🔄 Weekly Timeline with Team Member Breakdown

🚀 Week 1: Infrastructure, Auth, and Onboarding

👤 Member A

- **Backend**
  - Set up project repo (monorepo or split repos with workspace support)
  - Set up PostgreSQL + PostGIS database
  - Set up authentication system (OAuth2.0, MFA)
  - Create endpoints for:
    - Company registration
    - Employee invite & onboarding
    - Role-based access control
  - Encryption setup for sensitive data
  - Begin API for user sessions and access logs
- **DevOps**
  - Deploy backend on **Render or Railway** (GCP/AWS later if needed)
  - Secure API keys and third-party creds

👤 Member B

- **Frontend**
  - Build UI for:
    - Login/signup (with MFA)

        - Company admin dashboard (basic)
        - Invite form and onboarding flow
    - Implement secure auth flow with token storage
    - Create initial project layout using **React + Tailwind**
    - Build UI for Terms of Service, notification preferences

🟨 **Shared**

- Define API contracts for all auth/user-related actions
- Plan full schema for users, roles, companies
- Set up GitHub repo workflows and README

🗺️ Week 2: Mapping, Data Integration, and Property Systems

👤 **Member A**

- **Frontend**
    - Integrate map interface (Mapbox or Leaflet)
    - Add property markers, clustering logic
    - Implement zoom, pan, toggle views
    - Add filters: property value, size, location
    - Add heat map layer: **Wealth Distribution Heat Maps** ⭐
    - Start implementing **Saved Search UI** ⭐

👤 **Member B**

- **Backend**

- Integrate 3 third-party APIs:
  - Zillow (property value)
  - WealthEngine (net worth)
  - Fast People Search (ownership info)
- Build:
  - Consolidated owner profiles
  - Net worth estimates + confidence score
  - Transaction history endpoint
- Set up Redis for caching API responses
- Begin backend for "Wealth Trails" unique feature

🟨 Shared

- Design DB schema for properties, ownership, wealth data
- Write property detail and search endpoints
- Set up monitoring/logging for API and database
- Test frontend ↔ backend integration for core map features

📊 Week 3: Final Features, Visualizations, and Demo Polish

👤 Member A

- **Frontend**
  - Build detailed property card UI with expandable info
  - Add owner net worth panel

    - Create **Ownership Network Graph** ⭐ (D3.js or Vis.js)
    - Implement “Wealth Trails” frontend interface 🧠
    - Add bookmarking, saved search UX
    - Polish dashboards and transitions

👤 Member B

- **Backend**
    - Finalize data export & scheduled report generator ⭐
    - Finish “Wealth Trails” backend engine:
        - Track owner → property → owner chains
        - Cache historical wealth transitions
    - Implement access logs, analytics for admin dashboard
    - Write reports endpoint with export history
    - Implement **team sharing** of property collections ⭐

🟨 Shared

- Accessibility, performance optimization (lazy loading, map perf)
- Conduct security checks (input validation, rate limiting)
- Manual testing + bug fixing
- Record demo video walkthrough (5 mins)
- Final pitch deck + architecture diagram + deployment writeup

---

📦 Final Deliverables Checklist

Deliverable	Owner	Status
Working prototype (deployed)	Both	🔜
GitHub repo with clear commits	Both	🔜
Test accounts & setup docs	Member B	🔜
API documentation	Member A	🔜
System architecture diagram	Member A	🔜
Pitch deck	Both	🔜
5-minute demo video	Both	🔜

🧠 Unique Feature Highlight: *Wealth Trails*

What it is:
A traceable flow of ownership and wealth transfer across people, entities, and properties. Users can:

- Click on a property and view upstream/downstream connections (owners' other properties, linked people/entities)
- See estimated net worth trajectories over time
- Detect unusual patterns or clusters in high-value ownership chains

Why it stands out:

- Combines mapping, graph data, and temporal insights
- Creates a visual, investigative experience like a "financial forensics" tool
- No other similar product in your domain has this combined visualization

---

Let me know if you'd like this in **Notion**, **Markdown**, or **PDF** format for sharing with your team.