Educational Organisation Using ServiceNow

Team ID : NM2025TMID07546

Team Size : 5

Team Leader : SIVA A

Team member : SIVA V

Team member : SIVA BRINDHA S

Team member : SNEHA A SANIL

Team member : SRI KRISHNAN M

AIM:

The aim of this project is to **develop an Educational Management System using ServiceNow** that automates and streamlines the administrative processes of educational institutions by managing student and teacher information, simplifying admissions, and providing tools for monitoring academic progress.

ABSTRACT:

The project **"Educational Organisation Using ServiceNow"** focuses on developing an automated Educational Management System on the ServiceNow platform. The system is designed to simplify and streamline various administrative tasks within educational institutions, such as managing student and teacher data, handling admissions, maintaining academic records, and tracking student performance. By using ServiceNow's powerful tools like table creation, form design, client scripts, and process automation, the project demonstrates how non-IT processes can be efficiently managed on a cloud-based platform. This application enhances data accuracy, reduces manual effort, and improves overall institutional efficiency.

OBJECTIVE:

- The main objective of this project is to develop an Educational Management System using ServiceNow that streamlines the administrative operations of educational institutions.
- It enables efficient management of student and teacher information, simplifies the admission process, automates workflows, and provides tools for monitoring student progress.

METHODOLOGY:

The development of the **Educational Organisation Using ServiceNow** project followed a systematic process using the features and tools provided by the ServiceNow platform. The methodology involves the following steps:

1. **Setting Up ServiceNow Instance**

   - A personal developer instance of ServiceNow was created from the ServiceNow Developer Portal.
   - The instance was configured with admin privileges to allow application creation and customization.

2. **Creating an Update Set**

   - An **Update Set** was created to capture all the configurations, scripts, and changes made during development.
   - This helps in migrating the developed components between different instances if required.

3. **Creating a Table**

   - A custom table named **Educational Organisation** (or similar) was created to store student and teacher information.
   - Fields such as *Student ID*, *Student Name*, *Department*, *Course*, *Admission Date*, and *Grade* were added.
   - Relationships between related tables were established to manage linked data.

4. **Designing the Form Layout and Form Design**

   - The **Form Layout** was organized for a user-friendly interface.
   - The **Form Designer** was used to add and arrange fields logically for better data entry and viewing.
   - Sections and tabs were customized for clarity and usability.

5. **Number Maintenance Configuration**

   - Automatic numbering was configured using **Number Maintenance** to generate unique record IDs for each entry (e.g., STU0001, STU0002).
   - This ensures proper record tracking and consistency.

6. **Process Flow Creation**

   - Using **Flow Designer**, workflows were created to automate processes like admission approval, grade updates, and status tracking.
   - Notifications and task assignments were added to enhance communication and efficiency.

7. **Client Script Implementation**

   - **Client Scripts** were developed using JavaScript to add validations, dynamic field behaviors, and form interactivity.
   - Example: Auto-filling fields, restricting invalid data entries, and displaying alerts for missing information.

8. **Testing and Result Verification**

   - The developed application was tested by creating, updating, and viewing student records.
   - Automated workflows and scripts were verified for accuracy and functionality.
   - The final output was an efficient, user-friendly system for educational data management.

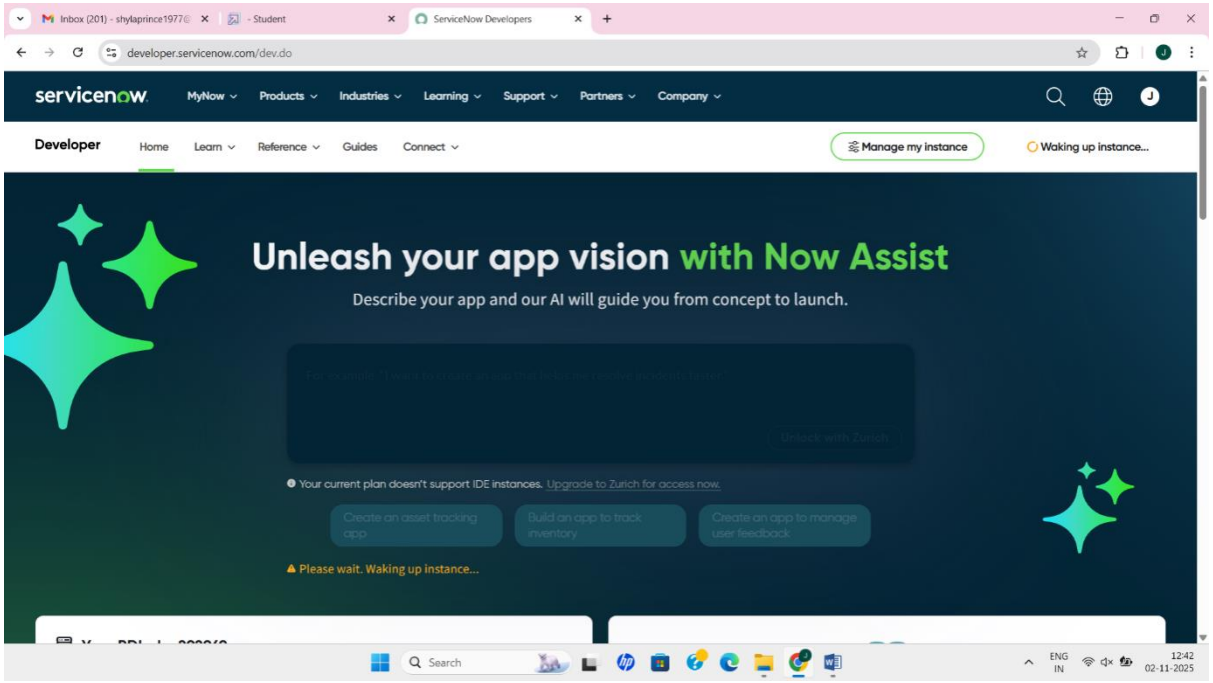
TOOLS AND TECHNOLOGIES USED:

- Service Now Platform
- Flow Designer
- Form Designer
- Client Scripts (JavaScript)
- Update Set Management

STEPS TO IMPLEMENT:

