

SMS Sending System with Web Interface

Luigi Matteo Girke

Agenda

01

User Journey

Everything a user can do

02

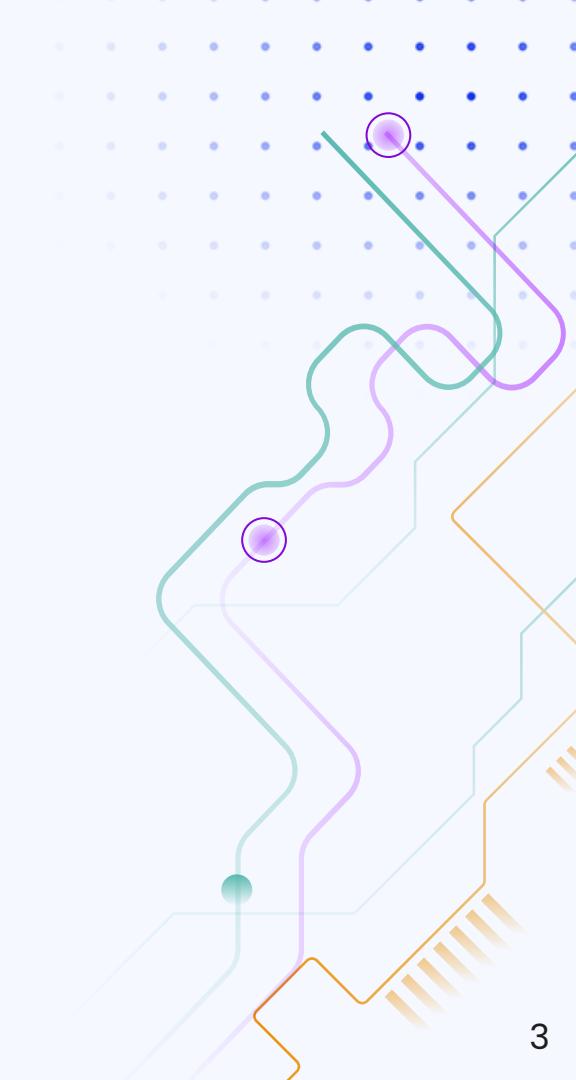
Tech Deep-Dive

How it works under the hood

03

Trials & Successes

Turning problems into solutions



01

User Journey

Everything a user can do

Login

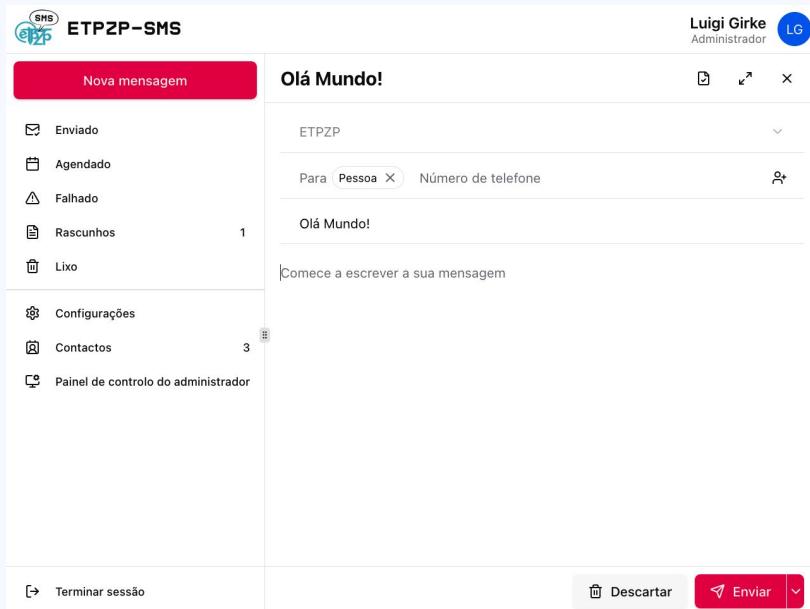
- AD integration for secure login
- No need for separate accounts
- Permission based on AD groups



The image shows a digital login interface for the website etpzp.pt. At the top left is the etpzp logo with a speech bubble containing the word "SMS". Below the logo is the heading "Iniciar sessão" followed by the text "para começar a enviar mensagens SMS". There are two input fields: one for "E-mail" containing the value "2812@etpzp.pt" and another for "Palavra-passe" showing a redacted password. A large black button at the bottom right contains the text "Iniciar sessão".

