AI-Based Need Analysis Tool

SRS Document

17-03-2025

Version 1.0

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Effective Date | Brief Description of Change | Prepared By | Approved By |
| 1.0 | 17-03-2025 | Initial Document | Ankit Verma |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Contents

[1 Introduction 5](#_Toc193275394)

[1.1 Purpose 5](#_Toc193275395)

[1.2 Business Objective 5](#_Toc193275396)

[1.3 Audience 5](#_Toc193275397)

[1.4 Abbreviations / Acronyms 5](#_Toc193275398)

[2 Scope 6](#_Toc193275399)

[2.1 In-Scope 6](#_Toc193275400)

[2.2 Out-of-Scope 6](#_Toc193275401)

[3 Assumptions & Dependencies 6](#_Toc193275402)

[Assumptions 6](#_Toc193275403)

[Dependencies 7](#_Toc193275404)

[4 Process Flow Diagram & Business Process 7](#_Toc193275405)

[4.1 General Flow 7](#_Toc193275406)

[5 Activity Flows 9](#_Toc193275407)

[5.1 User Intimation, Greeting & Initial Engagement 9](#_Toc193275408)

[5.2 Data Collection 11](#_Toc193275409)

[5.2.1 User Provides Incomplete or Conflicting Details 14](#_Toc193275410)

[5.2.2 User Skips Important Questions 15](#_Toc193275411)

[5.2.3 User Enters Free-Form Queries Instead of Selecting Options 15](#_Toc193275412)

[5.3 AI-Powered Need Analysis & Plan Matching 18](#_Toc193275413)

[5.3.1 User Wants More Customization 21](#_Toc193275414)

[5.3.2 User Wants to Compare Plans 22](#_Toc193275415)

[5.4 Plan Recommendation & Premium Estimation 23](#_Toc193275416)

[5.4.1 User Requests a Premium Estimation 24](#_Toc193275417)

[5.4.2 User Wants a Tax-Saving Plan 25](#_Toc193275418)

[5.5 Finalizing the Decision & Policy Issuance 25](#_Toc193275419)

[5.5.1 User Requests Advisory Support Before Purchase 27](#_Toc193275420)

[5.5.2 User Exits Midway & Wants to Resume Later 27](#_Toc193275421)

[5.6 Session Tracking, Retrieval & User Data Management 29](#_Toc193275422)

[5.6.1 User Selects "Saved Analysis" 31](#_Toc193275423)

[5.6.2 User Selects "My Profile" 32](#_Toc193275424)

[5.7Knowledge Base Search for Policy & FAQs (RAG-Based System) 33](#_Toc193275425)

[5.7.1 User Asks About Policy Payout Without Buying a Policy 35](#_Toc193275426)

[5.7.2 User Wants Insurance for Business Loan Collateral 35](#_Toc193275427)

[5.8 Input Validation & Data Consistency Checks 36](#_Toc193275428)

[6.8.1 User Provides Incomplete or Conflicting Details 38](#_Toc193275429)

[6.8.2 User Exaggerates Income or Coverage Needs 38](#_Toc193275430)

[6.8.3 User Skips Key Financial Questions 38](#_Toc193275431)

[5.9 Advanced User Scenarios & Edge Cases 39](#_Toc193275432)

[5.9.1User Exits Mid-Process Without Completing Any Steps 39](#_Toc193275433)

[5.9.2User Keeps Changing Answers 40](#_Toc193275434)

[5.9.3User's Internet Connection Is Lost Midway 41](#_Toc193275435)

[5.9.4User Requests Multiple Customizations 41](#_Toc193275436)

[5.9.5User Requests a Plan Comparison After Multiple Customizations 42](#_Toc193275437)

[5.10 Coverage Comparison & Plan Selection 43](#_Toc193275438)

[5.10.1 User Wants More Detailed Information on a Specific Plan 45](#_Toc193275439)

[5.10.2 User Adjusts Comparison Criteria 46](#_Toc193275440)

[5.10.3 User Saves Comparison Results for Future Review 46](#_Toc193275441)

[5.10.4 User Wants to Speak with an Advisor Before Finalizing 46](#_Toc193275442)

[5.10.5 User Profile & Dashboard Overview 47](#_Toc193275443)

[5.11 Policy Issuance & Follow-Up 49](#_Toc193275444)

[5.11.a User Proceeds with Purchase 51](#_Toc193275445)

[5.11.b User Needs More Time to Decide 51](#_Toc193275446)

[5.11.1 User Requests Advisory Support Before Purchase 52](#_Toc193275447)

[5.12Advanced Functionalities & AI Improvements 53](#_Toc193275448)

[5.12.1Premium Estimator & Affordability Check 53](#_Toc193275449)

[5.12.2 Automated Session Tracking & Retrieval 54](#_Toc193275450)

[5.12.3 Tax Planning & Life Insurance Optimization 55](#_Toc193275451)

[5.12.4. AI Model Learning & Continuous Improvement 56](#_Toc193275452)

[6 Integration 57](#_Toc193275453)

[7 Impact on User Management Module 58](#_Toc193275454)

[8 Appendix A - Wire Frames & Field Appendix 59](#_Toc193275455)

[8.1 WF\_Webpage\_001 59](#_Toc193275456)

[8.2 WF\_Chatbot\_Interface\_Greeting and Benefit\_offering\_002 61](#_Toc193275457)

[8.3 WF\_Chatbot\_Interface\_NeedCalculation\_003 64](#_Toc193275458)

[8.4 WF\_Chatbot\_Interface\_PlanRecommendation\_and \_CoverageComparsion\_004 67](#_Toc193275459)

[8.5 WF\_Chatbot\_Interface\_SavedAnalysis \_005 68](#_Toc193275460)

[8.6 WF\_User Profile Creation\_006 72](#_Toc193275461)

[8.7 Templet for gap report generated 74](#_Toc193275462)

[9. Appendix B – Requirements 77](#_Toc193275463)

[9.1Functional Requirement 77](#_Toc193275464)

[9.1.1User Interaction & Engagement Functionalities in LINAT 81](#_Toc193275465)

[9.1.2 Data Collection & User Profile Management Functionalities in LINAT 84](#_Toc193275466)

[9.1.3AI-Powered Need Analysis & Plan Matching in LINAT 88](#_Toc193275467)

[9.1.4Advanced Calculations & Estimations in LINAT 91](#_Toc193275468)

[9.1.5 Policy Issuance & User Decision Finalization 95](#_Toc193275469)

[9.1.6 Knowledge Base – Comprehensive Insurance Information System 99](#_Toc193275470)

[9.1.7 User Profile & Dashboard Functionalities 101](#_Toc193275471)

[9.1.8Security & Compliance 104](#_Toc193275472)

[9.1.9 AI Learning & System Updates 106](#_Toc193275473)

[9.1.10 Handling Advanced User Scenarios & Edge Cases 108](#_Toc193275474)

[10.2 Non-Functional Requirements 110](#_Toc193275475)

[10.2.1Performance 110](#_Toc193275476)

[10.2.2 Reliability 111](#_Toc193275477)

[10.2.3 Security 111](#_Toc193275478)

[10.2.4 Usability 111](#_Toc193275479)

[10.2.5 Maintainability 111](#_Toc193275480)

[10.2.6 Compatibility 112](#_Toc193275481)

[10.2.7 Audit and Monitoring 112](#_Toc193275482)

[11 Technical Requirement 113](#_Toc193275483)

[12 Reference Documents 117](#_Toc193275484)

[13 Open Items 117](#_Toc193275485)

[14 Sign Off 118](#_Toc193275486)

# Introduction

## 1.1 Purpose

The **Life Insurance Need Analysis Tool (LINAT)** is an AI-driven chatbot designed to assist individuals in evaluating their life insurance needs. Its primary goal is to simplify insurance selection by providing personalized recommendations based on user inputs, financial goals, and risk profiles. By leveraging AI, the system ensures accuracy and efficiency in policy recommendations while enhancing user understanding of insurance concepts.

LINAT enables users to make informed decisions by offering structured guidance and clarifying technical terms to minimize confusion. It also allows users to compare policies and store data for future reference. While the system focuses solely on need analysis, it provides a seamless transition to the policy issuance process when users decide to purchase a plan.

## 1.2 Business Objective

The client requires an AI-powered life insurance need analysis system that can:

* Analyze user financial goals and risk factors.
* Recommend the most suitable life insurance policies.
* Compare insurance plans across different insurers.
* Estimate premiums and provide cost projections.
* Seamlessly route users to policy issuance.
* Enhance customer engagement and improve policy conversions.

## 1.3 Audienceex

This document is intended for:

* + Insurance Company
  + iNube Internal Teams

## 1.4 Abbreviations / Acronyms

|  |  |
| --- | --- |
| Acronym | Definition |
| AI | Artificial Intelligence |
| NLP | Natural Language Processing |
| LINAT | Life Insurance Need Analysis Tool |
| API | Application Programming Interface |
| GDPR | General Data Protection Regulation |
| HIPAA | Health Insurance Portability and Accountability Act |
| GAPP | Generally Accepted Privacy Principles |

# Scope

## 2.1 In-Scope

* AI-based need analysis for life insurance.
* Chatbot-driven user interaction.
* Premium and benefits calculation tools.
* Policy comparison features.
* User-friendly interface with guided assistance.

## 2.2 Out-of-Scope

* Anything other than in scope is out of scope

# Assumptions & Dependencies

## 3.1 Assumptions

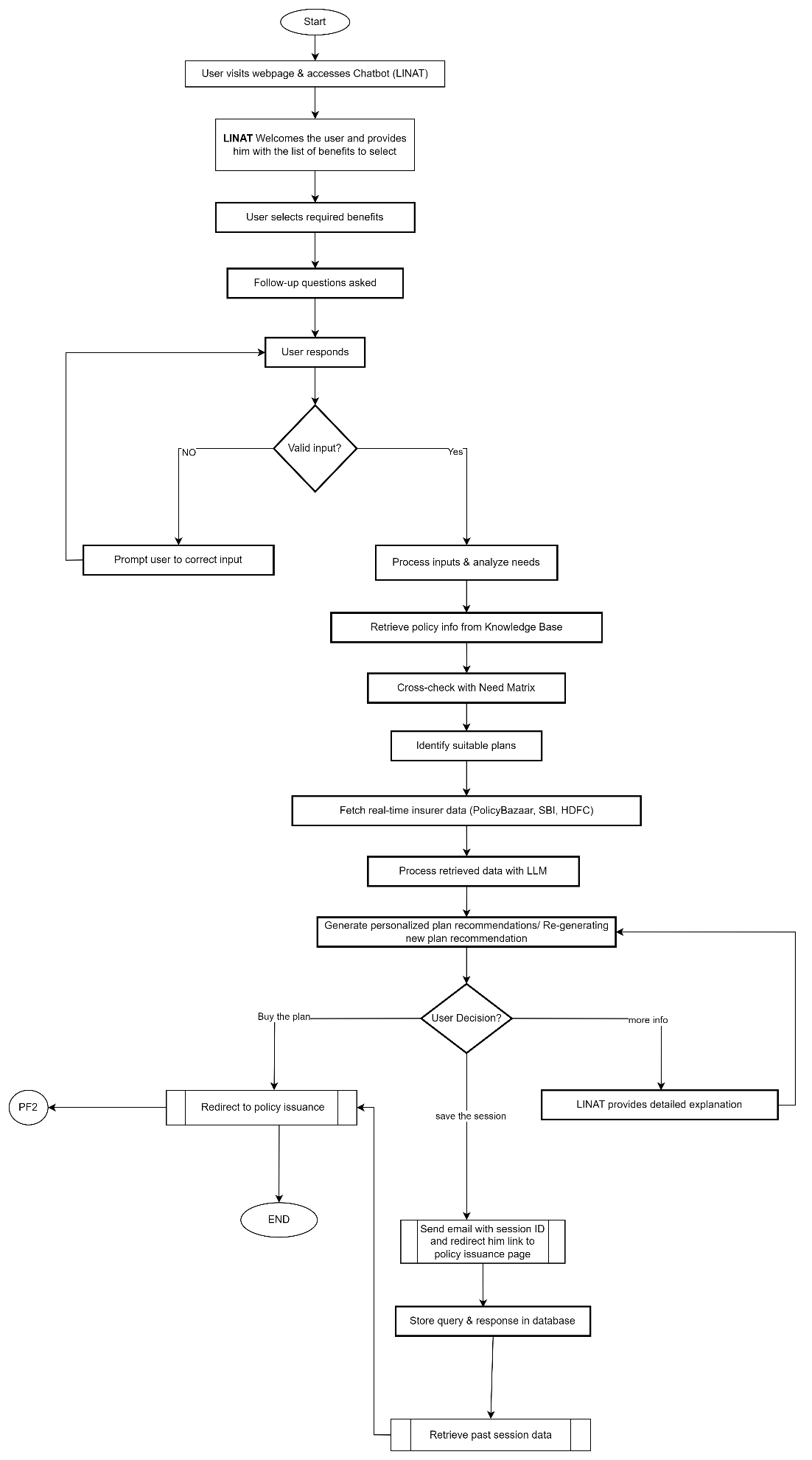
* The system currently only supports life insurance (Term, ULIP, Endowment, Whole Life, Money-Back, Pension Plans).
* The chatbot currently supports **English only**.
* The knowledge base is auto-manual updated with the latest insurance plans, IRDAI regulations, and insurer offerings.
* Users do not need to create an account to start the need analysis process, but registration is required to save sessions.
* AI models will be continuously trained on user interactions to improve accuracy and relevance.
* LINAT does not issue policies directly but redirects users to the insurer's platform for final purchase.

## 3.2 Dependencies

* Integration with third-party APIs for insurance data.
* Availability of the Open AI and Chat-GPT APIs for natural language processing.
* Access to a secure database for storing user data and interaction history.

# Process Flow Diagram & Business Process

## 4.1 General Flow



# Activity Flows

NOTE: - These are rule-based scenarios that we need to address during the training and development of the system.

## 5.1 User Intimation, Greeting & Initial Engagement

**Process Description:**

* The chatbot (LINAT) initiates the interaction with a structured greeting, introduces its capabilities, and provides users with guided options for selecting the appropriate insurance product based on their financial goals.

**Actors:**

* **User** – A customer seeking life insurance.
* **LINAT (Chatbot)** – AI-powered virtual assistant that analyzes insurance needs.

**Objective:**

* To engage the user effectively and guide them towards selecting the appropriate next step in the insurance decision-making process.

**Preconditions:**

* The user accesses LINAT through a website or integrated platform.

**Postconditions:**

* The chatbot provides users with multiple options to proceed, including:
* **Analyze My Insurance Needs** – Initiates the need analysis process.

**Process Flow:**

* The user visits LINAT through an available digital interface.
* The chatbot initiates the conversation by greeting the user and introducing its functionalities.
* The chatbot presents the available interaction options.
* The user selects an option to proceed.
* LINAT processes the selection and directs the user accordingly.

**Example Chatbot Interaction:**

* **Chatbot:**

**Welcome to [Insurance Company].**

I am here to help you find the best life insurance policy that meets your financial goals.

**What are you primarily looking for?**

* **Wealth Creation** – Grow your money over time
* **Child’s Fund** – Secure finances for education, marriage, or wealth transfer
* **Retirement Planning** – Ensure a steady income after retirement
* **Health Coverage** – Protection against medical expenses
* **Other** –

**Death Benefit** – Financial protection for dependents in case of the policyholder’s demise.

**Maturity Benefit** – Lump sum payout at the end of the policy term if the insured survives.

**Survival Benefit** – Periodic payouts during the policy tenure (common in money-back plans).

**Income Replacement** – Monthly or annual income for dependents in case of the policyholder’s death or disability.

**Loan Protection** – Ensures outstanding loans are covered in case of death or disability.

**Wealth Creation** – Investment-linked benefits offering market-based returns (ULIPs, Endowment).

**Retirement Income (Pension/Annuity)** – Guaranteed regular income post-retirement.

**Child Education Fund** – Ensures education expenses are covered even if the policyholder is no longer present.

**Marriage Fund for Children** – Financial support for children’s marriage expenses.

**Critical Illness Coverage** – Lump sum payout upon diagnosis of life-threatening illnesses.

**Accidental Death Benefit** – Additional sum assured payout in case of accidental death.

**Disability Protection** – Financial support in case of permanent or temporary disability.

**Waiver of Premium** – Future premiums are waived in case of disability or critical illness.

**Tax Benefits** – Tax savings under Sections 80C, 80D, and 10(10D) of the Income Tax Act.

**Estate Planning** – Ensures smooth inheritance and wealth transfer to legal heirs.

**Asset Protection** – Safeguarding personal/business assets against financial risks.

**Liquidity Benefit** – Allows partial withdrawals from certain policies for urgent financial needs.

**System Interactions and Components:**

**Frontend Components:**

* **Chat Interface:** Displays structured greeting messages and available options.
* **User Input Processor:** Captures and processes user selections dynamically.

**Backend Components:**

* **Session Management Module:** Retrieves past user interactions and stored sessions.
* **Need Analysis Engine:** Initiates financial assessment based on user responses.
* **Comparison Module:** Fetches insurance products and generates comparative reports.
* **Advisor Support System:** Facilitates advisor connection when required.

**Future Enhancements:**

* Personalize greeting messages based on insurance company requirements and past interactions.
* Integration of voice-based chatbot functionalities for hands-free engagement.
* Expansion of multi-language support to accommodate a diverse user base

## 5.2 Data Collection

**Process Description:**

To recommend the most suitable life insurance plan, the chatbot collects structured information regarding the user’s financial situation, risk appetite, and coverage needs. This ensures that the AI-powered need analysis and plan matching are accurate and personalized.

**Actors:**

1. **User –** A customer seeking insurance, providing details for analysis.
2. **LINAT (Chatbot) –** AI-powered chatbot collecting and processing user data.
3. **Database/Knowledge Base –** Stores and retrieves user inputs for further analysis.

**Objective:**

* Gather financial details to determine the best insurance plan for the user.
* Ensure AI-driven recommendations are personalized and data-driven.
* Validate user inputs and resolve inconsistencies before proceeding to plan recommendations.

**Preconditions:**

* The chatbot is active and ready to collect, validate, and store the data.

**Postconditions:**

* The chatbot successfully collects and stores structured user information.
* Data is processed for need-based analysis in the next step.
* If the user provides conflicting or incomplete details, the chatbot prompts them for corrections.

**Process Flow:**

**Step 1: Chatbot Initiates Data Collection**

* **Chatbot:** "To recommend the best life insurance plan for you, I need some details. Let's start with your age."
* **User Inputs Age:** "35 years"
* **Chatbot:** "Great! Are you married?"
* **User Selects: "Yes"**

**Step 2: Collecting Financial Information**

* **Chatbot:** "What is your annual income?"
* **User Inputs: "₹15 lakh"**
* **Chatbot:** "What are your average monthly expenses?"
* **User Inputs: "₹50,000"**

**Step 3: Identifying Liabilities & Existing Coverage**

* **Chatbot:** "Do you have any existing loans or financial liabilities?"
* **User Selects:** "Yes, ₹20 lakh home loan and ₹5 lakh personal loan"
* **Chatbot:** "Do you already have any life insurance coverage?"
* **User Inputs:** "Yes, a ₹10 lakh term plan from LIC."

**Step 4: Identifying Dependents & Financial Responsibilities**

* **Chatbot:** "How many people depend on you financially?"
* **User Inputs:** "2 (Spouse and 1 Child)"
* **Chatbot: "Are you planning for your child's future education or any other financial goals?"**
* **User Selects:** "Yes, saving for my child's higher education."

**Step 5: Data Validation & Confirmation**

* **Chatbot:** "Thanks for providing the details! Based on your responses, you have an annual income of ₹15 lakh, monthly expenses of ₹50,000, outstanding liabilities of ₹25 lakh, and a ₹10 lakh term plan. Does this look correct?"
* **User** Confirms or Requests Modifications.

**Diagram – Data Collection Process**



**System Interactions and Components:**

**Frontend Components:**

* User Input Forms: Capture structured financial data such as income, expenses, and liabilities.
* Interactive Chatbot UI: Guides users through the data collection process in a conversational manner.

**Backend Components:**

* Data Validation Module: Checks for missing or inconsistent user responses.
* User Profile Database: Stores collected information for future reference.
* Need Analysis Engine: Processes the collected data for further insurance recommendations.

**Error Handling & Edge Cases:**

1. **User Enters Incomplete or Conflicting Information**
   * Chatbot detects inconsistencies and prompts the user for clarification.
   * Example: If a user inputs ₹15 lakh income but ₹20 lakh monthly expenses, the chatbot flags an issue.
2. **User Declines to Provide Certain Details**
   * The chatbot provides estimated defaults and allows users to proceed.
   * Example: "Based on general financial benchmarks, we assume ₹5 lakh annual savings. Would you like to continue?"

### 5.2.1 User Provides Incomplete or Conflicting Details

**Scenario:**

* The user enters contradictory financial details, such as claiming high savings while also mentioning financial instability.
* **Example:** 
  + **User Input:** "I have ₹50 lakh in savings but also struggle with financial security."

**Chatbot Action:**

1. **Detects inconsistency in financial data.**
2. **Prompts the user for clarification:** 
   * **Chatbot:** "I noticed that you mentioned high savings but also financial struggles. Could you clarify if your savings are accessible for emergencies, or are they locked in investments?"
3. **User Provides Clarification:** 
   * **User:** "My savings are in a retirement account and cannot be used for daily expenses."
4. Chatbot updates the analysis by factoring in restricted savings as non-liquid assets.
5. Adjust the recommended insurance coverage accordingly.

### 5.2.2 User Skips Important Questions

**Scenario:**

* The user does not provide key financial details such as income, dependents, or liabilities.
* Example:
  + The user skips the "Annual Income" question.

**Chatbot Action:**

1. **Provides an estimated default based on user profile.**
2. **Chatbot**: "To calculate the right insurance coverage, I need an estimate of your annual income. If you're unsure, I can assume an average of ₹8 lakh based on similar users. Would you like to proceed with this estimate or enter your actual income?"
3. **User Options:** 
   * "Proceed with ₹8 lakh estimate."
   * "Enter my actual income."
4. **User Selects Actual Income:** 
   * **User Input:** "₹12 lakh"
5. Chatbot updates the analysis with the user’s actual income.

### 5.2.3 User Enters Free-Form Queries Instead of Selecting Options

**Scenario:**

* Instead of selecting predefined options, the user types a custom query, which might be vague or unrelated to structured responses.

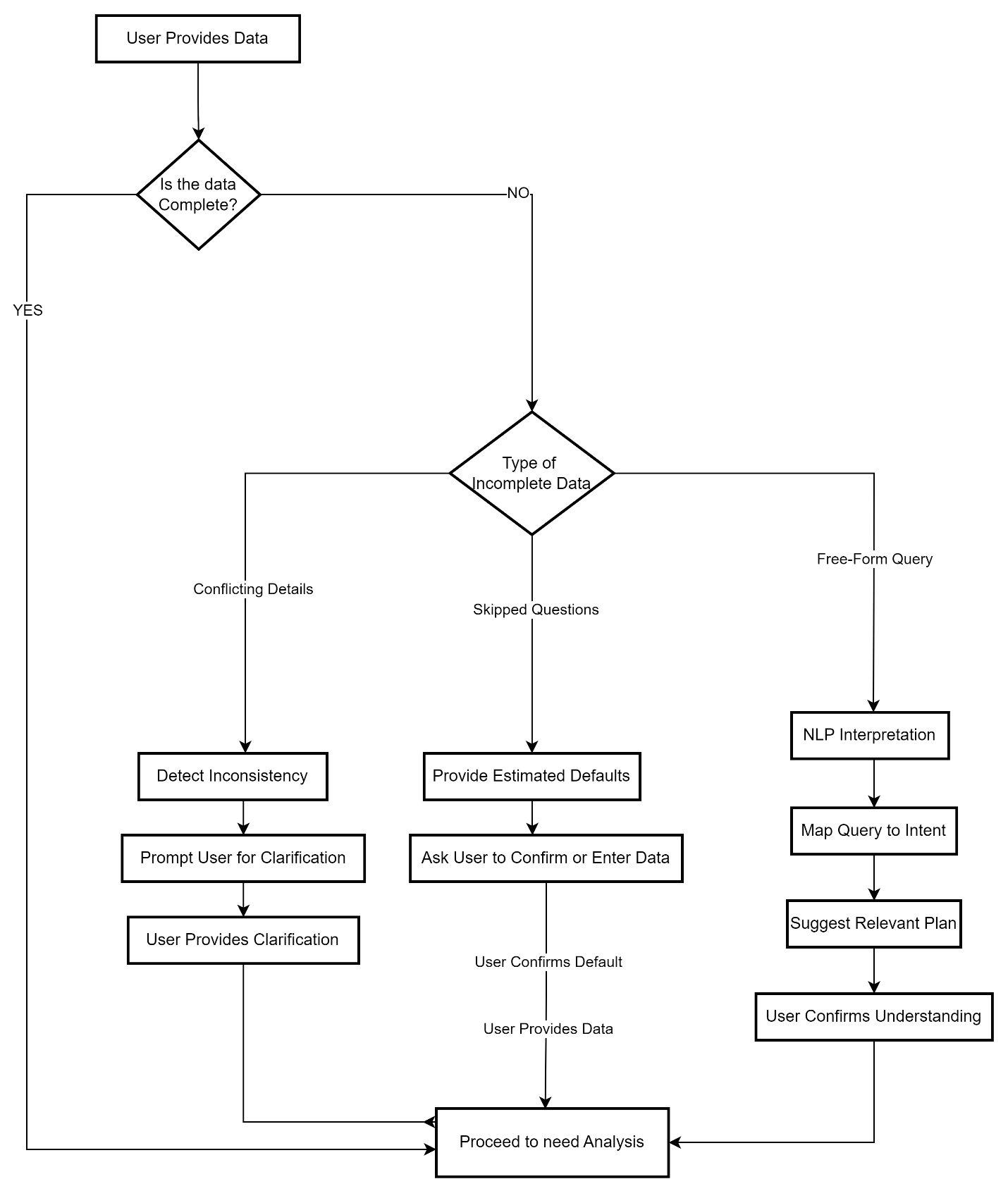
**Example:**

* + **User Input:** "I just want to secure my future without spending too much."

**Chatbot Action:**

1. **Uses Natural Language Processing (NLP) to interpret intent.**
2. **Chatbot:** "It looks like you're looking for a budget-friendly life insurance plan. Would you prefer a term insurance plan, which offers high coverage for a lower premium?"
3. **User Confirms Understanding:** 
   * **User: "Yes, that sounds good."**
4. **Chatbot presents the best term insurance options based on user profile and affordability.**

**Process Flow Diagram – Handling Edge Cases in Data Collection**



**System Interactions & Components Handling Sub-Use Cases**

**Frontend Components:**

* **Chatbot UI:** Dynamic prompts for inconsistent, missing, or free-form user responses.
* **User Input Validation:** Ensures logical consistency in financial data.

**Backend Components:**

* Data Validation Module: Checks for conflicts in user responses and prompts clarifications.
* NLP Engine: Interprets user queries and maps them to predefined insurance options.
* User Need Matrix: Adjusts the insurance recommendation based on updated user inputs.

**Error Handling & System Enhancements**

1. Handling Repeated User Edits: If a user continuously changes financial details, the chatbot provides a comparative analysis for better decision-making.
2. Advanced NLP for Free-Form Queries: Improves accuracy in interpreting custom user inputs.
3. Behavioral Analytics: Tracks user response patterns to improve chatbot learning for future interactions.

## 5.3 AI-Powered Need Analysis & Plan Matching

**Objective:**

The chatbot processes the user's financial data using AI-driven analysis to determine the optimal life insurance coverage amount. Based on income, liabilities, dependents, and financial goals, the system suggests suitable policies that align with the user's financial situation and risk tolerance.

**System Components Involved:**

1. **User Interaction Layer** – The chatbot collects financial data from the user.
2. **AI Processing & Need Analysis Engine** – AI algorithms process user inputs to calculate the required coverage.
3. **Recommendation Engine** – Matches user profiles with suitable insurance products.
4. **Insurance Database & API Integration** – Retrieves real-time insurance plan details.

