

# Assignment & Exam Management System

---

The **Assignment & Exam Management System** is a Salesforce-based CRM solution designed to streamline the management of academic workflows in educational institutions. It empowers teachers to create and manage assignments and exams, while enabling students to submit assignments, track grades, and receive timely deadline notifications. Key features include automated grade calculation, reminders for upcoming deadlines, and real-time reports and dashboards for tracking student performance and grade trends. This CRM addresses the pressing business need for reduced manual effort, greater accountability, and enhanced transparency in academic performance management.

The primary objective of this CRM is to centralize academic task management by offering a unified platform for assignment and exam workflows. It simplifies student submissions, automates grading, and enhances performance tracking, ensuring accuracy and efficiency for all stakeholders.

Automated notifications improve student accountability by reducing missed deadlines, while comprehensive reports and dashboards provide actionable insights for administrators and teachers. By leveraging Salesforce automation and analytics, the CRM delivers business value through improved operational efficiency, better student engagement, and informed academic decision-making.

# Phase 1:

## Requirement Gathering

- Understand the needs of students, teachers, and academic administrators.
- Identify functionalities such as:
  - Assignment submission and tracking.
  - Grade recording and tracking.
  - Performance reports and analytics.
- Ensure system supports real-time updates for accuracy and efficiency.

## Stakeholder Analysis

- Key stakeholders include:
  - Students: Need access to assignment submissions, grades, and timely alerts.
  - Teachers: Require tools to create, manage, and evaluate assignments and exams efficiently.
  - Academic Administrators: Need detailed performance reports, analytics, and monitoring tools.
- System must balance needs of all stakeholders with a user-friendly interface.

## Business Process Mapping

- Mapped academic workflows to ensure smooth operations.
- Processes include:
  - Assignment creation by teachers.
  - Submission of assignments by students.
  - Grading by teachers and feedback sharing.
  - Automated generation of performance reports.
  - Communication between stakeholders through system alerts.
- Automation ensures reduced manual work and improved efficiency.

## Industry-specific Use Case Analysis

- Focused on solving challenges in educational institutions.
- Analyzed common use cases such as:
  - Timely reminders for upcoming submissions and exams.
  - Automated grade calculations and publishing.
  - Student engagement through progress dashboards.
  - Early intervention for underperforming students via alerts.
- Designed to increase transparency and accountability.

## AppExchange Exploration

- Explored Salesforce AppExchange for add-ons and integrations.

- Identified useful tools such as:
  - Third-party SMS and Email services for alerts and reminders.
  - Analytics tools for advanced reporting and insights.
  - Document management and collaboration apps for better assignment handling.
  - Helps extend system capabilities beyond core Salesforce features.

## Conclusion

- Phase 1 successfully established the foundation for the Assignment & Exam Management System.
- Requirements were gathered with detailed focus on students, teachers, and administrators.
- Business processes were mapped to ensure automation and streamlined workflows.
- Industry-specific use cases were identified to address challenges like performance tracking and engagement.
- AppExchange integrations were explored to expand system functionality.

## Phase 2: Org Setup & Configuration

### Salesforce Editions - Developer Org

Set up the Salesforce Developer Org to build and test functionalities tailored to academic workflows.

*Process Flow: Setup → Salesforce Editions → Developer Org*

#### 1. Company Profile Setup

Configured the company profile with the following details:

- **Organization Name:** Prof. Ram Meghe Institute of Technology & Research
- **Default Locale:** English (India)
- **Default Currency:** INR
- **Default Time Zone:** (GMT+05:30) India Standard Time

The screenshot shows the 'Company Information' setup page under the 'SETUP' tab. The page title is 'Company Information' and the sub-section is 'Prof.Ram Meghe Institute Of Technology and Research'. A note says 'The organization's profile is below.' Below are tabs for 'User Licenses', 'Permission Set Licenses', 'Feature Licenses', and 'Usage-based Entitlements'. The 'Organization Detail' section contains the following data:

Organization Detail		Edit	
Organization Name	Prof.Ram Meghe Institute Of Technology and Research	Phone	
Primary Contact	OrgFarm EPIC	Fax	
Division		Default Locale	English (India)
Address	India	Default Language	English
Fiscal Year Starts In	Custom Fiscal Year	Default Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
Activate Multiple Currencies	<input type="checkbox"/>	Currency Locale	Hindi (India) - INR
Enable Data Translation	<input type="checkbox"/>	Used Data Space	501 KB (10%) <a href="#">[View]</a>
Newsletter	<input checked="" type="checkbox"/>	Used File Space	17 KB (0%) <a href="#">[View]</a>
Admin Newsletter	<input checked="" type="checkbox"/>	API Requests, Last 24 Hours	0 (15,000 max)
Hide Notices About System Maintenance	<input type="checkbox"/>	Streaming API Events, Last 24 Hours	0 (10,000 max)
Hide Notices About System Downtime	<input type="checkbox"/>	Restricted Logins, Current Month	0 (0 max)
Locale Formats	ICU	Salesforce.com Organization ID	00DgK000007ZvaW
		Organization Edition	Developer Edition
		Instance	CAN96

*Process Flow: Setup → Company Information → Edit*

## 2. Business Hours & Holidays

Created a Business Hours setting named 'School Hours' with all days available 9:00AM to 5:00PM hours to accommodate academic activities at any time.

The screenshot shows the 'Business Hours' setup page in Salesforce. At the top, there's a header with a gear icon labeled 'SETUP' and the title 'Business Hours'. Below the header, a section titled 'Organization Business Hours' contains instructions: 'Select the days and hours that your support team is available. These hours, when associated with escalation rules, determine the times at which cases can escalate.' A note says, 'If you enter blank business hours for a day, that means your organization does not operate on that day.' The main table, titled 'Business Hours Detail', shows the following data:

Business Hours Name	School Hours	Time Zone	Default Business Hours
Business Hours	Sunday: 9:00 AM to 5:00 PM Monday: 9:00 AM to 5:00 PM Tuesday: 9:00 AM to 5:00 PM Wednesday: 9:00 AM to 5:00 PM Thursday: 9:00 AM to 5:00 PM Friday: 9:00 AM to 5:00 PM Saturday: 9:00 AM to 5:00 PM	(GMT+05:30) India Standard Time (Asia/Kolkata)	<input type="checkbox"/>

Below the table, it says 'Active' with a checkbox, 'Created By' (Shruti Misra), 'Created Date' (9/15/2025, 4:57 AM), 'Last Modified By' (Shruti Misra), and 'Last Modified Date' (9/16/2025, 6:28 AM). There are 'Edit' and 'Add' buttons. A 'Holidays [0]' section is present, with a note 'No records to display'.

*Process Flow: Setup → Business Hours → New Business Hours*

## 3. Fiscal Year Settings

Defined the fiscal year from April 1, 2025, to March 31, 2026, aligning financial tracking and reporting to academic schedules.

The screenshot shows the 'Custom Fiscal Year Definition' page in Salesforce. At the top, there's a header with a gear icon labeled 'SETUP' and the title 'Custom Fiscal Year Definition'. Below the header, a note says, 'This page allows you to define a custom fiscal year record. Dates for the custom fiscal year will be generated based on the choices you make below. If you need to customize your fiscal year, expand the Advanced Customization section.' The main form has sections for 'Fiscal Year Information' and 'Reset Fiscal Year Structure'. In 'Fiscal Year Information', the following values are set:

Fiscal Year Start Date	4/1/2025
Fiscal Year Name	2026
Description	(empty text area)
Week Start Day	Tuesday

A note 'I Required Information' is shown next to the 'Fiscal Year Start Date' field. At the bottom, there are 'Preview', 'Save', and 'Cancel' buttons.

*Process Flow: Setup → Fiscal Year → Define*

## 4. User Setup & Licenses

Configured user roles, profiles, and licenses:

- Teacher User:
  - License: Salesforce
  - Profile: Teacher\_Profile
- Student User:
  - License: Salesforce
  - Profile: Student\_Profile
- Guardian User:
  - License: Salesforce
  - Profile: Guardian\_Profile

The screenshot shows the Salesforce Setup interface for cloning a profile. At the top, there's a header with a user icon and the word "SETUP". Below it, a "Profiles" section is visible. The main content area is titled "Clone Profile" and contains a sub-header "Enter the name of the new profile." A message box says "You must select an existing profile to clone from." Below this, there are three radio buttons: "Existing Profile" (selected), "Standard User", and "User License" (set to "Salesforce"). A text input field labeled "Profile Name" contains the value "Teacher\_Profile". At the bottom, there are "Save" and "Cancel" buttons.

SETUP

Users

### New User

Help for this Page 

User Edit Save | Save & New | Cancel

**General Information** ! Required Information

First Name	Harshal	Role	<None Specified>
Last Name	Alaspure	User License	Salesforce Platform
Alias	halas	Profile	Standard Platform User
Email		Active	<input checked="" type="checkbox"/>
Username		Marketing User	<input type="checkbox"/>
Nickname		Offline User	<input type="checkbox"/>
Title		Knowledge User	<input type="checkbox"/>
Company		Flow User	<input type="checkbox"/>
Department		Service Cloud User	<input type="checkbox"/>
Division		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	Default Limit (300)

*Process Flow: Setup → Users → New User → Assign Role/Profile*

---

## 5. Roles & Hierarchy

Defined hierarchy to control visibility and reporting structure in the academic system:

- **Academic\_Admin**  
↓
- **Teacher**  
↓
- **Student**

**Guardian** role is placed under **Admin** for independent visibility of child's records.

The screenshot shows a software interface for managing roles. At the top, there is a header with a user icon and the text 'SETUP' and 'Roles'. Below the header, the title 'Creating the Role Hierarchy' is displayed. A sub-header 'Your Organization's Role Hierarchy' follows. Under this, there is a tree view of roles. A collapse/expand button 'Collapse All Expand All' is at the top of the tree. The tree structure is as follows:

- Prof.Ram Meghe Institute Of Technology and Research
  - Add Role
  - Academic\_Admin
    - Edit | Del | Assign
    - Add Role
  - Guardian
    - Edit | Del | Assign
    - Add Role
  - Teacher
    - Edit | Del | Assign
    - Add Role
  - Student
    - Edit | Del | Assign

*Process Flow: Setup → Roles → New Role → Define Hierarchy*

## 6. Permission Sets

Created permission sets to manage access and rights:

- PS\_Grade\_Manager: Extra edit rights on Grade\_c
- Submission\_Reviewer: Manage rights on Submission\_c (assigned to Teacher user Anita)
- PS\_Assignment: Manage rights on Assignment\_c (assigned to Teacher user Anita)

The screenshot shows the 'Permission Sets' page in the Salesforce Setup. A specific permission set named 'PS\_Grade\_Manager' is selected. The 'Permission Set Overview' section displays the following details:

- Description:** Allows teachers to create/edit/delete Grade\_\_c records
- API Name:** PS\_Grade\_Manager
- Namespace Prefix:** ShruMisal
- Created By:** Shru Misal (9/14/2025, 8:23 AM)
- Last Modified By:** Shru Misal (9/15/2025, 5:29 AM)
- License:** Standard
- Session Activation Required:** Not checked
- Permission Set Groups Added To:** 0

The 'Apps' section lists various settings under 'Assigned Apps' and 'Assigned Connected Apps'. Other sections like 'Object Settings', 'App Permissions', 'Apex Class Access', and 'Visualforce Page Access' are also visible.

*Process Flow: Setup → Permission Sets → New → Assign*

## 7. Organization-Wide Defaults (OWD)

Set sharing model to ensure privacy and access control:

- Student\_\_c = Private
- Guardian\_\_c = Private
- Assignment\_\_c = Private
- Submission\_\_c = Private
- Grade\_\_c = Private
- Course\_\_c = Public Read Only

The screenshot shows the 'Sharing Settings' page in the Salesforce Setup. It displays a grid of sharing rules for various objects. Most objects are set to 'Private' sharing, except for 'Course' which is set to 'Public Read Only'. The 'Other Settings' section includes checkboxes for 'Standard Report Visibility', 'Manual User Record Sharing', 'Manager Groups', 'Secure guest user record access', and 'Require permission to view record names in lookup fields'.

*Process Flow: Setup → Sharing Settings → Edit Defaults*

## 8. Sharing Rules

Implemented sharing rules for effective collaboration and visibility:

- Grades (Grade\_c): Shared with Guardian role to view child's records
- Submissions (Submission\_c): Shared with Teacher role for management
- Assignments: Configured to ensure proper access as per role hierarchy

The screenshot shows the 'Sharing Settings' page in Salesforce. A new sharing rule is being created with the following details:

Label	Share_Grades_to_Guardian
Rule Name	Share_Grades_to_Guardian
Description	(empty)
Grade: owned by members of	Role: Student
Share with	Role: Guardian
Access Level	Read Only
Created By	Shruti Misal, 9/15/2025, 6:06 AM

At the bottom right of the form, it says 'Modified By Shruti Misal, 9/15/2025, 6:06 AM'. Below the form are 'Save' and 'Cancel' buttons.