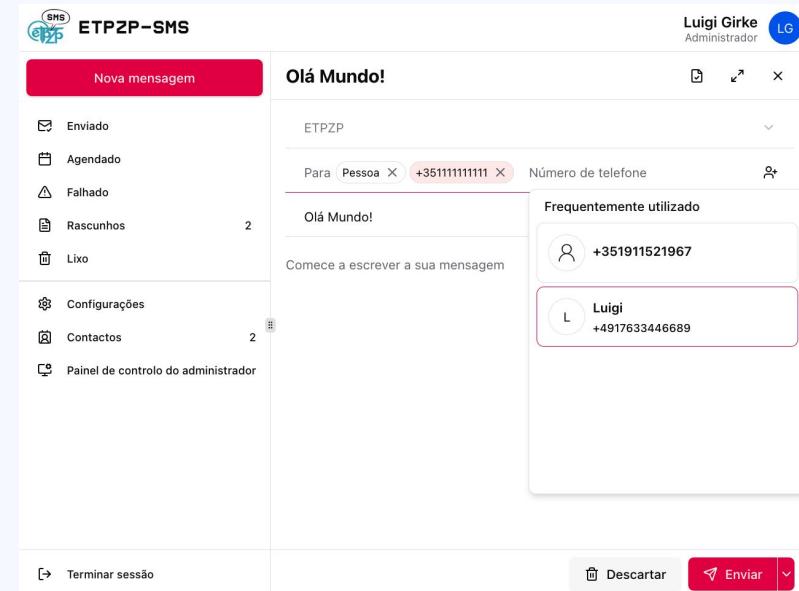
Compose & Send

- Dedicated new message page
- Fields: Sender (fixed), Recipients, Subject, Content
- Send immediately or schedule
- Manual + automatic draft saving



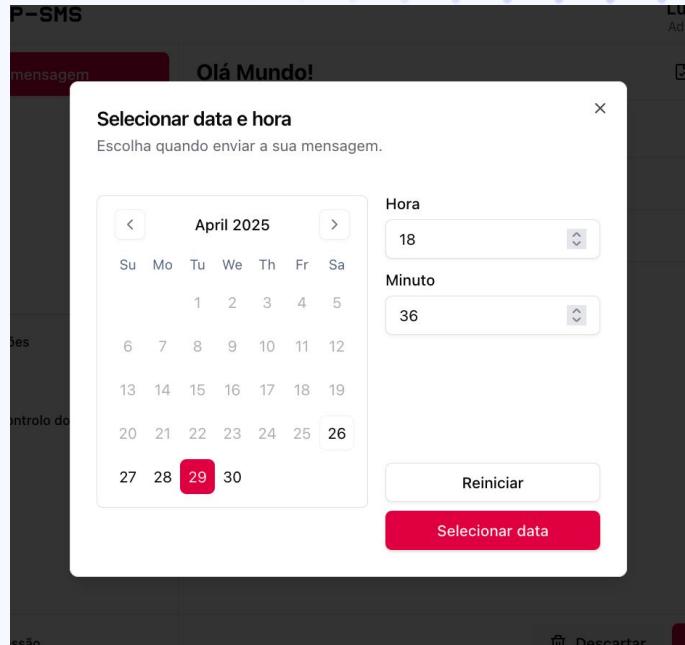
Smart Recipient Input

- Add multiple recipients with keyboard or mouse
- Real-time suggestions from contacts & past recipients
- Error handling for invalid or duplicate numbers



Scheduling Messages

- Choose future date & time
- Preview message status before send
- Cancel anytime before scheduled time



Message Management

- View: Sent, Drafts, Scheduled, Failed, Trash
- Each message shows delivery status, and recipients
- Ability to filter by name and content

