

# Xeno SDE Internship Assignment – 2025

Hi there! 🙌

Thanks for applying to the **SDE Internship position at Xeno**. We're excited to get to know you through this hands-on assignment that reflects the kind of real-world challenges we solve every day.


## **Assignment Goal:**

Build a **Mini CRM Platform** that enables customer segmentation, personalized campaign delivery, and intelligent insights using modern tools and approaches.

---

## **Scope of Work**

### **1. Data Ingestion APIs**

- Create secure, well-documented **REST APIs** to ingest **customers** and **orders** data.
  - Use **Postman** or **Swagger UI** to demonstrate API usage.
  -  **Brownie Points:** Implement a **pub-sub architecture** using a message broker (Kafka, RabbitMQ, Redis Streams, etc.) where:
    - API layer handles only validation
    - Actual data persistence happens asynchronously via a consumer service
- 

### **2. Campaign Creation UI**

Build a simple web application that lets a user:

- Define **audience segments** using flexible rule logic (e.g., spend > INR 10,000 AND visits < 3 OR inactive for 90 days)
- Combine conditions using AND/OR with a dynamic rule builder

- Preview **audience size** before saving the segment
- After saving, redirect to a **campaign history** page showing:
  - List of past campaigns
  - Delivery stats (sent, failed, audience size)
  - Most recent campaign at the top

✓ Bonus: Implement a clean, intuitive UX (drag-and-drop, visual rule blocks, or other creative approaches)

---

### ✓ 3. Campaign Delivery & Logging

- On saving a segment, initiate a new campaign:
  - Store campaign details in a `communication_log` table
  - Send a **personalized message** to each customer via a dummy vendor API
    - Example message: *"Hi Mohit, here's 10% off on your next order!"*
  - The Vendor API should simulate real-world delivery success/failure (~90% SENT, ~10% FAILED)
  - Vendor API must hit a **Delivery Receipt API** on your backend
- The Delivery Receipt API should:
  - Update the delivery status in the communication log
  - ✓ Brownie Points: Use a **consumer-driven process** that updates DB in batches, even if API hits are individual

---

### ✓ 4. Authentication

- Implement **Google OAuth 2.0-based authentication**
  - Ensure that only logged-in users can create audiences or view campaigns
- 

## ✓ 5. AI Integration

Incorporate **at least one AI-powered feature** into your CRM app. This is an opportunity to showcase your creativity and practical use of modern tools. Some ideas:

### 🧠 Suggested AI Use Cases

#### 1. Natural Language to Segment Rules

Allow users to write prompts like *"People who haven't shopped in 6 months and spent over ₹5K"* and convert them into logical rules.

#### 2. AI-Driven Message Suggestions

Given a campaign objective (e.g., "bring back inactive users"), generate 2–3 message variants to choose from.

✓ Bonus: Recommend relevant **product or offer images** based on message tone or audience.

#### 3. Campaign Performance Summarization

Instead of just showing sent/failed stats, generate a human-readable insight summary:

"Your campaign reached 1,284 users. 1,140 messages were delivered. Customers with > ₹10K spend had a 95% delivery rate."

#### 4. Smart Scheduling Suggestions

Recommend the best time/day to send a campaign based on customer activity patterns (you can mock or simulate this logic).

#### 5. Audience Lookalike Generator

Suggest additional audiences based on the characteristics of high-performing segments (e.g., people similar to those who responded last time).

#### 6. Auto-tagging Campaigns

Use AI to label campaigns (e.g., "Win-back", "High Value Customers") based on audience and message intent.

Use any public AI APIs (OpenAI, Google Vertex AI, Replicate, etc.) or local models. Just clearly document what you used and why.

---

## Preferred Tech Stack

- **Frontend:** React.js or Next.js
  - **Backend:** Node.js or Java (Spring Boot)
  - **Database:** MySQL or MongoDB
  - **Optional:** Kafka / RabbitMQ / Redis Streams for pub-sub
  - **Optional:** AI tools, LLM APIs, image APIs for personalization
- 

## Evaluation Criteria





We'll be assessing:

- Code quality and project structure
- Creative problem solving and clean UX
- Scalability (especially in ingestion & delivery flows)
- AI integration and thoughtful use of tools
- Completeness of your solution (demo, README, hosting)
- Communication skills in your **demo video**

We'll go through your project in detail during interviews, so **plagiarism is a strict no-go**. All work must be your own, and we'll ask you questions based on your implementation decisions.

---

## Submission Requirements

-  Public GitHub repo with well-organized code
-  Deployed project (can use Render, Vercel, Railway, etc.)
-  **Demo video** (max 7 mins) in your own voice explaining:
  - Features
  - How you approached the problem
  - Any trade-offs made
  - AI features added
-  **README .md** with:
  - Local setup instructions
  - Architecture diagram
  - Summary of AI tools and other tech used
  - Known limitations or assumptions

 Submit here by **EOD 12th May 2025**:


 [Google Form Link](#)

---

 **Let's Go!**

Curious what it's like working with us? [Check out what our past interns have to say](#) 🙄

This is your chance to show not just what you know — but how you solve, create, and ship.

We're excited to see what you build! 

Check us out at: <https://www.getxeno.com>