*Process Flow: Setup → Sharing Settings → New Sharing Rule*

## 9. Outcomes of Phase 2

This phase successfully established the Salesforce environment with proper configurations for users, roles, profiles, permissions, and data-sharing policies. It ensures secure, role-based access while enabling seamless academic workflows, timely communication, and transparent management of student records.

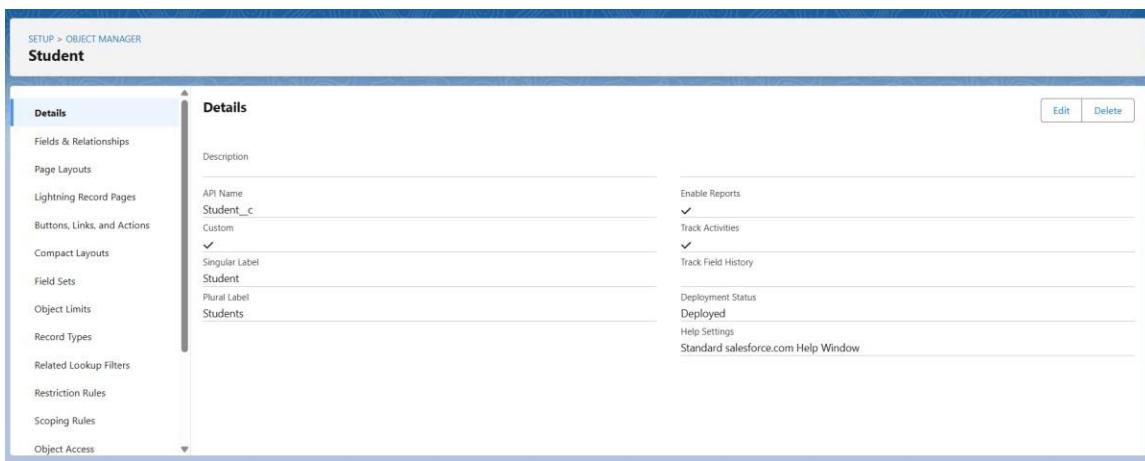
# Phase 3: Data Modeling & Relationships

## 1. Standard & Custom Objects

Custom objects created for the Assignment & Exam Management System include:

### Student\_c

- Student ID (Text, External ID, Unique)
- First Name, Last Name, Email, Phone
- Guardian (Lookup), Guardian Name, Guardian Contact
- Owner, Created By, Modified By



### Guardian\_c

- Guardian ID (Auto Number, External ID, Unique)
- First Name, Last Name, Email (Unique), Phone
- Relationship to Student (Picklist)
- Owner, Created By, Modified By

SETUP > OBJECT MANAGER  
**Guardian**

Details	
Fields & Relationships	Description
Page Layouts	
Lightning Record Pages	
Buttons, Links, and Actions	
Compact Layouts	
Field Sets	
Object Limits	
Record Types	
Related Lookup Filters	
Restriction Rules	
Scoping Rules	
Object Access	
<a href="#">Edit</a> <a href="#">Delete</a>	
<b>Details</b> API Name: <b>Guardian_c</b> Custom <input checked="" type="checkbox"/> Singular Label: <b>Guardian</b> <input checked="" type="checkbox"/> Plural Label: <b>Guardians</b>	
Enable Reports <input checked="" type="checkbox"/> Track Activities <input checked="" type="checkbox"/> Track Field History <input checked="" type="checkbox"/> Deployment Status: <b>Deployed</b> Help Settings: <b>Standard salesforce.com Help Window</b>	

## Course\_c

- Course Code (Unique), Name (Indexed)
- Credits, Instructor (Lookup), Schedule Info, Title
- Owner, Created By, Modified By

SETUP > OBJECT MANAGER  
**Course**

Details	
Fields & Relationships	Description
Page Layouts	
Lightning Record Pages	
Buttons, Links, and Actions	
Compact Layouts	
Field Sets	
Object Limits	
Record Types	
Related Lookup Filters	
Restriction Rules	
Scoping Rules	
Object Access	
<a href="#">Edit</a> <a href="#">Delete</a>	
<b>Details</b> API Name: <b>Course_c</b> Custom <input checked="" type="checkbox"/> Singular Label: <b>Course</b> <input checked="" type="checkbox"/> Plural Label: <b>Courses</b>	
Enable Reports <input checked="" type="checkbox"/> Track Activities <input checked="" type="checkbox"/> Track Field History <input checked="" type="checkbox"/> Deployment Status: <b>Deployed</b> Help Settings: <b>Standard salesforce.com Help Window</b>	

## Assignments\_c

- Assignment Name (Text)
- Course (Lookup Course), Description (Long Text), Due Date
- Max Marks, Title
- Owner, Created By, Modified By

SETUP > OBJECT MANAGER

### Assignment

<b>Details</b>	<b>Details</b>	<b>Actions</b>
Fields & Relationships	Description	<a href="#">Edit</a> <a href="#">Delete</a>
Page Layouts		
Lightning Record Pages		
Buttons, Links, and Actions		
Compact Layouts		
Field Sets		
Object Limits		
Record Types		
Related Lookup Filters		
Restriction Rules		
Scoping Rules		
Object Access		

**API Name:** Assignment\_\_c  
**Custom:** ✓  
**Singular Label:** Assignment  
**Plural Label:** Assignments

**Enable Reports:**  
✓ Track Activities  
✓ Track Field History  
✓ Deployment Status: Deployed  
Help Settings: Standard salesforce.com Help Window

## Grade\_c

- Grade Number (Auto Number)
- Enrollment (Lookup), Student (Lookup), Subject (Lookup Course)
- Teacher (Lookup), Marks, Grade Date
- Owner, Created By, Modified By

SETUP > OBJECT MANAGER

### Grade

<b>Details</b>	<b>Details</b>	<b>Actions</b>
Fields & Relationships	Description	<a href="#">Edit</a> <a href="#">Delete</a>
Page Layouts		
Lightning Record Pages		
Buttons, Links, and Actions		
Compact Layouts		
Field Sets		
Object Limits		
Record Types		
Related Lookup Filters		
Restriction Rules		
Scoping Rules		
Object Access		

**API Name:** Grade\_\_c  
**Custom:** ✓  
**Singular Label:** Grade  
**Plural Label:** Grades

**Enable Reports:**  
✓ Track Activities  
✓ Track Field History  
Deployment Status: Deployed  
Help Settings: Standard salesforce.com Help Window

## Exam\_c

- Exam Name (Text), Exam Date, Exam Type (Picklist)
- Duration, Max Marks
- Course (Lookup Course)
- Owner, Created By, Modified By

SETUP > OBJECT MANAGER  
**Exam**

**Details**

Description

API Name: Exam\_\_c  
Custom  
Singular Label: Exam  
Plural Label: Exams

Enable Reports  
✓  
Track Activities  
✓  
Track Field History  
✓  
Deployment Status: Deployed  
Help Settings: Standard salesforce.com Help Window

Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Restriction Rules  
Scoping Rules  
Object Access

Edit Delete

## Submissions\_c

- Submission Number (Auto Number), Submitted By (Lookup Student), Submitted Date
- Assignment (Lookup), Exam (Lookup)
- Owner, Created By, Modified By

**Process Flow:** Setup → Object Manager → New Custom Object → Add Fields. All objects were configured in the same way, with their respective fields as listed above.

SETUP > OBJECT MANAGER  
**Submission**

**Details**

Description

API Name: Submission\_\_c  
Custom  
Singular Label: Submission  
Plural Label: Submissions

Enable Reports  
✓  
Track Activities  
✓  
Track Field History  
✓  
Deployment Status: Deployed  
Help Settings: Standard salesforce.com Help Window

Fields & Relationships  
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Restriction Rules  
Scoping Rules  
Object Access

Edit Delete

## 2. Page Layouts

Configured for Student\_\_c as an example:

**Process Flow:** Setup → Object Manager → Student → Page Layouts → Student Layout → Drag fields into sections → Save.

Student Layout: Student Information section (Student ID, First Name, Last Name, Email, Phone), Guardian Info section (Guardian Name, Guardian Contact).

Related Lists added: Guardians, Enrollments, Grades.

Other objects (Guardian, Course, Assignment, Exam, Submissions, Grade) were configured similarly with relevant fields and related lists.

**Student Page Layout Configuration:**

- Fields:** Section, Blank Space, First Name, Last Modified By, Student ID, Last Name, Owner, Student Name, Created By, Guardian Contact, Email, Guardian Name, Phone.
- Student Sample Panel:** Highlights Panel, Quick Actions in the Salesforce Classic Publisher, Salesforce Mobile and Lightning Experience Actions.

**Guardian Page Layout Configuration:**

- Fields:** Section, Blank Space, First Name, Last Name, Last Modified By, Relationship to S...
- Information Panel:** Guardian Name (Sample Text), Guardian ID (GEN-2004-001234), First Name (Sample Text), Last Name (Sample Text), Email (sarah.sample@company.com), Phone (1-415-555-1212).
- System Information Panel:** Created By (Sample Text), Last Modified By (Sample Text).
- Custom Links Panel:** Mobile Cards (Salesforce mobile only).

**Assignment Page Layout Configuration:**

- Fields:** Section, Blank Space, Created By, Max Marks, Description, Owner, Title, Assignment Name, Due Date, Last Modified By, Course.
- Information Panel:** Assignment Name (Sample Text), Title (Sample Text), Description (Sample Text), Due Date (9/16/2025), Max Marks (98.145), Course (Sample Text).
- System Information Panel:** Created By (Sample Text), Last Modified By (Sample Text).
- Custom Links Panel:** Mobile Cards (Salesforce mobile only).

### **3. Lookup vs Master-Detail vs Hierarchical Relationships**

Lookup Relationship: Assignment linked to Course via Lookup.

Master-Detail Relationship: Exam- links to Student and Course.

Hierarchical Relationship: User roles such as Teacher under Academic\_Admin.

# Phase 4: Process Automation (Admin)

In this phase, we focus on implementing automation within Salesforce to ensure data integrity, reduce manual work, and improve efficiency. The core components include Validation Rules, Workflow Rules, Flow Builder, and Email Alerts. Each of these elements is described below along with process flows for their creation.

## Validation Rules

Validation rules enforce consistent and accurate data entry by applying conditions that must be met before saving a record. If conditions are not met, an error message is displayed, preventing incorrect data from being saved.

### Student Object

- Validate\_email\_format – Ensures student email is entered in a proper format.
- Guardian\_Phone\_requirement – Requires guardian phone number to be filled in.
- No\_self\_Reference – Prevents a student record from referencing itself.
- Validate\_email\_format (Guardian) – Ensures guardian email is entered correctly.

The screenshot shows the Salesforce Setup interface for creating a Validation Rule. The left sidebar lists various object settings like Details, Fields & Relationships, Page Layouts, etc. The main area is titled "Student Validation Rule" and contains the following fields:

- Validation Rule Edit**: A sub-section with "Rule Name" set to "Validate\_Email\_Format", "Active" checked, and a "Description" field.
- Error Condition Formula**: A formula editor with the following code:

```
NOT(ISBLANK(Email__c)) && NOT(CONTAINS(Email__c, "@"))
```
- Functions**: A dropdown menu listing various functions such as ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc.

The screenshot shows the Salesforce Object Manager interface for the 'Student' object. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area displays a validation rule configuration. The formula editor shows the condition: `NOT ISBLANK(Email__c) && NOT CONTAINS(Email__c, "%")`. A tooltip for the `NOT ISBLANK` function is visible. Below the formula, an 'Error Message' section contains the example: 'Discount percent cannot exceed 30%' and the message: 'Please enter a valid email address'. The error location is set to 'Field Email'. At the bottom, there are Save, Save & New, and Cancel buttons.

*Process Flow: Setup → Object Manager → Student → Validation Rules  
→ New*

### Assignment Object

- Assignment\_Dur\_Date\_Not\_Before\_Created – Ensures due date is not earlier than the creation date.
- Assignment\_MaxMarks\_positive – Ensures maximum marks entered are positive.
- Assignment\_required\_course – Ensures each assignment is linked with a course.

The screenshot shows the 'Assignment Validation Rule' configuration screen. The sidebar lists setup options for the Assignment object. The main area shows a single validation rule named 'Assignment\_DueDate\_NotBeforeCreated'. The 'Rule Name' is 'Assignment\_DueDate\_NotBeforeCreated'. The 'Error Condition Formula' is: `AND  
NOT ISBLANK(Due_Date__c),  
Due_Date__c < DATEVALUE(CreatedDate)`. The 'Error Message' is 'Due Date cannot be earlier than the Assignment creation date'. The 'Error Location' is 'Due Date'. The rule is marked as 'Active'. At the bottom, there are Edit and Close buttons.

