

# RazinSoft Limited

## Laravel Developer Technical Test

**Task:** Smart Auto-Reply Workflow Management System for Chat Bot

### Objective:

Build an advanced Laravel-based chat management system that automatically handles guest messages with context-relevant auto-replies, dynamic workflow control, and agent assignment automation.

When no human agent is assigned to a guest chat:

- The system must analyze the guest's message and send a context-aware auto-reply (not a static message).

When a human agent is assigned:

- Auto-reply is automatically disabled for that chat.

When the agent is unassigned or goes inactive:

- Auto-reply is reactivated automatically.

All workflows and bot behavior should be configurable from an Admin Panel.

### Core Requirements

#### 1. Entities

- Guest - Unauthenticated user who can start a chat.
- Agent - Authenticated user who handles assigned chats.
- Admin - Manages the entire system and workflow.
- Chat - Represents a guest conversation session.
- Message - Stores individual messages (sender, type, content).

## **2. Workflow Logic**

### **A. Auto-Reply System**

- When a guest sends a message:
- If no agent assigned -> trigger smart auto-reply.
- If agent assigned -> store message only (no auto-reply).

### **B. Smart Auto-Reply**

- Implement NLP-like logic (keyword-based or using an AI API).
- Example: 'price', 'support', 'refund' -> relevant replies.
- Auto-reply messages must be saved with `is_auto_reply = true`.

### **C. Workflow Events**

- `MessageReceived` -> triggers `HandleAutoReply` listener.
- `AgentAssigned` -> disables auto-reply.
- `AgentUnassigned` -> enables auto-reply.

### **D. Workflow State Control -**

Each chat has:

- `auto_reply_enabled`
- `agent assigned`
- `last_activity_at`
- Admin can manually override chat states.

## **Admin Panel Features**

- Dashboard with all guest chats and statuses.
- Assign/unassign agents to chats.
- Enable/disable auto-reply globally or per chat.
- Edit auto-reply rules: keyword -> custom reply.
- View chat transcripts.

## Technical Expectations

- Laravel 11+
- Events, Listeners, Queues, and Eloquent used properly.
- Policies for roles (Admin, Agent, Guest).
- Scheduled command for reactivating idle chats.
- Tests for workflow logic.
- README with setup and explanation.

## Bonus Challenges

1. AI Reply Integration (OpenAI, Hugging Face, etc.)
2. Real-time updates via Web Sockets/Laravel Echo.
3. Redis caching for chat states.
4. Workflow change logs.
5. Rate limiting for guests.

## Evaluation Criteria

- Architecture & Code Quality (25%)
- Workflow Logic (25%)
- Auto-Reply Intelligence (20%)
- Admin UI & UX (15%)
- Performance & Sync Handling (10%)
- Testing & Documentation (5%)

## Example Workflow

Guest sends message -> MessageReceived event -> Check if agent assigned ->

If yes -> store message only ->

If no -> Dispatch HandleAutoReply job -> Generate relevant reply -> Save and return to guest.