1. **Bot-Buddy - PRD**

Jira Epic	Assessments Feature for Bot-Buddy
Delivery Date	
Document Status	In Progress
Product Lead	Shreeja
Engineering Lead	
Developer	Manoj Acharya; Rashika Karki
Designer Lead	Deepak Joshi
Quality Assurance (QA) Lead	
Relevant Documents	All the document links: User Research: User Research Final User Stories (for wireframe): Final version of user stories 5.29 Roadmap: Project Charter: Working Agreement: Working Agreement- C0-Pilot.docx Stakeholder Requirements: Sketches : Figma Link UI/UX Requirements: JIRA: Jira board

Document owner:| Last Updated: [06/10/24]

Table of Contents

Version History

Any time someone comes and updates this document, you will write your name. Usually, you would need a sign off. In this case, we're going to give you an automatic Approval.

Version	Author	Approved By	Approval Date	Description of Change
1.1				
1.2				
1.3				
1.4				

COMPETITORS:

Drift

Answerly

GHL

Respond io

Hellorep ai

Competitor Analysis

Main application:

Answerly: Focused on AI-powered knowledge management and support.

Drift: Focused on conversational marketing and sales automation.

Chatfuel: Focused on social media chatbot creation and automation.

Target Audience:

Answerly: Businesses seeking enhanced internal knowledge management and customer support.

Drift: Marketing and sales teams aiming to increase lead conversion and engagement.

Chatfuel: SMBs, e-commerce, and marketers looking to automate interactions on social media.

Core Features:

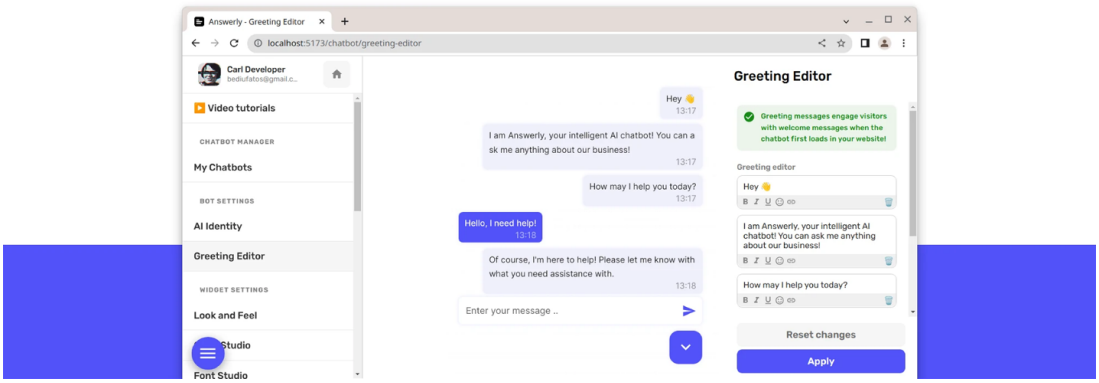
Answerly: AI-powered search, knowledge management, and seamless integrations.

Drift: Live chat, AI chatbots, lead qualification, and conversational landing pages.

Chatfuel: No-code bot building, multichannel support, automation, and analytics.

Answerly.io

Answerly is an AI agent creation platform that provides instant customer support 24/7 to your website visitors. It's customizable, easy to integrate, and works by understanding your business.



Introduction:

Answerly.io is a conversational AI platform designed to provide intelligent and personalized responses to user queries across various channels. Here's an overview of its key aspects:

Purpose: Answerly.io is primarily used for customer support automation, lead generation, and engagement enhancement. It serves as a virtual assistant capable of understanding natural language queries and delivering relevant answers in real-time.

History and Origin: Answerly.io was founded by a team of AI enthusiasts and developers with a vision to revolutionize customer support and engagement through AI-driven solutions. Its development likely stemmed from the growing demand for automation in customer service and the advancements in NLP technology. The company may have undergone iterations and enhancements based on user feedback and market trends to evolve into its current form. Further details on its history and origin would require more in-depth research or direct communication with the company.

Basic Features:

1. Natural Language Processing (NLP): Answerly.io leverages advanced NLP algorithms to comprehend user input and extract intent, enabling accurate responses.
2. Multi-channel Support: It integrates seamlessly with websites, messaging apps, social media platforms, and other communication channels, ensuring consistent and accessible support.
3. Personalization: The platform tailors responses based on user preferences, past interactions, and contextual information, enhancing user satisfaction.
4. Analytics and Insights: Answerly.io provides analytics dashboards to track performance metrics, user interactions, and sentiment analysis, facilitating data-driven decision-making.

Key Differentiators:

1. Advanced NLP Capabilities: Answerly.io boasts robust NLP capabilities, enabling it to understand complex queries and provide precise responses, setting it apart from competitors with basic rule-based systems.
2. Omnichannel Support: Its ability to seamlessly integrate across multiple channels offers a cohesive user experience, distinguishing it from competitors limited to specific platforms.
3. Personalization Engine: Answerly.io's sophisticated personalization engine enhances user engagement by delivering tailored responses, fostering stronger customer relationships.

Customer Base: Answerly.io caters to businesses across various industries, including e-commerce, finance, healthcare, and technology. Its customer base comprises both small businesses and enterprise clients seeking efficient and scalable customer support solutions.

Market Share: While specific market share data may not be readily available, Answerly.io has gained traction in the conversational AI market, competing with established players like IBM Watson Assistant, Google Dialogflow, and Microsoft Azure Bot Service.

KPIs:

1. Response Accuracy: Measure the platform's accuracy in understanding and addressing user queries.

2. User Satisfaction: Monitor customer satisfaction scores and feedback to gauge the effectiveness of Answerly.io's responses.
3. Time to Resolution: Evaluate the platform's efficiency in resolving user inquiries, aiming for minimal response times.
4. Conversion Rates: Track the impact of Answerly.io on lead generation, sales, and other conversion metrics to assess its ROI.
5. Retention Rate: Analyze the retention of users engaging with Answerly.io over time, indicating its long-term viability and effectiveness.

Additional Features (we can have):

As for **additional features to compete with Answerly.io**, consider the following:

1. Multilingual Support: Integrate language translation capabilities to serve a diverse user base, especially if your target market spans across different regions and languages.

2. Voice Recognition: Incorporate voice-enabled interactions for hands-free user engagement, allowing users to interact with the chatbot through speech commands.

3. Integration with CRM Systems: Enable seamless integration with popular Customer Relationship Management (CRM) systems like Salesforce, HubSpot, or Zoho CRM, facilitating efficient data management and personalized interactions based on customer profiles and history.

4. E-commerce Integration: Develop features that support e-commerce functionalities such as product recommendations, order tracking, and transaction processing within the chatbot interface, enhancing the user experience for online shoppers.

5. AI-Powered Insights and Recommendations: Implement AI-driven analytics capabilities to provide actionable insights, predictive analytics, and proactive recommendations for businesses based on user interactions and trends.

6. Emotional Intelligence: Enhance the chatbot's emotional intelligence by incorporating sentiment analysis and empathy algorithms to better understand and respond to users' emotions and moods, leading to more empathetic and human-like interactions.

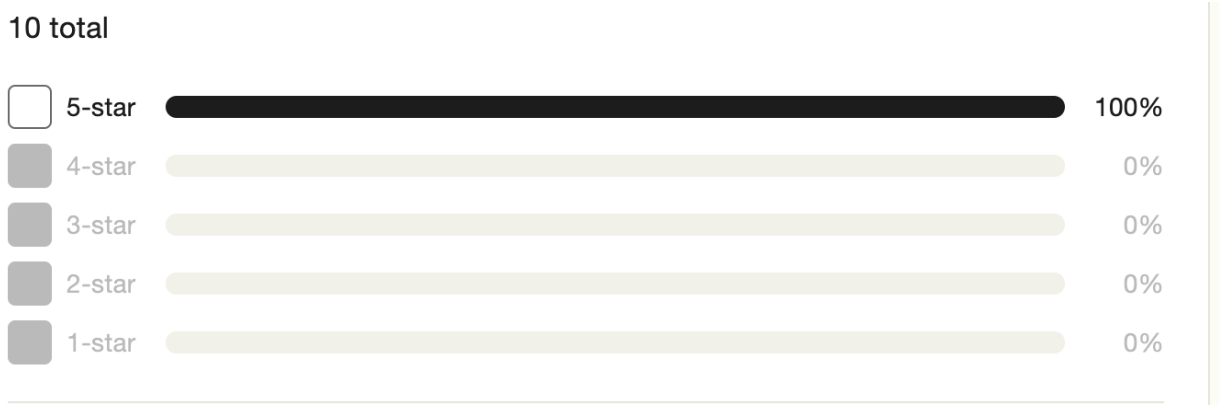
7. Augmented Reality (AR) Integration: Explore integrating AR technology into the chatbot interface for interactive product demonstrations, virtual try-ons, or immersive experiences, particularly beneficial for industries like retail, fashion, or interior design.

8. Continuous Learning and Improvement: Implement mechanisms for continuous learning and improvement, such as feedback loops, supervised learning algorithms, and user input validation, to enhance the chatbot's accuracy, relevance, and adaptability over time.

By incorporating these advanced features into your AI chatbot, you can not only compete with Answerly.io but also differentiate your product and provide added value to your users. Conducting user surveys, market analysis, and competitor benchmarking can further inform your feature prioritization and development roadmap.

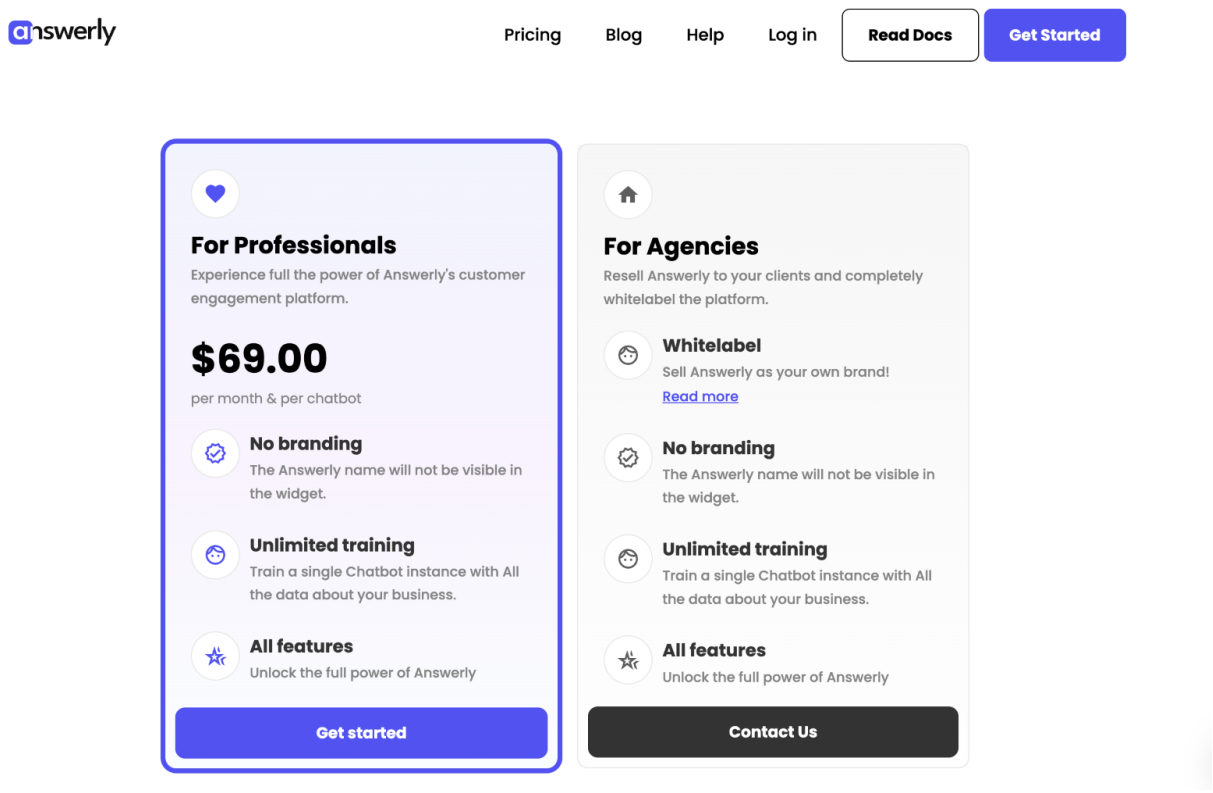
Links for Looking into Answerly.io Reviews/Customer Feedback

- https://www.trustpilot.com/review/answerly.io



- https://www.linkedin.com/pulse/answerly-review-lifetime-deal-worth-buying-bright-lifetime-deals-mehwf/
- https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.youtube.com/watch%3Fv%3D2EYsN8dLaRs&ved=2ahUKEwi2uKKYx5WGAxVA38kDHSBcAbsQwqsBegQIDhAG&usg=AOvVaw3QFSfXIc7bzrZ8G7I4UDC8

Pricing



Drift

Drift Chat is an intelligent tool that revolutionizes the way businesses interact with their customers. It utilizes advanced artificial intelligence algorithms to understand customer queries and provide relevant and helpful responses in real-time.