The screenshot shows the 'Assignment Validation Rule' creation page in the Salesforce Setup interface. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled 'Assignment Validation Rule' and contains a 'Validation Rule Detail' section. The rule is named 'Assignment\_MaxMarks\_Positive' and is active. The error condition formula is OR(ISBLANK(Max\_Marks\_\_c), Max\_Marks\_\_c <= 0). The error message is 'Max Marks must be greater than 0.' The error location is 'Max Marks'. The rule was created by Shruuti Misal on 9/17/2025 at 9:35 AM and modified by the same user on 9/17/2025 at 9:35 AM.

This screenshot shows another 'Assignment Validation Rule' creation page. The rule is named 'Assignment\_Requires\_Course' and is active. The error condition formula is ISBLANK(Course\_\_c). The error message is 'Assignment must be linked to a Course.' The error location is 'Course'. The rule was created by Shruuti Misal on 9/17/2025 at 9:36 AM and modified by the same user on 9/17/2025 at 9:36 AM. A note at the bottom states 'Assignment Validation Rule – Salesforce - Developer Edition.'

*Process Flow: Setup → Object Manager → Assignment → Validation Rules  
→ New*

---

## Exam Object

- Exam\_Duration\_Positive – Duration must be a positive number.
- Exam\_EndDate\_After\_StartDate – End date must come after start date.
- Exam\_Requires\_Course – Exam must be linked with a valid course.

SETUP > OBJECT MANAGER

## Exam

**Details**

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Restriction Rules

Scoping Rules

Object Access

### Exam Validation Rule

Back to Exam

**Validation Rule Detail**

Rule Name	Exam_Duration_Positive	Active	<input checked="" type="checkbox"/>
Error Condition Formula	OR( ISBLANK(Duration__c), Duration__c <= 0 )	Error Location	Duration
Error Message	Exam duration must be greater than 0.	Description	
Created By	Shruti Misal	Modified By	Shruti Misal
	<a href="#">Edit</a>	<a href="#">Clone</a>	

Help for this Page ⓘ

SETUP > OBJECT MANAGER

## Exam

**Details**

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Restriction Rules

Scoping Rules

Object Access

### Exam Validation Rule

Back to Exam

**Validation Rule Detail**

Rule Name	Exam_EndDate_After_StartDate	Active	<input checked="" type="checkbox"/>
Error Condition Formula	AND( NOT(ISBLANK(Exam_Start_Date__c)), NOT(ISBLANK(End_Date__c)), End_Date__c < Exam_Start_Date__c)	Error Location	Exam Date
Error Message	Exam End Date cannot be before Start Date.	Description	
Created By	Shruti Misal	Modified By	Shruti Misal
	<a href="#">Edit</a>	<a href="#">Clone</a>	

Help for this Page ⓘ

SETUP > OBJECT MANAGER

## Exam

**Details**

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Restriction Rules

Scoping Rules

Object Access

### Exam Validation Rule

Back to Exam

**Validation Rule Detail**

Rule Name	Exam_Requires_Course	Active	<input checked="" type="checkbox"/>
Error Condition Formula	ISBLANK(Course__c)	Error Location	Course
Error Message	Exam must be linked to a Course.	Description	
Created By	Shruti Misal	Modified By	Shruti Misal
	<a href="#">Edit</a>	<a href="#">Clone</a>	

Help for this Page ⓘ

*Process Flow: Setup → Object Manager → Exam → Validation Rules → New*

## Grade Object

- Grade\_Marks\_NonNegative – Prevents negative marks entry.
- Grade\_Marks\_NotExceed\_Max – Ensures entered marks do not exceed maximum marks.
- Grade\_Requires\_Submission – Ensures each grade is tied to a valid submission.

The screenshot shows the 'Grade Validation Rule' setup screen. On the left, a sidebar lists various object settings like Details, Fields & Relationships, Page Layouts, etc. The main area displays the validation rule details:

Validation Rule Detail	
Rule Name	Grade_Marks_NotExceed_Max
Error Condition Formula	AND( NOT(ISBLANK(Marks_Obtained__c)), NOT(ISBLANK(Submission__r_Assignment__rMax_Marks__c)), Marks_Obtained__c > Submission__r_Assignment__rMax_Marks__c)
Error Message	Marks Obtained cannot exceed Assignment Maximum Marks.
Description	
Created By	Shrujan Misal 9/17/2025, 10:05 AM
Modified By	Shrujan Misal 9/17/2025, 10:05 AM

*Process Flow: Setup → Object Manager → Grade → Validation Rules → New*

## Submission Object

- Submission\_Date\_NotBefore\_AssignmentCrea – Submission date cannot be before assignment creation date.

The screenshot shows the 'Submission Validation Rule' setup screen. On the left, a sidebar lists various object settings like Details, Fields & Relationships, Page Layouts, etc. The main area displays the validation rule details:

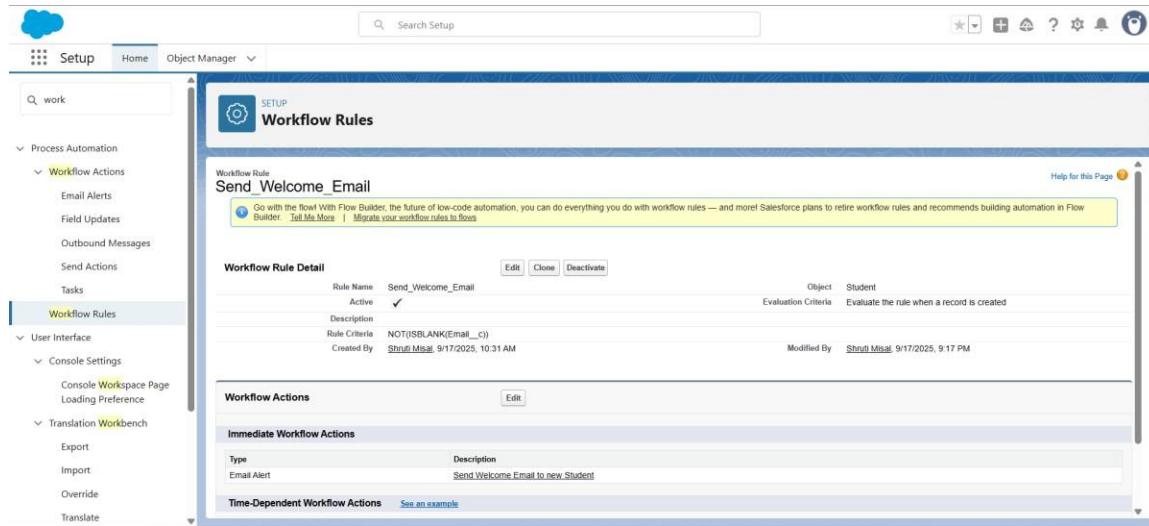
Validation Rule Detail	
Rule Name	Submission_Date_NotBefore_AssignmentCrea
Error Condition Formula	AND( NOT(ISBLANK(Submission_Date__c)), Submission_Date__c < Assignment__r_CreatedDate)
Error Message	Submission date cannot be before the assignment was created.
Description	
Created By	Shrujan Misal 9/17/2025, 9:47 AM
Modified By	Shrujan Misal 9/17/2025, 9:47 AM

*Process Flow: Setup → Object Manager → Submission → Validation Rules → New*

## Workflow Rule

Workflow rules automate standard internal procedures, such as sending alerts, updating fields, or creating tasks. They are triggered when records meet specific criteria.

- Send\_Welcome\_Email – Sends an automated welcome email to students after registration.



The screenshot shows the Salesforce Setup interface with the 'Workflow Rules' page open. The left sidebar shows various setup categories like Process Automation, Workflow Actions, and User Interface. The 'Workflow Rules' section is selected. A specific workflow rule, 'Send\_Welcome\_Email', is highlighted. The rule details show it's active, triggered by the condition 'NOT(ISBLANK(Email\_\_c))', and created by 'Shruti Misal' on 9/17/2025, 10:31 AM. It has one immediate workflow action: 'Email Alert' type, 'Send Welcome Email to new Student'. The 'Workflow Actions' section also includes a link to 'See an example'.

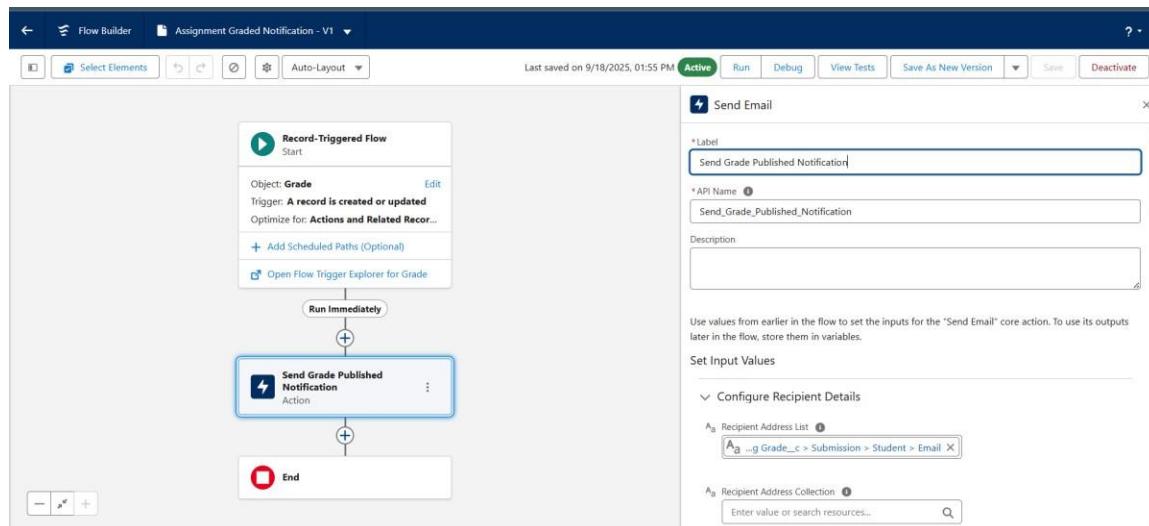
*Process Flow: Setup → Workflow Rules → New Rule*

## Flow Builder

Flow Builder allows the automation of complex processes. Record-triggered flows run automatically when a record is created or updated. These flows can send emails, update records, and perform actions without manual intervention.

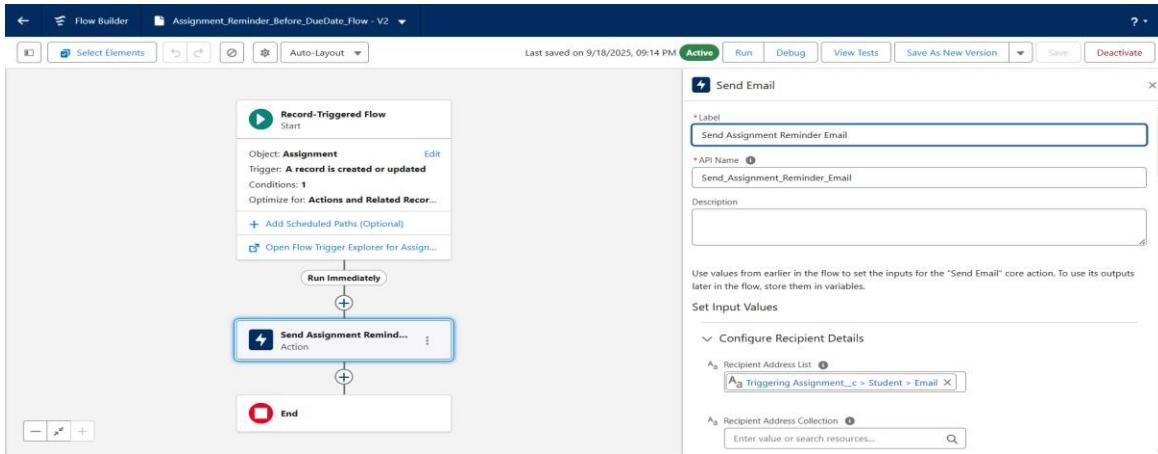
Flows implemented:

- Assignment Graded Notification – Sends notification when an assignment is graded.

