Leveraging Technology to Improve Customer Experience

A PROJECT REPORT

Submitted by,

Rachana Varadaraj	**20211CSE0391**
Mohammed Tauseef Ahmed	**20211CSE0353**
Dhriteshree M R	**20211CSE0365**
Mohammed Zahid Hasan	**20211CSE0359**

Under the guidance of,

Dr. Jothish C,
Associate Professor
in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **"Leveraging Technology to Improve Customer Experience"** being submitted by "Rachana Varadaraj, Mohammed Tauseef Ahmed, Dhriteshree M R, Mohammed Zahid Hasan" bearing roll number(s) "20211CSE0391, 20211CSE0353, 20211CSE0365, 20211CSE0359" in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a Bonafide work carried out under my supervision.

Dr. Jothish C
Associate Professor
School of CSE&IS
Presidency University

Dr. Asif Mohammed
HOD
School of CSE&IS
Presidency University

Dr. L. SHAKKEERA
Associate Dean
School of CSE
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vc School of Engineering
Dean -School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **Leveraging Technology to Improve Customer Experience** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. Jothish C, Associate Professor, School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Rachana Varadaraj(20211CSE0391)

Mohammed Tauseef Ahmed(20211CSE0353)

Dhriteshree M R(20211CSE0365)

Mohammed Zahid Hasan(20211CSE359)

ABSTRACT

In today’s fast-paced digital era, navigating the diverse landscape of insurance policies can be challenging for users. To simplify this process, our project introduces a comprehensive web platform that provides summarized insights into various insurance policies, catering to the needs of modern consumers.

The platform begins with a **Home Page** featuring options to log in or register. Upon successful registration, user credentials are securely stored in a **database**, ensuring data privacy and accessibility. After logging in, users are directed to an intuitive **Dashboard** showcasing four primary insurance categories: **Home**, **Life**, **Car**, and **Health**. Each category presents a curated list of companies offering related policies.

When a user selects a company, they are redirected to a detailed page containing the company's policy information. To enhance user convenience, a **Summarize Button** is provided on this page. There is a Download Button which will download the given insurance pdf which should be uploaded above the summarize option. Clicking the button generates a concise summary of the policy details on a new page.

The summarized page includes a **Redirect Button** that allows users to return seamlessly to the Home Page of the insurance company’s website. Additionally, every page is equipped with a **Navigation Bar** to facilitate easy movement between previous pages, ensuring a user-friendly browsing experience.

This project emphasizes a streamlined, accessible, and user-centric approach to exploring and understanding insurance policies. By integrating features such as text summarization, and a responsive design, our platform aims to empower users to make informed decisions with minimal effort. The system not only improves user engagement but also enhances accessibility, contributing to a more inclusive and efficient way to manage insurance-related tasks.

The application is designed with robust architecture, including a secure backend for data management and a dynamic frontend for intuitive user interaction, making it a valuable tool for individuals seeking clarity and simplicity in the insurance domain.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameer Uddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakeera L and Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and Dr."Asif Mohammed", Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr. Jothish C , Associate Professor** and Reviewer **Dr.Umapathi Gr, Assistant Professor**, School of Computer Science Engineering & Information Science, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman,** department Project Coordinators and Git hub coordinator Mr. **Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Rachana Varadaraj

Mohammed Tauseef Ahmed

Dhriteshree M R

Mohammed Zahid Hasan

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The insurance industry plays a vital role in providing financial security by covering risks such as health emergencies, accidents, or property damage. However, customers often struggle to understand their insurance policies due to the complex terms, lengthy documents, and legal jargon. Many individuals skip reading the full policy, leading to misunderstandings and dissatisfaction when claims are denied or unexpected clauses are invoked. This communication gap creates a sense of mistrust between customers and insurance providers, undermining the purpose of the policy itself. Modern customers expect more transparency in their dealings with companies. They prefer clear, concise explanations of policy terms, easy-to-access information, and trust-based relationships. Research shows that when customers clearly understand their coverage and exclusions, they are more likely to feel confident about their insurance choices and stay loyal to their insurer. However, traditional policy documents—usually filled with dense text—fail to meet these expectations, leading to low customer engagement and unresolved disputes during claims. Insurance companies are exploring ways to simplify communication using technology. This is how emerging technologies are improving transparency in the insurance industry, making policies more accessible, and fostering better trust between customers and insurers. The goal is to identify effective solutions that not only simplify policy documents but also ensure that customers fully understand their coverage, resulting in higher satisfaction and fewer disputes. In a competitive market, building trust through clear communication and accessible technology is essential for retaining customers and maintaining strong relationships over time.

CHAPTER 2

LITERATURE SURVEY

1.Overview of Insurance Management Systems

Modern digital platforms have significantly transformed the insurance sector by enabling seamless interaction between insurers and consumers. Studies suggest that user-friendly interfaces and advanced features like policy summaries, comparison tools, and read-aloud functionalities enhance user experience and accessibility (Rao et al., 2020). Platforms that provide easy access to policy information empower users to make informed decisions, a trend widely adopted in insurance management systems.

2. Importance of Summarization in Insurance Policies

Insurance documents often include lengthy and technical language, making them challenging for users to comprehend. Automatic text summarization simplifies this process by extracting key details, enhancing the user's ability to understand and compare policies (Ganesan et al., 2018). Projects that integrate summarization features improve accessibility and reduce cognitive load, particularly for users with limited domain knowledge.

3. Role of Dashboards in Information Systems

Dashboards are central to presenting categorized data in an organized manner. Research highlights that dashboards with categorized sections, such as health, life, car, and home insurance, enable intuitive navigation and improved user satisfaction (Patterson et al., 2014). Providing users with clear options and a centralized summary enhances decision-making and aligns with the principles of effective user experience design.

4. Impact of Comparative Tools in Decision Making

Comparison tools are increasingly valuable in digital platforms, particularly in sectors like insurance, where users evaluate multiple options. By presenting side-by-side comparisons of policy benefits, premiums, and coverage, these tools facilitate informed decision-making and improve transparency (Singh & Gupta, 2021).

5. Integration of Real-Time Databases in User Authentication

User authentication and registration systems backed by real-time databases enhance the security and scalability of web applications. These systems ensure that sensitive user data, such as login credentials, is stored securely while facilitating efficient retrieval for seamless user experiences (Kim et al., 2020).

6. Applications of API Integration in Web Development

API integrations, such as for language translations or text summarization, are instrumental in creating interactive and intelligent web applications. They allow developers to access powerful machine-learning models without extensive local resources, significantly enhancing functionality (Zhang et al., 2022).

7. Navigation in Web Applications

Effective navigation systems, including consistent navbars and redirect options, improve usability and user retention. Studies emphasize the importance of structured navigation to minimize cognitive effort and enable users to easily access desired information (Nielsen, 2018).

CHAPTER 3
RESEARCH GAPS OF EXISTING METHODS

The digital processing and summarizing of insurance papers has advanced significantly, but there are still a number of research gaps that limit the effectiveness of the present approaches. Finding these gaps aids in directing future advancements to produce systems that are more efficient and intuitive.

1. Low Summarization Model Accuracy:

- **Issue:** A lot of current NLP models, even cutting-edge transformers like BERT or GPT, have trouble accurately summarizing extremely complex and specialized insurance lingo.
- **Challenges:**
    - **Domain-Specific Knowledge:** General models frequently lack instruction on concepts and structures unique to the insurance business.
    - **Ambiguity in Policy Terms:** Summaries may misunderstand ambiguous passages or leave out important details.
- **Research Gap:** To increase summarization accuracy, domain-adapted or refined models tailored to the insurance industry are being developed.

2. Handling Diverse Document Formats:

- **Issue:** Insurance documents come in various formats (PDFs, scanned images, forms), making uniform text extraction challenging.
- **Challenges:**
    - **Inconsistent Data Quality:** OCR errors due to poor scan quality or complex layouts (tables, forms).
    - **Unstructured Data Processing:** Difficulty extracting structured data from unstructured text.
- **Research Gap:** Creating robust multi-format NLP pipelines that seamlessly process different document types while maintaining high accuracy.

3. Lack of Contextual Understanding:

- **Issue:** Current summarization methods often extract key sentences without understanding the context, leading to incomplete or misleading summaries.
- **Challenges:**
  - **Context Preservation:** Ensuring that the generated summary retains the original meaning, especially in legal and financial contexts.
  - **Contextual Dependencies:** Some terms or conditions are interdependent but may be separated in the document.
- **Research Gap:** Implementing advanced contextualization techniques to preserve logical and legal relationships in summaries.

4. Scalability and Real-Time Processing:

- **Issue:** Processing large volumes of documents in real-time remains challenging, especially when integrating AI/ML models on cloud platforms.
- **Challenges:**
  - **Resource Intensity:** Summarization models, particularly deep learning-based, require significant computational resources.
  - **Latency Issues:** Delays in generating summaries affect user experience, especially for mobile applications.
- **Research Gap:** Optimizing models for real-time processing with reduced computational overhead, possibly through model compression or distillation techniques.

5. Ethical and Regulatory Compliance:

- **Issue:** Summarization models must ensure data privacy and regulatory compliance, which is often overlooked in current implementations.
- **Challenges:**
  - **Data Security:** Summarizing sensitive information without breaching privacy laws (e.g., GDPR, HIPAA).
  - **Bias in Summarization:** Potential biases in training data could lead to inaccurate or unfair summaries.

- **Research Gap:** Developing explainable AI (XAI) systems that provide transparent, auditable summarization processes compliant with regulations.

6. User Experience and Customization:

- **Issue:** Existing systems often produce generic summaries that may not cater to the specific needs of different users (e.g., policyholders, agents).
- **Challenges:**
    - **Lack of Personalization:** Summaries do not adjust based on the user's role or preferences.
    - **Accessibility Issues:** Features like text-to-speech or multilingual support are often underdeveloped.
- **Research Gap:** Incorporating adaptive summarization mechanisms that personalize content based on user profiles and enhancing accessibility features.

7. Limited Integration with Existing Systems:

- **Issue:** Many current solutions do not integrate seamlessly with legacy insurance management systems or customer relationship management (CRM) tools.
- **Challenges:**
    - **Compatibility:** Ensuring new technologies work with older infrastructure.
    - **Data Synchronization:** Keeping summarized data consistent across multiple platforms.
- **Research Gap:** Developing modular, API-driven frameworks that ensure easy integration and data consistency.

8. Evaluation Metrics for Summarization Quality:

- **Issue:** Standard evaluation metrics (like ROUGE or BLEU) may not fully capture the quality of insurance document summarization.
- **Challenges:**
    - **Qualitative vs. Quantitative:** Metrics focus on text similarity rather than the accuracy or completeness of legal/financial information.
    - **Subjectivity:** Different users may have different expectations of what constitutes a "good" summary.

- **Research Gap:** Establishing more comprehensive evaluation frameworks tailored to the insurance domain, focusing on content relevance and accuracy.

9. Lack of Multi-Language Support:

- **Issue:** Most summarization tools are optimized for English, leaving non-English documents underserved.
- **Challenges:**
    - **Language-Specific Nuances:** Insurance terms can vary significantly across languages.
    - **Resource Constraints:** Limited training data for certain languages.
- **Research Gap**: Developing multilingual models that understand and summarize insurance documents accurately in various languages.

CHAPTER 4

PROPOSED METHODOLOGY

1.Analysis of Requirements

Stakeholder Engagement: Gather insights from stakeholders (customers and insurance professionals) to understand their needs for clearer policy communication.

Functional Requirements: Identify essential features, such as potential user account management.