How does Drift Chatbot work?

Drift Chatbot works by analyzing customer messages and using natural language processing techniques to understand the intent behind each message. It can handle a wide range of customer queries, from simple FAQs to complex requests, and provides accurate and personalized responses.

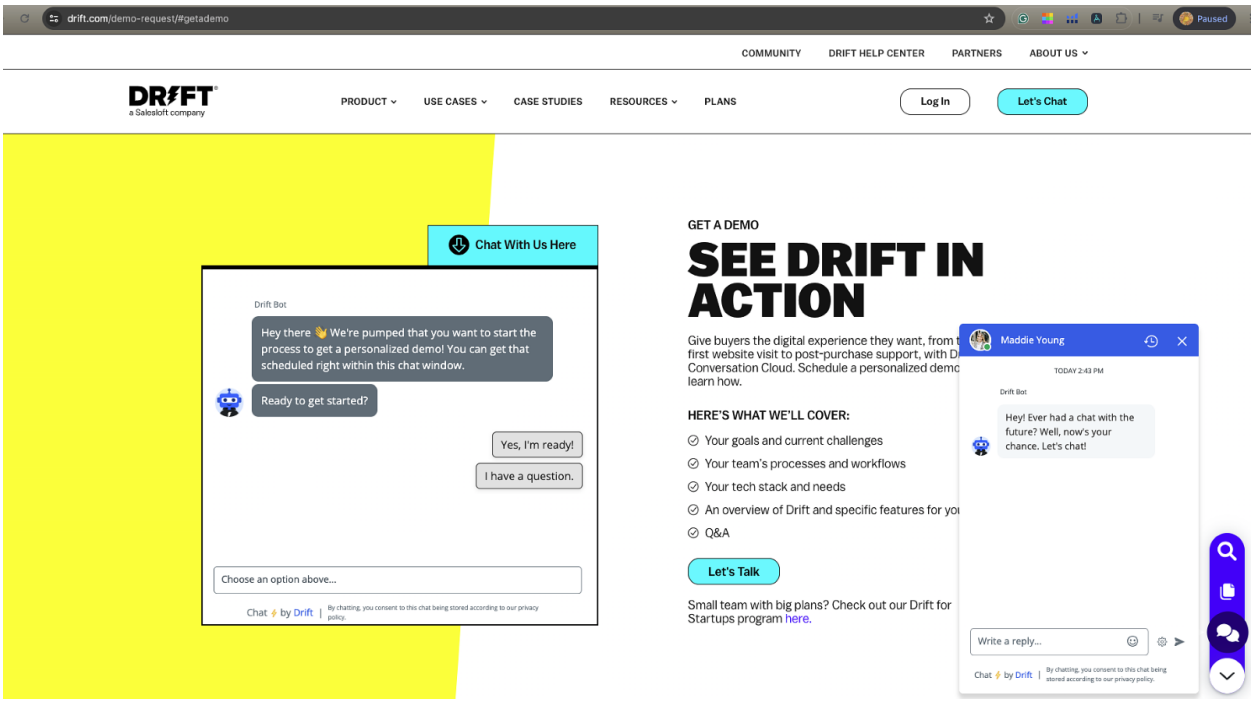
Features of Drift Chatbot:

- Instant Engagement: Drift Chatbot engages with customers immediately, ensuring quick response times and reducing the chance of customer frustration.
- Lead Qualification: The chatbot can ask relevant questions to qualify leads and provide valuable insights to your sales team.
- 24/7 Availability: Drift Chatbot is always available to handle customer queries, regardless of the time of day.
- Integration with CRM: The chatbot seamlessly integrates with your CRM system, allowing for efficient lead management and personalized customer experiences.
- Language Support: Drift Chatbot supports multiple languages, making it accessible to a global audience.

Key Performance Indicators (KPIs) of Drift Chatbot:

1. Response Time: Drift Chatbot reduces response time, ensuring faster customer resolutions and improved satisfaction.
2. Conversion Rate: The chatbot can contribute to an increase in lead conversion rates by engaging with potential customers and qualifying leads.
3. Customer Satisfaction: The chatbot's ability to provide accurate and helpful responses contributes to higher customer satisfaction levels.
4. Lead Generation: Drift Chatbot can generate leads by capturing customer information and qualifying them for further engagement.
5. Cost Savings: By automating customer service interactions, the chatbot reduces the need for manual intervention, resulting in cost savings for the business.

Overall, Drift Chatbot is a powerful tool that enhances customer engagement, improves lead management, and drives business growth by providing efficient and personalized customer e



ChatFuel

Chatfuel is a chatbot building platform that enables businesses to create AI-powered chatbots for various messaging platforms, including Facebook Messenger and Telegram.

How does Chat fuel work?

Chatfuel simplifies chatbot creation by providing a user-friendly interface and drag-and-drop builder. Users design conversation flows by adding message blocks and interactions. Once designed, chatbots can be integrated with platforms like Facebook Messenger or Telegram. Testing and optimization ensure smooth functionality, with ongoing support and updates provided by Chatfuel. Ultimately users can deploy their chatbots to engage with users and achieve their goals without coding knowledge.

Features of Chatfuel:

- Drag-and-Drop Interface: Users can easily design conversation flows using a simple drag and drop builder, making it accessible to users without coding knowledge.
- Multimedia Support: Chatfuel allows integration of images, videos, and GIFs into chatbot conversations, enhancing the user experience.
- Message Blocks: Users can create various message blocks such as text responses, buttons, quick replies, and user inputs to customize bot interactions.
- AI and Natural Language Processing (NLP): Incorporates AI and NLP capabilities to enable chatbots to understand and respond to user queries more intelligently.
- Integration with Messaging Platforms: Supports integration with popular messaging platforms like Facebook Messenger, Telegram, and others, allowing users to reach their audience across multiple channels.
- Analytics and Insights: Provides analytics and insights into chatbot performance, including metrics like user interactions, engagement rates, and more, helping users track and optimize bot performance.
- Templates and Plugins: Offers a variety of pre-built templates and plugins to expedite the chatbot creation process and add advanced functionalities.

Key Performance Indicators (KPIs) of Chatfuel:

- Engagement Metrics: Tracking metrics such as user interactions, session duration, and message open rates to gauge the level of engagement with the chatbot.
- Conversion Rates: Monitoring conversion metrics such as lead generation, sales, or other desired actions taken by users interacting with the chatbot.
- Retention Rates: Measuring the percentage of users who return to engage with the chatbot again over time, indicating user satisfaction and loyalty.
- User Satisfaction: Gathering feedback through surveys or sentiment analysis to assess user satisfaction with the chatbot's performance and interactions.
- Response Time: Monitoring the time taken for the chatbot to respond to user queries, ensuring prompt and efficient customer service.

- Error Rates: Tracking the frequency of errors or misunderstandings in user interactions to identify areas for improvement in the chatbot's functionality and performance.
- Usage Metrics: Analyzing metrics such as active users, session frequency, and user demographics to understand usage patterns and audience behavior.

SWOT analysis

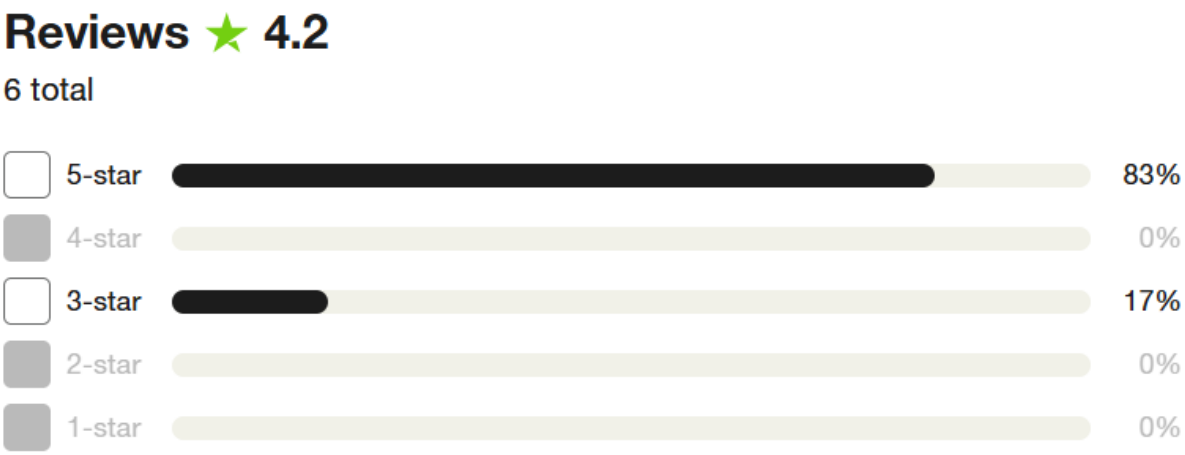
Strengths	**Weaknesses**
User-Friendly Interface Multichannel support AI and NLP capabilities Pre-built templates	Limited Customization Dependency on messaging platforms
Opportunities	**Threats**
Expansion of features Integration partnerships Industry-specific solutions	Competition Technology advancements

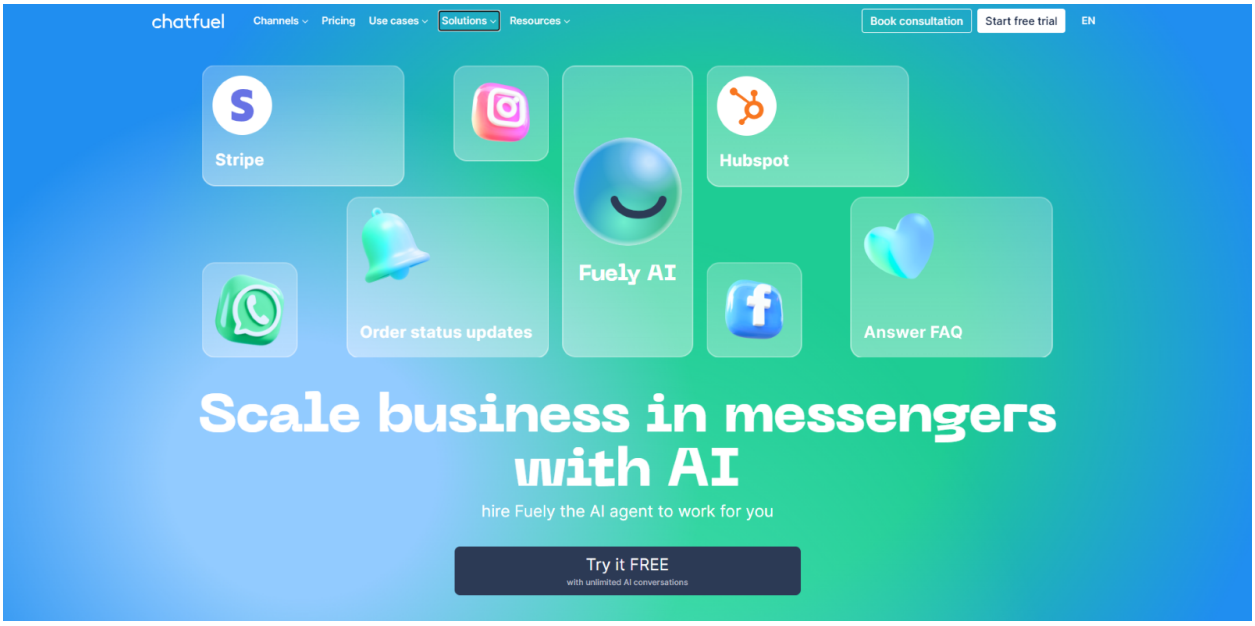
Overall, Chatfuel simplifies the process of chatbot creation by offering an intuitive interface and powerful features, allowing users to build effective and engaging chatbots for various purposes without the need for technical expertise.