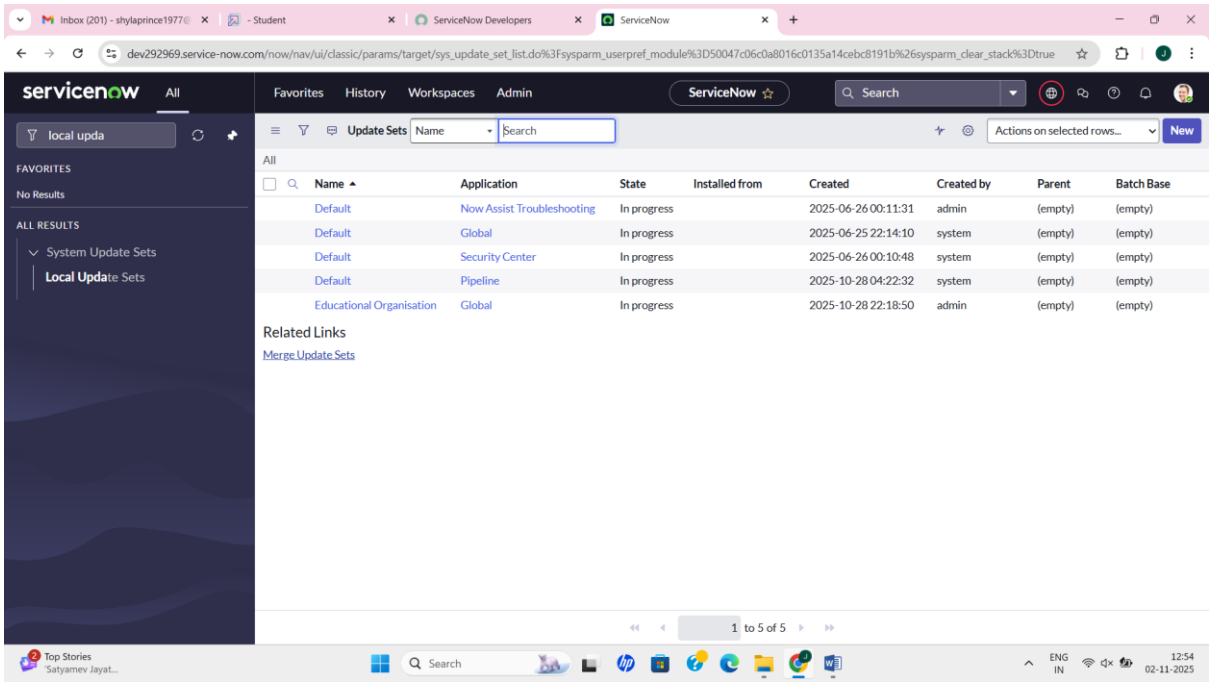
Step 1

Set up a service now instance



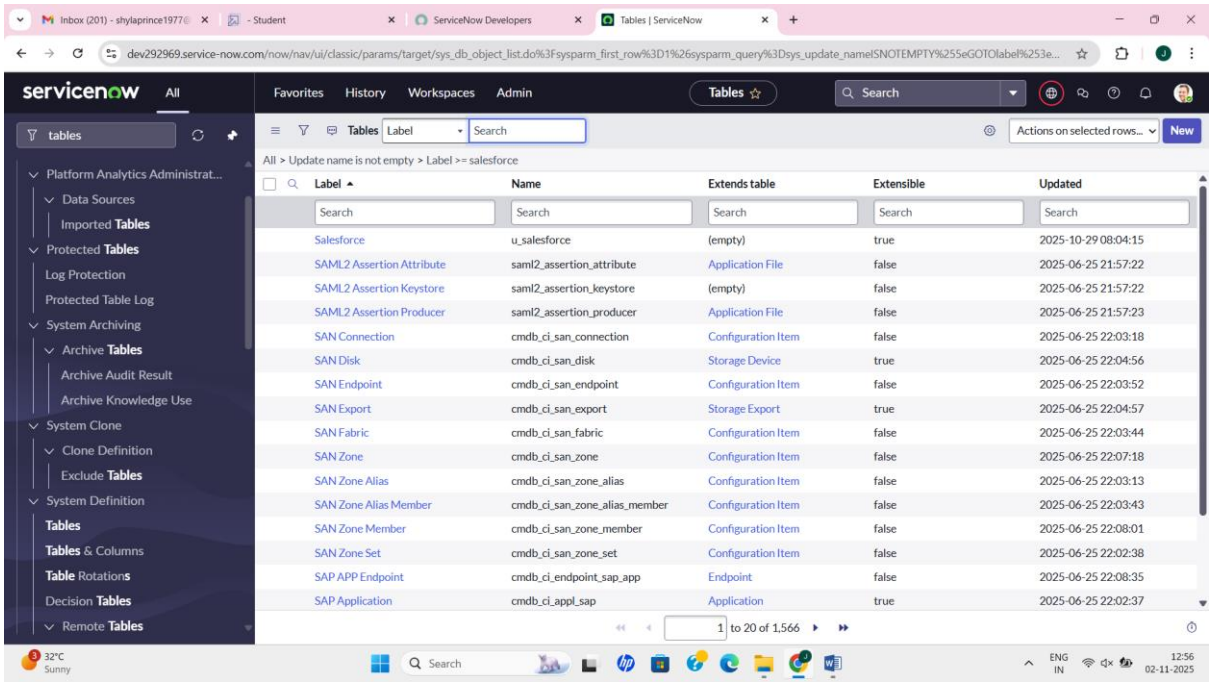
Step 2

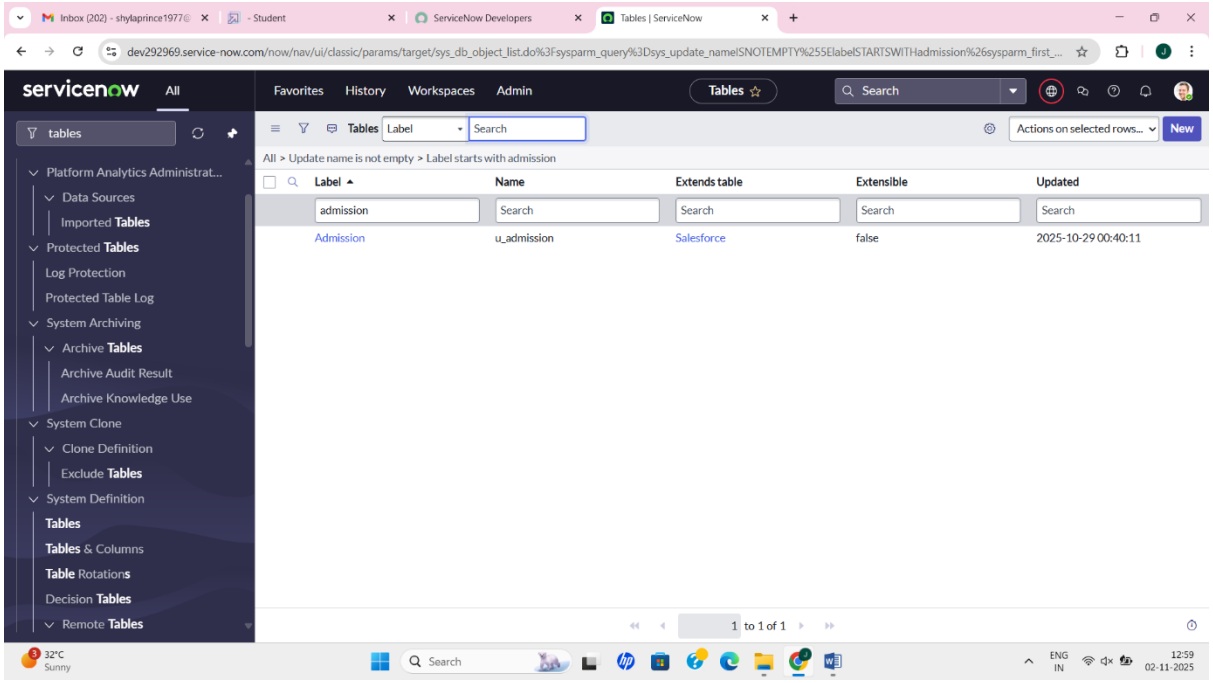
Create a update set on the local update set