Non-Functional Requirements: Define performance criteria, security measures, and usability goals to ensure a robust user experience.

2. System Design

Architecture Design: Outline how the React frontend, Node.js backend, and SQL database will interact.

Database Schema: Create a schema detailing tables for users, and policies, ensuring structured data management.

User Interface Design: Develop wireframes and mockups to visualize the user experience and ensure a user-friendly interface.

3. Technology Stack Selection

Frontend: Use React for building dynamic user interfaces.

Backend: Choose Node.js with Express for server-side development.

Database: Opt for an SQL database (e.g., MySQL or PostgreSQL) for structured data management.

Libraries/Tools: Identify additional libraries for multimedia handling.

4. Implementation

Project Setup: Initialize the project folder, set up Node.js with npm, and create a new React app.

Backend Development: Set up Express server and create RESTful APIs for managing policy and multimedia data.

Frontend Development: Build React components based on designs, including sections for policy details and multimedia content. It will fetch content dynamically from the backend to keep the user interface updated.

5.Monitoring and Maintenance

Monitoring: Implement tools to track user interactions and system performance. Focus on load times and error rates.

Feedback Collection: Establish mechanisms for users to provide feedback and suggestions for improvements.

Regular Maintenance: Schedule updates for software dependencies, fix bugs, and enhance features based on user input, ensuring database backups and security.

CHAPTER 5

OBJECTIVES

1. Simplifying Insurance Information Access:

The objective is to provide users with clear, concise summaries of complex insurance policies. Insurance documents can be lengthy and difficult to understand. By summarizing them, users can quickly grasp essential details, improving decision-making and reducing confusion.

2. **User-Friendly Navigation and Interface:**

The objective here is to ensure seamless navigation through a structured interface. The dashboard categorizes insurance types (home, life, car, and health), and a consistent navbar helps users move between pages efficiently. This structure enhances user experience by reducing cognitive load and improving accessibility.

3. **Personalized User Experience:**

It enables account creation and personalized access. User registration and login options ensure secure and customized experiences. Storing user data allows future enhancements, such as personalized recommendations based on previous interactions.

4. **Enhanced Insurance Detail Viewing:**

It will allow users to explore detailed policy information from different companies. Displaying detailed views for each company's policy helps users compare options in depth. This feature ensures transparency and empowers users to make informed choices.

5. **Summarization and Accessibility:**

The objective here is to provide summarized versions of insurance details with a read-aloud option. The summarization feature condenses complex information, while the read-aloud option enhances accessibility for users with visual impairments or those who prefer auditory learning.

6. **Comparison Functionality:**

It will develop a comparison tool for evaluating different insurance policies. Comparing policies side-by-side will help users identify the best option based on coverage, cost, and benefits. This feature will add significant value by supporting informed, comparative decision-making.

CHAPTER 6

SYSTEM DESIGN & IMPLEMENTATION

System Design

1. **Homepage and User Authentication:**
    - **Login and Registration**:
        - **Implementation**: Forms with field validation, secure password handling (hashed using libraries).
        - **Database Storage**: User credentials stored in a secure, encrypted format.
        - **Session Management**: Managed using session cookies for maintaining login states.
2. **Dashboard:**
    - **Insurance Categories**:
        - Four main categories: Home, Life, Car, and Health Insurance.
        - **Data Retrieval**: Fetching insurance company data from the database through API endpoints.
        - **Implementation**: Dynamically rendered category sections using front-end frameworks.
3. **Insurance Details Page:**
    - **Company-Specific Information**:
        - Data displayed dynamically based on the selected company.
        - **Implementation**: RESTful API calls to retrieve detailed policy data.
        - **Security Considerations**: Parameter validation to prevent SQL injection or unauthorized access.
4. **Summarization Functionality:**
    - **Summarize Button**:
        - **Backend Processing**: Utilizing OpenAI (gpt-3.5-turbo) to generate concise summaries.
    - **User Experience**: Summaries presented in a user-friendly format, with a focus on readability.
5. **Navigation and Redirection:**
    - **Nav Bar**:
        - Consistent across all pages, providing easy access to previous and main dashboard pages.
        - **Implementation**: Static component ensuring uniform navigation.
    - **Redirect Button**:
        - Takes users back to the homepage from the summarized page.
        - Implemented with simple event handling.

6. **Policy Comparison:**
    - **Design Considerations**:
        - Compare policies based on parameters like coverage, premium cost, and benefits.
    - **Technical Approach**: Implementing multi-select options and comparative analytics in the backend.

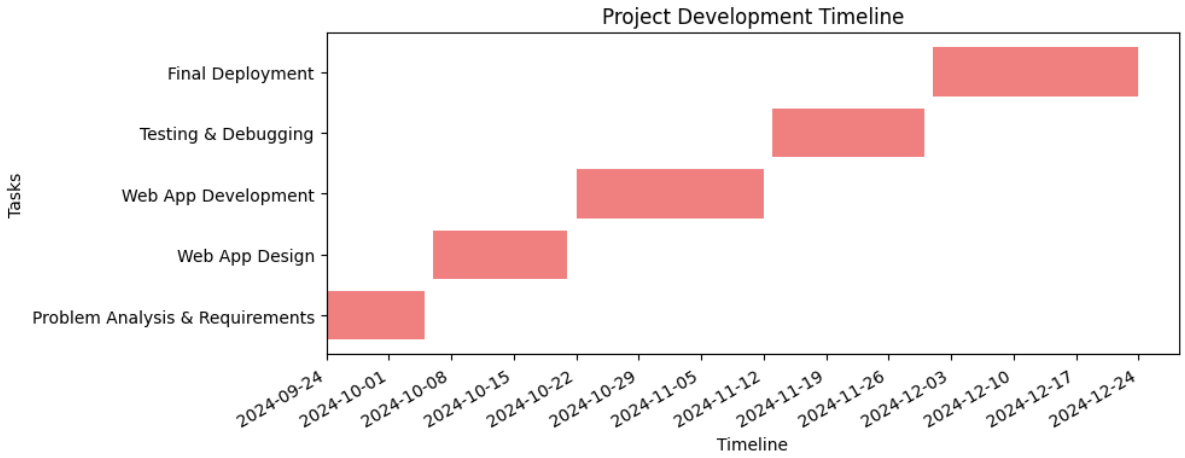
Implementation

1. **Setup and Configuration:**
    - Initialize the development environment with a version control system (Git).
    - Setup backend and frontend frameworks.
2. **User Authentication Implementation:**
    - Develop registration and login modules.
    - Integrate session management for user persistence.
3. **Dashboard and Data Retrieval:**
    - Create endpoints to fetch categories and insurance details.
    - Implement front-end components to display retrieved data dynamically.
4. **Summarization Module:**
    - Develop or integrate OpenAI model.
5. **Navigation and User Experience:**
    - Design the navbar for consistency.
    - Create event handlers for redirection and navigation controls.
6. **Testing and Deployment:**
    - **Unit Testing**: Test each module (backend APIs, frontend components).
    - **Integration Testing**: Verify end-to-end functionality.
    - **Deployment**: Deploy the application using cloud services like AWS or Azure.

CHAPTER 7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)



CHAPTER 8

OUTCOMES

- **Simplified Insurance Documents**: Automated generation of easily understandable summaries of policy documents by removing jargon and complex terms.
- **Improved Customer Engagement**: Increased policyholder interaction with documents, leading to better understanding and informed decision-making.
- **Reduced Miscommunication**: Fewer instances of misunderstandings in coverage, exclusions, and claims due to clearer information presentation.
- **Enhanced Trust**: Strengthened trust between insurers and customers, fostering long-term relationships.

CHAPTER 9

RESULTS AND DISCUSSIONS

Results

1. **User Authentication System:**
   - Successful implementation of a secure login and registration system with encrypted password storage.
   - **Results:** User data is securely stored and retrieved, ensuring privacy and accessibility.
2. **Dashboard Functionality:**
   - Implementation of a dynamic dashboard that categorizes insurance policies into home, life, car, and health.
   - **Results:** Users can seamlessly navigate through categories, and company-specific policies are displayed efficiently.
3. **Insurance Policy Summarization:**
   - OpenAI-based summarization models effectively condense detailed insurance information into concise summaries.
   - **Results:** Summaries provide accurate overviews, reducing the time users need to understand complex policies.
4. **Navigation and Redirect Functionality:**
   - Consistent navbar and redirect buttons enhance user navigation across the platform.
   - **Results:** Users experience smooth transitions between pages, reducing confusion and improving satisfaction.

Discussions

1. Impact on Users:

The platform significantly simplifies the process of understanding insurance policies by providing clear summaries. The read-aloud feature improves accessibility, making it a valuable tool for a broader audience.

2. Comparative Advantage:

- **Compared to Traditional Methods:**
    - Traditional methods require users to read lengthy, jargon-heavy documents.
    - The platform's summarization and navigation features save time and reduce cognitive load.

3. Scalability and Future Potential:

The architecture supports future enhancements, such as adding a comparison tool and personalized recommendations. These additions will further assist users in making informed decisions.

CHAPTER 10

CONCLUSION

The integration of technology into insurance processes has emerged as a powerful tool for enhancing transparency and accessibility. By scanning insurance documents and utilizing advanced technologies, insurers can improve customer experience, reduce disputes, enhance efficiency and strengthen compliance. These advancements will likely drive even greater improvements in insurance transparency, making it easier for policyholders to understand and manage their coverage.

By embracing these technological solutions, insurers can position themselves as customer centric organizations that prioritize transparency, trust, and a positive policyholder experience

REFERENCES

1. K. Ganesan *et al.*, "Text Summarization Techniques in Insurance Platforms: A Review," *Journal of Data and Information Systems*, 2018.

2. J. Kim *et al.*, "Database Integration for Secure Web Applications," in *Proceedings of Web Technologies and Systems*, 2020.

3. L. Morrison *et al.*, "Accessibility in Modern Web Applications: Best Practices," *International Journal of Human-Computer Interaction*, 2019.

4. J. Nielsen, "Web Usability and Navigation Principles," *Usability Research Lab*, 2018.

5. R. Patterson *et al.*, "Designing Effective Dashboards: A User-Centric Approach," *User Experience Journal*, 2014.

6. P. Rao *et al.*, "Digital Transformation in Insurance: Trends and Challenges," *Journal of Financial Technologies*, 2020.

7. A. Singh and N. Gupta, "Impact of Comparison Tools in Insurance Platforms," *Insurance Technology Review*, 2021.

8. L. Zhang *et al.*, "API-Driven Development: Case Studies in Insurance Platforms," in *ACM Web Science Conference Proceedings*, 2022.