https://www.youtube.com/watch?v=NcU7vTKykcY

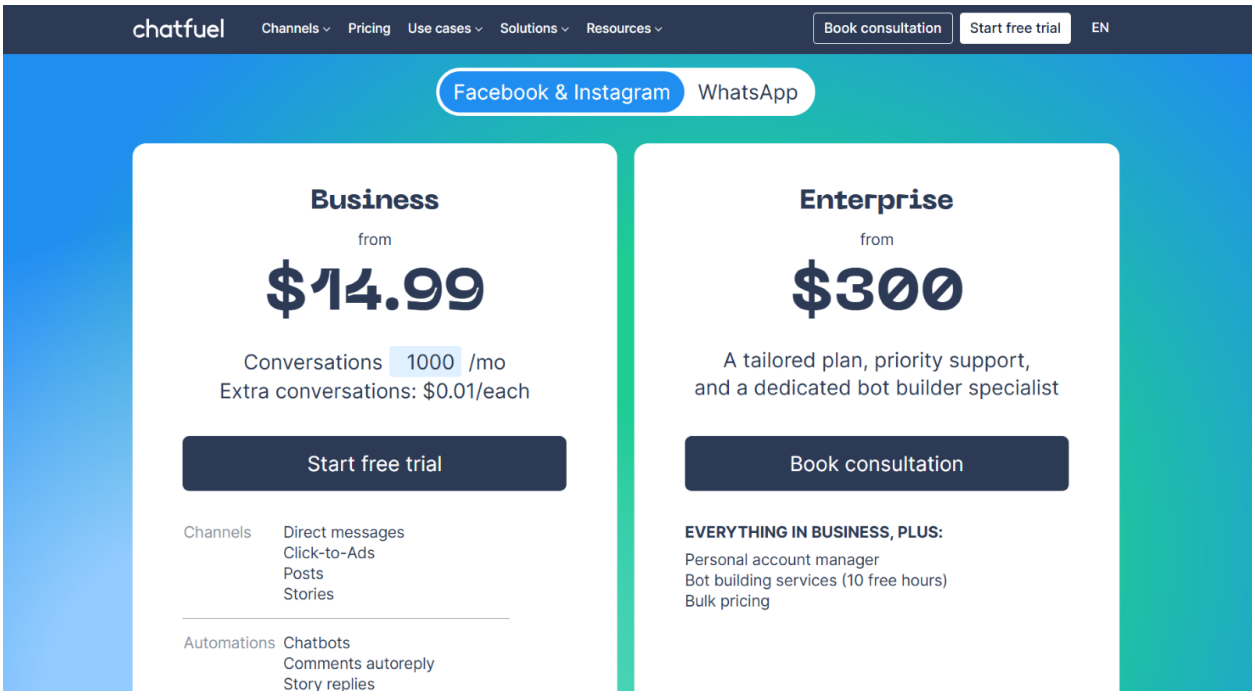
Chatfuel customer satisfaction:

Chatfuel Reviews | Read Customer Service Reviews of chatfuel.com (trustpilot.com)

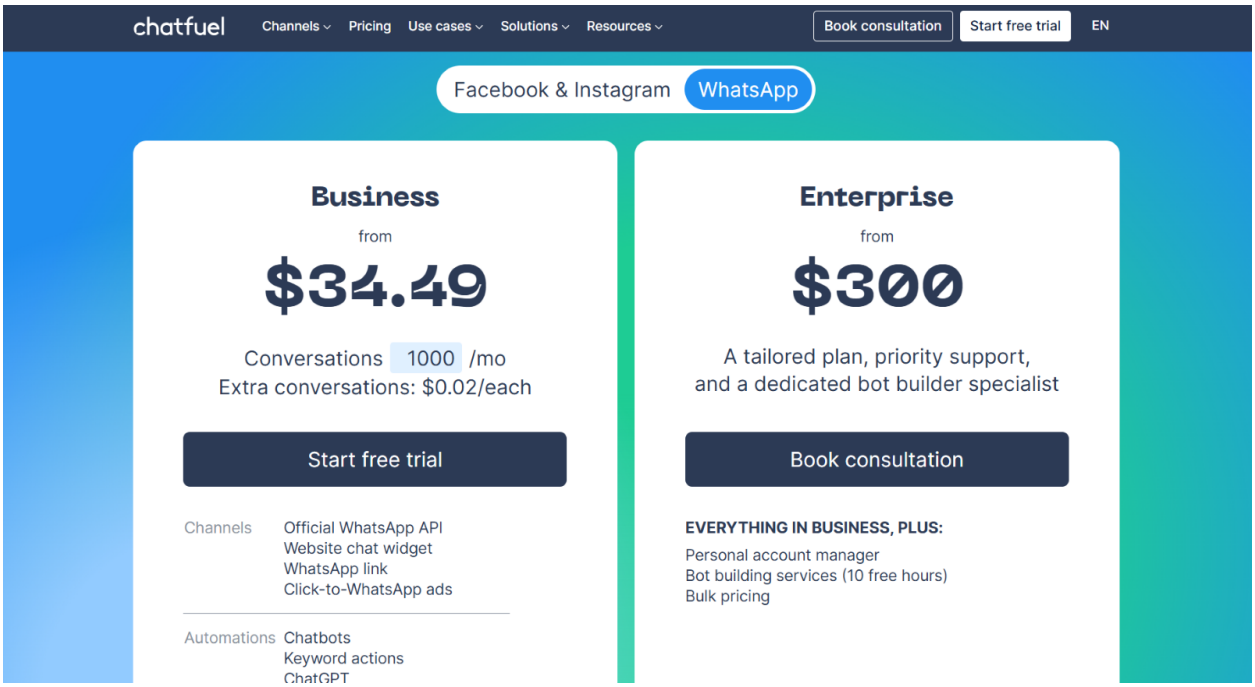




Pic: Home Screen



Pic: Pricing fb and instagram



Pic: Pricing for whatsapp

Summary

Chatfuel Pros

Easy-to-use Flow Builder

Unlimited Custom Field to Store User Data

Publish a chatbot on Facebook, Instagram & WhatsApp

Many marketing possibilities on Facebook, Instagram and WhatsApp

Built-in Live Chat

Integrations with ChatGPT, Zapier, Make, Shopify

Good analytics to improve answers of your chatbot

Cons

Duplicating chatbot flows across channels is not possible

It's hard to translate your chatbot to multiple languages

References

Chatfuel. (n.d.). Ai agents for automated sales: Meta's partner. https://chatfuel.com/

YouTube. (n.d.). *Chatfuel*. YouTube. https://www.youtube.com/@Chatfuel

Vishwakarma, Ashutosh. (2021). A Review & Comparative Analysis on Various Chatbots Design. International Journal of Computer Science and Mobile Computing. 10. 72-78. 10.47760/ijcsmc.2021.v10i02.011.

GHL

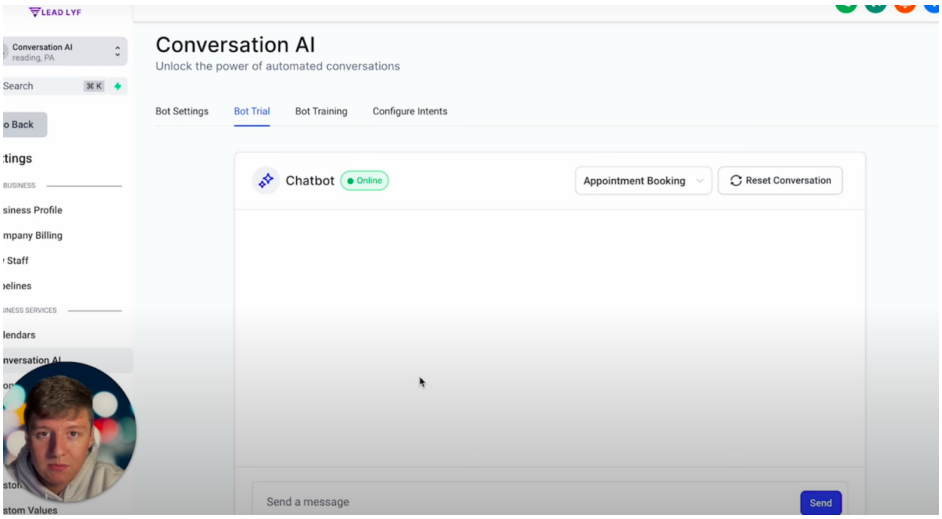
GoHighLevel is an all-in-one CRM platform designed primarily for marketing agencies and small businesses to streamline their operations and scale their services. The platform offers a comprehensive suite of tools, including CRM and pipeline management, two-way SMS and email communication, online scheduling, landing page and funnel builders, and reputation management.

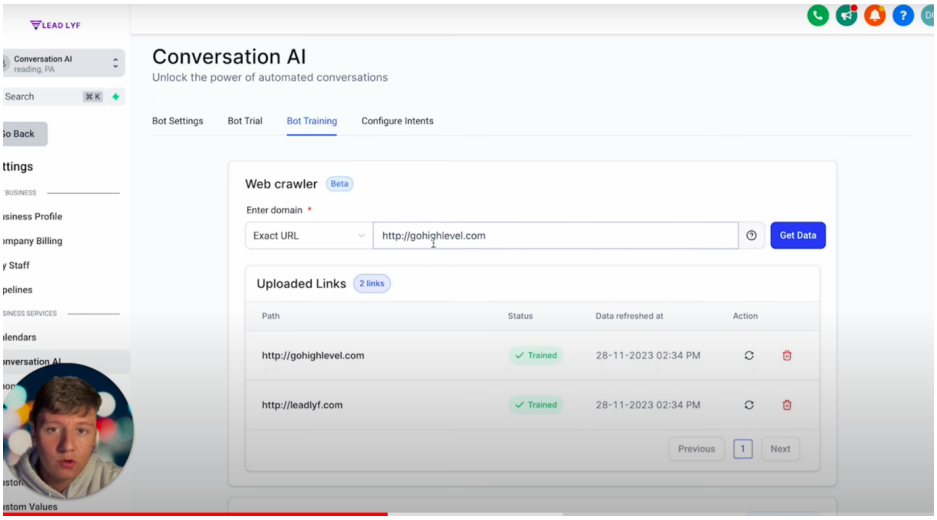
Key features of GoHighLevel include:

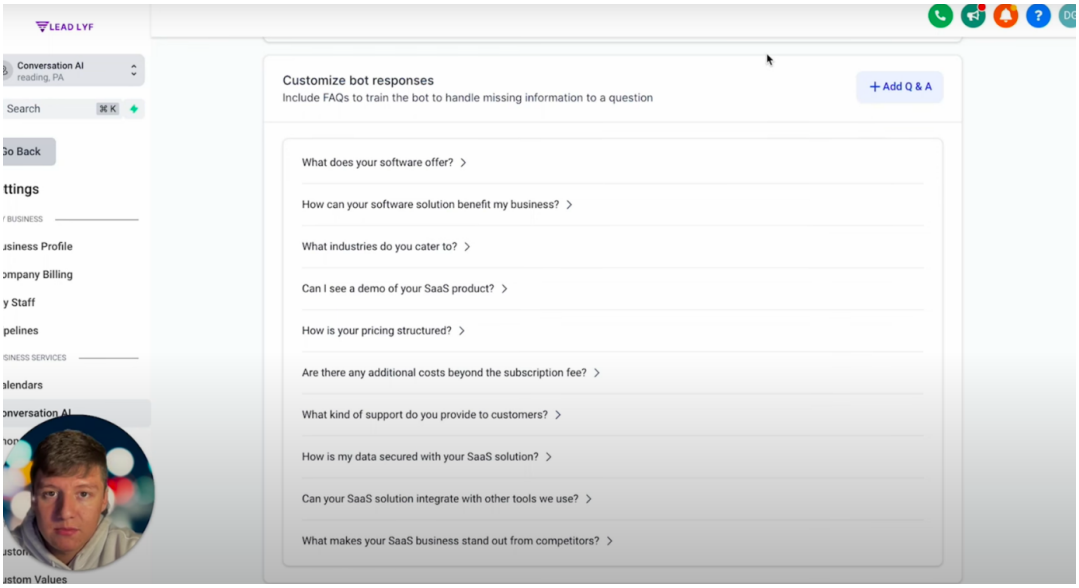
1. Lead Capture and Nurturing: Tools for building forms, surveys, landing pages, and websites to capture leads. Automated workflows can be set up to nurture these leads through email, SMS, voicemail drops, and call connects.
2. Multi-Channel Communication: Enables sending and receiving messages via SMS, emails, Facebook Messenger, Instagram DMs, Google Chats, and WhatsApp, with automation capabilities based on conversation keywords.
3. Customizable SaaS Platform: Agencies can white-label the platform with their branding, creating their own SaaS offerings and pricing models without needing to write code.
4. Sales and Marketing Automation: Automates various sales and marketing processes, including follow-ups, appointment scheduling, and analytics, helping businesses to close leads more efficiently.
5. Scalable Pricing: Designed with agency-friendly pricing, GoHighLevel allows unlimited users, contacts, and sub-accounts for a single monthly fee, which helps agencies manage costs predictably as they scale.

