**Child Insurance Management System**

Capstone Project Report

Team-Tech Hunters

Submitted in partial fulfilment of project requirements

**Rohan Patil**

**Yash Bora**

**Jayanth Gogada**

**JyotiPriya**

**Gopinadh**

**Aparna Kumari**

**CERTIFICATE**

This is to certify that the team consisting of:

**Rohan Patil,**

**Jayanth Gogada,**

**JyotiPriya,**

**Yash Bora,**

**Gopinadh,**

**Aparna Kumari.**

has successfully completed the capstone project entitled:

"Child Insurance Management System"

This project has been carried out with **sincerity, dedication, and adherence to academic standards**.

**Guide Name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**Project Coordinator:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**Head of Department:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DECLARATION**

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief,  
it contains no material previously published or written by another person, nor material which has been accepted  
for the award of any other degree or diploma of the university or other institute of higher learning,  
except where due acknowledgment has been made in the text.

**Submitted by:**

Rohan Patil,

Jayanth Gogada,

JyotiPriya,

Yash Bora,

Gopinadh,

Aparna Kumari.

**ACKNOWLEDGMENT**

We would like to express our heartfelt gratitude to all those who supported us throughout the completion of this project.

First and foremost, we extend our sincere thanks to our mentor for their invaluable guidance, constructive feedback, and continuous encouragement. Their insights greatly contributed to shaping the direction and quality of this work.

We are also thankful to our peers and teammates, whose collaboration and constant motivation kept us focused and determined.

A special note of appreciation goes to our families for their unwavering support, understanding, and patience during this journey. Their moral encouragement provided us the strength to persevere.

This project would not have been possible without the contribution of everyone involved, and we are truly grateful for the shared effort and commitment.

**TABLE OF CONTENTS**

* Abstract
* Table of Contents
* Introduction
* Problem Statement
* Objectives and Scope
* System Architecture
* Module Descriptions
* API Documentation
* Codebase Overview
* User Manual
* Admin Manual
* Working of the Project
* ER Diagram
* System Flow Diagram
* UI Mock-ups
* Testing and Results
* Conclusion
* References

**ABSTRACT**

The **Child Insurance Management System** is a web-based capstone project developed to simplify and automate the management of insurance policies for children. Built using **Spring Boot for the backend and React for the frontend**, the system offers core features such as secure parent registration, child profile management, policy selection, premium tracking, claim submissions, and admin approval workflows. The platform also includes a responsive dashboard for both users and administrators, ensuring transparency and ease of access across devices. By digitizing the insurance lifecycle, this project reduces administrative overhead, enhances policy tracking, and improves the overall experience for parents and insurers while maintaining data integrity and regulatory compliance..

**🏗️ SYSTEM ARCHITECTURE**

**The Child Insurance Management System (CIMS)** follows a robust three-tier architecture, enabling clear separation of concerns, enhanced scalability, and secure interactions across its components:

**⬤ Frontend (React + Vite)**

The client-facing layer ensures a smooth, responsive user experience for parents and administrators.

📝 Collects user input via forms for registration, policy application, and claim submission

📊 Displays real-time UI feedback, such as policy/claim status and transaction summaries

🔒 Implements protected routes using React Router and session-based access control

⚡ Developed using Vite for high performance and faster module loading

**⬤ Backend (Spring Boot)**

This is the core business layer, managing application logic and secure RESTful API communication.

🔐 Handles JWT-based authentication, parent/child registration, and role-based access

🧠 Encapsulates business logic for policies, claims, plans, and transactions through a service layer

📤 Supports file upload functionality for claim documents

📧 Sends email notifications for actions like policy approval, claim processing, etc.

🛡️ Enforces RBAC (Role-Based Access Control) to distinguish PARENT and ADMIN operations

**⬤ Database (PostgreSQL)**

The data layer securely persists all insurance-related information with structured relationships.

🗂️ Stores parent/child details, insurance plan data, claims, transactions, and policies

🔗 Maintains entity relationships to support end-to-end workflows

**📌 Tracks status flags such as:**

Policy: Pending, Approved, Rejected

Claim: Filed, Approved, Denied

Plan: Subscribed

🔒 Ensures secure storage of credentials and document links

🔐 Security Overview

🧾 Role-Based Authorization: Restricts sensitive operations (like claim approval, policy creation) to ADMIN users only

📥 Input & Document Validation: Ensures all form fields and file uploads follow strict validation checks for type, size, and integrity

**MODULE DESCRIPTIONS**

This section outlines the key modules of the Child Insurance Management System (CIMS), detailing their core functions and how they fulfil user needs across different workflows, from registration to claims and reminders.

**1. 👥 User Authentication and Registration**

**Easy On boarding and Account Management**

This module facilitates new user registration and account setup with a focus on simplicity and accessibility. Users can register with basic information, manage their profile, and securely log in to access the system. The platform supports profile updates, allowing users to maintain up-to-date child and guardian information.

**2. 🔐 Admin Access for Controlled Operations**

**Restricted Admin Login for Plan and Data Oversight**

This module is designed exclusively for administrators. Admins can log in to manage key system configurations—such as adding new insurance plans, managing policy data, and accessing a consolidated view of all parent and child records. This restricted access ensures operational control and data integrity.

**3. 📑 Policy Enrolment & Management**

**Streamlined Insurance Policy Application and Tracking**

Parents can browse and enrol in insurance plans for their children, providing necessary information and documents. This module supports the complete policy lifecycle, from creation to status tracking. Users can view all submitted policies along with their details and receive automatic premium calculations for financial clarity.

**4. 🧾 Claims Management (Claim-Service)**

**Efficient Claim Filing and Resolution**

This module allows users to file claims in case of incidents or emergencies by submitting relevant details and evidence. Parents can track the real-time status of each claim. On the admin side, claims are reviewed for accuracy and eligibility, after which they can be approved or rejected. This ensures a transparent and efficient claims process.

**5. 🔔 Reminders and Notifications (Reminder-Service)**

**Timely Alerts for Premiums and Renewals**

To reduce missed deadlines and improve user engagement, this module sends out notifications through email or WhatsApp. Users are reminded about upcoming renewals, premium due dates, and important policy updates. This feature plays a key role in maintaining active coverage and user awareness.

**CIMS API Documentation**

This section describes the core RESTful APIs used in the Child Insurance Management System (CIMS). Each endpoint includes a brief summary of its purpose and functionality.

**🔹 POST /api/admin/login**

**Authenticates admin user**

Accepts admin credentials and validates them against stored records. On success, returns a confirmation message or token. This is the entry point for admin-level access.

**🔹 POST /api/admin/addplan**

**Adds a new insurance plan**

Takes plan details such as name, coverage, and premium from the request body and adds it to the system. Only accessible to admins.

**🔹 GET /api/admin/getallplan**

**Retrieves all available plans**

Fetches a list of all insurance plans currently available in the system for review or management by the admin.

**🔹 POST /api/admin/delplan**

**Deletes an existing plan**

Removes the specified insurance plan based on the request body. Ensures that plan data is properly managed by administrators.

**🔹 GET /api/admin/getAllParent**

**Fetches all registered parents**

Retrieves the list of all parent users from the database. Used for administrative oversight.

**🔹 GET /api/admin/getAllChild**

**Fetches all registered children**

Returns a list of all children linked to registered parents in the system.

**🔹 POST /api/child/add?parentEmailId=**

**Adds a child to a parent profile**

Links a new child to an existing parent using their email. Accepts child details in the request body and returns the saved child entity.

**🔹 POST /api/claims/submit**

**Submits a new insurance claim**

Takes claim details such as policy ID, amount, and reason from the body and submits it for admin review and processing.

**🔹 GET /api/claims/pending**

**Lists all pending claims**

Retrieves all claims that are still under review and awaiting admin decision.

