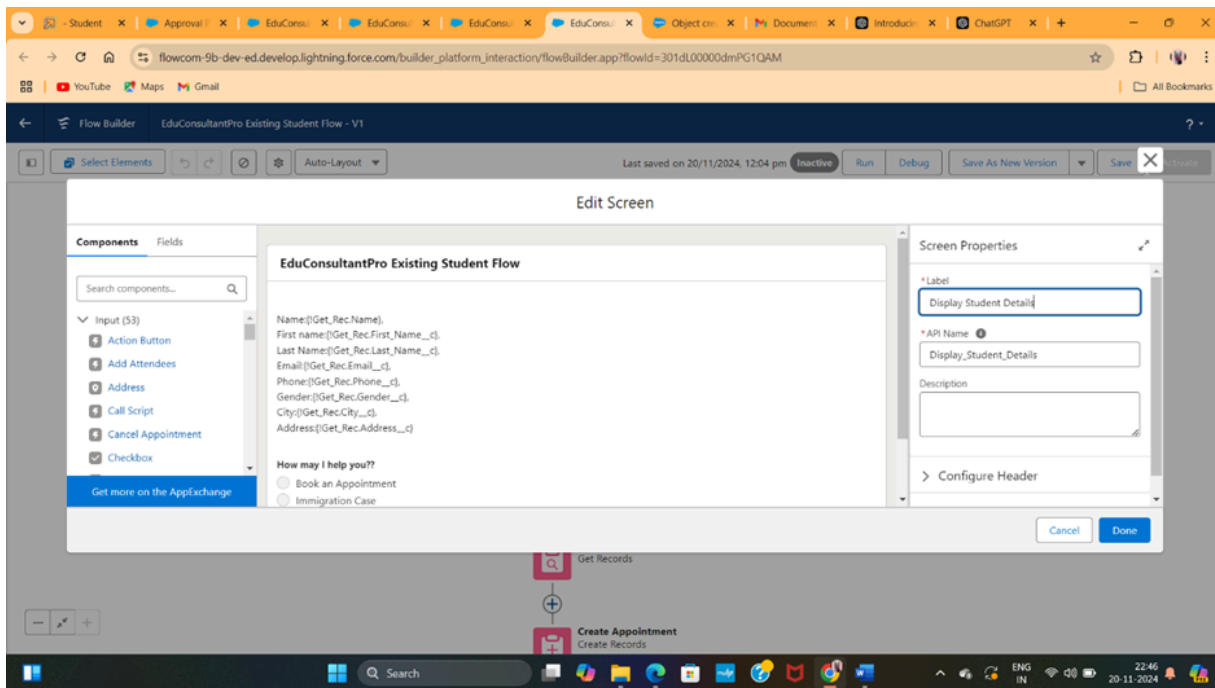
Last saved on 20/11/2024, 12:04 pm

Flow Diagram:

```
graph TD; Start([Start]) --> SelectedIELTS[Selected IELTS]; SelectedIELTS --> GetRecIELTS[Get Rec Get Records]; GetRecIELTS --> GetIELTSRec[Get IELTS Rec Get Records]; GetIELTSRec --> CreateIELTSRec[Create IELTS Rec Create Records]; SelectedGRE[Selected GRE] --> GetRecGRE[Get GRE Rec Get Records]; GetRecGRE --> CreateGRERec[Create GRE Rec Create Records];
```

Get Records Configuration:

- *Label: Get Rec
- *API Name: Get_Rec
- Description:
- Get Records of This Object:
 - *Object: Student - Students
- Filter Student - Students Records:
 - Condition Requirements: All Conditions Are Met (AND)
 - Field: Student_Name__c, Operator: Equals, Value: A... Enter_Student_Name X
 - Field: Email__c, Operator: Equals, Value: A... Enter_Student_Email X
 - AND



Step 7: Create a ScreenFlow to Combine all the flows at one place

1. Welcome Screen:

- Add a Screen Element labeled Welcome Screen.
- Use a Display Text component with SuccessMessage and paste the success message text.

2. Existing or New Student Confirmation Screen:

- Add a Screen Element labeled Existing or New Student Confirmation Screen.
- Add a Radio Button component:
 - Label: Are you an Existing Student.
 - Choices: Yes and No.

