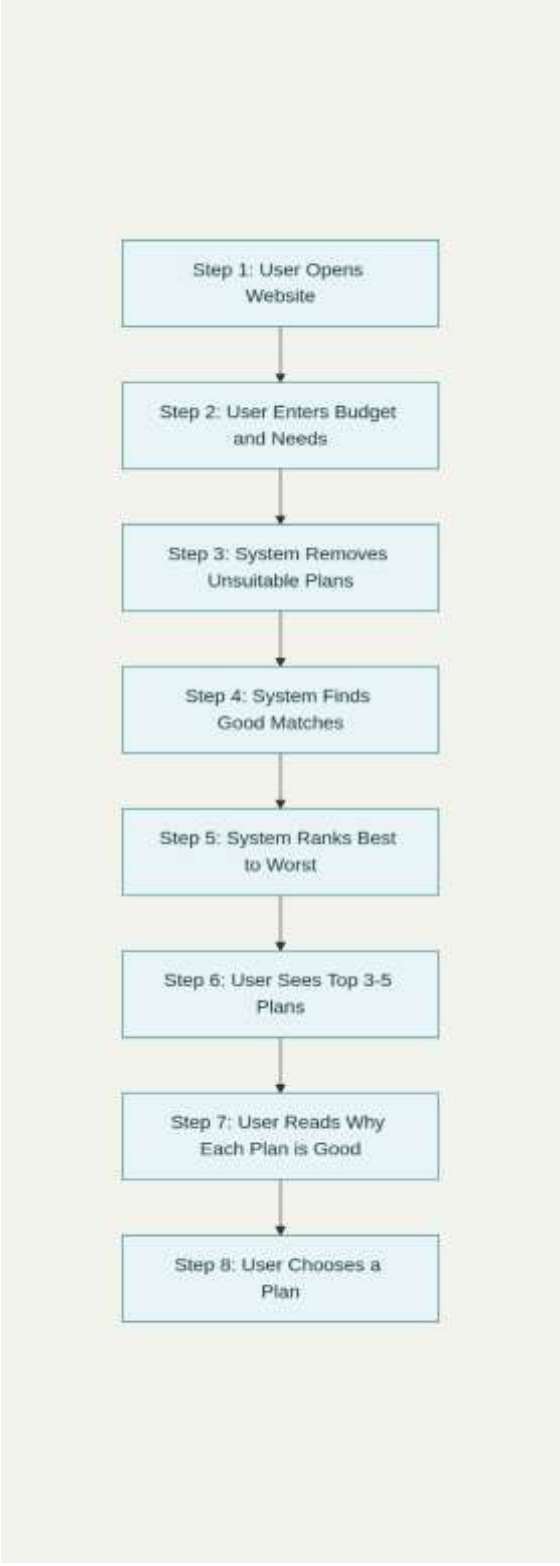
Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: CP	Aim: Documentation	
	Date: 25-09-2025	**Enrolment No: 92310133002**

Technical Report (2–3 pages)

Overview

Finsecure is a Vite + React + TypeScript web app that delivers explainable, privacy-aware insurance recommendations across life, health, and general insurance. The system integrates a lightweight knowledge graph for domain constraints, hybrid retrieval/ranking for personalization, and a transparent decision layer with multi-criteria decision-making (MCDM). A local-first privacy posture minimizes server-side exposure; optional federated learning (FL) can be adopted for cohort improvements.

Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: CP	Aim: Documentation	
	Date: 25-09-2025	**Enrolment No: 92310133002**



Key components:

- UI/Interaction: Built with React, shadcn-ui, Tailwind; routes under src/pages and reusable UI in src/components/ui.
- Knowledge Graph (conceptual): Encodes product constraints and relationships for pre-filtering and auditability.
- Hybrid Ranking: Combines structured features and learned embeddings to address cold-start and relevance.
- MCDM Layer: Applies user weights or archetypes to surface transparent trade-offs.
- Explainability: Generates layperson-readable reason codes and sensitivity sliders.
- Privacy: Defaults to client-side state; optional FL pathway if enabled server-side.

Implementation Highlights

- Frontend: Vite + React + TS; shadcn-ui components for accessible, consistent UI; Tailwind for utility styling.
- Pages: Home, Compare, Calculator, and insurance-specific pages under src/pages.
- State: Local state/hooks (src/hooks) for mobile and toast interactions; easily extended for preferences.
- Utilities: src/lib/utils.ts for class merging and helpers.
- Build tooling: vite.config.ts, Tailwind config, ESLint.

Key Outcomes (example KPIs to report)

- Faster task completion: users identify suitable plans within N minutes on average.
- Higher decision confidence: users rate explanations ≥ X/5.
- Privacy preference: ≥ Y% of users remain on default local-first mode.

User Manual (1–2 pages)

Primary Use Case: Compare Health Insurance Plans

1. Open the web app in a modern browser.
2. Navigate to Compare from the header.
3. Enter preferences (budget, deductible tolerance, network breadth) or select an archetype (e.g., Balanced).
4. Review recommended plans. Click a plan to see explanation reasons and what-if sliders.
5. Adjust weights to observe changes and finalize selection.

Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: CP	Aim: Documentation	
	Date: 25-09-2025	**Enrolment No: 92310133002**

Basic Navigation

- Header: Access Home, Compare, Calculator, About, and contact links.
- Calculator: Estimate premiums or coverage trade-offs (demo values included).
- Toasts and Dialogs: The UI provides confirmations and hints using shadcn-ui patterns.

Screenshot/Diagram

Include a screenshot of the Compare page or the architecture diagram above when exporting to PDF. If capturing a screenshot, use 1280×720 resolution for clarity.

Troubleshooting

- No results after filtering: Loosen constraints or reset weights to Balanced.
- UI components not loading: Hard refresh the browser (Ctrl/Cmd+Shift+R).
- Layout issues on mobile: Toggle the mobile menu; ensure a modern browser version.
- Build errors locally: Run npm i, then npm run dev; see dependencies in package.json.

Code Documentation (≈1 page)

Codebase Summary

- src/main.tsx: App bootstrap and provider wiring.
- src/App.tsx: Top-level routing and layout shell.
- src/pages/*: Page-level components (Home, Compare, Calculator, etc.).
- src/components/ui/*: Reusable shadcn-ui components (buttons, dialogs, tables, forms, etc.).
- src/components/Header.tsx: Navigation header.
- src/hooks/*: Custom hooks (use-mobile, use-toast).
- src/lib/utils.ts: Utility helpers (e.g., class name merging).
- src/assets/*: Static assets (e.g., hero background image).

Key Modules/Functions

- UI components expose props typed with TypeScript; prefer controlled components for form elements.
- Hooks return typed values and callbacks; use-toast demonstrates notification patterns.
- Pages are functional components receiving no global state by default; extend via context or local reducers.

Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: CP	Aim: Documentation	
	Date: 25-09-2025	**Enrolment No: 92310133002**

Dependencies

- Runtime: React, React DOM, shadcn-ui components, Tailwind CSS.
- Tooling: Vite, TypeScript, ESLint.

Build & Run

npm i

npm run dev