

# Smart Link Hub

Workflow Guide + Screen-by-Screen Walkthrough (for submission / judges)

Live Deployment (Netlify): <https://smartlink-hub.netlify.app/>

Github Repository: <https://github.com/abhijit-1011/Smart-Link-Hub.git>

Public Hub Example: <https://smartlink-hub.netlify.app/hub.html?slug=Abhishub>

## 1) Project Overview

Smart Link Hub is a Linktree-like app where users create a public hub page (by slug), add multiple links, track hub visits and link clicks, and optionally apply rule-based visibility (Time / Device / Country) per link. All management happens in dashboard.html (login required).

## 2) Directory Structure (as per your project)

```
Smart-Link-Hub/
  └── frontend/
    ├── index.html
    ├── hub.html
    ├── dashboard.html
    ├── about.html
    ├── login.html
    ├── signup.html
    └── assets/
      └── js/
        ├── analytics.js
        ├── auth.js
        └── dashboard.js
```

```
|   └── hub.js  
|   └── qr.js  
|   └── rules.js  
└── supabase.js
```

How files connect (simple map):

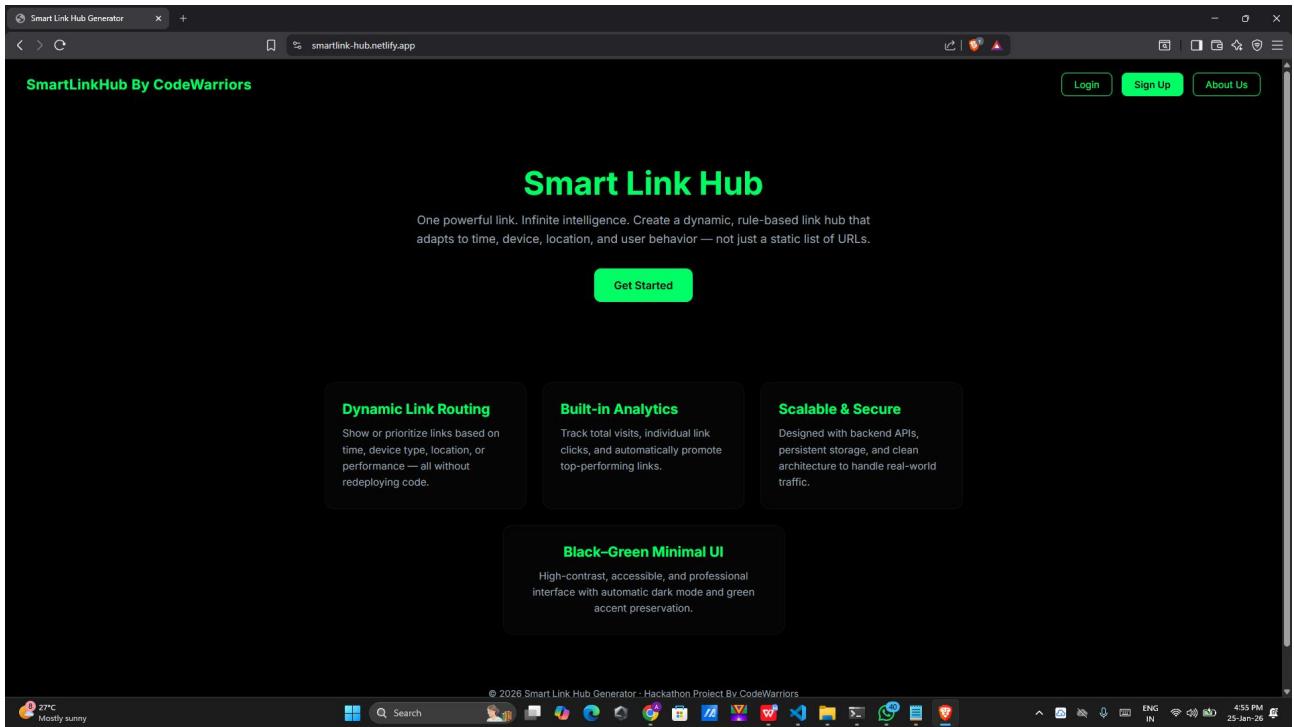
- assets/js/supabase.js → creates Supabase client (URL + ANON KEY). Used by all pages.
- login.html + signup.html → use auth.js (Supabase Auth: sign in / sign up).
- dashboard.html → uses dashboard.js + rules.js + analytics.js (CRUD hub, links, rules, analytics).
- hub.html → uses hub.js + qr.js + rules.js (loads public hub by slug, shows QR, applies rules).
- about.html + index.html → mostly static UI + navigation buttons.

