High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **SafeNet - Security Awareness Portal UI**. The portal aims to educate users on web security best practices through interactive guides and quizzes, utilizing a modern, responsive frontend built with React, TypeScript, and Tailwind CSS.

System Architecture Overview

Architecture Description:
SafeNet is a single-page application (SPA) structured into modular React components. The UI is fully client-side, with potential integration points for backend APIs (e.g., for quiz data or user progress). The design emphasizes modularity, reusability, and responsive layouts.

Module/Component	Role/Responsibility
App Shell	Main layout, routing, and global state management
Navigation Bar	Site navigation, responsive menu
Guides Module	Displays interactive security guides
Quiz Module	Renders quizzes, handles user input and scoring
User Progress Tracker	Tracks and displays user learning progress
Theming/Styling	Provides consistent, responsive UI via Tailwind CSS
API Integration Layer	(Optional) Fetches/sends data to backend services

Component Interactions

Source Component	Target Component	Interaction Description
App Shell	Navigation Bar	Renders navigation, manages route changes
Navigation Bar	Guides/Quiz Modules	Triggers module rendering based on selection
Guides Module	User Progress Tracker	Updates progress upon guide completion
Quiz Module	User Progress Tracker	Updates progress and scores after quizzes
API Integration Layer	Guides/Quiz Modules	(Optional) Supplies dynamic content

Flow Summary:
User navigates via the Navigation Bar, selecting guides or quizzes. Modules render content and update the User Progress Tracker. Data may be fetched from APIs if available.

Data Flow Overview

Data Source	Data Destination	Data Type/Flow Description
Static Content/Assets	Guides/Quiz Modules	Guide text, quiz questions, images
User Input	Quiz Module	Quiz answers, interactions
Quiz/Guide Modules	User Progress Tracker	Completion status, scores
API (optional)	Guides/Quiz Modules	Dynamic content, user data

---

Technology Stack

Layer/Area	Technology/Framework
UI Framework	React
Language	TypeScript
Styling	Tailwind CSS, CSS
Markup	HTML5
State Management	React Context/State
Routing	React Router
Testing (optional)	Jest, React Testing Library

Scalability & Reliability

- **Scalability:**
  The modular React architecture supports easy addition of new guides, quizzes, or features. Static content can be expanded without major refactoring.
- **Reliability:**
  Stateless frontend design ensures high reliability. Responsive layouts and accessibility best practices are enforced via Tailwind CSS.
- **Security:**
  No sensitive data is handled on the frontend. Follows secure coding practices and input validation for all user interactions.

End of Document