Step 3

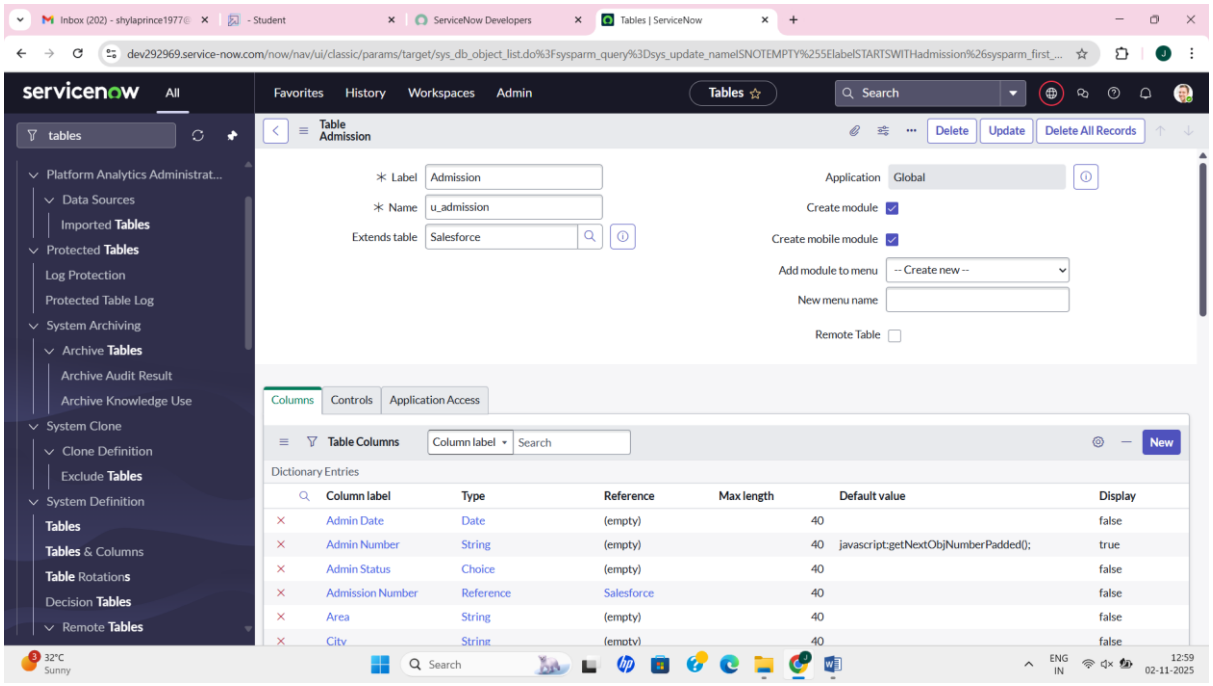
Create a tables in system definition name the tables as Admission,Salesforce and Student Progress