**Process Triggers:**

* The user completes the financial information input.
* The AI-driven analysis is initiated.
* The system generates suitable plan recommendations based on affordability and user needs.

**Process Steps:**

**Step 1: Chatbot Analyzes Financial Data**

* Chatbot: *"To provide the best life insurance recommendation, I need to analyze your financial details."*
* User provides the following information:
  + **Annual Income:** ₹10 lakh
  + **Existing Insurance Coverage:** ₹5 lakh
  + **Monthly Expenses:** ₹40,000
  + **Liabilities (Loans, Debts):** ₹20 lakh
  + **Number of Dependents:** 2 (Spouse & 1 Child)

**Step 2: AI Determines the User’s Primary Insurance Goal**

* Chatbot: *"Based on your financial inputs, what is your primary goal for buying life insurance?"*
* User selects one or more of the following:
  + **Family Protection** – Ensuring financial security for dependents in case of death.
  + **Wealth Creation** – Investment-linked insurance for financial growth.
  + **Child’s Education** – Long-term savings for future educational expenses.
  + **Retirement Planning** – Securing a steady income post-retirement.

**Step 3: AI Assesses Risk Level & Calculates Recommended Coverage**

**Formula Used for Coverage Calculation:**

Recommended Coverage=(Annual Income×10)+Liabilities−Existing Insurance\text{Recommended Coverage} = (\text{Annual Income} \times 10) + \text{Liabilities} - \text{Existing Insurance}Recommended Coverage=(Annual Income×10)+Liabilities−Existing Insurance

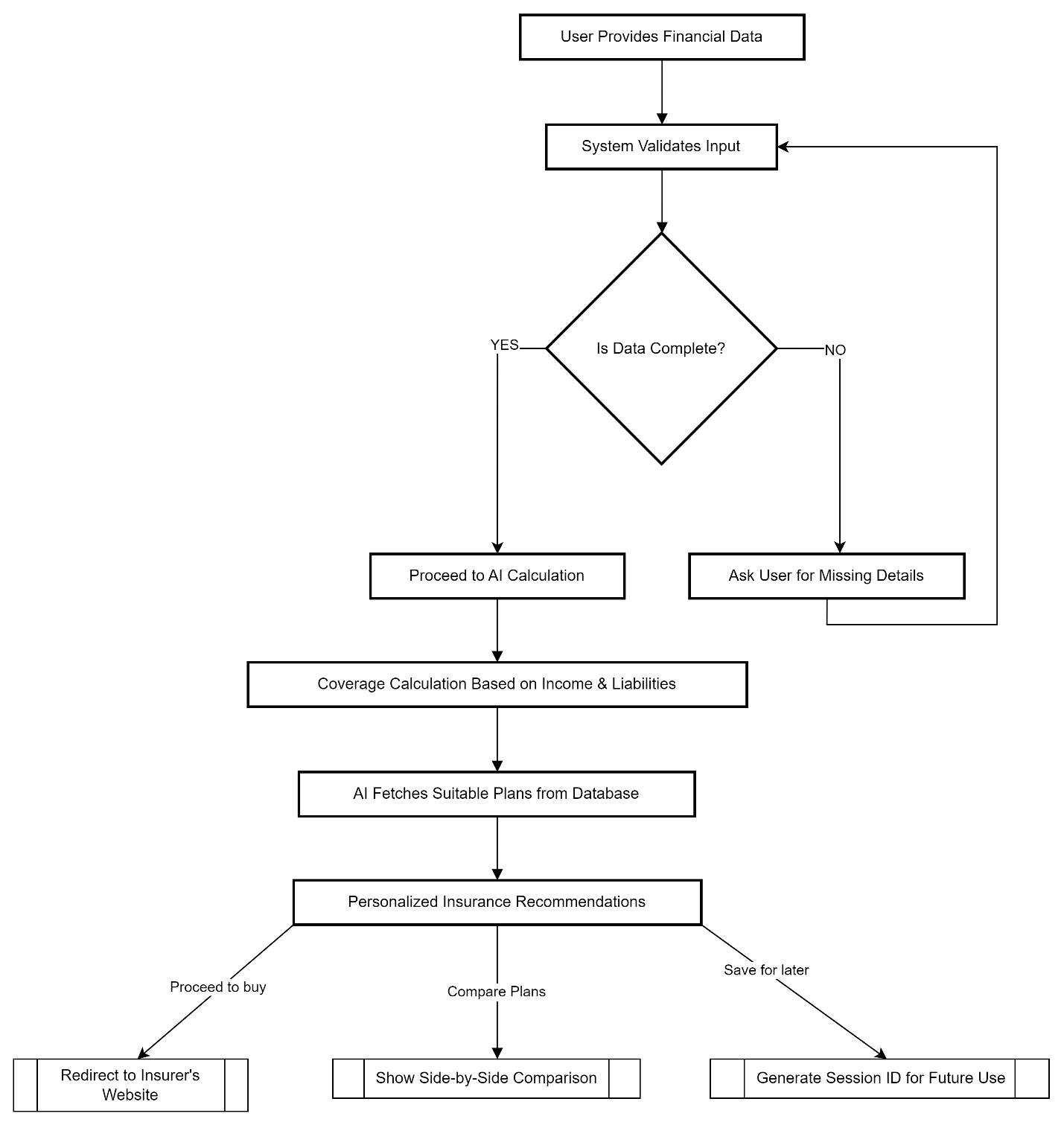
**Example Calculation:**

* **Income Protection (10x Annual Income):** ₹1 crore
* **Debt Repayment (Liabilities):** ₹20 lakh
* **Less Existing Coverage:** ₹5 lakh
* **Total Recommended Coverage:** ₹1.15 crore
* Chatbot: *"For your financial situation, I recommend a life insurance coverage of ₹1.15 crore to ensure your family's financial security."*
* Chatbot suggests **2-3 suitable plans** based on affordability and coverage.

**Expected Outcomes:**

* The user receives an **AI-powered recommendation** tailored to their financial situation.
* The system dynamically adapts based on user preferences and constraints.
* The backend stores session data for retrieval and analytics.

**Diagram: AI-Powered Need Analysis & Plan Matching – Process Flow**



**Sub-Process: -**

### 5.3.1 User Wants More Customization

**Scenario:**

* **The user wants to personalize their insurance plan by:** 
  + Modifying coverage amount (e.g., increasing from ₹1.15 crore to ₹1.5 crore).
  + Changing policy tenure (e.g., extending from 20 years to 30 years).
  + Adding riders (e.g., critical illness, accidental death benefit).

**Chatbot Action:**

1. Chatbot: *"Would you like to customize your policy? You can modify coverage, tenure, or add benefits."*
2. User selects customization options (e.g., increase coverage to ₹1.5 crore, add a critical illness rider).
3. The chatbot recalculates the premium and updates policy recommendations.
4. Updated options are displayed, allowing the user to finalize their selection.

### 5.3.2 User Wants to Compare Plans

**Scenario:**

* The user prefers to evaluate multiple insurance plans before making a decision.

**Chatbot Action:**

1. Chatbot: *"Would you like to compare plans from different insurers?"*
2. The chatbot fetches a side-by-side comparison of the best-matching policies.
3. The user reviews premium, coverage, tenure, riders, and benefits.
4. The user selects a preferred plan or requests further comparisons.

**Example Plan Comparison Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Plan A (Term Insurance)** | **Plan B (ULIP)** | **Plan C (Endowment)** |
| **Coverage** | ₹1.15 Cr | ₹75L | ₹50L |
| **Premium (Annual)** | ₹12,000 | ₹20,000 | ₹18,000 |
| **Investment Component** | No | Yes | Partial |
| **Riders Available** | Yes | Yes | Yes |
| **Maturity Benefit** | No | Yes | Yes |

## 5.4 Plan Recommendation & Premium Estimation

**Process Description:**

Once the chatbot has analyzed the user's financial details, goals, and risk preferences, it recommends suitable life insurance plans. Additionally, the chatbot provides a **premium estimation**, allowing the user to evaluate the affordability of different policies before making a decision.

**Actors:**

1. **User** – A customer seeking life insurance options.
2. **LINAT (Chatbot)** – AI-powered chatbot recommending and explaining insurance policies.
3. **Insurance Provider APIs** – External data sources fetching real-time insurance products and pricing.

**Objectives:**

* Present the most suitable **insurance plans** based on user needs.
* Estimate **premium costs** before the user selects a policy.
* Provide **comparisons between different plans** to assist decision-making.

**Preconditions:**

* The chatbot has **completed the needs analysis**, collecting financial data and risk preferences.
* The system has access to **real-time or stored insurance plans** from insurers.
* The user is ready to explore **policy options** based on recommendations.

**Postconditions:**

* The chatbot **displays multiple insurance options** for user comparison.
* The user can proceed with a **plan selection, request more details, or customize the coverage**.
* The chatbot provides a **clear breakdown of premium estimates**.

**Process Flow:**

**Step 1: Chatbot Presents Recommended Plans**

* **Chatbot:** "Based on your financial profile and goals, I have found some insurance plans that suit you. Here are your top recommendations:"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Plan Name** | **Coverage** | **Premium (₹/Year)** | **Policy Term** | **Key Features** |
| Secure Future Plan | ₹1 Cr | ₹8,500 | 25 Years | Term Plan, Critical Illness Rider |
| Wealth Builder ULIP | ₹75L | ₹12,000 | 20 Years | Market-Linked, High Growth Potential |
| Smart Retirement Endowment | ₹50L | ₹10,000 | 15 Years | Fixed Returns, Maturity Payout |

* **Chatbot Highlights Key Differences:**
  + Term vs. Investment-based Plans
  + Premium differences based on **coverage and tenure**
  + **Payout structure** and **flexibility of coverage**

**Step 2: User Selects a Plan or Requests More Details**

* **User:** "Tell me more about the Secure Future Plan."
* **Chatbot Provides Detailed Breakdown:**
  + **Coverage Details**: Sum assured, tenure, claim settlement process
  + **Payout Options**: Lump sum vs. monthly income
  + **Benefits & Tax Savings**: Eligibility under **Section 80C and 10(10D)**

**Step 3: User Requests a Premium Estimate**

* **Chatbot:** "Would you like to see an estimated premium based on your age and selected coverage?"
* **User:** "Yes, calculate my premium."

**Sub-Processes:**

### 5.4.1 User Requests a Premium Estimation

**Scenario:**

The user wants to **calculate the expected premium cost** before selecting a plan.

**Chatbot Actions:**

1. **Chatbot Asks for Inputs:** "Please confirm your age, coverage amount, and policy term."
2. Premium Calculation Formula:Premium=(1000Base Rate×Sum Assured)+Rider Cost+Taxes

Premium=(Base Rate×Sum Assured1000)+Rider Cost+Taxes\text{Premium} = \left( \frac{\text{Base Rate} \times \text{Sum Assured}}{1000} \right) + \text{Rider Cost} + \text{Taxes}

1. **Chatbot Displays Estimated Premium:**
   * "For a ₹1 Cr coverage, your estimated premium is ₹8,500 per year."
2. **User Selects Next Steps:**
   * **Modify Coverage**
   * **Proceed to Purchase**
   * **Compare with Other Plans**

### 5.4.2 User Wants a Tax-Saving Plan

**Scenario:**

The user specifically **looks for tax-saving insurance policies** to maximize benefits.

**Chatbot Actions:**

1. **Chatbot Asks for Tax Preferences:** "Are you looking to maximize tax savings under Section 80C or 10(10D)?"
2. **Chatbot Filters and Displays Tax-Efficient Plans:**

|  |  |
| --- | --- |
| **Plan** | **Tax Benefit** |
| Plan A | ₹1.5L tax deduction under **Section 80C** |
| Plan B | Tax-free maturity benefit under **Section 10(10D)** |

1. **User Explores Tax-Saving Options** before finalizing a plan.

**Importance of This Process:**

* Helps users compare and evaluate different life insurance plans based on affordability.
* Provides financial clarity by estimating premiums before finalizing a policy.
* Enhances decision-making by factoring in tax benefits and coverage flexibility.

## 5.5 Finalizing the Decision & Policy Issuance

**Process Overview**

Once a user has completed the need analysis, explored policy options, and selected an insurance plan, the system must guide them toward finalizing the purchase. The chatbot ensures a **smooth transition** to the insurer’s **policy issuance platform** while also offering advisory support if needed.

**Actors**

1. **User** – A customer ready to purchase a life insurance policy.
2. **LINAT (Chatbot)** – AI-powered chatbot assisting with policy selection and purchase.
3. **Insurance Provider’s Platform** – The external system where the final policy purchase takes place.

**Objectives**

* Seamlessly connect the user to the **insurance provider’s platform** for policy issuance.
* Ensure the user has **all the necessary details** before making a purchase.
* Provide **additional advisory support** for users requiring assistance.

**Preconditions**

* The user has **completed need analysis** and **selected a policy**.
* The chatbot has **gathered and validated** all necessary inputs.
* The user is ready to proceed with the purchase but may require **further assistance**.

**Postconditions**

* The chatbot **redirects the user** to the insurer’s **policy issuance page**.
* The system **saves session progress** in case the user wants to return later.
* If the user requests assistance, the chatbot **facilitates contact** with an advisor.

**Process Flow**

**Step 1: User Confirms Policy Selection**

1. The chatbot displays a **summary of the selected policy**, including:
   * **Coverage Amount:** ₹1 Crore
   * **Premium:** ₹8,500 per year
   * **Policy Term:** 25 years
   * **Riders Included:** Critical Illness, Accidental Death
2. **Chatbot:** "Are you happy with this policy, or would you like to make any changes?"
3. **User:** Selects **"Proceed to Buy"**

**Step 2: Chatbot Generates a Redirection Link**

1. The chatbot **fetches the direct purchase link** from the insurer’s database.
2. **Chatbot:** "You will now be redirected to [Insurer’s Name] to complete your purchase."
3. The chatbot displays a button:
   * **[Proceed to Buy]**

**Step 3: User Clicks "Proceed to Buy" and is Redirected**

1. The chatbot **routes the user to the insurer’s secure policy issuance page**.
2. If the user is **already logged in**, their details are **pre-filled** for quick completion.

**Sub-Processes**

### 5.5.1 User Requests Advisory Support Before Purchase

**Scenario:**

Some users may prefer **human assistance** before committing to a policy. They may have **questions about policy features, riders, or tax benefits**.

**Chatbot Actions:**

1. The chatbot detects **hesitation** if the user lingers on the **policy summary screen**.
2. **Chatbot:** "Would you like to speak to an insurance advisor before proceeding?"
3. If the user selects **“Yes”**, the chatbot:
   * **Schedules a call** with an advisor.
   * Displays a **phone number** or **live chat option** for immediate support.
   * Saves the **policy selection** so the advisor can access it during the conversation.

### 5.5.2 User Exits Midway & Wants to Resume Later

**Scenario:**

The user **gets distracted** or needs **more time** before making a decision. They may also need to **consult family members** before proceeding.

**Chatbot Actions:**

1. The chatbot detects **inactivity** and prompts:
   * **"Would you like to save your progress and continue later?"**
2. If the user selects **"Yes"**, the chatbot:
   * Generates a **unique session ID**.
   * Saves the **selected policy and user responses**.
   * Sends an **email/SMS** with a **session retrieval link**.
3. When the user **returns**, they can **resume from where they left off** using their **session ID or email**.
4. **Importance of This Process**

* Ensures a **seamless transition** from plan selection to **policy purchase**.
* Reduces **drop-offs** by allowing users to **save their progress** and return later.
* Increases **trust and engagement** by offering **expert assistance** when needed.

**Flow Diagram**

## 5.6 Session Tracking, Retrieval & User Data Management

**Process Overview**

Users may begin interacting with the chatbot but leave the session before completing their life insurance need analysis. To enhance user experience and prevent repeated data entry, the system ensures that user progress is **automatically saved** and can be retrieved when the user returns.

**Actors**

1. **User** – A customer who starts the process but may not complete it in one session.
2. **LINAT (Chatbot)** – AI-powered chatbot responsible for saving and retrieving user progress.
3. **Database** – Stores session history, including user responses and recommendations.

**Objectives**

* Enable users to **resume their insurance need analysis** at any time.
* Reduce **data re-entry** and enhance continuity in user interactions.
* Provide users with a **summary of past interactions** when they return.

**Preconditions**

* The user **starts interacting** with the chatbot and enters partial information but exits before completing the process.
* The chatbot **generates a session ID** or links progress to the user’s **email/phone number** for retrieval.

**Postconditions**

* The chatbot **saves user responses**, generates a **session ID**, or associates the progress with the user’s email/phone number.
* When the user **returns**, the chatbot retrieves the **saved session**, allowing them to continue from where they left off.

**Process Flow**

**Step 1: User Exits the Chatbot Midway**

* The **user stops responding**, closes the chat, or logs out before completing the need analysis.
* The chatbot **automatically saves the session** to prevent loss of data.

**Step 2: Chatbot Generates a Session ID**

* A **unique session ID** is created and stored in the **database**.
* If the user is **logged in**, progress is linked to their **email/phone number**.
* The chatbot **sends an email/SMS** with the session ID and a **link to resume later**.

**Step 3: User Returns & Selects "Retrieve My Saved Analysis"**

* When the user **revisits the chatbot**, they are given the option:
  + **Start a new analysis**
  + **Retrieve my saved session**

**Step 4: Chatbot Asks for Session ID or Email**

* The chatbot prompts:
  + **"Welcome back! Would you like to continue your last session? Please enter your session ID or email."**
* The user enters their **session ID** or registered **email/phone number**.

**Step 5: Chatbot Retrieves & Displays Saved Progress**

* The chatbot **loads previously entered responses** and asks if the user wants to:
  + **Resume from the last saved point**
  + **Modify any details before continuing**

**Step 6: User Resume Analysis**

* The chatbot **picks up from where the user left off** and continues the **recommendation process**.
* If any **details need updating**, the user can **modify inputs** before proceeding.

**Sub-Processes**

### 5.6.1 User Selects "Saved Analysis"

**Scenario:**

* A user wants to **access their previously saved need analysis report** instead of starting over.
* This is useful when the user wants to **revisit recommended insurance plans** before making a decision.

**Chatbot Actions:**

1. The chatbot **retrieves the saved analysis summary**, including:
   * **Recommended life insurance coverage**.
   * **Suggested policy types**.
   * **Estimated premium**.
   * **Financial insights** based on previous inputs.
2. The user can choose to:
   * **Continue exploring recommendations**.
   * **Modify details and recalculate insurance needs**.
   * **Proceed with policy selection**.

### 5.6.2 User Selects "My Profile"

**Scenario:**

* A user wants to **view their saved insurance policies**, past recommendations, and **personal details**.
* This applies to **logged-in users** who have already interacted with the chatbot multiple times.

**Chatbot Actions:**

1. The chatbot **redirects the user to their dashboard**, where they can:
   * **View previously saved policies**.
   * **Check past need analysis reports**.
   * **Update personal details** (income, dependents, etc.).
   * **Compare new insurance plans** with their existing policies.
2. If the user has **not logged in**, the chatbot **asks for authentication** before accessing their profile.
3. **Importance of This Process**

* Ensures a **seamless user experience** by allowing users to **pause and resume** their journey.
* Reduces **drop-offs** by enabling users to **retrieve saved progress** instead of starting over.
* Enhances **data accuracy and user engagement** by maintaining a **continuous interaction flow**

## 5.7 Knowledge Base Search for Policy & FAQs (RAG-Based System)

**Description:**

This process allows users to ask **free-form questions** related to **insurance policies, benefits, riders, claims, or terms** instead of selecting a predefined option. The chatbot retrieves **real-time, relevant, and contextual answers** from a **Retrieval-Augmented Generation (RAG)-based knowledge base** to ensure accuracy and up-to-date responses.

**Actors:**

1. **User** – A customer who wants **detailed information** about insurance policies before making a decision.
2. **LINAT (Chatbot)** – AI-powered chatbot that **searches the knowledge base** to retrieve answers.
3. **Insurance Knowledge Base (RAG-Based System)** – AI-enhanced repository containing:
   * Policy details, benefits, terms, and conditions.
   * FAQs related to claims, payouts, exclusions, and tax benefits.
4. **External APIs** – Fetching real-time policy and regulatory updates from **IRDAI or insurance provider APIs**.

**Goal:**

* To **instantly retrieve and present accurate answers** to user queries.
* To provide **personalized explanations** using AI-driven retrieval.
* To enhance **user trust and engagement** by eliminating manual searches.

**Preconditions:**

* The user has **specific questions** about life insurance policies, riders, payouts, tax benefits, or other related topics.
* The chatbot is **connected to a structured RAG-based knowledge base** for retrieving answers.

**Postconditions:**

* The chatbot provides a **precise, well-explained response** to the user’s query.
* The chatbot offers **further links, document references, or insurer contact details** for more clarity.

**Flow:**

**Step 1: User Asks a Free-Form Question**

* Instead of selecting a predefined option, the user types:
  + *"What’s the best term insurance for ₹1 crore?"*
  + *"How do I claim an insurance payout?"*
  + *"Can I add critical illness cover to my term plan?"*

**Step 2: Chatbot Searches the Knowledge Base**

* The chatbot **analyzes the query using Natural Language Processing (NLP)**.
* **RAG Model** retrieves **relevant sections** from the knowledge base.
* AI ranks the best responses based on **context, accuracy, and recency**.

**Step 3: AI Retrieves and Summarizes the Answer**

* The chatbot **filters out irrelevant details** and presents a **structured answer**.
* **Example Response:**
  + *"For a ₹1 crore term insurance plan, here are the top options based on premium and claim settlement ratio."*
  + *"Would you like to compare policies or proceed with a recommendation?"*

**Step 4: The user is Provided With Additional Resources**

* The chatbot **offers supporting links** to detailed policy documents.
* If required, the chatbot **offers to connect the user to an advisor**.

**Sub-Use Cases:**

### 5.7.1 User Asks About Policy Payout Without Buying a Policy

**Scenario:**

* A user is curious about **how insurance payouts work** before committing to a policy.
* They ask:
  + *"How will my family receive the insurance payout?"*
  + *"Can I get monthly payments instead of a lump sum?"*

**Chatbot Action:**

1. The chatbot **retrieves payout structures** from policy documents.
2. Provides **clear, structured explanations**:
   * **Lump Sum Payout:** A one-time full payment.
   * **Monthly Income Payout:** The sum assured is paid in **installments over a chosen period**.
   * **Combination Payout:** A mix of both options.
3. If the user wants to explore more, the chatbot **offers links to policies that support flexible payouts**.

### 5.7.2 User Wants Insurance for Business Loan Collateral

**Scenario:**

* A user requires **life insurance as security for a business loan**.
* They ask:
  + *"Can I use my life insurance policy as collateral for a business loan?"*

**Chatbot Action:**

1. The chatbot **identifies suitable loan protection insurance** options.
2. Suggest policies like **Decreasing Term Insurance** (where coverage reduces as the loan is repaid).
3. Provides a **comparison of business loan-friendly life insurance plans**.
4. Offers a **direct link to insurers who support collateralized insurance policies**.

## 5.8 Input Validation & Data Consistency Checks

**Description:**

The chatbot ensures **accuracy and reliability** in user-provided data while identifying **inconsistencies, incomplete entries, or unrealistic financial inputs**. This feature is essential to improve the accuracy of life insurance recommendations based on a user’s actual financial standing.

**Actors:**

1. **User** – A customer filling out personal and financial details in the chatbot.
2. **LINAT (Chatbot)** – AI-powered assistant that analyzes user responses for inconsistencies.
3. **Insurance Provider’s Backend System** – Stores and verifies inputted details based on industry benchmarks.

**Goal:**

* To **validate and cross-check** user inputs for accuracy.
* To **detect conflicting or unrealistic financial entries** that might affect policy recommendations.
* To **ensure accurate policy suggestions** by verifying that user responses align with practical financial scenarios.

**Preconditions:**

* The user is filling out the chatbot’s **insurance questionnaire** and providing personal and financial details.
* The chatbot has predefined **rules for inconsistencies** (e.g., unrealistic income vs. expense ratios, extreme coverage requests).

**Postconditions:**

* If all responses are valid, the chatbot proceeds with **need analysis and policy recommendations**.
* If conflicting or unrealistic responses are found, the chatbot will **prompt clarification questions** before continuing.

**Flow:**

**Step 1: User Enters Personal & Financial Details**

* The chatbot prompts users to enter key financial information such as:
  + **Age, Gender, Marital Status**
  + **Annual Income & Monthly Expenses**
  + **Liabilities (Loans, Mortgages, Credit Card Debts, etc.)**
  + **Dependents (Children, Spouse, Parents, etc.)**
  + **Existing Insurance Coverage**

**Step 2: AI Cross-Checks User Data**

* The chatbot applies **validation rules** to check for inconsistencies, including:
  + Does the reported income match the estimated savings and expenses?
  + Is the coverage amount requested reasonable based on income?
  + Are declared debts in proportion to income and liabilities?
* If all data appears valid, the chatbot proceeds to **need analysis and recommendations**.

**Step 3: Detecting Unusual Patterns or Mismatched Entries**

* If **data inconsistencies** are found, the chatbot flags them. Examples include:
  + A user earning ₹20,000 per month claiming ₹1 crore in savings.
  + A person with no declared income requesting ₹10 crore coverage.
  + A user with ₹50 lakh in loans but claiming ₹0 monthly expenses.

**Step 4: Clarification Questions for User Mismatches**

* If an inconsistency is found, the chatbot asks for clarification:
  + "I noticed that your expenses are ₹50,000 per month, but your monthly income is ₹30,000. Can you confirm these details?"
  + "You have mentioned ₹10 lakh in liabilities but no existing savings. Would you like to adjust your figures?"
* The user either **corrects their inputs** or **confirms the accuracy**.

**Step 5: Handling Unrealistic Data Entries**

* If a user enters **extreme values**, the chatbot provides guidance:
  + "Typically, life insurance coverage is **10-15x annual income**. Your request is significantly higher than the recommended range. Would you like to reconsider?"
* The chatbot **suggests a realistic coverage range** based on the user’s financial status.

**Sub-Use Cases:**

### 6.8.1 User Provides Incomplete or Conflicting Details

**Scenario:**

* A user enters **contradictory financial details** (e.g., claiming high savings but large debts).

**Chatbot Action:**

* The chatbot detects inconsistencies and prompts clarification:
  + "Your responses indicate high savings but also significant debt. Could you confirm if this is accurate or needs adjustment?"
* If the user provides valid corrections, the chatbot **continues the process smoothly**.
* If the user does not provide corrections, the chatbot **recommends a reassessment of financial inputs**.

### 6.8.2 User Exaggerates Income or Coverage Needs

**Scenario:**

* A user earning ₹25,000/month requests ₹10 crore in life insurance coverage.

**Chatbot Action:**

* The chatbot analyzes industry benchmarks and prompts:
  + "Based on your current income, we suggest a coverage range of ₹50 lakh - ₹1.5 crore for optimal protection. Would you like to adjust your request?"
* The chatbot **suggests a coverage range** that aligns with the user’s financial status.

### 6.8.3 User Skips Key Financial Questions

**Scenario:**

* A user **does not enter income, liabilities, or dependents**, which may lead to incorrect recommendations.

**Chatbot Action:**

* The chatbot prompts the user with:
  + "Providing accurate financial details ensures we recommend the best policy for you. Would you like to enter an estimated value for now?"
* If the user **still skips**, the chatbot makes an **educated assumption** and informs the user:
  + "Based on market data, we’ve estimated an annual income of ₹8 lakh for your profile. You can modify this if needed."

## 5.9 Advanced User Scenarios & Edge Cases

**Scenario:**

As users interact with the AI-based insurance need analysis system, various edge cases and complex user behaviors may arise. The chatbot must be designed to handle these efficiently to ensure a smooth and seamless user experience. Below are some of the most common advanced scenarios along with their resolutions.

### 5.9.1User Exits Mid-Process Without Completing Any Steps

**Description:**

Some users may initiate the chatbot conversation but exit without providing any details. This could be due to lack of time, distractions, or second thoughts about purchasing insurance. The chatbot must ensure that such users are re-engaged and encouraged to complete the process.

