**Project Audit: La Capitolare AI Fundraising System**

Let's review what we've accomplished so far and identify what remains to be completed in our pretotype for the La Capitolare Library AI Fundraising System.

**1. Completed Components**

**Backend (.NET Core)**

* ✅ Core domain models (Donor, Manuscript, DonorScore, etc.)
* ✅ Database context with Entity Framework Core
* ✅ Repository pattern implementation
* ✅ Basic seed data
* ✅ API controllers for the main entities

**Frontend (Angular)**

* ✅ Basic project structure
* ✅ Core modules (core, shared, features)
* ✅ API service for backend communication
* ✅ Mock API service for testing without backend
* ✅ Basic component implementations:
  + ✅ Donor Intelligence Dashboard (partial)
  + ✅ Manuscript-Donor Matching (partial)
  + ✅ Campaign Simulator (partial)

**2. Outstanding Issues**

**Technical Challenges**

* ❌ TypeScript errors in the simulation-results component
* ❌ Compatibility issues with charting libraries
* ❌ Need to resolve module import errors

**Incomplete Features**

* ❌ Proper data visualization (charts and graphs)
* ❌ Complete implementation of the Campaign Simulator
* ❌ Full integration between components
* ❌ Polished UI/UX for demonstration

**3. Next Steps**

**Short-term (To get a working demo)**

1. **Simplify component implementations**:
   * Focus on basic functionality first
   * Use simpler visualizations or tables instead of complex charts
   * Reduce dependencies on third-party libraries
2. **Fix critical TypeScript errors**:
   * Consider creating new components from scratch if necessary
   * Use more permissive TypeScript settings temporarily
3. **Ensure API connectivity**:
   * Confirm that frontend can communicate with backend
   * Validate data flow through the system

**Medium-term (For presentation quality)**

1. **Enhance visualization**:
   * Add more sophisticated visualizations once basics work
   * Implement responsive design for different screen sizes
2. **Complete core AI demonstrations**:
   * Ensure all three key features are working:
     + Donor Intelligence
     + Manuscript-Donor Matching
     + Campaign Simulation
3. **Add realistic data**:
   * Expand seed data to be more representative
   * Ensure AI insights feel authentic and valuable

**4. Suggested Immediate Actions**

1. **Simplify problem components**:
   * Create minimal versions of problematic components
   * Focus on one component at a time until it works
2. **Test individual features**:
   * Verify each component works in isolation
   * Then integrate components gradually
3. **Consider backend mock data**:
   * If frontend components remain problematic, focus on making the backend API robust
   * Demonstrate features through the API directly if needed

**5. Overall Status**

The project has made good progress with the foundational architecture in place, but we're facing some technical challenges with the Angular components. The backend looks relatively solid, with the core domain models, repositories, and controllers implemented.

To move forward efficiently, we should focus on simplifying the problematic components and ensuring we have a working demonstration of the core AI capabilities, even if the UI is less polished than initially planned. Once the basic functionality is working, we can incrementally enhance the UI and visualization aspects.

**La Capitolare AI Fundraising System - Development Audit**

**Project Context**

We're developing an AI-powered fundraising optimization system for La Capitolare Library in Verona, Italy (the world's oldest operational library). The system leverages artificial intelligence to enhance donor acquisition, optimize fundraising campaigns, and provide personalized engagement for potential supporters.

**Current Development Status**

**Backend (.NET Core)**

* **Architecture**: Clean architecture with domain models, repositories, and controllers
* **Database**: Entity Framework Core with SQL Server, configured with proper relationships
* **Domain Models**: Implemented core entities (Donor, Manuscript, DonorScore, etc.)
* **Repositories**: Generic repository pattern with specialized repositories for donors and manuscripts
* **API Controllers**: Basic controllers for donors, manuscripts, and campaign simulation
* **Seed Data**: Sample data for testing including manuscripts, donors, and donor scores

**Frontend (Angular)**

* **Project Structure**: Organized with core, shared, and feature modules
* **Service Layer**: API service for backend communication and mock service for testing
* **Components**: Initial implementation of three core features:
  1. Donor Intelligence Dashboard - Shows AI-scored donor data
  2. Manuscript-Donor Matching - Demonstrates AI matching manuscripts to potential donors
  3. Campaign Simulator - Simulates fundraising campaigns with AI optimization

**Technical Challenges**

* Angular dependency compatibility issues with charting libraries
* TypeScript errors in component implementations
* HTML/template syntax issues in some components
* Challenges with type safety in the simulation results component

**Core Features Implementation Status**

**1. Donor Intelligence Dashboard**

* Basic UI structure implemented
* Donor scoring display partially working
* Visualization components need refinement

**2. Manuscript-Donor Matching**

* Basic matching functionality implemented
* UI for displaying matching donors in place
* AI insight generation implemented in mock service

**3. Campaign Simulator**

* Form for campaign parameters implemented
* Simulation logic in mock service working
* Results visualization component needs fixing

**Data Models Overview**

* **Donors**: Basic donor information, interests, and capacity metrics
* **Manuscripts**: Historical manuscript data with preservation needs
* **Donor Scores**: AI-generated affinity scores and predictions
* **Campaign Simulations**: Parameters and results for fundraising simulations

**Next Implementation Steps**

1. Resolve component errors in simulation results display
2. Implement simplified visualizations that work with current Angular version
3. Complete integration between components
4. Enhance the mock data to better demonstrate AI capabilities
5. Ensure proper data flow between backend and frontend

**Technical Stack**

* Backend: .NET Core 6.0+ (Web API, ML.NET)
* Frontend: Angular 14+ with Material components
* Database: SQL Server
* Styling: Tailwind CSS (partially implemented)

**Pretotype Goal**

The immediate goal is to have a working demonstration that showcases how AI can enhance fundraising efforts through the three core features, with sufficient realism to validate the concept with La Capitolare stakeholders.