perplexity

Create a customer onboarding Page, that includes a SignUp/SignIn Form for the page for users to signup and sign-in to place their request or submit tickets for SaaS products purchased through a subscription. Add a postgres database where the data will be saved. Once the Users complete SignUp, create a dashboard which they will be redirected. This dashboard should be designed as a CRM for each customer. Finally, Create a button for the customer to submit tickets.

This section outlines the design and flow for a customer onboarding page that enables users to register, log in, and—once authenticated—access a personalized CRM dashboard with ticket submission capabilities. It also defines the Postgres schema for storing user and ticket data.

Onboarding Page Overview

The onboarding page combines sign-up and sign-in forms in a single, streamlined interface to minimize friction and accelerate first-time user engagement [1]. It uses a clean, responsive layout with clear calls to action and social login options.

Form Design and Best Practices

- Minimize Fields: Only request email and password on initial signup, postponing additional profile details until after first login [1].
- Combined Sign-In/Sign-Up Flow: Detect existing emails and either log in or create an account seamlessly without separate pages [2].
- Social Login: Offer Google and GitHub OAuth to reduce typing and boost conversions [1].
- Clear Feedback: Display inline validation messages and a progress indicator for multi-step flows [3].

Page Components

‘use client’ Directive & Styling

All interactive form elements use the `‘use client’` directive and Tailwind CSS for responsive design. Icons are provided by Lucide React.

Component Structure

- `<AuthForm>`: Toggles between signup and login modes, handles form state and submissions.
- `<SocialLoginButtons>`: Renders OAuth options for Google and GitHub.
- `<FormField>`: Reusable input component with label, error text, and proper ARIA attributes.

Postgres Database Schema

```
-- Users table
CREATE TABLE users (
  id SERIAL PRIMARY KEY,
  email VARCHAR(255) UNIQUE NOT NULL,
  password_hash VARCHAR(255) NOT NULL,
  created_at TIMESTAMP WITH TIME ZONE DEFAULT now()
);

-- Tickets table
CREATE TABLE tickets (
  id SERIAL PRIMARY KEY,
  user_id INTEGER REFERENCES users(id),
  subject TEXT NOT NULL,
  description TEXT NOT NULL,
  status VARCHAR(20) DEFAULT 'open',
  created_at TIMESTAMP WITH TIME ZONE DEFAULT now()
);
```

This schema ensures secure storage of credentials and a one-to-many relationship between users and support tickets.

Authentication Flow

1. User visits `/onboarding`—sees combined signup/login form.
2. On **Signup**, the form calls a server endpoint to hash the password and insert into `users`.
3. On **Login**, the form validates credentials against `users`.
4. Successful auth sets an HTTP-only session cookie and redirects to `/dashboard`.

CRM Dashboard Design

Layout Components

- **Header**: Shows user name, logout button, and global navigation (Dashboard, Tickets, Profile).
- **Sidebar**: Collapsible menu linking to “My Tickets,” “New Ticket,” and “Account Settings.”
- **Main Content**:
  - **KPI Cards**: Overview of open tickets and response SLA metrics [4].
  - **Recent Activity Feed**: List of latest ticket updates.

Ticket Submission CTA

A persistent “New Ticket” button in the header and sidebar opens a `<TicketForm>` component. The form includes subject and description fields with real-time validation and “Submit” button.

User Stories

- **As a visitor**, I can sign up with email/password or via OAuth so I can quickly access the service [1].
- **As a returning user**, I can log in without confusion and be redirected to my personalized dashboard [2].
- **As an authenticated user**, I can view an overview of my support tickets and key service metrics [4].
- **As a customer**, I can click “New Ticket,” fill in subject/description, and submit a request that is saved to the database [5].
- **As a user**, I can log out securely and end my session.

This design ensures a low-friction onboarding experience and a CRM-style dashboard for ongoing customer support interactions.

⁂

1. https://www.vakulski-group.com/blog/essay/saas-signup-page-examples/
2. https://www.indiehackers.com/post/6-tips-for-the-perfect-saas-signup-flow-79b8d1d4d1
3. https://procreator.design/blog/saas-onboarding-design-ultimate-checklist/
4. https://www.minimaldashboard.com/blog/our-guide-to-saas-dashboard-features-benefits-examples-tips-for-success
5. https://workdo.io/documents/support-ticket-integration-in-dash-saas/