**Actors:**

**1. User** – Starts the chatbot session but does not proceed with entering details.

**2. LINAT (Chatbot)** – Detects inactivity and initiates follow-up actions.

**3. Notification/Email System** – Sends reminders to users.

**Flow:**

1. **User visits chatbot but leaves before answering any questions.**
2. **Chatbot detects inactivity** (e.g., no response for 1-2 minutes).
3. **Sends a reminder message:**
   * "It looks like you left before we could complete your insurance analysis. Would you like to continue where you left off?"
4. **If the user does not return within 24 hours, the system sends an automated reminder email/SMS.**

**Example Chatbot Conversation:**

* **Chatbot:** "I noticed you left before completing your insurance need analysis. Would you like to pick up where we left off?"
* **User:** "Yes, resume my session."
* **Chatbot:** "Great! Let’s continue from where you stopped."

### 5.9.2User Keeps Changing Answers

**Description:**

Some users may be indecisive and continuously switch between different insurance goals (e.g., switching between "Wealth Creation" and "Retirement Planning" multiple times). The chatbot must help users compare options clearly to reduce confusion.

**Actors:**

1. **User** – Provides inconsistent responses regarding their insurance needs.
2. **LINAT (Chatbot)** – Detects inconsistent answers and guides the user effectively.

**Flow:**

1. **User initially selects one goal (e.g., Wealth Creation).**
2. **Later changes to another goal (e.g., Retirement Planning).**
3. **Keeps switching between different options.**
4. **Chatbot detects the pattern and offers a side-by-side comparison:**
   * "I noticed you are considering both Wealth Creation and Retirement Planning. Would you like to compare them side by side?"
5. **User reviews the comparison and finalizes their decision.**

**Example Chatbot Conversation:**

* **User:** "I think I want a plan for Wealth Creation."
* **Chatbot:** "Great! Let’s explore the best wealth-building insurance plans for you."
* **User:** "Wait, I’m not sure. Maybe I should focus on Retirement Planning instead."
* **Chatbot:** "I see you’re considering both options. Let’s compare Wealth Creation vs. Retirement Planning to help you decide."

### 5.9.3User's Internet Connection Is Lost Midway

**Description:**

A user may experience an unexpected **internet disconnection** while interacting with the chatbot. To prevent frustration, the chatbot should **save session progress** and allow easy resumption once the user reconnects.

**Actors:**

1. **User** – Experiences an internet disconnection while interacting with the chatbot.
2. **LINAT (Chatbot)** – Saves progress and prompts the user to resume upon reconnection.

**Flow:**

1. **User is actively providing details to the chatbot.**
2. **Internet connection is lost or the user unintentionally closes the chat.**
3. **Chatbot saves session progress with a session ID.**
4. **When the user reconnects, the chatbot prompts:**
   * "Would you like to resume where you left off?"
5. **User enters session ID or email to retrieve their saved progress.**

**Example Chatbot Conversation:**

* **User:** "I need a term plan with ₹1 crore coverage for 20 years."
* **Chatbot:** "Sure! I’ll get the best options for you..."
* **[User loses internet connection]**
* **[Later, user returns]**
* **Chatbot:** "Welcome back! I see your last session was interrupted. Would you like to resume where you left off?"
* **User:** "Yes, please continue."

### 5.9.4User Requests Multiple Customizations

**Description:**

Some users may wish to make **multiple customizations** to their insurance policy before finalizing their choice. They may want to **adjust coverage amounts, change policy terms, add or remove riders, or modify their investment strategy multiple times.** The chatbot should handle **dynamic recalculations** and suggest optimized plans based on each modification.

**Actors:**

1. **User** – Requests several changes to coverage, tenure, or riders.
2. **LINAT (Chatbot)** – Recalculates recommendations dynamically.

**Flow:**

1. **User selects an insurance plan.**
2. **Requests changes to sum assured, tenure, and riders.**
3. **Chatbot updates premium calculations and re-suggests optimized plans.**
4. **User makes further modifications.**
5. **Chatbot dynamically adapts and refines recommendations.**

**Example Chatbot Conversation:**

* **User:** "I need ₹1 crore coverage with a 20-year term."
* **Chatbot:** "Your estimated premium is ₹7,500/year. Would you like to customize it?"
* **User:** "Yes, increase my coverage to ₹1.5 crore."
* **Chatbot:** "With ₹1.5 crore coverage, the new premium is ₹9,200/year. Would you like to add a Critical Illness Rider?"
* **User:** "Yes, add that too."
* **Chatbot:** "With the Critical Illness Rider, your total premium is ₹10,500/year. Here’s the updated breakdown: [Plan Details]"

### 5.9.5User Requests a Plan Comparison After Multiple Customizations

**Description:**

After customizing a policy, users may still **want to compare it with other available plans** to ensure they are getting the best option. The chatbot must efficiently **compare customized plans vs. standard plans** for easy decision-making.

**Actors:**

1. **User** – Requests a comparison between their customized plan and other options.
2. **LINAT (Chatbot)** – Retrieves and displays **comparison data.**
3. **External Insurance APIs** – Fetches real-time policy details.

**Flow:**

1. **User modifies a plan (e.g., increases coverage, adds riders).**
2. **Requests comparison with similar policies.**
3. **Chatbot presents a side-by-side comparison.**
4. **User reviews details and finalizes a choice.**

**Example Chatbot Conversation:**

* **User:** "This customized plan looks good, but how does it compare to other similar plans?"
* **Chatbot:** "I’ll fetch similar policies from different insurers for comparison..."
* **[Displays comparison table]**
* **User:** "Plan B has a lower premium, but Plan C has better coverage. Let me go with Plan C!"

## 5.10 Coverage Comparison & Plan Selection

**Description:**

Many users may want to compare different insurance plans before making a final decision. The **Coverage Comparison & Plan Selection** use case enables users to analyze various insurance policies based on parameters such as **coverage amount, premium, policy term, riders, claim settlement ratio, exclusions, and additional benefits.** The chatbot facilitates this by fetching real-time data from different insurance providers, presenting the comparison in an easy-to-understand format, and guiding the user toward selecting the most suitable policy.

**Actors:**

1. **User** – A potential customer looking to compare multiple insurance policies before selecting one.
2. **LINAT (Chatbot)** – The AI-driven chatbot that retrieves, processes, and presents insurance comparison data.
3. **External Insurance APIs** – Third-party services or insurer APIs that provide **real-time policy details**, including coverage, premium, exclusions, and other features.

**Goal:**

To **empower users** with a transparent and **data-driven comparison** of insurance policies, enabling them to select the best option based on their financial goals, affordability, and risk appetite.

**Preconditions:**

* The user has completed the **need analysis process** or manually chooses to compare plans.
* The chatbot has access to **insurance provider APIs** or a **knowledge base of insurance policies.**

**Postconditions:**

* The user receives a **side-by-side comparison** of multiple policies tailored to their financial needs.
* The user can **proceed to purchase, modify criteria, or request an advisor’s help** for further clarification.

**Flow:**

**1. User Initiates Coverage Comparison**

* **User selects “Compare Plans”** either after completing a need analysis or from the main chatbot menu.
* **Chatbot asks what type of insurance the user wants to compare:**
  + Term Insurance
  + Endowment Plan
  + ULIP
  + Child Education Plan
  + Retirement Plan
* **User selects the desired insurance type.**

**2. Chatbot Fetches Top Insurance Plans**

* **LINAT retrieves policy details** from multiple insurers using:
  + External insurance APIs
  + A pre-built product database
  + A real-time pricing engine
* **Factors considered for comparison:**
  + **Coverage Amount (Sum Assured)**
  + **Premium Amount**
  + **Policy Term**
  + **Riders Available (Critical Illness, Accidental Death, Waiver of Premium, etc.)**
  + **Claim Settlement Ratio (CSR%)**
  + **Exclusions & Waiting Periods**
  + **Tax Benefits**

**3. Displaying the Comparison Table**

**Example Table Format:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Plan A (Insurer X)** | **Plan B (Insurer Y)** | **Plan C (Insurer Z)** |
| Coverage Amount | ₹1 Cr | ₹1 Cr | ₹1.2 Cr |
| Premium (Annual) | ₹8,500 | ₹9,000 | ₹8,200 |
| Policy Term | 25 years | 30 years | 20 years |
| Critical Illness Rider | Included | Not Included | Available |
| Accidental Cover | Included | Included | Not Included |
| Waiver of Premium | Not Available | Available | Available |
| Claim Settlement Ratio | 97% | 95% | 96% |
| Tax Benefits (80C, 10(10D)) | Yes | Yes | Yes |

**4. User Interacts with the Comparison Results**

Once the chatbot presents the **side-by-side comparison table**, the user can:

1. **Select a policy** and proceed with the purchase.
2. **Request detailed information** on a specific plan.
3. **Modify search filters** (adjust sum assured, policy term, premium amount).
4. **Save comparison results** under a session ID for later review.
5. **Speak with an insurance advisor** for additional guidance.

**Sub-Use Cases**

### 5.10.1 User Wants More Detailed Information on a Specific Plan

**Scenario:** The user wants additional details about a specific plan before making a decision.

**Chatbot Actions:**

* Fetches and displays **detailed policy documents**, including terms, benefits, exclusions, and fine print.
* Provides an option to **download a PDF summary of the policy.**

**Example Chatbot Conversation:**

* **User:** "I like Plan B, but can I get more details?"
* **Chatbot:** "Certainly! Here’s a summary of **Plan B (Insurer Y)** including benefits, coverage breakdown, and exclusions. Would you like to download a PDF?"
* **User:** "Yes, send it to my email."

### 5.10.2 User Adjusts Comparison Criteria

**Scenario:** The user realizes that their coverage amount or premium preference needs adjustment.

**Chatbot Actions:**

* Offers options to **modify the sum assured, policy term, or riders.**
* Fetches an **updated comparison table** based on the revised criteria.

**Example Chatbot Conversation:**

* **User:** "These plans are good, but I need ₹1.5 Cr coverage instead of ₹1 Cr."
* **Chatbot:** "No problem! Let me update the comparison with **₹1.5 Cr sum assured**. Here’s the revised table:"

### 5.10.3 User Saves Comparison Results for Future Review

**Scenario:** The user is not ready to decide immediately and wants to review the comparison later.

**Chatbot Actions:**

* Saves the **comparison session with a unique Session ID.**
* Offers to **send the comparison details via email or SMS.**

**Example Chatbot Conversation:**

* **User:** "I want to compare later."
* **Chatbot:** "I have saved your comparison results under **Session ID: 12345ABC**. You can retrieve them anytime using this ID."

### 5.10.4 User Wants to Speak with an Advisor Before Finalizing

**Scenario:** The user needs additional guidance from a human insurance expert before making a decision.

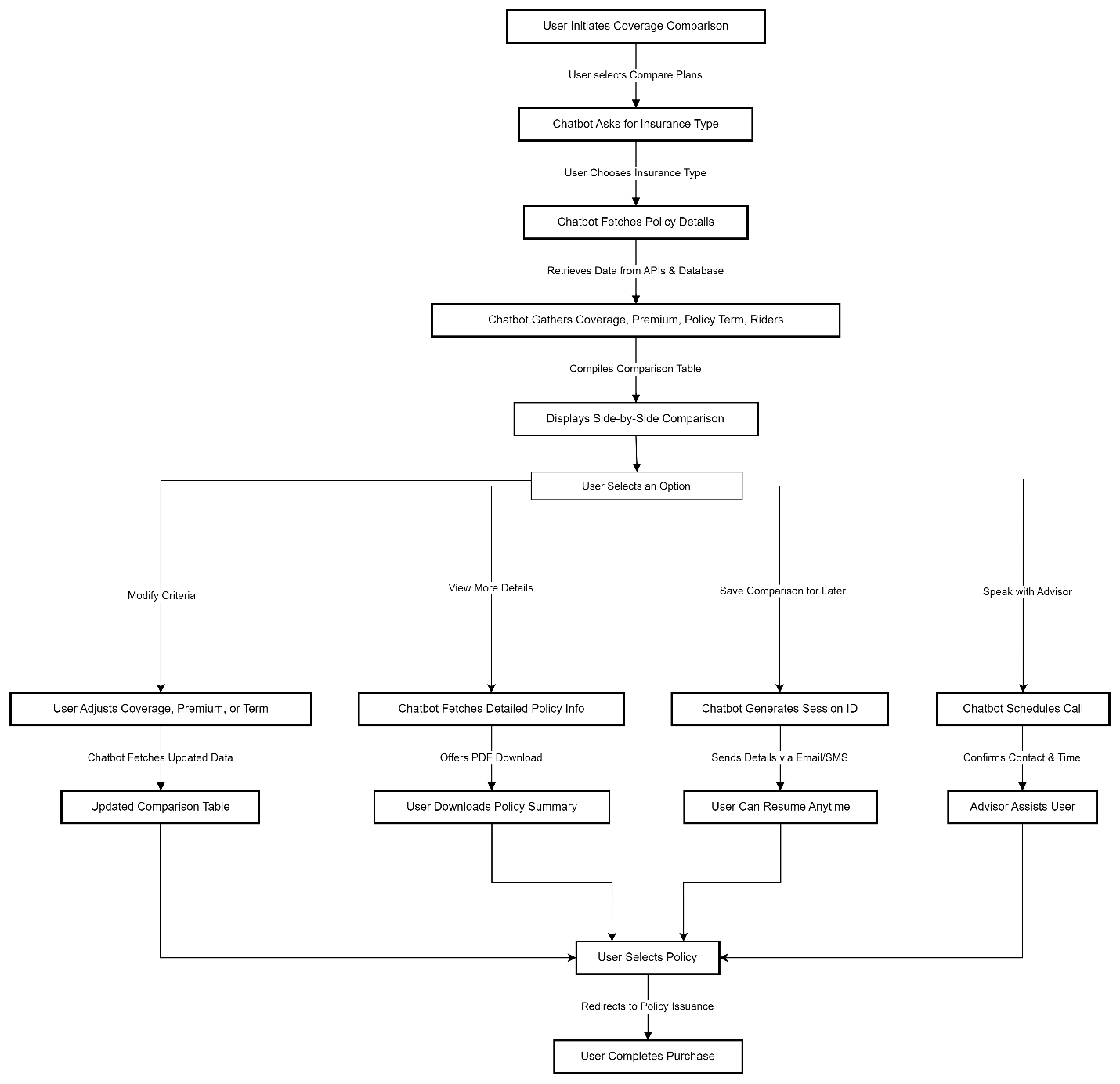
**Chatbot Actions:**

* Provides an option to **schedule a call with an advisor**.
* Collects contact details and preferred call timings.
* Sends confirmation and schedules an **advisor callback.**

**Example Chatbot Conversation:**

* **User:** "I’m still confused. Can I talk to someone?"
* **Chatbot:** "Absolutely! I can schedule a call with an advisor. What’s your preferred time?"
* **User:** "Tomorrow at 4 PM."
* **Chatbot:** "Done! Our insurance advisor will call you tomorrow at 4 PM. You’ll receive a confirmation SMS."

Flow Diagram



### 5.10.5 User Profile & Dashboard Overview

**Description**

The **"My Profile"** option allows users to access their personal dashboard, where they can view and manage their insurance details. This option is available through:

* The **chatbot's main menu** (e.g., "View My Profile").
* **After completing a need analysis**, where users can save and revisit their details.
* The **saved analysis page**, allowing users to check previous recommendations.

**Dashboard Contents**

Once the user selects **"My Profile,"** they are redirected to a personalized dashboard displaying all relevant insurance information.

**1. User's Basic Information**

* **Personal Details** (Name, Age, Gender, Marital Status).
* **Financial Snapshot** (Annual Income, Monthly Expenses, Liabilities).
* **Risk Profile & Insurance Goal** (e.g., Wealth Creation, Retirement Planning).

**2. Saved Need Analysis Reports**

* List of **previous insurance need calculations** with timestamps.
* Ability to **recalculate recommendations** based on updated financial data.

**3. Purchased or Recommended Policies**

* **Policies already purchased** with coverage details.
* **Recommended policies** based on past interactions.

**4. Premium Due Reminders & Policy Status**

* **Upcoming premium payment reminders** with due dates.
* **Active policy details**, including sum assured, tenure, and riders.

**5. Saved Comparisons & Premium Estimates**

* **Previously compared insurance plans** side by side.
* **Past premium calculations** for different coverage amounts.

**6. Policy Issuance & Claim Assistance**

* **Track policy issuance progress** (if the policy is not yet purchased).
* **Quick access to claim process & FAQs**.

**7. Tax Benefits & Financial Insights**

* **Estimated tax savings** under **Section 80C & 10(10D)**.
* **Personalized financial insights** based on insurance planning.

**8. Session History & Retrieval**

* **Access to previous chatbot interactions** for reference.
* Ability to **resume past conversations** using a session ID.

**User Actions in the Profile Dashboard**

**Update Financial Details** – Modify income, expenses, dependents, and liabilities.

**Recalculate Need Analysis** – Update coverage recommendations based on new data.

**Download Reports** – Get detailed PDFs of insurance recommendations.

**Proceed to Purchase** – Complete policy issuance from saved recommendations.

**Contact an Advisor** – Schedule a call for expert guidance.

**Flow Diagram :-**

## 5.11 Policy Issuance & Follow-Up

**Description:**

Once a user finalizes their insurance plan, they need a seamless **transition to the insurer’s website** to complete the policy purchase. The chatbot ensures that the user receives the correct information, is routed to the right page, and is provided with any additional support if needed.

**Actors:**

* **User** – A customer who has selected an insurance plan and is ready to proceed with the purchase.
* **LINAT (Chatbot)** – Provides the final policy details and generates a direct purchase link.
* **Insurance Company API** – Validates the user’s selected plan and redirects them to the insurer’s purchase portal.
* **Advisor (Optional Human Support)** – Provides expert guidance if the user requests additional help before purchasing.

**Goal:**

Ensure that the user **completes the policy purchase efficiently** and receives any necessary support along the way.

**Preconditions:**

* The user has successfully completed the **insurance need analysis and plan selection process**.
* The chatbot has retrieved the **selected policy details** and a valid purchase link from the **insurance company API**.

**Postconditions:**

* The user is successfully redirected to the insurer’s policy issuance page.
* The chatbot provides **post-purchase support options** (e.g., policy document download, customer support, follow-up reminders).

**Flow:**

1. **User confirms their insurance plan selection.**
2. **Chatbot retrieves final plan details** from the insurer’s API.
3. **Chatbot displays the final summary of the selected policy, including:**
   * Coverage amount
   * Policy term
   * Premium amount
   * Any selected riders (e.g., accidental death, critical illness cover)
   * Tax benefits and payout details
4. **Chatbot asks if the user wants to proceed with the purchase.**
   * **If yes**, chatbot provides a **secure policy purchase link** and routes the user to the insurer’s website.
   * **If no**, chatbot allows the user to:
     + Compare other plans
     + Modify inputs (coverage, term, riders)
     + Speak with an advisor
5. **User clicks "Proceed to Buy"** → Redirected to the insurer’s secure payment portal.
6. **Chatbot confirms redirection and offers follow-up options:**
   * "Would you like a reminder to review your policy details later?"
   * "Would you like an advisor to help you with the final steps?"
7. **If the user completes the purchase:**
   * Chatbot offers a **policy document download option**.
   * Provides **customer service contact** for future queries.
8. **If the user exits without purchasing:**
   * Chatbot saves the user’s selected plan under a **Session ID** for future retrieval.
   * Offers to **send an email reminder** with policy details.

Example Chatbot Interaction:

### 5.11.a User Proceeds with Purchase

* **Chatbot:** "You have selected the **Secure Future Plan** with ₹1 crore coverage for 25 years. Your premium is ₹6,000 per year. Do you want to proceed with the purchase?"
* **User:** "Yes, proceed to buy."
* **Chatbot:** "Great! You will be redirected to [Insurer Name]’s secure payment portal. Click below to continue."
* **User clicks the provided link and completes the purchase.**
* **Chatbot:** "Congratulations! Your policy has been successfully issued. You can download your policy document here: [Download Link]. Do you need help with anything else?"

### 5.11.b User Needs More Time to Decide

* **Chatbot:** "You have selected the **Wealth Plus Plan** with ₹75 lakh coverage for 20 years. Your premium is ₹5,500 per year. Would you like to proceed with the purchase?"
* **User:** "I need more time."
* **Chatbot:** "No problem! I have saved your selection under **Session ID: 98765XYZ**. Would you like me to send you a reminder email with your policy details?"
* **User:** "Yes, send an email."
* **Chatbot:** "Done! You can access your selected plan anytime using your Session ID. Let me know if you need any further assistance."

Sub-Use Cases:

### 5.11.1 User Requests Advisory Support Before Purchase

**Scenario:**

Some users may prefer **expert guidance before finalizing their policy purchase.** They might have **questions about coverage details, exclusions, premium flexibility, or tax benefits.** In such cases, the chatbot should provide an option to connect with a human advisor.

**Flow:**

1. **User selects "Speak to an Advisor" instead of proceeding to purchase.**
2. **Chatbot collects user details** (Name, Contact Number, Preferred Time for Call).
3. **Chatbot provides advisor contact options:**
   * Schedule a **call-back** from a licensed insurance advisor.
   * Connect immediately via **live chat** (if available).
4. **User selects the preferred option.**
5. **Chatbot confirms the request and provides a summary:**
   * "Your call with our advisor is scheduled for **tomorrow at 3 PM**. You will receive a confirmation SMS."
6. **If the user later decides to proceed with the purchase, the chatbot offers the policy link again.**

Example Chatbot Interaction:

#### User Requests Advisor Support

* **Chatbot:** "You have selected the **Smart Life Plan** with ₹1 crore coverage for 30 years. Your premium is ₹7,200 per year. Would you like to proceed with the purchase?"
* **User:** "I have some questions before buying."
* **Chatbot:** "I can connect you with one of our insurance experts. Would you like a call-back or a live chat?"
* **User:** "Schedule a call-back."
* **Chatbot:** "Sure! What’s your preferred time for the call?"
* **User selects:** "Tomorrow at 4 PM."
* **Chatbot:** "Done! Your call with our advisor is scheduled for tomorrow at 4 PM. You will receive a confirmation SMS. Would you like me to save your plan details for reference?"
* **User:** "Yes."
* **Chatbot:** "Your selection has been saved under **Session ID: 45678XYZ**. Let me know if you need anything else!"

NOTE: - For a detailed understanding of different Chatbot module and training Q&A, please refer to the document attached in the header as the reference document [(Doc\_008).](#_11_Reference_Documents)

# Integration

* OpenAI API for chatbot interactions
* Database for storing user inputs
* Insurance provider APIs for fetching plan details
* Analytics tools (Power BI, Tableau) for insights

NOTE: - For a detailed understanding of integration, please refer to the document attached in the header as the reference document [(Doc\_003).](#_11_Reference_Documents)

# 7 User Management Module

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Privilege Name** | **Description** | **Access Level** | **Impact on System** | **Restrictions/Notes** |
| **User Registration** | Users can sign up using email, phone, or SSO. | Create, View | Enables new users to access the system. | Public access, but verification is required. |
| **User Authentication** | Secure login via OTP, email, or multi-factor authentication. | Authenticate, Access | Ensures account security and controlled access. | Mandatory for all users. |
| **Profile Management** | Users can update personal details, financial goals, and policy preferences. | Edit, Update | Personalizes user experience. | Limited fields editable after verification. |
| **Role-Based Access Control** | Grants different access permissions based on user roles. | Assign, Modify | Controls feature accessibility based on roles. | Managed by Admin. |
| **User Activity Tracking** | Logs user interactions for analytics and insights. | View Only | Helps in system analytics and behavior analysis. | Restricted to Admin for privacy reasons. |
| **Saved Conversations** | Users can save chatbot conversations and revisit them later. | Save, Retrieve | Allows continuity in interactions. | Secured storage; accessible only to the user. |
| **Plan Recommendation History** | View past insurance recommendations and chatbot interactions. | View Only | Provides reference for past plan selections. | Only visible to users and advisors. |
| **Custom Plan Creation** | Users can create custom insurance plans based on their needs. | Create, Modify, Delete | Enhances user flexibility in financial planning. | Only available for logged-in users. |
| **Policy Management** | Users can view, edit, and track existing policies. | View, Edit | Facilitates ongoing policy modifications. | Editing is restricted post-policy issuance. |
| **Compare Plans** | Allows users to compare multiple insurance plans side by side. | View, Compare | Helps users make informed decisions. | Limited to selected insurance providers. |
| **Download Gap Report** | Generate a gap analysis report highlighting coverage shortfalls. | Generate, View | Provides insights into financial protection gaps. | Restricted to registered users. |
| **Live Plan Redirection** | Routes users to insurer websites for policy purchase. | Redirect, View | Ensures smooth transition from analysis to policy purchase. | Users cannot modify external plan details. |
| **Email & Session Storage** | Allows users to save interactions and receive session details via email. | Save, Send | Enables follow-up on past interactions. | User consent required before email is sent. |
| **Tax & Benefit Estimator** | Calculates tax benefits based on selected policies. | Auto-Calculate, View | Helps users understand policy tax advantages. | Automated calculations; no manual inputs allowed. |
| **Financial Need Assessment** | Conducts an in-depth analysis of user financial goals and risk appetite. | Analyze, Generate Report | Provides customized insurance recommendations. | Based on user-input financial details. |
| **Backend Data & Analytics** | Stores and processes user data for AI training and trend analysis. | View Only | Improves chatbot recommendation accuracy. | Admin-only access for security. |
| **Audit Logs** | Maintains records of all system activities for compliance tracking. | View Only | Ensures system transparency and security. | No modifications allowed. |
| **Account Deactivation** | Allows users to deactivate or delete their accounts upon request. | Deactivate, Delete | Provides users with control over their data. | Requires identity verification before action is completed. |

# 8 Appendix A - Wire Frames & Field Appendix

## 8.1 WF\_Webpage\_001



| **Field Name** | **Field Type** | **Data Type** | **Data Source** | **Read-only/Editable** | **Validation** |
| --- | --- | --- | --- | --- | --- |
| Need Analysis LINAT | Button | String (Text) | Static UI Component | Read-only | N/A (Navigational) |

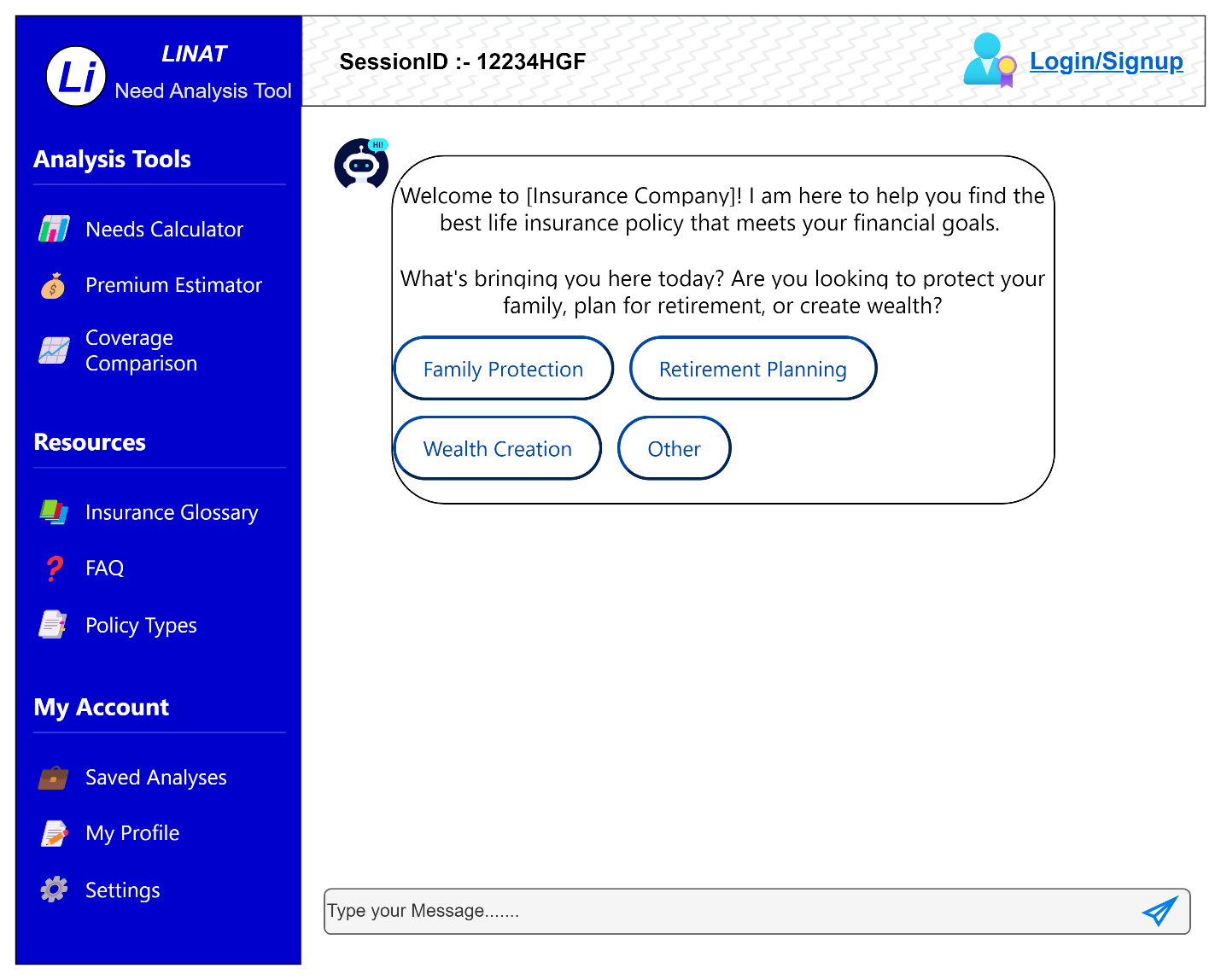
Explanation of Screen01 –

This is a **home page wireframe** for the **Life Insurance Company** from where the user can access our tool **“AI Need Analysis Tool (LINAT)”**. It serves as the **entry point** for users looking to assess their life insurance needs. Below is a detailed breakdown of each element on the screen.