APPENDIX-A

PSUEDOCODE

backend > pdfProcessor.js > ...

```
const pdf = require('pdf-parse');
const fs = require('fs');

const extractTextFromPDF = async (filePath) => {
  try {
    // Verify that the file exists
    if (!fs.existsSync(filePath)) {
      throw new Error('File does not exist');
    }

    console.log('Reading file for PDF extraction:', filePath);
    const dataBuffer = fs.readFileSync(filePath);

    // Attempt to extract text from the PDF
    const data = await pdf(dataBuffer);
    console.log('Extracted PDF data:', data.text);

    if (!data.text || data.text.trim() === '') {
      throw new Error('PDF contains no extractable text');
    }

    return data.text;
  } catch (error) {
    console.error('PDF Extraction Error:', error.message);
    throw new Error('Failed to extract text from PDF');
  }
};
```

backend > pdfSaver.js > ...

```
const PDFDocument = require('pdfkit');
const fs = require('fs');

// Function to save text to a new PDF
function saveTextToPDF(text, outputPath) {
  const doc = new PDFDocument();
  doc.pipe(fs.createWriteStream(outputPath));
  doc.fontSize(12).text(text);
  doc.end();
}

module.exports = { saveTextToPDF };
```

backend > rephraser.js > ...

```
const { OpenAI } = require('openai'); // If using OpenAI API (make sure to install the OpenA
const openai = new OpenAI({ apiKey: 'sk-Dz5KTQ8v8QLuucdhQHZQcz1ujxCQy2ZzyUd-fZwIEGT3BlbkFJTP

// Function to rephrase text
const rephraseText = async (text) => {
  try {
    // Check if the text is too long; if so, break it into chunks
    const chunkSize = 1000; // Max tokens for OpenAI, adjust as needed
    const chunks = splitTextIntoChunks(text, chunkSize);

    // Rephrase each chunk and collect the results
    let rephrasedText = '';
    for (const chunk of chunks) {
      const rephrasedChunk = await rephraseChunk(chunk);
      rephrasedText += rephrasedChunk + ' '; // Append rephrased chunk
    }

    return wrapTextInHTML(rephrasedText.trim()); // Wrap the final rephrased text in HTML fo
  } catch (error) {
    console.error("Error during rephrasing:", error);
    throw new Error("Rephrasing failed");
  }
};

// Helper function to split text into manageable chunks for rephrasing
const splitTextIntoChunks = (text, chunkSize) => {
  const words = text.split(' ');
```

backend > rephraser.js > ...

```
const splitTextIntoChunks = (text, chunkSize) => {
  let chunks = [];
  let currentChunk = [];

  for (const word of words) {
    currentChunk.push(word);
    if (currentChunk.join(' ').length >= chunkSize) {
      chunks.push(currentChunk.join(' '));
      currentChunk = [];
    }
  }
  if (currentChunk.length > 0) {
    chunks.push(currentChunk.join(' ')); // Push any remaining words
  }

  return chunks;
};

// Function to rephrase a single chunk using the OpenAI API
const rephraseChunk = async (chunk) => {
  try {
    // Make an API request to OpenAI to rephrase the chunk using the newer model gpt-3.5-tur
    const response = await openai.chat.completions.create({
      model: 'gpt-3.5-turbo', // You can change this to gpt-4 if you have access
      messages: [{
        role: 'user',
        content:`Rephrase the following text in a more easy to read format give it in html f
```

backend > rephraser.js > ...

```
46  const rephraseChunk = async (chunk) => {
49      const response = await openai.chat.completions.create({
51        messages: [{
54        }],
55        max_tokens: 1000,  // Adjust max tokens if needed
56      });
57
58      // Return the rephrased text
59      return response.choices[0].message.content.trim();
60    } catch (error) {
61      console.error("Error rephrasing chunk:", error);
62      throw new Error("Failed to rephrase chunk");
63    }
64  };
65
66  // Function to wrap rephrased text in HTML for formatting
67  const wrapTextInHTML = (text) => {
68    // Example: Add bold headings and highlight important points
69    let formattedText = text
70      .replace(/(\b[A-Z][A-Z]+\b)/g, '<b>$1</b>')  // Bold uppercase words (e.g., headings)
71      .replace(/(\b(?:important|key|crucial)\b)/gi, '<span class="highlight">$1</span>')  // H
72
73    return `<div class="gpt-response"><p>${formattedText}</p></div>`;
74  };
75
76  module.exports = { rephraseText };
```

backend > server.js > PORT

```
2   const mysql = require('mysql2');
3   const cors = require('cors');
4   const multer = require('multer');
5   const fs = require('fs');
6   const path = require('path');
7   const { extractTextFromPDF } = require('./pdfProcessor'); // Ensure this is implemented corr
8   const { rephraseText } = require('./rephraser.js');
9
10  const app = express();
11  const PORT = 5000; // Use this port to avoid conflict
12
13  const corsOptions = {
14    origin: 'http://localhost:3000', // Allow frontend to make requests
15    methods: ['GET', 'POST', 'PUT', 'DELETE'], // Allow the required HTTP methods
16    allowedHeaders: ['Content-Type', 'Authorization'], // Allow Content-Type header
17  };
18
19  // Use CORS middleware
20  app.use(cors(corsOptions));
21  app.use(express.json());
22
23  // Database connection
24  const db = mysql.createConnection({
25    host: 'localhost',
26    user: 'root',
27    password: 'root',
28    database: 'login_db',
```

E:\all_project\frontend\src\components\SummaryPage.js • Untracked

```
});

db.connect((err) => {
  if (err) {
    console.error('Database connection failed: ' + err.stack);
    return;
  }
  console.log('Connected to the database');
});

// Routes
app.post('/signup', (req, res) => {
  const sql = "INSERT INTO users (username, email, password) VALUES (?)";
  const values = [req.body.name, req.body.email, req.body.password];
  db.query(sql, [values], (err, data) => {
    if (err) {
      console.error("Error inserting data:", err);
      return res.json({ error: err.message });
    }
    return res.json({ success: true, data });
  });
});

app.post('/login', (req, res) => {
  const { email, password } = req.body;
  const sql = "SELECT * FROM users WHERE email = ? AND password = ?";
  db.query(sql, [email, password], (err, data) => {
```

backend > server.js > PORT

```
});

// Multer configuration for file uploads
const upload = multer({
  dest: 'uploads/',
  limits: { fileSize: 10 * 1024 * 1024 }, // 10 MB limit
});

// PDF Rephrase API
app.post('/api/rephrase-pdf', upload.single('pdf'), async (req, res) => {
  if (!req.file) {
    return res.status(400).json({ error: 'No file uploaded' });
  }

  const { category, provider } = req.body;

  try {
    const filePath = req.file.path;

    console.log(`Received file: ${filePath}`);

    // Extract text from the uploaded PDF
    const extractedText = await extractTextFromPDF(filePath);

    if (!extractedText || extractedText.trim() === '') {
      console.error('Error: No text extracted from PDF');
      return res.status(500).json({ error: 'No text extracted from PDF' });
```

frontend > src > components > AckoDisplay.js > RelianceDisplay

```
1
2   import { Worker, Viewer } from '@react-pdf-viewer/core';
3   import '@react-pdf-viewer/core/lib/styles/index.css';
4   import { useState } from 'react';
5   import axios from 'axios';
6
7   // Component for displaying individual PDFs
8   function PdfDisplay({ pdfUrl }) {
9       if (!pdfUrl) {
10          return <p>No PDF URL provided</p>;
11      }
12
13      return (
14          <div style={{ textAlign: 'center', margin: '20px auto' }}>
15              {/* PDF.js worker for rendering */}
16              <div style={{ height: '800px', width: '80%', marginBottom: '10px' }}>
17                  <Worker workerUrl="https://unpkg.com/pdfjs-dist@3.0.279/build/pdf.worker.min
18                      <Viewer fileUrl={pdfUrl} defaultScale={1} />
19                  </Worker>
20              </div>
21          </div>
22      );
23  }
24
25  // Main Component to display PDFs for Reliance
26  export default function RelianceDisplay() {
27      const pdfUrl = 'https://acko-cms.ackoassets.com/Stand_alone_Own_Damage_Private_Car_Polic
```

frontend > src > components > BajajDisplay.js > RelianceDisplay

```
1
2   import { Worker, Viewer } from '@react-pdf-viewer/core';
3   import '@react-pdf-viewer/core/lib/styles/index.css';
4   import { useState } from 'react';
5   import axios from 'axios';
6
7   // Component for displaying individual PDFs
8   function PdfDisplay({ pdfUrl }) {
9       if (!pdfUrl) {
10          return <p>No PDF URL provided</p>;
11      }
12
13      return (
14          <div style={{ textAlign: 'center', margin: '20px auto' }}>
15              {/* PDF.js worker for rendering */}
16              <div style={{ height: '800px', width: '80%', marginBottom: '10px' }}>
17                  <Worker workerUrl="https://unpkg.com/pdfjs-dist@3.0.279/build/pdf.worker.m
18                      <Viewer fileUrl={pdfUrl} defaultScale={1} />
19                  </Worker>
20              </div>
21          </div>
22      );
23  }
24
25  // Main Component to display PDFs for Reliance
26  export default function RelianceDisplay() {
```

frontend > src > components > Dashboard.js > Dashboard

```
import React, { useState } from 'react';
import {
    AppBar,
    Box,
    Container,
    Grid,
    Typography,
    Toolbar,
    Collapse,
    IconButton,
    Button,
    Table,
    TableBody,
    TableCell,
    TableContainer,
    TableHead,
    TableRow,
    Paper,
    Divider,
    Menu,
    MenuItem,
} from '@mui/material';
import { ExpandMore, ExpandLess } from '@mui/icons-material';

import { HealthAndSafety, DirectionsCar, House, Security } from '@mui/icons-material';
import { Link } from 'react-router-dom';
import { Carousel } from 'react-responsive-carousel';
```

frontend > src > components > Dashboard.js > Dashboard

```
function Dashboard() {
    const [expandedSection, setExpandedSection] = useState(null);
    const [anchorEl, setAnchorEl] = useState(null);
    const open = Boolean(anchorEl);

    const handleMenuClick = (event) => {


        setAnchorEl(event.currentTarget);
    };

    const handleMenuClose = () => {
        setAnchorEl(null);
    };

    const handleToggleSection = (type) => {
        setExpandedSection((prev) => (prev === type ? null : type));
    };
    const handleLogout = () => {
      // Clear authentication tokens or session data
      localStorage.removeItem('authToken'); // Example for token-based auth
      // Redirect to login page
      window.location.href = '/';
  };

```

```
frontend > src > components > Dashboard.js > Dashboard
 66  function Dashboard() {
225                 {/* Types of Insurance Section */}
226      <Box id="types-of-insurance" sx={{ padding: '40px 20px', textAlign: 'center', backgroundC
227          <Typography variant="h4" sx={{ color: '#2E8BC0', marginBottom: '30px' }}>
228              Types of Insurance
229          </Typography>
230          <Grid container spacing={3} sx={{ justifyContent: 'center' }}>
231              {insuranceTypes.map((insurance) => (
232                  <Grid item xs={12} sm={6} md={3} key={insurance.type}>
233                      <Box
234                          sx={{
235                              cursor: 'pointer',
236                              textAlign: 'center',
237                              boxShadow: 3,
238                              padding: '20px',
239                              backgroundColor: '#fff',
240                              '&:hover': { boxShadow: 5 },
241                          }}
242                          onClick={() => handleToggleSection(insurance.type)}
243                      >
244                          {insurance.icon}
245                          <Typography variant="h6" sx={{ color: '#2E8BC0' }}>
246                              {insurance.type}
247                          </Typography>
248                      </Box>
249                      <Collapse in={expandedSection === insurance.type}>
```

```
frontend > src > components > Dashboard.js > Dashboard
 66  function Dashboard() {
373          >
374            <Typography variant="h6" sx={{ color: '#7e94f0', marginBottom: '10px' }}>
375              💰 Affordable Pricing
376            </Typography>
377            <Typography
378              variant="body1"
379              sx={{ color: '#555', lineHeight: '1.6' }}
380            >
381              We help you find policies that offer great coverage at competitive prices.
382            </Typography>
383          </Box>
384
385          {/* Card 4 */}
386          <Box
387            sx={{
388              position: 'relative',
389              padding: '20px',
390              borderRadius: '8px',
391              boxShadow: '0px 4px 6px rgba(0, 0, 0, 0.1)',
392              backgroundColor: '#f5f9f6',
393              borderLeft: '5px solid #6378d1 ',
394            }}
395          >
396            <Typography variant="h6" sx={{ color: '#6378d1 ', marginBottom: '10px' }}>
397              🤝 Hassle-Free Claims
398            </Typography>
```