### 3) End-to-End Workflow (User Journey)

Below is the correct order of screens in real usage. Each step includes what the user does and a screenshot.

#### Step 1: Open Home (index.html)

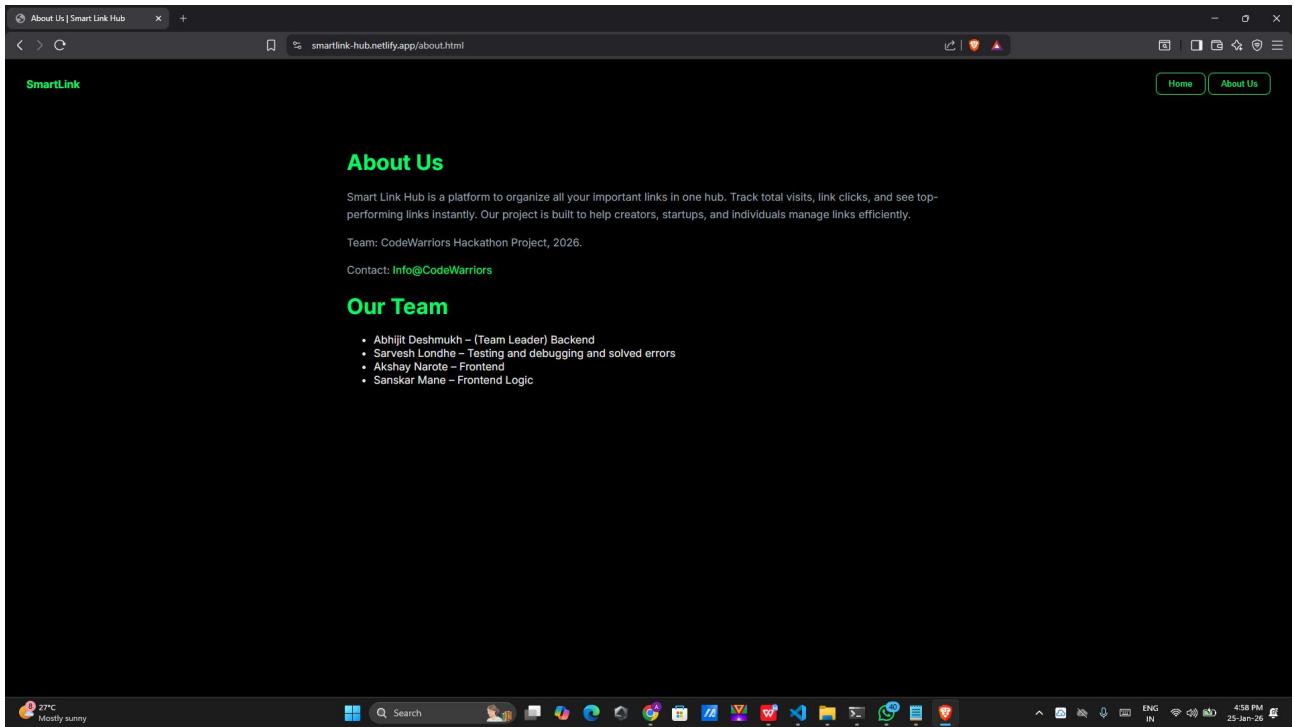
Go to the deployed site. Click \*\*Login\*\* (existing user) or \*\*Sign Up\*\* (new user).