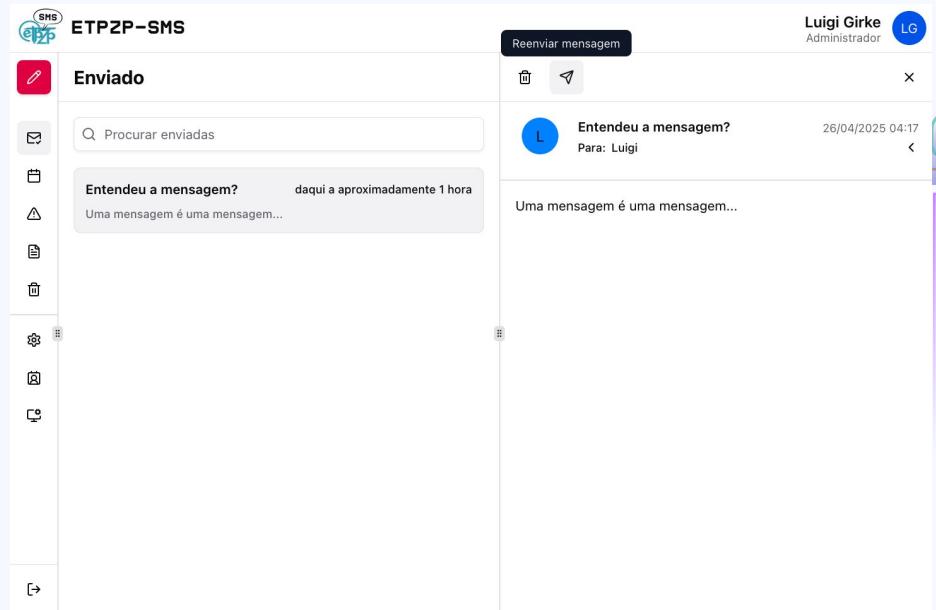
The screenshot displays the ETPZP-SMS application interface. At the top, there's a header with the logo 'ETPZP-SMS' and a user profile for 'Luigi Girke Administrador'. Below the header, the main area is titled 'Rascunhos' (Drafts). A search bar is present with the placeholder 'Procurar rascunhos'. Two draft messages are listed:

- A message to 'Olá Mundo!' scheduled for 37 minutes from now.
- A message with 'Nenhum assunto' scheduled for 36 minutes from now.

On the left side, there's a vertical sidebar with icons for creating new messages, managing contacts, and settings. At the bottom right of the main window, there's a red button labeled 'Continuar rascunho' (Continue draft).

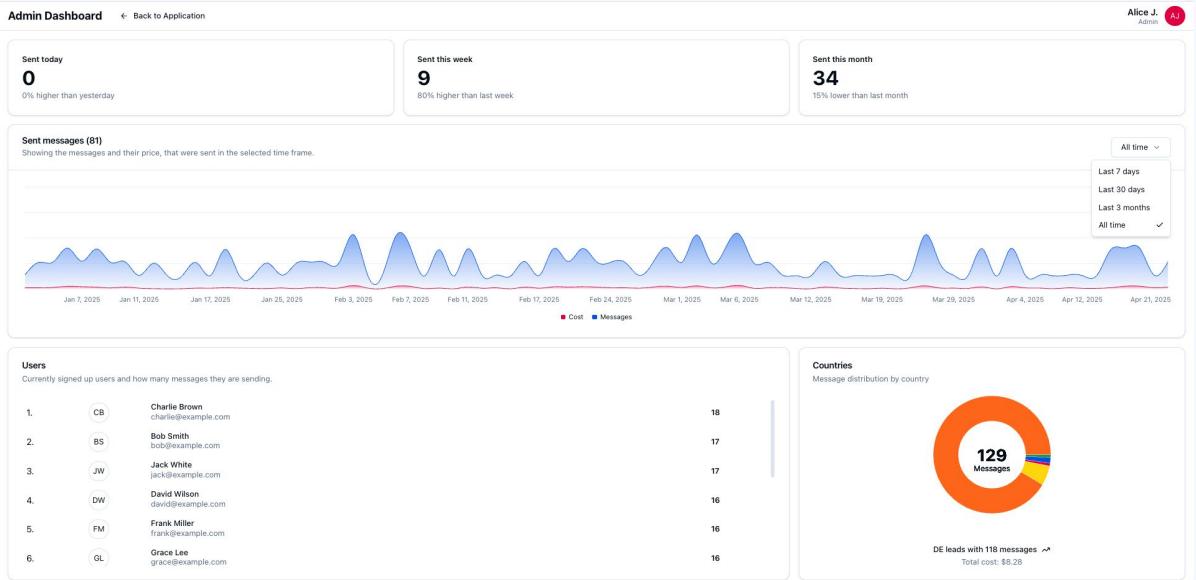
Full Message Actions

- Resend or edit messages
- Move to trash or recover
- Cancel scheduled messages
- Detailed recipient chips with profile tooltips



