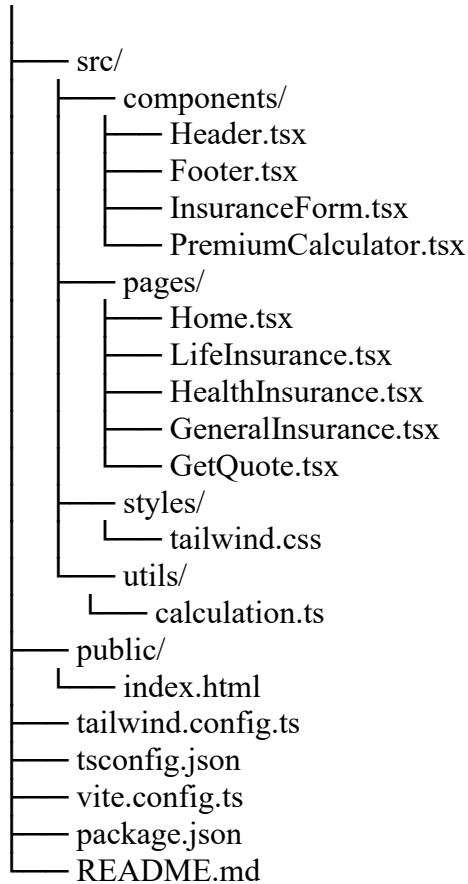
 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: CP	Aim: Implementation and Technical Documentation	
	Date: 25-09-2025	Enrolment No: 92310133002

1. Code Structure & Organization


Repository: Financialadvisor on GitHub (<https://github.com/Dhairya5300/Financialadvisor>)

File/Folder hierarchy:

Financialadvisor/



This separation ensures modularity: UI, logic, and styles are cleanly separated.

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

2. Implementation Details

Premium Calculation Logic: Implemented in calculation.ts with static formulas for life, health, and general insurance.

Example function:

```
export function calculateLifePremium(age: number, sumAssured: number): number {
  let baseRate = 0.002;
  let ageFactor = 1 + (age - 25) * 0.01;
  return sumAssured * baseRate * ageFactor;
}
```

Form Handling: InsuranceForm.tsx validates inputs and submits data to Formspree.

UI & Styling: Built with TypeScript + Tailwind CSS for responsive design.

Deployment: Hosted on Vercel, auto-deployed from GitHub main branch.

3. System Architecture & Data Flow

System Architecture:

[User Browser] → [Frontend (TypeScript + Tailwind UI)] → [Premium Calculation Module] → [Form Submission via Formspree] → [Formspree Server]

Data Flow:

1. User navigates website and inputs details.
2. Premium is calculated and displayed.
3. For quotes, data is submitted via Formspree.
4. Confirmation message shown to user.


4. Testing Procedures & Results

Testing Strategy:

- Unit Testing: Manual testing of premium functions with expected vs actual results.
- Form Validation Tests: Tested missing/invalid inputs.
- Integration/UI Tests: Verified navigation, calculation, and form submissions.
- Cross-Browser & Responsiveness: Verified on multiple browsers and devices.

Sample Test Case:

Life Premium Calculation | Input: age=30, sumAssured=100,000 | Expected: matches formula | Actual: correct.

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

5. Setup & Deployment Instructions

Prerequisites: Node.js ($\geq 14.x$), npm/yarn, Git.

Clone & Run:

```
git clone https://github.com/Dhairya5300/Financialadvisor.git
cd Financialadvisor
npm install
npm run dev
```

Deployment: Connect GitHub repo to Vercel, configure build, and auto-deploy from main branch.

6. Code Quality, Modularity & Best Practices

- Clean and commented TypeScript code.
- Modular separation of UI, logic, and styles.
- Validation on forms ensures robust inputs.
- GitHub repo with version control and commits.
- Premium calculation logic separated into utils for clarity.

7. Summary

The FinSecure prototype fulfills all implementation requirements:

- Functional TypeScript + Tailwind website.
- Static premium calculation for life, health, and general insurance.
- Formsfree integration for quotes.
- Deployed on Vercel.
- Clean structure, tested functionality, and robust integration.