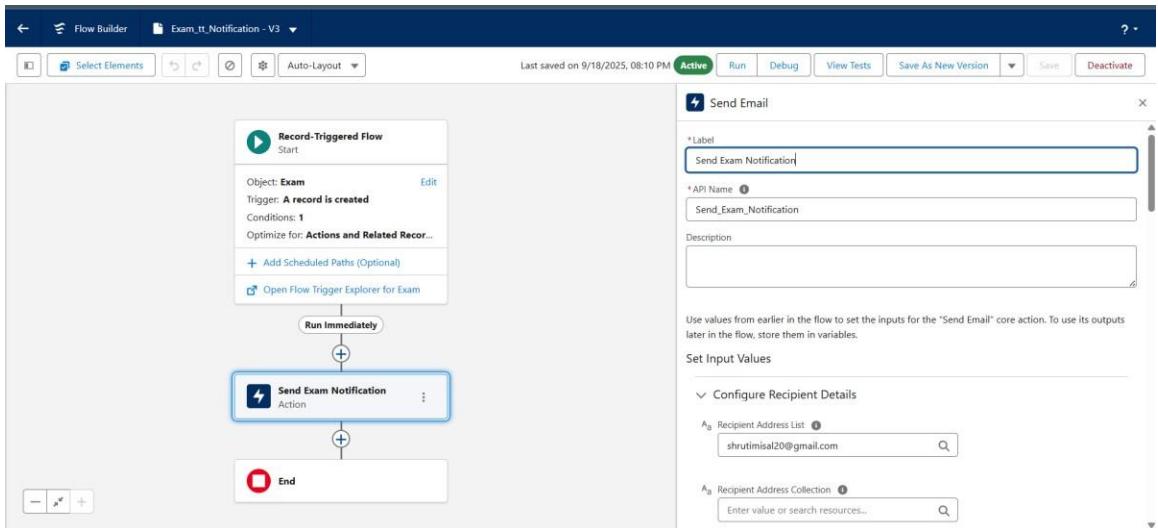


The screenshot shows the Flow Builder interface with a flow named 'Assignment Graded Notification - V1'. The flow starts with a 'Record-Triggered Flow' element for 'Grade'. The trigger is 'record is created or updated'. The flow runs immediately and contains a single action: 'Send Grade Published Notification'. This action uses the 'Send Email' core action with the API name 'Send\_Grade\_Published\_Notification'. The 'Recipient Address List' is set to 'A\_g ...g Grade\_\_c > Submission > Student > Email'. The flow ends with an 'End' element.

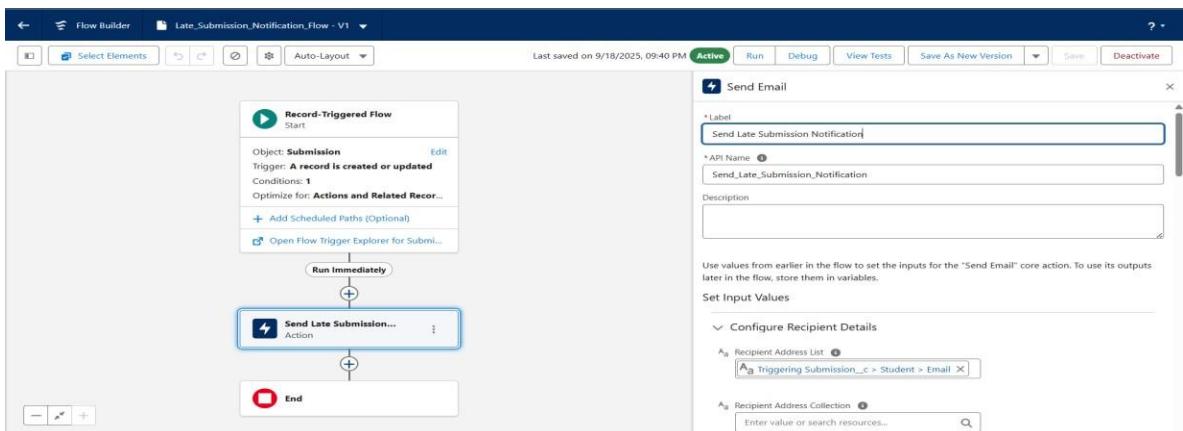
- Assignment\_Reminder\_Before\_DueDate\_Flow – Sends reminders before due dates.



- Exam\_tt\_Notification – Notifies students about exam timetables.



- Late\_Submission\_Notification\_Flow – Sends alerts when submissions are late.



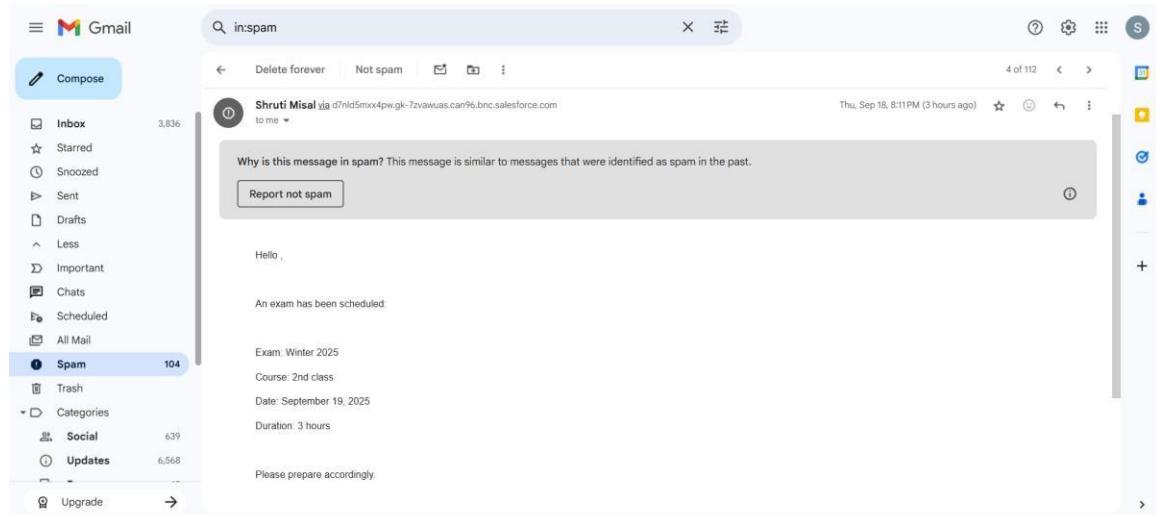
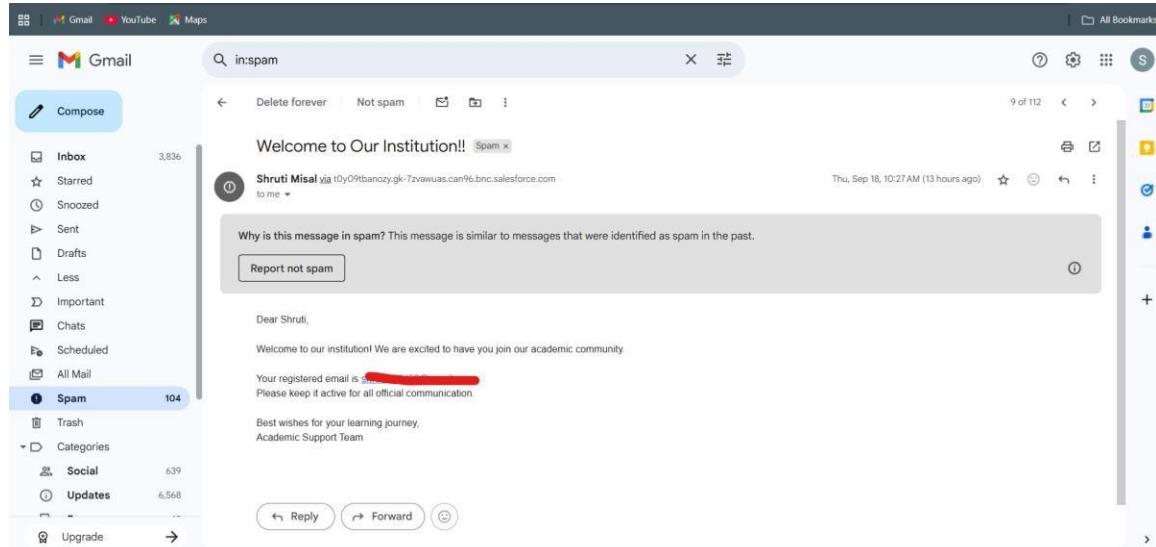
## *Process Flow: Setup → Flows → New Flow → Record-Triggered Flow*

---

### Email Alerts

Email alerts are automated messages sent to users when triggered by workflow rules or flows. They are essential for keeping students, guardians, and teachers informed about important events.

In this phase, email alerts are configured for events like assignment grading, exam scheduling, and submission deadlines. Screenshots of each email alert can be added in this section for reference.



## *Process Flow: Setup → Email Alerts → New Email Alert*

---

# Phase 5 Completion Report – Apex Programming (Developer)

## 1. Classes & Objects

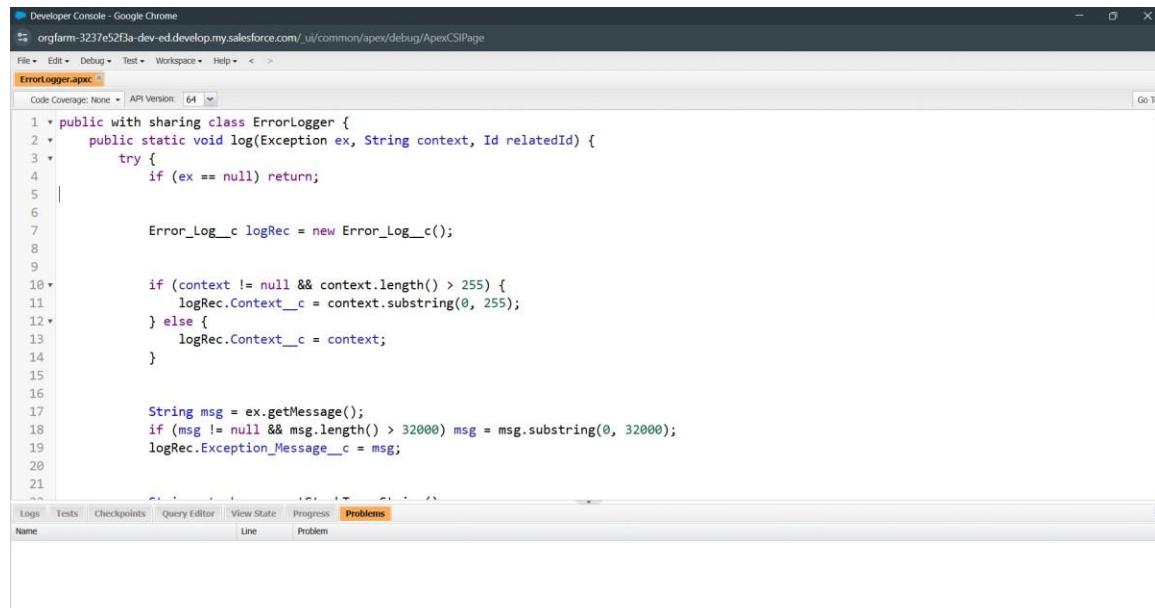
Custom Apex classes were created to handle specific business logic such as error logging, grade aggregation, performance recalculation, and guardian notifications.

Steps to Create Apex Class (applies to all classes):

1. Developer Console → File → New → Apex Class.
2. Enter Class Name → Click OK.
3. Write logic as per business requirement.
4. Save and test with Execute Anonymous or test classes.

Classes Implemented:

- **ErrorLogger** – Logs exceptions to Error\_Log\_c for debugging and auditing.

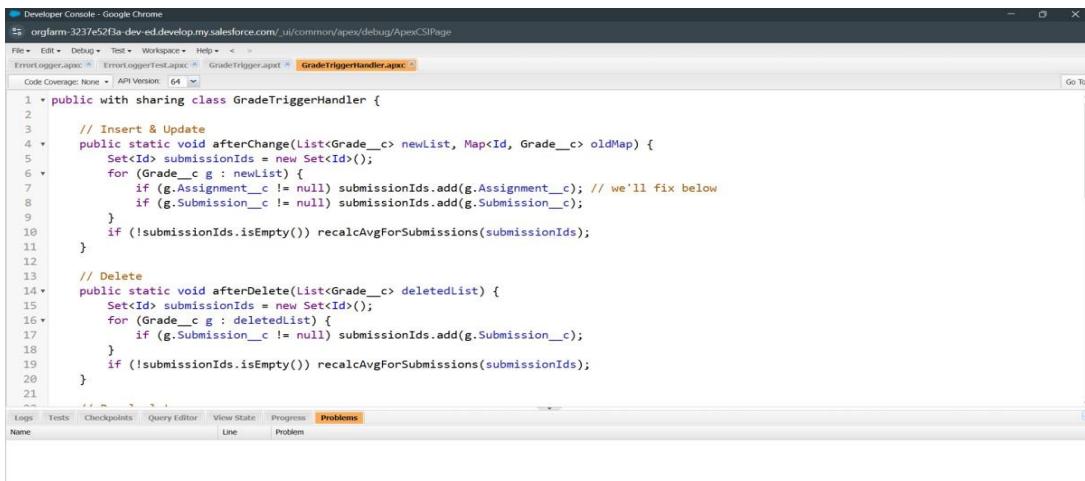


The screenshot shows the Salesforce Developer Console interface. The title bar reads "Developer Console - Google Chrome" and the URL is "orgfarm-3237e52Ba-dev-ed.develop.my.salesforce.com/ui/common/apex/debug/ApexCsPage". The menu bar includes File, Edit, Debug, Test, Workspace, Help, and a Go To button. The main area displays the code for the ErrorLogger class. The code is as follows:

```
1 public with sharing class ErrorLogger {
2     public static void log(Exception ex, String context, Id relatedId) {
3         try {
4             if (ex == null) return;
5         }
6
7         Error_Log__c logRec = new Error_Log__c();
8
9
10        if (context != null && context.length() > 255) {
11            logRec.Context__c = context.substring(0, 255);
12        } else {
13            logRec.Context__c = context;
14        }
15
16
17        String msg = ex.getMessage();
18        if (msg != null && msg.length() > 32000) msg = msg.substring(0, 32000);
19        logRec.Exception_Message__c = msg;
20
21    }
}
```

At the bottom of the code editor, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected, showing a table with columns for Name, Line, and Problem. The table is empty.