Additionally, GoHighLevel offers a user-friendly interface that is easy to set up and use, with robust customer support available 24/7 via email, chat, and live Zoom sessions. The platform is particularly favored by agencies due to its extensive feature set, scalability, and ability to simplify complex marketing and sales workflows into a single, integrated system

- GHL workflow: https://youtube.com/shorts/vvPfP0YoWqA?si=LSohkQDLgB8g2whR
- Bot training https://www.youtube.com/watch?v=sZrMi7hzVjE

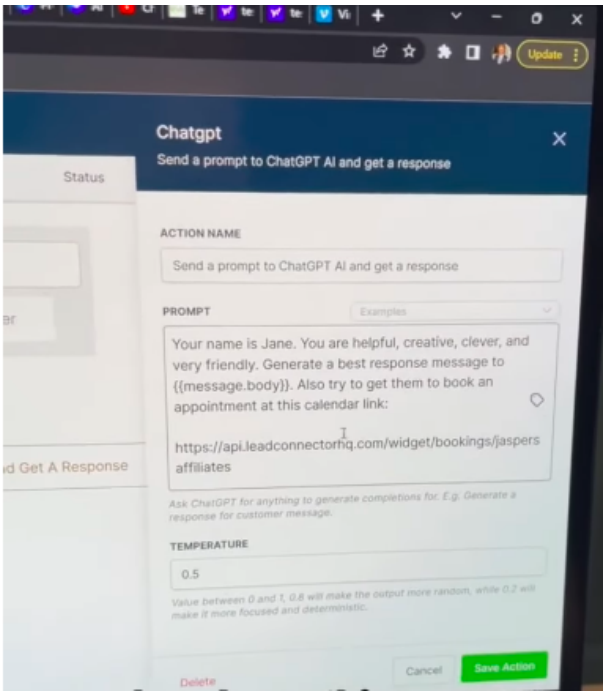

Conversation AI
Unlock the power of automated conversations
Bot Settings
Bot Trial
Bot Training
Configure Intents
Chatbot
Online
Appointment Booking
Reset Conversation
Send a message
Send

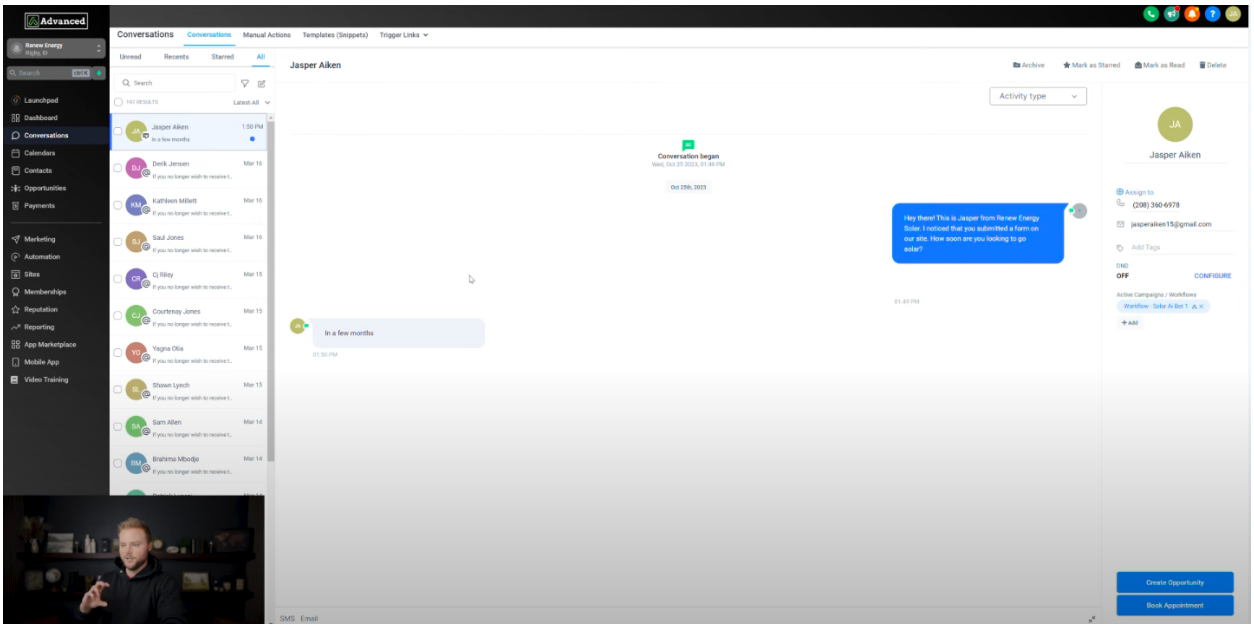

Conversation AI
Unlock the power of automated conversations
Bot Settings
Bot Trial
Bot Training
Configure Intents
Web crawler
Beta
Enter domain *
Exact URL
http://gohighlevel.com
Get Data
Uploaded Links
2 links
Path
Status
Data refreshed at
Action
http://gohighlevel.com
Trained
28-11-2023 02:34 PM
http://leadlyf.com
Trained
28-11-2023 02:34 PM
Previous
1
Next

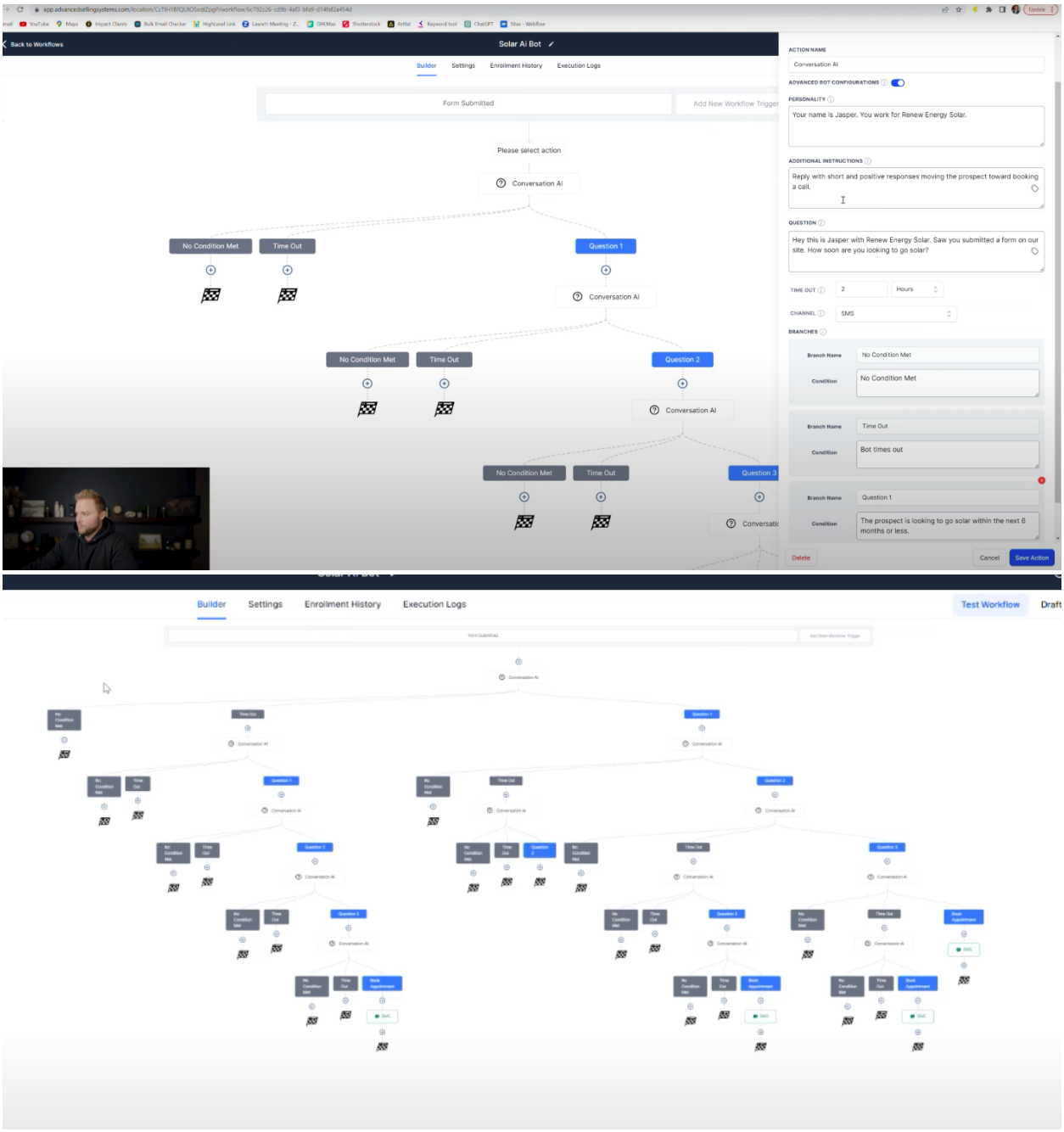

Customize bot responses
Include FAQs to train the bot to handle missing information to a question
+ Add Q & A
What does your software offer?
How can your software solution benefit my business?
What industries do you cater to?
Can I see a demo of your SaaS product?
How is your pricing structured?
Are there any additional costs beyond the subscription fee?
What kind of support do you provide to customers?
How is my data secured with your SaaS solution?
Can your SaaS solution integrate with other tools we use?
What makes your SaaS business stand out from competitors?

