**1. Project Structure**

* **Frontend (HTML, CSS, JavaScript):**
  + **User Interface:**
    - **Homepage:** Display the DAC's vision, mission, core principles, and key features.
    - **Membership Page:** Information about membership tiers, benefits, and how to acquire NFTs.
    - **Content Hub:**
      * Display and search for content (text, images, videos, etc.).
      * Content filtering and sorting options.
      * User profiles and content creation tools.
    - **Governance Page:**
      * View and vote on proposals.
      * Track governance decisions and amendments.
      * Access information about the DAO's constitution.
    - **Impact Investing Page:**
      * View investment proposals, track portfolio performance, and access impact reports.
      * Information about the DAC's impact investing criteria and methodology.
    - **Community Forum:**
      * Discussion boards for members to interact, share ideas, and provide feedback.
  + **User Authentication:**
    - Implement secure user authentication and authorization using JavaScript and suitable libraries.
    - Integrate with blockchain wallets for NFT and token management.
* **Backend (Java):**
  + **API Development:**
    - Create a RESTful API using Java (e.g., Spring Boot) to handle:
      * User authentication and authorization.
      * Content management (upload, storage, retrieval, and moderation).
      * Governance processes (proposal submission, voting, result tabulation).
      * Impact investing data management and reporting.
      * Community forum interactions.
    - **Blockchain Interaction:**
      * Develop Java libraries to interact with the Solidity smart contracts on the blockchain.
      * Handle token transfers, NFT minting and transfers, and on-chain governance actions.
      * Integrate with blockchain oracles for real-time data and off-chain computations.
  + **Database:**
    - Utilize a suitable database (e.g., PostgreSQL, MySQL) to store user data, content metadata, community forum posts, and other relevant information.
* **Smart Contracts (Solidity):**
  + **Membership NFTs:**
    - Implement ERC-1155 or similar standards for tiered membership NFTs.
    - Define access levels and privileges associated with each NFT tier.
  + **Governance Token:**
    - Implement ERC-20 token standard with appropriate tokenomics.
    - Define voting power based on token holdings and membership tiers.
  + **Constitution:**
    - Encode the DAO's constitution as a set of rules and parameters within smart contracts.
    - Implement on-chain governance mechanisms for proposing and voting on amendments.
  + **Investment Contracts:**
    - Implement smart contracts for managing investment funds and distributing returns.

**2. Key Considerations**

* **User Experience (UX):**
  + Design an intuitive and user-friendly interface that is accessible to a wide range of users.
  + Ensure a smooth and seamless user experience across all devices.
* **Security:**
  + Implement robust security measures to protect user data and prevent fraud.
  + Regularly audit and update security protocols.
* **Scalability:**
  + Design the system to handle a growing number of users and transactions.
  + Consider using scalable blockchain solutions and efficient database technologies.
* **Decentralization:**
  + Minimize reliance on central servers and data centers.
  + Explore decentralized storage solutions (e.g., IPFS) for content hosting.
* **Fair Use Implementation:**
  + Develop robust mechanisms for enforcing fair use principles and protecting intellectual property rights.
  + Integrate with copyright and licensing APIs.

**3. Development Roadmap**

1. **Prototype Development:**
   * Build a basic prototype of the frontend and backend with core functionalities.
   * Develop and deploy initial smart contracts.
2. **Community Testing & Feedback:**
   * Conduct thorough testing and gather feedback from the community.
   * Iterate on the design and functionality based on user input.
3. **Scalability & Performance:**
   * Optimize performance and scalability as the community grows.
   * Implement load balancing and caching strategies.
4. **Continuous Improvement:**
   * Regularly update and improve the platform based on user needs and technological advancements.
   * Continuously refine the governance structure and amend the constitution.

**Note:** This is a high-level overview. The actual implementation will involve many more details and considerations.

This project requires a multidisciplinary team with expertise in frontend development (HTML, CSS, JavaScript), backend development (Java), blockchain development (Solidity), and potentially other areas like UX/UI design, database administration, and security.

I hope this comprehensive outline provides a solid foundation for your project. Feel free to ask if you have any further questions.