Amnaya Website Project Plan

1. Project Objectives

The primary objective of the Amnaya website is to provide a cohesive digital platform that accurately represents the parent company and its subsidiaries, while serving as a hub for its projects and business initiatives. The website will act as an informative and engaging resource for visitors, including potential investors, partners, and job seekers.

Specific objectives include:

- Showcasing Amnaya's "Core Projects," "Turnkey Projects," and "Center of Excellence."
- Providing detailed descriptions and downloadable PDF and PPT resources for each project.
- Creating media, investor relations, and career pages to enhance transparency and engagement.
- Ensuring an easy transition from the React.js framework to Next.js, taking advantage of Next.js's enhanced performance and SEO capabilities.

2. Key Deliverables

- **What We Do** Page: Displaying a summary of Amnaya's "Core Projects," "Turnkey Projects," and "Center of Excellence" with links to dedicated project pages.
- **Project Pages**: Each project will have a dedicated page with detailed descriptions and downloadable resources (PDF and PPT).
- **Media Page**: A section containing press releases, news, and multimedia content related to Amnaya and its subsidiaries.
- **Investor Relations Page**: Information and reports relevant to investors, including financial statements, press releases, and downloadable reports.
- **Career Page**: Job listings, application forms, and information about working with Amnaya.
- **Next.js Transition**: The new architecture will be implemented using Next.js, ensuring better SEO, performance optimization, and server-side rendering.
- **Responsive, User-friendly Design**: The design will reflect Amnaya's brand identity and offer an intuitive user experience across all devices.

3. Timeline

The estimated duration for this project is **8-9 weeks**, broken down into phases:

Phase 1: Planning (Week 0-1)

- **Tasks**:
    - Finalize design using Figma and technical specifications.

Phase 2: Development (Week 2–7)

- **Tasks**:
    - Implement static website design as per Figma design.
    - Backend integration (with database/API).
    - Ensure Next.js-specific optimizations (server-side rendering, static generation, SEO).

Phase 3: Testing and Quality Assurance (Week 7-8)

- **Tasks**:
    - Functional testing of all pages.

4. Team Members and Their Roles

- Frontend Developer
- Backend Developer
- UI/UX Designer
- Content Writer

5. Resources Required

- **Technical Tools**:
    - Next.js for frontend development.
    - Cloud hosting solution (Vercel, AWS or DigitalOcean) to deploy the website.
    - Strapi CMS or headless CMS integration for backend content management or can create own admin panel.
- **Design Resources**:
    - Design tools like Figma.
- **Project Management**:
    - Tools like Jira or Trello for task management and collaboration.
- **Development Resources**:
    - Cloud storage for hosting PDF/PPT downloads.

6. Potential Risks and Mitigation Strategies

- **Risk**: Delays in design approval may delay development.
    - **Mitigation**: Set strict deadlines for design reviews and approvals. Regular updates with the client to minimize delays.
- **Risk**: Performance issues post-launch due to heavy media content.
    - **Mitigation**: Use Next.js's image optimization and server-side rendering (SSR) features to minimize load times.

7. Benefits of Switching from React.js to Next.js

- **Server-side Rendering (SSR)**: Next.js offers SSR, improving the website's performance and loading speeds. This is crucial for dynamic content such as project pages, investor relations, and media-heavy sections.
- **Static Site Generation (SSG)**: Certain pages (like downloadable resources and careers) can benefit from static site generation, ensuring quicker load times.
- **Enhanced SEO**: SSR and SSG inherently improve SEO rankings. This will make Amnaya's website more discoverable, attracting more investors and potential employees.
- **Better Performance**: Next.js offers better caching and performance optimization compared to a traditional React.js setup.
- **Ease of Maintenance**: Next.js allows incremental static regeneration, meaning specific pages can be re-rendered as needed without redeploying the entire site. This will ease the process of updating content (such as media or careers).

8. Some key points.

- **List of pages:**
    1. Home
    2. What we do
        - Core projects
            - Project detail page
        - Turnkey projects
            - Project detail page.
        - Center of excellence
    3. Career
        - Life At AMNAYA
        - Network Companies
        - Job listing
        - Application form
    4. Investors
    5. Financial information
        - Convertible Bond
    6. Leadership & Governance
    7. Financial Calendar
    8. Contact
    9. Media
        - AMNAYA News
        - Portfolio News
    10. Privacy Policy.
- Figma design and content should be freezed before starting development.
- If we want to implement register and login pages then we've to provide purpose of authentication.
- We've to create an Admin panel for job listing page or can use external forms like Google forms.