frontend > src > components > InsuranceTypePage.js > insuranceData > "home-insurance"

```
import React from "react";
import { useParams, Link } from "react-router-dom";
import PropTypes from "prop-types"; // Import PropTypes for type checking
import "../styles/compare.css"; // Import the CSS file

// Static data for insurance comparison
const insuranceData = {
  "health-insurance": [
    {
      company: "Reliance General Insurance Company Limited",
      coverage: [
        "In-patient treatment",
        "Day care treatment",
        "Pre and post-hospitalization",
        "Rehabilitation",
      ],
      premium: "Varies based on the plan and coverage selected.",
      deductibles: "Customizable plans with higher deductibles, ideal for low healthcare ne
      coPayments: "20% on dental claims\n"+"other co-payment terms may apply.",
      claimSettlementRatio: "98.75% ",
      waitingPeriods:
        "Specific waiting periods apply for some conditions",
      exclusions: [
        "Non-medically necessary treatments, cosmetic surgeries, and specific conditions ar
      ],
      additionalBenefits: [
        "No Claim Bonus",
```

frontend > src > components > LandingPage.js > ...

```
import React from 'react';
import { Box, Typography, Button } from '@mui/material';
import { styled } from '@mui/system';
import { Link } from 'react-router-dom';
import Carousel from 'react-material-ui-carousel';

const StyledLandingPage = styled(Box)(({ theme }) => ({
  backgroundColor: '#243B55',
  color: '#fff',
  textAlign: 'center',
  paddingBottom: '50px',
}));

const Header = styled(Box)(({ theme }) => ({
  display: 'flex',
  alignItems: 'center',
  justifyContent: 'space-between',
  padding: '20px 50px',
  backgroundColor: '#1E293A',
  color: "white"
}));

const Logo = styled(Box)(({ theme }) => ({
  display: 'flex',
  alignItems: 'center',
  gap: '10px',
  color: '#fff',
```

frontend > src > components > LandingPage.js > ...

```
67  const LandingPage = () => {
90                Login
91              </Button>
92              <Button variant="contained" color="primary" component={Link} to="/signup">
93                Sign Up
94              </Button>
95            </Box>
96          </Header>
97
98          {/* Welcome Section */}
99          <Box sx={{ padding: '50px', textAlign: 'center' }}>
100           <Typography variant="h3" component="h1" sx={{ fontWeight: 'bold', marginBottom: '16p
101             Welcome to InsureEase
102           </Typography>
103           <Typography variant="body1" sx={{ fontSize: '1.25rem', marginBottom: '24px', color:
104             Building trust with transparency. Simplifying insurance for everyone.
105           </Typography>
106         </Box>
107
108         {/* Carousel Section */}
109         <CarouselContainer />
110
111         {/* Why Us Section */}
112         <WhyUsSection>
113           <Typography variant="h4" component="h2" sx={{ fontWeight: 'bold', marginBottom: '20p
114             Why Choose Us?
```

frontend > src > components > Login.js > ...

```
1   import React, {useState} from "react";
2   import { Link, useNavigate } from "react-router-dom";
3   import Validation from "./LoginValidation";
4   import axios from 'axios'
5
6
7   function Login() {
8       const [values , setValues] = useState({
9           email: '',
10          password: ''
11      })
12
13      const navigate = useNavigate();
14      const [errors, setErrors] = useState({})
15
16      const handleInput = (event) => {
17          setValues(prev => ({...prev, [event.target.name]: [event.target.value]}))
18      }
19
20      const handleSubmit = (event) => {
21          event.preventDefault();
22
23          // Run validation
24          const validationErrors = Validation(values);
25          setErrors(validationErrors);
26
```

frontend > src > components > Login.js > ...

```
function Login() {
    const handleSubmit = (event) => {
                })
                .catch(err => console.log("Axios error:", err));
        }
    };


return (
    <div className="d-flex justify-content-center align-items-center bg-light vh-100">
        <div className="d-flex w-75">
            {/* Image Section */}
            <div className="w-50" style={{ marginRight: '0px' }}>
                <img
                    src="https://www.renewbuy.com/sites/default/files/2023-10/Asset%205%4030
                    alt="Login Illustration"
                    className="img-fluid h-100 rounded"
                    style={{ objectFit: 'cover', height: '100%' }}
                />
            </div>

            {/* Form Section */}
            <div className="bg-white p-5 rounded shadow-lg w-50 d-flex flex-column justify-c
                <div className="text-center mb-4">
                    <i className="fas fa-user-circle" style={{ fontSize: '80px', color: '#33
                </div>
                <h2 className="text-center mb-4" style={{ color: '#333', fontWeight: 'bold'
```

frontend > src > components > LoginValidation.js > Validation

```
function Validation(values) {
    let error = {};

    const email_pattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    //const password_pattern = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])[a-zA-Z0-9]{8,}$/;
    const password_pattern = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[@$!%*?&])[A-Za-z\d@$!%*?

    // Email validation
    if (values.email === "") {
        error.email = "Email should not be empty";
    } else if (!email_pattern.test(values.email)) {
        error.email = "Email didn't match";
    }

    // Password validation
    if (values.password === "") {
        error.password = "Password should not be empty";
    } else if (!password_pattern.test(values.password)) {
       error.password = "Password didn't match";
    }

    return error;
}

export default Validation;
```

frontend > src > components > ManipalDisplay.js > PdfDisplay

```
1
2   import { Worker, Viewer } from '@react-pdf-viewer/core';
3   import '@react-pdf-viewer/core/lib/styles/index.css';
4   import { useState } from 'react';
5   import axios from 'axios';
6
7   // Component for displaying individual PDFs
8   function PdfDisplay({ pdfUrl }) {
9       if (!pdfUrl) {
10          return <p>No PDF URL provided</p>;
11      }
12
13      return (
14          <div style={{ textAlign: 'center', margin: '20px auto' }}>
15              {/* PDF.js worker for rendering */}
16              <div style={{ height: '800px', width: '80%', marginBottom: '10px' }}>
17                  <Worker workerUrl="https://unpkg.com/pdfjs-dist@3.0.279/build/pdf.worker.min
18                      <Viewer fileUrl={pdfUrl} defaultScale={1} />
19                  </Worker>
20              </div>
21          </div>
22      );
23  }
24
25  // Main Component to display PDFs for Manipal Cigna
26  export default function RelianceDisplay() {
27      const pdfUrl = 'https://www.manipalcigna.com/documents/20124/0/Arogya%20Sanjeevani_Accor
```

frontend > src > components > MaxDisplay.js > RelianceDisplay

```
26  export default function RelianceDisplay() {
63      const buttonStyle = {
70          cursor: 'pointer',
71          textDecoration: 'none',
72      };
73
74      return (
75          <div>
76              <a
77      href="https://drive.google.com/file/d/1nOeQBQcLDq2Cw6Yfarw1hK70ywX-strj/edit" // Direct
78      download="maxlife-policy.pdf" // Desired file name for the download
79      style={buttonStyle}
80  >
81      Download PDF
82  </a>
83              <h1>Max Life</h1>
84              <PdfDisplay pdfUrl={pdfUrl} />
85              {/* <h2>PDF Rephraser</h2> */}
86              <input type="file" accept="application/pdf" onChange={handleFileChange} />
87              <button onClick={handleRephrase}>Summarize PDF</button>
88              {error && <p style={{ color: 'red' }}>{error}</p>}
89              {rephrasedText && (
90                  <div>
91                      <h3>Summarized Content:</h3>
92                      <div
93                          style={{ textAlign: 'left', margin: '20px' }}
94                          dangerouslySetInnerHTML={{ __html: rephrasedText }} // Render HTML
```

frontend > src > components > PdfRephraser.js > default

```
import React, { useState } from 'react';
import axios from 'axios';
const PdfRephraser = () => {
  const [pdfFile, setPdfFile] = useState(null);
  const [rephrasedText, setRephrasedText] = useState('');
  const [error, setError] = useState(null);

  const handleFileChange = (e) => {
    setPdfFile(e.target.files[0]);
  };

  const handleRephrase = async () => {
    if (!pdfFile) {
      console.log('No PDF file selected');
      return;
    }

    console.log('PDF file selected, preparing to send request');
    const formData = new FormData();
    formData.append('pdf', pdfFile);

    try {
      const response = await axios.post('http://localhost:5000/api/rephrase-pdf', formData,
        headers: {
          'Content-Type': 'multipart/form-data',
        },
      });
```

frontend > src > components > Signup.js > ...

```
import React,{useState}from "react";
import { Link, useNavigate } from "react-router-dom";
import Validation from "./SignupValidation";
import axios from 'axios'


function Signup(){

    const [values , setValues] = useState({
        name: '',
        email: '',
        password: ''
    })

    const navigate = useNavigate();
    const [errors, setErrors] = useState({})

    const handleInput = (event) => {
        setValues(prev => ({...prev, [event.target.name]: event.target.value}))
    }


    const handleSubmit = (event) => {
        event.preventDefault();

        // Run validation and store results in a variable
        const validationErrors = Validation(values);
```

frontend > src > components > SignupValidation.js > ...

```
function Validation(values) {
    let error = {};

    const email_pattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    //const password_pattern = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])[a-zA-Z0-9]{8,}$/;
    const password_pattern = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&

    // Name validation
    if (values.name === "") {
        error.name = "Name should not be empty";
    }

    // Email validation
    if (values.email === "") {
        error.email = "Email should not be empty";
    } else if (!email_pattern.test(values.email)) {
        error.email = "Invalid Email ";
    }

    // Password validation
    if (values.password === "") {
       error.password = "Password should not be empty";
    } else if (!password_pattern.test(values.password)) {
       error.password = "Invalid Password ";
```