**1. LINAT Chatbot Icon (Bottom Right Corner)**

* A **blue AI chatbot icon** with a shield symbol is present, likely representing LINAT’s assistant.
* This chatbot icon suggests:
  + **Instant AI-driven insurance guidance**.
  + **Automated customer support** for queries about policies, coverage, and premium calculations.

## 8.2 WF\_Chatbot\_Interface\_Greeting and Benefit\_offering\_002



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Data Source** | **Read-only/Editable** | **Validation** |
| Logo (LI - LINAT) | Image/Icon | PNG/JPEG | Static UI Component | Read-only | N/A |
| Tool Name (LINAT) | Text Label | String | Static UI Component | Read-only | N/A |
| Tool Description | Text Label | String | Static UI Component | Read-only | N/A |
| Needs Calculator | Navigation Button | Hyperlink | Internal Routing | Read-only | Valid Page Route |
| Premium Estimator | Navigation Button | Hyperlink | Internal Routing | Read-only | Valid Page Route |
| Coverage Comparison | Navigation Button | Hyperlink | Internal Routing | Read-only | Valid Page Route |
| Insurance Glossary | Navigation Button | Hyperlink | Internal Routing | Read-only | Valid Page Route |
| FAQ | Navigation Button | Hyperlink | Internal Routing | Read-only | Valid Page Route |
| Policy Types | Navigation Button | Hyperlink | Internal Routing | Read-only | Valid Page Route |
| Saved Analyses | Navigation Button | Hyperlink | Internal Routing | Read-only | Valid Page Route |
| My Profile | Navigation Button | Hyperlink | Internal Routing | Read-only | Dashboard page Route |

Explanation of screen02 –

This screen represents the **LINAT Need Analysis Chatbot Interface** where users interact with the system to assess their life insurance needs. Below is a breakdown of each section and its significance.

**1. Left Sidebar – Navigation Panel**

The left sidebar contains **three sections** for easy navigation:

**(A) Analysis Tools – Key Functionalities for Need Assessment**

* **Needs Calculator** – Helps users determine how much life insurance coverage they require.
* **Premium Estimator** – Calculates premium amounts based on user inputs.
* **Coverage Comparison** – Allows users to compare different policy benefits and premium costs.

**(B) Resources – Informational Support for Users**

* **Insurance Glossary** – Provides definitions for insurance-related terms.
* **FAQ** – Answers common user queries about life insurance.
* **Policy Types** – Lists various insurance products available.

|  |  |  |
| --- | --- | --- |
| **Policy Type** | **Purpose** | **Best For** |
| **Term Insurance** | Pure life cover with high sum assured | Family protection, income replacement |
| **Endowment Plan** | Life cover + savings with guaranteed maturity benefits | Secure savings with assured returns |
| **ULIP (Unit-Linked Insurance Plan)** | Life cover + market-linked investment | Wealth creation with flexible investment |
| **Money-Back Policy** | Periodic payouts + life cover | Those needing liquidity during the policy term |
| **Child Plan** | Savings + insurance for child’s future | Education or marriage planning |
| **Retirement/Pension Plan** | Guaranteed income post-retirement | Long-term financial security |

**(C) My Account – User-Specific Features**

* **Saved Analyses** – Stores previous need analysis reports for future reference.
* **My Profile** – Allows users to update personal information.
* **Settings** – Enables customization of chatbot interaction and notifications.

**2. Top Navigation Bar**

* **Session ID: 12234HGF**
  + A unique identifier for each user session, allowing them to resume conversations later.
* **Login/Signup Button** (Top Right Corner)
  + Provides an option for users to log in or create an account for personalized recommendations and saved analyses.

**3. Chatbot Welcome Message (Main Interaction Box)**

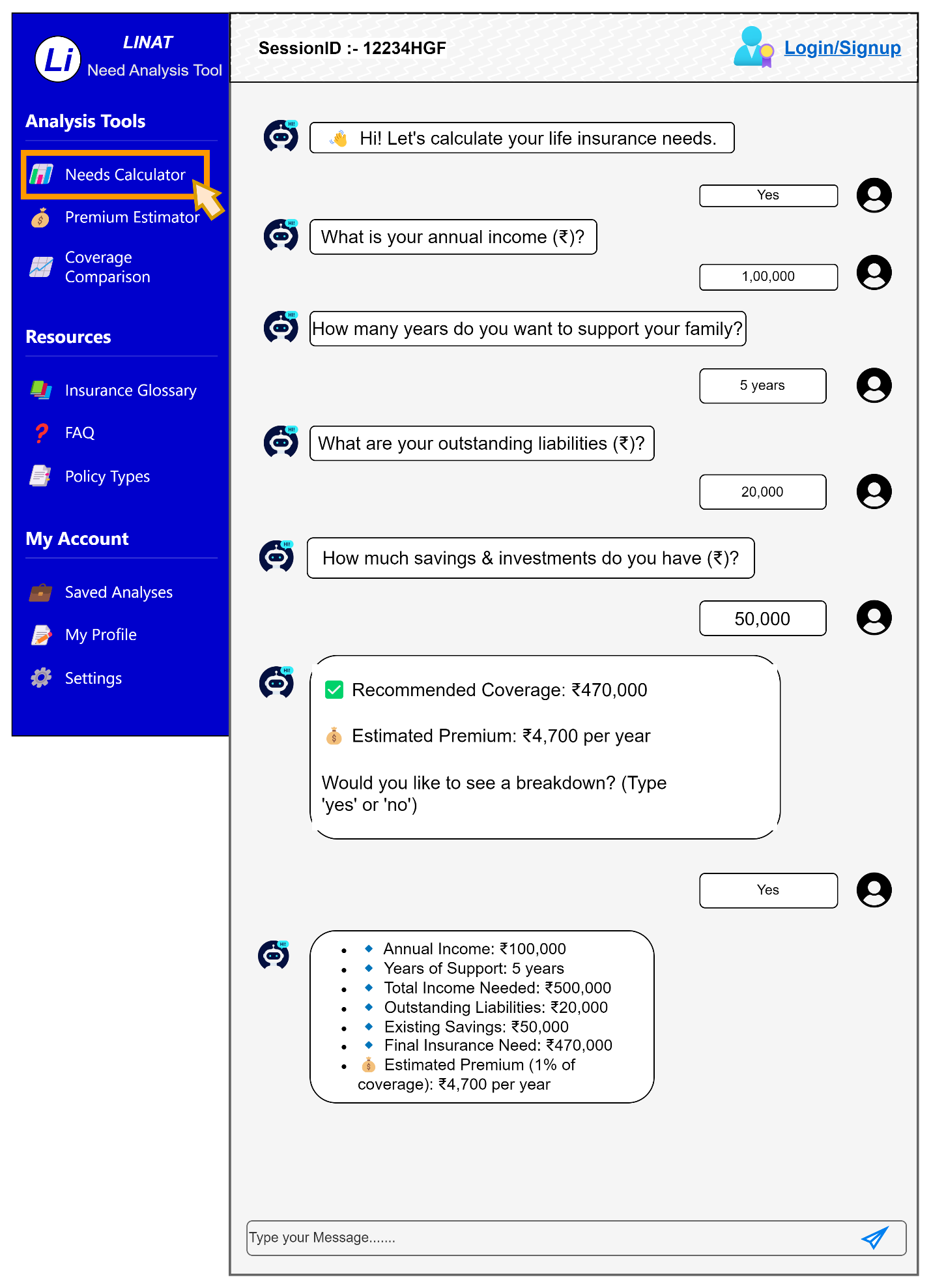
* **Chatbot Greeting:**
  + The AI bot welcomes the user with a personalized message from "[Insurance Company]."
  + It explains that the chatbot is designed to **help find the best life insurance policy** aligned with financial goals.
* **Question to User:**
  + "What’s bringing you here today?" – This establishes user intent by categorizing insurance needs.
* **Response Options (Predefined User Inputs):**
  + **Family Protection** – For users who want to secure their family's financial future in case of unforeseen events.
  + **Retirement Planning** – For users seeking stable income post-retirement.
  + **Wealth Creation** – For users interested in investment-linked insurance products.
  + **Other** – For users with unique needs not covered by the primary options.

These predefined choices **streamline the conversation** by guiding users into structured need-based pathways.

**4. Message Input Box (User Interaction)**

* A text box at the bottom allows users to **type their own queries** instead of selecting predefined options.
* A **send button (paper plane icon)** is available for submission.

## 8.3 WF\_Chatbot\_Interface\_NeedCalculation\_003



Explanation of screen03 –

This screen captures the **Life Insurance Needs Calculation Process** within the LINAT chatbot. It guides users step-by-step through key financial inputs to determine an appropriate coverage amount.

**Left Sidebar – Navigation Highlights**

The **"Needs Calculator"** is selected, highlighted with a yellow box and a cursor, indicating user interaction.

Other available tools:

* **Premium Estimator** – Calculates the cost of different policies.
* **Coverage Comparison** – Compares features and costs of different insurance plans.

**Chatbot-Based Needs Analysis Process**

**Step 1: Chatbot Greeting & Purpose Introduction**

💬 **Chatbot:** *"Hi! Let’s calculate your life insurance needs."*

The bot **sets the context** for the user, ensuring clarity in the process.

**Step 2: Gathering Financial Inputs**

1. **Annual Income (₹)?** – *User inputs ₹1,00,000.*
2. **How many years of family support?** – *User enters 5 years.*
3. **Outstanding liabilities (₹)?** – *User enters ₹20,000.*
4. **Savings & investments (₹)?** – *User enters ₹50,000.*

**Calculated Life Insurance Coverage**

**Recommended Coverage:** ₹4,70,000  
 **Estimated Premium:** ₹4,700 per year *(assuming 1% of coverage as a rough estimate).*

The chatbot **calculates the coverage using this formula:**

Total Income Needed=Annual Income×Years of Support\text{Total Income Needed} = \text{Annual Income} \times \text{Years of Support} Total Income Needed=Annual Income×Years of Support Final Insurance Need=Total Income Needed+Outstanding Liabilities−Existing Savings\text{Final Insurance Need} = \text{Total Income Needed} + \text{Outstanding Liabilities} - \text{Existing Savings} Final Insurance Need=Total Income Needed+Outstanding Liabilities−Existing Savings =(₹1,00,000×5)+₹20,000−₹50,000=₹4,70,000= (₹1,00,000 \times 5) + ₹20,000 - ₹50,000 = ₹4,70,000 =(₹1,00,000×5)+₹20,000−₹50,000=₹4,70,000

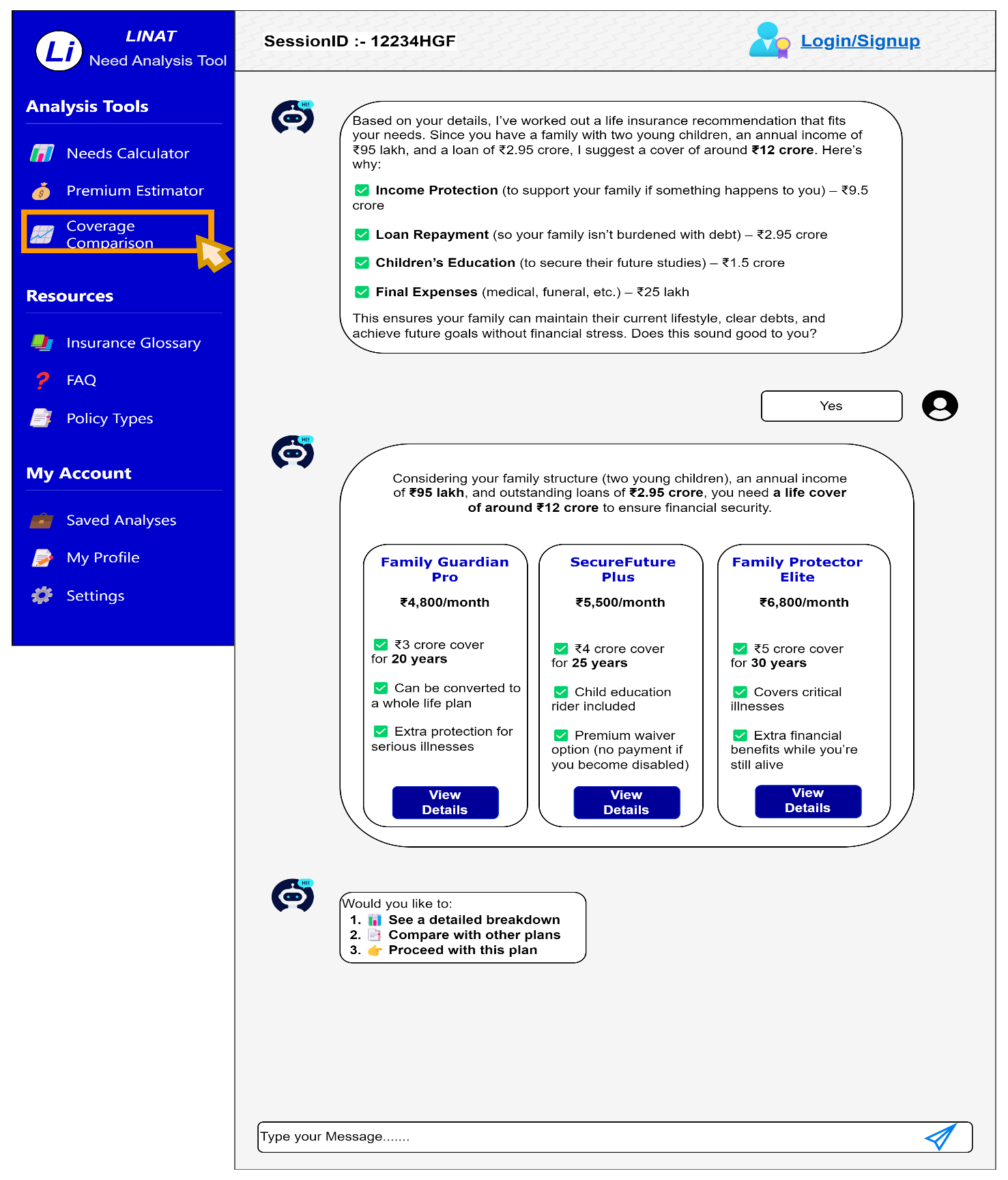
**Detailed Breakdown (Upon User Request)**

When the user types *"Yes"*, the chatbot provides a structured breakdown:

* **Annual Income:** ₹1,00,000
* **Years of Support:** 5
* **Total Income Needed:** ₹5,00,000
* **Outstanding Liabilities:** ₹20,000
* **Existing Savings:** ₹50,000
* **Final Insurance Need:** ₹4,70,000
* **Estimated Premium (1% rule):** ₹4,700 per year

This transparency helps users understand **how coverage is calculated.**

## 8.4 WF\_Chatbot\_Interface\_PlanRecommendation\_and \_CoverageComparsion\_004



**Screen Description: Coverage Comparison**

This screen belongs to the **Coverage Comparison** section of the **LINAT (Life Insurance Need Analysis Tool)**. It helps users compare different life insurance plans based on their unique financial needs and preferences. The AI-powered chatbot provides **a personalized recommendation**, explaining the logic behind the suggested coverage amount and breaking it down into specific financial needs.

**Key UI Elements and Features:**

**1. Chatbot-Based Insurance Recommendation**

* The chatbot **analyzes user data** (income, loans, dependents, and future financial needs) and suggests an appropriate **coverage amount (~₹12 crore)**.
* The coverage breakdown includes:
  + **Income Protection** – ₹9.5 crore
  + **Loan Repayment** – ₹2.95 crore
  + **Children’s Education** – ₹1.5 crore
  + **Final Expenses** – ~₹25 lakh
* The chatbot asks for confirmation (**"Does this sound good to you?"**) with a **"Yes"** button to proceed.

**2. Insurance Plan Comparison Table**

The screen displays a **side-by-side comparison** of three life insurance plans:

|  |  |  |  |
| --- | --- | --- | --- |
| **Plan Name** | **Premium Amount** | **Coverage & Duration** | **Key Features** |
| **Family Guardian Pro** | ₹4,800/month | ₹3 crore for 20 years | Convertible to whole life, extra protection for serious illnesses |
| **SecureFuture Plus** | ₹5,500/month | ₹4 crore for 25 years | Child education rider, premium waiver if disabled |
| **Family Protector Elite** | ₹6,800/month | ₹5 crore for 30 years | Covers critical illnesses, additional financial benefits |

Each plan includes a "**View Details**" button for further exploration.

**3. User Action Options**

Below the plan comparison, the chatbot provides three interactive choices:

1. **See a Detailed Breakdown** – Provides a more in-depth analysis of the selected policy.
2. **Compare with Other Plans** – Allows users to explore alternative insurance plans.
3. **Proceed with This Plan** – Directs the user to initiate the policy purchase process.

**4. Sidebar Navigation (Highlighted Coverage Comparison)**

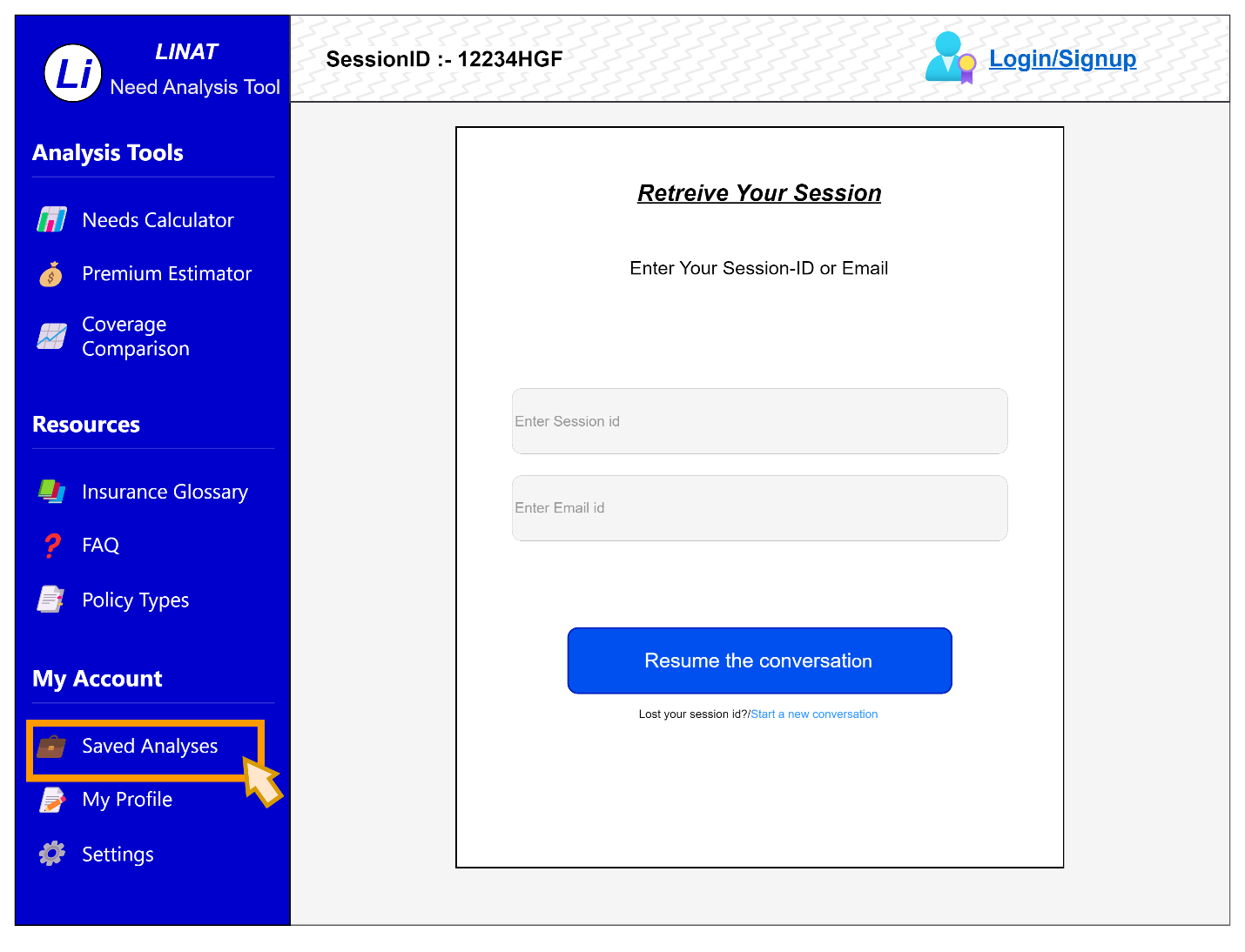
* The **left-side navigation panel** contains different **analysis tools and resources**, including:
  + **Needs Calculator**
  + **Premium Estimator**
  + **Coverage Comparison** (currently highlighted, showing the user is in this section)
  + **Insurance Glossary, FAQs, and Policy Types**
  + **Saved Analyses, Profile, and Settings**

**5. User Input & Chat Functionality**

* A **chat input field** at the bottom allows users to type messages and interact with the chatbot for further assistance.

## 8.5 WF\_Chatbot\_Interface\_SavedAnalysis \_005

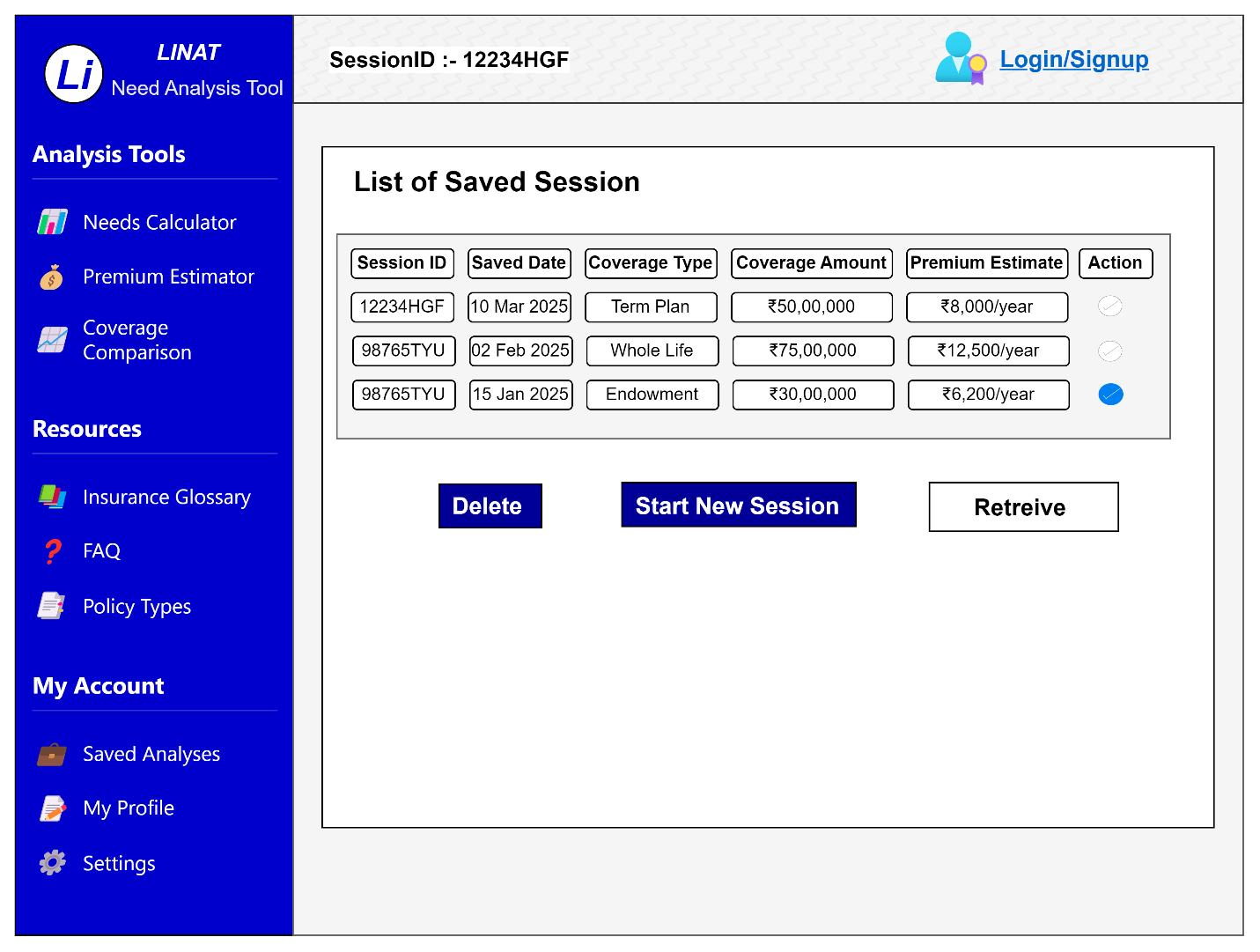
#### If User have saved sessions



**Field Appendix for "Retrieve Your Session" Page**

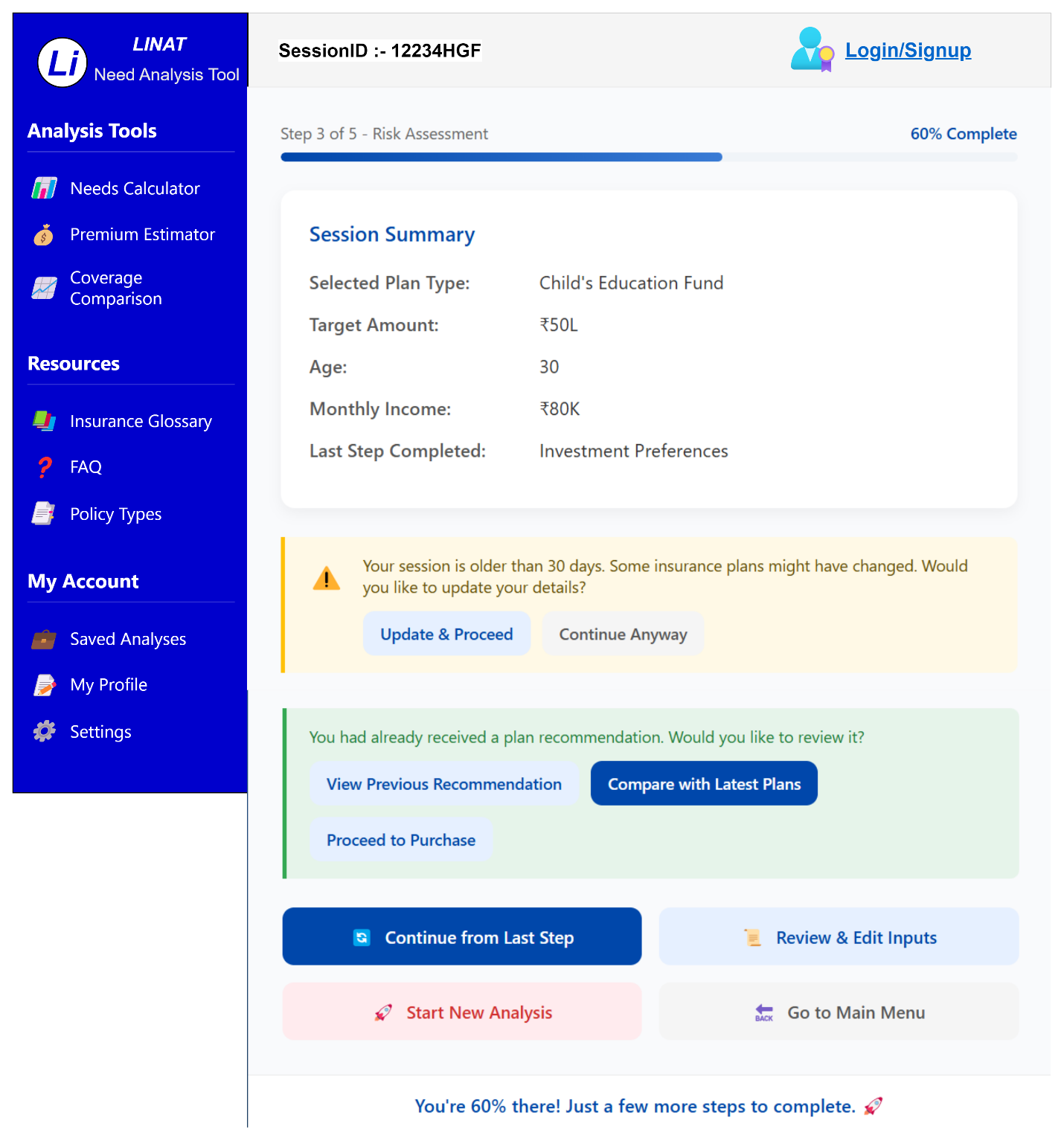
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Data Source** | **Read-only/Editable** | **Validation Rules** | **System Response** |
| **Session ID Input** | Text Field | Alphanumeric String | User Input | Editable | Required, Max Length: 8, No Special Characters. Format: First 5 Numeric + 3 Alphabetical (e.g., 12345ABC). | **If incorrect:** Show error message: "Invalid Session ID. Please try again." |
| **Email ID Input** | Text Field | Email String | User Input | Editable | Required, Valid Email Format (e.g., [user@example.com](mailto:user@example.com)). | **If format is incorrect:** Show error message: "Please enter a valid email address." |
| **Resume Conversation** | Button | Action Trigger | System | Read-only | Enabled only if a valid Session ID or Email is entered. | **If both fields are empty:** Disable "Resume Conversation" button. |
| **Start a New Conversation** | Hyperlink | URL String | System | Read-only | No validation required. | **On click:** Redirect to the first step of LINAT analysis. |
| **Session ID Display** | Label | Alphanumeric String | System-generated | Read-only | Auto-updated when a session is active. | No specific error handling. |
| **Error Message** | Text Alert | String | System | Read-only | Displays error messages as needed. | **Triggers if incorrect input is detected.** |
| **Session Expiry Alert** | UI Notification | String | System | Read-only | Shown if session is older than 30 days. | **Warning message:** "Your session may be outdated. Update details?" |
| **Back to Dashboard** | Button | Action Trigger | System | Read-only | No validation required. | Redirects user to the LINAT home screen. |