GoHighLevel's Conversation AI Bot can be trained to improve its responses by using either automatic or manual methods. Automatic training involves crawling web content from specific URLs, domains, or paths, or using Google Docs with appropriate sharing settings. Manual training allows the addition of custom question-answer pairs. Continuous refinement is achieved through feedback mechanisms, enabling the bot to learn from user interactions.

https://help.gohighlevel.com/support/solutions/articles/155000000996-training-the-conversation-ai-bot






Back to Workflows
Solar Ai Bot
Builder
Settings
Enrollment History
Execution Logs
Form Submitted
Add New Workflow Trigger
Please select action
Conversation AI
No Condition Met
Time Out
Question 1
Conversation AI
No Condition Met
Time Out
Question 2
Conversation AI
No Condition Met
Time Out
Question 3
ACTION NAME
Conversation AI
ADVANCED BOT CONFIGURATIONS
PERSONALITY
Your name is Jasper. You work for Renew Energy Solar.
ADDITIONAL INSTRUCTIONS
Reply with short and positive responses moving the prospect toward booking a call.
QUESTION
Hey this is Jasper with Renew Energy Solar. Saw you submitted a form on our site. How soon are you looking to go solar?
TIME OUT
2
Hours
CHANNEL
SMS
BRANCHES
Branch Name
No Condition Met
Condition
No Condition Met
Branch Name
Time Out
Condition
Bot times out
Branch Name
Question 1
Condition
The prospect is looking to go solar within the next 6 months or less.
Delete
Cancel
Save Action

Builder
Settings
Enrollment History
Execution Logs
Test Workflow
Draft

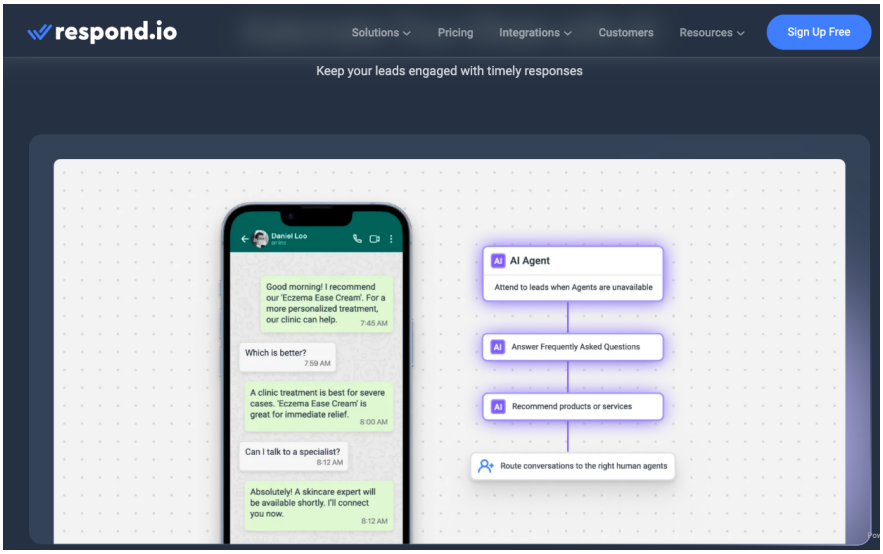
Respond io

Respond.io is a comprehensive messaging platform designed to streamline and enhance business communication across multiple channels. It integrates various messaging apps, such as WhatsApp, Facebook Messenger, Instagram, Telegram, and more, into a single unified interface. This allows businesses to manage customer interactions, support, marketing, and sales conversations from one central location.

Key features of respond.io include:

1. Omnichannel Messaging: Consolidates messages from various platforms, enabling businesses to respond promptly and consistently, regardless of the customer's preferred channel.
2. Automation and AI: Automates repetitive tasks and workflows with the help of Respond AI, which can handle routine inquiries, qualify leads, and expedite conversions.
3. Conversational Sales and Marketing: Enhances sales strategies and marketing efforts by engaging with customers directly through chat, converting leads more efficiently.
4. Customer Support: Provides tools to improve customer support, ensuring timely and effective responses, thereby boosting customer satisfaction and loyalty.
5. Integration Capabilities: Connects with various third-party applications like CRM systems (HubSpot, Salesforce), eCommerce platforms (WooCommerce), and workflow automation tools (Zapier), enhancing overall business operations.

Respond.io aims to optimize and simplify business communication, making it easier for companies to engage with their customers, automate processes, and ultimately drive better business outcomes .



Hellorep ai

HelloRep.ai offers an advanced AI-driven platform designed to enhance sales and support for eCommerce stores, particularly those using Shopify. The core product, Rep AI, acts as an "AI Concierge," engaging with visitors to improve conversions and customer service.

Key Features:

1. Behavioral AI: Detects when customers are about to leave the site and proactively engages them to increase conversions (REP AI Home) (REP AI Home).
2. Product Assistance: Provides product recommendations, answers FAQs, and assists with order management directly within the chat interface (REP AI Home) (REP AI Home).
3. 24/7 Operation: The AI works around the clock to handle customer interactions, significantly reducing the need for human intervention and support tickets (REP AI Home) .
4. Customizable and On-Brand Interactions: Trained to understand the specific product catalog and maintain the brand's voice and personality across different languages (REP AI Home) (REP AI Home).
5. Integration: Seamlessly integrates with existing tools like Shopify, Gorgias, Zendesk, and others to ensure a smooth handoff to human support if needed (REP AI Home).
6. Real-time Analytics: Provides a dashboard for monitoring performance metrics such as conversion rates, total sales, and average order values influenced by the AI interactions (REP AI Home).

HelloRep.ai aims to transform the eCommerce experience by offering personalized, scalable, and effective customer engagement solutions.

Overview: Entire Application

Bot-Buddy is an AI-based chatbot solution designed to revolutionize customer support for businesses. By leveraging advanced artificial intelligence technology, this product will offer 24/7 customer support, enhancing the way businesses engage with their customers.

Scope: Trying To Achieve

The scope of this project is to develop an advanced chatbot application designed specifically for providing customer support through a customizable and embeddable widget for our clients' websites. This chatbot will be capable of handling multiple types of queries by utilizing specialized agents for different domains such as logistics, customer service, and finance. Each client will have the ability to upload documents specific to their business, enabling the chatbot agents to become proficient in the client's industry and provide highly accurate and contextual support.

The goal of this project is to create a versatile, intelligent, and secure chatbot solution that enhances the customer support capabilities of our clients, leading to improved customer satisfaction and operational efficiency.

In addition to enhancing customer support capabilities, the goal of this project is also to provide a scalable solution that can handle a high volume of customer interactions simultaneously, reducing response times and ensuring customers receive immediate assistance.

Requirements (Within time frame/ What are you planning on releasing)

Features:

1. **Enhance Landing Page (Sprint 1)**
2. **Sign-up/Login Feature using email = User can successfully log in or sign up (Sprint 1)**
3. **Dashboard - Create AI Agent (Sprint 2)**
4. **Dashboard - Import Knowledge Base (Sprint 2)**

Key Features:

1.Document Upload and Training:

- Clients can upload business-specific documents to the platform.
- The chatbot will use these documents to train and adapt its responses, ensuring it provides relevant and accurate information tailored to the client's business.

2.Embeddable Widget:

- The chatbot will be available as a widget that can be easily embedded into the client's website.

3.Contextual Understanding and Response:

- The chatbot will leverage natural language processing (NLP) to understand and respond to customer inquiries contextually.
- Continuous learning capabilities will allow the chatbot to improve its responses over time based on user interactions and feedback.

4.Security and Privacy:

- The application will adhere to industry-standard security and privacy practices to protect client data.
- Data encryption, secure authentication, and regular security audits will be implemented to maintain data integrity and confidentiality.

Out of scope

1. Multiple Specialized Agents (more than 2):

    - The application will support the creation of multiple chatbot agents, each specializing in a specific area (e.g., logistics, customer service, finance).
    - Clients can assign specific agents to handle particular types of inquiries based on their specialization.

2. Analytics and Reporting:

    - The application will include analytics and reporting tools to provide clients with insights into chatbot performance, user interactions, and common inquiries.
    - These insights will help clients refine their support strategies and improve customer satisfaction.

3. Customizable widget

    - Customization options will be provided to match the look and feel of the client's website and brand.

4. Integration with Existing Systems:

    - The chatbot will be designed to integrate seamlessly with existing client systems such as CRM, inventory management, and other relevant tools.
    - This integration will ensure the chatbot has access to up-to-date information and can provide accurate responses.

Non-Functional Requirements:

1. **Performance:**
    - The chatbot should respond to user queries within 2 seconds under normal operating conditions.
    - The system should handle a minimum of 1,000 concurrent users without performance degradation.
2. **Scalability:**

- The application must be scalable to support the growing number of clients and users.
- It should be capable of scaling horizontally to handle increased load.

3. **Reliability:**
   - The chatbot service must have an uptime of 99.9%.
   - Automated failover mechanisms should be in place to ensure continuous service availability.
4. **Security:**
   - All data transmitted between the client and server must be encrypted using industry-standard protocols (e.g., TLS).
   - The system must comply with data protection regulations such as GDPR and CCPA.
   - Regular security audits and vulnerability assessments must be conducted.
5. **Maintainability:**
   - The codebase should be modular and well-documented to facilitate easy maintenance and updates.
   - Continuous integration and continuous deployment (CI/CD) pipelines should be implemented for efficient code deployment and testing.
6. **Usability:**
   - The user interface of the chatbot widget should be intuitive and easy to use.
   - Documentation and tutorials should be provided to help clients integrate and customize the chatbot.
7. **Interoperability:**
   - The chatbot must be able to integrate with various third-party systems such as CRMs, inventory management systems, and other relevant tools through APIs.
   - Standard data exchange formats (e.g., JSON, XML) should be used to facilitate integration.
8. **Portability:**
   - The chatbot widget should be easily embeddable on various platforms (e.g., websites, mobile apps).
   - The system should support deployment in different environments for clients (e.g. dev, staging, prod).
9. **Compliance:**
   - The application must comply with all relevant industry standards and regulations.
   - Regular compliance checks and updates should be performed to adhere to changing regulations.
10. **Localization:**
    - The chatbot should support multiple languages to cater to a global audience.
    - Localization settings should be customizable by the client.
11. **Backup and Recovery:**

- Regular data backups must be performed to prevent data loss.
- Disaster recovery plans should be in place to restore services within an acceptable time frame after a failure.

These nonfunctional requirements ensure that the chatbot application is robust, secure, and user-friendly, providing a reliable and high-quality service to clients and their customers.

Technical Stack:

Ideal Tech Stack

The ideal tech stack for a customer support chatbot application should be chosen based on factors such as scalability, performance, ease of development, and integration capabilities. Here's a suggested tech stack along with the rationale for each technology:

1. **Front-end:**
    - **React.js:**
        - React is a popular JavaScript library for building user interfaces, particularly single-page applications. It allows for the creation of a responsive and dynamic user interface. Its component-based architecture promotes reusability and maintainability.
2. **Back-end:**
    - **Node.js with Express.js:**
        - Node.js is a runtime environment that enables JavaScript to be used for server-side scripting. Express.js is a minimalist web framework for Node.js that simplifies the development of robust APIs. Together, they offer high performance and are well-suited for handling asynchronous operations, making them ideal for real-time chatbot interactions.
        - **ChatGPT AI :**

          ChatGPT is an AI conversational model developed by OpenAI, known for its ability to generate human-like text responses. It's based on the GPT (Generative Pre-trained Transformer) architecture, which uses deep learning algorithms to understand and generate language. ChatGPT has been trained on a diverse range of internet text, allowing it to respond to various prompts with informative and contextually relevant answers. It can perform tasks like answering questions, writing essays, summarizing information, and even creating code.

3. **Database:**
    - **MongoDB:**

- MongoDB is a NoSQL database that provides flexibility in storing unstructured data. Its schema-less nature is suitable for storing diverse and dynamic data generated by chat interactions. It also offers high scalability and performance.

4. **Natural Language Processing (NLP):**
   - **ChatGPT:**
     - ChatGPT is an advanced NLP model developed by OpenAI, which utilizes deep learning techniques to produce human-like text. It's designed to engage in conversation, answer questions, and perform a variety of language-based tasks. ChatGPT stands out for its ability to understand context, generate coherent and contextually relevant responses, and even exhibit a degree of creativity in its outputs.
5. **Machine Learning Framework:**
   - **ChatGPT:**
     - ChatGPT itself is an AI language model developed by OpenAI, designed to understand and generate human-like text. It's capable of engaging in conversations, answering questions, and providing information across various topics.
6. **API Layer:**
   - **node.js:**
     - The platform provides a rich library of various JavaScript modules which simplifies the development of web applications. Node.js is widely used for developing backend services like APIs, web app frameworks, and more4. Its ability to handle numerous simultaneous connections with high throughput makes it a popular choice for online gaming, communication applications, and other environments where real-time data is crucial.
7. **Hosting and Deployment:**
   - **Herokus:**
     - Heroku is a cloud platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. It's designed to be flexible and easy to use, providing a fully managed experience that allows developers to focus on their core product rather than the infrastructure. Heroku supports several programming languages and offers a range of services and tools to enhance developer productivity.

8. **Authentication and Security:**
   - **Yet to figure out:**

9. **Deployment:**
    - **Github:**
        - The deployment process can be triggered by various GitHub events, such as a push to a branch or a pull request. It can also be configured to require manual approval, ensuring that deployments are controlled and secure.
10. **Testing:**
    - **Replit:**
        - The ELK stack provides powerful tools for log management, data analysis, and visualization. Elasticsearch enables fast search and analytics, Logstash handles data processing and transformation, and Kibana offers interactive data visualization.