BLACKBOX Chat | Add Logs | CyberCoder | Improve Code | Share Code Link | Search Error | JavaScript | Go Live | AI Code Cha

frontend > src > components > SummaryPage.js > ...

```
import React, { useEffect, useState } from 'react';
import { useParams } from 'react-router-dom';
import axios from 'axios';

const SummaryPage = () => {
  const { category, provider } = useParams();
  const [summary, setSummary] = useState('Loading...');
  const [error, setError] = useState(null);

  useEffect(() => {
    const fetchSummary = async () => {
      try {
        const response = await axios.post('http://localhost:8081/api/rephrase-pdf', {
          category,
          provider,
        });
        setSummary(response.data.rephrasedText);
      } catch (err) {
        setError('Failed to fetch the summary.');
      }
    };

    fetchSummary();
  }, [category, provider]);

  if (error) {
    return <p style={{ color: 'red' }}>{error}</p>;
```

frontend > src > components > Support.js > ...

```
// src/Support.js
import React from 'react';
import { Typography, Container, Button } from '@mui/material';  // Added Button import
import { Link } from 'react-router-dom';  // Added Link import

function Support() {
    return (
        <Container sx={{ marginTop: '30px', textAlign: 'center' }}>
            <Typography variant="h4" sx={{ marginBottom: '20px', color: '#2E8BC0' }}>
                Support
            </Typography>
            <Typography variant="h6" sx={{ color: '#333' }}>
                For Troubleshooting, Please Contact Us:
            </Typography>
            <Typography variant="h6" sx={{ color: '#2E8BC0' }}>
                +1-800-123-4567
            </Typography>
            <Typography variant="body1" sx={{ marginTop: '20px', color: '#555' }}>
                Our support team is available 24/7 to help you with any issues you may face.
            </Typography>
            <Button
                variant="contained"
                sx={{ backgroundColor: '#2E8BC0', color: '#fff', '&:hover': { backgroundColo
                component={Link}
                to="/dashboard"
            >
                Go to Home
```

frontend > src > App.js > App

E:\all_project\frontend • Contains emphasized items

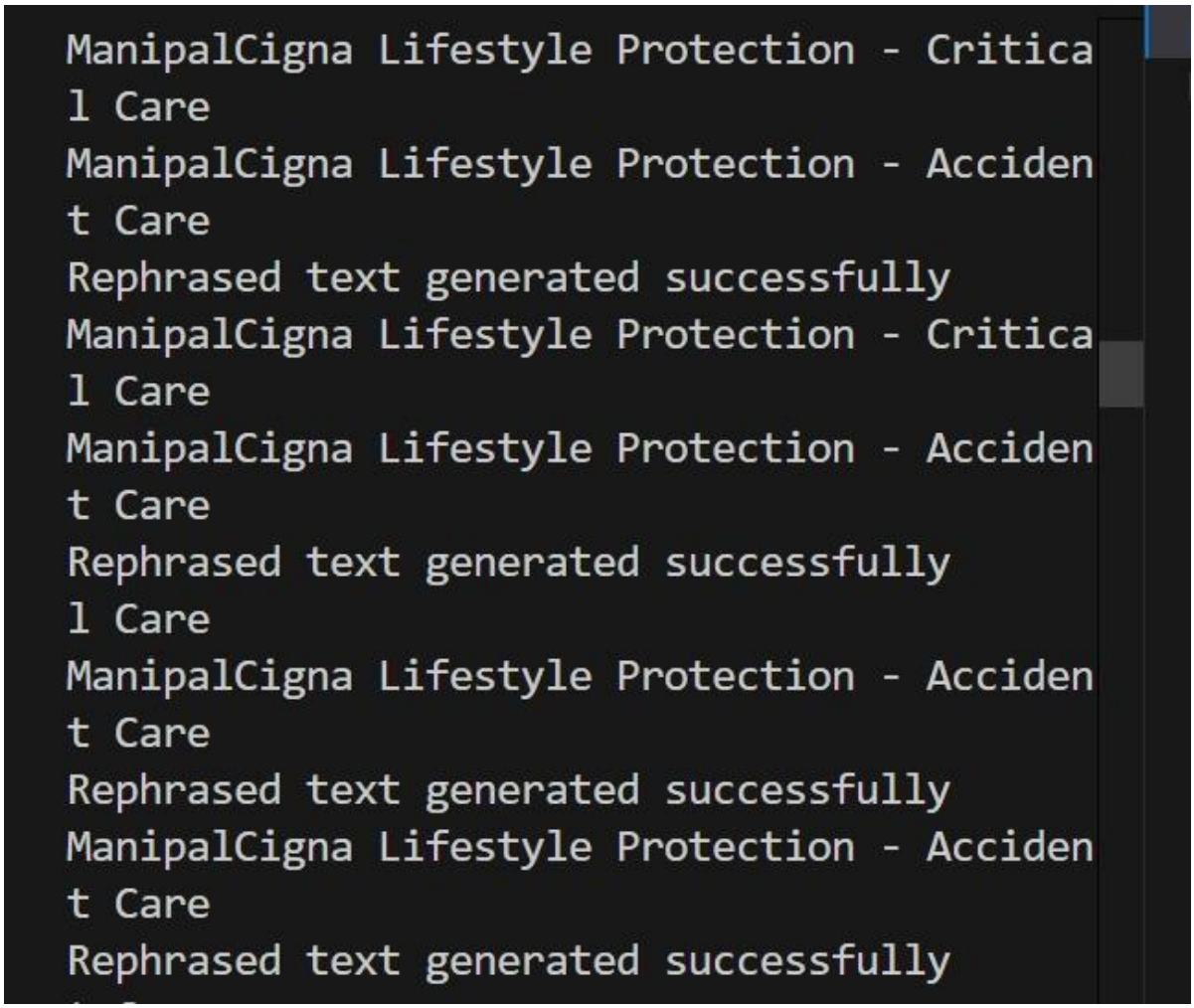
```
./components/PolicyBazaarDisplay.js';
import SBILifeDisplay from './components/SBILifeDisplay.js';
import RelianceDisplay from './components/RelianceDisplay.js';
import BajajDisplay from './components/BajajDisplay.js';
function App() {
return (
  <BrowserRouter>
    <Routes>
      <Route path = '/' element = {<LandingPage />}></Route>
      <Route path = '/login' element = {<Login />}></Route>
      <Route path = '/signup' element = {<Signup />}></Route>
      <Route path = '/dashboard' element = {<Dashboard />}></Route>
      <Route path="/support" element={<Support />} />
      <Route path="/landingpage" element={<LandingPage />} />
      <Route path='/company/health-co-1' element={<RelianceDisplay />}></Route>
      <Route path="/insurance-type/:type" element={<InsuranceTypePage/>}></Route>
      <Route path="/reliance-text-summarizer" element={<RelianceTextSummarizer />} />
      <Route path='/company/health-co-2' element={<ManipalDisplay />}></Route>
      <Route path='/company/car-co-1' element={<AckoDisplay />}></Route>
      <Route path='/company/car-co-2' element={<BajajDisplay />}></Route>
      <Route path='/company/home-co-1' element={<PolicyBazaarDisplay />}></Route>
      <Route path='/company/home-co-2' element={<SBIDisplay />}></Route>
      <Route path='/company/life-co-1' element={<MaxDisplay />}></Route>
      <Route path='/company/life-co-2' element={<SBILifeDisplay />}></Route>
    </Routes>
  </BrowserRouter>
  )
```

frontend > src > components > Dashboard.js > Dashboard

```
28  import 'react-responsive-carousel/lib/styles/carousel.min.css';
29
30  // Insurance Types and Companies Data
31  const insuranceTypes = [
32      {
33          type: 'Health Insurance',
34          icon: <HealthAndSafety />,
35          companies: [
36              { name: 'Reliance General', logo: 'health_logo_1.png', link: '/company/health-co
37              { name: 'Manipal Cigna', logo: 'health_logo_2.png', link: '/company/health-co-2'
38          ],
39      },
40      {
41          type: 'Car Insurance',
42          icon: <DirectionsCar />,
43          companies: [
44              { name: 'Acko', logo: 'car_logo_1.png', link: '/company/car-co-1' },
45              { name: 'Bajaj Allianz', logo: 'car_logo_2.png', link: '/company/car-co-2' },
46          ],
47      },
48      {
49          type: 'Home Insurance',
50          icon: <House />,
51          companies: [
52              { name: 'Policy Bazaar', logo: 'home_logo_1.png', link: '/company/home-co-1' },
53              { name: 'SBI General', logo: 'home_logo_2.png', link: '/company/home-co-2' },
54          ],
```

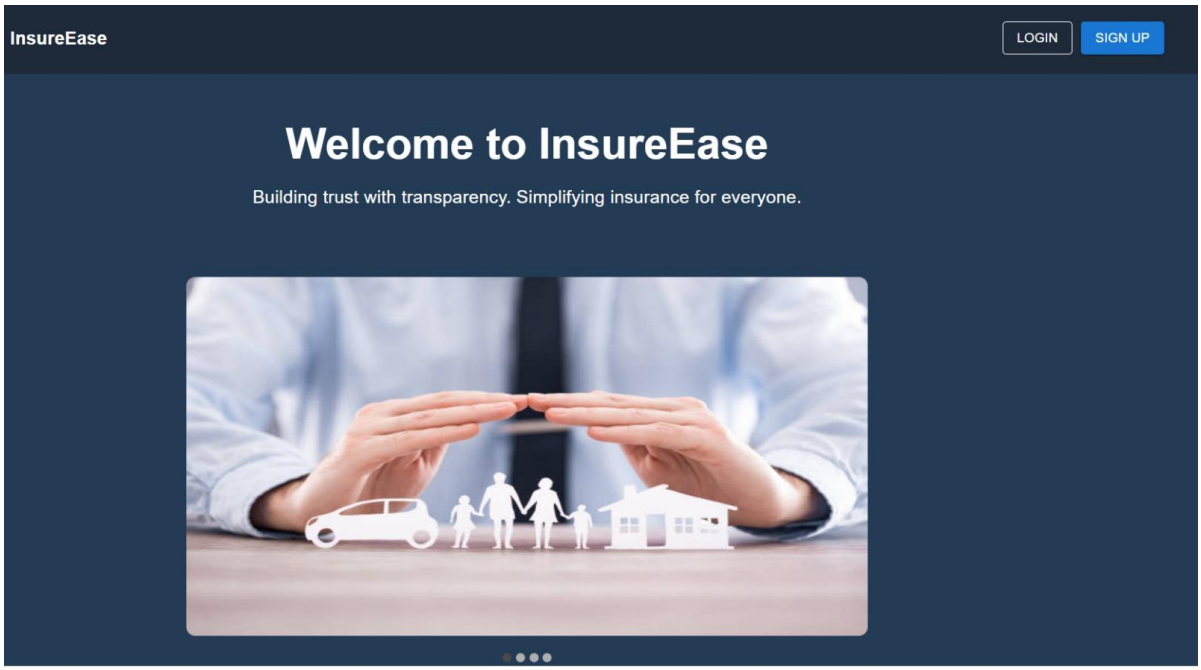
frontend > src > components > Navbar.js > ...

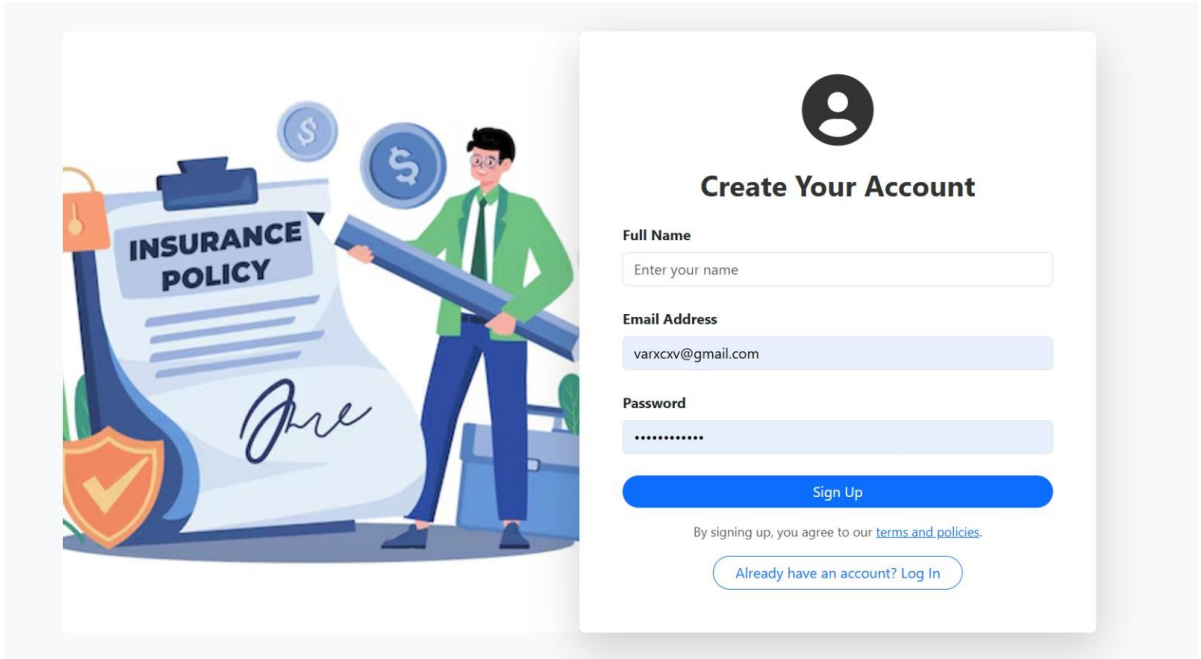
```
1   import React from 'react';
2   import { Link } from 'react-router-dom';
3   import { AppBar, Toolbar, Button, Box } from '@mui/material';
4
5   const Navbar = () => {
6       return (
7           <AppBar position="sticky" sx={{ backgroundColor: '#2E8BC0' }}>
8               <Toolbar>
9                   <Box sx={{ flexGrow: 1 }}>
10                      <Button color="inherit" component={Link} to="/">Home</Button>
11                  </Box>
12                  <Button color="inherit" component={Link} to="/login">Login</Button>
13                  <Button color="inherit" component={Link} to="/signup">Sign Up</Button>
14              </Toolbar>
15          </AppBar>
16      );
17  };
18
19  export default Navbar;
20
```