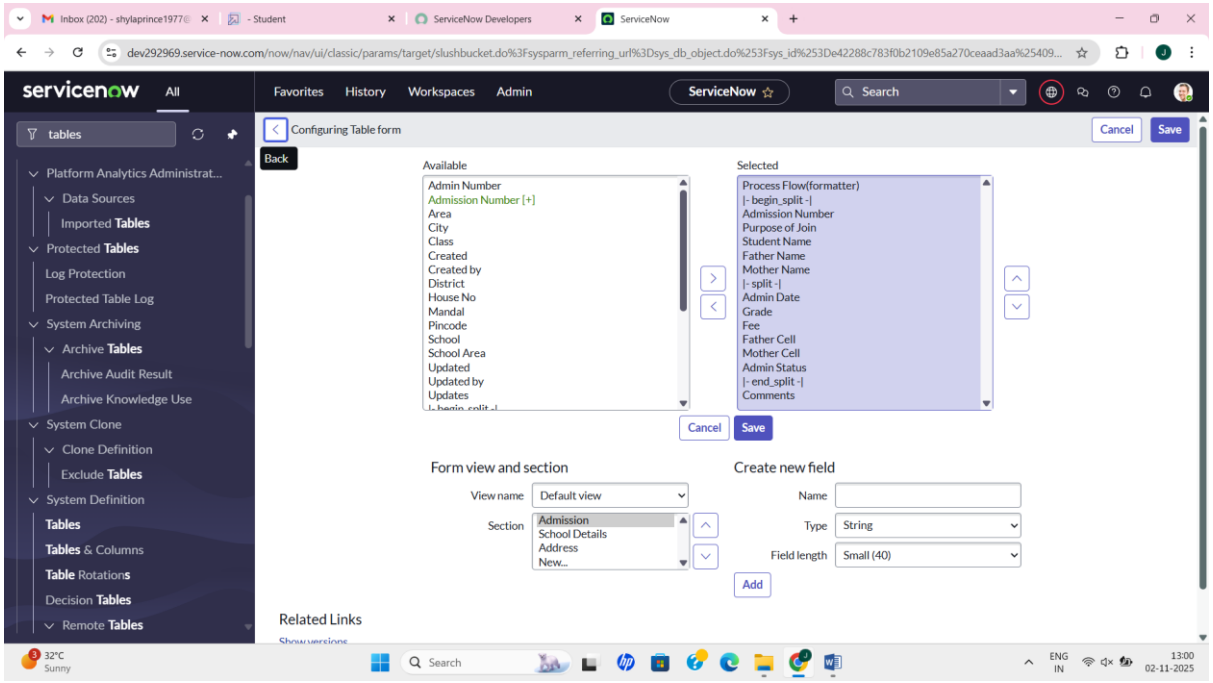
Step 4

Create the label for the tables



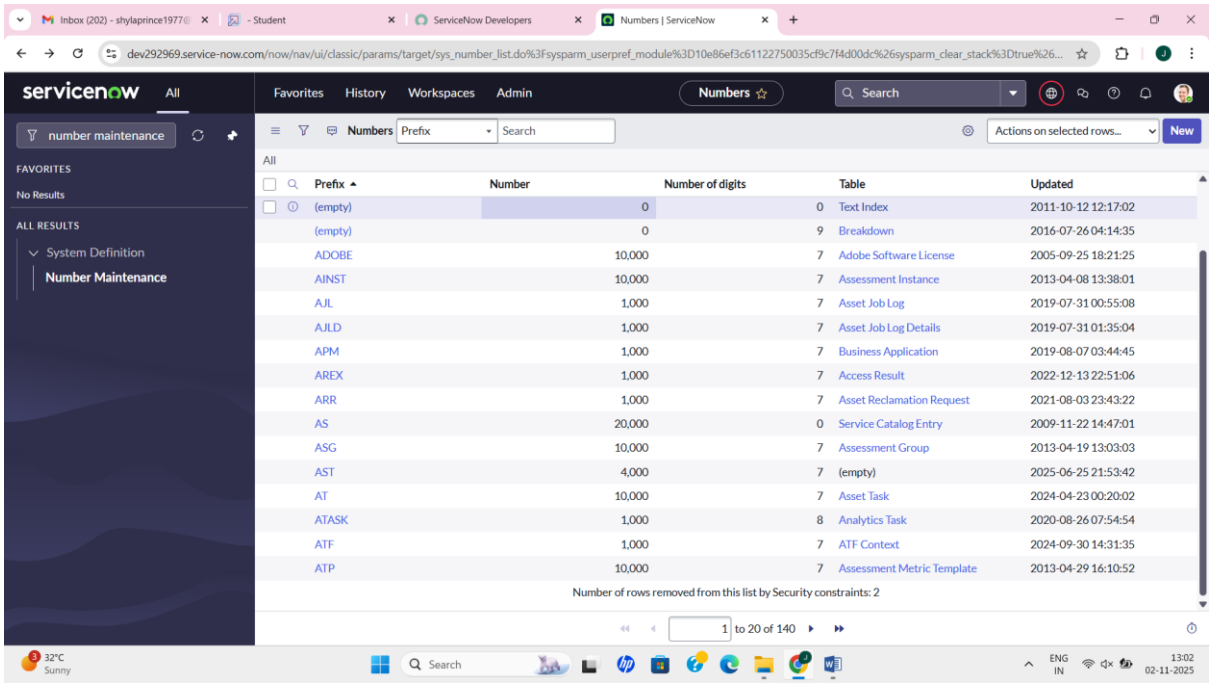
Step 5

Design the form for the table content then configure the table in the service now administrator