- **GradeTriggerHandler** – Recalculates student average marks after Grade changes.

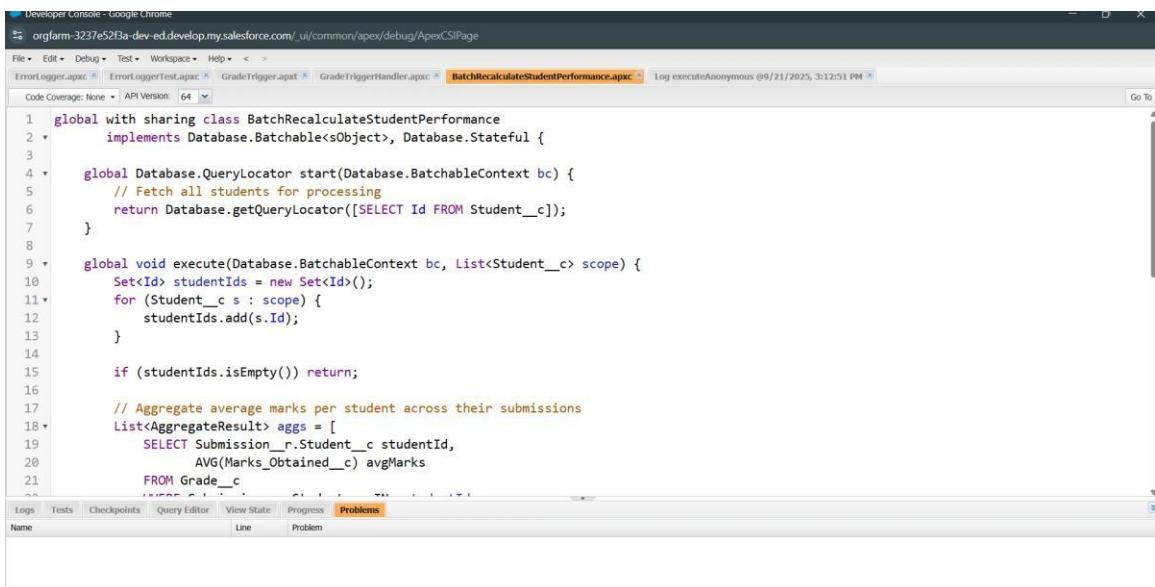


The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-3237e52Ba-dev-ed.develop.my.salesforce.com/ui/common/apex/debug/ApexCSIPage. The tab bar includes ErrorLogger.apxc, ErrorLoggerTest.apxc, GradeTrigger.apxc (selected), and GradeTriggerHandler.apxc. The API Version is set to 64. The code editor displays the GradeTriggerHandler.apxc class:

```
1 * public with sharing class GradeTriggerHandler {  
2  
3     // Insert & Update  
4     public static void afterChange(List<Grade__c> newList, Map<Id, Grade__c> oldMap) {  
5         Set<Id> submissionIds = new Set<Id>();  
6         for (Grade__c g : newList) {  
7             if (g.Assignment__c != null) submissionIds.add(g.Assignment__c); // we'll fix below  
8             if (g.Submission__c != null) submissionIds.add(g.Submission__c);  
9         }  
10        if (!submissionIds.isEmpty()) recalcAvgForSubmissions(submissionIds);  
11    }  
12  
13    // Delete  
14    public static void afterDelete(List<Grade__c> deletedList) {  
15        Set<Id> submissionIds = new Set<Id>();  
16        for (Grade__c g : deletedList) {  
17            if (g.Submission__c != null) submissionIds.add(g.Submission__c);  
18        }  
19        if (!submissionIds.isEmpty()) recalcAvgForSubmissions(submissionIds);  
20    }  
21}
```

The bottom navigation bar includes Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing no issues.

- **BatchRecalculateStudentPerformance** – Batch Apex to recalculate all students' average marks.



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-3237e52Ba-dev-ed.develop.my.salesforce.com/ui/common/apex/debug/ApexCSIPage. The tab bar includes ErrorLogger.apxc, ErrorLoggerTest.apxc, GradeTrigger.apxc, GradeTriggerHandler.apxc (selected), and BatchRecalculateStudentPerformance.apxc. The API Version is set to 64. The code editor displays the BatchRecalculateStudentPerformance.apxc class:

```
1 global with sharing class BatchRecalculateStudentPerformance  
2     implements Database.Batchable<sObject>, Database.Stateful {  
3  
4     global Database.QueryLocator start(Database.BatchableContext bc) {  
5         // Fetch all students for processing  
6         return Database.getQueryLocator([SELECT Id FROM Student__c]);  
7     }  
8  
9     global void execute(Database.BatchableContext bc, List<Student__c> scope) {  
10        Set<Id> studentIds = new Set<Id>();  
11        for (Student__c s : scope) {  
12            studentIds.add(s.Id);  
13        }  
14  
15        if (studentIds.isEmpty()) return;  
16  
17        // Aggregate average marks per student across their submissions  
18        List<AggregateResult> aggs = [  
19            SELECT Submission__r.Student__c studentId,  
20            AVG(Marks_Obtained__c) avgMarks  
21            FROM Grade__c
```

The bottom navigation bar includes Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing no issues.

- **GuardianNotificationQueueable** – Queueable Apex to send email notifications to guardians.

```

1 * public with sharing class GuardianNotificationQueueable implements Queueable {
2     private Set<Id> studentIds;
3
4     public GuardianNotificationQueueable(Set<Id> studentIds) {
5         this.studentIds = studentIds;
6     }
7
8     public void execute(QueueableContext ctx) {
9         // Query Students with their linked Guardian and Guardian Email
10        List<Student__c> studs = [
11            SELECT Id, Name, Guardian__c, Guardian__r.Email__c
12            FROM Student__c
13            WHERE Id IN :studentIds AND Guardian__c != null
14        ];
15
16        List<Messaging.SingleEmailMessage> mails = new List<Messaging.SingleEmailMessage>();
17
18        for (Student__c s : studs) {
19            if (s.Guardian__r != null && String.isNotBlank(s.Guardian__r.Email__c)) {
20                Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
21
22                mail.setTo(s.Guardian__r.Email__c);
23                mail.setSubject('Guardian Notification');
24
25                String message = 'Hello ' + s.Name + ',';
26                message += ' Your student has been added to the system.';
27                message += ' Please review their information and update if necessary.';
28
29                mail.setPlainTextBody(message);
30
31                mails.add(mail);
32            }
33        }
34
35        Messaging.sendEmail(mails);
36    }
37
38 }

```

- **WeeklyPerformanceScheduler** – Scheduled Apex to run the batch weekly.

```

1 * global with sharing class WeeklyPerformanceScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         Database.executeBatch(new BatchRecalculateStudentPerformance(), 200);
4     }
5 }

```

## 2. Apex Triggers

Triggers automate actions after DML events on key objects like Grade\_\_c.

Steps to Create Trigger:

1. Developer Console → File → New → Apex Trigger.
2. Enter Trigger Name → Select Object.
3. Choose events (before/after insert/update/delete).
4. Call handler class methods.
5. Save.

## Triggers Implemented:

- **GradeTrigger** – Fires after insert, after update, after delete on Grade\_c to recalc averages.
  - Calls GradeTriggerHandler.afterChange() and GradeTriggerHandler.afterDelete().

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, Debug, Test, Workspace, Help, and Go To. Below the menu is a toolbar with tabs for Error.logger.apex, Error.loggerTest.apex, GradeTrigger.apxt (which is selected and highlighted in orange), and GradeTriggerHandler.apxc. A dropdown menu shows Code Coverage: None and API Version: 64. The main workspace displays the following PL/SQL code:

```
1 trigger GradeTrigger on Grade__c (after insert, after update, after delete) {
2
3     if (Trigger.isAfter) {
4         if (Trigger.isInsert) GradeTriggerHandler.afterChange(Trigger.new, null);
5         if (Trigger.isUpdate) GradeTriggerHandler.afterChange(Trigger.new, Trigger.oldMap);
6         if (Trigger.isDelete) GradeTriggerHandler.afterDelete(Trigger.old);
7     }
8 }
```

At the bottom, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems, with Problems being the active tab. A status bar at the bottom shows Name, Line, and Problem.

## 3. Trigger Handler Classes

Handler classes separate logic from triggers (Trigger Design Pattern).

## GradeTriggerHandler:

- Recalculates average marks for students when grades are inserted, updated, or deleted.
  - Uses try-catch with ErrorLogger.log() to safely log exceptions.

Developer Console - Google Chrome

orgfarm-3237e52Ba-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

ErrorLogger.apxc ErrorLoggerTest.apxc GradeTriggerHandler.apxc

Code Coverage: None API Version: 64 Go To

```
1 * public with sharing class GradeTriggerHandler {  
2  
3     // Insert & Update  
4     public static void afterChange(List<Grade__c> newList, Map<Id, Grade__c> oldMap) {  
5         Set<Id> submissionIds = new Set<Id>();  
6         for (Grade__c g : newList) {  
7             if (g.Assignment__c != null) submissionIds.add(g.Assignment__c); // we'll fix below  
8             if (g.Submission__c != null) submissionIds.add(g.Submission__c);  
9         }  
10        if (!submissionIds.isEmpty()) recalcavgForSubmissions(submissionIds);  
11    }  
12  
13    // Delete  
14    public static void afterDelete(List<Grade__c> deletedList) {  
15        Set<Id> submissionIds = new Set<Id>();  
16        for (Grade__c g : deletedList) {  
17            if (g.Submission__c != null) submissionIds.add(g.Submission__c);  
18        }  
19        if (!submissionIds.isEmpty()) recalcavgForSubmissions(submissionIds);  
20    }  
21}
```

Steps:

1. Collect affected Submission\_c IDs from grades.
2. Map submissions to students.
3. Aggregate grades and calculate averages per student.
4. Update Student\_c records with Average\_Marks\_c.

## 4. Batch Apex

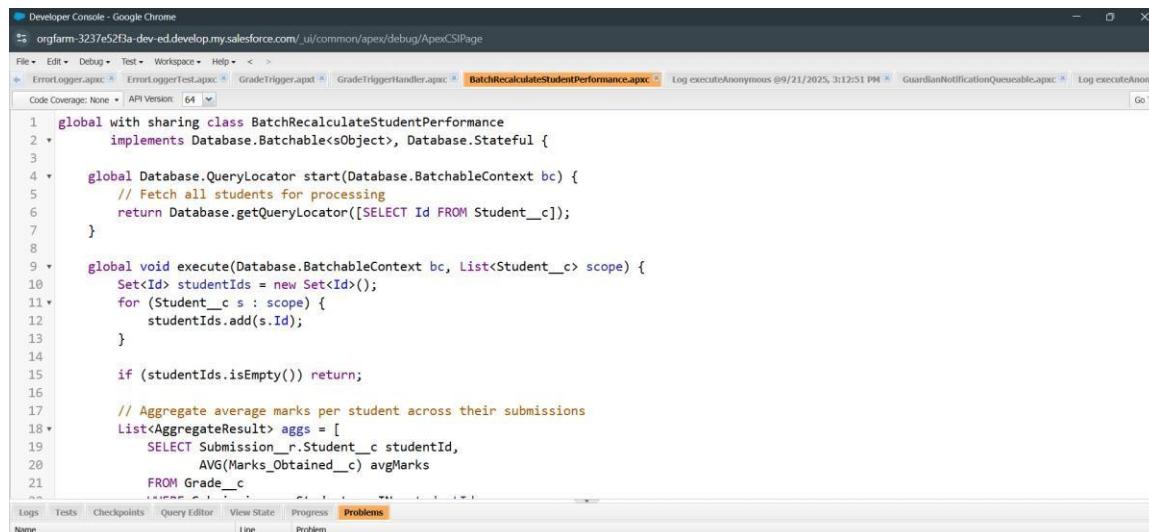
Batch Apex handles large datasets asynchronously.

BatchRecalculateStudentPerformance:

- Recalculates average marks for all students.
- Implements Database.Batchable<sObject> and Database.Stateful.

Steps:

- start() queries all Student\_c records.
- execute() calculates average marks for each student in batch scope.
- Updates Student\_c records.
- finish() logs completion (can enqueue further actions).



