📌 System Design Documentation

Introduction

When building the Alma Leads App, I wanted to create a system that was intuitive while keeping things simple. The app consists of two main parts: a public lead submission form and an internal lead management system guarded by authentication. To achieve this, I carefully selected tools that would balance performance, developer experience, and maintainability. Below, I break down my design choices and why they made sense for this project. 🚀

1. Handling Form Submissions with Formidable

Initially, I considered using `useState` to handle form input and then sending the data to an API. However, that wouldn't work well for file uploads, which is also a part of the form. So, I needed a solution that could handle multipart form data efficiently.

After exploring options, I chose Formidable, which is specifically designed for handling file uploads in Next.js API routes. It allows me to parse incoming form data without blocking the main thread.

- 🔷 Uploaded files are stored in `public/uploads/`.
- 🔷 The directory is dynamically created if it doesn't exist, preventing errors.
- 🔷 Error handling ensures smooth file processing.
- 🔷 I added logging to track successful uploads.

2. Mock Authentication for Admin Panel

Authentication was a key aspect of this project because the internal leads list should only be accessible to authorized users. However, since full authentication (e.g., OAuth, JWT) wasn't needed, I opted for a mock authentication system using React Context API.

- 🔷 The authentication state is stored in Context and accessed via a custom hook (`useAuth`).
- 🔷 Since this is a mock system, auth resets on refresh, meaning there's no persistent login session.
- 🔷 Admins can also log out, which clears the authentication state.

3. Seamless Navigation with Next.js Router

Instead of manually managing page transitions, I used file-based routing in Next.js and `useRouter`. This allows for smooth client-side navigation without unnecessary reloads.

- `useRouter.push()` is used to programmatically navigate between pages.
- `<Link>` is used for faster client-side navigation.
- The welcome page (`/`) acts as a hub, making it easier to access different sections.

4. Managing Leads via Next.js API Routes

A crucial decision was how to store and update leads. Using `useState` for temporary storage would mean data is lost on refresh. Instead, I opted for Next.js API routes to handle lead retrieval and updates.

- Leads are fetched via API calls to maintain persistence.
- Some lead starts as `PENDING` and can be manually updated to `REACHED_OUT`. While some leads are initialized as `REACHED_OUT`.
- API endpoints ensure data remains available across sessions.

5. Why I Chose Material React Table

Displaying a list of leads in a searchable, sortable, and filterable format is crucial for usability. Instead of manually implementing these features, I decided to use Material React Table.

- Built-in search, sorting, and column filtering.
- Supports responsive tables with fixed headers.
- Easy to integrate with Next.js.

6. Structuring the App with Next.js Pages

Instead of using complex client-side state management for routing, I relied on page-based routing in Next.js. This made navigation straightforward and kept the project structure clean.

- `/` → Welcome Page
- `/lead-form` → Public Lead Submission
- `/leads` → Internal Admin Panel

7. Leveraging TypeScript for Type Safety

As someone who has faced runtime errors due to weakly typed data, I made it a priority to use TypeScript. This ensures every component and API response has a clearly defined structure.

Defined an interface for leads to prevent inconsistent data:

```
interface Lead {
    id: number;
    name: string;
    submitted: string;
    status: string;
    country: string;
}
```

8. Styling with Styled Components

Instead of using plain CSS or Tailwind, I opted for styled-components (from @mui/system). This approach allowed me to create reusable, theme-friendly UI elements.

- Used styled buttons to update lead status dynamically.
- Ensured consistent design across components.

```
const GreenButton = styled(Button)({
    color: "#046135",
    "&:hover": "none",
    textTransform: "none",
    width: "70%"
});
```

9. Handling Validation & Errors

To prevent users from submitting incomplete forms, I added form validation with error message.

- Users are notified if required fields are missing.
- File uploads are restricted to prevent invalid formats.
- Errors are logged for debugging purposes.

10. Why I Added a Welcome Page

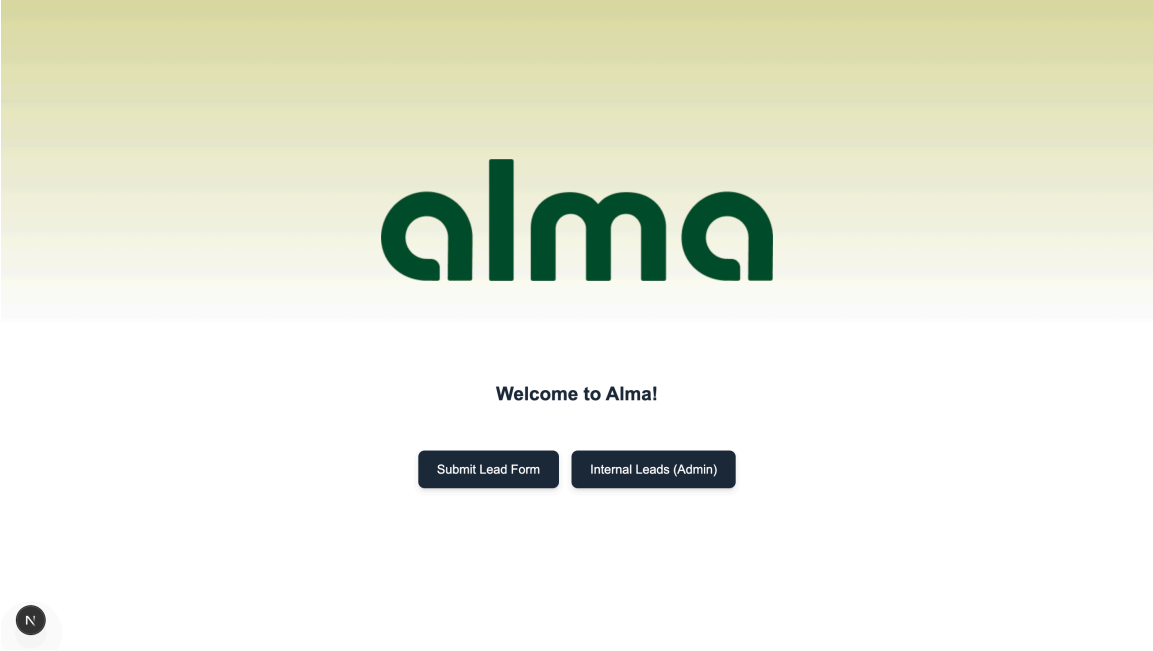
To simplify navigation between the public and admin sections, I included a Welcome Page as a central hub. This makes it easy for users to choose their path without manually entering URLs.