Screenshot: Home page (index.html)

## Step 2: About Page (about.html) – optional

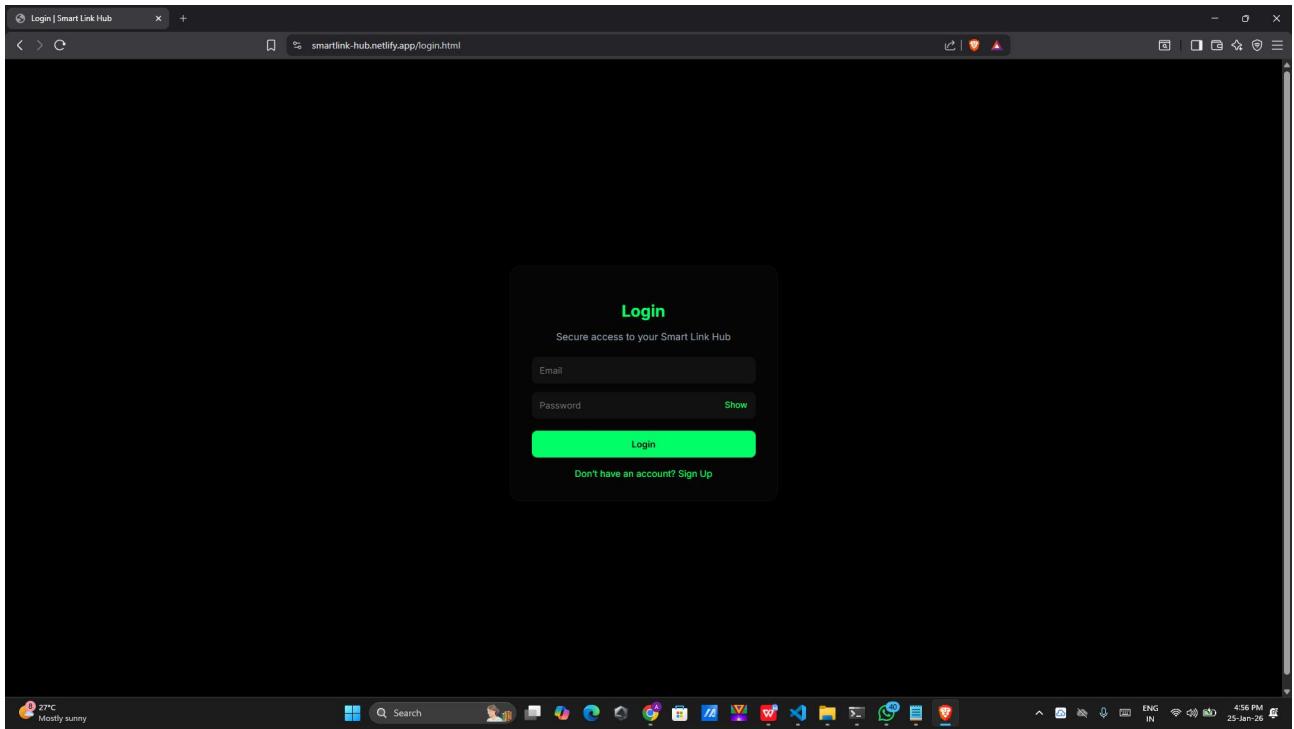
From the top menu, open \*\*About Us\*\* to read project + team details, then return Home.