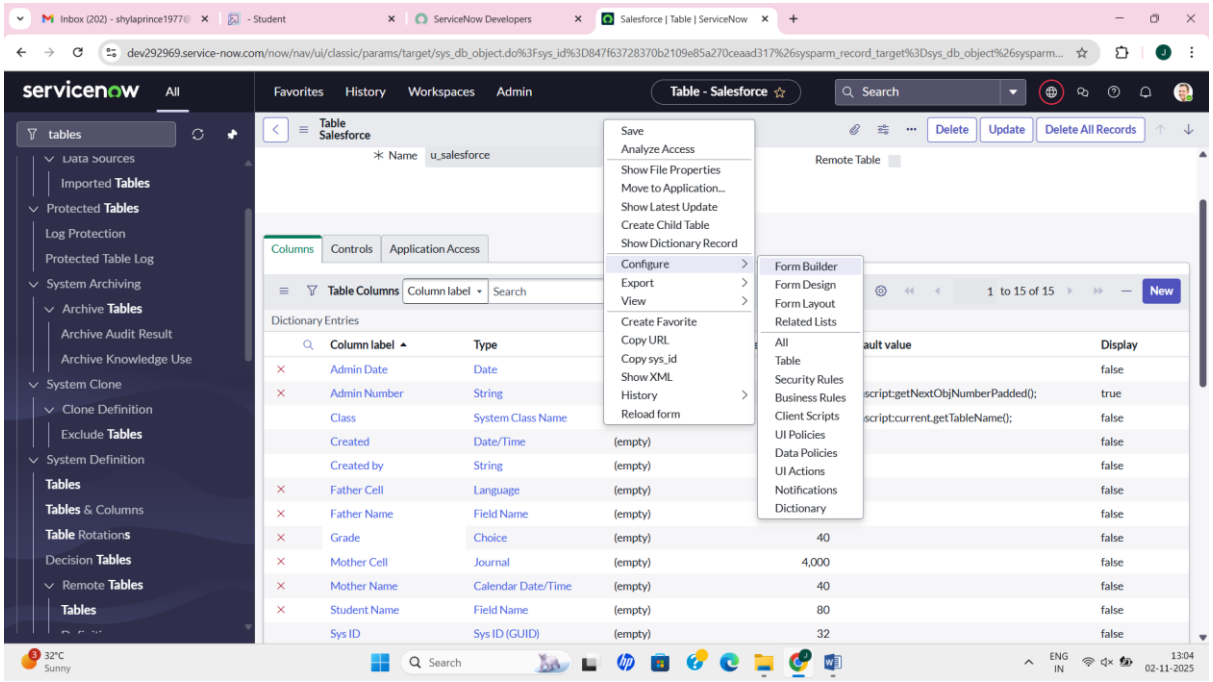
Step 6

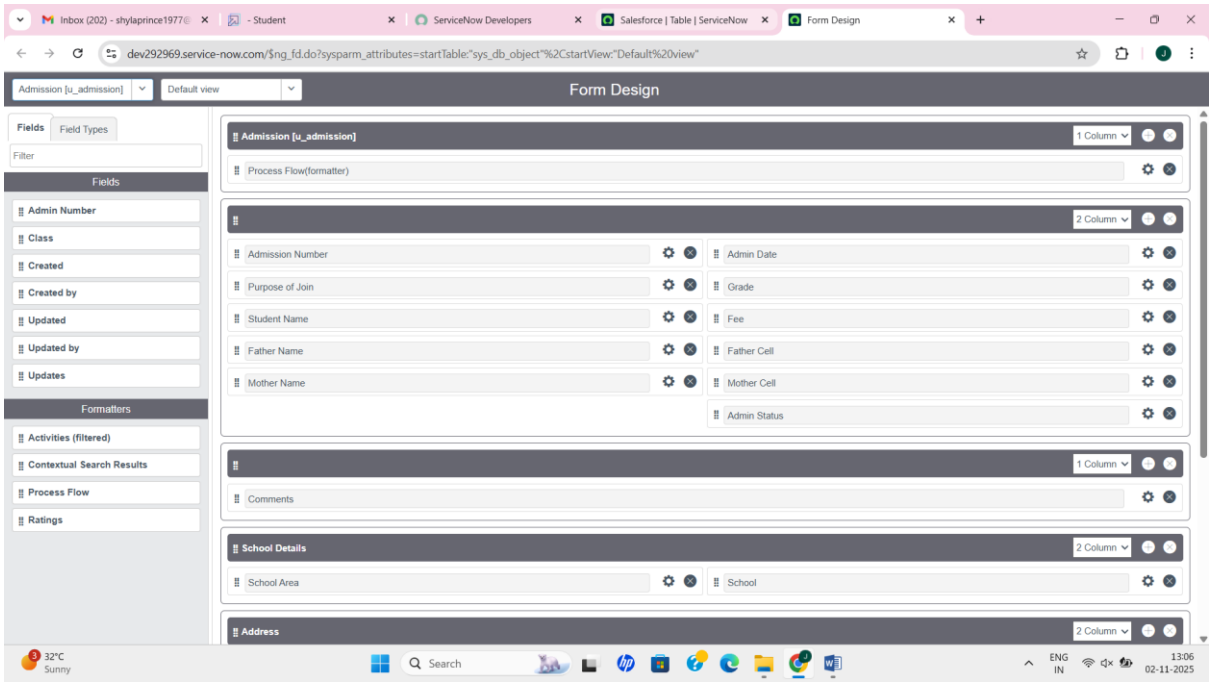
In the Service Now Administrator,Number Maintenance Configuration



Step 7

The process flow was created for designing the form and layout




Inbox (202) - shylaprince1977
- Student
ServiceNow Developers
Salesforce | Table | ServiceNow
Form Design
dev292969.service-now.com/$ng_fd.do?sysparm_attributes=startTable:"sys_db_object"%2CstartView:"Default%20view"
Admission [u_admission]
Default view
Form Design
Fields
Field Types
Filter
Fields
Admin Number
Class
Created
Created by
Updated
Updated by
Updates
Formatters
Activities (filtered)
Contextual Search Results
Process Flow
Ratings
Admission [u_admission]
1 Column
Process Flow(formatter)
2 Column
Admission Number
Admin Date
Purpose of Join
Grade
Student Name
Fee
Father Name
Father Cell
Mother Name
Mother Cell
Admin Status
1 Column
Comments
School Details
2 Column
School Area
School
Address
2 Column
32°C
Sunny
Search
ENG
IN
13:06
02-11-2025

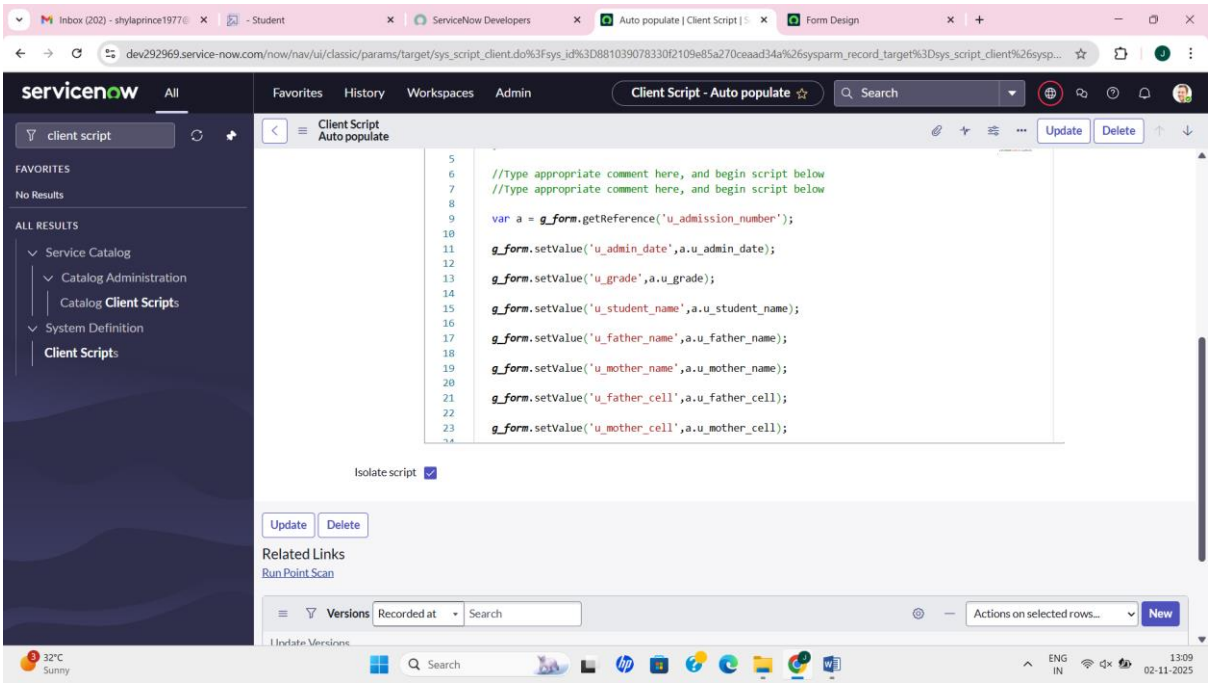
Step 8

Write the script for the client

```
functiononChange(control, oldValue, newValue, isLoading,
isTemplate) {
  if (isLoading || newValue === '') {
    return;
  }


  //Type appropriate comment here, and begin script below
  var a = g_form.getReference('u_admission_number');
  g_form.setValue('u_admin_date',a.u_admin_date);
  g_form.setValue('u_grade',a.u_grade);
  g_form.setValue('u_student_name',a.u_student_name);
  g_form.setValue('u_father_name',a.u_father_name);
  g_form.setValue('u_mother_name',a.u_mother_name);
  g_form.setValue('u_father_cell',a.u_father_cell);
  g_form.setValue('u_mother_cell',a.u_mother_cell);



  g_form.setDisabled('u_admin_date',a.u_admin_date);
  g_form.setDisabled('u_grade',a.u_grade);
  g_form.setDisabled('u_student_name',a.u_student_name);
  g_form.setDisabled('u_father_name',a.u_father_name);
  g_form.setDisabled('u_mother_name',a.u_mother_name);
  g_form.setDisabled('u_father_cell',a.u_father_cell);
  g_form.setDisabled('u_mother_cell',a.u_mother_cell);
}
```

