

# Troubleshooting GuideSignal's Blank Page Issue

## Overview of the Issue

**GuideSignal** (hosted via GitHub Pages on the custom domain **guide-signal.com**) is currently loading as a blank page for users. Instead of the expected content (job listings, sign-up forms, etc.), visitors see an empty white screen (as confirmed by user-provided screenshots). We need to determine why the site isn't rendering properly and how to fix it. Potential culprits include mis-served HTML (wrong file at the root), missing or failing assets (CSS/JS), client-side script errors (especially related to the `auth-optimized.js` script and Firebase auth), or misconfiguration in the GitHub Pages/GoDaddy domain setup.

## Site Structure and Deployment Configuration

GuideSignal appears to be a static multi-page site. Several HTML pages are present (either as separate pages or sections), including:

- **index.html** – the homepage (“Jobs that reply.” tagline on GuideSignal) <sup>1</sup> .
- **how.html** – “How GuideSignal Works” page <sup>2</sup> .
- **jobs.html** – job listings (“Jobs that reply” page with sample job posts) <sup>3</sup> .
- **post.html** – a “Post a Job” form page <sup>4</sup> .
- **apply.html** – an application page prompting sign-in <sup>5</sup> .
- **auth.html** – the combined Sign In / Sign Up page (contains login form, registration form, etc.) <sup>6</sup> <sup>7</sup> .
- **faq.html**, **privacy.html**, **terms.html** – informational pages (FAQ, Privacy Policy, Terms) <sup>8</sup> <sup>9</sup> .

The site is deployed via **GitHub Pages**. According to search results, the underlying GitHub Pages URL is `https://ideamlabs.github.io/guidesignal-site/` (the repository likely named “guidesignal-site”). A custom domain **guide-signal.com** is configured. Typically, a CNAME file or setting on GitHub Pages ties this domain to the site. The presence of a valid HTTPS lock in the browser and the content being accessible to search engines indicates that the DNS is mostly correct – the domain is pointing to GitHub's servers and a certificate is in place.

### However, potential deployment misconfigurations to double-check:

- **Custom Domain and DNS:** Ensure both the apex domain (guide-signal.com) and the “www” subdomain are properly configured. In GoDaddy DNS, the apex should have A records pointing to GitHub's Pages IPs or an ALIAS, and “www” should have a CNAME to `ideamlabs.github.io` (if you intend to use www). Currently, searching `www.guide-signal.com` returns a result but it might not load properly <sup>10</sup> . If www is not set up, visiting that could fail. Make sure to add a CNAME for www or forward it to the apex. Also, in the GitHub Pages settings, “**Enforce HTTPS**” should be enabled for the custom domain.

- **Root File Served:** GitHub Pages will serve **index.html** by default at the root. Confirm that **index.html** exists in the published branch and contains the intended homepage content. If the site is showing blank, it's possible the wrong file is being served. For example, if **index.html** is missing or empty, or if another file (like **auth.html**) was mistakenly set as the start page (GitHub Pages itself doesn't allow custom default beyond **index.html**, but a redirect script could cause navigation). In our case, the homepage *should* be **index.html** showing the landing content ("Jobs that reply" heading, etc.). We see evidence that **index.html** has at least a header and two links <sup>1</sup>, but earlier content ("Where a Job Finds You." tagline) <sup>11</sup> suggests the homepage content might have been updated or partially removed. Ensure that the **index.html** file is the correct one and contains the full intended markup for the landing page. If content was accidentally placed in a different file or a template, users might be hitting a near-empty page.
- **Base URL or Paths:** Since the project was originally accessible at `ideamlabs.github.io/guidesignal-site/`, any hard-coded references to that path need updating. For example, if the HTML uses a base URL or absolute paths including `"/guidesignal-site/"`, those would break on the custom domain. All asset references and links should either be relative or updated for the new domain. Check the HTML for `<base href>` or any script/style src that still contains `"guidesignal-site"`. A mis-set base href could cause the browser to look for files in the wrong location, leading to 404s and thus a blank or unstyled page.
- **Jekyll vs. Nojekyll:** If any files or directories start with an underscore (e.g., `_assets/`), GitHub Pages' Jekyll might ignore them. In such cases, a blank page could result if critical files aren't published. The solution is adding an empty `.nojekyll` file in the publishing branch to disable Jekyll processing. This may not apply if your repository doesn't use underscore-prefixed folders, but it's worth ensuring `.nojekyll` is present if you had any asset naming that Jekyll could skip.

## HTML Structure and DOM Loading

From the static content we gathered, the HTML markup on each page is mostly straightforward. For instance, the **auth.html** page contains the sign-in form (email, password, "Remember me" checkbox, etc.) and the sign-up form (name, email, password, role selection) in one page <sup>6</sup> <sup>7</sup>. This suggests there may be some JavaScript to toggle between the "Sign In" and "Sign Up" views (perhaps showing one form at a time).

One thing to consider is whether any CSS is hiding content by default. For example, some sites apply `display: none` or an overlay on the body until scripts load (to prevent flicker). If GuideSignal uses such a tactic (e.g., a `.loading` class on body), a script error could leave the page hidden. Check the CSS for any such rules. If you find something like `body.loading { opacity: 0 }` or a full-screen modal that is meant to be removed, that's a clue: the JS needs to remove that class or overlay. A failure in the script would leave the page invisible. On Safari/iOS, a common symptom of this is a completely white page even though the HTML is there.