*Screenshot: About page (about.html)*

### **Step 3: Login (login.html)**

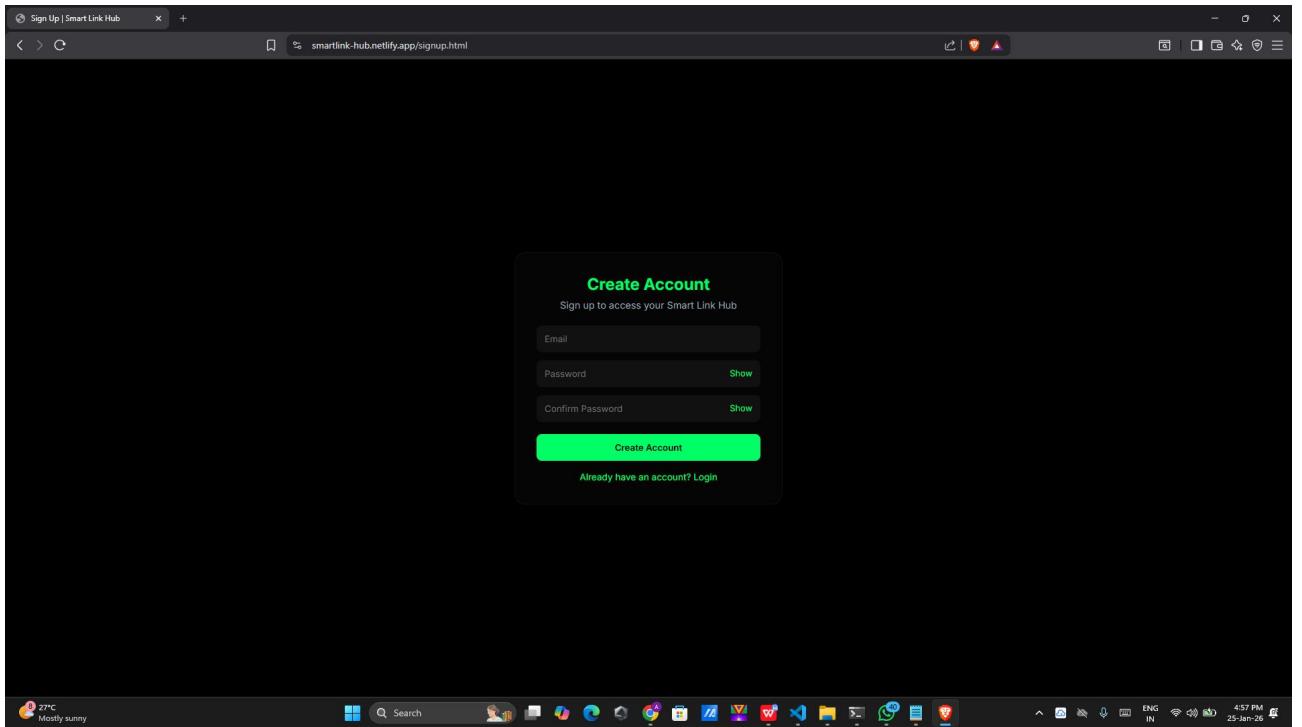
Enter email + password, then click \*\*Login\*\*. After success, user is redirected to \*\*dashboard.html\*\*.



Screenshot: Login page (`login.html`)

#### Step 4: Sign Up (`signup.html`) – only if new

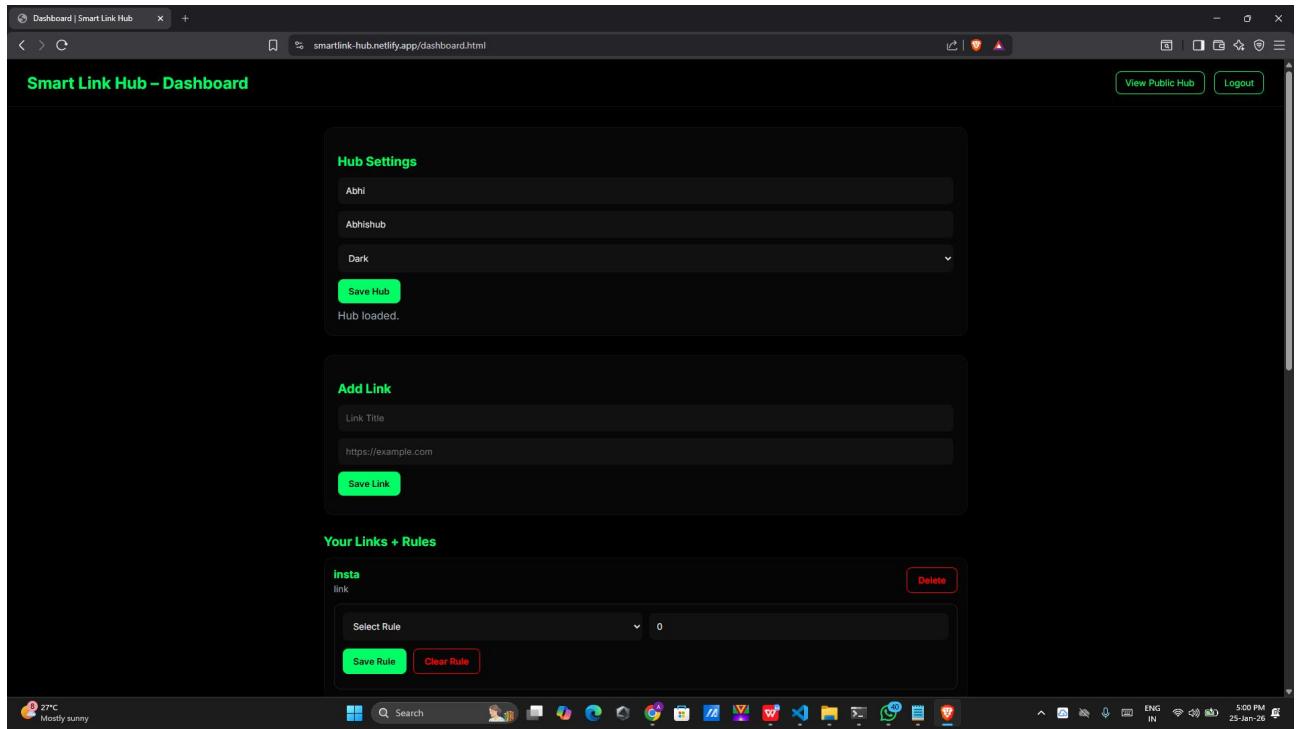
Create account using email + password + confirm password, then login and go to dashboard.