3. Decision Element:

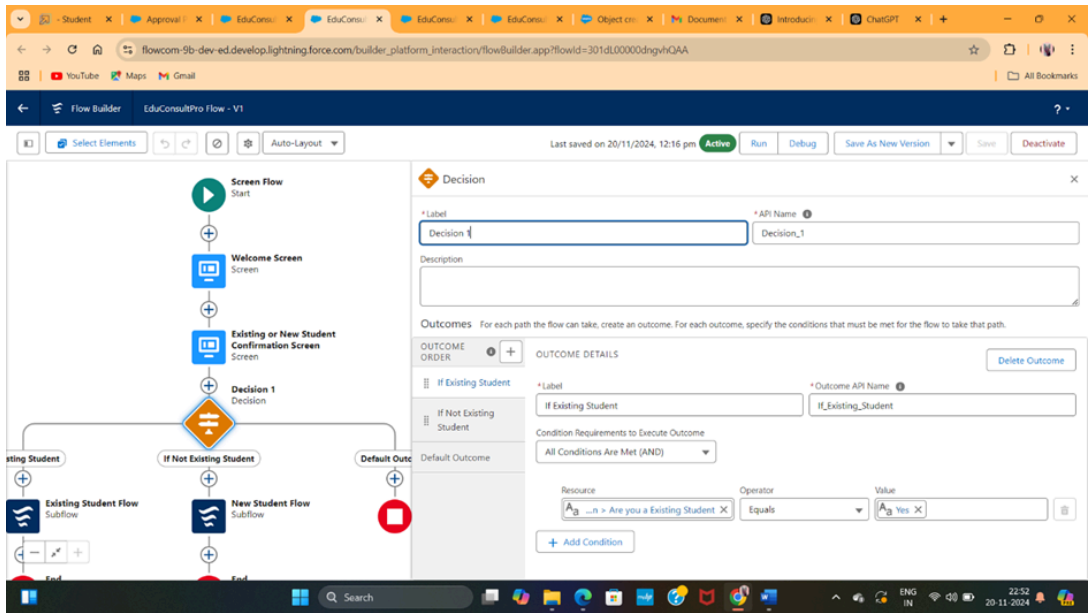
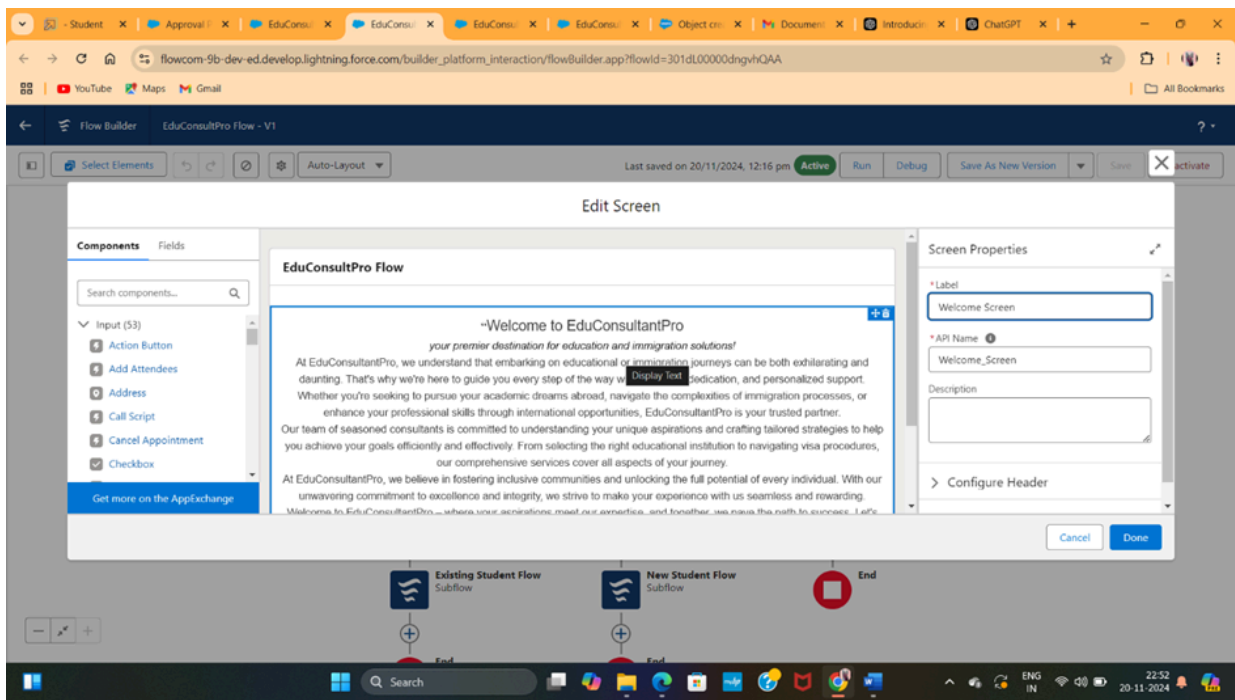
- Add a Decision Element labeled Decision 1.
- Outcome 1: If Existing Student:
 - Condition: `{!Are_you_a_Existing_Student}` equals Yes.