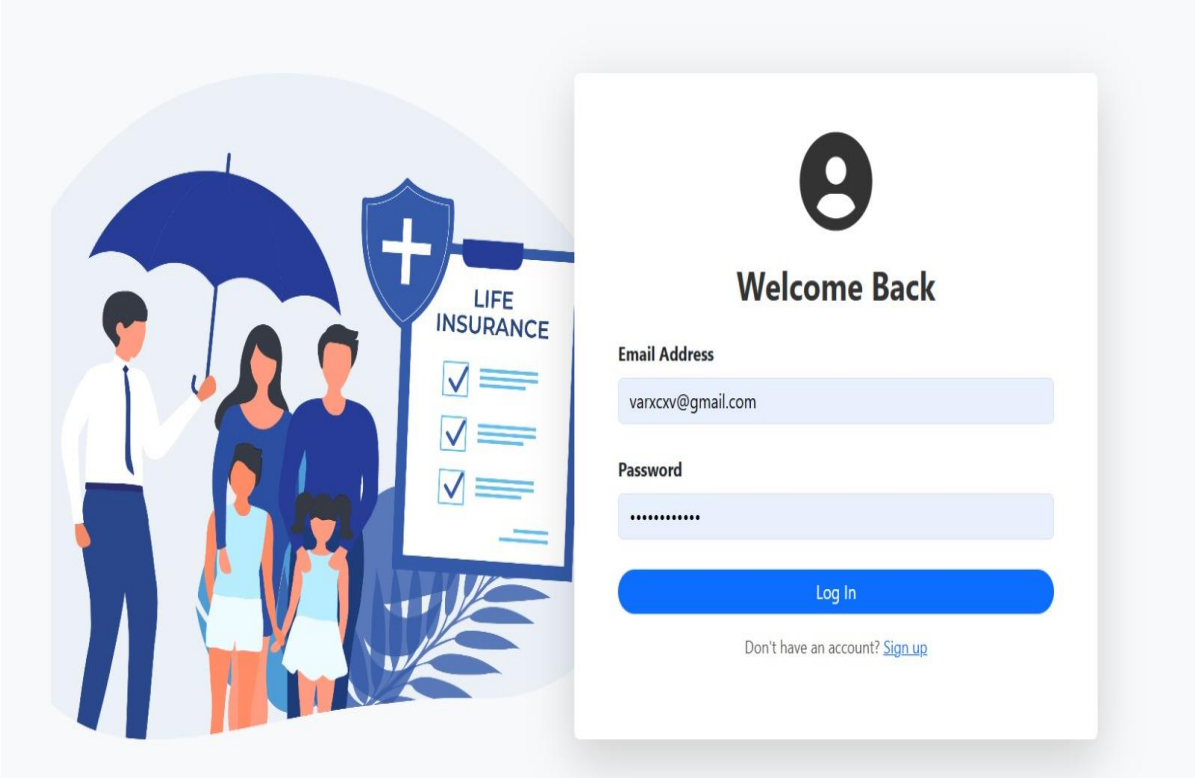

ManipalCigna Lifestyle Protection - Critica
l Care
ManipalCigna Lifestyle Protection - Acciden
t Care
Rephrased text generated successfully
ManipalCigna Lifestyle Protection - Critica
l Care
ManipalCigna Lifestyle Protection - Acciden
t Care
Rephrased text generated successfully
l Care
ManipalCigna Lifestyle Protection - Acciden
t Care
Rephrased text generated successfully
ManipalCigna Lifestyle Protection - Acciden
t Care
Rephrased text generated successfully