Admin Dashboard

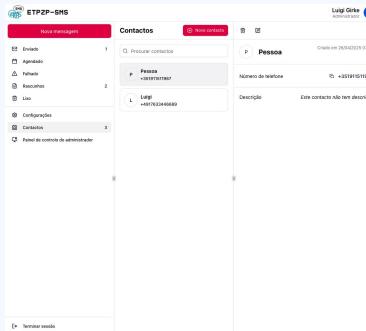
- Stats by user/date/country
- Area and pie charts
- Toggle date range



Other Pages

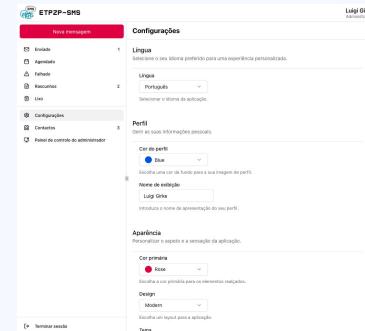
Contacts

- Add, edit, delete contacts
- View contact details



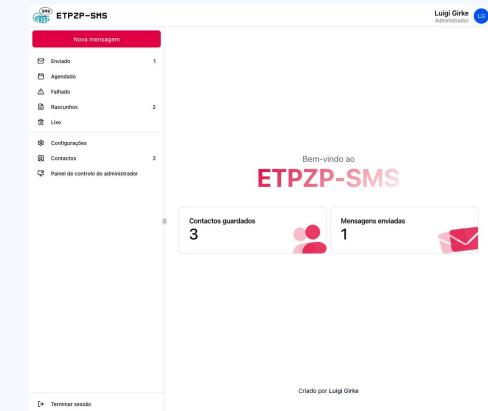
Settings

- Change themes and language
- Update user preferences



Home

- Quick links to pages





02

Tech Deep-Dive

How it works under the hood



Client-Server Model

Client (Browser):

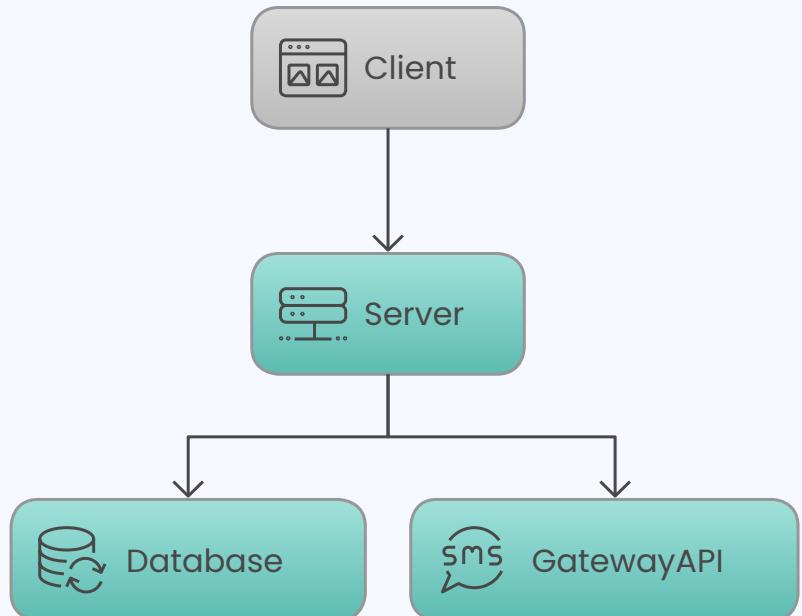
- Renders the UI
- Sends requests to server

Server:

- Handles logic and routing
- Connects to database and GatewayAPI

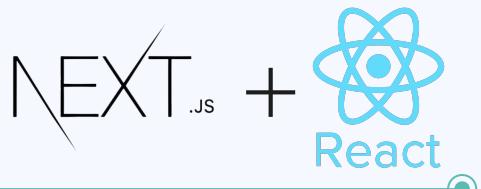
External Services:

- Database: Stores users, messages, contacts
- GatewayAPI: Sends and cancels SMS



Framework Basics

- Built with React + Next.js
- File-based routing
- Layout and server components



UI & Theming

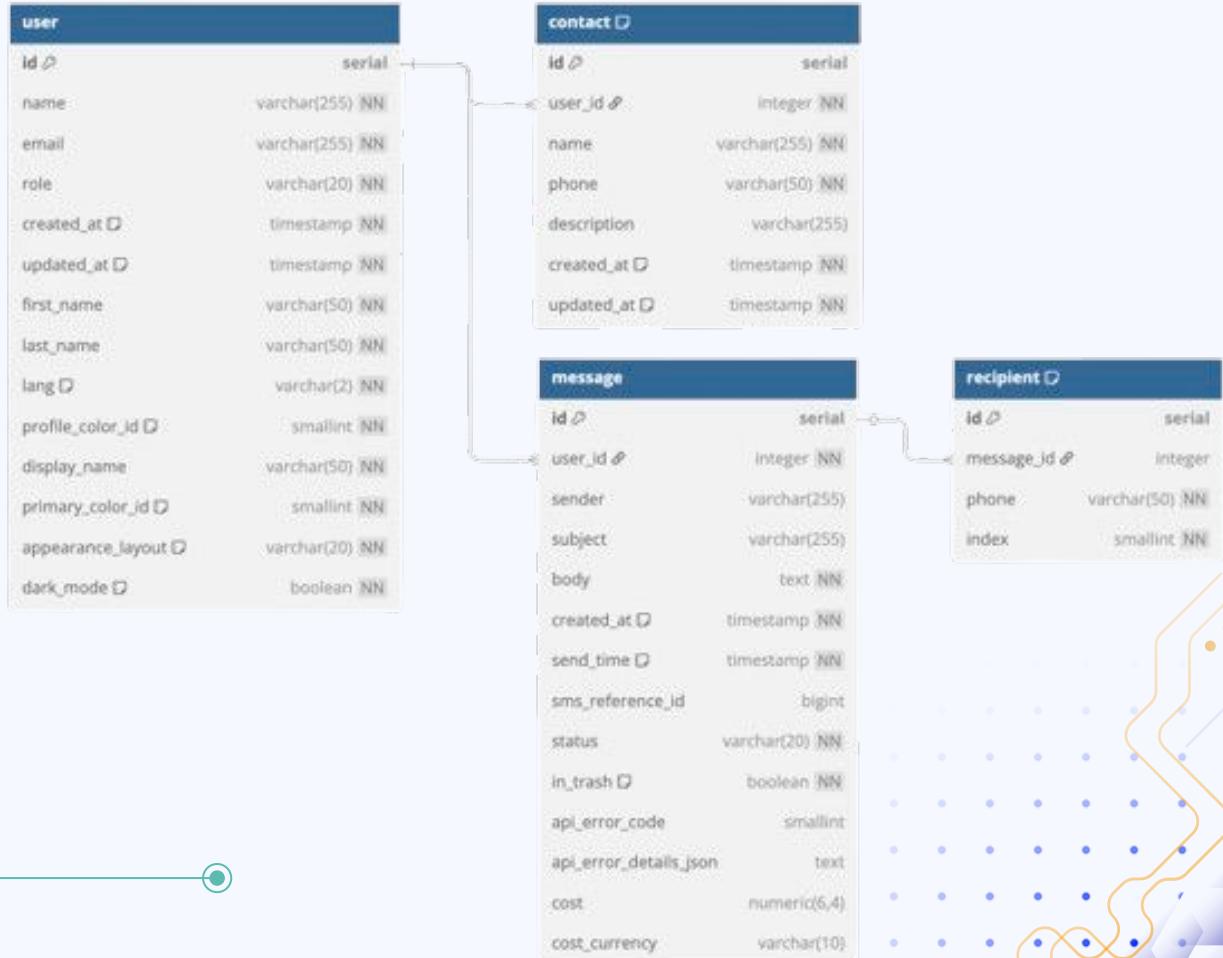
- ShadCN + TailwindCSS
- Resizable panels
- Dynamic light/dark + color themes



The screenshot displays a dark-themed user interface for managing application settings. At the top, there's a navigation bar with the logo 'ETPZP-SMS' and a user profile for 'Luigi Girke'. Below the navigation, a sidebar lists categories: 'Nova mensagem' (1), 'Enviado', 'Agendado', 'Falhado', 'Rascunhos' (2), 'Lixo', 'Configurações' (3), 'Contactos', and 'Painel de controlo do administrador'. The main content area is titled 'Configurações' and contains sections for 'Cor do perfil' (set to 'Blue'), 'Nome de exibição' (set to 'Luigi Girke'), and 'Aparência'. Under 'Aparência', there are sections for 'Cor primária' (set to 'Yellow'), 'Design' (set to 'Modern'), and 'Tema'. The 'Tema' section shows two options: 'Claro' (light mode) and 'Escuro (ativo)' (dark mode, which is selected). A note at the bottom right of the 'Tema' section says 'Selecionar o modo claro ou escuro para a aplicação.' At the bottom left, there's a link 'Terminar sessão'.