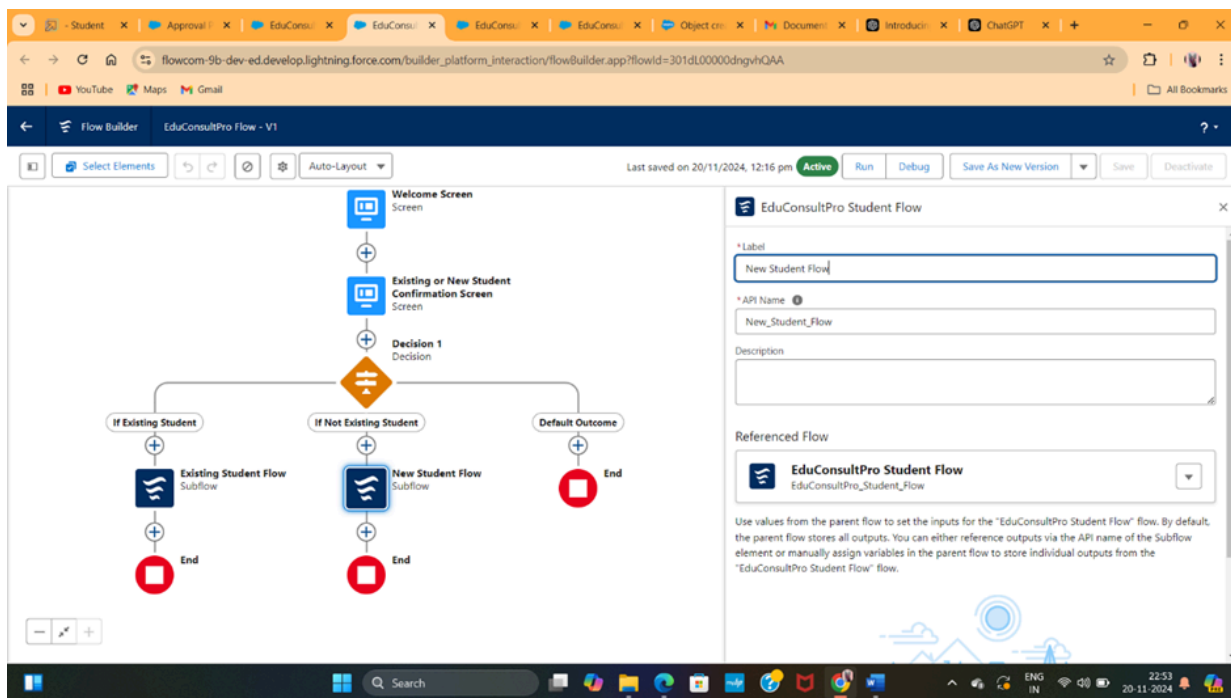
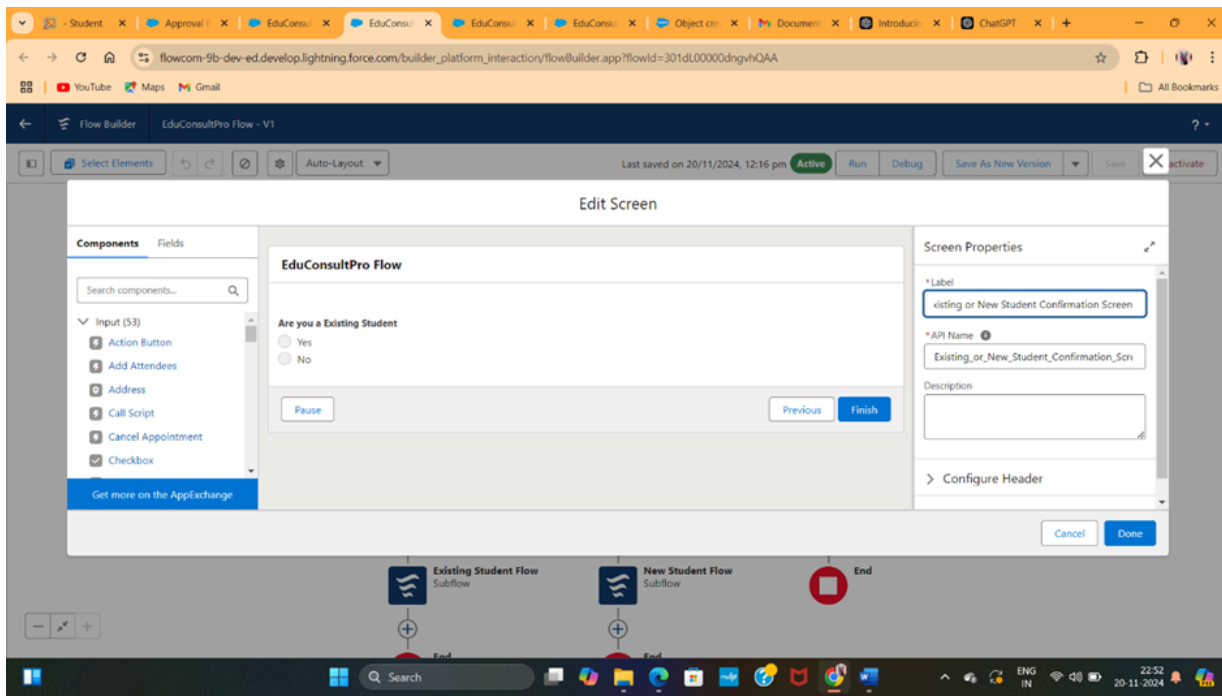
4. Subflow for Existing Student:

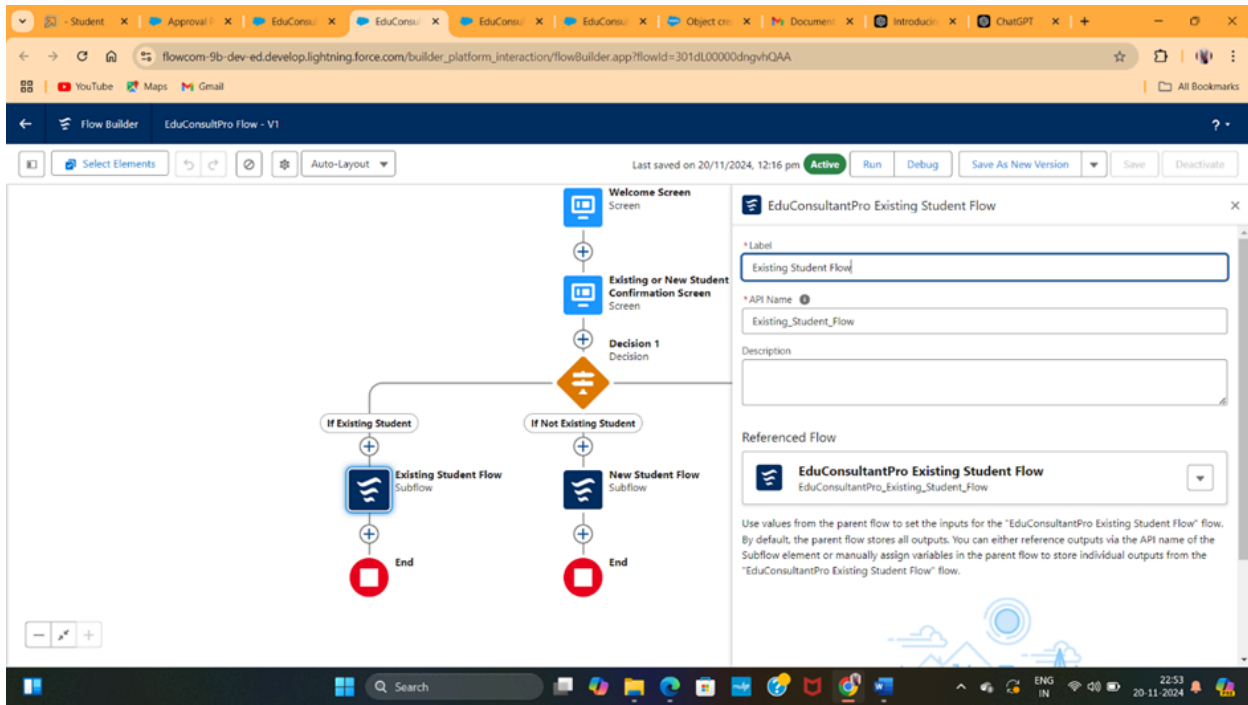
- Add a Subflow Element after the If Existing Student path.
- Select EduConsultPro Existing Student Flow.