By utilizing this tech stack, the chatbot application can achieve high performance, scalability, ease of development, and robust integration capabilities, ensuring a seamless and efficient customer support experience for clients and their users.

Avoided Tech Stacks:

1. AngularJS:
    - Why Avoided: Although once popular, AngularJS is now considered outdated and has been replaced by Angular (a complete rewrite). Its steep learning curve and complex architecture make it less favorable compared to React.js, which offers simpler and more flexible development.
2. jQuery:
    - Why Avoided: jQuery is less relevant in modern web development due to the rise of powerful frameworks like React, Angular, and Vue.js. It doesn't provide the component-based architecture and state management needed for complex applications like a chatbot.
3. PHP with Laravel:
    - Why Avoided: While PHP and Laravel are robust for many web applications, they are not as well-suited for real-time, event-driven applications compared to Node.js. Node.js offers better performance for handling asynchronous operations, which are crucial for a responsive chatbot.
4. MySQL:
    - Why Avoided: MySQL, being a relational database, is less flexible for handling the unstructured and dynamic data typically generated by chatbot interactions. MongoDB, as a NoSQL database, offers better performance and scalability for these use cases.
5. Apache HTTP Server:

- Why Avoided: Apache is known for its stability but is generally slower than modern alternatives like Nginx, which offers better performance and scalability for handling high concurrency, making it more suitable for real-time applications like a chatbot.

6. Ruby on Rails:
    - Why Avoided: Ruby on Rails, while excellent for rapid development, can face scalability issues under high loads. Node.js, with its non-blocking I/O model, provides better performance and scalability for applications that require real-time interaction.
7. TensorFlow.js:
    - Why Avoided: While TensorFlow.js allows for machine learning in JavaScript, it is not as mature or efficient for heavy NLP tasks compared to Python-based TensorFlow or PyTorch. Python's ecosystem offers more robust libraries and tools for NLP and AI development.
8. PostgreSQL:
    - Why Avoided: Although PostgreSQL is a powerful relational database with advanced features, its relational nature is not as suited for the flexible schema requirements of a chatbot. MongoDB's document-based approach is more adaptable to the varying data structures of chat interactions.
9. Firebase Realtime Database:
    - Why Avoided: Firebase Realtime Database is great for simple applications but may not scale as well for complex, high-traffic applications. Its data structure can become challenging to manage with complex relationships and queries compared to MongoDB or Redis.
10. Flask:
    - Why Avoided: Flask is a lightweight Python web framework that's great for simple applications, but it lacks the built-in features and scalability of Node.js with Express.js for handling real-time, high-load applications like a chatbot.
11. Microsoft Azure Bot Service:
    - Why Avoided: While Azure Bot Service provides a comprehensive solution for building chatbots, it can be more restrictive in terms of customization and may have higher costs compared to building a custom solution with the chosen tech stack.
12. Cassandra:
    - Why Avoided: Cassandra is a powerful distributed database but is often overkill for applications that do not require massive scale and complex data distribution. MongoDB and Redis provide a more balanced and manageable approach for the expected scale of this chatbot application.

By avoiding these technologies, we aim to ensure the chatbot application is built on a modern, flexible, and scalable tech stack that aligns with the project's performance, maintainability, and integration requirements.

Security Concerns and Mitigations Using Our Tech Stack

Authentication (JWT)

Security Concerns:

- Token Theft: JWT tokens can be stolen if not properly secured, leading to unauthorized access.
- Token Expiry: If tokens have long expiration times, stolen tokens can be used for an extended period.

Mitigations:

- Secure Storage: Store JWT tokens securely in HTTPS-only cookies or local storage with appropriate security settings.
- Short Lifespan: Set short expiration times for JWT tokens and use refresh tokens for prolonged sessions.
- Token Revocation: Implement mechanisms to revoke tokens in case of suspicious activity.

HTTPS Requests

Security Concerns:

- Man-in-the-Middle Attacks: Data transmitted over HTTP can be intercepted and tampered with.
- Data Integrity: Ensuring data integrity during transmission is crucial to prevent unauthorized modifications.

Mitigations:

- SSL/TLS Encryption: Enforce HTTPS for all data transmissions to encrypt data between the client and server.
- Certificates Management: Regularly update SSL/TLS certificates and use trusted Certificate Authorities (CAs).

ChatGPT Safe Document Upload

Security Concerns:

- Data Privacy: Uploaded documents may contain sensitive information that needs to be protected.
- Malicious Files: There is a risk of malicious files being uploaded and executed.

Mitigations:

- Data Encryption: Encrypt documents during upload and storage to protect sensitive information.
- File Validation: Implement robust file validation and scanning processes to detect and prevent malicious files.
- Access Controls: Restrict access to uploaded documents to authorized personnel and systems only.

Prompt Not to Give Private Info to Users

Security Concerns:

- Data Leakage: The chatbot might inadvertently share private or sensitive information with users.
- Social Engineering: Users may attempt to trick the chatbot into revealing confidential information.

Mitigations:

- Data Masking: Ensure the chatbot never reveals sensitive information by masking or omitting such data in responses.
- Prompt Filtering: Implement filtering mechanisms to identify and block prompts that seek private information.
- Training and Monitoring: Train the chatbot model to recognize and handle sensitive data appropriately and continuously monitor interactions for compliance.

General Security Measures

1. Input Validation:
    - Concern: Injections and malicious inputs can exploit vulnerabilities.
    - Mitigation: Validate and sanitize all user inputs to prevent SQL injection, cross-site scripting (XSS), and other injection attacks.
2. Access Controls:
    - Concern: Unauthorized access to sensitive data and functionalities.
    - Mitigation: Implement role-based access controls (RBAC) to ensure users only access data and actions they are authorized for.
3. Logging and Monitoring:
    - Concern: Security breaches going undetected.
    - Mitigation: Maintain detailed logs of all interactions and monitor for unusual activities, implementing alerts for potential security incidents.

4. Regular Security Audits:
    - Concern: Undetected vulnerabilities and outdated security practices.
    - Mitigation: Conduct regular security audits and penetration testing to identify and fix vulnerabilities.
5. User Education:
    - Concern: Users unknowingly compromising security.
    - Mitigation: Educate users on best security practices, such as recognizing phishing attempts and securing their accounts.
6. Dependency Management:
    - Concern: Vulnerabilities in third-party libraries and dependencies.
    - Mitigation: Regularly update and audit third-party libraries and dependencies for security patches and vulnerabilities.

By addressing these security concerns and implementing the corresponding mitigations, we can ensure a robust security posture for our chatbot application, protecting both the application and its users from potential threats.

Roadmap

1. **Enhance Landing Page** - Sprint 2

2. **Sign up / Login Feature using Email** - Sprint 1

3. **Dashboard - Create AI Agent** - Sprint 2

4. **Dashboard - Import Knowledge Base** - Sprint 2
   Businesses can easily import existing content or data from their knowledge base into their Agent’s Knowledge Hub.

   This feature streamlines the process of training the chatbot and ensures that it has access to relevant information to provide accurate responses to customer inquiries.
   The Knowledge Hub serves as the AI brain of the chatbot platform, allowing businesses to train their chatbot with customized knowledge about their products, services, and industry. This feature enables businesses to continuously improve the accuracy and relevance of the chatbot's responses over time.

5. **AI-Powered Chatbot** - Sprint 3
   Our platform will feature an AI-powered chatbot that can interact with customers in real-time, providing accurate and timely responses to their inquiries. Leveraging advanced

natural language processing (NLP) algorithms, the chatbot will understand and interpret customer queries, enabling seamless communication.

6. **Conversation History**
   Businesses can monitor and review all live chats and interactions between the chatbot and customers in real-time. This feature provides valuable insights into customer behavior, frequently asked questions, and areas for improvement, enabling businesses to optimize their chatbot's performance.

7. **Instant Embed**
   Our platform offers a streamlined process for embedding the chatbot widget onto business websites quickly and effortlessly. This feature enables businesses to launch their AI-powered chatbot with minimal setup time, allowing them to start providing 24/7 customer support without delay.

Next Steps

With the MVP features outlined above, we aim to test the market viability of our chatbot platform and gather feedback from early adopters. By focusing on delivering personalized, efficient, and 24/7 customer support, we strive to address the pain points faced by businesses and enhance their ability to engage with customers effectively in the digital age.

Expectations

1. Research and use competitor products to get an idea of these chatbot tools
2. Check their overall user flow, what is happening, how and try answering some chatbots

Roadmap for the coming year

1. **Answers with Links**
   Allow the chatbot to include relevant web links in its responses, providing users with additional resources or information to explore further. This feature enhances the depth of information available to users and increases the utility of the chatbot.

2. **Personalization Options**
   Businesses will have the ability to personalize their chatbot by assigning a unique name and character, reflecting their brand identity. Additionally, they can customize the chatbot's traits and behavior to create engaging and interactive conversations with customers, fostering a more personalized experience.

3. **Widget Customization and White Labeling**
   Businesses can customize the appearance and behavior of the chatbot widget to seamlessly integrate it into their website design. This feature includes options to adjust the widget's placement, colors, width, and shadow, ensuring a cohesive user experience across all touchpoints.
   Enhance online presence by customizing the domain of chatbot links with white labeling feature. This feature allows businesses to personalize chatbot URLs, aligning them with their brand and creating a seamless user experience.

4. **Discover Content**
   Analyze existing knowledge within the Knowledge Hub and generate new questions that the chatbot may struggle with, helping to enhance its knowledge and effectiveness. This feature assists businesses in continuously improving the chatbot's capabilities and ensuring it remains up-to-date with

5. **Workspace Collaboration**
   Enable team collaboration within the platform, allowing multiple users to collectively manage and improve the performance of the chatbot. This feature facilitates knowledge sharing, collaboration, and coordination among team members for optimal chatbot performance.

6. **Chatbot SDK**

Adapt the chatbot widget to suit unique requirements and unlock limitless possibilities by leveraging the products' SDK. This feature empowers businesses to extend the functionality of the chatbot and integrate it with custom applications or platforms.

Success Metrics (Work with stakeholders to define)

1. **User Satisfaction:**

- Metric:
    - Survey: Rate the accuracy of summaries out of 10 (1 being worst, 10 being best).
    - Reviews: Rating on a scale from 1 to 5 (1 being worst, 5 being best).
    - Customer Interviews: Qualitative feedback from direct interactions.
- Why: Measures user satisfaction with interactions and assistance.

2. **Revenue Targets:**

- Metric:

  i) Specification: Define target revenue for a period (month, quarter, year).

  ii) Subscription Model:

    - Paid Version for professionals and companies

  iii) Market Adaptation: Adjust revenue targets based on market conditions, competition, and trends.

- Why: Measure company's financial performance and guiding its strategic direction.

3. **Conversion Rate:**

- Metric: Target conversion rates for key user actions, such as subscription sign-ups.
- Why: Measures the product's effectiveness in driving desired actions.

4. **Retention Rate:**

- Metric: Desired retention rates over specific periods (e.g., one month, six months).
- Why: Indicates how well the product maintains user engagement.

5. **Churn Rate:**

- Metric: Frequency of customer loss.
- Why: Identifies areas for improvement to retain customers.

6. **Response Time:**
- Metric: Average response time of the chatbot to user queries.
- Why: Ensures timely assistance and responsiveness.

7. **Resolution Rate:**
- Metric: Percentage of user inquiries resolved by the chatbot without human intervention.
- Why: Indicates the chatbot's efficiency and effectiveness.

8. **Engagement Rate:**
- Metric: Number of interactions per user session.
- Why: Measures user engagement and interaction quality.

9. **Fallback Rate:**
- Metric: Percentage of interactions escalated to a human agent.
- Why: Identifies areas for chatbot improvement or additional training.

10. **Usage Metrics:**
- Metric: Number of active users, total interactions, and peak usage times.
- Why: Provides insights into the chatbot’s adoption and usage patterns.

11. **Error Rate:**
- Metric: Number of errors or failed interactions divided by total interactions.
- Why: Identifies performance issues for improvement.