Database Schema

- PostgreSQL
- Users, contacts, messages, recipients
- One message table — one status field



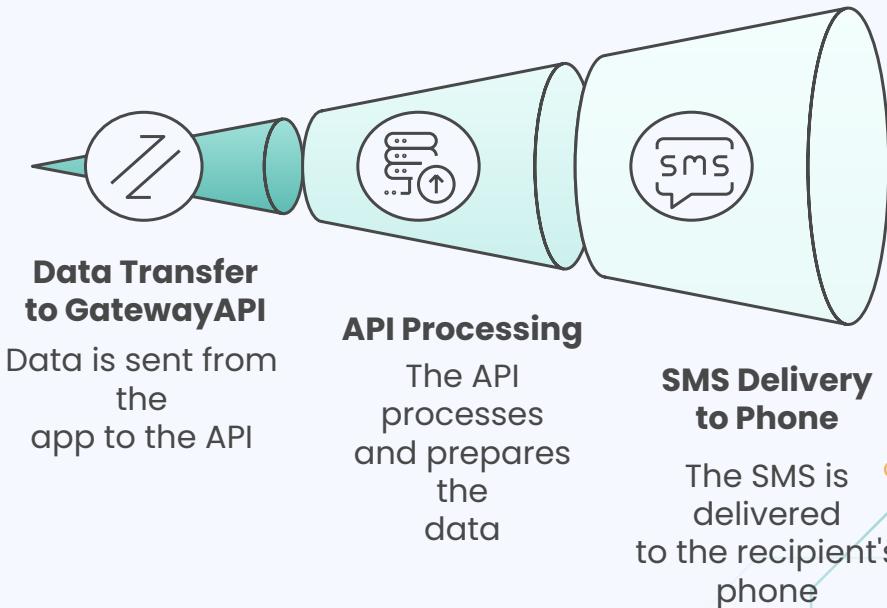
Database Querying

- PostgreSQL + `pg`
- Helper for safe pool-based queries
- Query separation in `/lib/db`

```
lib > db > ts index.ts > ...
1 import { Pool, QueryResult } from "pg";
2
3 const pool = new Pool({
4   host: process.env.POSTGRES_HOST,
5   port: Number(process.env.POSTGRES_PORT),
6   user: process.env.POSTGRES_USER,
7   password: process.env.POSTGRES_PASSWORD,
8   database: process.env.POSTGRES_DB,
9 });
10
11 async function db(query: string, params?: any[]): Promise<QueryResult> {
12   const client = await pool.connect();
13   try {
14     const res = await client.query(query, params);
15     return res;
16   } catch (err) {
17     console.error("Database query error", err);
18     throw err; // Rethrow the error for handling in the calling function
19   } finally {
20     client.release(); // Always release the client back to the pool
21   }
22 }
23
24 export default db;
```

