[Internship] Kandid - Full Stack Developer Next.js Intern Assignment

Platform UI Replication

Overview

You are tasked with replicating the user interface of two key sections from our Linkbird.ai platform: **Leads** and **Campaigns**. This assignment will test your ability to build modern, responsive web applications using our current tech stack.

Demo Video - 🎬 [Internship]Assignment-video.mp4

Please watch the provided demo video carefully to understand:

- The overall layout and navigation structure
- User interactions and expected behaviors
- Data presentation patterns
- UI/UX design principles used

Required Tech Stack

- **Next.js 15+** - React framework with App Router
- **Tailwind CSS + shadcn/ui** - Styling and component library
- **PostgreSQL + Drizzle ORM** - Database and type-safe ORM
- **Better Auth** - Authentication (credentials + Google OAuth)
- **TanStack Query (React Query)** - Server state management
- **Zustand** - Client-side state management

Assignment Requirements

1. Authentication System

Implement a complete authentication flow using Better Auth:

Features to implement:

- Login page with email/password form
- Google OAuth integration
- User registration functionality

- Protected routes middleware
- Session management
- Logout functionality

Acceptance Criteria:

- Users can sign up with email/password
- Users can sign in with Google OAuth
- Protected pages redirect unauthenticated users to login
- Clean, responsive authentication UI
- Proper error handling and validation

2. Application Layout & Navigation

Create the main application shell with:

Sidebar Navigation:

- Collapsible sidebar with navigation items
- Active state indicators
- User profile section with logout option
- Navigation items: Dashboard, Leads, Campaigns, Settings

Main Layout:

- Header with breadcrumbs
- Consistent spacing and typography

3. Leads Section

Replicate the leads management interface with all functionality shown in the demo.

Main Leads Table:

- Infinitely scrollable table displaying all leads across campaigns
- Columns should include:
  - Lead Name/Contact
  - Email
  - Company
  - Campaign Name
  - Status (Pending, Contacted, Responded, Converted)
  - Last Contact Date
- Search/filter capabilities
- Loading states and skeleton UI

Lead Detail Side Sheet:

- Clicking on any lead opens a detailed side sheet
- Display comprehensive lead information:

    - Contact details
    - Associated campaign information
    - Interaction history
    - Lead status and progression
    - Action buttons (Contact, Update Status)
- Smooth slide-in animation
- Close functionality (X button, ESC key, click outside)

4. Campaigns Section

Build the campaigns overview and management interface.

Campaigns Table:

- Display all campaigns in a structured table format
- Columns should include:
    - Campaign Name
    - Status (Draft, Active, Paused, Completed)
    - Total Leads
    - Successful Leads
    - Response Rate (%)
    - Progress Bar
    - Created Date
    - Actions (Edit, Pause/Resume, Delete)
- Sortable columns
- Campaign status filters

Campaign Statistics:

- Progress indicators for each campaign
- Success rate calculations
- Visual progress bars
- Color-coded status indicators
- Summary cards showing overall metrics

Technical Requirements

Database Schema

Design and implement the following database tables using Drizzle ORM:

-- Users table (handled by Better Auth)
-- Campaigns table
-- Leads table

State Management

- Use **Zustand** for:
    - Sidebar collapse state
    - Selected leads/campaigns
    - Filter and search states
    - UI state (modals, side sheets)
- Use **TanStack Query** for:
    - Data fetching and caching
    - Infinite scrolling for leads table
    - Optimistic updates
    - Background refetching

Performance Considerations

- Optimize database queries with proper indexing
- Use React.memo and useMemo where appropriate
- Implement proper loading states and error boundaries.

Design Requirements

UI/UX Standards

- Follow the exact design patterns shown in the demo video
- Use shadcn/ui components consistently
- Implement proper hover states and transitions

Color Scheme & Branding

- Match the Linkbird.ai color palette from the demo
- Use consistent spacing (Tailwind spacing scale)
- Implement dark/light theme support (bonus)
- Professional, clean interface

Deliverables

1. Source Code

- Complete Next.js application
- Well-organized file structure
- Clean, commented code

- TypeScript throughout
- Git repository with meaningful commit history

2. Documentation

- README.md with setup instructions
- API documentation
- Database schema documentation
- Deployment instructions

3. Demo Deployment

- Deploy to Vercel or similar platform
- Provide live demo URL
- Make sure Sign-up and registration works

Evaluation Criteria

Technical Excellence (40%)

- Code quality and organization
- Proper use of TypeScript
- Performance optimization
- Error handling
- Security best practices

Feature Completeness (30%)

- All required features implemented
- Matches demo functionality
- Smooth user interactions
- Data persistence

UI/UX Quality (20%)

- Visual accuracy to demo
- Responsive design
- User experience
- Accessibility

Code Architecture (10%)

- Proper separation of concerns
- Reusable components
- Scalable patterns

- Testing considerations

Submission Guidelines

Timeline

- **Deadline:** 10th September 2025
- **Recommended time investment:** 4-5 development hours

How to Submit

1. Create a public GitHub repository
2. Deploy the application to Vercel
3. Submit the following Google Form - https://forms.gle/skdhcpaAx2rN8ePh6
    - GitHub repository URL
    - Live demo URL
    - Short Video Explaining the product
    - Any additional notes or considerations

Questions?

If you have any questions about the requirements or need clarification on any aspect of the assignment, please don't hesitate to reach out to pulkit@kandid.ai.

Good luck! We're excited to see your implementation and discuss your approach during the technical interview.