**🔹 PUT /api/claims/approve/{claimId}**

**Approves a submitted claim**

Admin endpoint to approve a pending claim by ID. Marks the claim as "Approved" in the system.

**🔹 PUT /api/claims/reject/{claimId}**

**Rejects a submitted claim**

Allows the admin to reject a claim by specifying its ID, updating the status to "Rejected".

**🔹 POST /api/user/registration**

**Registers a new parent account**

Accepts parent information like name, email, password, and stores the data in the system.

**🔹 POST /api/user/login**

**Logs in a parent and returns dashboard data**

Verifies parent credentials and, on success, returns profile and dashboard-related data.

**🔹 GET /api/user/changepassword?email=&pw=**

**Changes the parent's password**

Takes the email and new password as parameters and updates the password if the user exists.

**🔹 POST /api/user/updateemail**

**Updates parent’s email address**

Receives old and new email addresses in the request body and updates the parent’s contact information accordingly.

**🔹 POST /api/plan/addplan**

**Adds a new plan (alternative endpoint)**

Same functionality as /api/admin/addplan, used from another controller context.

**🔹 POST /api/plan/delplan**

**Deletes a plan (alternative endpoint)**

Same as /api/admin/delplan, used independently for plan deletion operations.

**🔹 POST /api/policy/buypolicy**

**Buys a policy for a child**

Receives policy details and links it to a child. Represents a new policy purchase action by a parent.

**🔹 PUT /api/policy/updatepremium**

**Updates the premium of a policy**

Takes policy ID and new premium value and updates it in the database.

**🔹 GET /api/policy/isExpired**

**Checks if a policy is expired**

Evaluates whether a given policy has passed its expiration date based on the current date.

**🔹 GET /send-test-email**

**Sends a test email**

Used for verifying the email service setup. Sends a hardcoded message to a predefined address.

**🔹 POST /api/test/send-renewal-email**

**Sends renewal notification email**

Triggers an email to notify users about their upcoming policy renewal. Used for scheduled notifications or testing.

**CODEBASE OVERVIEW**

***This project follows a modular full-stack architecture using Spring Boot for the backend and React for the frontend. The codebase is cleanly separated into functional layers, promoting maintainability, scalability, and security.***

**Backend (Spring Boot):**

/controller –

The /controller package contains all the REST API endpoints. These classes handle HTTP requests and route them to the appropriate service methods. Examples include AdminController, ClaimController, ParentController, and PolicyController. Each controller is mapped to a specific business domain and is responsible for interacting with the frontend via JSON-based HTTP requests.

/service –

The /service layer encapsulates the business logic of the application. It includes processing for admin login, plan and policy management, claim approval and rejection, user and child registration handling, email and SMS notifications, as well as dashboard analytics. Services serve as the core of the backend by orchestrating workflows and validating application logic before interacting with the database layer.

/model –

The /entity package defines the data structures used in the application as JPA entity classes. This includes Admin, Parent, Child, Plan, Policy, Claim, and Transaction. Each class maps directly to a relational database table and defines relationships such as OneToMany and ManyToOne that enforce business rules and data integrity within the application.

/repository –

The /repository package contains interfaces that extend Spring Data JPA repositories. These interfaces enable standard database operations like save, update, find, and delete with minimal boilerplate code. Repositories are tightly coupled to the entity classes and are used throughout the service layer to execute queries and perform CRUD operations efficiently.

/exception –

The /exception package includes centralized exception handling logic for the application. Custom exception classes such as InvalidClaimException and InvalidCredentialsException are thrown during error scenarios, and the GlobalExceptionHandler ensures that the frontend receives consistent and meaningful error messages through structured HTTP responses.

/test –

Includes unit and integration test classes written using JUnit and Mockito. These test files validate the correctness of service logic, controller responses, and repository interactions. Important service flows such as claim approval, policy creation, and data retrieval are covered with mock data and assertions to ensure code reliability and to catch regressions early during development.

**Frontend (React):**

/pages –

Houses all the main views of the application such as HomePage, ParentPage, AdminPage, LoginPage, Dashboard, ContactUsPage, PremiumCalcPage, EmailReminderPage, PolicyPage, PoliciesPage, CalculatorPage, and BrochurePage. It also includes a MapLocation page for visualizing the physical location of services. Each page corresponds to a major user flow and is mapped via React Router.

/components –

Reusable UI components like Navbar, Table, Modal, Button, FormField, and Footer. These elements promote code reuse and ensure consistent UI design across all pages and sections of the website.

/services –

Contains Axios-based API handler files that manage HTTP communication with the backend. These services are responsible for tasks such as plan retrieval, claim submission, user data access, and sending email notifications. Error handling is managed gracefully throughout each service interaction.

/context –

Implements React’s Context API to manage global state. This includes user session state, active page tracking, and other shared values required across components and views.

/testing –

Includes React testing files using tools such as Jest and React Testing Library. Components and pages are tested for rendering, user interaction, and API integration. Mock service responses and assertions ensure frontend components behave correctly under various conditions and input scenarios.

**USER & ADMIN MANUAL**

User Manual

This manual outlines the steps required for users and administrators to effectively use the Child Insurance Management System (CIMS). It provides comprehensive guidance from registration and policy selection to claim submission and admin-side claim management.

**1. USER SIDE MANUAL**

**1.1 Parent Registration**

**Parents must register on the platform to access the services and manage child insurance policies**.

**Steps:**

Click the Register button on the homepage.

Fill in your Name, Email, Password, and Phone Number.

Click Submit to complete registration.

You will be redirected to the Login page.

**1.2 Login to Dashboard**

**After successful registration, users can log in and access their dashboard.**

**Steps:**

Navigate to the Login page.

Enter your registered Email and Password.

Click Login to authenticate and access your parent dashboard.

Upon login, you'll be redirected to the Parent Dashboard.

**1.3 Navigating the Dashboard**

The dashboard is the control centre for parents. From here, users can view their children, associated policies, and any active claims or transactions.

**Features:**

View all registered children.

View active plans and policies.

See current and historical claims.

Track premium payment history and recent transactions.

**1.4 Viewing Policies**

**Parents can explore available insurance plans before choosing a suitable one.**

**Steps:**

Click the Policies tab or go to the Policy Page.

Browse the list of available insurance plans and their features.

Click Download Brochure for more detailed documentation.

Use the Premium Calculator to estimate costs.

**1.5 Using the Premium Calculator**

The Premium Calculator allows parents to estimate the monthly premium based on the child’s age and chosen plan.

**Steps:**

Go to the Premium Calculator page.

Enter the child’s age and select a plan.

Click Calculate to view the estimated premium.

Adjust values as needed for comparison.

**1.6 Submitting a Claim**

**Parents can submit insurance claims directly through the system.**

**Steps:**

From the dashboard, navigate to the Submit Claim section.

Select the child, associated policy, and upload the required documents.

Provide a description of the reason for the claim.

Click Submit Claim.

The claim will appear in a Pending state until reviewed by an admin.

**1.7 Tracking Claims**

Parents can track the status of their claims in real time.

**Status Labels:**

Pending – Claim is submitted and awaiting admin review.

Approved – Claim is reviewed and accepted.

Rejected – Claim is declined (reasons will be shown).

Real-time feedback banners notify users of submission success, errors, or status updates.

**1.8 Additional Features**

Parents can access extra pages from the navigation bar.

**Contact Us:** Submit inquiries or support requests.

Email Reminder: Set and receive premium due reminders.

Map Location: View the insurance company or branch location on an integrated map.

**2. ADMIN SIDE MANUAL**

**2.1 Admin Login**

Admins can log into the platform to manage plans, users, and claim requests.

**Steps:**

Go to the Admin Login page.

Enter valid Admin ID and Password.

Click Login to access the admin dashboard.

**2.2 Viewing Parents and Children**

Admins can access the list of all parent and child users.

