

# Zaya - LeyLine Bot PRD

## Product Requirement Document (PRD):

Leyline Website Chatbot MVP – Intent Detection + Feishu Data Sync

Owner: Product Intern

Reviewer: Product Lead / CTO

Version: v1.0

Date: June 2025

---

### Objective

Build an embedded website chatbot (Zaya) that can:

- Identify the user's intent (Creator, Developer, or Bug Reporter)
  - Respond with relevant information or links
  - Capture and log user inputs into a Feishu (Lark) multi-dimensional table for internal tracking
- 




### User Flow

1. User opens chatbot on any Leyline webpage

2. Bot welcomes user:

“Hi, I'm Zaya – your creative guide. What brings you here today?”

3. Bot presents three quick reply buttons:

-  I'm a **Content Creator**
-  I'm a **Developer**
-  I want to **Report a Bug**

4. Bot provides next-step guidance:

**If Content Creator:**

“Awesome! You can try StormEye for AI video generation or Thunderr for growing your content. What would you like help with?”

### If Developer:

~~“We love builders. Our API and CLI tools are coming soon — want early access or documentation?”~~

### If Bug Reporter:

“Thanks for reporting! Please briefly describe what went wrong, and we’ ll take care of the rest.”

#### 5. Bot captures:

- User selection
- User's first typed message
- Timestamp
- Page URL (optional / Or simply record the link of the currently opened **Zaya** page.)



#### 6. Data is sent to **Feishu Sheet**

## ✓ Feature Scope (MVP)

Feature	Description
Quick-reply buttons	For identifying user intent
Contextual response logic	Based on selected role
Data collection	ID, intent, message, timestamp
Feishu integration	Logged into a structured table
Low-code implementation	Using Crisp / Tidio + Zapier/Make

## Tech Stack & Tools

- **Development Workflow:**
  - Both frontend and backend should be **developed and tested locally first.**

- Once initial testing is successful, the code should be **integrated into the official project repositories**.
- **Frontend:**
  - Built with **React**
  - Final integration into: [mktp-web-demo](#)
- **Backend:**
  - Preferred languages: **Java** or **Python**
  - Responsible for handling data processing, API logic, and integration with automation tools
- **Use of AI Tools or Third-Party APIs:** If the project requires AI capabilities or external services, please contact  **Anran** for centralized resource coordination and access approval
- **Data Storage:** All user selections and input messages should be automatically synced to a **Feishu Multi-dimensional Table** for internal tracking and analysis
  - 📌 **Data Destination:** Feishu Multi-dimensional Table
  -  [Feishu Developer Docs](#)

Feishu Table Structure	
Field	Source
User ID	Generated by chat tool or session ID
User Intent	Button selected (Creator / <del>Developer</del> / Bug)
User Message	Typed messages
Timestamp	Time of entry
Page URL	Auto-captured from browser context (if possible)

## Milestones & Timeline

Please review the project scope and **propose a development plan with estimated timeline** based on your own assessment.

## Acceptance Criteria

- Bot launches automatically on webpage
- User can choose their role from buttons

- Bot gives meaningful next-step responses
  - User message + selection is logged to Feishu
  - Sheet is updated in real time via automation tool
- 

## Notes for Intern

- Keep messages short, friendly, and on-brand (Zaya = creative guide + helpful assistant)
- Use emoji or visual tags in buttons for better UX
- Validate Feishu tokens & test submission via Postman before automating