12. **Human Agent Hand-Off Rate:**
- Metric: Percentage of interactions transferred to human agents.
- Why: Measures the chatbot’s autonomy and hand-off effectiveness.

13. **Time to Resolution:**
- Metric: Average time to resolve an inquiry, whether by chatbot or human.
- Why: Indicates overall support efficiency.

14. Cost Savings:
- Metric: Reduction in support costs due to the chatbot.
- Why: Measures financial benefits of chatbot deployment.

15. Learning and Improvement Rate:
- Metric: Frequency and effectiveness of updates and improvements.
- Why: Ensures continuous improvement based on feedback and performance analytics.

These success metrics will help in evaluating the performance, effectiveness, and impact of the chatbot application, guiding further development and optimization efforts.

Anticipated Risks

1. **Development Risks:**
    - **Technical Complexity:** Developing a sophisticated NLP-based chatbot with multiple specialized agents and seamless integration can be technically challenging.
        - **Mitigation:** Invest in skilled developers, use established frameworks, and implement thorough testing.
    - **Integration Issues:** Ensuring compatibility with various third-party systems and existing client infrastructure may pose challenges.
        - **Mitigation:** Develop robust APIs, provide comprehensive documentation, and offer integration support services.
2. **Data Security and Privacy:**
    - **Data Breaches:** Handling sensitive customer data poses a risk of data breaches and compliance violations.
        - **Mitigation:** Implement strong encryption, secure authentication, and regular security audits. Ensure compliance with regulations like GDPR and CCPA.
    - **Client Data Handling:** Improper handling of client-specific documents and training data could lead to confidentiality issues.
        - **Mitigation:** Establish clear data handling protocols and ensure all data processing complies with client agreements and legal requirements.
3. **Deployment Risks:**
    - **Scalability Issues:** The application might face performance bottlenecks or downtime during high traffic periods.
        - **Mitigation:** Use scalable cloud infrastructure, implement load balancing, and conduct stress testing.
    - **Deployment Failures:** Issues during deployment can lead to service interruptions.

        - **Mitigation:** Use CI/CD pipelines, automated testing, and rollback mechanisms.
4. **Cost Risks:**
    - **Development Costs:** The cost of developing and maintaining a sophisticated chatbot can be high.
        - **Mitigation:** Plan a detailed budget, secure sufficient funding, and monitor expenses closely.
    - **Operational Costs:** Ongoing costs for cloud services, maintenance, and support can add up.
        - **Mitigation:** Optimize resource usage, negotiate better rates with service providers, and explore cost-effective solutions.
5. **Client Acquisition and Retention Risks:**
    - **Market Competition:** The market for chatbot solutions is competitive, with many established players.
        - **Mitigation:** Differentiate the product through unique features, superior performance, and exceptional customer support.
    - **Client Onboarding:** Clients may face challenges in integrating and customizing the chatbot for their specific needs.
        - **Mitigation:** Offer comprehensive onboarding support, training, and customization services.
6. **User Experience Risks:**
    - **User Acceptance:** Users may be reluctant to interact with a chatbot or may find it less effective than human support.
        - **Mitigation:** Focus on delivering a high-quality, intuitive, and helpful user experience. Collect and act on user feedback.
    - **Accuracy of Responses:** The chatbot may provide incorrect or irrelevant responses, leading to user frustration.
        - **Mitigation:** Continuously update and train the chatbot using real-world interactions and client-specific data.
7. **Regulatory and Compliance Risks:**
    - **Changing Regulations:** Keeping up with changing data protection and industry-specific regulations can be challenging.
        - **Mitigation:** Stay informed about regulatory changes, and ensure the application is adaptable to new requirements.
8. **Technological Changes:**
    - **Rapid Advancements:** The field of AI and machine learning is evolving rapidly, and the technology used may become obsolete.
        - **Mitigation:** Stay updated with technological advancements, invest in ongoing research and development, and be prepared to adopt new technologies.

9. **Integration with Existing Systems**:

- The chatbot needs to be integrated with the existing IT infrastructure. Any compatibility issues could lead to delays and increased costs.
    - **Mitigation:** To mitigate the risk of integration issues with existing systems, a thorough assessment of the current IT infrastructure is crucial. This includes understanding the APIs, databases, CRM systems, and other software that the chatbot will interact with. Using standardized interfaces and protocols can ensure compatibility and ease of integration. Pilot testing can help identify potential issues early on.

10. **Language and Cultural Nuances:**

- The chatbot needs to understand and respond appropriately to different languages and cultural nuances. Misunderstandings could lead to customer dissatisfaction.
    - **Mitigation:** To mitigate the risk of misunderstandings due to language and cultural nuances, the chatbot should be designed to understand and respond appropriately to different languages. This can be achieved through Natural Language Processing (NLP) and Machine Learning (ML) techniques. Additionally, the chatbot should be trained on a diverse dataset that includes various cultural nuances. Regular updates and improvements should be made based on user feedback and analysis of chatbot interactions.

Assumptions & Dependencies - review

Assumptions	**Dependencies**
Clients have a basic understanding of chatbot technology and its benefits.	Availability of skilled developers with expertise in AI, NLP, and full-stack development.
Clients are willing to provide necessary business-specific documents for training the chatbot.	Reliable and scalable cloud infrastructure (e.g., AWS, GCP) for hosting and deployment.
The chatbot will be used in environments with stable internet connectivity.	Third-party integration support (e.g., CRM, inventory systems) via APIs.

Users are comfortable interacting with a chatbot for customer support.	Access to up-to-date and comprehensive NLP libraries and frameworks (e.g., spaCy, TensorFlow).
Clients will provide timely feedback and participate in the onboarding process.	Compliance with data protection regulations (e.g., GDPR, CCPA) by the development and deployment team.
The application will have a sufficient budget and resources allocated for development and maintenance.	Reliable third-party services for additional functionalities (e.g., authentication, analytics).
Clients have the necessary infrastructure to embed the chatbot widget on their websites.	Regular updates and patches from the underlying technologies used in the stack.
The application will receive ongoing support and updates based on user feedback and technological advancements.	Stable and high-speed internet connectivity for optimal chatbot performance.
The chatbot will handle a variety of user queries effectively across different domains.	Adequate training data provided by clients for specific business needs and domains.
Data security and privacy measures will be sufficient to protect sensitive client and user information.	Continuous monitoring and support from cloud service providers to ensure uptime and performance.
Clients will follow the recommended guidelines for integrating and using the chatbot.	Efficient customer support and maintenance teams to handle client issues and feedback.

This table outlines the key assumptions and dependencies that should be considered for the successful development, deployment, and use of the chatbot application.

User Stories - review*Let's work as a team to create these****

List or link to the user stories involved. Also link to customer interviews, and include screenshots of what you've seen. Provide enough detail to make a complete story.

#	Title	User Story	Acceptance Criteria	Priority	Jira Link	Notes
1	Epic HA-1: Assessment Feature:		•			•
2	Epic HA-2:		•			•

	Generative AI Integration					

Cost Factor Estimations for 6 Months

Cost Factor	Estimated Cost Range (6 Months)	Details
Development Costs	$100,000 - $200,000	Includes salaries for a smaller team of full-stack developers, AI/ML engineers, and front-end developers. Costs based on team size and expertise.
Operational Costs	$25,000 - $50,000	Includes cloud hosting (AWS/GCP), database management (MongoDB, Redis), and third-party services (APIs, monitoring tools).
Research and Development (R&D)	$25,000 - $75,000	Budget for initial research, prototype development, and ongoing improvements in AI/NLP capabilities.
Legal and Compliance	$10,000 - $25,000	Costs for ensuring compliance with regulations (GDPR, CCPA), legal consultations, and drafting necessary legal documents and agreements.
Security Measures	$15,000 - $30,000	Investment in data encryption, regular security audits, vulnerability assessments, and compliance with industry standards.
Marketing and Sales	$20,000 - $40,000	Costs for promoting the chatbot solution, sales team salaries, marketing campaigns, and promotional materials.
Customer Support and Maintenance	$15,000 - $35,000	Ongoing customer support, maintenance, and updates to ensure smooth operation and address any client issues or feedback.
Training and Documentation	$5,000 - $10,000	Creating training materials, user guides, and documentation for clients to effectively use and integrate the chatbot.

Infrastructure Setup	$10,000 - $20,000	Initial setup of development environments, CI/CD pipelines, and necessary software tools and licenses.
Testing and Quality Assurance	$10,000 - $20,000	Rigorous testing to ensure the chatbot performs as expected, including unit tests, integration tests, and user acceptance testing.
Contingency Fund	$15,000 - $25,000	Reserved for unforeseen expenses and risks that may arise during development and deployment.

Total Estimated Costs for 6 Months:

- **Initial Development:** $210,000 - $530,000
- **Operational and Maintenance:** $50,000 - $100,000

Breakdown of Estimated Costs for 6 Months:

1. **Development Costs:**
    - **Full-Stack Developers:** 2 developers at $80,000/year each (6 months) = $80,000
    - **AI/ML Engineers:** 1 engineer at $100,000/year (6 months) = $50,000
    - **Front-End Developers:** 1 developer at $70,000/year (6 months) = $35,000
2. **Operational Costs:**
    - **Cloud Hosting (AWS/GCP):** $15,000 (6 months)
    - **Database Management:** $5,000 (6 months)
    - **Third-Party Services:** $5,000 (6 months)
3. **R&D Costs:**
    - **Initial Research:** $25,000
    - **Ongoing Improvements:** $25,000 (6 months)
4. **Legal and Compliance:**
    - **Legal Consultations:** $10,000
    - **Compliance Measures:** $15,000
5. **Security Measures:**
    - **Data Encryption:** $10,000
    - **Security Audits:** $5,000 (6 months)
6. **Marketing and Sales:**
    - **Marketing Campaigns:** $20,000
    - **Sales Team Salaries:** $20,000 (6 months)
7. **Customer Support and Maintenance:**
    - **Support Team Salaries:** $20,000 (6 months)
    - **Maintenance Costs:** $15,000 (6 months)
8. **Training and Documentation:**
    - **Training Materials:** $5,000
    - **User Guides:** $5,000

9. **Infrastructure Setup:**
    - **Development Environments:** $10,000
    - **CI/CD Pipelines:** $10,000
10. **Testing and Quality Assurance:**
    - **Testing Tools:** $5,000
    - **QA Team Salaries:** $15,000 (6 months)
11. **Contingency Fund:**
    - **Reserved Budget:** $20,000

These modest estimates provide a more budget-conscious view for developing, deploying, and maintaining the chatbot application over a six-month period, ensuring all critical aspects are covered while controlling costs.

Potential Revenue Streams for the Chatbot Application

1. **Subscription Fees:**
    - **Description:** Charge clients a recurring monthly or annual fee for using the chatbot service.
    - **Estimated Revenue:** $100 - $1,000 per client per month, depending on the size of the business and the level of service provided.
        - **Example:** With 50 clients, at an average of $500/month, annual revenue = $500 * 50 * 12 = $300,000.
2. **Setup and Integration Fees:**
    - **Description:** Charge a one-time fee for initial setup and integration of the chatbot into the client's website and systems.
    - **Estimated Revenue:** $1,000 - $10,000 per client, depending on the complexity of the integration.
        - **Example:** With 20 new clients annually at an average fee of $5,000, annual revenue = $5,000 * 20 = $100,000.
3. **Customization and Development Fees:**
    - **Description:** Offer additional customization and feature development services tailored to specific client needs.
    - **Estimated Revenue:** $5,000 - $50,000 per project, depending on the scope and complexity.
        - **Example:** With 10 customization projects annually at an average fee of $20,000, annual revenue = $20,000 * 10 = $200,000.
4. **Usage-Based Fees:**
    - **Description:** Charge clients based on the number of interactions or messages handled by the chatbot.
    - **Estimated Revenue:** $0.01 - $0.10 per interaction, depending on the volume.