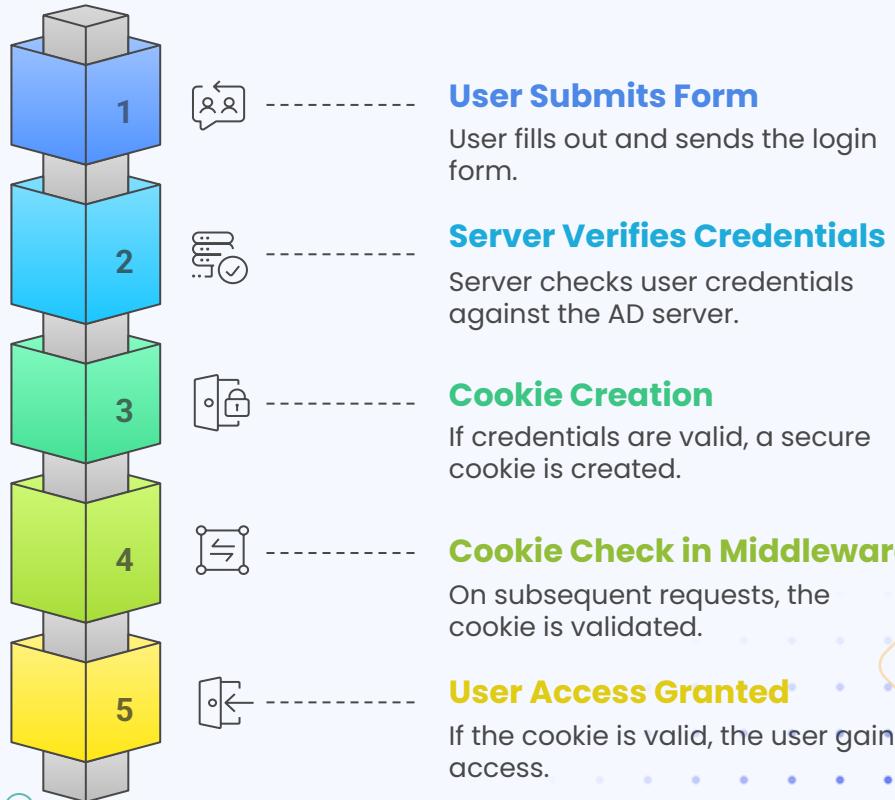
SMS Gateway Integration

- GatewayAPI
- REST calls with error feedback
- Cancel via API for refund
- Used SMSAPI first



Authentication

- Active Directory
- Iron-session for cookies
- Admin detection from AD groups



Self-Hosting



Docker

- Runs node.js server and Postgres in isolated containers
- Ensures consistent environment and easy restarts



Nginx

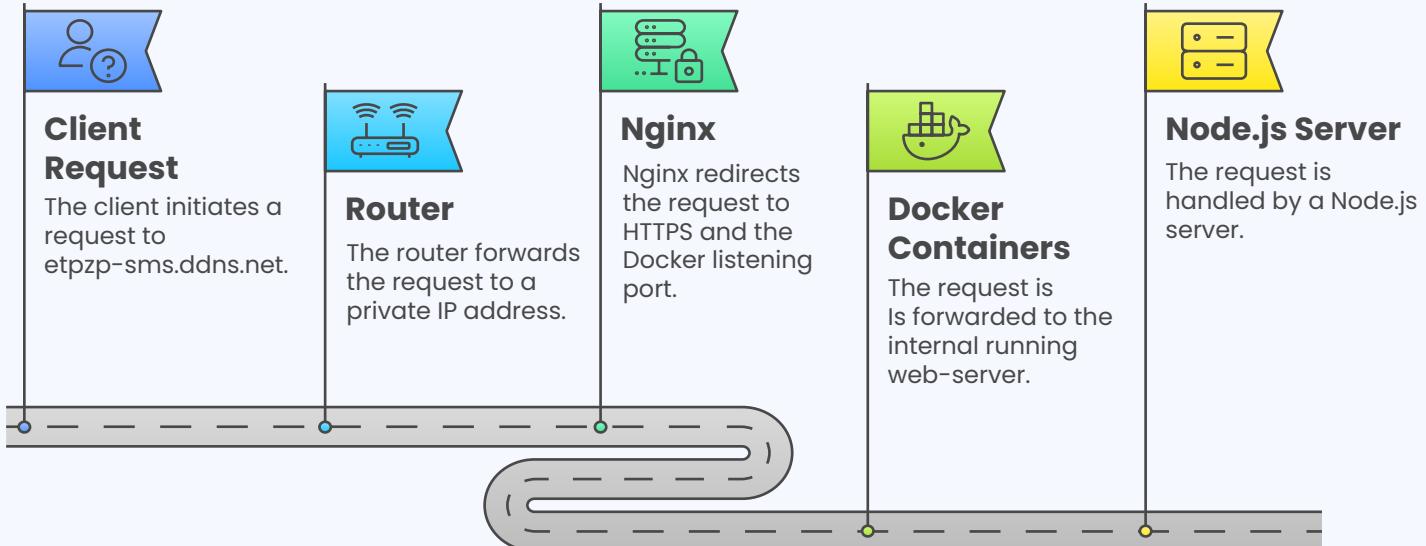
- Reverse proxy for routing traffic to Docker containers
- Sets up SSL and redirects HTTP to HTTPS



