# **Zivana – AI Assistant**

## **1. Introduction**

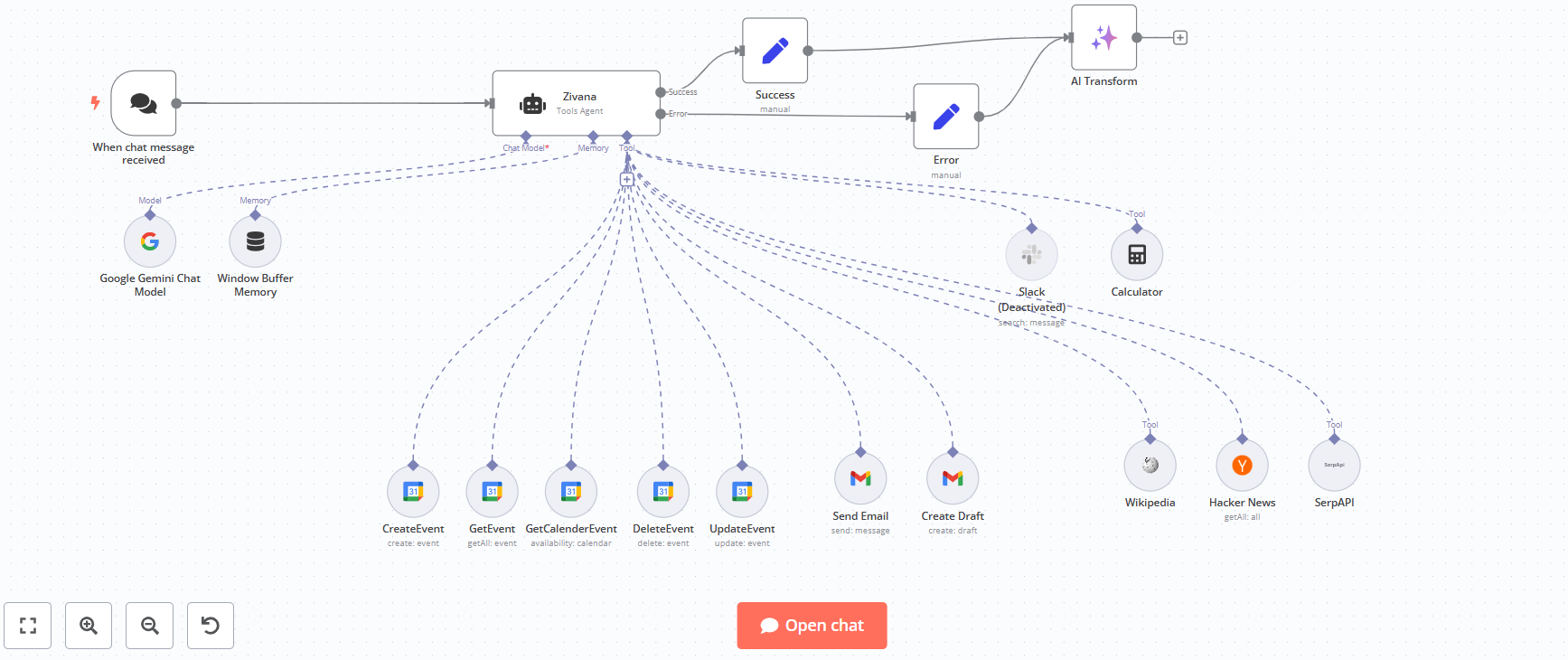
This document outlines the **architecture design flow** of Zivana, detailing its system components, interactions, and data flow. The architecture is designed to provide seamless automation for email management, scheduling, web searches, and voice-based interactions.

## **2. High-Level Architecture**

Zivana follows a **modular, event-driven architecture**, integrating AI, automation, and communication tools.

### **2.1 Core Architectural Components**

1. **User Interface Layer**
   1. Telegram Bot
   2. Slack Integration
   3. Web-based Commands
2. **Processing & AI Layer**
   1. Google Gemini AI for NLP and Conversational Intelligence
   2. n8n Workflow Engine for Task Automation
3. **Integration & API Layer**
   1. Google Calendar API (Event Scheduling)
   2. Gmail API (Email Management)
   3. SerpAPI, Wikipedia API (Web Search)
   4. Airtable API (Contacts Management)
4. **Data Storage Layer**
   1. Airtable (Contacts Database)
   2. Google Calendar (Event Data)
   3. Local Memory Buffer (Short-Term Context)



## **3. Workflow and Data Flow**

### **3.1 User Request Handling Flow**

1. **User Interaction:**
   1. Users send a request via Telegram, Slack, or Web.
2. **Processing & Task Determination:**
   1. Google Gemini AI processes natural language input.
   2. n8n automation decides the appropriate action (Email, Calendar, Search, etc.).
3. **Data Fetching & Execution:**
   1. Relevant APIs are called to fetch required information.
   2. Actions such as scheduling or email handling are performed.
4. **Response Generation & Delivery:**
   1. AI formats the response based on execution output.
   2. Response is sent back to the user through the requested interface.

### **3.2 Event Scheduling Flow**

1. **User requests a meeting setup.**
2. **Zivana checks for available time slots using Google Calendar API.**
3. **If a slot is available, it creates the event; if not, it suggests alternatives.**
4. **User confirms, and event details are sent as a response.**

### **3.3 Email Management Flow**

1. **User requests email actions (read, send, delete).**
2. **Zivana calls the Gmail API to fetch or modify email data.**
3. **Summarized results or email drafts are generated and sent back.**

**4. Future Enhancements**

* Implementing **AI-driven workflow optimization** for better automation.
* Enhancing **multi-platform support** beyond Slack and Telegram.
* **Real-time event conflict detection** for improved scheduling.
* **Predictive AI suggestions** for smarter decision-making.

## **5. Conclusion**

Zivana’s architecture is designed for **scalability, efficiency, and AI-driven automation**, making task management seamless and intelligent. Future iterations will focus on **deep learning-based personalization and expanded integration.**

## **6. Team – Gully Coders**

1. Mrunal Ghodeswar
2. Shailendra Murmu
3. Siddhartrh Moradiya
4. Aas Vora Jain
5. Jatin Suthar