This flow handles the welcome screen, asks if the student is existing or new, and branches to an existing

student flow.







Step 8: Create a lightning app page

1. Create Home Page:

- Go to Setup > App Builder > Lightning App Builder.
- Click New, select Home Page, and click Next.
- Name the page EduConsultPro Home Page, select Standard Home Page template, and click Done.

2. Add Flow Component:

- Drag the Flow Component to the top-right region of the page.
- Search for EduConsultPro Flow and add it.

3. Save and Activate:

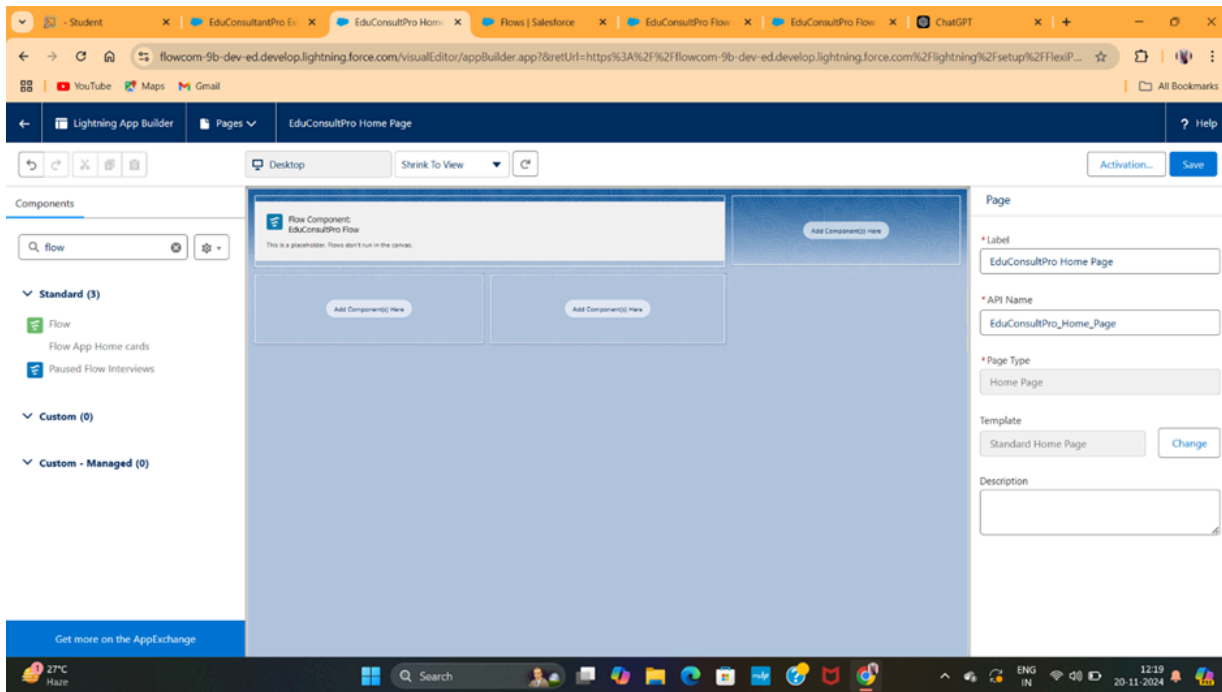
- Click Save, then Activate.