**Steps:**

Navigate to Parent List to see all registered parents.

Click Child List to view all enrolled children under different policies.

**2.3 Managing Plans**

Admins are responsible for creating and deleting insurance plans.

**Add Plan Steps:**

Go to the Add Plan page.

Fill in plan details like name, duration, premium, and benefits.

Click Submit to add it to the system.

**Delete Plan Steps:**

Navigate to the Plan Management section.

Select the plan to remove.

Click Delete to permanently remove it from the database.

**2.4 Claim Management**

Admins review and act upon submitted claims.

**Steps:**

Navigate to the Pending Claims page.

Review each submitted claim with attached documents.

To accept, click Approve.

To reject, click Reject and optionally enter a rejection reason.

Upon action, the parent is notified of the decision on their dashboard.

**✅ System Feedback**

Throughout the system, banners and toast messages provide users and admins with real-time feedback for every major action:

Registration and login confirmation

Claim submission status

Plan updates or deletions

Policy assignments and document uploads

**WORKING OF THE PROJECT**

The Child Insurance Management System (CIMS) is a full-stack web application developed using Spring Boot (Java) for the backend and React with Tailwind CSS for the frontend. The system streamlines the process of child registration, insurance plan management, and claim processing for parents and admins.

**1. User Registration and Email Notification**

Parents can sign up through a registration form by providing details such as name, email, and password.

After successful registration:

A confirmation email is automatically sent to the user’s registered email address.

The system stores the user information securely in the backend database.

Users can then log in using their email and password.

**2. Parent Dashboard Access**

After logging in, parents are redirected to a personalized dashboard showing:

Registered children

Insurance plans enrolled

Claim history and current claim status

Payment and transaction history

The interface is dynamic, responsive, and provides real-time data.

**3. Child Registration & Plan Subscription**

Parents can add children by providing their name, gender, date of birth, and other relevant details.

After child registration, parents can browse available insurance plans and subscribe based on eligibility.

Subscribed plans are linked with the specific child’s profile and stored in the system.

**4. Claim Submission by Parents**

In case of a medical event or other claimable situation, parents can submit a claim form.

Required claim information includes:

Incident description

Date of occurrence

Document uploads (supporting evidence)

Uploaded documents are securely stored and associated with the claim.

**5. Admin Review and Claim Approval**

Admins access a dedicated Admin Dashboard with role-based access.

**Admin functionalities include:**

Viewing and verifying all registered users, children, and submitted claims

Approving or rejecting claims after reviewing the submitted information and documents

Once a decision is made, the claim status is updated in the system and visible to the parent.

**6. Status Updates and Notifications**

**The system keeps parents informed through:**

Real-time status messages on the dashboard (e.g., "Claim Approved", "Plan Active")

Email confirmation upon registration (optional for future enhancements: email on claim status)

Clear visual cues guide users through their actions and system updates.

**7. Security & Usability Features**

User credentials are stored securely using password hashing.

The frontend implements route protection and prevents unauthorized access.

Axios is used for secure HTTP calls between frontend and backend.

The UI is designed for responsive behaviour across devices and ensures easy navigation for both parents and admins.

**🧪 TESTING AND RESULTS**

Thorough testing was carried out at different levels of the application to ensure performance, correctness, and a seamless user experience.

**✅ Backend Testing**

Technologies Used: JUnit 5, Mockito

**Test Coverage Includes**:

Parent and Child registration

Insurance plan enrolment logic

Claim submission and approval workflows

Email sending functionality

Outcome: All business logic and data flows passed unit and integration tests successfully.

**✅ Frontend Testing**

Methodology: Manual UI testing in browser

**Key Areas Tested:**

Form validations (required fields, correct data format)

Route restrictions based on user role (Parent/Admin)

Dynamic display of data (dashboard updates)

Claim form submission and document upload handling

Outcome: All interactive elements and flows behaved correctly under different test conditions.

**✅ Integration Testing**

Tool Used: Postman

**Scenarios Tested:**

Parent login and child registration workflows

API communication between frontend and backend

Document upload and retrieval

Admin-only actions such as claim approval

Outcome: Full workflows from registration to claim approval were tested and verified to work smoothly.

**✅ Test Case Results (Pass)**

Parent registration and email confirmation worked successfully.

Child and policy records were accurately stored and retrieved.

Admin access control was enforced properly.

Claim status changes were reflected in real-time on both user and admin dashboards.

**CONCLUSION**

The Child Insurance Management System (CIMS) delivers a secure, scalable, and intuitive platform for managing child-focused insurance operations. Designed with modern architectural principles, it streamlines policy issuance, claim processing, and plan management while ensuring clarity and control for both users and administrators.

By leveraging Spring Boot for backend services, React with Vite for a high-performance frontend, and PostgreSQL for reliable data persistence, the system ensures:

🔹 High modularity and scalability to support growing user demands

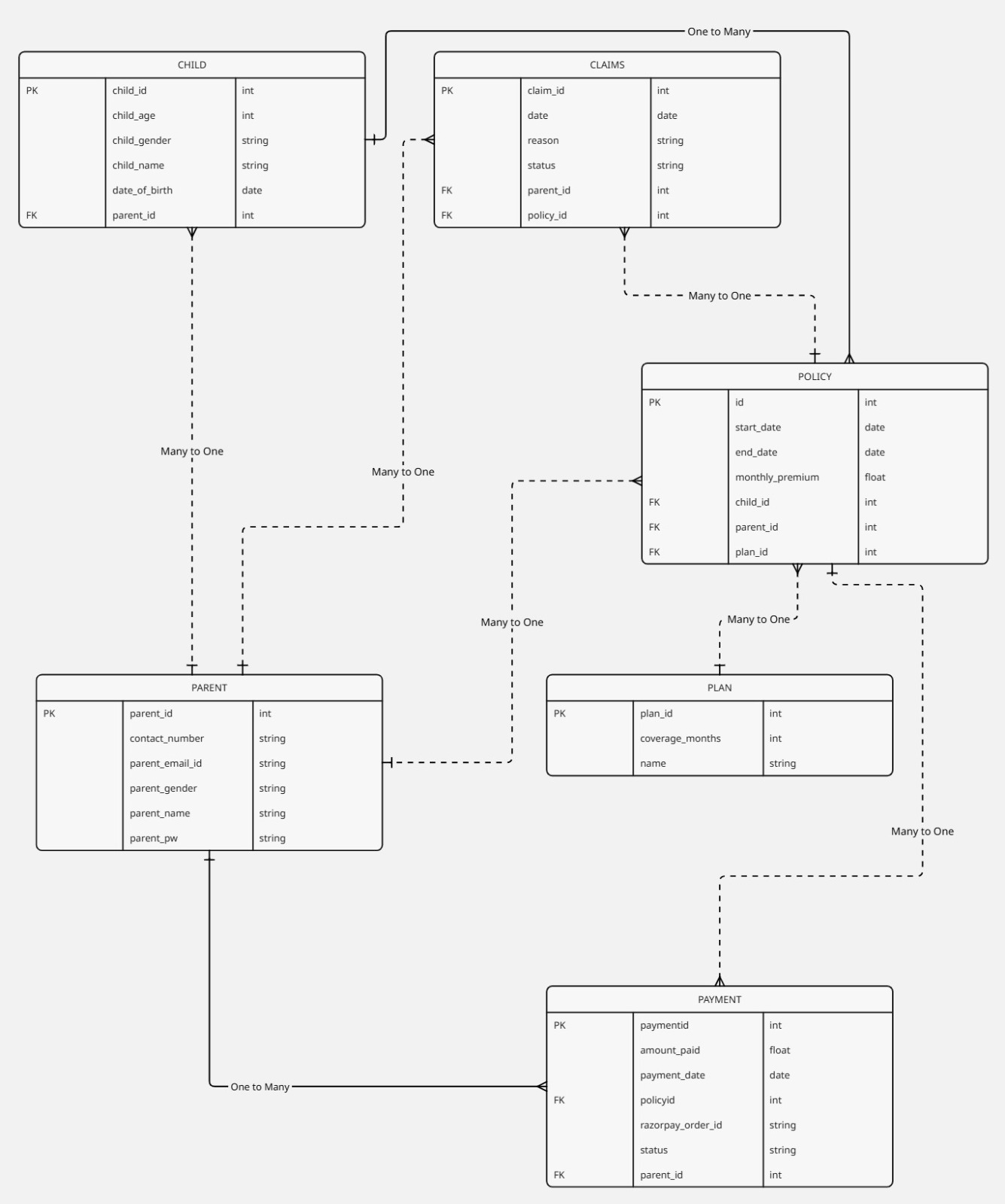
🔹 Efficient policy and claim workflows with real-time status tracking