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-3237e523a-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage. The tab bar includes ErrorLogger.apc, ErrorLoggerTest.apc, GradeTrigger.apc, GradeTriggerHandler.apc, and BatchRecalculateStudentPerformance.apc. The status bar shows Log executeAnonymous @9/21/2025, 3:12:51 PM, GuardianNotificationQueueable.apc, and Log executeAnonymous. The code editor contains the following Apex code:

```
1 global with sharing class BatchRecalculateStudentPerformance {
2     implements Database.Batchable<sObject>, Database.Stateful {
3
4         global Database.QueryLocator start(Database.BatchableContext bc) {
5             // Fetch all students for processing
6             return Database.getQueryLocator([SELECT Id FROM Student_c]);
7         }
8
9         global void execute(Database.BatchableContext bc, List<Student_c> scope) {
10            Set<Id> studentIds = new Set<Id>();
11            for (Student_c s : scope) {
12                studentIds.add(s.Id);
13            }
14
15            if (studentIds.isEmpty()) return;
16
17            // Aggregate average marks per student across their submissions
18            List<AggregateResult> aggs = [
19                SELECT Submission__r.Student__c studentId,
20                    AVG(Marks_Obtained__c) avgMarks
21            FROM Grade_c
22        ];
23    }
24}
```

Process Flow Example:

Developer Console → File → New → Apex Class → Write Batch Logic → Save → Execute via Execute Anonymous or Scheduled Apex.

## 5. Queueable Apex

Queueable Apex processes asynchronous tasks in a chainable, flexible manner.

GuardianNotificationQueueable:

- Sends email notifications to guardians.

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-3237e52Ba-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage. The tab bar includes ErrorLogger.apcpx, ErrorLoggerTest.apcpx, GradeTrigger.apcpx, GradeTriggerHandler.apcpx, BatchRecalculateStudentPerformance.apcpx, Log executeAnonymous @9/21/2025, 3:12:51 PM, and GuardianNotificationQueueable.apcpx. The API Version is set to 64. The code editor displays the following Apex class:

```

1 * public with sharing class GuardianNotificationQueueable implements Queueable {
2     private Set<Id> studentIds;
3
4     public GuardianNotificationQueueable(Set<Id> studentIds) {
5         this.studentIds = studentIds;
6     }
7
8     public void execute(QueueableContext ctx) {
9         // Query Students with their linked Guardian and Guardian Email
10        List<Student__c> studs = [
11            SELECT Id, Name, Guardian__c, Guardian__r.Email__c
12            FROM Student__c
13            WHERE Id IN :studentIds AND Guardian__c != null
14        ];
15
16        List<Messaging.SingleEmailMessage> mails = new List<Messaging.SingleEmailMessage>();
17
18        for (Student__c s : studs) {
19            if (s.Guardian__r != null && String.isNotBlank(s.Guardian__r.Email__c)) {
20                Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
21                mail.setToAddresses(new String[] { s.Guardian__r.Email__c });
22            }
23        }
24    }
25
26    ...
27
28 }

```

The bottom navigation bar shows Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected.

Steps:

1. Accepts student IDs in constructor.
2. Queries students with linked guardians.
3. Builds email messages.
4. Sends via Messaging.sendEmail().

Process Flow Example:

Developer Console → File → New → Apex Class → Implement Queueable → Save → Enqueue using System.enqueueJob(new ClassName(params)).

## 6. Scheduled Apex

Scheduled Apex automates batch processing on a time-based schedule.

WeeklyPerformanceScheduler:

- Runs the BatchRecalculateStudentPerformance weekly.

Steps:

1. Developer Console → File → New → Apex Class.
2. Implement Schedulable interface.
3. In execute(), call Database.executeBatch(new BatchRecalculateStudentPerformance(), 200).
4. Save and schedule via Setup → Apex Classes → Schedule Apex.

```

1 * global with sharing class WeeklyPerformanceScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         Database.executeBatch(new BatchRecalculateStudentPerformance(), 200);
4     }
5 }

```

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help', and tabs for 'Performance.apxc', 'Log executeAnonymous @9/21/2025, 3:12:51 PM', 'GuardianNotificationQueueable.apxc', 'Log executeAnonymous @9/21/2025, 3:23:11 PM', 'WeeklyPerformanceScheduler.apxc' (which is the active tab), and 'Log executeAnonymous @9/21/2025, 3:28:35 PM'. The main area displays the Apex code for 'WeeklyPerformanceScheduler.apxc'. Below the code editor is a toolbar with tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Problems' tab is selected, indicated by an orange border.

## 7. Exception Handling

Implemented via ErrorLogger class:

- Wraps risky operations in try-catch blocks.
- Logs exceptions to Error\_Log\_c to prevent system failures and allow debugging.

## 8. Test Classes

- ErrorLoggerTest – Validates that exceptions are correctly logged.

```

1 *isTest
2 *private class ErrorLoggerTest {
3
4     @isTest
5     static void testLogCreatesRecord() {
6         Test.startTest();
7         try {
8             // Force an exception
9             Integer i = 1 / 0; // This will throw a System.MathException
10        } catch (Exception ex) {
11            // Call your logger with the caught exception
12            ErrorLogger.log(ex, 'UnitTestContext', null);
13        }
14        Test.stopTest();
15
16        // Query the inserted error log
17        List<Error_Log_c> logs = [
18            SELECT Id, Context__c, Exception_Message__c
19            FROM Error_Log__c
20            WHERE Context__c = 'UnitTestContext'
21            ORDER BY CreatedDate DESC
22        ];

```

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help', and tabs for 'ErrorLogger.apxc' and 'ErrorLoggerTest.apxc' (which is the active tab). The main area displays the Apex test code for 'ErrorLoggerTest.apxc'. Below the code editor is a toolbar with tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Tests' tab is selected, indicated by an orange border. At the bottom, there is a table for 'Overall Code Coverage' showing results for 'ErrorLogger'.

Status	Test Run	Enqueued Time	Duration	Failures	Total	Class	Percent	Lines
✓	TestRun @ 11:27:47 am			0	1	Overall	77%	
✓	TestRun @ 11:31:24 am			0	1	ErrorLogger	77%	14/18

- GradeTriggerHandlerTest – (if created) would verify average mark recalculations.

The screenshot shows a Salesforce Lightning interface for a student profile. The profile details are as follows:

- Student ID:** PRM123
- First Name:** Shruti
- Last Name:** Misal
- Email:** [Redacted]
- Phone:** [Redacted]
- Guardian Name:** raju misal
- Guardian Contact:** [Redacted]
- Average Marks:** 40.00

**Created By:** Shruti Misal, 9/17/2025, 9:52 PM    **Last Modified By:** Shruti Misal, 9/21/2025, 2:29 AM

**Activity Section:**

- Upcoming & Overdue:** No activities to show. Get started by sending an email, scheduling a task, and more.
- No past activity:** Past meetings and tasks marked as done show up here.

Steps to Create Test Classes:

1. Developer Console → File → New → Apex Class.
2. Annotate with `@isTest`.
3. Insert test data → Perform DML → Assert expected outcomes.

## 9. Asynchronous Processing

Batch, Queueable, and Scheduled Apex ensure heavy or repeated calculations (attendance, grades) run asynchronously.

- Reduces governor limits issues.
- Improves performance for large datasets.

# Phase 6: User Interface Development

## Phase 6: User Interface Development

### Lightning App Builder -> Record Pages

Setup → Lightning App Builder → New → Record Page → Object: Assignment\_c → Add Highlights Panel, Related Lists (Submissions, Grades).

The screenshot shows the Salesforce Setup interface with the Lightning App Builder selected in the sidebar. The main page displays the configuration for a Lightning Page named "Assignment\_Record\_Page". The "Information" section shows the Name as "Assignment\_Record\_Page" and the Label as "Assignment Record Page". Below this, the "Assignments By App" section indicates "No Assignments to display". The "Assignments By App, Record Type, and Profile" section lists assignments for the Assignment and Exam Management System object across various record types (Master), profiles (Guardian\_Profile, Standard User, Student\_Profile, System Administrator, Teacher\_Profile), and form factors (Desktop). A note at the bottom right says "Always show me ▾ more records per related list".

The screenshot shows the Salesforce Setup interface with the Lightning App Builder selected in the sidebar. The main page displays a list of Lightning Pages. The table has columns for Action, Label, Name, Namespace Prefix, Description, Type, Created By, and Last Modified By. The table shows five pages: Assignment\_Record\_Page, Exam\_Record\_Page, Grade\_Record\_Page, Home\_Page\_Default, and Submission\_Record\_Page. The "Assignment\_Record\_Page" and "Exam\_Record\_Page" were created by "str" on 9/22/2025 at 7:19 PM, while the others were created by "str" on 9/22/2025 at 7:46 PM. The "Home\_Page\_Default" was last modified by "str" on 8/14/2025 at 6:12 AM, and the others on 9/22/2025 at 7:37 PM.

Action	Label	Name	Namespace Prefix	Description	Type	Created By	Last Modified By
Edit   Clone   Del	Assignment_Record_Page	Assignment_Record_Page			Record Page	str 9/22/2025, 7:19 PM	str 9/22/2025, 7:20 PM
Edit   Clone   Del	Exam_Record_Page	Exam_Record_Page			Record Page	str 9/22/2025, 7:46 PM	str 9/22/2025, 7:46 PM
Edit   Clone   Del	Grade_Record_Page	Grade_Record_Page			Record Page	str 9/22/2025, 7:43 PM	str 9/22/2025, 7:43 PM
Edit   Clone   Del	Home_Page_Default	Home_Page_Default			Home Page	str 8/14/2025, 6:12 AM	str 8/14/2025, 6:12 AM
Edit   Clone   Del	Submission_Record_Page	Submission_Record_Page			Record Page	str 9/22/2025, 7:36 PM	str 9/22/2025, 7:37 PM

<https://orgfarm-3237e52f3a-dev-ed.lightning.force.com/lightning/setup/les/PageList/home>

Setup → Lightning App Builder → New → Record Page → Object: Exam\_c → Add Exam Details, Related Assignments.

Setup → Lightning App Builder → New → Record Page → Object: Grade\_c → Show Submission, Marks, Teacher Comments.

Setup → Lightning App Builder → New → Record Page → Object: Submission\_c → Show Submission details and related Grade.

Setup → Lightning App Builder → New → Home Page → Default → Add Recent Items, Reports, List View.

## Tabs (Custom Object Tabs)

The screenshot shows the Salesforce Setup interface with the 'Tabs' tab selected under 'User Interface'. The main content area displays a table titled 'Custom Object Tabs' with columns for Action, Label, Tab Style, and Description. The table lists various custom tabs, each with an 'Edit | Del' link. The 'Tab Style' column shows small icons representing different styles: Books, Bank, Safe, Bottle, Chest piece, Helicopter, Factory, and Compass. The 'Description' column provides a brief description of each tab's purpose.

Action	Label	Tab Style	Description
Edit   Del	Assignments	Books	
Edit   Del	Courses	Bank	
Edit   Del	Exams	Safe	
Edit   Del	Grades	Bottle	
Edit   Del	Guardians	Chest piece	
Edit   Del	Integration Logs	Helicopter	
Edit   Del	Students	Factory	
Edit   Del	Submissions	Compass	

The screenshot shows the Salesforce Setup interface with the 'Assignment' custom tab selected under 'User Interface'. The main content area displays the 'Custom Tab Definition Detail' for the 'Assignment' tab. The table includes fields for Tab Label (Assignments), Object (Assignment), Tab Style (Books), Splash Page Custom Link, Description, Created By (Shruti Misal), and Modified By (Shruti Misal). The 'Edit' and 'Delete' buttons are located at the top right of the table.

Tab Label	Assignments	Tab Style	Splash Page Custom Link		
Object	Assignment	Books			
Description					
Created By	Shruti Misal	9/16/2025, 10:54 AM	Modified By	Shruti Misal	9/16/2025, 10:54 AM

Setup → Tabs → New → Custom Object Tab → Exam

Setup → Tabs → New → Custom Object Tab → Courses

Setup → Tabs → New → Custom Object Tab → Grades

Setup → Tabs → New → Custom Object Tab → Guardians

Setup → Tabs → New → Custom Object Tab → Integration Logs

Setup → Tabs → New → Custom Object Tab → Students

## LWC (Lightning Web Components)

VS Code → Create LWC → assignmentSummary → Deploy to Salesforce → Add to Assignment Record Page.