*Screenshot: Signup page (signup.html)*

## **Step 5: Dashboard: Hub Settings + Add Link (dashboard.html)**

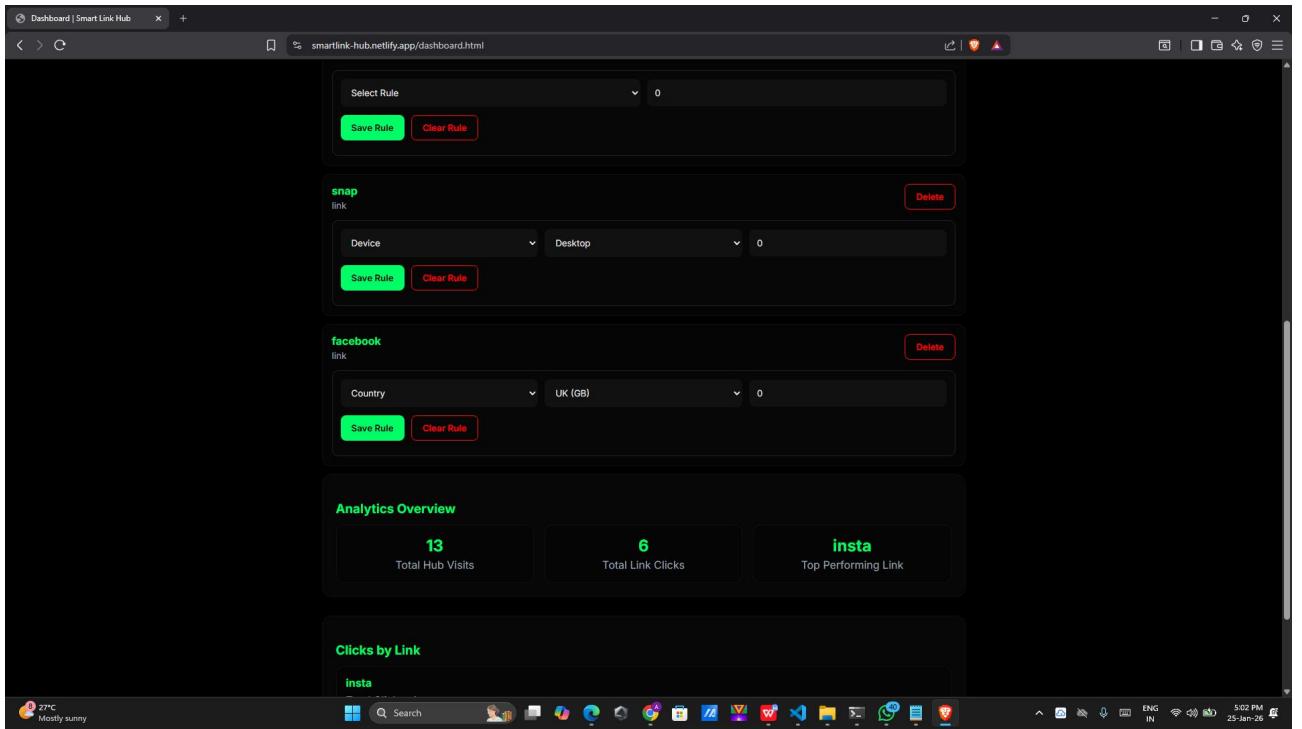
Create or edit your hub (Title, Slug, Theme) and click \*\*Save Hub\*\*. Then add links using \*\*Save Link\*\*.