🔹 Role-based access control to separate parent and admin functionality

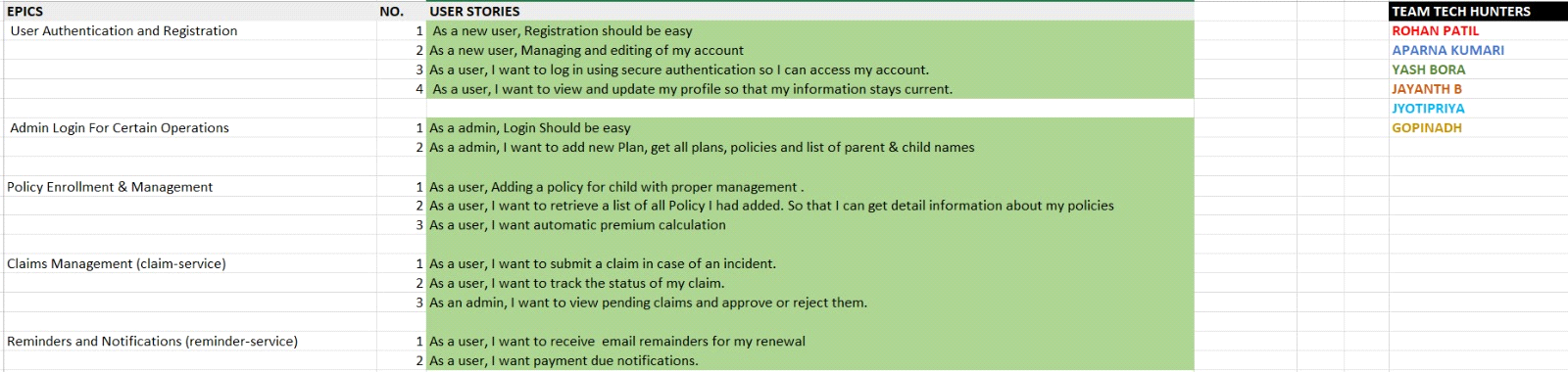
🔹 Improved user experience through responsive design and informative dashboards

This platform not only reduces manual overhead and processing delays but also builds transparency and trust among parents and administrators. With a focus on automation, usability, and control, CIMS serves as a comprehensive solution for modern child insurance operations.