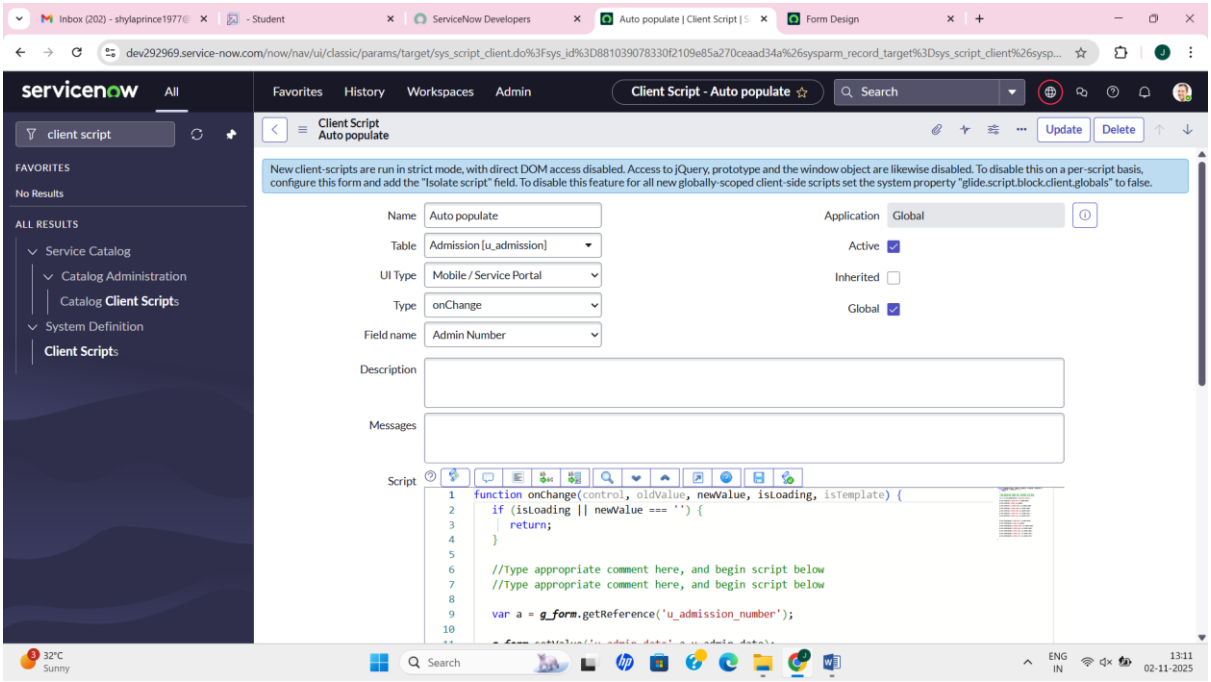
Step 9

The client script was isolated and saved



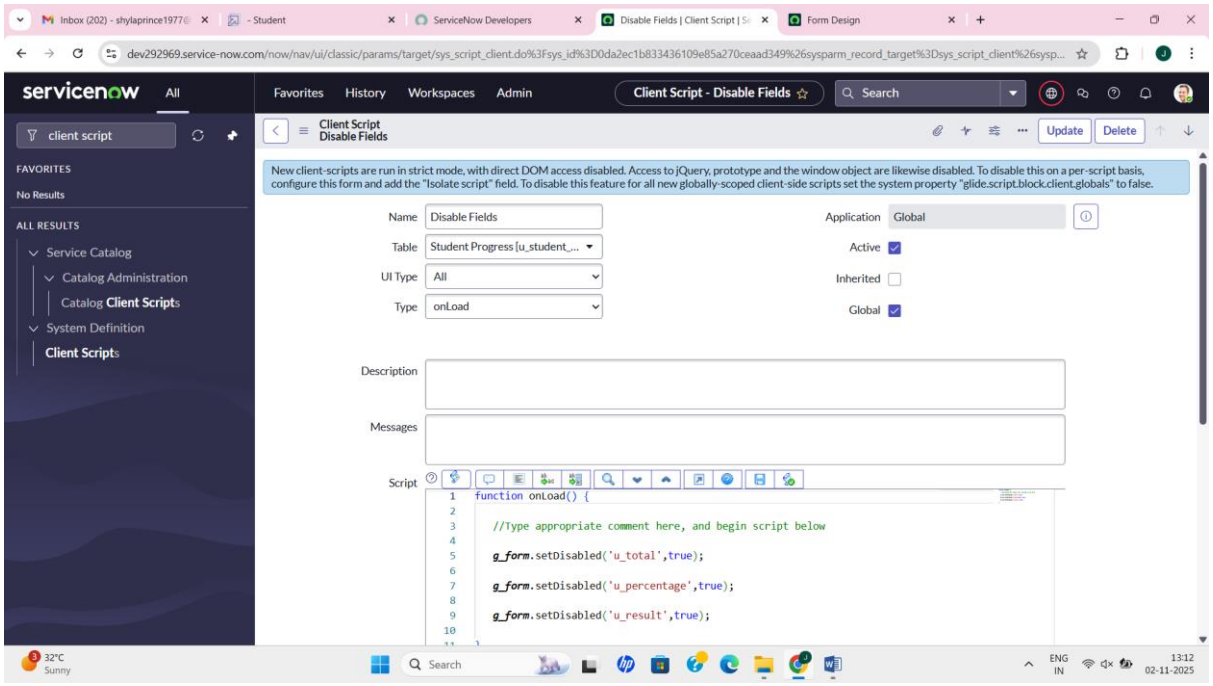
Step 10

The testing of the client script for the autopopulate in the admission table



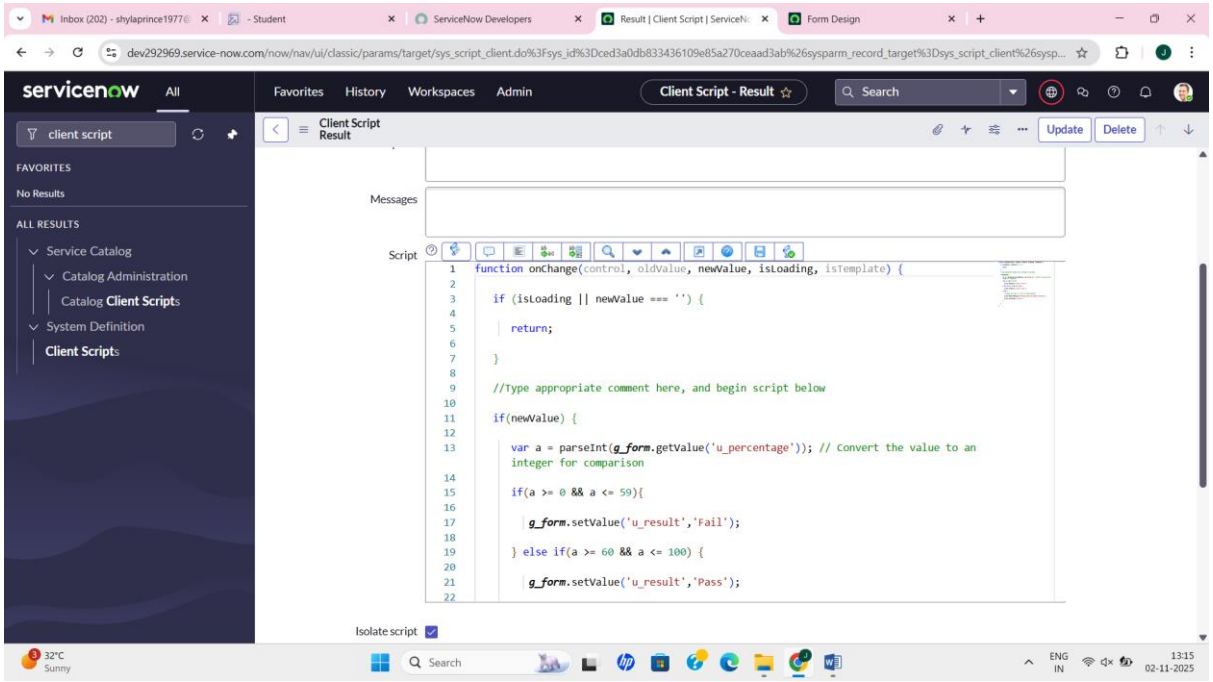
Step 11

Create the disable fields for student progress table



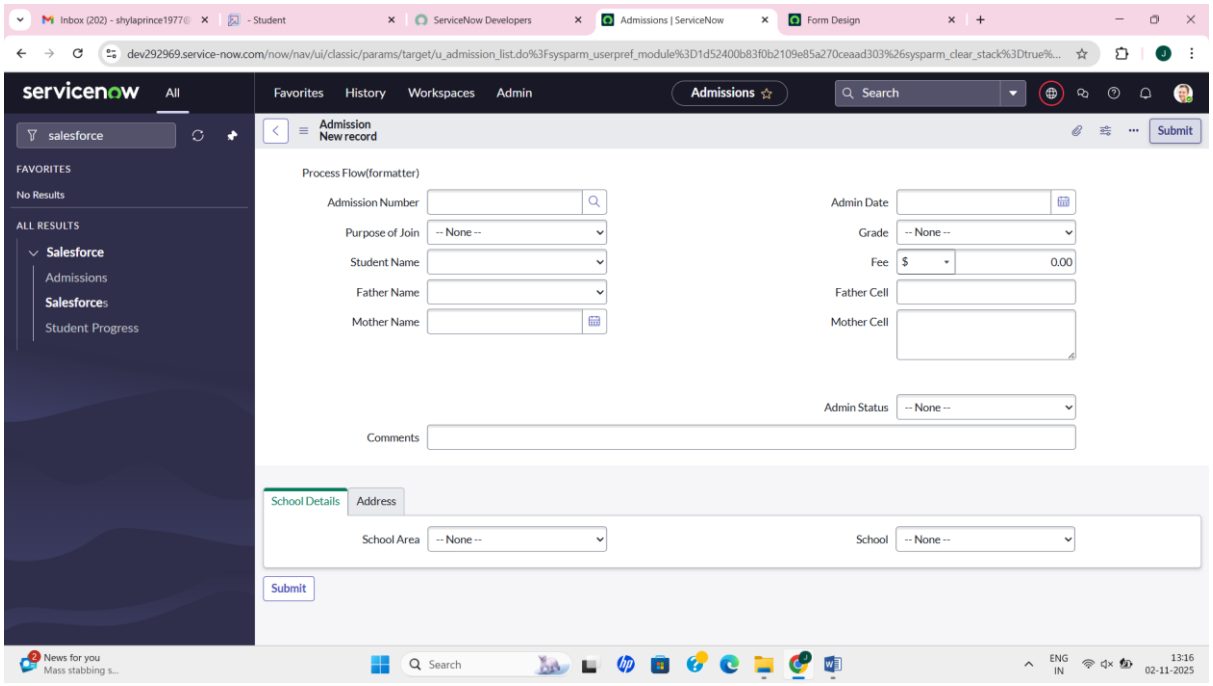
Step 12

Create result and percentage script for the student progress table

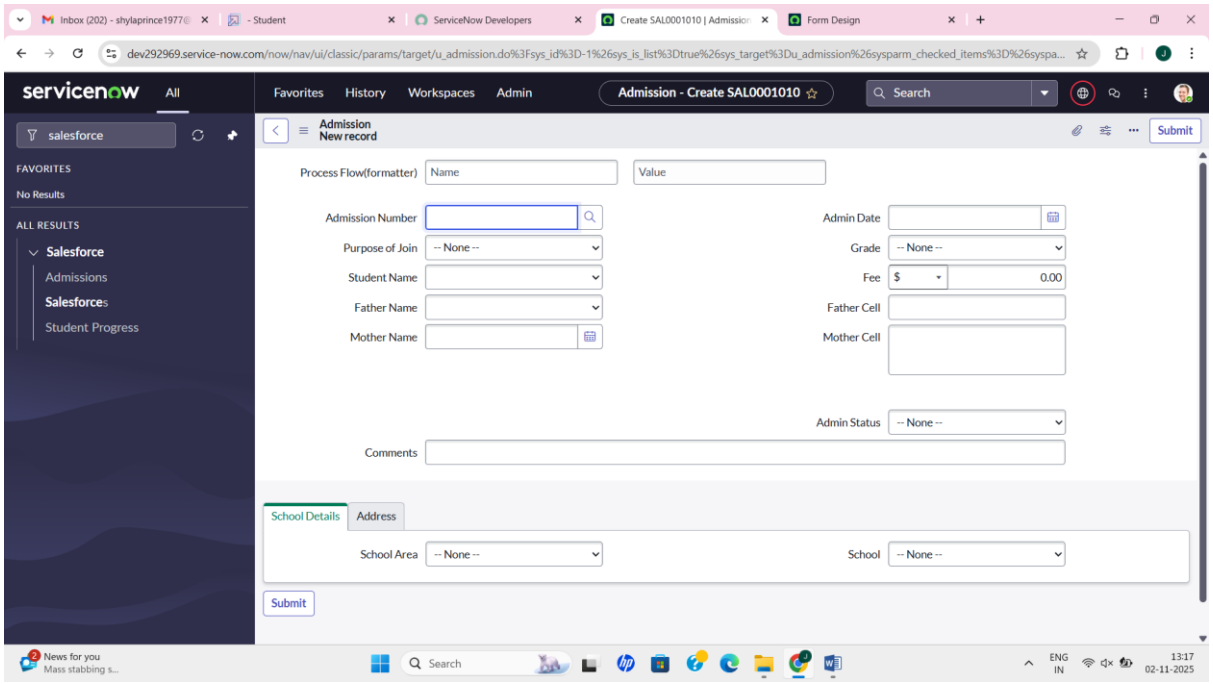


