Insurance Policy Generator using MERN & Chatbot

Name: Aravindhan V

Internship Duration: Feb 26 – April 30

College/Institute: [Your College Name]

Company (if applicable): [Company Name]

# Introduction

This project is about simplifying the process of insurance policy creation using a chatbot and the MERN stack. The main goal was to build a web-based system where users can interact with a chatbot to select an insurance type, provide necessary information, get their premium calculated, and generate a downloadable policy document. This helps reduce the manual work and makes the process user-friendly and interactive.

# Tools & Technologies Used

* MongoDB – For storing user and policy data
* Express.js – Backend server
* React.js – Frontend interface
* Node.js – Server runtime
* Botpress – Used for building the chatbot
* PDFKit – For generating policy documents in PDF format
* Mongoose – For MongoDB data modeling

# Main Features

* Chatbot to guide users through the process
* Supports Life, Health (Individual & Family), and Vehicle Insurance
* Dynamic forms based on selected insurance type
* Premium is calculated based on input data
* Policy PDF is generated and downloaded
* All details are saved in MongoDB

# System Flowchart

Below is the basic chatbot flow:  
  
1. User starts chat  
2. Bot greets and shows available insurance types  
3. User selects an option (e.g., Vehicle Insurance)  
4. Bot opens a form link  
5. User fills form and submits  
6. Backend calculates premium  
7. Bot asks if user wants to proceed  
8. If yes → Save data + Generate policy  
9. If no → Discard info

# How It Works – Step by Step

* The user opens the application and starts a chat with the bot.
* The chatbot gives a welcome message and shows 4 insurance options: Life, Health, Vehicle, Home.
* Based on the user’s selection, a new form opens in a pop-up.
* The form collects user-specific and policy-related details.
* After submitting the form, the backend calculates the premium.
* The bot shows the premium and asks the user whether to proceed.
* If the user agrees, the data is saved and a PDF policy is generated.
* If not, the chatbot discards all data.

# Premium Calculation Logic

Health Insurance:  
- Individual Plan: Base amount + condition-based risk  
- Family Plan: Includes all members, adds risk from family medical history  
  
Vehicle Insurance:  
- Based on vehicle age, type of coverage (comprehensive or third-party), and driver history  
- Example: New car with no accidents and comprehensive cover → lower premium  
- Old car with accident history → higher premium  
  
Life Insurance:  
- Depends on age, term, and sum assured  
- Older age = higher premium  
  
Frequency Type:  
- Premiums adjust based on Monthly / Quarterly / Semi-Annual / Annual selection  
  
Example:  
If base annual premium = ₹12,000  
- Monthly = ₹1,200  
- Quarterly = ₹3,200 (a little more than 3x due to extra charges)

# What I Learned

* How to build an end-to-end MERN application
* Working with real-time chatbots using Botpress
* Handling dynamic forms and user input
* Creating PDFs dynamically using PDFKit
* Connecting frontend, backend, and database smoothly

# Challenges Faced

* Getting the chatbot and React app to sync smoothly
* Designing the form submission flow and verifying data
* Creating flexible premium logic for different types
* Handling popup forms and checking if user submitted or closed it

# Future Improvements

* Add Home Insurance fully
* Add user authentication and login
* Create an admin dashboard to manage policies
* Allow users to make payments and get receipts
* Send policy PDF to email or WhatsApp automatically

# Conclusion

This project helped me understand how to build smart, interactive systems using full stack development. I learned how to connect a chatbot with backend logic and handle real user data. The whole insurance process became simpler and user-friendly through automation.