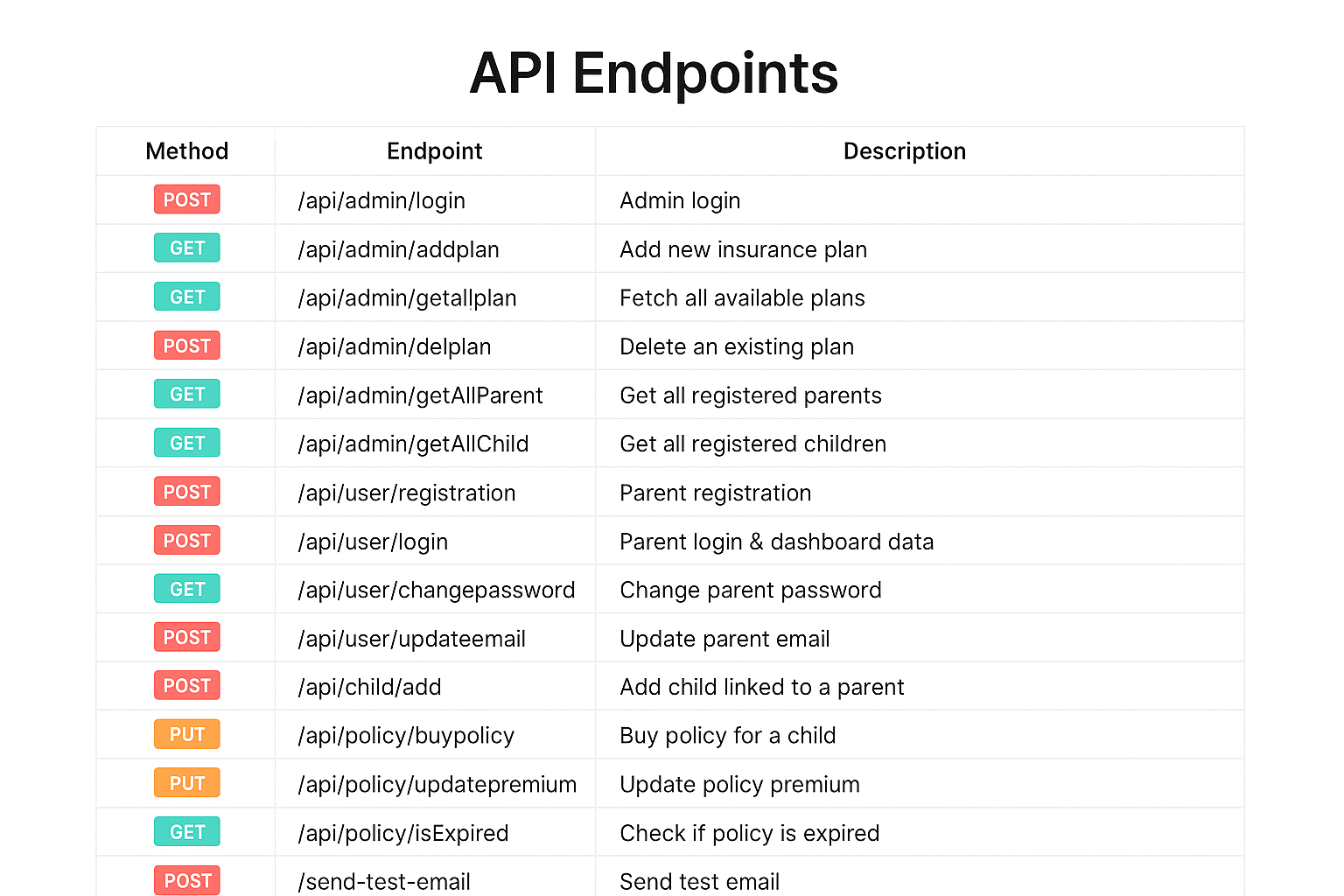
**ER DIAGRAM**

****

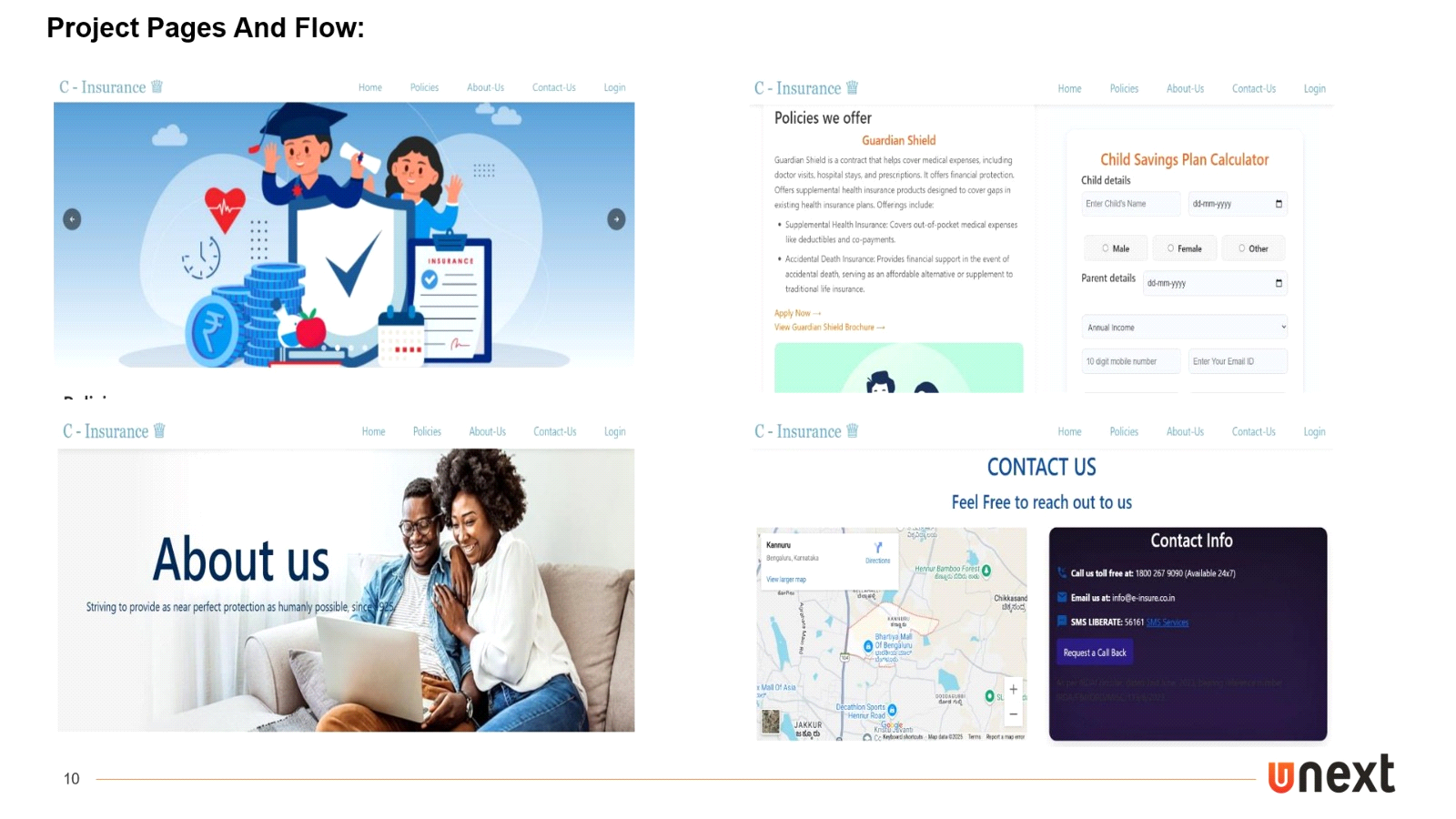
**Epics:**

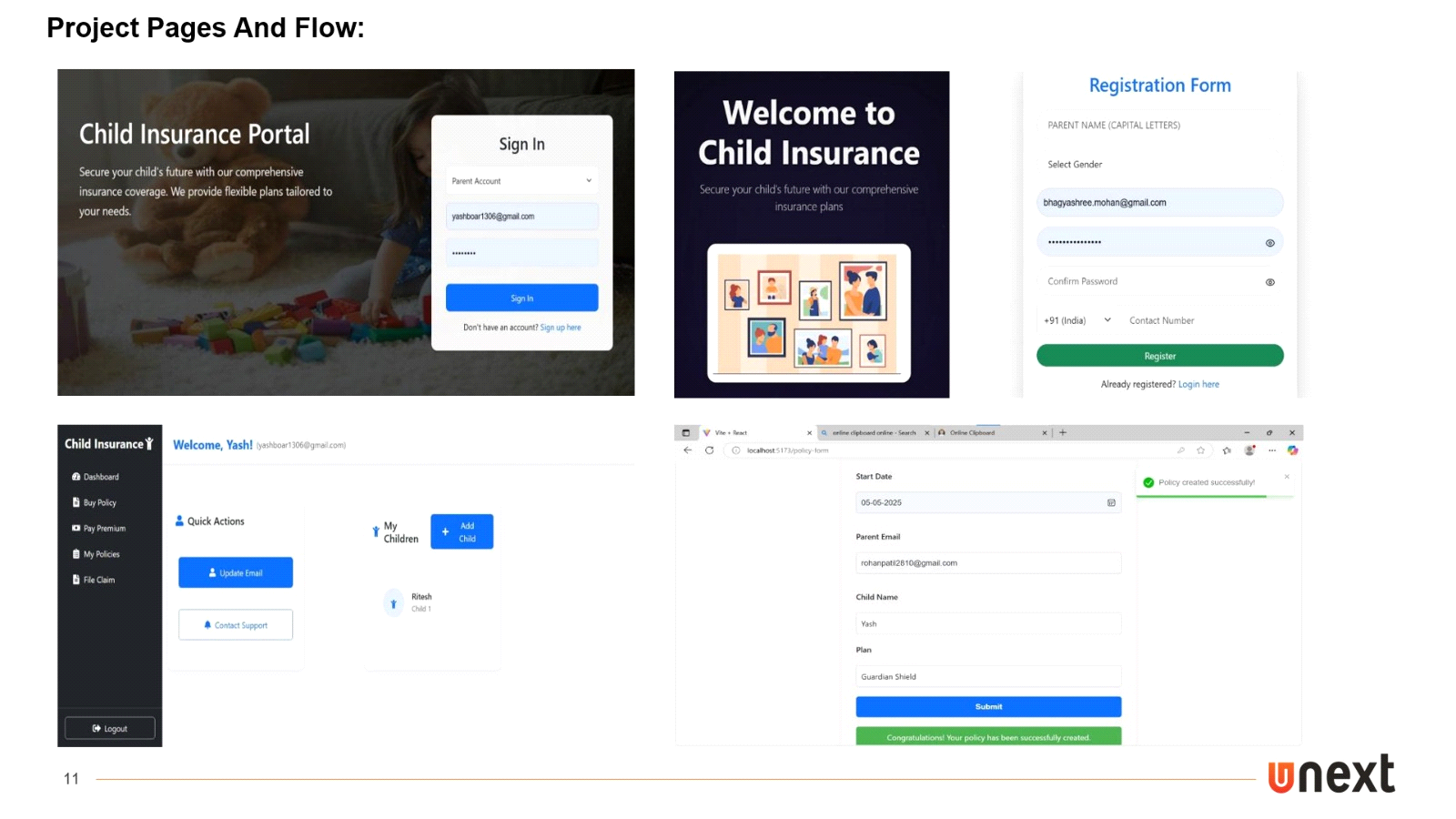


