

Title:

🧠 Community Skill Exchange Platform – Frontend (Next.js + React + Tailwind)

Prompt:

Create a **multi-page Next.js 14 (App Router)** frontend called “**Community Skill Exchange Platform**”, where users can **trade skills** (like coding for design) through an interactive, visually warm and modern UI.

The theme should feel **creative, social, and collaborative**, with **soft warm color tones** (coral, peach, amber, and cream gradients) and elegant typography using **Pinterest Sans** (use a close alternative like *Poppins* or *Inter* if unavailable).

The app must be **fully responsive**, built with **Tailwind CSS**, **Framer Motion animations**, and clean, reusable React components.

The goal is to make the frontend production-ready for future backend integration.

🎨 Overall Design Direction

- Warm, welcoming color palette (peach, coral, amber, off-white background)
- Rounded cards and buttons with soft shadows
- Minimalist icons (use Lucide icons)
- Smooth page transitions (Framer Motion fade/slide)
- Semi-glassmorphic panels for dashboards
- Use Pinterest Sans or similar geometric sans-serif
- Light/dark toggle for later

📁 Pages to Build (Next.js App Router Structure)

1. Home Page (/)

- Hero section with gradient background, large heading: “Trade Skills. Learn Anything. Grow Together.”
- “How It Works” section with 3-step visual cards (Join → Match → Learn)
- Featured Skill Categories (cards like *Programming*, *Design*, *Music*, *Writing*)
- Testimonials section (3–4 rounded cards with quotes)
- CTA footer with “Join the Skill Exchange” button

2. Login Page (/login)

- Centered card layout with input fields (email, password)
- Soft gradient background
- “Login” and “Continue with Google” button
- “Don’t have an account? Register” link below
- Subtle motion hover effects on buttons

3. Register Page (/register)

- Two-column layout: left panel with illustration, right with signup form
- Input fields for: Name, Email, Password, Skills Offered, Skills Wanted (tags input style)

- Button: “Join Now”
- Gradient accent buttons and form field focus animations

4. Dashboard (/dashboard)

- Layout: Sidebar (left) + Top Navbar + Main Content Area
- Sidebar: links to Dashboard, Matches, Chats, Profile, Settings
- Main Area: Feed of suggested users (cards with avatars, skill tags, and “Trade Skill” button)
- Include “Your Matches” preview at the top
- Animate skill cards with hover lift effect
- Placeholder data from mock JSON

5. Match Feed Page (/matches)

- Tinder-style interface with swipeable skill cards
- Show user name, bio, offered/needed skills
- Buttons: “Trade Skill” / “Skip”
- Framer Motion swipe animations
- Toast message or confetti when a match occurs

6. Chat Page (/chat/[chatId])

- Layout: Chat list sidebar + main chat area
- Chat UI:
 - Messages in rounded bubbles
 - Scrollable area auto-scrolls to last message
 - Typing indicator animation
- Input bar with emoji and send button
- Use mock chat data for now

7. Profile Page (/profile/[id])

- Banner section with gradient overlay and avatar
- Profile details: name, location, description
- Sections for “Skills Offered” and “Skills Wanted” (tag chips)
- Edit profile button (modal form)
- Show connection stats (matches, exchanges, followers)

8. Settings Page (/settings)

- Tabs for Account, Preferences, Notifications
- Theme toggle (light/dark mode)
- Update email/password form fields (UI only)
- Save button with animated success toast

Components to Include

- Navbar → top navigation bar with logo, profile avatar
- Sidebar → dashboard and chat navigation links
- SkillCard → reusable card for displaying user/skill details

- ProfileCard → user summary for dashboard feed
- ChatBox → chat message list UI
- MatchCard → swipeable skill match interface
- Tag → styled skill chips with color-coded backgrounds
- Button → warm, rounded button with gradient hover
- Modal → for editing profile or confirming actions

🌟 UI Animation Guidelines

- Use **Framer Motion** for:
 - Page transitions (fade-in, slide-up)
 - Card hover lifts
 - Button press scaling
 - Match success animation (confetti or color flash)
- Use Tailwind transitions for hover/focus animations

🧠 Frontend State (mock only)

Use temporary mock JSON data for users, chats, and skills.

Example mock fields:

```
{
  id: 1,
  name: "Ava Sharma",
  skillsOffered: ["UI Design", "Figma"],
  skillsWanted: ["React.js", "Next.js"],
  bio: "Creative designer looking to learn web development.",
  image: "/avatars/ava.png"
}
```

📁 Folder Structure (to generate)

/src

```
├── app
│   ├── page.jsx
│   ├── login/page.jsx
│   ├── register/page.jsx
│   ├── dashboard/page.jsx
│   └── matches/page.jsx
```

```
|   |— chat/[chatId]/page.jsx
|   |— profile/[id]/page.jsx
|   |— settings/page.jsx
|   |— components
|   |— Navbar.jsx
|   |— Sidebar.jsx
|   |— SkillCard.jsx
|   |— ProfileCard.jsx
|   |— MatchCard.jsx
|   |— ChatBox.jsx
|   |— Tag.jsx
|   |— Button.jsx
|   |— Modal.jsx
|   |— data
|   |— mockUsers.js
|   |— styles
|   |— globals.css
```

Goals for v0 AI Output

- Generate **production-quality** Next.js + Tailwind + Framer Motion code
- All pages functional with placeholder data
- Responsive layout
- Warm and human-friendly visual style
- Smooth animations and polished UX
- Easily connectable to future backend (Django REST or Node API)

Extra Notes

- Use **Next.js App Router** (/app folder structure)
- Use **Client Components** for interactive pages (dashboard, chat, matches)
- Don't include backend logic — focus only on the frontend design and flow
- Optimize for readability and reusability
- Add SEO metadata for home page