Screenshot: Dashboard (Hub Settings + Add Link)

## Step 6: Dashboard: Add Rules (Time / Device / Country)

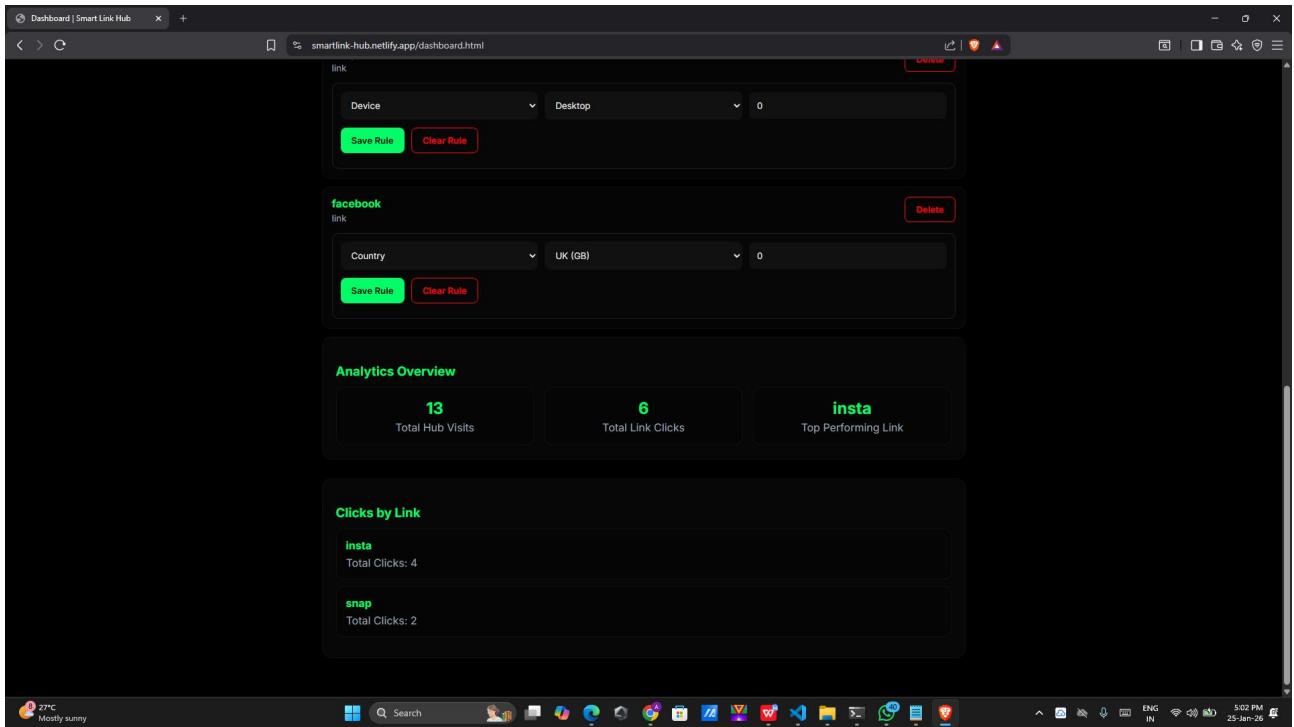
For each link, select a rule type, select the value (e.g., Desktop / UK) and click \*\*Save Rule\*\*. Use \*\*Clear Rule\*\* to remove.