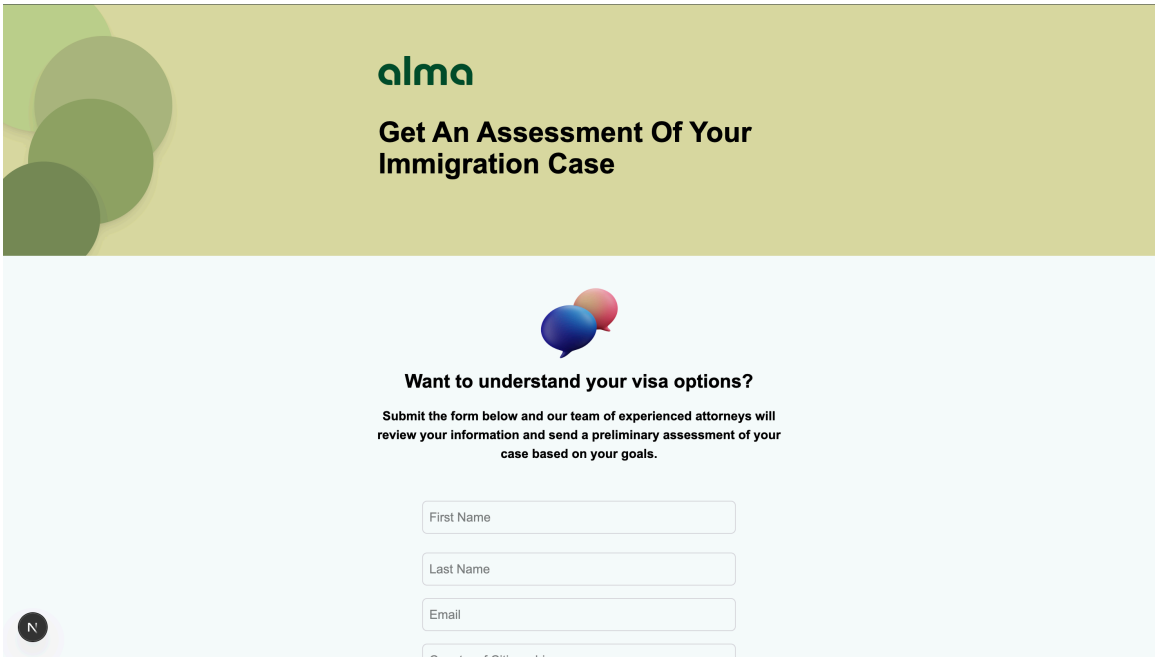
🔷 Provides two clear buttons: Submit Lead Form and Internal Leads (Admin).

🔷 Improves user experience by avoiding unnecessary page reloads.

Conclusion

This project was designed with a balance of simplicity and efficiency. By using Next.js, TypeScript, Formidable, Material React Table, and Context API, I was able to build a structured, user-friendly, and maintainable application. 🚀





LinkedIn / Personal Website URL

Upload Resume / CV

Visa categories of interest?

- [ ] O-1
- [ ] EB-1A
- [ ] EB-2 NIW
- [ ] I don't know

How can we help you?

What is your current status and when does it expire? What is your past immigration history? Are you looking for long-term permanent residency of short term employment visa or both? Are there any timeline considerations?

Submit

alma

Leads

Settings

Admin Logout

Leads

Search

All

Name	Submitted	Status	Action	Country
Carlos Mendoza	02/02/2024, 3:15 PM	Pending	Change Status	Argentina
Sofia Petrov	02/02/2024, 3:30 PM	Contacted	Reached Out	Russia
Yuki Nakamura	02/02/2024, 3:40 PM	Pending	Change Status	Japan
Ahmed Al-Farsi	02/02/2024, 3:45 PM	Interested	Reached Out	Oman
Ethan Williams	02/02/2024, 4:00 PM	Pending	Change Status	USA
Lisa Müller	02/02/2024, 4:15 PM	Pending	Change Status	Germany
Hassan Kabir	02/02/2024, 4:30 PM	Follow-up	Reached Out	Pakistan

Rows per page 10 11-20 of 30