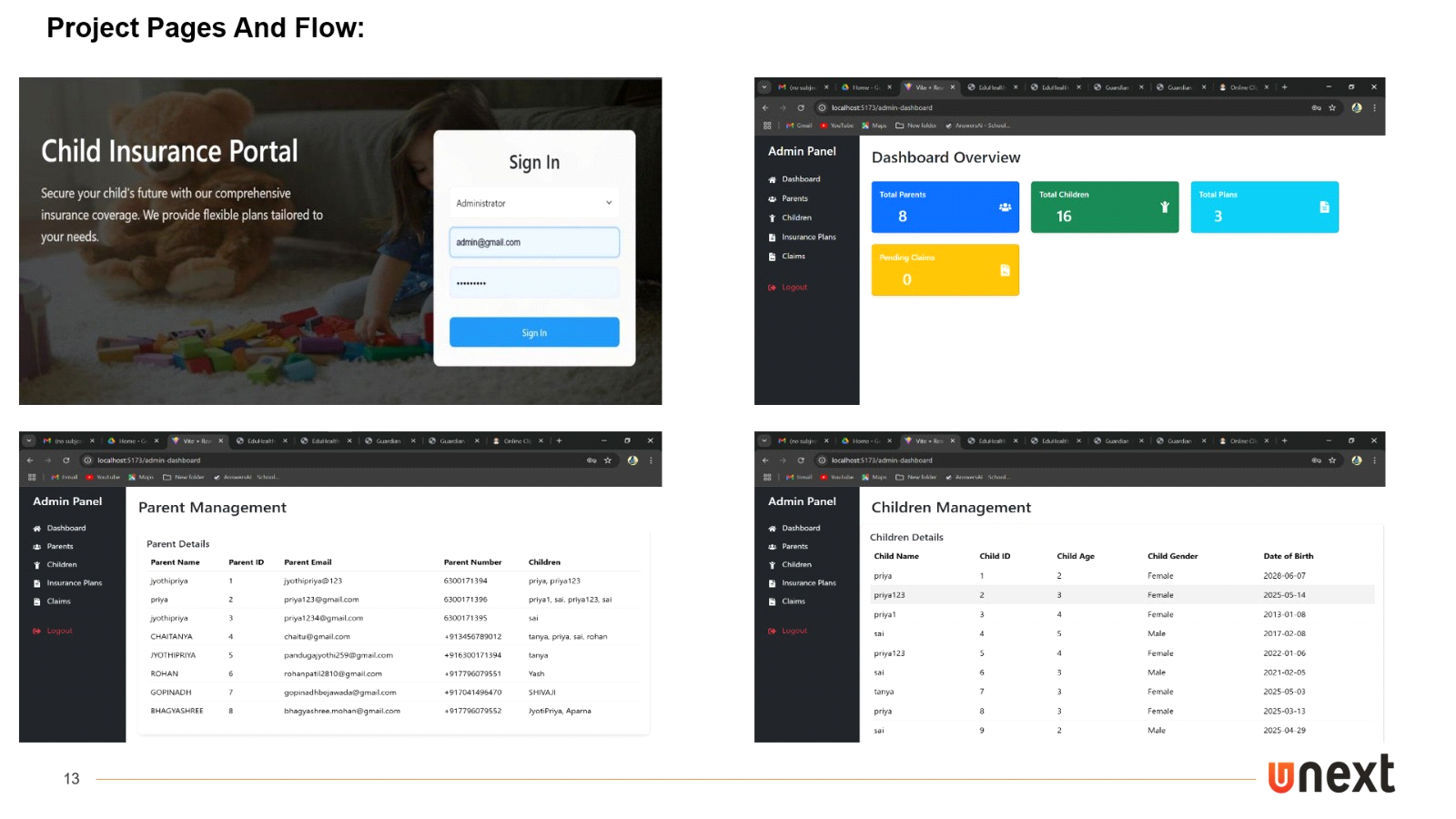
**API Endpoints**

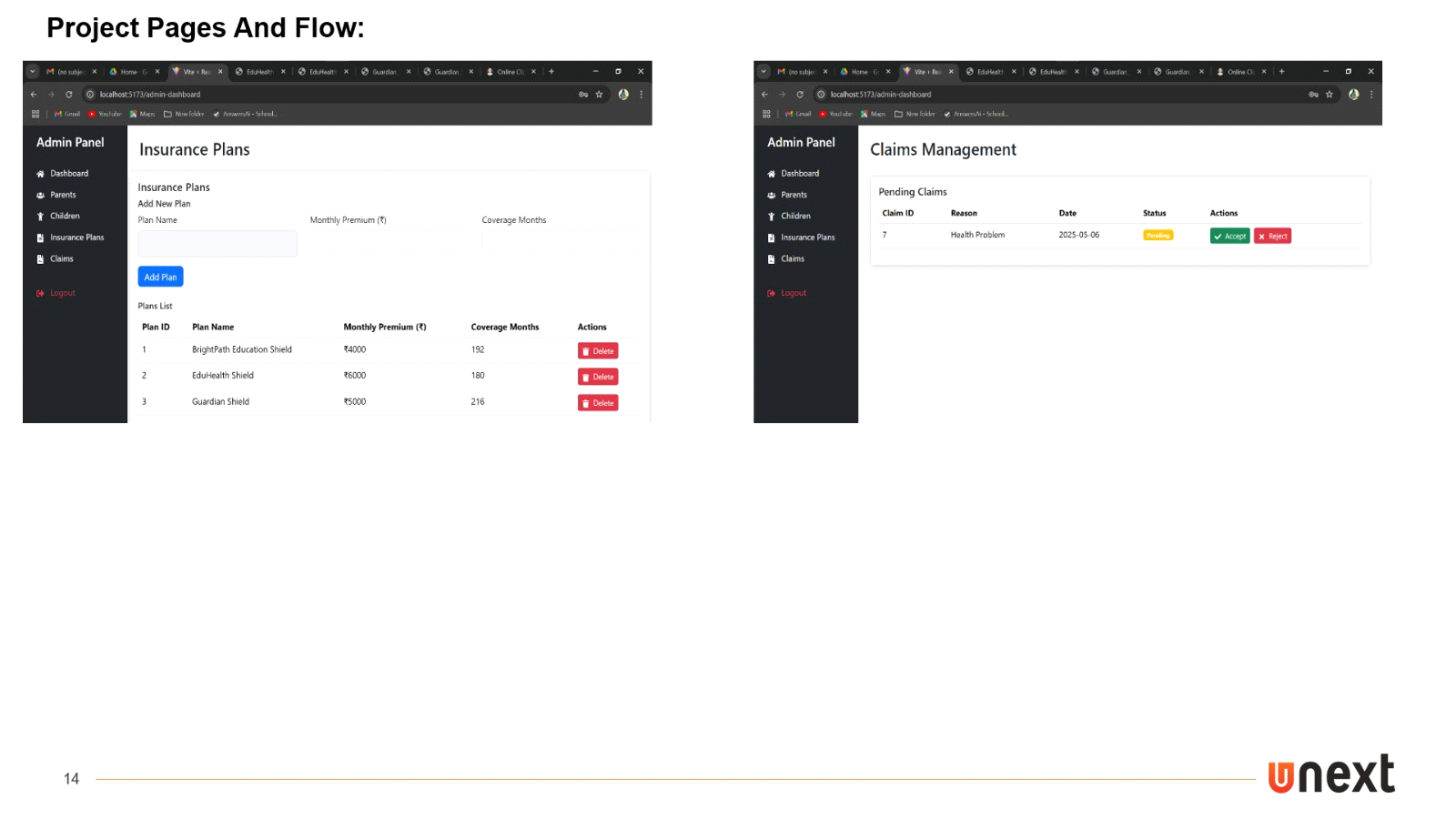


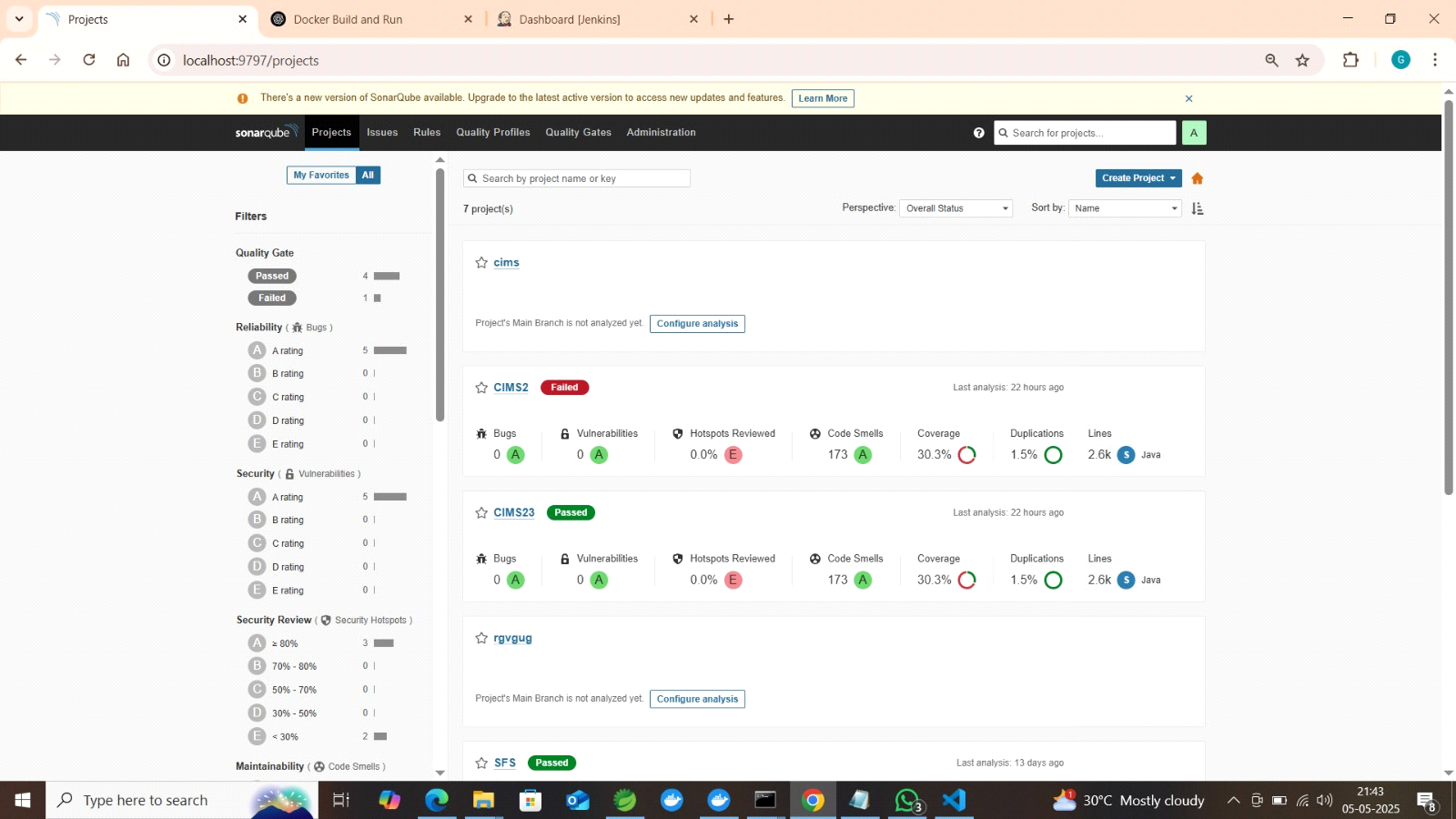