#### History of saved sessions



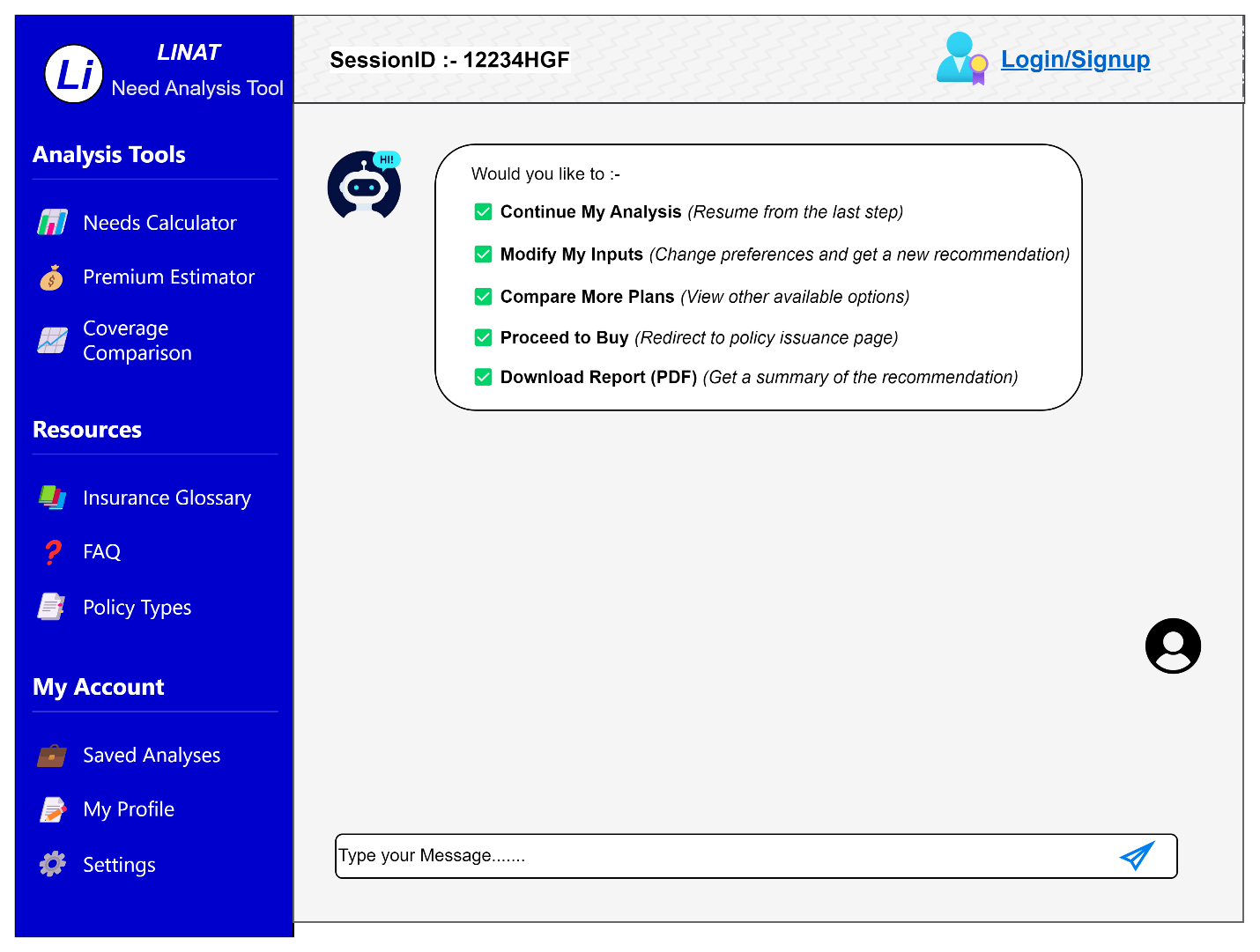
| **Field Name** | **Field Type** | **Data Type** | **Data Source** | **Read-only/Editable** | **Validation** |
| --- | --- | --- | --- | --- | --- |
| Session ID | Input Box | Alphanumeric | User Input | Auto-generated | Required, Unique, 8-12 characters |
| Data Saved | Date Field | Date | System Generated | Read-only | Valid Date Format (DD MMM YYYY) |
| Coverage Type | Dropdown | String | Predefined List | Editable | Required |
| Coverage Amount | Input Box | Currency (₹) | User Input | Editable | Required, Numeric Only |
| Premium Estimate | Input Box | Currency (₹) | System Calculated | Read-only | Positive Number Only |
| Action | Radio Button | Boolean | User Selection | Editable | One Selection Mandatory |
| Enter Session ID | Input Box | Alphanumeric | User Input | Editable | Optional, Match Existing Session |
| Enter Email ID | Input Box | Email | User Input | Editable | Valid Email Format |
| Resume Conversation | Button | Action Trigger | User Click | Read-only | Enabled when valid session/email |
| Delete | Button | Action Trigger | User Click | Read-only | Enabled when session is selected |
| Start New Session | Button | Action Trigger | User Click | Read-only | Always Enabled |
| Retrieve | Button | Action Trigger | User Click | Read-only | Enabled when session is selected |

**Screen for retrieving past history**



**Field Appendix for "Resume Need Analysis" Page**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Data Source** | **Read-only/Editable** | **Validation Rules** | **System Response** |
| Session ID Display | Label | Alphanumeric String | System-generated | Read-only | Auto-generated based on session retrieval. | No specific error handling. |
| Progress Bar | UI Component | Percentage | System-generated | Read-only | Displays session progress dynamically. | Updates based on completed steps. |
| Selected Plan Type | Label | String | System | Read-only | Displays plan type from retrieved session. | No specific error handling. |
| Target Amount | Label | Currency (INR) | User Input | Read-only | Value should be formatted in ₹ (e.g., ₹50L). | No specific error handling. |
| Age | Label | Numeric (Years) | User Input | Read-only | Must be between 18-99 years. | No specific error handling. |
| Monthly Income | Label | Currency (INR) | User Input | Read-only | Value should be formatted in ₹ (e.g., ₹80K). | No specific error handling. |
| Last Step Completed | Label | String | System-generated | Read-only | Auto-updated to show the last completed step. | No specific error handling. |
| Session Expiry Warning | UI Notification | String | System | Read-only | Triggered if session is older than 30 days. | Shows warning: "Your session is older than 30 days. Some insurance plans might have changed. Would you like to update your details?" |
| Update & Proceed | Button | Action Trigger | System | Read-only | Enabled when session is outdated (older than 30 days). | Redirects to edit session details. |
| Continue Anyway | Button | Action Trigger | System | Read-only | No validation required. | Proceeds with existing session data without updating. |
| View Previous Recommendation | Button | Action Trigger | System | Read-only | Enabled if a previous plan recommendation exists. | Shows last recommended plan. |
| Compare with Latest Plans | Button | Action Trigger | System | Read-only | Enabled if a previous recommendation exists. | Fetches latest insurance plans for comparison. |
| Proceed to Purchase | Button | Action Trigger | System | Read-only | Enabled if recommendation exists. | Redirects to policy purchase process. |
| Continue from Last Step | Button | Action Trigger | System | Read-only | Enabled if session exists. | Resumes from the last completed step. |
| Review & Edit Inputs | Button | Action Trigger | System | Read-only | Enabled if session exists. | Allows users to modify previously entered details. |
| Start New Analysis | Button | Action Trigger | System | Read-only | No validation required. | Clears session data and restarts LINAT. |
| Go to Main Menu | Button | Action Trigger | System | Read-only | No validation required. | Redirects user to LINAT home screen. |
| Completion Message | UI Notification | String | System-generated | Read-only | Displays motivational progress message. | Example: "You're 60% there! Just a few more steps to complete. 🚀" |

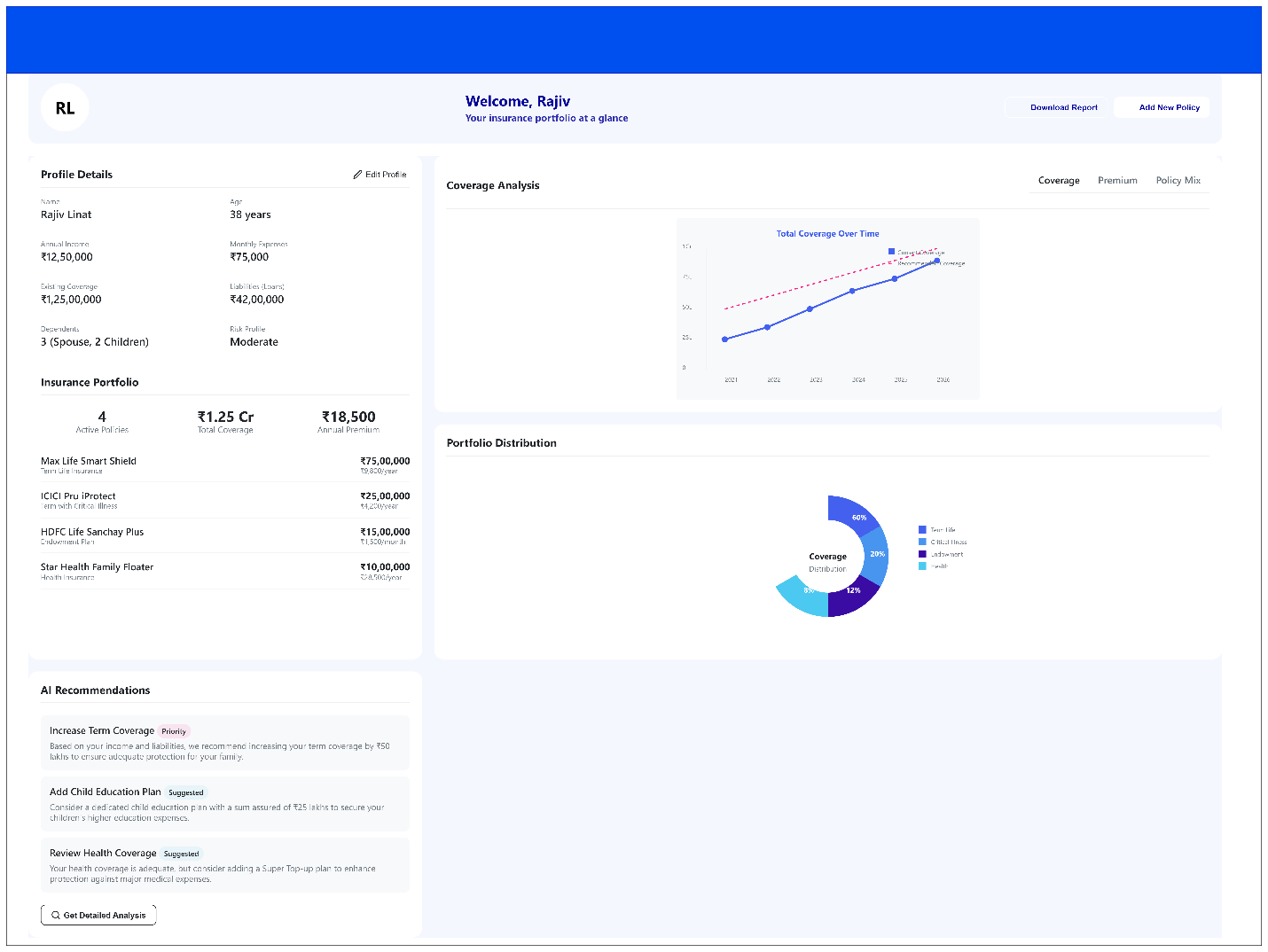


#### If the user has no saved session

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Data Source** | **Read-only/Editable** | **Validation** |
| **No Saved Session Message** | Text | String | System-Generated | Read-only | Auto-Generated |
| **Start New Conversation Button** | Button | Action Trigger | System Action | Read-only | Starts a New Session (Route you back to the chatbot) |

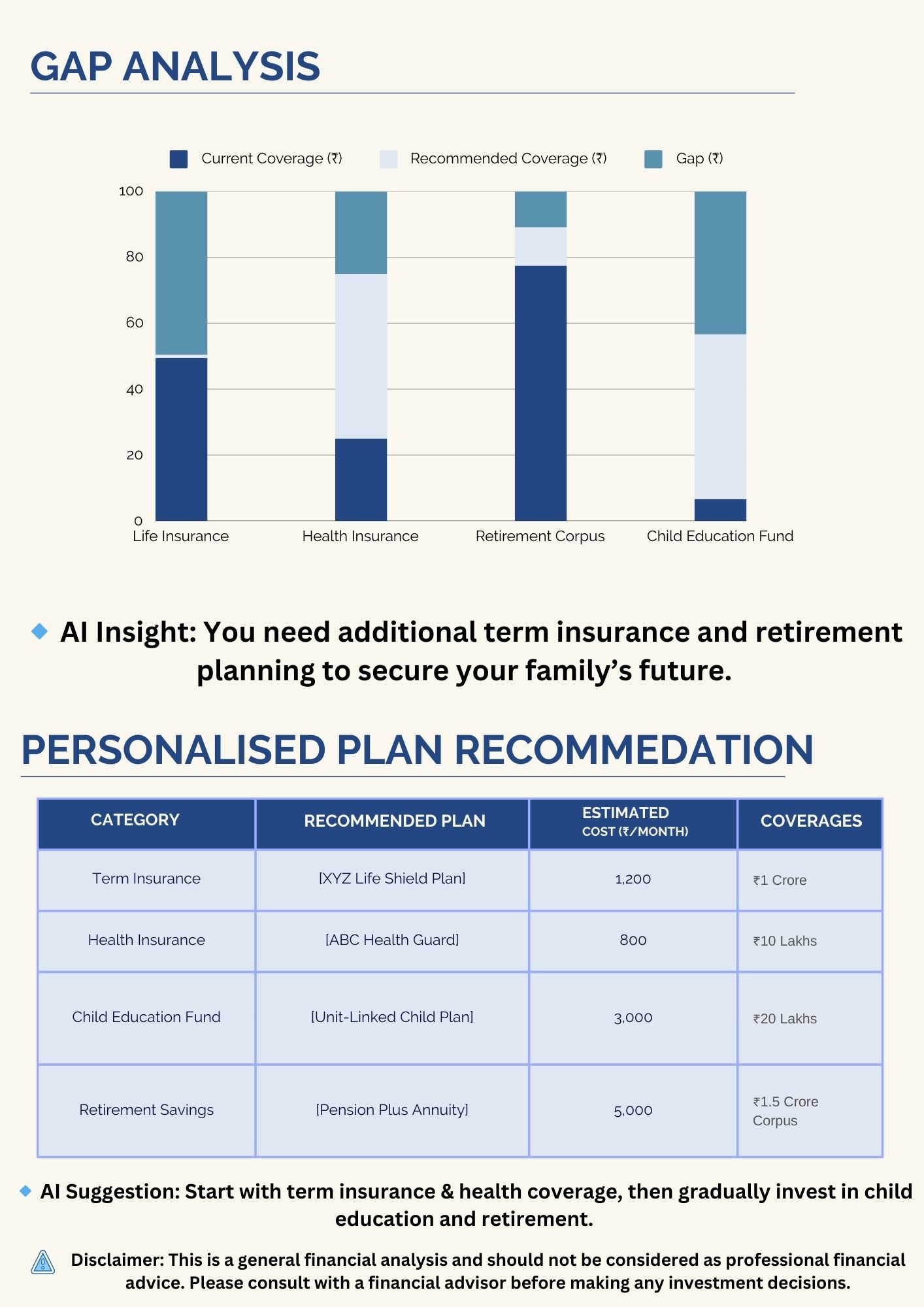
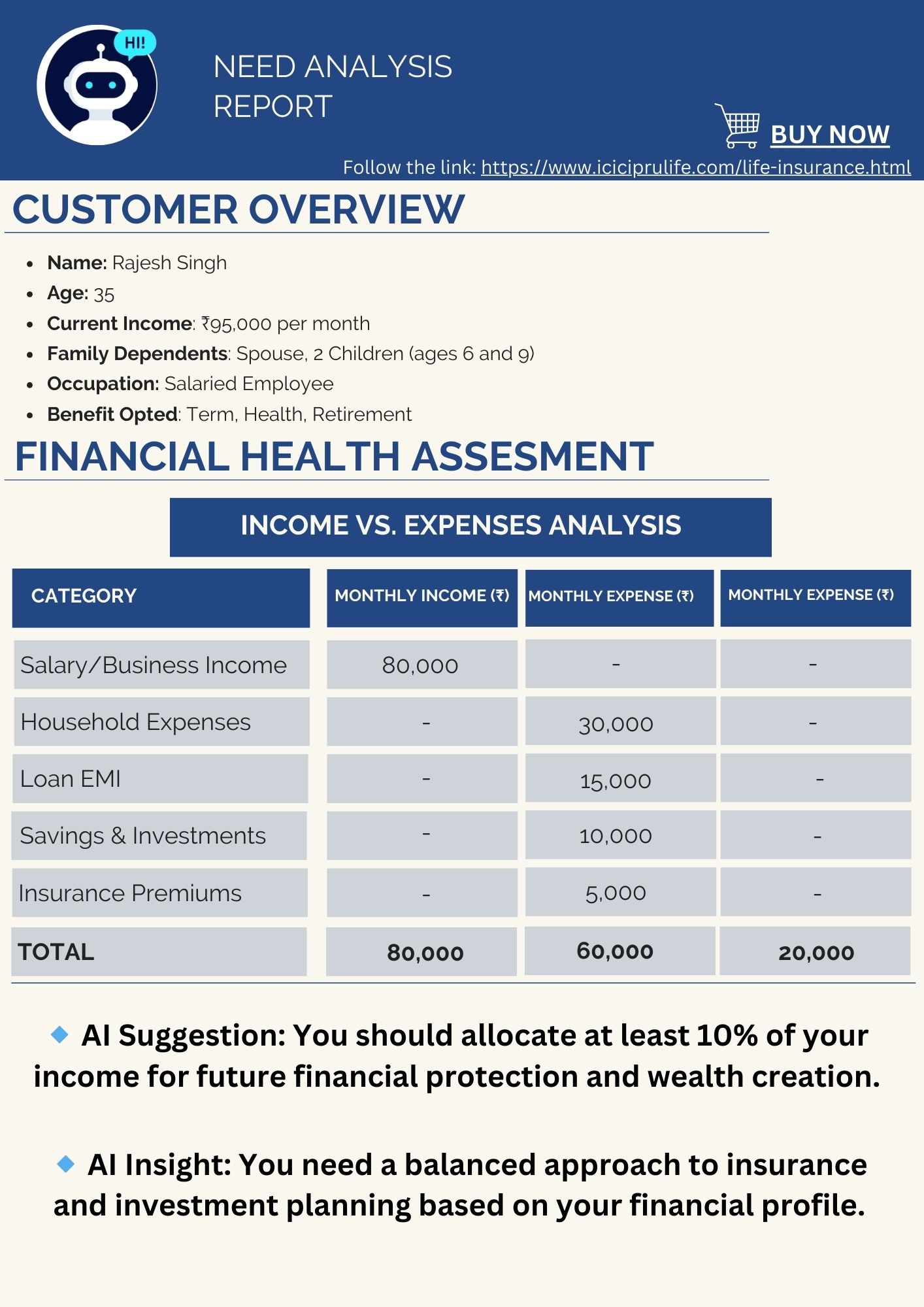
## 8.6 WF\_User Profile Creation\_006

**Field Description**: When the user selects the My Profile Icon will be directed to This Dashboard been generated



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Data Source** | **Read-only/Editable** | **Validation** |
| Name | Text | String | User Input/Profile Data | Editable | Required, Max 50 characters |
| Age | Numeric | Integer | User Input/Profile Data | Read-only | Range: 18-100 |
| Annual Income | Currency | Float | User Input/Profile Data | Editable | Min ₹1,00,000 |
| Monthly Expenses | Currency | Float | User Input/Profile Data | Editable | Min ₹5,000 |
| Existing Coverage | Currency | Float | Policy Database | Read-only | Min ₹0 |
| Liabilities (Loan) | Currency | Float | User Input/Profile Data | Editable | Min ₹0 |
| Dependents | Numeric | Integer | User Input/Profile Data | Editable | Range: 0-10 |
| Risk Profile | Dropdown | String | AI Risk Assessment | Read-only | Options: Low, Moderate, High |
| Total Coverage | Currency | Float | Policy Database | Read-only | Min ₹0 |
| Annual Premium | Currency | Float | Policy Database | Read-only | Min ₹0 |
| Insurance Portfolio (Policies) | List (Table) | Object | Policy Database | Read-only | Min 1 Policy Required |
| Coverage Analysis Chart | Visualization | Graph | System-Generated Data | Read-only | Auto-Generated |
| Portfolio Distribution Chart | Visualization | Pie Chart | System-Generated Data | Read-only | Auto-Generated |
| AI Recommendations | List (Text) | String | AI Model | Read-only | Based on User Data |
| Download Report Button | Button | Action Trigger | System Action | Read-only | Click to Download |
| Add New Policy Button | Button | Action Trigger | System Action | Read-only | Redirect to Policy Page |

## 8.7 Templet for gap report generated



**PDF Download – Gap Report Generation**

**1. Overview**

The PDF download feature in LINAT allows users to generate and download a **Gap Report**, summarizing their insurance needs, financial gaps, policy recommendations, and comparative insights. This ensures users have a structured and easily accessible report for decision-making.

**2. Trigger Points for PDF Generation**

The Gap Report PDF is generated when a user:

* Completes the need analysis and requests a summary.
* Clicks on the **"Download PDF"** button on the chatbot interface.
* Requests an email copy of their report (in which case, a downloadable link is provided via email).

**3. Content Structure of the Gap Report**

The PDF report is structured as follows:

**A. Cover Page**

* **Report Title:** "Insurance Gap Analysis Report"
* **User Name & Session ID** (if applicable)
* **Date & Time of Report Generation**
* **Company Branding & Contact Details**

**B. User Profile & Inputs**

* **Age, Income, Dependents**
* **Financial Goals (Retirement, Child’s Education, Loan Protection, etc.)**
* **Current Insurance Coverage**
* **Policy Preferences (Sum Assured, Tenure, Riders, etc.)**

**C. Financial Gap Analysis**

* **Current Financial Status vs. Future Needs**
* **Coverage Shortfall Calculation**
* **Risk Factor Analysis**
* **Recommended Sum Assured Based on AI Evaluation**

**D. Policy Recommendations**

* **AI-Suggested Insurance Plans**
* **Premium Estimations**
* **Comparison of Available Policies**
* **Graphical Representation of Coverage vs. Needs**

**E. Premium Breakdown & Affordability**

* **Monthly vs. Annual Premium Cost**
* **Rider Costs & Additional Charges**
* **Tax Benefits Overview**
* **Customizable Plan Options**

**F. Next Steps**

* **"Proceed to Buy"** with a clickable link redirecting users to the insurer’s live plan.
* **Advisor Assistance Contact Information**
* **Save for Future Reference (with Email Link Option)**

**4. PDF Generation Process & Technology Stack**

1. **User Triggers Report Generation**
   * User selects "Download PDF" in the chatbot interface.
   * System validates session data and fetches stored responses.
2. **Data Processing & Formatting**
   * User inputs, need analysis results, and policy recommendations are compiled.
   * Structured layout is formatted using **HTML to PDF conversion (wkhtmltopdf, Puppeteer, ReportLab, or jsPDF)**.
3. **PDF Styling & Branding**
   * Company logo, fonts, and layout styling applied.
   * Tables, charts, and summary sections formatted using CSS and PDF libraries.
4. **Secure File Generation & Storage**
   * PDF is generated and temporarily stored in a secure server.
   * User is provided with a direct **download link** (valid for a limited time).
5. **Email PDF Option (If Requested)**
   * PDF is attached to an email and sent via **SendGrid, AWS SES, or SMTP API**.

**5. Security & Compliance Measures**

* **Data Encryption**: PDF files containing sensitive financial data are encrypted before storage.
* **Access Control**: Only authenticated users can download the report.
* **Session-Based Download Links**: Links expire after a set period to prevent unauthorized access.
* **Compliance**: Ensures GDPR & DPDP Act compliance for user data privacy.

