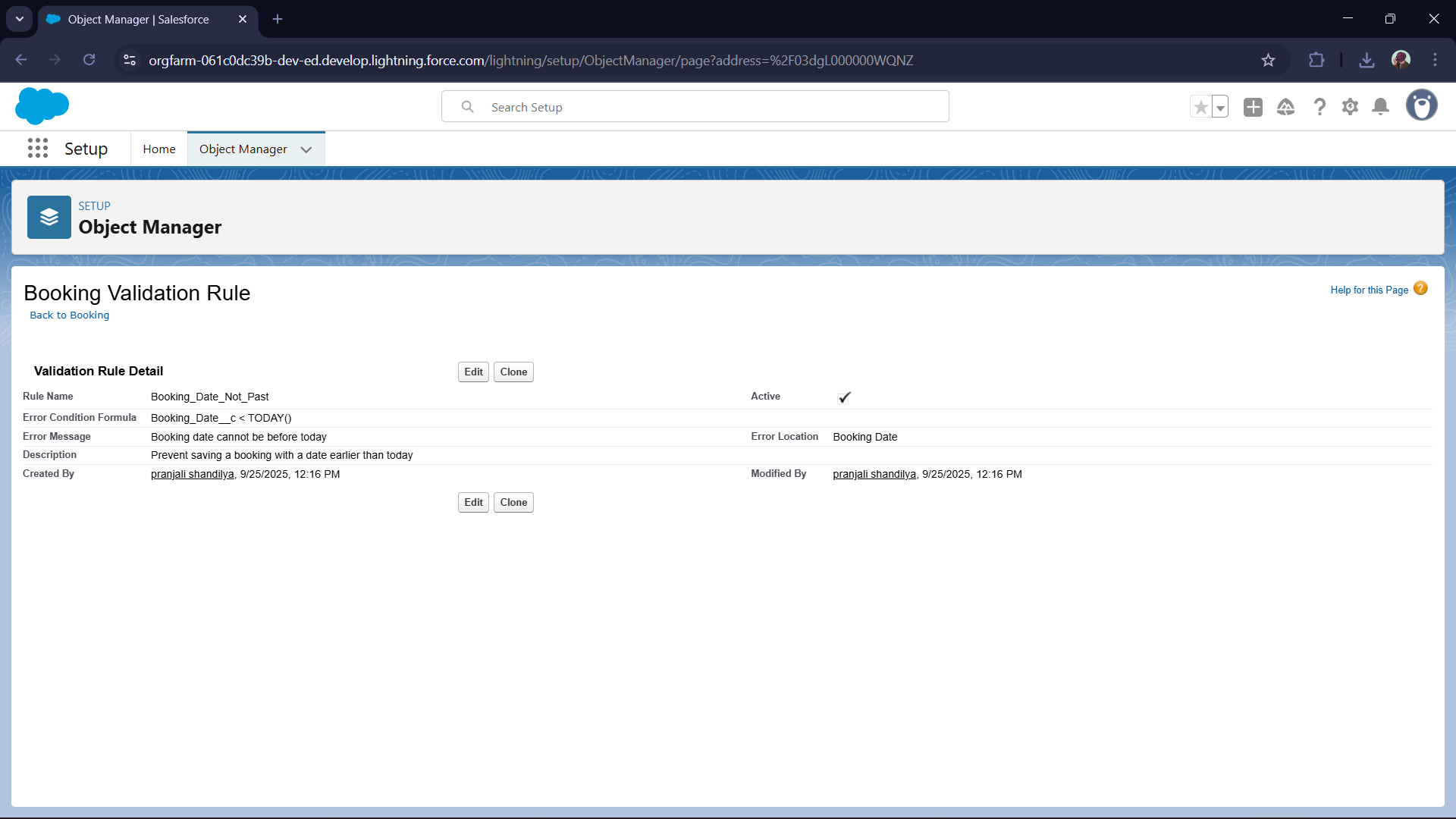
**Phase 4: Process Automation (Admin) — Tutor Match CRM**

In this phase, we automate the essential operational workflows of Tutor Match CRM using Salesforce declarative tools. The objective: reduce manual work for parents, tutors, and admin staff; ensure notifications, follow-ups, and data consistency.  
  