**UI MOCKUPS**



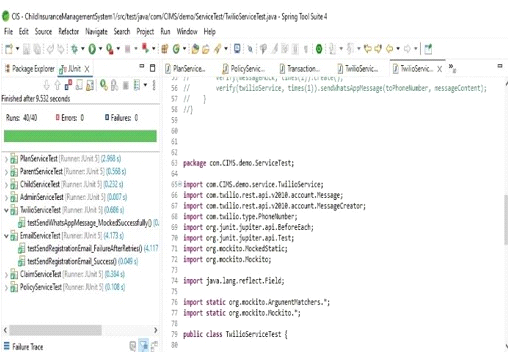






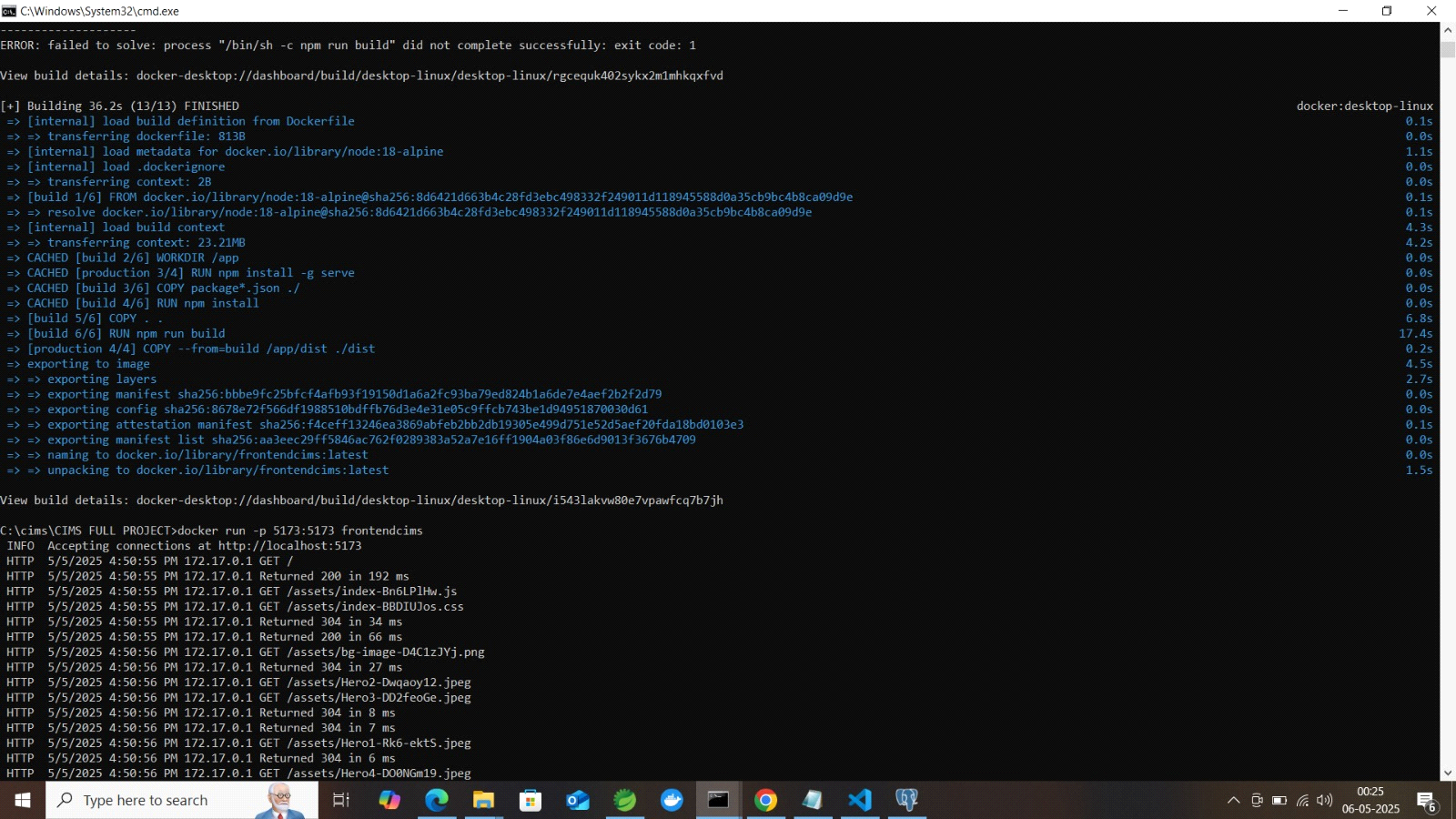
**SonarQube:**

**Testing**