**6. Error Handling & User Notifications**

* If PDF generation fails: User sees an error message with retry option.
* If download link expires: System prompts user to regenerate the report.

## 8.8 WF\_Admin Login Page



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Placeholder/Label** | **Validation Rules** | **Error Message** |
| **Email/Username** | Text Input | "Enter your email or username" | Required, Valid email format (if email) | "Please enter a valid email or username." |
| **Password** | Password Input | "Enter your password" | Required, Min. 8 characters, Strong password (includes letters, numbers, special characters) | "Password must be at least 8 characters long." |
| **Show/Hide Icon** | Button/Toggle | Eye Icon | Toggles password visibility | N/A |
| **Login Button** | Button (Primary) | "Login" | Disabled if fields are empty | N/A |
| **Forgot Password?** | Link/Button | "Forgot Password?" | Redirects to password reset page | N/A |
| **Need Help?** | Link/Button | "Need Help?" | Opens customer support/contact form | N/A |

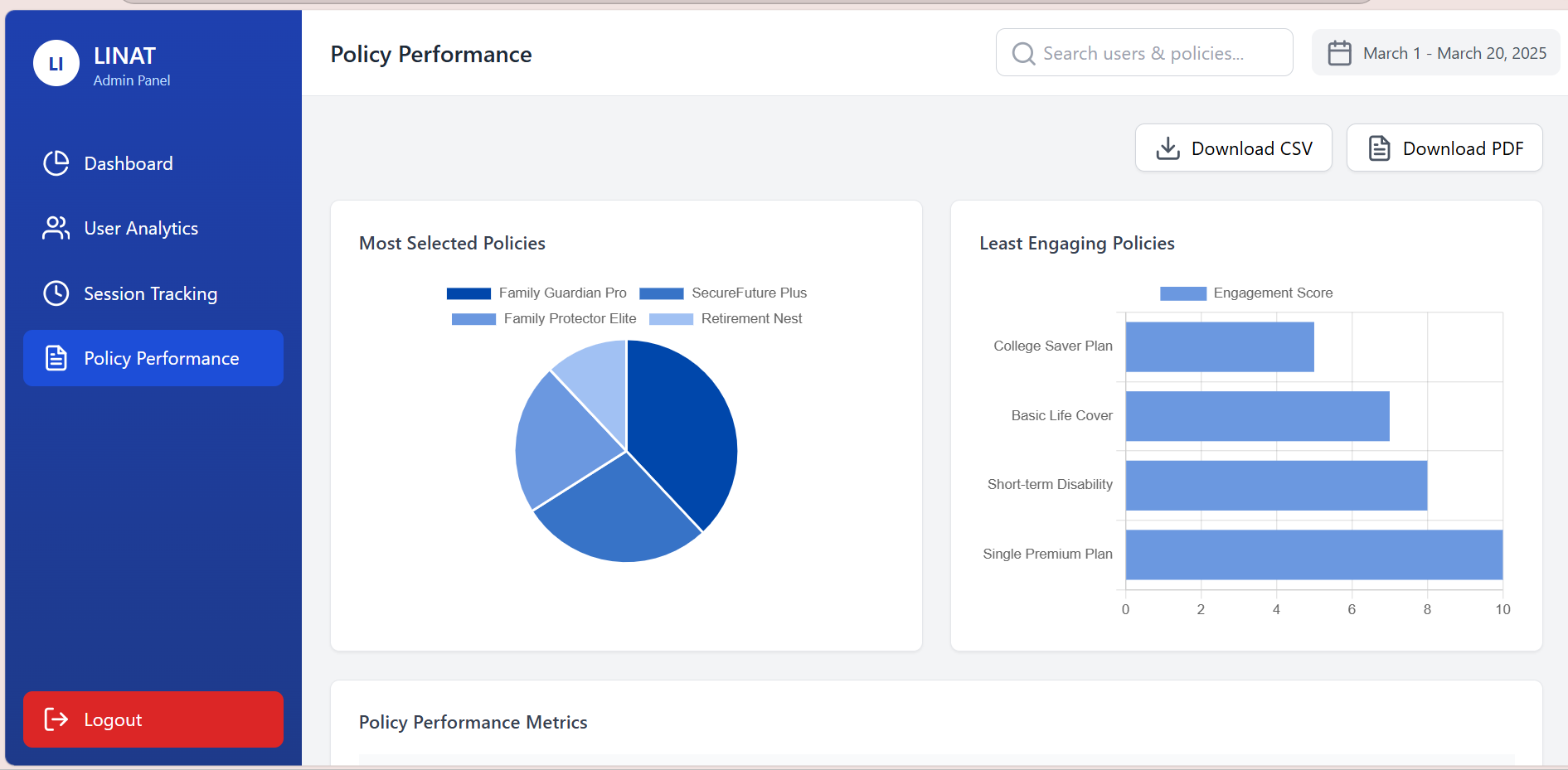
**Additional Notes:**

* If login fails due to invalid credentials, display an **error banner**:

**"Invalid email or password. Please try again."**

* If the user **forgets the password**, they should be redirected to the **password recovery page**.

## 8.9 WF\_Admin Dashboard



Screen Reference – <file:///C:/Users/hp/Downloads/admin-dashboard-html.html>

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Page/Tab** | **Field Name** | **Type** | **Placeholder/Label** | **Validation Rules** | **Field Type** | **Data Type** | **Data Source** | **Read-only/Editable** |
| **Dashboard** | total\_sessions\_today | Number | Total Sessions Today | None | Text | Integer | Backend API | Read-only |
|  | active\_users\_now | Number | Active Users Now | None | Text | Integer | Backend API | Read-only |
|  | policy\_selection\_rate | Number | Policy Selection Rate | None | Text | Float | Backend API | Read-only |
| **User Analytics** | user\_demographics | Chart | User Demographics | None | Chart | JSON | Backend API | Read-only |
|  | user\_engagement | Chart | User Engagement | None | Chart | JSON | Backend API | Read-only |
| **Session Tracking** | session\_duration | Chart | Session Duration | None | Chart | JSON | Backend API | Read-only |
|  | bounce\_rate | Number | Bounce Rate | None | Text | Float | Backend API | Read-only |
|  | conversion\_rate | Number | Conversion Rate | None | Text | Float | Backend API | Read-only |
| **Policy Performance** | most\_selected\_policies | Chart | Most Selected Policies | None | Chart | JSON | Backend API | Read-only |
|  | least\_engaging\_policies | Chart | Least Engaging Policies | None | Chart | JSON | Backend API | Read-only |
|  | policy\_performance\_metrics | Table | Policy Performance Metrics | None | Table | JSON | Backend API | Read-only |
| **Recent Activities** | user | Text | User | None | Text | String | Backend API | Read-only |
|  | action | Text | Action | None | Text | String | Backend API | Read-only |
|  | time | Text | Time | None | Text | String | Backend API | Read-only |
| **Filters** | date\_filter | Date | Select Date Range | None | Date Picker | Date | User Input | Editable |
| **Download Options** | download\_csv | Button | Download CSV | None | Button | Action | N/A | Editable |
|  | download\_pdf | Button | Download PDF | None | Button | Action | N/A | Editable |
| **Email List** | recent\_user\_emails | Text | Recent User Emails | None | Text | String | Backend API | Read-on |

**Screen: Life Insurance Need Analysis Dashboard**

**1. When is this screen triggered?**

* This screen is **triggered immediately after the user logs in** to the **Life Insurance Need Analysis Tool (LINAT)**.
* The system validates the user credentials and, upon successful authentication, redirects them to the **dashboard screen**.
* If the login fails, the user remains on the login page and receives an error message.

**2. What does the user see first after logging in?**

Once authenticated, the user is taken to the **Dashboard screen**, where they see the following:

**A. Header Section (Top Navigation Bar)**

* **Welcome Message**: A greeting message such as *"Welcome, [User Name]!"*
* **Date Filter**: Allows users to select a custom date range for viewing reports.
* **Download Options**:
  + **Download CSV**: Exports the displayed data in CSV format.
  + **Download PDF**: Exports the displayed data in PDF format.

**B. Dashboard Section (Main Analytics Overview)**

This section provides a quick summary of key metrics related to user engagement and policy selection:

* **Total Sessions Today** – Number of user sessions on the platform today.
* **Active Users Now** – Number of users currently active on the system.
* **Policy Selection Rate** – Percentage of users selecting an insurance policy after need analysis.

**C. User Analytics Section**

* **User Demographics** (Chart): Displays user distribution based on age, gender, location, or other demographics.
* **User Engagement** (Chart): Shows user activity trends, such as session duration, page views, and interaction frequency.

**D. Session Tracking Section**

* **Session Duration** (Chart): Displays the average session length per user.
* **Bounce Rate** (Percentage): Shows how many users leave the platform without completing an analysis.
* **Conversion Rate** (Percentage): Indicates the percentage of users who complete the need analysis and proceed with a policy recommendation.

**E. Policy Performance Section**

* **Most Selected Policies** (Chart): Displays the top policies chosen by users.
* **Least Engaging Policies** (Chart): Lists policies that receive the least engagement.
* **Policy Performance Metrics** (Table): Detailed breakdown of policy performance, including selection rates, drop-off points, and feedback metrics.

**F. Recent Activities Section**

* **User Activity Log**: Displays recent interactions, such as:
  + User name
  + Actions performed (e.g., completed need analysis, viewed policy details)
  + Timestamp of the action

**G. Email List Section**

* Shows a **list of recent user emails** who have interacted with the system.

1. **3. User Actions on This Screen**

* **View Reports**: Users can analyze different metrics related to life insurance need analysis.
* **Filter Data**: Select a date range to modify report views.
* **Download Reports**: Export data as CSV or PDF.
* **Track User Behavior**: Check engagement patterns and conversion rates.

1. **4. Summary of User Journey**
2. **User logs in** → Redirected to the **Dashboard**.
3. **First view**: Overview of sessions, engagement, and policy selection.
4. **User interacts** with charts and reports.
5. **User applies filters** (if needed) to refine data.

**User downloads** reports if required

# Appendix B – Requirements

## 9.1 Functional Requirement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Functionality** | **Criticality** | **Response Type** | **Comments** |
| **Modular Design** | The system will be modular, component-based, and service-oriented to support scalability and reusability. | **Vital** | C/ER/F/CR/NC | Ensures future scalability and ease of maintenance. |
| **System Architecture** | The architecture will be designed for seamless integration with third-party insurers. | **Vital** | C/ER | Supports industry standards and open architecture. |
| **AI-Powered Chatbot** | LINAT will use conversational AI to guide users through insurance selection, leveraging NLP and RAG-based retrieval models. | **Vital** | F | Enhances user experience through dynamic, intelligent responses. |
| **Web-Based Design** | The entire application will be web-based, accessible through modern browsers (Chrome, Firefox, Edge, Safari). | **Vital** | ER | Ensures cross-platform accessibility. |
| **User Intimation & Greeting** | The chatbot will greet users, detect returning customers, and personalize interactions based on previous conversations. | **Essential** | F | Improves customer engagement and user experience. |
| **User Profile & Data Collection** | Collects key user details such as age, income, financial goals, dependents, and liabilities. | **Vital** | ER/F | Required for accurate need analysis. |
| **Data Validation & Error Handling** | AI will validate user inputs, check for missing/inconsistent data, and prompt users for corrections. | **Vital** | ER | Ensures accuracy and prevents invalid calculations. |
| **AI-Powered Need Analysis** | Uses financial modeling to determine ideal coverage and insurance type based on user inputs. | **Vital** | F | Leverages actuarial formulas and user profiles for recommendations. |
| **Plan Matching & Recommendations** | AI recommends suitable policies based on user risk appetite, coverage needs, and financial capability. | **Vital** | F | Ensures personalized policy selection. |
| **Premium Estimator** | Calculates estimated premium based on sum assured, policy term, and risk factors. | **Essential** | ER | Helps users understand affordability before purchasing. |
| **Coverage Comparison** | Compares insurance plans from different providers with side-by-side feature analysis. | **Essential** | F | Enables informed decision-making. |
| **Customization of Plans** | Users can adjust sum assured, riders, tenure, and other parameters before finalizing a plan. | **Essential** | ER/F | Enhances flexibility in choosing policies. |
| **Policy Issuance & Redirection** | If the user chooses to purchase, they are redirected to the insurer’s issuance system with pre-filled details. | **Vital** | ER | Ensures seamless transition to the insurer’s platform. |
| **Session Tracking & Retrieval** | Saves user progress with a session ID, allowing them to return later and continue. | **Essential** | ER/F | Improves continuity for users who leave mid-process. |
| **Knowledge Base & FAQs (RAG-Based Search)** | Allows users to search for policy details, insurance terms, and regulatory information. | **Essential** | F | Enhances self-service capabilities. |
| **Tax Planning Optimization** | Suggests policies that offer tax benefits under **Section 80C, 80D, and 10(10D)**. | **Essential** | F | Helps users optimize tax-saving opportunities. |
| **Lead Management & Advisor Integration** | Users can request a call with an advisor for policy guidance. | **Essential** | ER | Supports direct interaction for complex cases. |
| **AI Model Learning & User Behavior Analysis** | The system improves recommendations by learning from past user interactions. | **Essential** | ER | Uses machine learning to refine future responses. |
| **User Dashboard** | Displays past interactions, saved policies, and profile information. | **Essential** | ER/F | Allows users to track their policy interests. |
| **Security & Compliance** | Implements data encryption, role-based access control, and compliance with **IRDAI, GDPR, and HIPAA**. | **Vital** | C/ER | Ensures regulatory compliance and data privacy. |
| **Multi-Language Support (Future Scope)** | Future versions will support additional languages beyond English. | **Desirable** | F | Expands accessibility in diverse markets. |
| **Mobile-Optimized UI** | Ensures that the chatbot and dashboard are fully responsive for mobile users. | **Essential** | ER | Improves accessibility and usability. |
| **Data Storage & Backup** | Secure database infrastructure to store policy details and user profiles with automated backups. | **Vital** | ER | Prevents data loss and ensures high availability. |
| **Fraud Detection & Risk Monitoring** | AI monitors inconsistencies in user inputs to detect potential fraud risks. | **Essential** | ER | Helps flag suspicious activity. |
| **Automated Reporting & Insights** | Generates downloadable need analysis reports and premium estimates. | **Essential** | ER/F | Provides users with structured financial insights. |
| **Performance Monitoring & Scalability** | Ensures the system can handle large user volumes with real-time response optimization. | **Essential** | ER | Supports future expansion and high availability. |
| **MIS Reports & Business Intelligence Dashboard** | Admins can generate reports on user behavior, plan preferences, and system performance. | **Essential** | ER | Supports data-driven decision-making. |
| **Regulatory Updates & Compliance Checks** | AI keeps track of new insurance regulations and ensures compliance in recommendations. | **Vital** | ER | Prevents outdated policy suggestions. |
| **System Admin Panel** | Admins can configure chatbot workflows, update knowledge bases, and manage insurer API integrations. | **Essential** | ER | Provides backend control over chatbot behavior. |

##### Explanation of Table Columns

* **Requirement ID (SF.)** → Unique identifier for each system functionality.
* **Feature** → The name of the functionality.
* **Functionality** → A brief explanation of what it does.
* **Criticality** → Indicates whether the feature is **Vital, Essential, or Desirable**.
* **Response Type** → Defines whether the system will handle this as **Core (C), Error Handling (ER), Functionality (F), Compliance Requirement (CR), or Non-Critical (NC)**.
* **Comments** → Additional remarks or future enhancement possibilities.

### 9.1.1User Interaction & Engagement Functionalities in LINAT

The success of **LINAT (Life Insurance Need Analysis Tool)** relies heavily on **how effectively it interacts with users**. This section details how the chatbot ensures a **smooth, personalized, and engaging experience** for users.

#### 9.1.1.a Conversational AI Chatbot

**Description:**  
The chatbot serves as the primary **user interface** for interacting with the system. It is designed to **engage users in a natural conversation**, ask relevant questions, process responses, and guide them through the **insurance selection process**.

**Example:**

* **Chatbot:** "Welcome! I’m LINAT, your personal insurance assistant. I can help you find the best life insurance plan based on your financial goals. Would you like to start by analyzing your insurance needs?"
* **User:** "Yes, let’s begin."

**Implementation Considerations:**  
 Uses **NLP (Natural Language Processing)** to understand free-text responses.  
 Provides **predefined options** for quicker user input.  
 Ensures that **users don’t feel overwhelmed** with too many questions at once.

#### 9.1.1.b Multi-Turn Conversations

**Description:**  
Rather than presenting a long, static form, LINAT engages users in a **step-by-step conversation**, refining inputs based on previous responses.

**Example:**

* **Chatbot:** "Do you have any existing life insurance policies?"
* **User:** "Yes, I have a term plan for ₹50 lakh."
* **Chatbot:** "Great! Would you like to consider increasing your coverage for better protection?"

**Why This is Important:**  
 Allows users to **think about their responses** without feeling rushed.  
 Adjusts the **flow dynamically**, skipping irrelevant questions based on user answers.  
 Helps **improve accuracy** by clarifying information before making recommendations.

#### 9.1.1.c Interactive UI Elements

**Description:**  
Instead of relying on **text-based responses**, LINAT incorporates **various UI elements** to make input easier and more engaging.

**Types of UI Elements Used:**  
 **Buttons & Quick Reply Options** – Users can tap on predefined choices rather than typing responses.  
 **Sliders** – Used for selecting values such as **coverage amount, premium affordability, or risk level**.  
 **Dropdown Menus** – Allows users to **pick from multiple predefined options**.  
 **Text Fields** – Enables users to **input custom responses** when needed.

**Example:**

* **Chatbot:** "How much coverage do you need?"
  + [ ₹25 Lakh] [ ₹50 Lakh] [ ₹1 Crore] [ Custom Amount]

**Benefits:**  
 **Speeds up the user journey** by reducing the need for manual input.  
 **Minimizes errors** as users select from structured options.  
 **Enhances the chatbot experience** by making interactions more intuitive.

#### 9.1.1.d User Personalization & Context Awareness

**Description:**  
LINAT remembers user preferences and adjusts the **conversation flow** based on past interactions. If a user has previously interacted with the chatbot, it **personalizes the experience** to avoid redundant questions.

**Example:**

* **Returning User:**
  + **Chatbot:** "Welcome back, Ankit! Last time, you were exploring plans for child education. Would you like to continue from where you left off or start fresh?"
* **New User:**
  + **Chatbot:** "Welcome to LINAT! Let’s get started with your insurance analysis."

**Implementation Considerations:**  
 Uses **session tracking** to recognize returning users.  
 **Saves key user preferences** such as preferred coverage type, investment horizon, and selected plans.  
 **Reduces redundant questions** for a seamless experience.

#### 9.1.1.e Session Tracking & Retrieval

**Description:**  
If a user **leaves midway**, LINAT ensures they can **resume the conversation** from where they left off instead of restarting the process.

**Example:**  
1️ User starts the insurance needs analysis but **exits before completing it**.  
2️ The chatbot **generates a session ID** and offers to save progress.  
3️ When the user returns, LINAT **retrieves the last session** using an email or session ID.

**Example Chatbot Interaction:**

* **Chatbot:** "I see that you started an analysis session earlier. Would you like to resume where you left off?"
  + [ Resume My Analysis] [ Start a New Analysis]

**Why This Matters:**  
 Users do not have to **repeat steps unnecessarily**.  
 Increases **completion rates** for insurance selection.  
 Allows users to take their time **without losing progress**.

#### 9.1.1.f Greeting & User Intimation

**Description:**  
The **first interaction** sets the tone for the user experience. LINAT greets users with a **friendly, informative introduction** and **guides them towards selecting the right insurance option**.

**Example Chatbot Interaction:**

* **Chatbot:** "Welcome to [Insurance Company]! I am LINAT, your AI-powered insurance assistant. I can help you choose the best insurance plan that fits your financial goals. What would you like to do today?"
  + [ Analyze My Insurance Needs]
  + [ Compare Plans]
  + [ Retrieve My Saved Session]

**Why This is Important:**  
 **Engages users immediately**, reducing drop-off rates.  
 **Clarifies chatbot capabilities**, helping users navigate options easily.  
 **Encourages interaction**, making the process feel effortless.

### 9.1.2 Data Collection & User Profile Management Functionalities in LINAT

LINAT gathers and manages user financial data and preferences to provide accurate insurance recommendations. The system must validate inputs, resolve missing or conflicting data, and allow users to retrieve and manage their profile details.

#### 9.1.2.a Collecting Financial Data

Description:

The chatbot asks structured questions to gather essential financial information from users. This helps in estimating the required life insurance coverage based on the user's current financial situation and future goals.

Key Data Points Collected:

Income Details – Annual income, monthly salary.

Expense Breakdown – Monthly expenses (rent, utilities, loans, education, etc.).

Liabilities & Loans – Any outstanding debts (home loan, car loan, personal loans).

Dependents & Family Status – Spouse, children, elderly parents, or other dependents.

Existing Insurance Coverage – Whether the user already has life insurance and how much coverage they have.

Example Chatbot Interaction:

Chatbot: "To recommend the best policy for you, I need to understand your finances. What’s your annual income?"

User: "₹12,00,000 per year."

Chatbot: "Got it! Do you have any existing life insurance coverage?"

User: "Yes, ₹20 lakh term plan."

Chatbot: "Noted! Let’s move on to your financial responsibilities."

Why This is Important?

Helps in calculating the right insurance coverage.

Avoids over-insurance or under-insurance by considering existing policies.

Provides accurate premium estimates based on financial capacity.

#### 9.1.2.b Storing User Preferences

Description:

The system records and stores the user’s choices regarding insurance plans, policy types, and additional coverage preferences.

Stored Preferences Include:

Preferred Policy Type – Term insurance, ULIP, Endowment plans, etc.

Coverage Amount – ₹25 lakh, ₹50 lakh, ₹1 crore, etc.

Policy Duration – 10 years, 20 years, 30 years.

Additional Riders – Critical illness, accidental death, waiver of premium, etc.

Example Chatbot Interaction:

Chatbot: "Do you want to add extra protection like Accidental Death Cover or Critical Illness Rider?"

User: "Yes, I’d like both."

Chatbot: "Great! I’ll include these in my recommendations."

Why This is Important?

Users don’t have to re-enter preferences every time they interact with the system.

Helps in providing personalized recommendations based on saved preferences.

#### 9.1.2.c Profile Dashboard

Description:

Users can access their profile dashboard, which displays previous interactions, saved recommendations, and purchased policies.

Dashboard Features:

Saved Insurance Plans – Policies the user has considered but not purchased.

Past Interactions – Displays previous chatbot conversations and recommendations.

Session History – Allows users to resume previous need analysis sessions.

Personalized Recommendations – Suggests plans based on the user’s profile.

Example Dashboard View:

Section Details Displayed

My Insurance Plans Saved plans with coverage details.

My Recommendations AI-powered policy suggestions.

My Profile Financial details, dependents, existing policies.

Saved Sessions Incomplete insurance analyses for retrieval.

Why This is Important?

Users can manage all their insurance needs in one place.

Encourages users to complete the policy selection process.

#### 9.1.2.d Validating User Input

Description:

Ensures that user-entered data follows the correct format and is within valid ranges.

Validation Rules:

Numeric Data Validation – Age must be between 18-70 years, income must be a positive number.

Currency Formatting – Amounts (₹) must be formatted correctly (e.g., ₹10,00,000 instead of "10L").

Dropdown Selections – Ensures valid plan selection instead of free-text input.

Example Chatbot Interaction:

Chatbot: "How much coverage do you need?"

User: "₹50 Crore"

Chatbot: "That seems too high based on your income. Let’s adjust to a more realistic amount."

Why This is Important?

Prevents errors in calculations.

Ensures accurate premium estimates and policy matching.

#### 9.1.2.e Handling Missing or Conflicting Data

Description:

If a user provides incomplete or conflicting details, the chatbot prompts them to clarify their responses before proceeding.

Example Scenarios:

User Input Issue Detected Chatbot Response

Income: ₹10L, Expenses: ₹12L Expenses exceed income "Your expenses seem higher than your income. Would you like to adjust them?"

Savings: ₹0, Investment Plan: High-Risk ULIP Conflict in risk-taking ability "ULIPs involve risk. Are you comfortable with a high-risk investment?"

Age: 20, Retirement Plan: 30 Years Plan term exceeds working years "This policy lasts beyond typical retirement age. Would you like to reconsider the term?"

Why This is Important?

Avoids misleading recommendations.

Helps users correct mistakes before finalizing a plan.

#### 9.1.2.f Free-Form Query Processing

Description:

Instead of forcing users to select from predefined options, LINAT allows them to ask questions in their own words and uses AI to interpret the intent.

Example Chatbot Interaction:

User: "Can I get a plan that covers my family if something happens to me?"

Chatbot: "Yes! A family protection term plan provides financial security to your loved ones. Would you like to explore options?"

Why This is Important?

Provides a more natural, flexible experience for users.

Reduces frustration for users who prefer typing over selecting predefined choices.

### 9.1.3 AI-Powered Need Analysis & Plan Matching in LINAT

This module is **the core intelligence** of LINAT, ensuring that **users receive the most relevant insurance recommendations** based on their unique financial needs. It processes user inputs, **estimates coverage**, assesses **risk tolerance**, dynamically adjusts **follow-up questions**, and suggests **tax-saving policies**.

10.1.3.a Need Analysis Based on Financial Goals

**Description:**

LINAT identifies the **user’s primary insurance goal** to **personalize recommendations**. Each goal aligns with **specific policy types**, ensuring that users only receive **relevant plan options**.

**Supported Financial Goals & Matching Insurance Types:**

|  |  |  |
| --- | --- | --- |
| **User Goal** | **Suitable Policy Type** | **Purpose** |
| **Wealth Creation** | ULIP, Endowment Plan | Investment-linked growth with life cover. |
| **Retirement Planning** | Pension Plans, Annuities | Provides financial security after retirement. |
| **Child Education** | Child Insurance Plans | Ensures funds for higher education. |
| **Family Protection** | Term Insurance | Provides high coverage at low cost. |
| **Tax Savings** | ULIP, Endowment, Term Plan | Offers benefits under **Section 80C & 10(10D)**. |

**Example Chatbot Interaction:**

* **Chatbot:** "What is your primary goal for life insurance?"
  1. Wealth Creation
  2. Retirement Planning
  3. Child’s Education
  4. Family Protection
  5. Tax Savings
* **User:** "Child’s Education"
* **Chatbot:** "Great! Planning early ensures your child gets the best education. Let’s estimate how much coverage you’ll need."
* **Why This is Important?**

**Aligns policy recommendations with user intent.**  
 **Simplifies complex insurance choices.**

#### 10.1.3.b Coverage Requirement Estimation

**Description:**

The chatbot **calculates the required sum assured** based on the user's financial situation, future obligations, and inflation.

**Formula for Coverage Calculation:**

Required Coverage=(Annual Income×10−15)+(Liabilities)−(Existing Insurance Cover)\text{Required Coverage} = (\text{Annual Income} \times 10-15) + (\text{Liabilities}) - (\text{Existing Insurance Cover})Required Coverage=(Annual Income×10−15)+(Liabilities)−(Existing Insurance Cover)

**Example Chatbot Interaction:**

* **Chatbot:** "What is your annual income?"
* **User:** "₹10,00,000"
* **Chatbot:** "Do you have any loans or outstanding debts?"
* **User:** "Yes, ₹20 lakh home loan."
* **Chatbot:** "Considering your income and liabilities, you may need around ₹1.2 crore coverage to ensure your family’s financial security."

**Why This is Important?**

**Helps users understand how much insurance they actually need.**  
 **Avoids under-insurance or over-insurance.**

10.1.3.c Risk Profile Assessment

**Description:**

LINAT **evaluates risk tolerance** and **adjusts policy recommendations** based on how much risk a user is willing to take.