Screenshot: Dashboard rules area

## Step 7: Dashboard: Analytics Overview

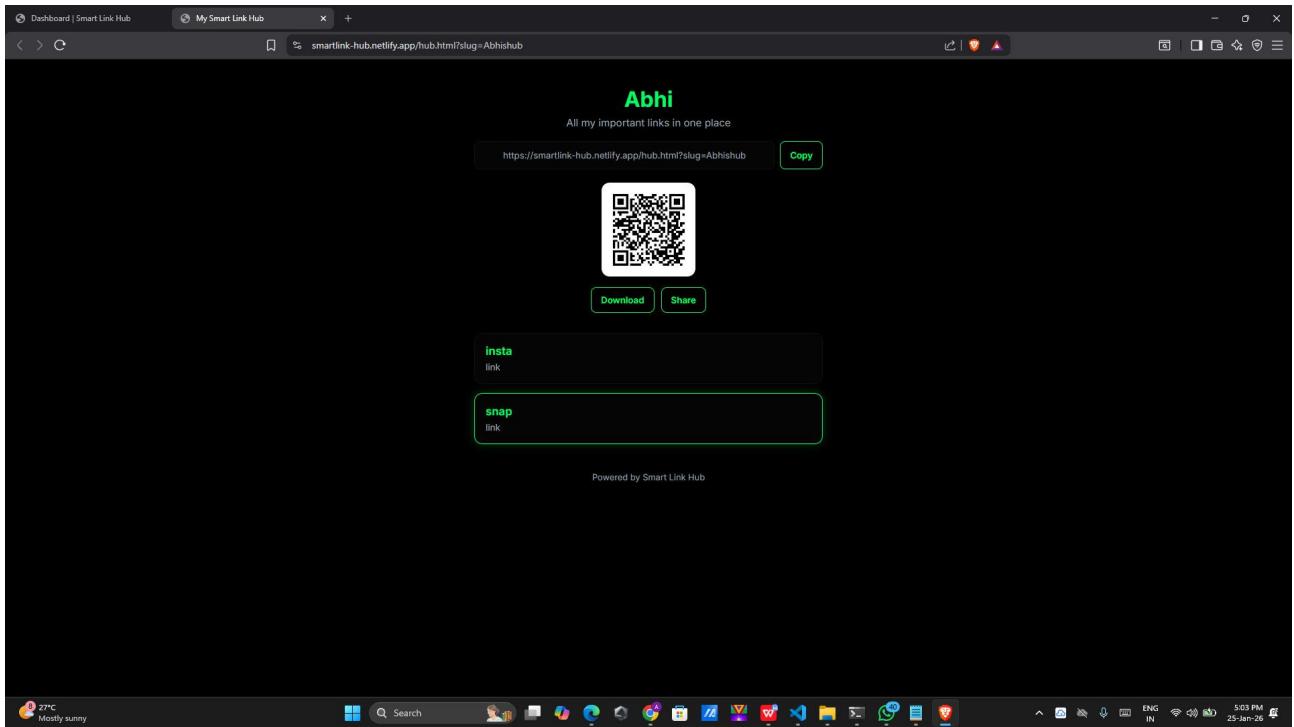
View total hub visits, total link clicks, and the top performing link. Scroll down to see clicks per link.



*Screenshot: Analytics section (dashboard.html)*

## Step 8: Public Hub Page (hub.html?slug=...)

Click \*\*View Public Hub\*\* to open the public hub. Users see your links and QR code, and rules decide which links appear.



Screenshot: Public Hub page (hub.html)

## 4) Notes for Judges (what to check quickly)

- Hub creation works: title/slug/theme saved in Supabase table `hubs`.
- Links work: added under the correct hub in `links` table.
- Rules work: each link can have a rule in `rules` (Time/Device/Country).
- Analytics work: `hub\_visits` increments when public hub is opened; `click\_events` increments on link clicks.
- Public hub loads by slug and shows QR + copy/share/download.

## 5) Local Setup (quick)

Recommended: Use VS Code + Live Server extension.

1. Clone repository from GitHub.

2. Open the `frontend/` folder in VS Code.
3. In `frontend/assets/js/supabase.js`, add your Supabase URL + ANON KEY.
4. Right-click `index.html` → Open with Live Server.

## 6) Deployment (Netlify)

- Push code to GitHub (public repo).
- In Netlify: New site from Git → select repository.
- Set base directory/publish directory to `frontend` (so Netlify serves index.html).
- Deploy. Copy the generated live URL for submission.