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure under "ASSIGNMENT-AND-EXAM-MANAGEMENT-SYSTEM". It includes "force-app/main/default" with "classes", "contentassets", "flexipages", "flows", "layouts", and "lwc" folders. The "lwc" folder contains "assignmentSummary" with "assignmentSummary.html", "assignmentSummary.js", and "assignmentSummary.js-meta.xml". There are also "lastmileapp" and "tests" folders.
- Code Editor:** Displays two files:
  - assignmentSummary.html:** An LWC template file containing HTML and JS code. It includes a lightning card with a title "Assignments Summary" and an icon "standard:assignment". It iterates over "assignments" and displays each assignment's name, status, and due date.
  - assignmentSummary.js:** An LWC JavaScript file extending "LightningElement". It handles custom events, makes API calls via @wire, and maps submissions to assignments.
- Bottom Status Bar:** Shows the current file path: "force-app/main/default/lwc/assignmentSummary" and the file name "assignmentSummary.js".

EXPLORER ...

ASSIGNMENT-AND-EXAM-MANAGEMENT-SYSTEM

- force-app\main\default
  - classes
    - UpdateContactPhone.cls
    - UpdateContactPhone.cls-meta.xml
    - UpdatingContactException.cls
    - UpdatingContactException.cls-meta.xml
    - WeeklyPerformanceScheduler.cls
    - WeeklyPerformanceScheduler.cls-meta.xml
  - contentsets
  - flexpages
  - flows
  - layouts
  - lwc
    - assignmentSummary
      - \_tests\_
      - assignmentSummary.html
      - assignmentSummary.js
      - assignmentSummary.js-meta.xml
    - lastmileapp
      - \_tests\_
      - lastmileapp.test.js
      - lastmileapp.html
      - lastmileapp.js
      - lastmileapp.js-meta.xml
  - manifest
  - manifests.xml

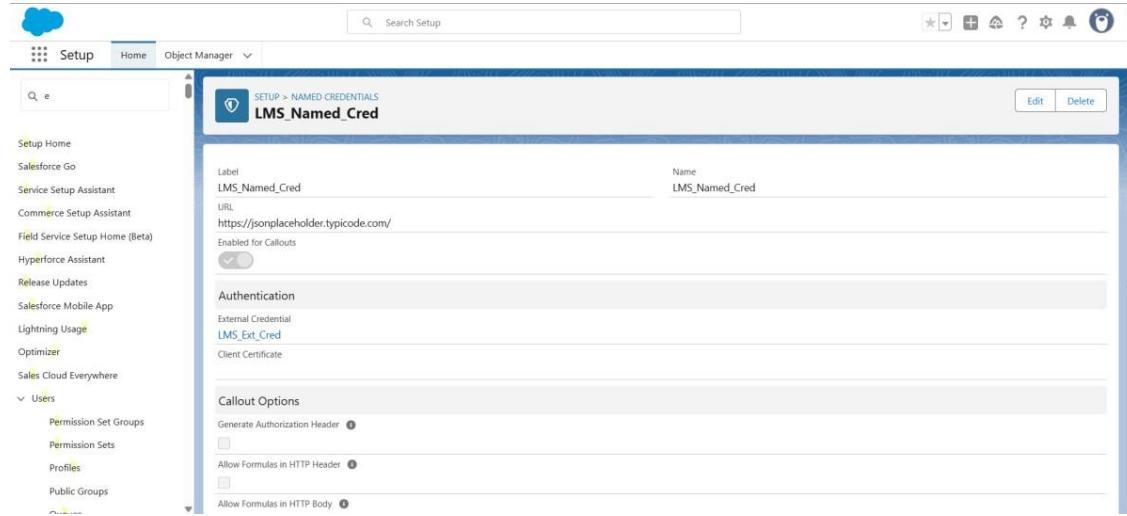
README.md assignmentSummary.html assignmentSummary.js package.xml assignmentSummary.js-meta.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
3   <apiVersion>58.0</apiVersion>
4   <isExposed>true</isExposed>
5   <targets>
6     <target>lightning__RecordPage</target>
7     <target>lightning__AppPage</target>
8     <target>lightning__HomePage</target>
9   </targets>
10  <targetConfigs>
11    <targetConfig targets="lightning__RecordPage">
12      <objects>
13        <object>Student__c</object>
14      </objects>
15    </targetConfig>
16  </targetConfigs>
17 </LightningComponentBundle>
```

# Phase 7: Integration & External Access

## Named Credentials

Setup → Named Credentials → New → Label: LMS\_Named\_Cred → URL:  
<https://jsonplaceholder.typicode.com/> → External Credential: LMS\_Ext\_Cred → Save.



## External Credentials

Setup → External Credentials → New → Label: LMS\_Ext\_Cred → Authentication Protocol: Basic Authentication → Save.

The screenshot shows the Salesforce Setup interface with the following details:

- Header:** Search Setup, Home, Object Manager.
- Page Title:** SETUP > NAMED CREDENTIALS - LMS\_Ext\_Cred
- Form Fields:**
  - Label:** LMS\_Ext\_Cred
  - Name:** LMS\_Ext\_Cred
  - Authentication Protocol:** Basic Authentication
- Related Named Credentials:** A table showing one entry:

Label	Name	URL
LMS_Named_Cred	LMS_Named_Cred	https://jsonplaceholder.typicode.com/
- Principals:** A table with columns: Se... (dropdown), Parameter Name, Identity Type, Authentication Status, Username, Actions.

# Phase 8: Data Management & Deployment

## Data Loader

Open Data Loader → Login with Salesforce → Select Insert → Object: Student\_c → Choose CSV file → Map fields → Finish.

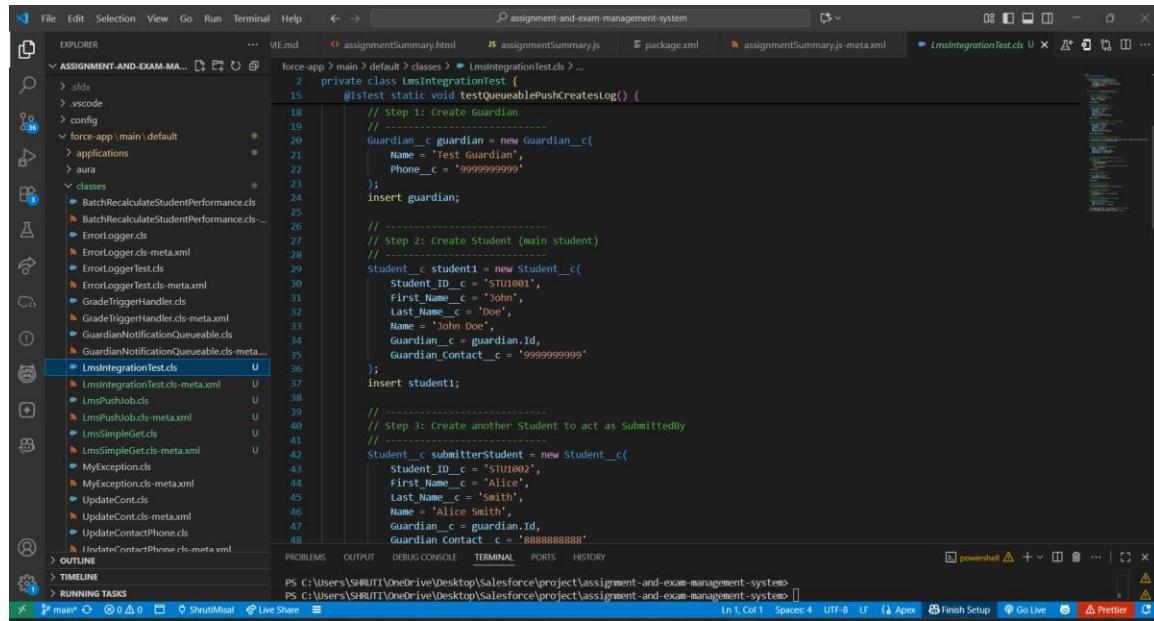
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Record ID	Average_N	Created By	Created Di	Deleted	Email	First Name	Guardian	Guardian C	Guardian I	Last Activit	Last Modif	Last Modif	Last Name	Last Refer	Last Viewe	Owner ID	Phone	Student ID	Student Ni	System Mc	Owner.ID	Owner.Act O
a02gk000	40_005gk000	09/18/202	FALSE	shrutimisa	Shruti	a08gk000	9.77e+09	raju misal	005gk000	09/21/202	005gk000	8.79e+09	PRM123	Shruti Misal	09/21/202	005gk000	TRUE	sh				
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3
4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6

## Data Export & Backup

Setup → Data Export → Schedule Export → Select Student\_c → Export to CSV → Download.

## VS Code & SFDX

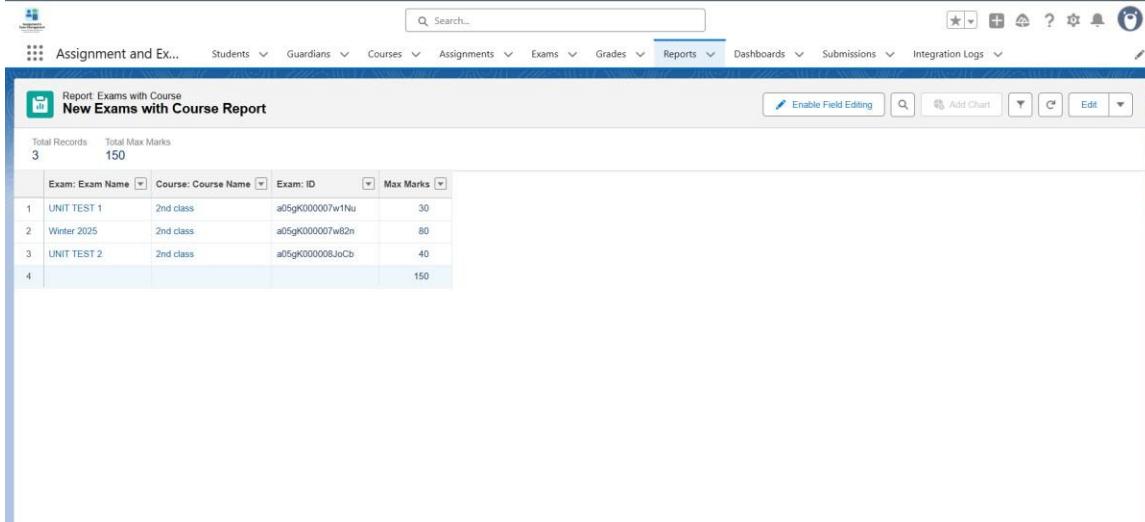
Install VS Code → Install Salesforce Extension Pack → Connect to Org (sfdx force:auth:web:login) → Push/Deploy metadata.