4. Assign to Apps and Profiles:

- Click App and Profile, then Assign to Apps and Profiles.
- Select Sales App, click Next.

- Choose System Administrator profile, click Next and then Save.

This assigns the EduConsultPro Flow to the newly created home page and configures it for the Sales App and System Administrator profile.



5. Testing and Validation

For the CRM application to manage services offered by an institution, testing involves two key areas: **Unit Testing** and **User Interface (UI) Testing**. Unit testing focuses on validating back-end components like Apex classes and triggers, ensuring service workflows, automation, and data handling operate correctly while meeting Salesforce's governor limits and achieving at least 75% code coverage. UI testing ensures the functionality and usability of customer and agent-facing interfaces, including service request workflows, self-service portals, and dashboards. It includes cross-browser/device compatibility, accessibility compliance, and end-to-end scenario testing to guarantee a seamless and efficient user experience..

6. Key Scenarios Addressed by Salesforce in the Implementation Project

Service Request Management

- Enable customers to submit service requests via multiple channels, such as a self-service portal, email, or phone.
- Automate service request assignment to the appropriate team based on predefined rules.
- Track service request status, escalate unresolved issues, and ensure SLA compliance.

2. Customer Interaction Tracking

- Provide a 360-degree view of customer profiles, including interaction history, preferences, and previous requests.
- Consolidate customer touchpoints across email, chat, and phone for seamless communication.

3. Case Management

- Efficiently log, monitor, and resolve service-related issues or complaints.
- Automate escalation processes for high-priority cases to ensure timely resolutions.
- Provide agents with real-time dashboards for case monitoring and workload management.

4. Self-Service Enablement

- Offer customers a portal to access services, check statuses, and browse a knowledge base for self-resolution.
- Reduce dependency on support agents by empowering customers with self-help resources.

5. Service Delivery Optimization

- Automate approval processes for service requests, reducing delays and manual intervention.
- Schedule and manage appointments with Salesforce Scheduler for streamlined service delivery.
- Track resource availability and optimize allocation for operational efficiency.

6. Analytics and Reporting

- Enable real-time reporting on service performance, customer satisfaction, and operational metrics.
- Provide insights into trends, bottlenecks, and improvement areas through dashboards.

7. Customer Engagement and Communication

- Use automated notifications to keep customers informed about request statuses, appointment confirmations, and case updates.
- Implement targeted campaigns using Marketing Cloud or Pardot to engage customers and promote relevant services.

8. Data Security and Compliance

- Ensure secure handling of customer data using role-based access controls and encryption.
- Adhere to data protection regulations like GDPR or HIPAA, depending on institutional requirements.

9. Scalability and Customization

- Address growing customer and service volumes with Salesforce's scalable architecture.
- Customize workflows, data models, and UI elements to meet the institution's specific requirements.

10. Collaboration and Productivity

- Facilitate teamwork with tools like Chatter for internal collaboration on cases and projects.
- Provide mobile access via the Salesforce Mobile App for agents to manage requests on the go.

These scenarios ensure that Salesforce addresses both customer-facing and operational needs, driving efficiency, customer satisfaction, and growth for the institution.

7. Conclusion

In conclusion, the Salesforce CRM implementation project for managing the services offered by the institution aims to transform service delivery, streamline operations, and enhance customer satisfaction. By leveraging Salesforce's robust features such as Service Cloud, Experience Cloud, and automation tools, the solution provides a unified platform for efficient service request handling, proactive customer engagement, and real-time insights. The integration of self-service capabilities, compliance measures, and scalability ensures that the system meets both current and future institutional needs. This implementation not only optimizes resource utilization and response times but also fosters a more personalized and seamless experience for both customers and service agents, positioning the institution for sustained success and growth.