**Risk Levels & Suitable Policy Types:**

|  |  |
| --- | --- |
| **Risk Level** | **Policy Type Recommended** |
| **Low Risk** | Endowment Plan, Money-Back Policy |
| **Moderate Risk** | ULIP with Balanced Fund |
| **High Risk** | ULIP with Equity Fund |

**Example Chatbot Interaction:**

* **Chatbot:** "Are you comfortable with market-linked returns, or do you prefer guaranteed returns?"
  1. **Low Risk** – I prefer guaranteed returns.
  2. **Moderate Risk** – I can take some risk for higher returns.
  3. **High Risk** – I want maximum growth potential.
* **User:** "Moderate Risk"
* **Chatbot:** "A **ULIP with Balanced Fund** might be a good choice for you. It offers both **growth and stability**."

**Why This is Important?**

**Ensures the policy aligns with user comfort level.**  
 **Prevents users from choosing unsuitable policies.**

#### 10.1.3.d Dynamic Questioning

**Description:**

Instead of **fixed scripts**, the chatbot adapts its **next question based on previous responses**.

**Example Scenarios:**

|  |  |
| --- | --- |
| **User Response** | **Chatbot’s Next Question** |
| **"I want a ULIP for wealth creation."** | "Would you like your money to grow aggressively in equities, or prefer a balanced approach?" |
| **"I have ₹5 lakh savings and want family protection."** | "Would you like a high-cover term plan or a return-of-premium plan?" |
| **"I’m a freelancer with irregular income."** | "Would you prefer flexible premium payments?" |

**Why This is Important?**

**Makes the chatbot feel intelligent and responsive.**  
 **Ensures users only answer relevant questions.**

#### 10.1.3.e Tax-Saving Optimization

**Description:**

Many users want **tax benefits** along with life coverage. LINAT suggests **tax-efficient insurance plans** under **Sections 80C and 10(10D)**.

**Tax-Saving Policies:**

|  |  |  |
| --- | --- | --- |
| **Policy Type** | **Tax Benefit Under** | **Tax-Saving Limit** |
| **Term Insurance** | **Section 80C** | Up to ₹1.5 lakh deduction |
| **ULIP** | **Section 80C + 10(10D)** | Tax-free maturity payout |
| **Endowment Plan** | **Section 80C + 10(10D)** | Tax-free returns |

**Example Chatbot Interaction:**

* **Chatbot:** "Are you looking for tax-saving insurance options?"
* **User:** "Yes."
* **Chatbot:** "Term insurance allows a **₹1.5 lakh tax deduction under Section 80C**. ULIP plans also provide **tax-free maturity benefits under Section 10(10D)**. Would you like to explore these?"

**Why This is Important?**

**Educates users on how insurance helps in tax planning.**  
 **Encourages users to choose tax-efficient policies.**

### 9.1.4 Advanced Calculations & Estimations in LINAT

This module enhances the accuracy of **insurance planning** by providing users with **data-driven estimations** related to premiums, affordability, long-term cost projections, and specific financial goals such as **education planning and business loan collateral protection**.

#### 9.1.4.a Premium Estimator

**Description:**

The **Premium Estimator** helps users understand **how much they will pay for a selected insurance policy** before committing to a purchase. The system considers **user age, sum assured, policy term, and additional riders** to calculate an accurate premium estimate.

**Formula for Premium Estimation:**

Premium=(Base Rate×Sum Assured1000)+Rider Cost+18% GST\text{Premium} = \left(\frac{\text{Base Rate} \times \text{Sum Assured}}{1000}\right) + \text{Rider Cost} + 18\% \text{ GST}Premium=(1000Base Rate×Sum Assured​)+Rider Cost+18% GST

**Example Chatbot Interaction:**

* **Chatbot:** "Would you like to estimate your insurance premium before making a decision?"
* **User:** "Yes."
* **Chatbot:** "What is your age?"
* **User:** "35 years."
* **Chatbot:** "How much coverage do you need?"
* **User:** "₹1 crore."
* **Chatbot:** "For how many years do you want coverage?"
* **User:** "20 years."
* **Chatbot:** "Do you want to add riders like **Accidental Cover** or **Critical Illness**?"
* **User:** "Yes, add **Accidental Cover**."
* **Chatbot:** "**Calculating...** Based on your inputs, your estimated monthly premium is **₹2,500 per month** including **Accidental Cover** and taxes. Would you like to proceed?"

Why This is Important?

**Gives users an upfront idea of costs** before purchasing.  
 **Helps users adjust coverage based on affordability.**

9.1.4.b Affordability Analysis

**Description:**

If a user finds the **calculated premium too high**, LINAT suggests **alternative plans** that fit their budget.

**Example Chatbot Interaction:**

* **User:** "₹2,500 per month is too expensive. Can I get a lower premium?"
* **Chatbot:** "Certainly! Here are some alternative options:"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Plan Name** | **Coverage** | **Premium (₹/Month)** | **Policy Term** | **Changes Made** |
| **Plan A** | ₹1 crore | ₹2,500 | 20 years | Default Selection |
| **Plan B** | ₹75 lakh | ₹1,800 | 20 years | Lower Sum Assured |
| **Plan C** | ₹1 crore | ₹2,000 | 15 years | Shorter Policy Term |

* **Chatbot:** "Would you like to proceed with one of these options?"
* **User:** "I’ll go with Plan C."
* **Chatbot:** "Great! You have selected **₹1 crore coverage for 15 years at ₹2,000/month.** Proceeding to the next step!"

**Why This is Important?**

**Ensures users don’t abandon the process due to cost concerns.**  
 **Helps users find the best coverage within their budget.**

#### 9.1.4.c Long-Term Cost Projection

**Description:**

LINAT helps users understand how their **premium costs and insurance coverage may change over time**, taking into account factors like **inflation** and policy renewals.

**Example Chatbot Interaction:**

* **Chatbot:** "Would you like to see how your insurance costs will change over time?"
* **User:** "Yes."
* **Chatbot:** "Considering an inflation rate of **6% per year**, your future costs would be:"

|  |  |  |
| --- | --- | --- |
| **Year** | **Projected Premium (₹/Year)** | **Cumulative Cost (₹)** |
| **Year 1** | ₹30,000 | ₹30,000 |
| **Year 5** | ₹35,000 | ₹1,75,000 |
| **Year 10** | ₹42,000 | ₹4,20,000 |
| **Year 15** | ₹50,000 | ₹7,50,000 |

* **Chatbot:** "As seen above, **your premium may increase over time due to inflation.** Would you like to adjust your coverage now to plan ahead?"

**Why This is Important?**

**Helps users plan for the long-term affordability of their insurance.**  
 **Prevents users from underestimating future costs.**

Education Cost Inflation Calculator

**Description:**

For users **saving for their child’s education**, LINAT calculates **future education costs** based on **expected inflation rates** (6-10% per year).

**Formula for Future Education Cost:**

Future Cost=Current Cost×(1+Inflation Rate)Years\text{Future Cost} = \text{Current Cost} \times (1 + \text{Inflation Rate})^\text{Years}Future Cost=Current Cost×(1+Inflation Rate)Years

**Example Chatbot Interaction:**

* **Chatbot:** "How much do you estimate your child’s college education will cost today?"
* **User:** "₹25,00,000."
* **Chatbot:** "By what age do you expect your child to start college?"
* **User:** "At age 18."
* **Chatbot:** "Considering a **6% annual increase**, the estimated cost when your child turns 18 will be **₹50 lakh.** You may want to plan your savings accordingly."

|  |  |
| --- | --- |
| **Year** | **Projected Education Cost (₹)** |
| **Today** | ₹25,00,000 |
| **5 Years Later** | ₹33,45,000 |
| **10 Years Later** | ₹44,77,000 |
| **15 Years Later** | ₹50,00,000 |

**Chatbot:** "Would you like to explore an **Education Insurance Plan** that grows with inflation?"

**Why This is Important?**

**Helps parents plan accurately for future education expenses.**  
 **Encourages long-term savings with insurance-backed investments.**

#### 9.1.4.d Business Loan Collateral Protection Calculation

**Description:**

If a user **has taken a business loan and wants to secure it with life insurance**, LINAT suggests a **decreasing term plan** where the **coverage reduces as the loan is repaid**.

**Example Chatbot Interaction:**

* **Chatbot:** "Would you like life insurance to cover a business loan in case of unexpected events?"
* **User:** "Yes, I have a ₹50 lakh business loan for 10 years."
* **Chatbot:** "A **Decreasing Term Insurance Plan** is best for loan protection. Here’s how coverage works over time:"

|  |  |  |
| --- | --- | --- |
| **Year** | **Loan Balance (₹)** | **Suggested Insurance Coverage (₹)** |
| **Year 1** | ₹50,00,000 | ₹50,00,000 |
| **Year 5** | ₹30,00,000 | ₹30,00,000 |
| **Year 10** | ₹0 | ₹0 |

* **Chatbot:** "This plan ensures your loan is covered at all times, reducing as you repay. Would you like to proceed?"

**Why This is Important?**

**Protects businesses from financial risk.**  
 **Ensures the loan is repaid in case of unexpected events.**

### Advanced Functionalities & AI Improvements

This section covers additional AI-powered features that improve user experience, data retention, and financial planning capabilities. These functionalities ensure a seamless, intelligent, and adaptive chatbot system.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Premium Estimator & Affordability Check

Description:

Many users want to estimate how much they will need to pay before committing to a plan. This feature allows them to calculate insurance premiums based on their preferred coverage, policy term, and optional riders.

Actors:

• User – Seeks to estimate and compare premium costs.

• LINAT (Chatbot) – Processes user inputs and calculates premiums.

Goal:

To help users understand the affordability of different life insurance plans based on their financial situation.

Preconditions:

• The user selects the Premium Estimator option.

• The chatbot must have access to premium calculation formulas based on insurer pricing data.

Postconditions:

• The system calculates and displays estimated premium amounts.

• The user may choose to proceed with a plan, modify inputs, or compare plans.

Flow:

1. User selects "Estimate Premium".

2. Chatbot asks for key details:

o Age

o Desired coverage amount (sum assured)

o Policy term (years)

o Additional riders (accidental cover, critical illness, etc.)

3. Chatbot processes inputs and applies calculation formula:

o Basic Formula:

Premium = (Base Rate × Sum Assured ÷ 1000) + Rider Cost + GST (18%)

o The Base Rate depends on factors like age, policy type, and term length.

o If riders are selected, the additional cost is added to the base premium.

4. Chatbot displays the estimated premium:

o Monthly, quarterly, and annual breakdown.

o Tax benefits, if applicable.

5. User Actions:

o Proceed with Plan → Chatbot suggests available insurance products.

o Modify Inputs → User adjusts values and recalculates premium.

o Save Estimate → Premium estimate is stored under a session ID.

Example Chatbot Interaction:

• Chatbot: "Let's estimate your insurance premium! What type of plan are you interested in? (Term, Whole Life, Endowment, etc.)"

• User selects: "Term Plan"

• Chatbot: "What is your age?"

• User enters: "30 years"

• Chatbot: "How much coverage do you need?"

• User enters: "₹1 crore"

• Chatbot: "Would you like to add riders (Accidental Cover, Critical Illness)?"

• User selects: "Yes, Accidental Cover"

• Chatbot: "Calculating... Your estimated premium is ₹7,500 per year. Would you like to compare similar plans or proceed with this estimate?"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### Automated Session Tracking & Retrieval

Description:

If a user loses connection, closes the chat, or exits the session midway, their progress is saved automatically. This allows them to resume from where they left off instead of restarting the process.

Actors:

• User – Exits the chatbot but wants to resume later.

• LINAT (Chatbot) – Saves and retrieves past sessions.

• Database – Stores session information securely.

Goal:

To ensure users do not lose their progress, making the experience seamless and frustration-free.

Preconditions:

• The chatbot should be configured to store user progress at key checkpoints.

• Users must be able to retrieve previous sessions using an email or session ID.

Postconditions:

• The user resumes their insurance analysis from where they left off.

Flow:

1. User interacts with the chatbot and provides inputs.

2. User exits or gets disconnected midway.

3. Chatbot automatically saves progress:

o Stores the user’s responses up to that point.

o Generates a unique Session ID.

4. When the user returns, the chatbot asks:

o "Would you like to continue where you left off?"

5. User enters email or session ID.

6. Chatbot retrieves session data and resumes the conversation.

Example Chatbot Interaction:

• User initiates need analysis but exits midway.

• Chatbot saves session ID: "Your session has been saved. Use ID: 12345XYZ to continue later."

• User returns after a few hours and enters session ID.

• Chatbot: "Welcome back! Shall we continue where you left off?"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### Tax Planning & Life Insurance Optimization

Description:

Many users seek tax benefits when purchasing life insurance. This feature helps them understand tax savings and optimize their policy selection accordingly.

Actors:

• User – Wants to maximize tax benefits.

• LINAT (Chatbot) – Guides users on tax exemptions.

Goal:

To help users choose policies that align with tax-saving strategies.

Preconditions:

• User must be aware of tax benefits related to life insurance.

• The chatbot must be updated with the latest tax regulations (Section 80C, 80D, 10(10D)).

Postconditions:

• The user selects an insurance policy with tax-saving benefits.

Flow:

1. User asks about tax benefits.

2. Chatbot explains applicable tax laws:

o Section 80C: Premiums up to ₹1.5 lakh per year are tax-deductible.

o Section 80D: Health riders on life insurance qualify for additional deductions.

o Section 10(10D): Maturity proceeds are tax-free under certain conditions.

3. User provides income details.

4. Chatbot suggests plans that optimize tax savings.

Example Chatbot Interaction:

• User: "Can I save tax with life insurance?"

• Chatbot: "Yes! Under Section 80C, premiums up to ₹1.5 lakh per year are tax-deductible. Would you like to see policies that maximize your savings?"

• User: "Yes, show me options."

• Chatbot: "Here are 3 policies that offer high tax benefits…"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### AI Model Learning & Continuous Improvement

Description:

The system continuously learns from user interactions and market updates to improve recommendations.

Actors:

• AI Model – Learns from user behavior.

• LINAT (Chatbot) – Enhances responses over time.

• Insurance API – Updates real-time policy details.

Goal:

To ensure the chatbot adapts to market trends and user preferences.

Preconditions:

• The chatbot should have access to historical user data and policy updates.

Postconditions:

• The chatbot improves accuracy and relevance of future recommendations.

Flow:

1. User interacts with the chatbot.

2. Chatbot collects anonymous data on:

o User preferences

o Most selected plans

o Common questions

3. AI analyzes patterns and updates responses.

4. Insurance policies and premium rates are refreshed from APIs.

Example Improvement:

• If most users prefer plans with waiver-of-premium riders, the chatbot prioritizes those recommendations in similar cases.

### 9.1.5 Policy Issuance & User Decision Finalization

Once users have completed the need analysis and plan selection, they are given the option to **finalize their choice, proceed to purchase, save their selection, or request expert assistance** before making a decision. This stage ensures a smooth transition from recommendation to actual policy issuance.

#### Plan Finalization & Confirmation

**Description:**

Before proceeding to purchase, users can **review the details of their selected insurance plan**, including **coverage, premium amount, policy term, and any riders added**.

**Flow:**

1. **User selects an insurance plan** from the recommended options.
2. **Chatbot presents a detailed summary**, including:
   * Sum Assured
   * Policy Term
   * Monthly/Annual Premium
   * Riders Included
   * Tax Benefits (if applicable)
3. **User confirms the selection** or requests modifications.
4. **If modifications are requested**, chatbot provides customization options:
   * Adjust sum assured
   * Add/remove riders
   * Change policy term
5. **Once finalized**, chatbot provides next steps:
   * Proceed to policy issuance
   * Save for later
   * Request expert assistance

**Example Chatbot Interaction:**

* **Chatbot:** "Great choice! Here’s a summary of your selected plan:"  
  **Plan Name:** Secure Future Term Plan  
  **Coverage:** ₹1 crore  
  **Policy Term:** 25 years  
  **Premium:** ₹1,500/month  
  **Riders Included:** Accidental Death Benefit  
  **Tax Savings:** Eligible under Section 80C
* **Chatbot:** "Would you like to **proceed to purchase, modify the plan, save for later, or speak with an advisor**?"
* **User:** "I want to proceed to purchase."

#### Policy Issuance Redirection

**Description:**

Once a user confirms their insurance selection, LINAT **redirects them to the insurer’s policy issuance page** to complete the purchase process.

**Flow:**

1. **User confirms their selection.**
2. **Chatbot generates a secure redirection link** to the insurer’s website.
3. **User clicks "Proceed to Buy."**
4. **User is redirected to the insurer’s purchase portal** to enter personal details, verify identity, and complete payment.

**Example Chatbot Interaction:**

* **Chatbot:** "You’re all set! Click the link below to complete your purchase on the insurer’s website."
* **Proceed to Buy**

**Why This is Important?**

**Seamless transition from need analysis to policy purchase.**  
 **Ensures a guided user experience without losing engagement.**

#### Save for Later Feature

**Description:**

If a user is **not ready to make a purchase**, they can **save their preferred plan** and return later. LINAT stores their selection and generates a **session ID for retrieval**.

**Flow:**

1. **User decides to save their selection for later.**
2. **Chatbot generates a session ID or links the plan to the user’s profile.**
3. **User can retrieve the saved plan anytime from the dashboard.**
4. **Chatbot can send a reminder notification/email after a certain period.**

Example Chatbot Interaction:

* **User:** "I’m not ready to buy yet."
* **Chatbot:** "No problem! I’ve saved your selected plan. You can retrieve it anytime using **Session ID: 12345ABC**."
* **Chatbot:** "Would you like me to send a reminder in **3 days, 7 days, or not at all?**"

**Why This is Important?**

**Reduces drop-offs by allowing users to return later.**  
 **Encourages follow-ups without forcing a purchase.**

#### Request Expert Assistance

**Description:**

Some users prefer speaking with a **human advisor** before making a purchase. LINAT offers an option to **schedule a call or chat with an expert**.

**Flow:**

1. **User selects "Request Expert Assistance."**
2. **Chatbot asks for preferred contact method:**
   * Phone Call
   * WhatsApp Chat
   * Email
3. **User selects a time slot for a call or chat.**
4. **System schedules the session and confirms the appointment.**
5. **Advisor reaches out at the scheduled time.**

**Example Chatbot Interaction:**

* **User:** "I want to talk to an expert before deciding."
* **Chatbot:** "Sure! How would you like to be contacted?"  
  **1. Phone Call**  
  **2. WhatsApp Chat**  
  **3. Email Response**
* **User:** "Phone Call."
* **Chatbot:** "Got it! Please select a preferred time slot:"  
  **1. Today (4 PM - 6 PM)**  
  **2. Tomorrow (10 AM - 12 PM)**  
  **3. Custom Time**
* **User:** "Tomorrow, 10 AM."
* **Chatbot:** "Your appointment is confirmed! Our insurance expert will call you at **10 AM tomorrow**."

**Why This is Important?**

**Provides human assistance for complex queries.**  
 **Increases trust and improves conversion rates.**

### 9.1.6 Knowledge Base – Comprehensive Insurance Information System

The **Knowledge Base** serves as the **central repository** for all insurance-related data, enabling **AI-powered need analysis, policy recommendations, and user education**. It ensures users have access to **accurate, up-to-date, and structured information** to make informed decisions.

Insurance Information Repository

**Description:**

A **structured database** that holds extensive details about various life insurance products, ensuring the chatbot can **fetch relevant data** for personalized recommendations and user inquiries.

**Key Components:**

* **Comprehensive Product Database:**
  + Stores details of **different types of insurance policies**, including:
    - **Term Insurance Plans** (Pure risk cover)
    - **Endowment Plans** (Savings + protection)
    - **Unit-Linked Insurance Plans (ULIPs)** (Market-linked growth)
    - **Child Education Plans** (Future financial security for children)
    - **Pension & Annuity Plans** (Retirement-focused income)
    - **Group Insurance Plans** (Corporate employee benefits)
  + Contains **policy features**, **sum assured options**, **premium structures**, **benefit payouts**, and **exclusions**.
* **Dynamic Content Updates:**
  + Insurance regulations and **IRDAI compliance updates** are **automatically reflected**.
  + **New product launches** from insurers are periodically added.
  + **Market trends** affecting insurance pricing and benefits are incorporated.
* **User-Centric Categorization:**
  + Policies are grouped based on **user needs** (e.g., wealth creation, family protection, child education, retirement planning).
  + Allows **quick filtering** of plans based on financial goals.

**Example Scenario:**

* **User:** "What is the difference between Term Insurance and ULIP?"
* **Chatbot:**  
  "**Term Insurance** provides a **pure risk cover**, meaning your family gets a **payout only if something happens to you**. It’s affordable and ideal for protection.  
  **ULIP (Unit-Linked Insurance Plan)** is a **market-linked investment option** that combines insurance and wealth creation. Your premium is partly invested in funds that can grow over time. Would you like a comparison table?"

#### **Search Capability – AI-Powered Information Retrieval**

**Description:**

Users can **search the knowledge base** to get detailed information about insurance policies, concepts, and terminology.

**Key Features:**

* **Natural Language Processing (NLP) Search:**
  + Users can **type full questions** (e.g., "What’s the best insurance plan for retirement?") instead of selecting predefined options.
  + AI understands the context and retrieves the most relevant answers.
* **Structured Search Results:**
  + Displays **simple definitions** for common insurance terms.
  + Provides **detailed explanations with examples** for complex topics.
  + Highlights **related policies and their features**.
* **Contextual Assistance:**
  + If a user asks a general question (e.g., "What is a critical illness rider?"), chatbot explains its meaning.
  + If a user asks a specific question (e.g., "Does my policy include a critical illness rider?"), chatbot retrieves policy-specific details.

**Example Scenario:**

* **User:** "What’s the claim settlement ratio of ABC Insurance?"
* **Chatbot:**  
  "The latest **claim settlement ratio** for **ABC Insurance is 98.5%**, meaning **98.5% of claims were successfully paid out** in the last financial year. Would you like to compare this with other insurers?"

#### Embedded Knowledge Base API

**Description:**

The knowledge base is **not static**—it integrates with **external data sources** to ensure the chatbot provides **real-time and verified information**.

**Key Functionalities:**

* **Integration with Insurance Provider APIs:**
  + Fetches **latest policy details, updated benefits, exclusions, and pricing**.
  + Ensures information remains **accurate and compliant** with regulatory changes.
  + Retrieves **real-time competitor data** for policy comparison.
* **Regulatory Compliance & Legal Updates:**
  + Connects with **IRDAI and government insurance guidelines** to reflect policy changes.
  + Ensures chatbot provides **legally correct** information to users.
* **Document Retrieval System:**
  + Enables chatbot to **fetch historical data** for users retrieving past interactions.
  + AI-powered **indexing** speeds up searches across large datasets.

**Example Scenario:**

* **User:** "Can I see the latest term plans available in the market?"
* **Chatbot:** "Sure! Fetching the latest term insurance options..."
  + **Plan A** – ₹1 crore cover, ₹8,000/year premium.
  + **Plan B** – ₹1 crore cover, ₹9,500/year premium, includes critical illness rider.
  + **Plan C** – ₹1.2 crore cover, ₹10,000/year premium, flexible payout options.

### 9.1.7 User Profile & Dashboard Functionalities

The **User Profile & Dashboard** acts as a centralized space where users can access their **insurance recommendations, saved analyses, purchase history, and profile settings**. This feature improves **user experience** by allowing **seamless access to past interactions and easy modifications** to personal details.

#### View Past Insurance Recommendations

**Description:**

Users can **revisit previous insurance analyses** to see recommendations they received in the past. This is useful for users who:

* Want to compare past and current recommendations.
* Need to review details before making a decision.
* Wish to resume their insurance selection process later.

**Flow Explanation:**

1. **User logs into their profile dashboard.**
2. **Selects "Past Recommendations".**
3. **Chatbot displays previous recommendations**, including:
   * Policy Name
   * Coverage Amount
   * Premium Estimate
   * Riders & Additional Benefits
4. **User actions available:**
   * "Compare with New Plans" → Fetches updated recommendations.
   * "Proceed to Buy" → Redirects to the insurer’s website for purchase.
   * "Modify Inputs & Recalculate" → Allows users to adjust their needs and get revised suggestions.

**Example Scenario:**

* **User:** "I want to check the term insurance recommendation you gave me last time."
* **Chatbot:** "Here’s your last analysis from [Date]: ₹1 crore term insurance plan for ₹9,000 per year. Would you like to compare with updated options?"

#### Saved Analysis Retrieval

**Description:**

Users can **save their insurance analysis progress** and **retrieve it later** for modifications or purchase.

**Flow Explanation:**

1. **User selects "Saved Analysis" from the dashboard.**
2. **Chatbot fetches previously saved sessions**, displaying:
   * Coverage Requirement
   * Policy Recommendations
   * Estimated Premium
   * Riders & Add-ons
3. **User actions available:**
   * "Continue Where I Left Off" → Resumes past session.
   * "Modify Coverage & Recalculate" → Adjusts details for new recommendations.
   * "Delete Analysis" → Removes unwanted saved sessions.

**Example Scenario:**

* **User:** "I saved my analysis last week. Can I access it now?"
* **Chatbot:** "Yes! Here’s your saved analysis from [Date]. Would you like to continue or make changes?"

#### Policy Purchase History

**Description:**

Users can view policies they have **purchased through the system**, allowing them to track active policies, renewals, and claims.

**Flow Explanation:**

1. **User navigates to "My Policies".**
2. **Chatbot retrieves purchased policies**, displaying:
   * Insurer Name
   * Policy Type
   * Sum Assured
   * Premium Amount
   * Policy Term
   * Policy Status (Active/Expired)
3. **User actions available:**
   * "Download Policy Documents" → Retrieves soft copies of purchased policies.
   * "View Policy Details" → Expands information on riders, maturity benefits, etc.
   * "Renew Policy" → Redirects to the insurer’s renewal page.
   * "File a Claim" → Provides claim assistance & required documentation.

**Example Scenario:**

* **User:** "I want to check the policy I purchased last month."
* **Chatbot:** "You purchased a ₹50 lakh ULIP plan from XYZ Insurance on [Date]. Would you like to download your policy document?"

#### Update User Profile

**Description:**

Users can update their **personal details, income, and financial preferences** to receive updated insurance recommendations.

**Flow Explanation:**

1. **User selects "Update Profile".**
2. **Chatbot displays editable fields**, such as:
   * Personal Information (Name, Age, Gender, Marital Status)
   * Financial Details (Income, Expenses, Liabilities)
   * Dependents (Children, Spouse, Parents)
   * Risk Appetite (Low, Moderate, High)
3. **User makes changes & submits updates.**
4. **Chatbot confirms updates and suggests recalculating insurance needs if required.**

**Example Scenario:**

* **User:** "I got a salary hike recently. How do I update my profile?"
* **Chatbot:** "You can update your financial details under 'My Profile'. Adjust your income, and I’ll recalculate your recommended coverage!"

### 9.1.8 Security & Compliance

The LINAT system ensures a high level of security for user data, system integrity, and compliance with regulatory requirements. The security framework is designed to protect against unauthorized access, data breaches, and cyber threats while maintaining seamless operations.

#### Security Integration & Compliance

**9.1.8.a User Authentication & Access Control**

* **Multi-Factor Authentication (MFA)** – Users authenticate via OTP (SMS/Email) or app-based MFA.
* **Role-Based Access Control (RBAC)** – Different permissions are assigned for customers, advisors, and administrators.
* **Single Sign-On (SSO) Integration** – Enables login via Google, Microsoft, or enterprise authentication.

**9.1.8.b Data Protection & Encryption**

* **End-to-End Encryption:** Data is encrypted in transit (TLS 1.2/1.3) and at rest (AES-256).
* **Tokenization:** Sensitive financial details are tokenized to prevent direct exposure.
* **Data Masking:** Partial masking of sensitive fields (e.g., displaying only the last four digits of policy numbers).

**9.1.8.c Secure API & System Communication**

* **OAuth 2.0 & JWT Authentication:** Secure API interactions between the chatbot, backend, and external services.
* **API Rate Limiting & Throttling:** Prevents abuse by limiting the number of API requests per user/session.
* **Web Application Firewall (WAF):** Protects against common cyber threats (SQL injection, XSS, CSRF).