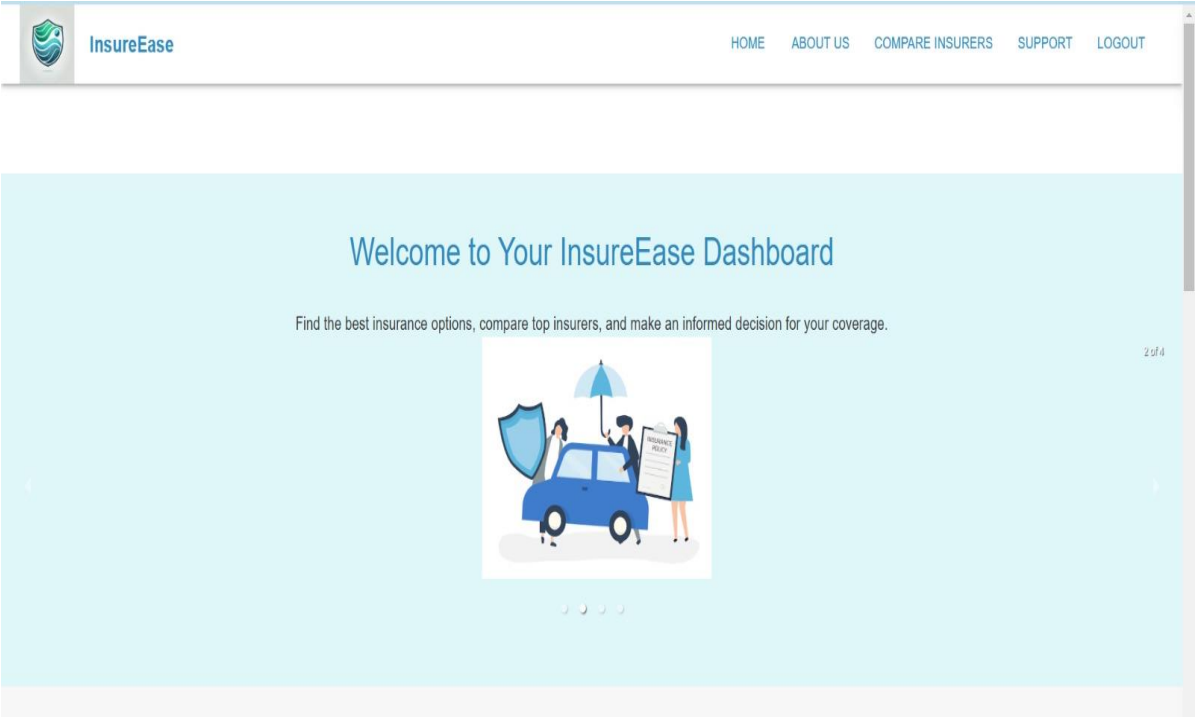
APPENDIX-B

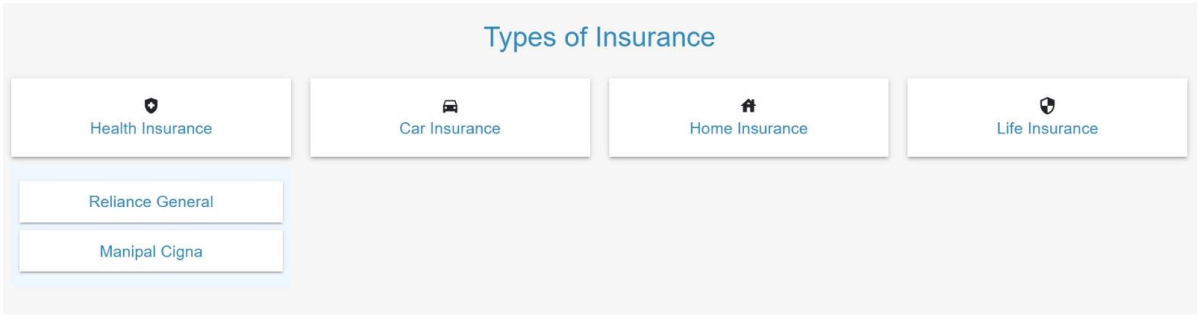
SCREENSHOTS

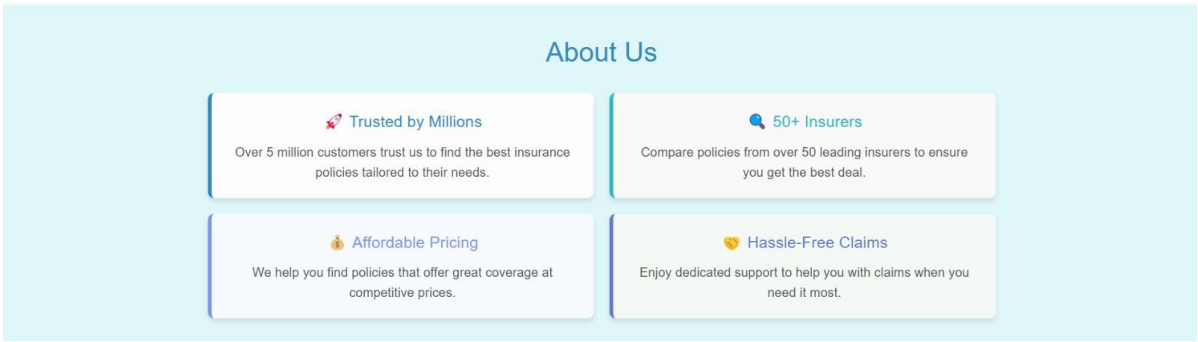


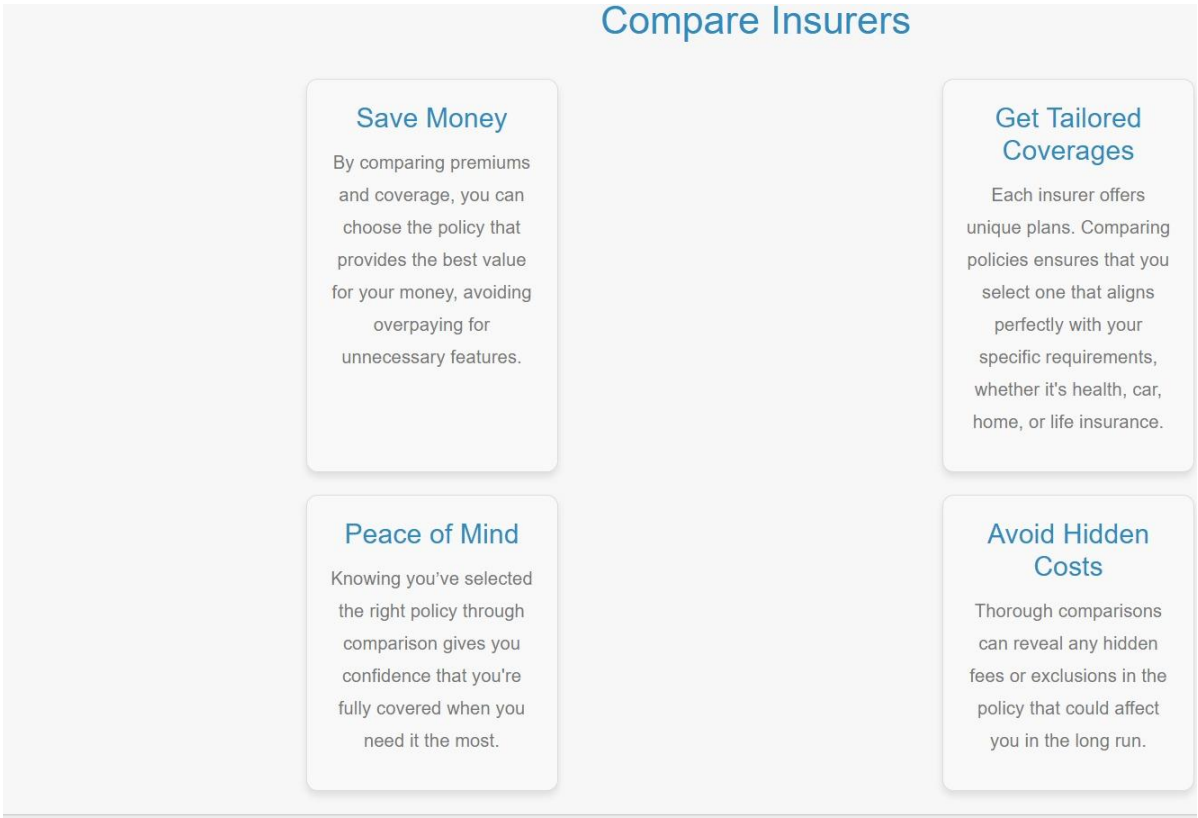



LIFE
INSURANCE
Welcome Back
Email Address
varxcxv@gmail.com
Password
Log In
Don't have an account? Sign up


InsureEase
HOME
ABOUT US
COMPARE INSURERS
SUPPORT
LOGOUT
Welcome to Your InsureEase Dashboard
Find the best insurance options, compare top insurers, and make an informed decision for your coverage.


Types of Insurance
Health Insurance
Car Insurance
Home Insurance
Life Insurance
Reliance General
Manipal Cigna


About Us
Trusted by Millions
Over 5 million customers trust us to find the best insurance policies tailored to their needs.
50+ Insurers
Compare policies from over 50 leading insurers to ensure you get the best deal.
Affordable Pricing
We help you find policies that offer great coverage at competitive prices.
Hassle-Free Claims
Enjoy dedicated support to help you with claims when you need it most.


Compare Insurers
Save Money
By comparing premiums and coverage, you can choose the policy that provides the best value for your money, avoiding overpaying for unnecessary features.
Get Tailored Coverages
Each insurer offers unique plans. Comparing policies ensures that you select one that aligns perfectly with your specific requirements, whether it's health, car, home, or life insurance.
Peace of Mind
Knowing you've selected the right policy through comparison gives you confidence that you're fully covered when you need it the most.
Avoid Hidden Costs
Thorough comparisons can reveal any hidden fees or exclusions in the policy that could affect you in the long run.

HEALTH INSURANCE COMPARISION

Home

Property	Reliance General Insurance Company Limited	ManipalCigna Health Insurance Company.
Coverage	In-patient treatment, Day care treatment, Pre and post-hospitalization, Rehabilitation	Arogya Sanjeevani Health Insurance Policy.
Premium	Varies based on the plan and coverage selected.	Offers a variety of health insurance plans with different premium features.
Rating	4.3/5	4.2/5
Deductibles	Customizable plans with higher deductibles, ideal for low healthcare needs.	Offers flexibility, mainly suited for comprehensive coverage and employer-based group plans.
Co-Payments	20% on dental claims other co-payment terms may apply.	A fixed co-pay of 5% applicable across all ages for every claim.
Claim Settlement Ratio	98.75%	99.96%
Waiting Periods	Specific waiting periods apply for some conditions	It offers flexibility in reducing waiting times for pre-existing conditions, depending on the plan.
Exclusions	Non-medically necessary treatments, cosmetic surgeries, and specific conditions are excluded.	Maternity and related expenses are not covered
Additional Benefits	No Claim Bonus, Inflation protection, Unlimited reinstatement	In-patient hospitalization, pre and post-hospitalization, daycare treatment and modern treatments.

CAR INSURANCE COMPARISION

Home

Property	Acko General Insurance Limited	Bajaj Allianz General Insurance Co. Ltd.
Coverage	Coverage for own vehicle damage, personal accident for owner-driver, theft, fire, and specified risks.	Own Damage covers accidental car damage from fire, theft, and natural disasters.
Premium	The premium varies based on the vehicle and chosen coverage options.	Determined based on the information provided in the Private Car Package Policy Proposal Form.
Rating	4.7/5	4.4/5
Deductibles	The insured must pay a deductible as specified in the schedule for each claim under Section I.	Has both compulsory and voluntary deductibles.
Co-Payments	Acko prioritizes minimal to zero co-payment across its products.	Bajaj Allianz offers flexibility with optional co-payment for voluntary deductibles.
Claim Settlement Ratio	86.87%	98.50%
Waiting Periods	Offers flexibility to reduce pre-existing condition waiting periods based on the plan.	Not applicable as this is a car insurance policy.
Exclusions	Excludes loss from mechanical breakdown, tyre damage unless with vehicle damage, and claims under intoxication.	Exclusions include damage outside the geographical area, use of the car beyond policy, consequential loss.
Additional	Includes personal accident cover for the owner-driver, accessory coverage, and options	Optional extensions such as loss of accessories, legal liability to paid driver/cleaner, and

HOME INSURANCE COMPARISION

Home

Property	SBI General Insurance Company Limited	Bajaj Allianz General Insurance Company Ltd.
Coverage	Covers fire, lightning, explosion, riots, storms, floods, and more.	It covers damage to the home building, contents, and offers optional covers for valuables and personal accident.
Premium	Varies based on the insured value and coverage options.	The premium amount is not specified; it varies based on the insured value and coverage options.
Rating	4.4/5	4.5/5
Deductibles	Vary by policy.	Fixed amount payable by the policyholder before the insurer settles the claim.
Co-Payments	10% co-pay applies to admissible claims for non-network hospitalization.	A percentage of the claim amount borne by the policyholder, as per policy terms.
Claim Settlement Ratio	95%	High claim settlement ratio, indicating efficient and reliable claim processing.
Waiting Periods	30-day waiting period for general illnesses, 12 months for specific diseases, and 36 months for pre-existing conditions.	Waiting periods vary depending on the policy, often between 30 to 90 days.
Exclusions	Losses from official orders, unoccupied homes, livestock, vehicles, securities.	Loss or damage caused by deliberate acts., War, invasion, or civil commotion., Pollution or contamination unless resulting from an insured event., Loss or damage due to mechanical breakdown or electrical failure.
Additional	Personal accident cover, Key replacement and All risk cover for portable	Loss of rent and rent for alternative accommodation., Coverage for architect's, surveyor's, and

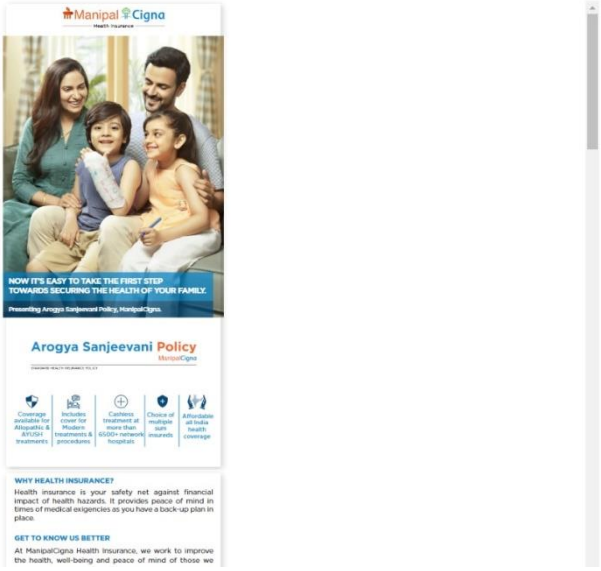
LIFE INSURANCE COMPARISION

Home

Property	Max Life Insurance Company Limited	SBI Life Insurance Company Limited
Coverage	Provides life insurance coverage with options for death benefits, critical illness benefits, and additional riders.	description: Life insurance cover with return of premium., optionalRider: Optional accident benefit rider available.
Premium	The premium varies based on the sum assured, age, and selected options. Minimum annual premium is ₹2,200.	Example premium for a ₹1 crore sum assured is ₹22,284 per annum (excluding taxes).
Rating	4.6/5	4.7/5
Deductibles	Policyholder may need to pay an amount before coverage kicks in.	, deductibles are usually based on the specific plan chosen, but not commonly applicable to life insurance products.
Co-Payments	Co-payment percentages may vary depending on the plan and the terms selected by the policyholder.	Co-payments apply primarily to health insurance policies and vary by plan.
Claim Settlement Ratio	Claim settlement ratio is around 99%, demonstrating a high success rate in claim processing.	Claim settlement ratio is approximately 96-97%, reflecting efficient and reliable claim settlement.
Waiting Periods	A waiting period of 90 days applies for the Critical Illness benefit.	No waiting period for life insurance policies, except for specific riders like critical illness.
Exclusions	Pre-existing conditions., Suicide within the first year., Claims arising from war or criminal acts.	Death due to suicide within the first 12 months., Other exclusions as per the policy terms.

Manipal Cigna

Download PDF



Arogya Sanjeevani Health Insurance Plan by ManipalCigna

The Arogya Sanjeevani health insurance policy by ManipalCigna provides financial backup during medical emergencies.

Benefits of Arogya Sanjeevani Health Insurance Plan:

- Covers hospitalisation needs from illness and injury
- Offers cashless network hospitals
- Competitive policy premium

Payment and Grace Periods:

For yearly, half yearly, and quarterly payments, a grace period of 30 days is allowed. For monthly payments, a grace period of 15 days is allowed.

Free Look Period:

The Free Look Period is applicable at the inception of the policy.

Insured individuals have 30 days from receiving the policy to review the terms and conditions and return it if not acceptable.

DOCTYPE html>

Why Buy Arogya Sanjeevani Health Insurance Policy?

Increasing Health Conditions and Healthcare Expenses

The onset of various health conditions is increasing amongst people in India. Healthcare expenses are also on the rise. Having a basic health insurance policy is the key to smart financial planning.

Benefits of Arogya Sanjeevani Policy

- Cost-effective option covering both Covid and other illnesses
- One of the most affordable health insurance policies in the market
- Sum insured options from Rs. 50,000 to 10 Lakhs

Refund Policy

Q.19) What is the Grace Period for Arogya Sanjeevani Policy?

For Yearly, half yearly and quarterly payment mode, a fixed period of 30 days will be allowed as Grace Period. For monthly mode of payment, a fixed period of 15 days will be allowed as Grace Period.

