CafeNet Frontend Development

# Introduction

This document outlines the full frontend architecture, design system, pages, components, and logic for the CafeNet platform.   
It provides a clear roadmap for frontend developers to begin implementation using Next.js and Tailwind CSS.

# Tech Stack

- Framework: Next.js (App Router)  
- Styling: Tailwind CSS + Shadcn/UI  
- State Management: Context API (or Redux Toolkit)  
- API Handling: Axios  
- Form Validation: React Hook Form + Yup  
- Authentication: JWT   
- Animation: Framer Motion  
- Icons: Lucide or Heroicons  
- Fonts: Albert Sans or Inter

# Design Theme

- Primary Color: #1e1e1e (Background)  
- Accent Color: #5d493d  
- Highlight Color: #927a5e  
- Style: Aesthetic, modern, soft shadows, rounded corners

# Pages Overview

* / - Home Page
* /login - Login
* /register - Register
* /dashboard - User Dashboard
* /cafes - List of Cafes
* /cafes/[id] - Cafe Detail Page
* /profile - User Profile
* /profile/edit - Edit Profile
* /profile/password - Change Password
* /rewards - My Rewards
* /refer - Referral System
* /history/visits - Visit History
* /history/points - Points History
* /admin - Admin Dashboard
* /admin/cafes - Manage Cafes
* /admin/users - Manage Users

# Reusable Components

* Navbar.tsx – dynamic navigation based on auth
* Footer.tsx – site-wide footer
* CafeCard.tsx – reusable cafe card
* PointTracker.tsx – shows current user points
* ReferBox.tsx – displays referral code and share button
* ProtectedRoute.tsx – protects routes from unauthorized access

# Layout Strategy

- Mobile First: Stack vertically with padding and scroll  
- Desktop View: Use grid layout, split panels, sidebars  
- Tailwind breakpoints: sm, md, lg, xl used for responsive design  
- Container: 'container mx-auto px-4' to center content

# Features by Role

* User Can See:
* Cafe Can See:
* Admin Can See:

# Leaderboards & Tags

Leaderboards include:  
- Top Referrers  
- Top Visitors  
- Top Redeemers  
- Top Cafes  
- Tag-Based Leaderboards (WiFi, Pet Friendly, etc.)  
Tags are created by cafes during onboarding/editing and used to filter and sort cafes.

# Authentication & API

- JWT Auth via HttpOnly Cookies or LocalStorage  
- AuthContext used globally  
- API calls with Axios  
- Protect dashboard and admin routes via middleware and ProtectedRoute

# Developer Notes

1. Start with layout.tsx and global components like Navbar/Footer.  
2. Implement AuthContext and basic API handlers.  
3. Begin from /login and /register, then /dashboard and cafes.  
4. Use responsive utilities from Tailwind.  
5. Always keep UX clean, mobile-first, and smooth.