**9.1.8.d Compliance & Audit Controls**

* **GDPR & Data Privacy Compliance:** Adheres to regulatory requirements for data handling.
* **Audit Logging & Monitoring:** Tracks all user and admin activities for security and compliance.
* **Automated Session Expiry & Logout:** Ends idle sessions to prevent unauthorized access.

**9.1.8.e Fraud Detection & Anomaly Detection**

* **AI-Based Anomaly Detection:** Monitors unusual login behavior and financial inconsistencies.
* **IP & Device Tracking:** Flags suspicious logins from unrecognized locations or devices.
* **Automated Alerts & Notifications:** Sends real-time security alerts for unauthorized access attempts.

#### SSL/TLS Security Integration

**3.1 Secure Data Transmission**

* **TLS 1.2/1.3 Encryption:** Ensures secure data exchange between users and the system.
* **HTTPS Enforcement:** All communications use HTTPS for encrypted connections.
* **End-to-end Encryption:** Protects sensitive information from interception.

**3.2 Authentication & Certificate Management**

* **SSL Certificates:** Uses Extended Validation (EV) SSL certificates for trust and verification.
* **Certificate Pinning:** Prevents MITM (Man-in-the-Middle) attacks by allowing only trusted certificates.
* **Auto-Renewal & Monitoring:** Ensures SSL certificates are automatically renewed and monitored.

**3.3 Protection Against Cyber Threats**

* **Prevention of Eavesdropping:** Encrypted communication prevents unauthorized access to sensitive data.
* **Defending Against Phishing Attacks:** Verified SSL certificates ensure users interact with the legitimate LINAT system.
* **Data Integrity:** Ensures that transferred data remains unchanged during transmission.

**3.4 Compliance & Best Practices**

* **PCI DSS Compliance:** Required for handling sensitive financial data.
* **GDPR & DPDP Act Compliance:** Enforces secure data handling and encryption standards.
* **HSTS (HTTP Strict Transport Security):** Prevents protocol downgrades and ensures only secure connections are used.

#### System Security Monitoring & Risk Mitigation

**4.1 Continuous Monitoring**

* **24/7 Security Monitoring:** Automated security monitoring tools track anomalies and breaches.
* **Intrusion Detection Systems (IDS):** Identifies and alerts about potential security threats.
* **SIEM (Security Information and Event Management):** Logs and analyzes security events.

**4.2 Penetration Testing & Vulnerability Assessment**

* **Regular Security Audits:** Periodic security audits identify vulnerabilities.
* **Automated Penetration Testing:** Simulates cyber-attacks to identify system weaknesses.
* **Patch Management:** Ensures security patches are applied regularly to mitigate threats.

#### User Data Protection & Privacy Measures

**5.1 Data Access & Storage Controls**

* **Role-Based Data Access (RBAC):** Different user roles restrict access to sensitive information.
* **Zero Trust Security Model:** Users must verify identity at multiple levels before accessing data.
* **Secure Cloud Storage:** Data is stored in ISO 27001-certified environments with restricted access.
* **Data Residency Compliance:** User data is stored based on regional regulations (e.g., GDPR, DPDP Act).

**5.2 User Privacy & Transparency**

* **User Consent Management:** Users must explicitly agree to data collection policies.
* **Right to Erasure:** Users can request deletion of their data at any time.
* **Download User Data:** Users can obtain a copy of their stored information.
* **User Activity Log:** Users can review login history and chatbot interactions.
* **Two-Step Verification for Critical Actions:** Major actions require additional authentication.

#### Incident Response & Disaster Recovery

**6.1 Incident Response Plan**

* **Security Incident Management:** Defines procedures for handling security breaches.
* **Real-Time Threat Analysis:** Uses AI-driven threat intelligence to respond to security threats.
* **Immediate User Notification:** Users are informed about potential breaches or unauthorized activities.

**6.2 Disaster Recovery & Business Continuity**

* **Automated Data Backups:** Ensures periodic data backups for recovery purposes.
* **Redundant Cloud Storage:** Enables quick restoration of lost data.
* **Failover Mechanisms:** Ensures uninterrupted service in case of system failure.

The LINAT system employs multi-layered security measures to protect user data, ensure system integrity, and comply with legal regulations. These non-functional security activities help mitigate risks, enhance user trust, and maintain a robust security posture.

### 9.1.9 AI Learning & System Updates

The system continuously **learns from user interactions** and updates itself based on **market changes, policy updates, and evolving financial needs**.

#### Continuous AI Training

**Description:**

AI models improve over time by learning from past user queries and **adapting recommendations**.

**Flow Explanation:**

1. **AI analyzes user interactions** to understand trends in insurance selection.
2. **Patterns are identified**, such as:
   * Frequently asked questions
   * Popular insurance choices
   * Common financial concerns
3. **The model is retrained regularly** to enhance accuracy.

**Example Scenario:**

* **User:** "I want a tax-saving insurance plan."
* **Chatbot:** "Many users in your income bracket prefer policies under Section 80C. Let me show you some options!"

#### Market & Policy Updates via API

**Description:**

Insurance policies and pricing **frequently change**. LINAT stays updated by integrating with **insurer APIs** for real-time data retrieval.

**Flow Explanation:**

1. **API fetches updated policy details** (coverage, premium, riders).
2. **If new policies become available**, the chatbot incorporates them into recommendations.
3. **Regulatory changes are monitored** to ensure compliance with IRDAI rules.

**Example Scenario:**

* **User:** "Are there any new life insurance plans?"
* **Chatbot:** "Yes! A new term insurance plan was launched this month with better tax benefits. Would you like to explore it?"

#### Personalized Recommendations Over Time

**Description:**

As a user’s financial situation evolves, **the system dynamically adjusts recommendations** based on past interactions.

**Flow Explanation:**

1. **User’s past inputs are stored** (income, dependents, coverage needs).
2. **When a user returns, LINAT suggests adjustments**, such as:
   * Increasing coverage due to salary growth
   * Changing investment options for better returns
   * Adding new riders for better protection
3. **Users receive periodic policy updates** based on market trends.

**Example Scenario:**

* **User:** "I just got a promotion. Should I update my coverage?"
* **Chatbot:** "Yes! A higher income means your insurance coverage should increase. Let’s calculate your new requirements."

### 9.1.10 Handling Advanced User Scenarios & Edge Cases

Sometimes, users may **change preferences, lose connection, or have special insurance needs**. The system is designed to **adapt and provide solutions** for such cases.

#### Session Auto-Save & Recovery

**Description:**

If a user **loses connection or exits** the chatbot midway, **their progress is saved automatically**.

**Flow Explanation:**

1. **Chat session auto-saves every few interactions.**
2. **A session ID is generated**, linked to the user’s email.
3. **When the user returns, chatbot asks: "Would you like to continue where you left off?"**

**Example Scenario:**

* **User:** "I was in the middle of selecting a plan, but my internet disconnected."
* **Chatbot:** "No worries! Your session is saved. Would you like to resume from your last step?"

#### Handling Changing Preferences

**Description:**

If a user keeps changing answers **(e.g., switching between term and ULIP plans)**, the chatbot provides **side-by-side comparisons** to aid decision-making.

**Flow Explanation:**

1. **User starts with one insurance need** (e.g., wealth creation).
2. **User changes their mind multiple times** (e.g., shifts to retirement planning).
3. **Chatbot detects indecision** and suggests:
   * A comparison table
   * Pros & cons of each option

**Example Scenario:**

* **User:** "I want wealth creation… no wait, maybe a retirement plan?"
* **Chatbot:** "You seem unsure. Here’s a side-by-side comparison of both options to help you decide!"

#### Fraud Prevention & Input Validation

**Description:**

To ensure **accurate financial analysis**, the chatbot **flags unrealistic inputs** and requests clarifications.

**Flow Explanation:**

1. **User enters financial details (e.g., ₹10,000 income but requests ₹5 crore coverage).**
2. **Chatbot detects inconsistencies** and asks:
   * "Your requested coverage is unusually high for your income. Would you like to adjust your inputs?"
3. **User can modify details or proceed with an explanation.**

**Example Scenario:**

* **User:** "I earn ₹20,000 per month but want a ₹3 crore policy."
* **Chatbot:** "That coverage seems high for your income. Let’s explore a plan that fits your budget."

#### User Wants to Explore Other Needs

**Description:**

If a user **starts with one goal** (e.g., retirement) but later wants to explore others **(e.g., child education, tax savings)**, the chatbot dynamically **adapts**.

**Flow Explanation:**

1. **User initially selects a need (e.g., retirement).**
2. **Later, they express interest in another category.**
3. **Chatbot asks: "Would you like to see how both plans compare?"**

**Example Scenario:**

* **User:** "I was looking for retirement plans, but now I’m also interested in saving for my child’s education."
* **Chatbot:** "No problem! I’ll show you plans that can help with both. Would you like to see a combination plan?"

### 9.1.11 Context-Aware Decision Mechanism for Plan Recommendations

**Overview**

The LINAT system is designed to allow users to ask questions and provide information in an unstructured manner. To ensure a seamless experience, the system must intelligently decide when to stop gathering data and transition to plan recommendations. This is achieved through a **context-aware decision mechanism** that dynamically evaluates user inputs and determines the optimal point for recommendation generation.

**Decision Criteria for Plan Recommendations**

The system will rely on multiple adaptive mechanisms to transition from data collection to plan recommendations:

**1. Minimum Required Data Points Check**

* The system maintains a predefined set of **essential data points** required for an accurate plan recommendation. These include, but are not limited to:
  + Age
  + Income
  + Dependents
  + Financial goals
  + Risk preference
  + Desired payout structure (lump sum, annuity, etc.)
* Once these essential data points are collected, the chatbot will **stop asking further questions** and proceed to plan recommendations.
* If any critical data points are missing, the chatbot will prompt the user for the necessary details before transitioning.

**2. Confidence Score Threshold**

* The system assigns a **confidence score** based on the completeness and clarity of user responses.
* The confidence score increases as more relevant data is provided.
* When the score meets or exceeds a predefined threshold, the chatbot transitions to plan recommendations.
* If the score remains low due to vague or missing details, the chatbot continues prompting the user with clarifying questions to improve recommendation accuracy.

**3. Adaptive Questioning Approach**

* The chatbot dynamically adjusts its questioning approach based on user behavior:
  + If the user engages in **exploratory inquiries**, the chatbot provides relevant information without transitioning to recommendations.
  + If the user provides **specific financial details**, the system accelerates the transition to recommendations.
  + If the user explicitly asks **“What plan do you recommend?”**, the chatbot immediately provides plan options based on available data.

**4. User-Triggered Transition**

* Users will have the ability to **manually trigger plan recommendations** at any point by using:
  + A command such as **“Show my plan recommendations”** or **“Proceed with recommendations”**.
  + A dedicated button labeled **"Show My Plan Recommendations Now"**.
* This feature ensures that users who prefer a quick recommendation process can bypass additional questioning.

**5. Time-Based or Interaction-Based Transition**

* To prevent excessive questioning, the system implements an interaction threshold:
  + If a user has interacted for more than **X questions** or **Y minutes**, the chatbot will suggest transitioning to recommendations.
  + Example prompt: *“I have enough information to recommend plans. Would you like to see them now?”*
* If the user continues to provide more details, the chatbot refines the recommendations accordingly, ensuring flexibility in the decision-making process.

**Implementation Considerations**

* The context-aware decision mechanism will be integrated into the chatbot's AI-powered rule engine.
* Machine learning models will be utilized to fine-tune the confidence score thresholds over time, ensuring higher accuracy in plan recommendations.
* A user feedback loop will be incorporated to monitor user satisfaction and refine decision-making criteria as needed.

**Conclusion**

By implementing this context-aware decision mechanism, the LINAT system ensures that users receive timely and accurate plan recommendations while maintaining a seamless and personalized experience. This adaptive approach enhances user engagement and improves the overall effectiveness of the life insurance need analysis process.

### 9.1.12 Admin Panel Functionalities for LINAT System

#### 1. User Management

* **Login & Authentication**
  + Secure admin login (Multi-Factor Authentication - MFA recommended).
  + Role-based access control (Admin).
  + Logout and session timeout.
* **User Account Management**
  + View all registered users.
  + Search, filter, and sort users.
  + Edit user details (name, email, role).
  + Reset passwords & account recovery options.
  + Activate/deactivate user accounts.

#### 2. Dashboard & Reports Management

* **View Overall System Metrics**
  + Total active users, total sessions today, active users now.
  + Policy selection rate, bounce rate, and conversion rate.
  + User demographics and engagement trends.
* **Custom Reports Generation**
  + Generate, filter, and download reports (CSV, PDF).
  + Date-range-based filtering.
  + Compare engagement trends over different time periods.
* **Recent Activities & Logs**
  + View user activity logs (who accessed what, when).
  + Track policy selections and user drop-offs.

#### 3. Policy & Product Management

* **Manage Insurance Products**
  + Add/edit/remove policies available for recommendation.
  + Configure policy details (name, description, benefits, coverage amount).
  + Set selection criteria based on need analysis.
* **Policy Performance Analytics**
  + Track most & least selected policies.
  + Analyze user interaction with different policy options.

#### 4. Chatbot & Need Analysis Configuration

* **Manage Chatbot Responses**
  + Update predefined chatbot responses.
  + Add new FAQ responses.
  + Adjust chatbot rules for better need analysis.
* **Configure Need Analysis Criteria**
  + Modify user input parameters (income, age, dependents, financial goals).
  + Adjust risk assessment rules.
  + Enable/disable features like tax benefit suggestions.

#### 5. Data & Security Management

* **User Data Protection**
  + Ensure compliance with data privacy regulations.
  + Restrict access to sensitive user data.
  + Encrypt stored user information.
* **Audit & Monitoring**
  + Track system logs for unauthorized access.
  + Set alerts for unusual user behavior.

#### 6. Communication & Support

* **Email Notifications**
  + Send bulk emails for updates & policy promotions.
  + Notify users about incomplete need analysis sessions.
* **Live Support & Feedback**
  + Monitor chatbot feedback.
  + Respond to user queries via admin panel.

#### 7. System Configuration & Settings

* **Manage Admin Roles & Permissions**
  + Add new admin users.
  + Assign different permission levels (Full Access, Read-Only, etc.).
* **Backup & Restore Data**
  + Enable periodic system backups.
  + Restore previous data if needed.

The Admin Dashboard is an analytics and management system designed to monitor and analyze real-time data generated by the LINAT chatbot. LINAT serves as the primary data source, collecting customer responses, need analysis results, user sessions, and policy selection insights. This data is processed and displayed in various dashboard sections, such as User Analytics, Session Tracking, and Policy Performance.

**Data Flow and Dependency**

The Admin Dashboard relies entirely on LINAT for data retrieval and analysis. The following subsections outline the flow of data from LINAT to the dashboard.

**Data Flow from LINAT to Admin Dashboard**

1. User interacts with LINAT.
2. LINAT processes responses and stores need analysis data.
3. Backend API collects user session data.
4. The Admin Dashboard retrieves data via API calls.
5. Data is visualized in charts, tables, and performance metrics.

**Key Data Points Fetched from LINAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Point** | **Source (LINAT)** | **Admin Dashboard Section** | **Usage** |
| Total Sessions Today | User interactions | Dashboard | Displays total chatbot sessions |
| Active Users Now | Live session data | Dashboard | Shows real-time active users |
| User Demographics | Profile data | User Analytics | Displays age, location, and financial profile |
| Conversion Rate | Policy selection | Session Tracking | Measures policy adoption rate |
| Most Selected Policies | User selections | Policy Performance | Displays trending insurance plans |
| Recent User Emails | User chat logs | Admin Panel | Assists in customer follow-ups |

**API and Data Integration**

The Admin Dashboard fetches real-time analytics data from LINAT through secure REST APIs. LINAT's backend aggregates session logs, policy selection insights, and chatbot interaction data, exposing it through API endpoints.

**API Endpoints Used by the Admin Dashboard**

|  |  |  |  |
| --- | --- | --- | --- |
| Endpoint | Method | Description | Used In |
| /api/sessions/today | GET | Fetches total chatbot sessions today | Dashboard |
| /api/users/active | GET | Retrieves the number of currently active users | Dashboard |
| /api/users/demographics | GET | Provides user age, income, and location data | User Analytics |
| /api/policies/popular | GET | Fetches the most selected policies | Policy Performance |
| /api/chats/recent | GET | Retrieves the last 100 chatbot interactions | Admin Panel |

**Functional Dependencies**

Since the Admin Dashboard is entirely dependent on LINAT, its availability is crucial for dashboard functionality. If LINAT encounters an outage or API failure, the following measures will be implemented:

* **Retry Mechanism:** The dashboard will attempt automatic retries for fetching data.
* **Fallback Data:** If real-time data is unavailable, the system will display cached statistics from the last successful fetch.
* **Error Handling and Notifications:** Admins will receive alerts if LINAT data is not retrieved for more than five minutes.

**Admin-Specific Functionalities**

The Admin Dashboard provides system monitoring capabilities but does not allow modification of chatbot-generated data. The following functionalities are available to administrators:

|  |  |  |
| --- | --- | --- |
| Feature | Description | Access Level |
| View User Analytics | See demographics and engagement data | Admin |
| Monitor Policy Performance | Track the most selected policies | Admin |
| Download Reports | Export chatbot interaction data | Admin |
| Adjust Chatbot Settings | Modify chatbot responses (if required) | Super Admin |
| View Error Logs | Monitor API failures or system downtime | Super Admin |

**Error Handling and Monitoring**

If LINAT fails to send data to the Admin Dashboard, the system will:

1. Log the error using ELK Stack (Elasticsearch, Logstash, Kibana).
2. Notify administrators via email and dashboard alerts.
3. Display a fallback message in place of missing data visualizations.

**Conclusion**

The Admin Dashboard does not generate data independently but relies entirely on LINAT's backend. It is essential that:

* LINAT’s APIs remain highly available and responsive.
* Any downtime in LINAT directly impacts the reporting and insights available on the Admin Dashboard.
* Admin functionalities focus on monitoring and analysis rather than data modification.
* Effective error handling mechanisms ensure business continuity by providing fallback data and notifications during disruptions.

## 9.2 Non-Functional Requirements

### 9.2.1 Performance

* **Response Time**
  + The system should respond to user queries within 2 seconds for most interactions.
  + The recommendation engine should generate personalized plan recommendations within 3 seconds for optimal user experience.
* **Scalability**
  + The system should accommodate up to 10,000 concurrent users with the ability to scale horizontally to handle increased traffic.
  + Must support future growth in user base and data volume without degradation in performance.

### 9.2.2 Reliability

* **Availability**
  + The system should be available 99.9% of the time, ensuring minimal downtime for users.
  + Scheduled maintenance should be communicated to users in advance, with minimal disruption.
* **Fault Tolerance**
  + The system should gracefully handle failures and provide fallback mechanisms to ensure availability (e.g., caching strategies, backup servers).
  + In case of partial system failure, the chatbot must still provide basic functionalities.

### 9.2.4 Usability

* **User Interface Design**
  + The user interface should be intuitive and user-friendly, minimizing the learning curve for new users.
  + Provide accessible design features to accommodate users with disabilities (e.g., screen reader compatibility, keyboard navigation).
* **Documentation**
  + Comprehensive user documentation and help resources should be available, including FAQs, tutorials, and troubleshooting guides.

### 9.2.5 Maintainability

* **Code Quality**
  + The system should follow best coding practices (e.g., modular design, clear documentation) to facilitate easy maintenance and updates.
  + Code should be well-structured and commented, allowing for easy understanding and modification by developers.
* **Update Process**
  + The deployment process should allow for seamless updates to the system without significant downtime.
  + Automated testing should be implemented to ensure new features do not introduce regressions.

### 9.2.6 Compatibility

* **Cross-Platform Support**
  + The system should be compatible with multiple platforms, including web browsers (Chrome, Firefox, Safari) and mobile devices (iOS, Android).
  + Ensure responsiveness of the chatbot interface across different screen sizes and resolutions.
* **Integration with Third-Party Services**
  + The system should support integration with various insurance company APIs and payment gateways, adhering to their respective standards.

### 9.2.7 Audit and Monitoring

* **Logging**
  + All user interactions, system events, and errors should be logged for auditing and troubleshooting purposes.
  + Logs should be accessible to authorized personnel for analysis and compliance audits.
* **Monitoring**
  + Implement monitoring tools to track system performance, usage metrics, and error rates in real-time.
  + Alerts should be set up to notify the development team of any critical issues or anomalies in system behavior.

# 10 Technical Requirement

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Feature** | **Functionality** | **Criticality** | **Response Type** | **Comments** |
| **TR.1** | Microservices Architecture | The system will follow a microservices-based approach for modularity, independent scalability, and fault tolerance. | **Vital** | **C/ER** | Ensures efficient system scaling and maintenance. |
| **TR.2** | API-First Design | All components will communicate via RESTful APIs with JSON payloads, allowing seamless integration with third-party services. | **Vital** | **C/ER** | Ensures easy integration with insurers and external services. |
| **TR.3** | Cloud-Based Deployment | The system will be deployed on cloud infrastructure (AWS/Azure/GCP) for high availability and disaster recovery. | **Vital** | **C/ER** | Ensures reliability, scalability, and cost-efficiency. |
| **TR.4** | AI Model Hosting | AI models for LINAT will be hosted using scalable GPU-optimized cloud instances or API-based AI services. | **Essential** | **C/ER** | Optimizes AI performance for need analysis and chatbot functions. |
| **TR.5** | **NLP & RAG-based AI Integration** | The chatbot will use NLP models and **RAG-based retrieval** for intelligent conversations and relevant product recommendations. | **Vital** | **ER** | Enhances chatbot accuracy and user experience. **ML is not required unless justified.** |
| **TR.6** | Web-Based UI Framework | The front-end will be built using **React with Tailwind CSS** for a responsive UI and smooth cross-device experience. | **Essential** | **ER** | Ensures a modern and interactive user experience. |
| **TR.7** | Secure Authentication & Authorization | Implements OAuth 2.0 / JWT for user authentication and RBAC (Role-Based Access Control) for authorization. | **Vital** | **C/ER** | Ensures secure user management and access control. |
| **TR.8** | Data Encryption | All sensitive data, including user profiles and financial details, will be encrypted using AES-256 and TLS 1.3. | **Vital** | **C/ER** | Prevents unauthorized access and ensures data privacy. |
| **TR.9** | **Knowledge Base Retrieval System** | The system will retrieve product information from a structured knowledge base instead of real-time data sources. | **Vital** | **ER** | **Real-time retrieval is not needed; focus on structured data queries.** |
| **TR.10** | Session Management | Implements secure session tracking with expiration and user re-authentication for stored conversations. | **Essential** | **ER** | Enhances user experience while maintaining security. **Session ID & email alone are not sufficient for security—add more security layers.** |
| **TR.11** | Logging & Monitoring | Uses centralized logging (ELK Stack/Splunk) and real-time monitoring (Prometheus/Grafana) for system health tracking. | **Essential** | **C/ER** | Helps in proactive issue resolution and performance monitoring. |
| **TR.12** | High Availability & Load Balancing | Implements auto-scaling with load balancers to manage high traffic and prevent downtime. | **Vital** | **C/ER** | Ensures system uptime and smooth performance under heavy loads. |
| **TR.13** | Disaster Recovery & Backups | Automatic backups and failover mechanisms to ensure data recovery in case of failures. | **Vital** | **C/ER** | Protects against data loss and minimizes downtime. |
| **TR.14** | AI Model Training Pipeline | Implements automated pipelines (MLOps) for continuous training and deployment of AI models. | **Essential** | **C/ER** | Ensures that AI models improve over time with new data. **Optional if RAG suffices.** |
| **TR.15** | Compliance with IRDAI & GDPR | System architecture will adhere to IRDAI, GDPR, HIPAA, and other regulatory requirements. | **Vital** | **C/ER** | Ensures legal and regulatory compliance. |
| **TR.16** | Multi-Tenancy Support | System will support multiple insurers and product configurations under a single architecture. | **Essential** | **C/ER** | Allows expansion to multiple insurance providers. |
| **TR.17** | Cross-Platform Browser Compatibility | The web app will support all major browsers (Chrome, Firefox, Edge, Safari) and adapt to different screen sizes. | **Essential** | **ER** | Ensures maximum accessibility. |
| **TR.18** | **Gamification & User Engagement** | Introduce gamification and **sentiment analysis** to retain users and ensure a smooth chatbot experience. | **Essential** | **ER** | Keeps users engaged and reduces drop-off rates. |
| **TR.19** | **Multi-Turn Conversation & Chain of Thought** | The chatbot will support **goal-based branching** and **context-aware responses** to dynamically adjust to user inputs. | **Vital** | **ER** | Ensures an **interactive and adaptive conversation flow**. |
| **TR.20** | **Fraud Detection Algorithms** | AI will monitor anomalies in user inputs to detect fraudulent activity. | **Essential** | **ER** | Helps flag potential fraud risks. |
| **TR.21** | API Gateway & Rate Limiting | Uses API Gateway (Kong/Apigee) with rate limiting to prevent abuse and optimize traffic handling. | **Essential** | **C/ER** | Protects against API overuse and DDoS attacks. |
| **TR.22** | Auto-Scaling Infrastructure | Implements Kubernetes/Docker for containerized deployment and auto-scaling. | **Essential** | **C/ER** | Ensures scalability as traffic grows. |
| **TR.23** | CI/CD Pipeline | Implements DevOps CI/CD pipelines for automated testing, deployment, and rollback. | **Essential** | **C/ER** | Ensures smooth updates without downtime. |
| **TR.24** | Business Intelligence & Analytics | Uses BI tools (Power BI/Tableau) for real-time analytics on user behavior and insurance trends. | **Essential** | **ER** | Supports decision-making with actionable insights. |
| **TR.25** | **Premium Estimator API Consideration** | The premium estimator API may not always be available or updated. The system should have fallback calculations. | **Essential** | **ER** | API unavailability is a risk—note it and keep a backup method. |
| **TR.26** | API Rate Limiting & Security | Implements API rate limiting, token expiration, and anomaly detection for secure API interactions. | **Vital** | **C/ER** | Protects against API abuse and unauthorized access. |
| **TR.27** | **Web Page Design Clarification** | Web page design is **not a functional requirement**—focus on need analysis chatbot instead. | **Vital** | **ER** | Aligns expectations with functional scope. |
| **TR.28** | **Session ID & Access Control Enhancements** | Session IDs will be **securely stored** and tied to additional authentication layers for better security. | **Vital** | **ER** | Security beyond session ID and email is required. |
| **TR.29** | **Auto-Update Mechanism (Future)** | The system will include an automatic update mechanism to push software patches seamlessly. | **Desirable** | **ER** | Reduces manual maintenance efforts. |
| **TR.30** | **Edge Computing for AI Models (Future)** | Future AI model inference can be optimized using Edge Computing to reduce latency. | **Desirable** | **ER** | Improves chatbot response times in high-traffic situations. |

# 11 Reference Documents

| Document ID No | Date | Document Name | Version No | Source |
| --- | --- | --- | --- | --- |
| **Doc\_001.1** | **07-02-2025** | **This link provides a published URL offering a detailed explanation of the entire assignment. It covers all important aspects** | **1.1** | [**https://rainbow-platinum-367.notion.site/Assignment-1992918661d380f4b84fd5f162064adb**](https://rainbow-platinum-367.notion.site/Assignment-1992918661d380f4b84fd5f162064adb) |
| Doc\_002 | 13-03-2025 | Integration | 1.0 |  |
| Doc\_003 | 13-03-2025 | Gene AI in LINAT | 1.0 |  |
| Doc\_004 | 13-03-2025 | System Building Lifecycle | 1.0 |  |
| Doc\_005 | 20-02-2025 | Objective of Conducting the Need Analysis | 1.0 |  |
| Doc\_006 | 10-02-2025 | Chatbot Training Modules | 1.0 |  |
| Doc\_007 | 16-02-2025 | Need Matrix | 1.0 |  |
| Doc\_008 | 16-02-2025 | Data Flow in LINAT | 1.0 |  |

# 12 Open Items

# 13 Sign Off

|  |  |
| --- | --- |
| iNube | Customer |
| Name: | Name: |
| Title: | Title: |
| Date: | Date: |
| Signature: | Signature: |