Follow Us

Contact Us

- Callers from India: Toll-free number 1800-102-4462
- Callers outside India: +91 22 4985 4100 (Call charges as per the caller's tariff plan will apply)

Email: For any queries email us at customercare@manipalcigna.com and for any policy alterations email us

DOCTYPE html>

Contact Email:

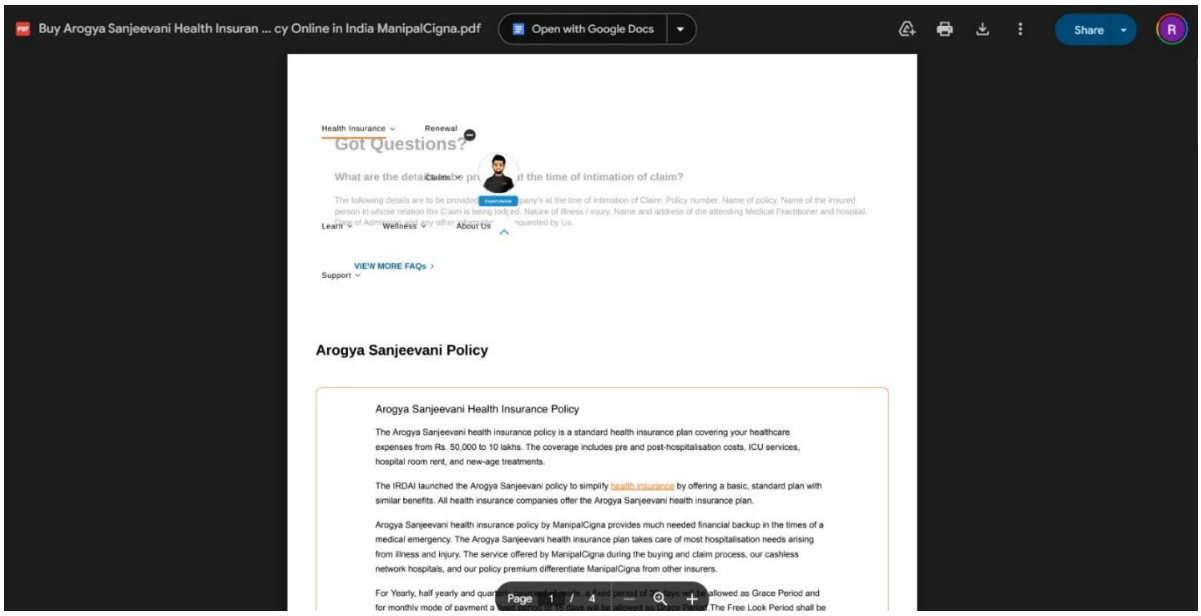
mychangerequest@manipalcigna.com

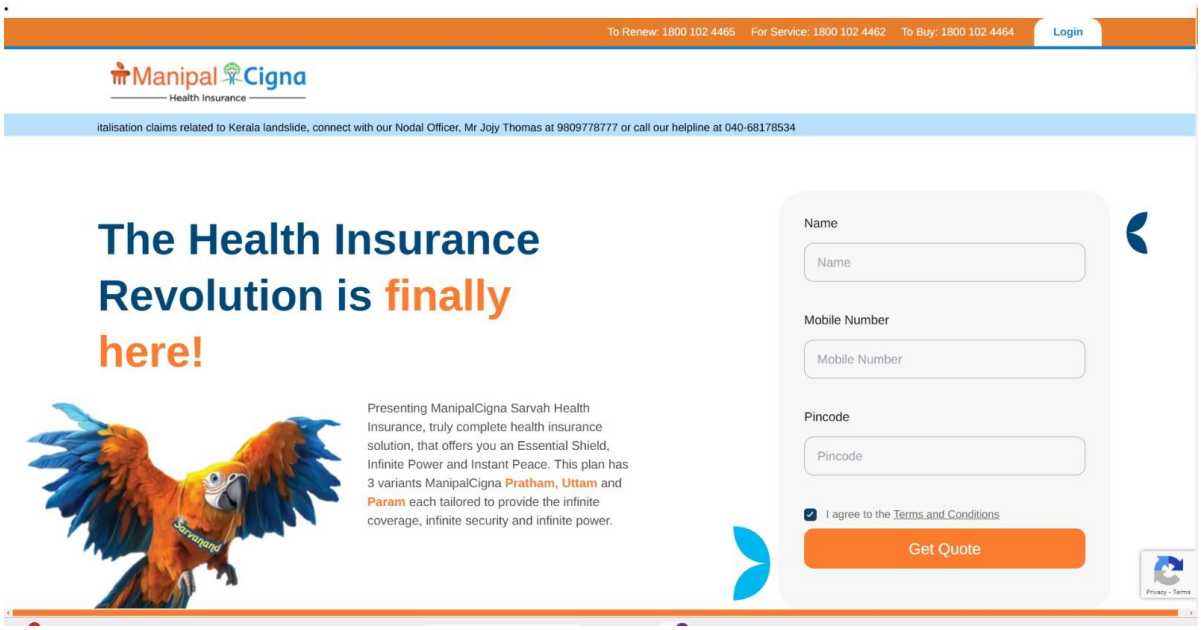
Download myManipalCigna App

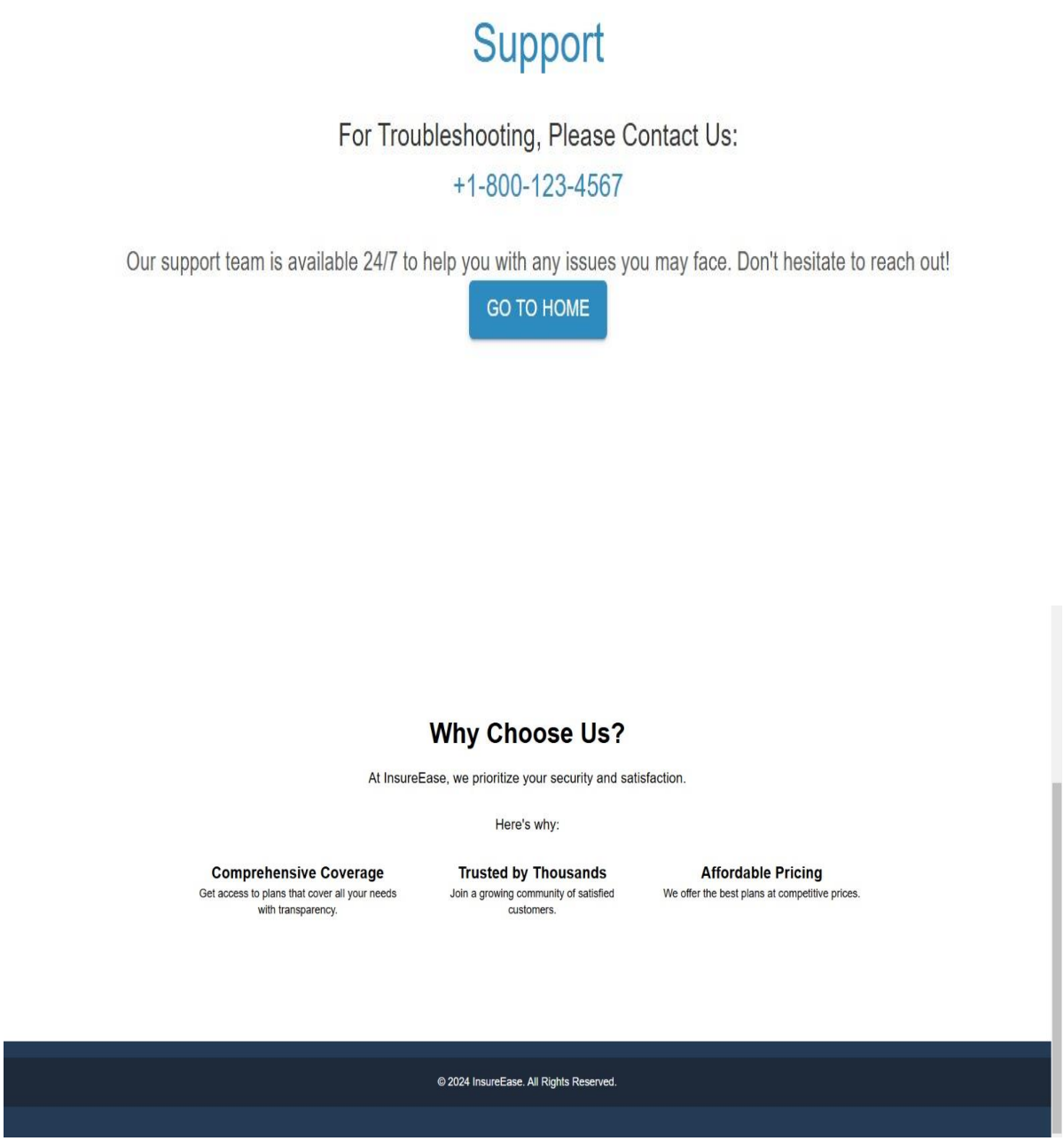
Health Insurance Plans:

- ManipalCigna ProHealth Prime
- ManipalCigna Lifetime Health
- ManipalCigna ProHealth Insurance
- ManipalCigna Super Top Up
- ManipalCigna Prime Senior
- ManipalCigna ProHealth Select
- ManipalCigna ProHealth Cash
- Arogya Sanjeevani Policy, ManipalCigna
- ManipalCigna Lifestyle Protection - Critical Care
- ManipalCigna Lifestyle Protection - Accident Care

Buy Insurance


Buy Arogya Sanjeevani Health Insuran ... cy Online in India ManipalCigna.pdf
Open with Google Docs
Share
Got Questions?
VIEW MORE FAQs
Arogya Sanjeevani Policy
Arogya Sanjeevani Health Insurance Policy
The Arogya Sanjeevani health insurance policy is a standard health insurance plan covering your healthcare expenses from Rs. 50,000 to 10 lakhs. The coverage includes pre and post-hospitalisation costs, ICU services, hospital room rent, and new-age treatments.
The IRDAI launched the Arogya Sanjeevani policy to simplify health insurance by offering a basic, standard plan with similar benefits. All health insurance companies offer the Arogya Sanjeevani health insurance plan.
Arogya Sanjeevani health insurance policy by ManipalCigna provides much needed financial backup in the times of a medical emergency. The Arogya Sanjeevani health insurance plan takes care of most hospitalisation needs arising from illness and injury. The service offered by ManipalCigna during the buying and claim process, our cashless network hospitals, and our policy premium differentiate ManipalCigna from other insurers.
Page 1 / 4


To Renew: 1800 102 4465
For Service: 1800 102 4462
To Buy: 1800 102 4464
Login
ManipalCigna
Health Insurance
italisation claims related to Kerala landslide, connect with our Nodal Officer, Mr Joiy Thomas at 9809778777 or call our helpline at 040-68178534
The Health Insurance Revolution is finally here!
Presenting ManipalCigna Sarvah Health Insurance, truly complete health insurance solution, that offers you an Essential Shield, Infinite Power and Instant Peace. This plan has 3 variants ManipalCigna Pratham, Uttam and Param each tailored to provide the infinite coverage, infinite security and infinite power.
Name
Mobile Number
Pincode
I agree to the Terms and Conditions
Get Quote


Support
For Troubleshooting, Please Contact Us:
+1-800-123-4567
Our support team is available 24/7 to help you with any issues you may face. Don't hesitate to reach out!
GO TO HOME
Why Choose Us?
At InsureEase, we prioritize your security and satisfaction.
Here's why:
Comprehensive Coverage
Get access to plans that cover all your needs with transparency.
Trusted by Thousands
Join a growing community of satisfied customers.
Affordable Pricing
We offer the best plans at competitive prices.
© 2024 InsureEase. All Rights Reserved.