        - **Example:** With 1,000,000 interactions per month at an average fee of $0.05, annual revenue = $0.05 * 1,000,000 * 12 = $600,000.
5. **Premium Support Packages:**
    - **Description:** Offer premium support services for faster response times, dedicated account managers, and additional support options.
    - **Estimated Revenue:** $500 - $5,000 per month per client, depending on the level of support.
        - **Example:** With 10 clients opting for premium support at an average of $2,000/month, annual revenue = $2,000 * 10 * 12 = $240,000.
6. **Data Analytics and Reporting:**
    - **Description:** Provide advanced analytics and reporting features as an add-on service to help clients gain insights from chatbot interactions.
    - **Estimated Revenue:** $100 - $1,000 per month per client for advanced analytics.
        - **Example:** With 30 clients using advanced analytics at an average of $500/month, annual revenue = $500 * 30 * 12 = $180,000.
7. **Advertising and Affiliate Marketing:**
    - **Description:** Partner with relevant third-party services and display targeted ads or affiliate links within the chatbot.
    - **Estimated Revenue:** $0.10 - $1.00 per click or conversion, depending on the partner agreement.
        - **Example:** With 10,000 clicks per month at an average revenue of $0.50 per click, annual revenue = $0.50 * 10,000 * 12 = $60,000.
8. **Training and Certification Programs:**
    - **Description:** Offer training sessions and certification programs for clients' staff to effectively use and manage the chatbot.
    - **Estimated Revenue:** $500 - $5,000 per training session, depending on the depth and duration.
        - **Example:** With 20 training sessions annually at an average fee of $2,000, annual revenue = $2,000 * 20 = $40,000.

Total Potential Annual Revenue:

- **Low Estimate:** $1,220,000
- **High Estimate:** $1,720,000

By diversifying revenue streams, the chatbot application can ensure a stable and growing income, catering to various client needs and maximizing the potential for profitability.

User Engagement Strategies

1. **Personalized Interactions:**
    - **Description:** Tailor responses and suggestions based on user behavior, preferences, and past interactions.
    - **Implementation:** Use AI and machine learning to analyze user data and personalize the chatbot's responses accordingly.
2. **Interactive Onboarding:**
    - **Description:** Provide a guided onboarding process to help users understand the chatbot's capabilities and features.
    - **Implementation:** Design an engaging, step-by-step tutorial that highlights key functionalities and how to interact with the chatbot.
3. **Gamification:**
    - **Description:** Incorporate gamified elements such as badges, rewards, and points for frequent interactions or achieving specific milestones.
    - **Implementation:** Develop a reward system where users earn points or badges for using the chatbot regularly or completing certain actions.
4. **Regular Updates and New Features:**
    - **Description:** Continuously introduce new features, updates, and improvements to keep the chatbot experience fresh and engaging.
    - **Implementation:** Implement a feedback loop with users to gather suggestions and prioritize feature development based on user needs.
5. **Multichannel Support:**
    - **Description:** Enable users to interact with the chatbot across various platforms such as web, mobile apps, social media, and messaging apps.
    - **Implementation:** Integrate the chatbot with popular platforms like Facebook Messenger, WhatsApp, and Slack to provide seamless multichannel support.
6. **Proactive Engagement:**
    - **Description:** The chatbot should initiate conversations based on user behavior, such as offering assistance when a user is stuck or suggesting relevant information.
    - **Implementation:** Use behavioral analytics to identify key moments for proactive engagement and set triggers for the chatbot to reach out to users.
7. **Contextual Awareness:**
    - **Description:** Ensure the chatbot maintains context throughout the conversation to provide relevant and coherent responses.
    - **Implementation:** Use advanced NLP techniques and context management algorithms to track and understand the conversation flow.
8. **Feedback Mechanism:**
    - **Description:** Encourage users to provide feedback on their experience and the chatbot's performance.

- **Implementation:** Integrate feedback prompts within the chatbot and use the collected data to make continuous improvements.

9. **Educational Content and Resources:**
    - **Description:** Offer educational content, FAQs, and resources that users can access to learn more about the chatbot and the services it provides.
    - **Implementation:** Create a knowledge base and integrate it into the chatbot's interface, allowing users to easily find and access helpful information.
10. **Personalized Follow-Ups:**
    - **Description:** Send personalized follow-up messages based on previous interactions to re-engage users and provide additional value.
    - **Implementation:** Use CRM integration to track user interactions and schedule automated follow-up messages with relevant content or offers.
11. **Live Agent Handoff:**
    - **Description:** Provide a seamless transition from the chatbot to a live human agent when necessary, ensuring complex queries are handled effectively.
    - **Implementation:** Implement a smart escalation system that transfers the conversation context to human agents without losing information.
12. **Visual and Interactive Elements:**
    - **Description:** Enhance the user experience with visual aids, interactive buttons, quick replies, and multimedia content.
    - **Implementation:** Incorporate rich media elements such as images, videos, and interactive carousels to make interactions more engaging.
13. **Localized Content:**
    - **Description:** Offer localized content and support multiple languages to cater to a global audience.
    - **Implementation:** Implement language detection and translation capabilities to provide a personalized experience for users from different regions.
14. **Incentives and Promotions:**
    - **Description:** Provide special offers, discounts, and promotions to users through the chatbot to incentivize engagement.
    - **Implementation:** Collaborate with marketing teams to create targeted campaigns and deliver them through the chatbot.
15. **Regular Communication:**
    - **Description:** Keep users informed about updates, new features, and important announcements through regular communication.
    - **Implementation:** Use email notifications, in-app messages, and chatbot notifications to keep users engaged and informed.

By implementing these user engagement strategies, the chatbot application can foster a more interactive, personalized, and satisfying user experience, leading to higher user retention and satisfaction.

UX Mocks

https://www.figma.com/file/SqpAkkjkUYyexSRaLbzZku/hireworks?type=design&node-id=0-1&mode=design

Miro Firewrames: https://miro.com/app/board/uXjVKUlrgRg=/

Key Milestones (Release Plan)

Release Plan (7 Weeks Remaining)

Key Milestone	**Estimated Start Date**	**Estimated End Date**
Requirement Gathering	May 20, 2024	May 26, 2024
Architecture Design	May 27, 2024	June 2, 2024
Prototype Development	June 3, 2024	June 9, 2024
NLP and AI Model Training	June 3, 2024	June 16, 2024
Backend Development	May 27, 2024	June 9, 2024
Frontend Development	June 3, 2024	June 16, 2024
Integration and API Development	June 10, 2024	June 16, 2024
Internal Testing and QA	June 17, 2024	June 23, 2024
Beta Release and Client Feedback	June 24, 2024	June 30, 2024
Feedback Analysis and Iteration	June 24, 2024	June 30, 2024
Final Testing and QA	July 1, 2024	July 3, 2024
Documentation and Training	July 1, 2024	July 3, 2024
Marketing and Launch Preparation	July 1, 2024	July 3, 2024
Official Launch	July 4, 2024	July 4, 2024
Post-Launch Support and Updates	July 5, 2024	Ongoing

- **Requirement Gathering (May 20 - May 26):** Collect detailed requirements from stakeholders.
- **Architecture Design (May 27 - June 2):** Design the system architecture and technical specifications.

- **Prototype Development (June 3 - June 9):** Develop a basic prototype to demonstrate key functionalities.
- **NLP and AI Model Training (June 3 - June 16):** Train NLP models using client-specific documents and data.
- **Backend Development (May 27 - June 9):** Develop the server-side components and APIs.
- **Frontend Development (June 3 - June 16):** Create the user interface and chatbot widget.
- **Integration and API Development (June 10 - June 16):** Integrate various components and develop necessary APIs.
- **Internal Testing and QA (June 17 - June 23):** Conduct thorough testing to identify and fix bugs.
- **Beta Release and Client Feedback (June 24 - June 30):** Release a beta version to select clients for feedback.
- **Feedback Analysis and Iteration (June 24 - June 30):** Analyze client feedback and make necessary improvements.
- **Final Testing and QA (July 1 - July 3):** Conduct final testing to ensure readiness for launch.
- **Documentation and Training (July 1 - July 3):** Prepare user guides, training materials, and documentation.
- **Marketing and Launch Preparation (July 1 - July 3):** Plan and execute marketing strategies for the launch.
- **Official Launch (July 4):** Release the chatbot application to the public.
- **Post-Launch Support and Updates (July 5 - Ongoing):** Provide ongoing support and release updates based on user feedback.

Questions

Questions for Developers and Expected Outcomes

Question	Expected Outcome
What technologies and frameworks do you recommend for the front-end?	Identification of suitable front-end technologies (e.g., React.js) and frameworks for the project.
How will you ensure the chatbot can handle multiple specialized agents?	Explanation of the approach and architecture for managing multiple specialized agents within the chatbot.
What methods will you use to train the NLP models with client-specific data?	Understanding of the techniques and tools (e.g., TensorFlow, spaCy) for training NLP models using client data.

How will you integrate the chatbot with existing client systems?	Detailed plan for integrating the chatbot with CRM, inventory management, and other client systems through APIs.
What security measures will you implement to protect user data?	Description of security protocols (e.g., encryption, authentication) to ensure data protection and compliance.
How will you handle scalability to accommodate increasing user traffic?	Strategy for scaling the application, including cloud infrastructure and load balancing.
Can you outline the testing process to ensure the chatbot's reliability?	Overview of the testing procedures (e.g., unit testing, integration testing, QA) to ensure the chatbot functions properly.
How will you manage and store the unstructured data generated by the chatbot?	Explanation of the database solutions (e.g., MongoDB, Redis) for handling and storing unstructured chatbot data.
What is your approach to maintaining contextual awareness in conversations?	Understanding of techniques for maintaining conversation context using advanced NLP and context management algorithms.
How will you handle updates and maintenance post-launch?	Plan for ongoing updates, bug fixes, and maintenance to ensure the chatbot remains functional and up-to-date.
What analytics tools will you integrate for performance monitoring?	List of analytics and monitoring tools (e.g., ELK Stack, Prometheus, Grafana) for tracking performance and user interactions.
How will you ensure the chatbot is accessible to users with disabilities?	Implementation of accessibility features (e.g., compliance with WCAG standards) to ensure inclusivity.
What is your plan for user feedback integration?	Process for collecting, analyzing, and incorporating user feedback into future updates and improvements.
How will you support multilingual capabilities in the chatbot?	Plan for implementing language detection and translation to support multiple languages.
Can you provide a timeline for each development phase?	Detailed timeline outlining the expected duration and milestones for each phase of development.

This table provides a structured approach to understanding the development process, ensuring key aspects are addressed and expectations are clear.

Out of Scope

Features That Could Be Out of Scope

Feature	**Reason for Being Out of Scope**
Voice Interaction Capabilities	Adding voice interaction would significantly increase complexity and development time.
Virtual Reality (VR) Integration	VR integration is not essential for a customer support chatbot and requires substantial resources.
Augmented Reality (AR) Features	Similar to VR, AR is complex and not critical for the primary functionality of the chatbot.
Blockchain for Data Security	Implementing blockchain is advanced and may not be necessary for the security requirements.
Advanced Sentiment Analysis	Basic sentiment analysis may suffice; advanced features could be added later based on need.
Hyper-Personalization Using Big Data	Requires extensive data processing capabilities and may not be feasible initially.
Integration with All Possible Third-Party Apps	Focus on key integrations first; extensive integration can be added based on client demand.
Custom Animated Avatars	Developing custom avatars can be resource-intensive and is not essential for core chatbot functions.
Social Media Monitoring and Engagement	Advanced social media features can be complex and may extend beyond primary customer support needs.
Predictive Analytics for Business Insights	Requires significant data and complex algorithms, which may be out of scope for initial development.
Natural Language Generation (NLG)	Advanced NLG capabilities could be deferred to future updates to ensure initial release stability.
In-App Purchase Handling	Complex financial transactions could be deferred to focus on core support functions initially.
Comprehensive User Behavior Analytics	Basic analytics should suffice initially; comprehensive tracking can be built over time.
Support for unusual Languages	Focus on widely-used languages first; rare language support can be added based on client demand.
Real-Time Collaboration Tools	Adding features for real-time collaboration among users may complicate the initial development.

This list helps in focusing the initial development on core functionalities and ensuring timely delivery while considering these advanced features for future iterations based on user feedback and demand.

Persona
https://docs.google.com/presentation/d/1IItRXmeXMtSxK7S1LiK_DQT-m2N7XN50nLpinnkWFoI/edit
https://docs.google.com/presentation/d/1NtACExwgxqhb5Dk74LC07tX_g8OA4c6fEhg6Y98BMZ8/edit