MyCircle Project Documentation

1. Aim

MyCircle is a hyper-local community platform designed to connect people within their immediate vicinity. Unlike traditional platforms, MyCircle is **not purely money-centric**; it actively promotes a **Favor/Barter System**.

Core Philosophy:

- **Community First:** Build a trusted circle of neighbors who help each other.
- **Favor & Barter:** Users can exchange services or goods without exchanging money, fostering genuine community spirit.

  > **Example:** A young man wants to celebrate his sister's birthday but needs help with decorations. He posts a request looking for 2 people to help. In return, instead of just money, he might offer to pay them, treat them as guests at the party, or offer a return favor later.

- **Hyper-Local:** Focus on connections within a specific radius to ensure physical accessibility.

Key Features:

- **Find Jobs & Services:** Connect with local service providers.
- **Buy, Sell & Rent:** A marketplace for local trading.
- **Favor Exchange:** A dedicated way to ask for help or offer skills in return for favors.
- **Real-time Communication:** Chat instantly to negotiate barters or services.

---

2. Tech Stack

Mobile App (MyCircleMobileBare)

- **Framework:** React Native (Bare Workflow).
- **Language:** TypeScript / JavaScript.
- **UI/UX:** NativeWind (Tailwind CSS), Lucide React Native icons.
- **Navigation:** React Navigation (Stack & Bottom Tabs).
- **Services:** Geolocation, Axios, Socket.io-client.

Web App (MyCircleClient)

- **Framework:** React.js (Vite).
- **Visuals:** Tailwind CSS, Framer Motion, Three.js (for immersive elements).
- **Routing:** React Router DOM.

Server (MyCircleServer)

- **Runtime:** Node.js / Express.js.
- **Database:** MongoDB (Mongoose ODM).
- **Real-time:** Socket.io (Events: `new_post`, `receive_message`).
- **Auth:** JWT & Google OAuth (Passport.js).
- **Media:** Cloudinary.

---

3. System Design

Architecture

Standard Client-Server model with a real-time WebSocket layer for instant communication.

```
graph TD
    Client[Mobile/Web App] <-->|Rest API| Server[Express Server]
    Client <-->|Events| Socket[Socket.io]
    Server <-->|Data| DB[(MongoDB)]
    Server <-->|Media| Cloud[Cloudinary]
```

Database Schema (Key Models)

1. **User**
   - `googleId`, `displayName`, `avatar`: Identity.
   - `location` (GeoJSON or string): For hyper-local matching.
   - `skills`: List of skills user can offer for barter.
   - `reputation`: Community rating based on successful favors/transactions.
   - `connections`: Network of trusted neighbors.
2. **Post**
   - `type`: 'favor', 'job', 'service', 'sell', 'rent'.
   - `barter_request`: (Optional) What the user wants in return (e.g., "Home cooked meal").
   - `location`: Geotagging.
   - `status`: 'open', 'completed'.
3. **Conversation**
   - `participants`: Users involved in the negotiation.
   - `context`: Link to the Post being discussed.
   - `messages`: Chat history.

4. Workflow

A. Favor/Barter Flow

1. **Create Post:** User A posts a request: "Need help moving a sofa."
    - *Payment Mode:* Selects "Favor" or "Barter".
    - *Offer:* "Will bake a cake in return."
2. **Discovery:** User B (neighbor) sees the post in their feed.
3. **Negotiation:** User B chats with User A to agree on time and terms.
4. **Completion:** Once done, both users mark the transaction complete.
5. **Reputation:** They rate each other, building their "Community Trust Score".

B. Service/Job Flow

1. **Search:** User searches for "Electrician".
2. **Connect:** Finds a verified local provider.
3. **Engage:** Direct message or "Request Service".
4. **Transaction:** Service delivered and paid (cash/online) or bartered.

---

5. Examples

Post Object (Barter Type)

```
{
  "title": "Garden cleanup help needed",
  "type": "favor",
  "description": "Need someone to help weed my garden for 2 hours.",
  "barter_offer": "Private Math tutoring session for kids",
  "location": "Sector 4, Bilaspur",
  "user": "userId_123"
}
```

Socket Event (New Favor Request)

```
// When someone offers to help
socket.emit('offer_help', {
  postId: 'post_abc',
  offererId: 'user_xyz',
  message: 'I can help this Sunday!'
});
```

---

6. Diagrams

Barter Negotiation Flow

```
sequenceDiagram
    participant Requester
    participant Server
    participant Helper

    Requester->>Server: Post Favor ("Need help moving")
    Server->>Helper: Feed Update (New Local Favor)
    Helper->>Server: Send Message ("I can help, I have a truck")
    Server->>Requester: Notification (New Offer)
    Requester->>Helper: Chat ("Great! Sunday work?")
    Helper->>Requester: Chat ("Yes, deal.")
    Requester->>Server: Mark as "In Progress"
```

Community Trust System

```
graph LR
    A[User Completes Favor] --> B{Verify}
    B -->|Confirmed| C[Increase Reputation Score]
    B -->|Dispute| D[Moderator Review]
    C --> E[Higher Visibility in Feed]
```