No-IP

- Maps public IP to local network
- Enables remote access without a static IP

SMS Application Request Flow



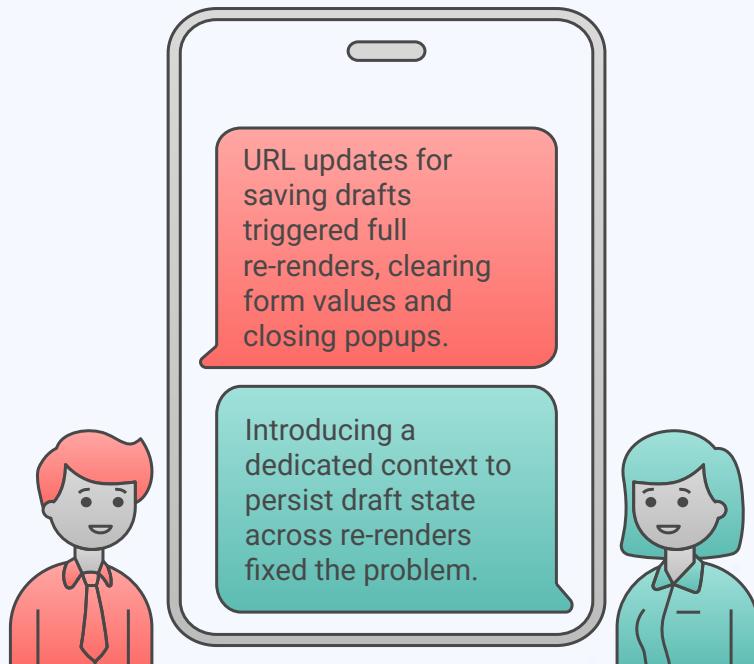


03

Trials & Successes

Turning Problems Into Solutions

Draft Saving Challenge



Recipient Input Complexity

- Building the autocomplete/suggestions dropdown was time-intensive and error-prone
- Ensured up-to-date values by carefully managing context and removal logic

Removal Logic

Context Management

Error-Prone Implementation

Time-Intensive Development

**Effective
Autocomplete**

Settings Architecture Challenge

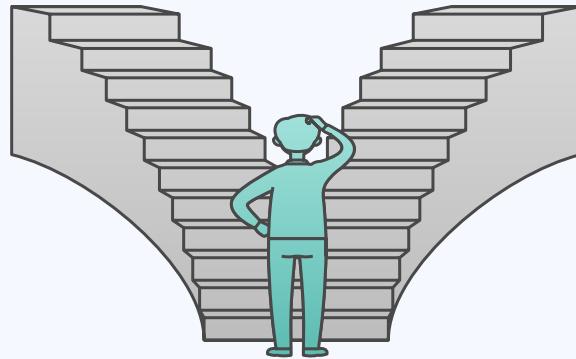
- Local storage writes don't trigger React re-renders
- Settings context enables per-item updates

One Form for All Settings (rejected)

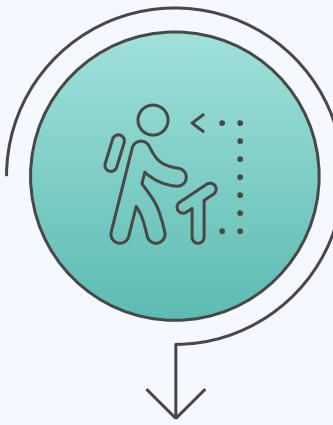
whole-payload saves made validation/error handling cumbersome

Multiple Per-Item Forms (chosen)

per-item saves provide instant persistence, clear errors, and kept UI reactive

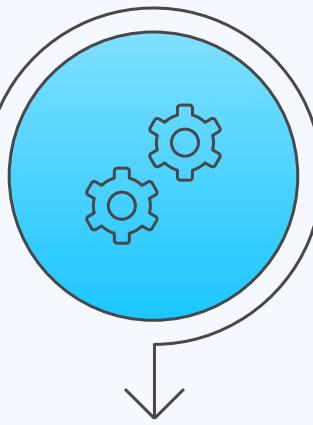


Summary



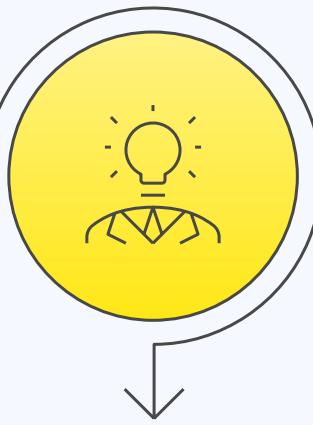
User Journey

Outline of customer experience



Tech Deep-Dive

Technologies used in the project



Trials & Successes

Problems faced and resolutions

Thank you

I'm happy to take your questions now