Step 13

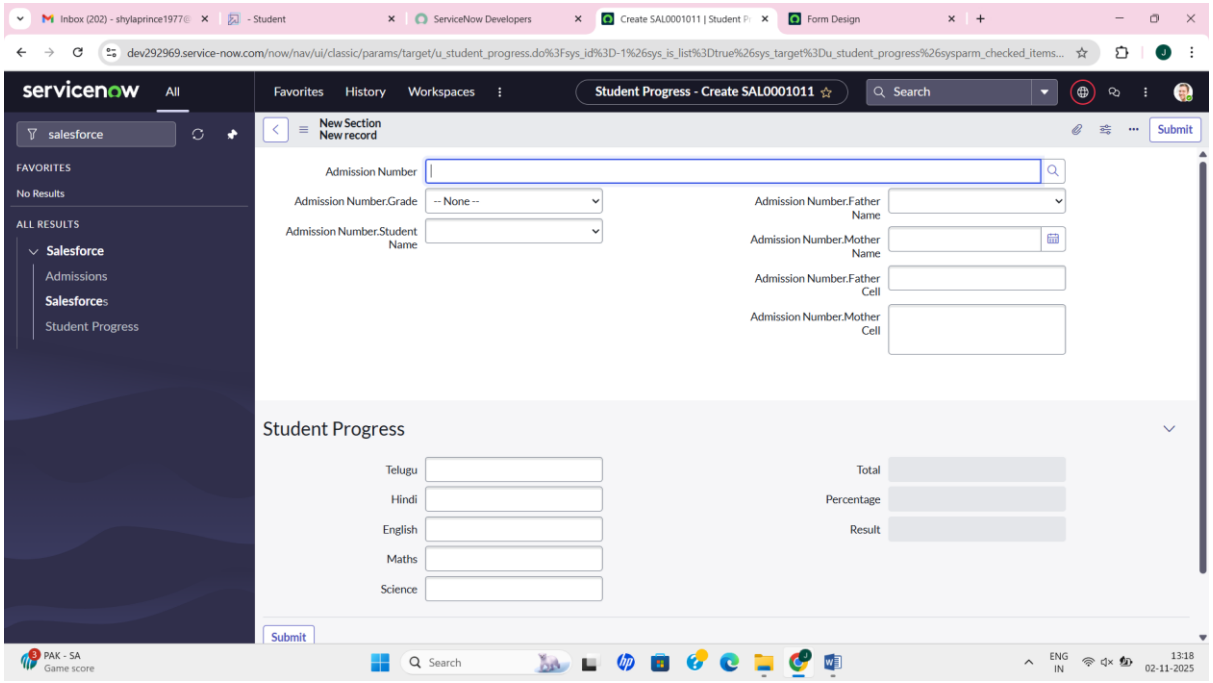
The testing and result verification can be output in the salesforce

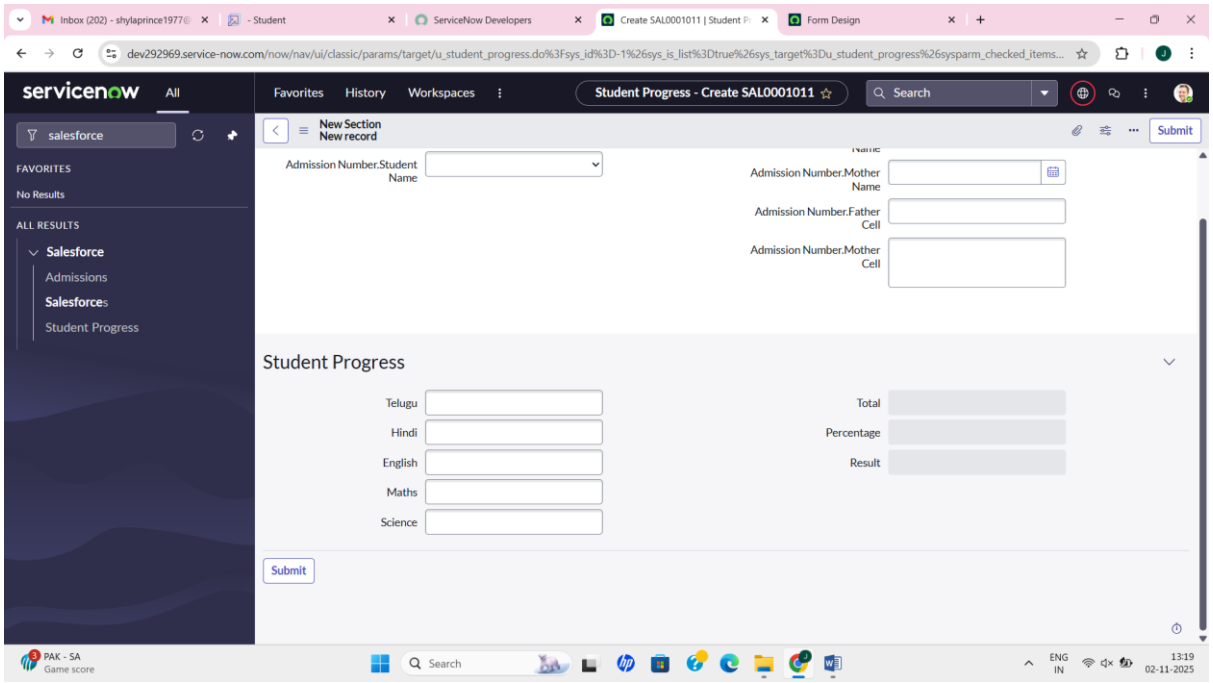


Admission:



Student Progress:





RESULT:

- A working **Educational Management Application** was successfully developed on ServiceNow.
- It enables:
    - Easy record creation and tracking
    - Workflow automation
    - Real-time data management for educational institutions
- The project demonstrates how ServiceNow can be used beyond IT services — to manage education-related operations efficiently.

CONCLUSION:

This project successfully demonstrates the implementation of a basic Educational Management System using ServiceNow.Through this project, the administrative activities such as maintaining student details, admissions, and grades are automated and streamlined.It shows the flexibility of ServiceNow as a platform for developing custom enterprise applications in different domains.