**1) Validation Rules**

**Use:** Enforce data integrity before saving records.  
**Implementation in Tutor Match:**

  
  
**2) Workflow Rules (legacy)**

**Use:** Automate simple actions like email alerts or field updates.  
**Implementation in Tutor Match:**

* **Workflow on Booking**: When booking is created with status = “Confirmed”
  + **Immediate Action:** Send an Email Alert to tutor & parent
  + **Result:** Stakeholders are instantly notified of confirmed bookings
  + (Note: for new projects, replace with Flow whenever possible because Salesforce is pushing automation to Flow.)

A screenshot of a computer

AI-generated content may be incorrect.

**3) Process Builder**

**Use:** Intermediate automation with branching, invocation, etc.  
**Implementation in Tutor Match:**

* On Booking creation, if booking’s **Source = “Website”**, then set a custom field Booking\_Channel\_\_c = "Website"

**4) Approval Process**

**Use:** Standardize approval steps for critical requests or conditions.  
**Implementation in Tutor Match:**

* Example: **Discount Approval for Package**
  + A tutor or parent applies a discount request on a package purchase
  + **Step 1:** Counselor submits discount on Booking
  + **Step 2:** Manager reviews the request
  + **If approved:** update Discount\_\_c, status becomes Approved
  + **If rejected:** status = Rejected
  + This ensures unauthorized discounts or changes are controlled

A screenshot of a computer

AI-generated content may be incorrect.  
  
**Email Alerts**

**Use:** Automatically notify users of important events.  
**Implementation in Tutor Match:**

* Create email templates (Lightning / Classic) like Booking\_Confirmation\_Email
* Use via Workflow or Flow: e.g. after Booking is confirmed, send email to tutor & parent
* Templates include merge fields (Booking Name, Date, Tutor name, etc.)

A screenshot of a computer screen

AI-generated content may be incorrect.  
  
  
  
A screenshot of a computer

AI-generated content may be incorrect.  
  
  
**7) Field Updates**

**Use:** Automatically change field values based on conditions.  
**Implementation in Tutor Match:**

* Example: When a Booking is approved via Approval Process, set Booking\_Status\_\_c = "Confirmed"
* In a Flow: if package has a discount, automatically update Discounted\_Price\_\_c = Package\_Price\_\_c \* (1 – Discount\_Percent\_\_c)

**8) Tasks**

**Use:** Automatically assign follow-up or action items for users.  
**Implementation in Tutor Match:**

* Flow triggers creating a **Follow-up Task** when a Booking is created
  + **Subject:** Follow up: {Booking Name}
  + **Due Date:** Booking Date or next day
  + **Owner:** Tutor or Parent (or internal coordinator)
* Ensures no appointment or booking is forgotten

**9) Custom Notifications**

**Use:** Send in-app and mobile real-time alerts (instead of or alongside emails).  
**Implementation in Tutor Match:**

* Configure a **Custom Notification Type**: e.g. New Booking Alert
* In Flow: after a new Booking is created, use **Send Custom Notification** action to alert the assigned tutor / parent
  + **Title:** “New Booking Received”
  + **Body:** “You have a new booking: {Booking Name}”
  + **Recipient:** Tutor user ID / Parent user Id

Result: real-time, in-app/mobile push notifications, improving responsiveness  
  
  
A screenshot of a computer

AI-generated content may be incorrect.

**Overall Result (for Tutor Match)**

* Automations enforce data correctness (via Validation Rules)
* Flow automations reduce manual steps (auto-create sessions, cancel sessions, reminders)
* Approval process ensures discounts or special cases don’t bypass oversight
* Notifications and tasks guide users so no booking, session, or payment slips through
* The system becomes proactive rather than reactive