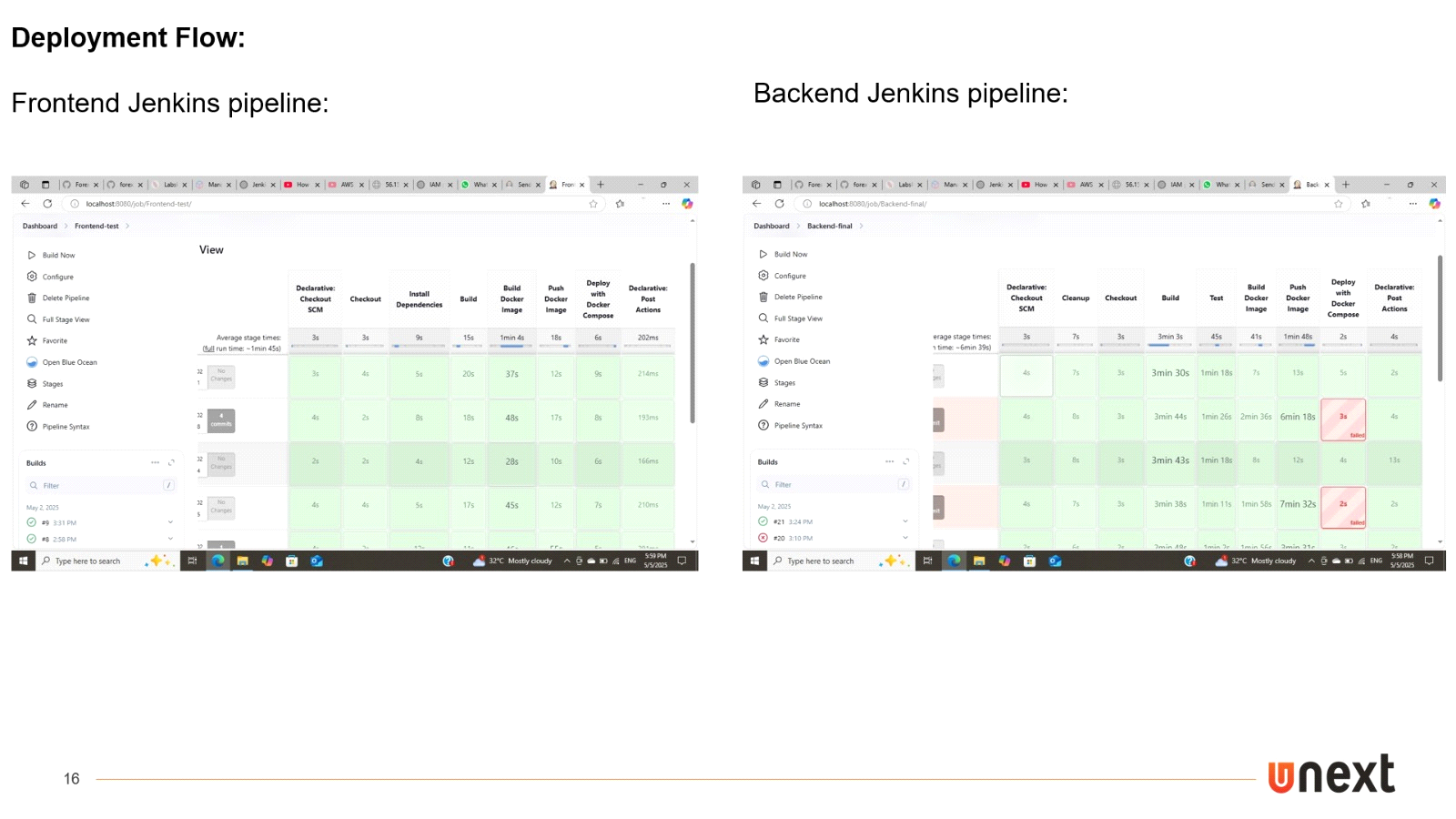


**Deployment Flow:**

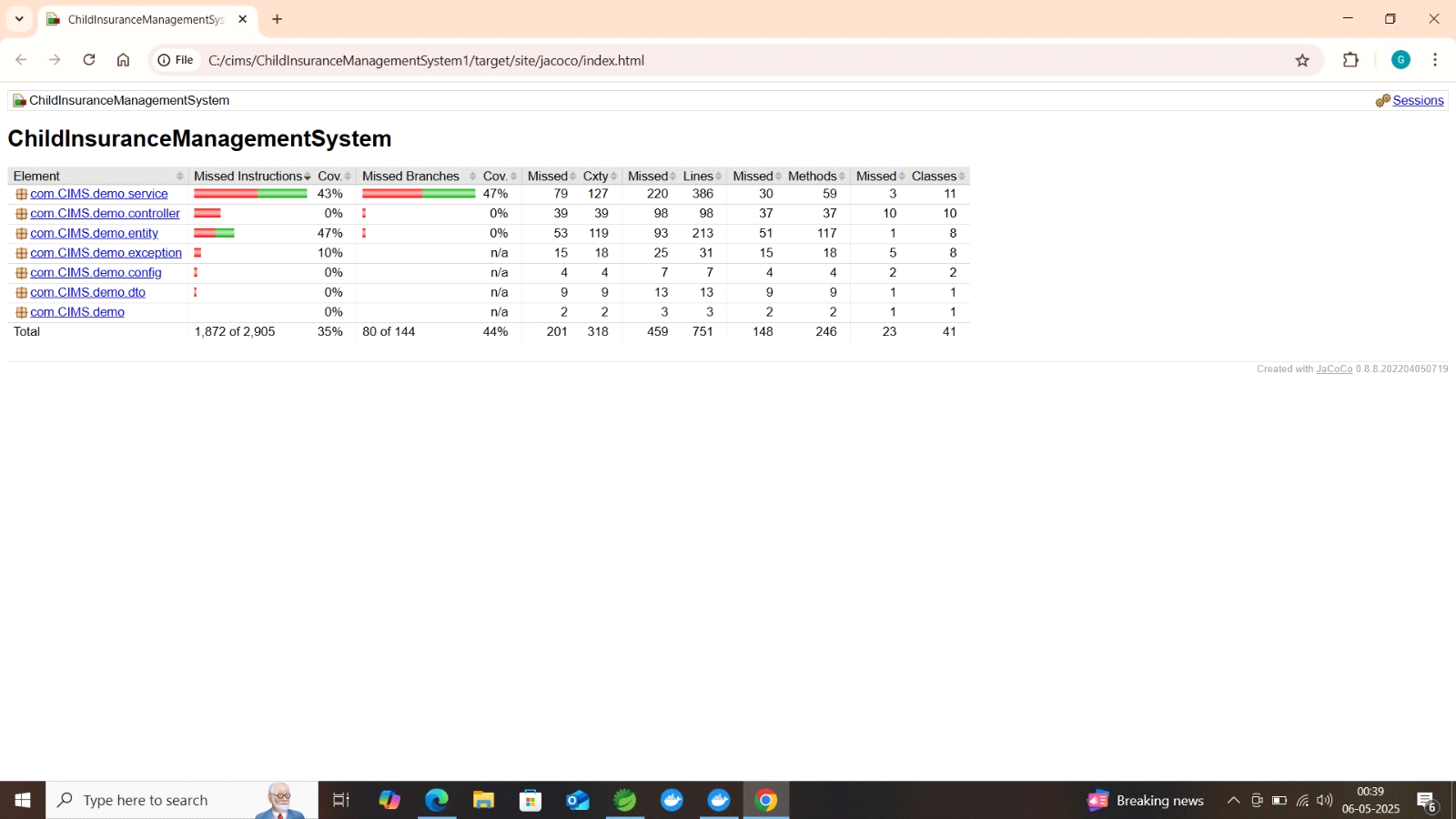
***Docker***



***Jenkins***



***Jacoca***



*Page 20*