```
force-app > main > default > classes > LmsIntegrationTest.cls
  2 private class LmsIntegrationTest {
15     @isTest static void testQueueablePushCreatesLog() {
18         // Step 1: Create Guardian
19         // -----
20         Guardian__c guardian = new Guardian__c{
21             Name = 'Test Guardian',
22             Phone__c = '9999999999'
23         };
24         insert guardian;
25
26         // -----
27         // Step 2: Create Student (main student)
28         // -----
29         Student__c student1 = new Student__c(
30             Student_ID__c = 'STU1001',
31             First_Name__c = 'John',
32             Last_Name__c = 'Doe',
33             Name = 'John Doe',
34             Guardian__c = guardian.Id,
35             Guardian_Contact__c = '9999999999'
36         );
37         insert student1;
38
39         // -----
40         // Step 3: Create another Student to act as SubmittedBy
41         // -----
42         Student__c submitterStudent = new Student__c(
43             Student_ID__c = 'STU1002',
44             First_Name__c = 'Alice',
45             Last_Name__c = 'Smith',
46             Name = 'Alice Smith',
47             Guardian__c = guardian.Id,
48             Guardian_Contact__c = '8888888888'
```

# Phase 9: Reporting, Dashboards & Security Review

## Reports



The screenshot shows a reporting interface with a header bar containing various navigation links such as Assignment and Ex., Students, Guardians, Courses, Assignments, Exams, Grades, Reports, Dashboards, Submissions, and Integration Logs. Below the header is a search bar and a toolbar with icons for star, plus, home, question mark, gear, and notifications. The main content area is titled "Report: Exams with Course" and "New Exams with Course Report". It displays a table with the following data:

	Exam: Exam Name	Course: Course Name	Exam: ID	Max Marks
1	UNIT TEST 1	2nd class	a05gK000007w1Nu	30
2	Winter 2025	2nd class	a05gK000007w62n	80
3	UNIT TEST 2	2nd class	a05gK000008JoCb	40
4				150

Reports → New Report → Object: Exam\_c → Report: New Exams with Course Report

Reports → New Report → Object: Course\_c → Report: New Courses with Student Report

Screenshot: \_\_\_\_\_

Reports → New Report → Object: Grade\_c → Report: New Grades with Assignment Report

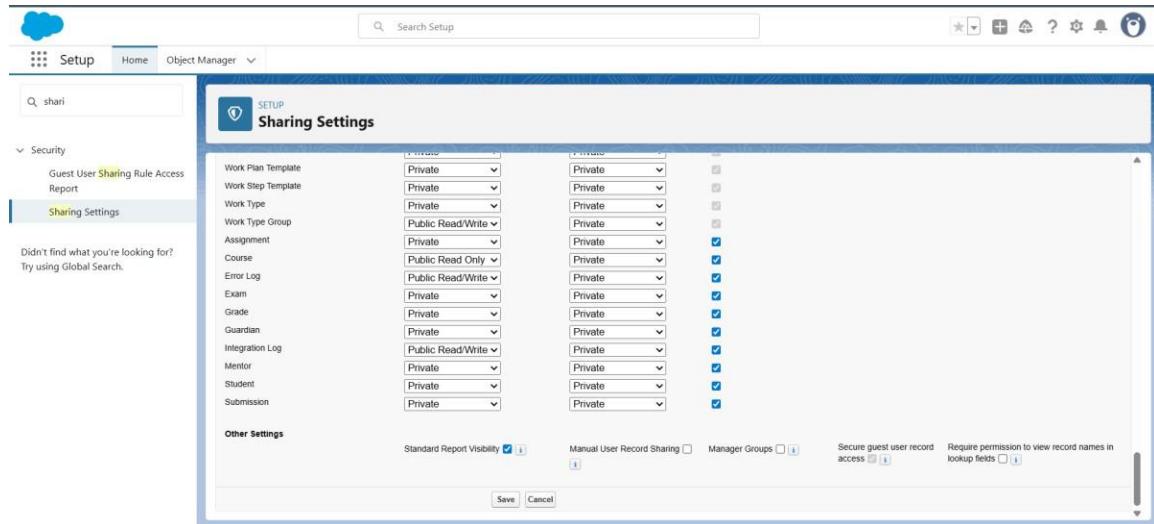
Screenshot: \_\_\_\_\_

Reports → New Report → Object: Submission\_c → Report: New Submissions Report

Screenshot: \_\_\_\_\_

## Sharing Settings (OWD)

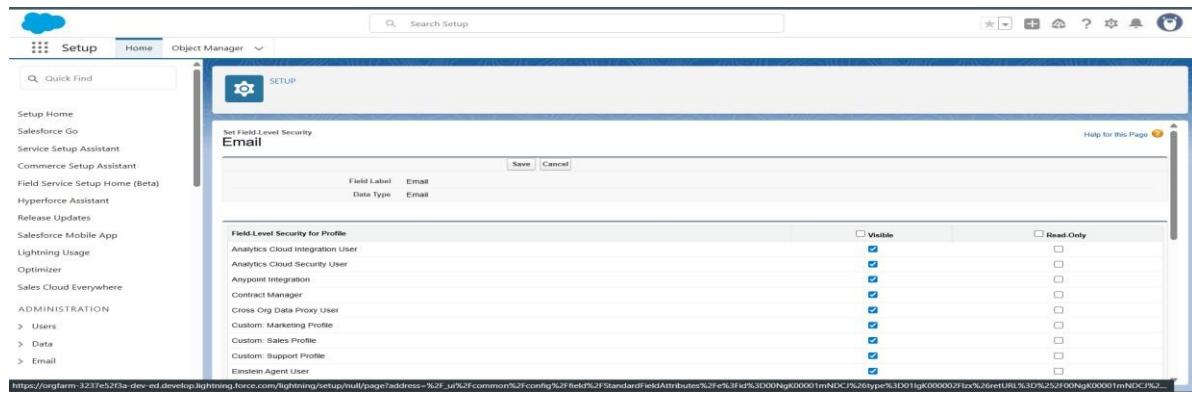
Setup → Sharing Settings → Edit OWD → Define object-level access for Assignment, Submission, Grade, etc.



The screenshot shows the 'Sharing Settings' page in the Salesforce Setup. The left sidebar has 'Sharing Settings' selected under 'Security'. The main area displays a grid of sharing rules for various objects. The columns are labeled 'Object' and 'Profile'. Most objects have their sharing set to 'Private', indicated by a dropdown menu icon. Some objects like 'Assignment' and 'Grade' have 'Public Read/Write' or 'Public Read Only' options. The 'Profile' column shows checkboxes for 'Visible' and 'Read Only'. At the bottom of the grid, there are checkboxes for 'Standard Report Visibility', 'Manual User Record Sharing', 'Manager Groups', 'Secure guest user record access', and 'Require permission to view record names in lookup fields'. There are 'Save' and 'Cancel' buttons at the bottom right.

## Field Level Security

Setup → Object Manager → [Select Object] → Fields & Relationships → Select Field → Set Field-Level Security.



The screenshot shows the 'Set Field Level Security' page for the 'Email' field. The left sidebar lists various setup categories. The main area has 'Field Label' set to 'Email' and 'Data Type' set to 'Email'. Below this is a table titled 'Field Level Security for Profile'. It lists several profiles on the left and checkboxes for 'Visible' and 'Read Only' on the right. Most checkboxes for 'Visible' are checked, while most for 'Read Only' are unchecked. At the bottom right of the table, there are 'Save' and 'Cancel' buttons.

## Session Settings

Setup → Session Settings → Session Timeout = 30 minutes → Save.

The screenshot shows the Salesforce Setup interface with the following details:

- Header:** Cloud icon, Setup, Home, Object Manager, Search Setup.
- Left Sidebar:** Security, Session Settings (selected).
- Page Title:** Session Settings.
- Section:** Session Settings.
- Sub-section:** Session Timeout.
- Form Fields:**
  - Timeout Value: 30 minutes (selected).
  - Disable session timeout warning popup.
  - Force logout on session timeout.
- Section:** Session Settings.
- List:** A series of checkboxes for session security settings:
  - Lock sessions to the IP address from which they originated.
  - Lock sessions to the domain in which they were first used.
  - Terminate all of a user's sessions when an admin resets that user's password.
  - Force relogin after Login-As-User.
  - Use POST requests for cross-domain sessions.
  - Enforce login IP ranges on every request.
  - When embedding a Lightning application in a third-party site, use a session token instead of a session cookie.
- Section:** Extended use of IE11 with Lightning Experience.
- Note:** EXTENDED USE OF IE11 WITH LIGHTNING EXPERIENCE HAS NOW ENDED. AS OF DECEMBER 31, THE EXTENDED PERIOD HAS ENDED, AND USE OF INTERNET EXPLORER 11 (IE 11) WITH LIGHTNING EXPERIENCE IS NO LONGER SUPPORTED. ISSUES WITH PERFORMANCE OR FUNCTIONALITY.

## Login IP Ranges

Setup → Profiles → [Select Profile] → Login IP Ranges → No restrictions added.

## Audit Trail

Setup → View Setup Audit Trail → Download CSV if required.

The screenshot shows the Salesforce Setup interface with the following details:

- Header:** Cloud icon, Setup, Home, Object Manager, Search Setup.
- Left Sidebar:** Security, View Setup Audit Trail (selected).
- Page Title:** View Setup Audit Trail.
- Section:** View Setup Audit Trail.
- Note:** The last 20 entries for your organization are listed below. You can download your organization's setup audit trail for the last six months (Excel, csv file).
- Table:** A list of audit trail entries with columns: Date, User, Source Namespace Prefix, Action, Section, and Delegate User.

Date	User	Source Namespace Prefix	Action	Section	Delegate User
9/26/2025, 1:23:56 AM PDT	shrunmisad0524@agentrforce.com		Changed Session Timeout Value from 120 to 30 minutes	Session Settings	
9/26/2025, 1:23:56 AM PDT	shrunmisad0524@agentrforce.com		Session Security Level for Multi-Factor Authentication was set to High Assurance	Session Settings	
9/26/2025, 1:23:56 AM PDT	shrunmisad0524@agentrforce.com		Session Security Level for Passwordless Login was set to Standard	Session Settings	
9/26/2025, 1:23:56 AM PDT	shrunmisad0524@agentrforce.com		Session Security Level for Lightning Login was set to Standard	Session Settings	
9/26/2025, 1:23:56 AM PDT	shrunmisad0524@agentrforce.com		Session Security Level for Activation was set to Standard	Session Settings	
9/26/2025, 1:23:56 AM PDT	shrunmisad0524@agentrforce.com		Session Security Level for Delegated Authentication was set to Standard	Session Settings	
9/26/2025, 1:23:56 AM PDT	shrunmisad0524@agentrforce.com		Session Security Level for Username Password was set to Standard	Session Settings	
9/26/2025, 10:41:43 PM PDT	shrunmisad0524@agentrforce.com		Changed LmsIntegrationTest Apex Class code	Apex Class	
9/25/2025, 10:40:16 PM PDT	shrunmisad0524@agentrforce.com		Changed LmsIntegrationTest Apex Class code	Apex Class	
9/25/2025, 10:38:50 PM PDT	shrunmisad0524@agentrforce.com		The custom field option Required for Submitted By on object Submissions has changed from on to off	Custom Objects	
9/25/2025, 10:36:33 PM PDT	shrunmisad0524@agentrforce.com		Changed LmsIntegrationTest Apex Class code	Apex Class	
9/25/2025, 10:33:12 PM PDT	shrunmisad0524@agentrforce.com		Changed LmsIntegrationTest Apex Class code	Apex Class	
9/25/2025, 10:31:34 PM PDT	shrunmisad0524@agentrforce.com		Changed LmsIntegrationTest Apex Class code	Apex Class	
9/25/2025, 10:29:57 PM PDT	shrunmisad0524@agentrforce.com		Changed LmsIntegrationTest Apex Class code	Apex Class	
9/25/2025, 10:28:10 PM PDT	shrunmisad0524@agentrforce.com		Changed LmsIntegrationTest Apex Class code	Apex Class	
9/25/2025, 10:26:24 PM PDT	shrunmisad0524@agentrforce.com		Changed LmsIntegrationTest Apex Class code	Apex Class	

# Phase 10: Quality Assurance Testing

## Object- Record-Creation Testing(Student\_c):

The screenshot displays two screenshots of the Salesforce Lightning interface, illustrating the process of creating and viewing a student record.

**Screenshot 1: New Student Record Creation**

This screenshot shows the "New Student" form. The "Information" section contains the following data:

- \* Student Name: Atharva Misal
- \* Student ID: PRM234
- \* First Name: Atharva
- \* Last Name: Misal
- Email: shrumisal0205@gmail.com
- Phone: 9999999999

The "Owner" field is set to Shruti Misal. At the bottom, there are "Cancel", "Save & New", and "Save" buttons.

**Screenshot 2: Student Record View**

This screenshot shows the details of the newly created student record, "Atharva Misal".

**Details Tab:**

- Name: Atharva Misal
- Owner: Shruti Misal
- Phone: 9999999999
- Email: shrumisal0205@gmail.com
- Contact: 3-8888

**Activity Tab:**

- Header: Activity
- Filters: All time • All activities • All types
- Buttons: Refresh, Expand All, View All
- Section: Upcoming & Overdue
- Text: No activities to show. Get started by sending an email, scheduling a task, and more.
- Text: No past activity. Past meetings and tasks marked as done show up here.

## Workflow rule Testing(Send\_Welcome\_email):

The screenshot shows an email in the spam folder. The subject is "Welcome to Our Institution!!". The recipient is "Shruti Misal shrutimisal20@gmail.com via mzs9lgj14g6dlw.gk-7zvawus.can96.bnc.salesforce.com to me". The email body contains a welcome message, a registered email address, and a note about keeping it active for official communication. It ends with best wishes from the Academic Support Team.

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Report not spam

Dear Atharva,

Welcome to our institution! We are excited to have you join our academic community.

Your registered email is [shrumisal0205@gmail.com](mailto:shrumisal0205@gmail.com). Please keep it active for all official communication.

Best wishes for your learning journey,  
Academic Support Team

Reply Forward

## Trigger Testing:

The screenshot shows a grade entry screen in a web-based application. The student's name is "Atharva Misal". The grade details include a submission number "SUB-0001", marks obtained "40", graded date "9/26/2025", graded by "Anita Deshpande", assignment "Science 2", and student "Atharva Misal". The "Save" button is visible at the bottom right.

Information

Grade Number

Submission: SUB-0001

\*Marks Obtained: 40

Graded Date: 9/26/2025

Graded By: Anita Deshpande

Assignment: Science 2

Student: Atharva Misal

Cancel Save & New Save

A screenshot of a software application interface, likely a school management system. The top navigation bar includes links for Assignment and Ex., Students, Guardians, Courses, Assignments, Exams, Grades, Reports, Dashboards, Submissions, and Integration Logs. A search bar and various icons are also present. A modal window titled "Grade" shows the creation of a new grade entry with the number "GRD-0002". The form fields include Grade Number (GRD-0002), Owner (Shruti Misal), Submission (SUB-0001), Marks Obtained (40), Grade Letter (A), Graded Date (9/26/2025), Graded By (Anita Deshpande), Assignment (Science 2), and Student (Atharva Misal). A success message at the top right says "Grade \*GRD-0002\* was created." Buttons for New Contact, Edit, and New Opportunity are visible at the bottom right of the modal.

### Average\_Marks

40.00



Created By

Last M