Also, verify that all sections of the HTML are properly closed and nested. An unclosed `<div>` or a stray `<script>` tag in the wrong place can break rendering. Given that search engines indexed the content, the HTML is likely well-formed, but a quick validation couldn't hurt.

## JavaScript Behavior: auth-optimized.js and Firebase Integration

The `auth-optimized.js` file is a prime suspect in the blank page issue. This script likely contains the client-side logic for user authentication (sign in/up form handling) and possibly other interactive features. It probably initializes Firebase (to use Firebase Auth, and maybe Firestore if profiles are stored) and manages UI state (like showing a user's name when logged in, or toggling the correct form). If this script fails to execute properly, it could prevent parts of the page from rendering or functioning. Key things to examine:

- **Script Loading Failure:** Check if `auth-optimized.js` is actually loading. In the browser dev tools, look at the Network tab for a 404 or other error when fetching this file. A few possibilities:
- **Wrong Path or Filename:** Ensure the `<script src="">` in the HTML points to the correct path. For example, if the file is in the root of the repository, using `src="auth-optimized.js"` (relative path) from pages in subdirectories might fail – it should likely be `src="/auth-optimized.js"` (root-relative) so all pages fetch the same file. If it's in a folder (e.g., `assets/js/auth-optimized.js`), the path in all pages must match. A common mistake is a case-sensitivity mismatch (e.g., the file on disk is `auth-optimized.js` but HTML calls `Auth-Optimized.js`). Such a mistake wouldn't show up locally on Windows/macOS (case-insensitive filesystems) but would 404 on GitHub's servers (case-sensitive). A 404 script leads to a silent failure (except an error in console). The page's HTML would still be there, but any functionality that relies on JS (like revealing content) wouldn't run.
- **MIME Type Errors:** If the script is present but served with the wrong MIME type (e.g., as HTML due to a misconfiguration), browsers might block it. Check the Console for messages about MIME or "unexpected token `<`" which indicates the JS file might actually be an HTML error page (common when a missing file returns a GitHub Pages 404 HTML). If you see that, it confirms the script path or name is wrong – fix the reference so the correct file is found.
- **Load Order:** If `auth-optimized.js` depends on Firebase SDKs, ensure those are included *before* it in the HTML. For example, if Firebase Auth and App scripts aren't imported, any calls to `firebase.initializeApp(...)` or `firebase.auth()` will error out. Look for `<script src="https://www.gstatic.com/firebasejs/...">` tags in your HTML. If missing, you need to include Firebase's libraries or bundle them into your JS. If they are present, they must come before the custom script. A missing Firebase SDK include could cause a `ReferenceError` in `auth-optimized.js`.
- **Firebase Configuration & Authorized Domains:** If the Firebase initialization is present, verify the config object. The `authDomain` in the Firebase config should typically be your Firebase project's domain (like `your-project.firebaseio.com` or `.web.app`), **not** the custom domain. The custom domain needs to be added as an *Authorized Domain* in your Firebase Auth settings (Firebase console). If `guide-signal.com` isn't authorized, Firebase Auth calls (particularly OAuth sign-ins or even email link sign-ins) will fail with an "unauthorized domain" error. This error would appear in the browser console. While this alone might not blank the page, it would break the sign-in flow. So ensure in Firebase console under Authentication > Settings, **guide-signal.com** is listed. Likewise, check that the Firebase API keys in the config are correct for the project.
- **Code Errors in auth-optimized.js:** If the script is loading, any unhandled exceptions early on can halt execution and potentially affect the UI. Some scenarios:

- The script might be executing on pages where certain DOM elements don't exist. For example, if `auth-optimized.js` runs on every page (included in the base template for convenience), it may try to bind event listeners to the login form fields or "Sign In" button. On a page like the home or `how.html`, those elements won't be in the DOM, so `document.getElementById('password')` would return null, and then trying `null.value` or `addEventListener` on it throws an error. Once that error happens, any subsequent code in the script (perhaps code that would remove a loading overlay or populate content) is skipped. This **could leave the page in a semi-initialized, blank state**. To fix this, wrap such code in conditionals (only run if element exists) or include the auth script only on the relevant page (`auth.html`). It's generally better to not load heavy auth scripts on pages that don't need it.
- There may be a logic that immediately redirects or alters the page. For instance, some sites redirect to an auth page if the user is not logged in. If GuideSignal had a plan to redirect `/` to `/auth.html` for new users, a bug in that logic could cause a blank page. Check if any `window.location` or routing code is running on page load. If it's redirecting to a route that doesn't exist or gets stuck in a loop, that would explain the blank screen. (No evidence of a redirect loop yet, but worth examining the script for any location change.)
- Use console logging or breakpoints. If you have access to the site files, temporarily add `console.log` statements or enable debug to see how far the script gets. The browser console error message (if any) will point to a line in `auth-optimized.js`. For example, an error like "Uncaught TypeError: Cannot read property '...' of null at auth-optimized.js:123" would tell exactly what line and give a clue (maybe an element ID). Trace that back in your source code.
- **"Remember me" and Safari Quirks:** The `auth.html` has a "Remember me" checkbox <sup>12</sup>. If the script handles this by using `localStorage` or cookies to persist sessions, note that Safari in Private Mode (and sometimes normal mode) can throw errors when using `localStorage` (in Private mode it's disabled, calling `setItem` will throw a `QuotaExceededError`). If an uncaught exception from `localStorage` access occurs, it could break the script on iPhones. If many users see a blank page on Safari only, this could be the cause. The fix is to wrap `localStorage` usage in `try/catch`. For example:

```
try {
  localStorage.setItem('rememberMe', true);
} catch(e) {
  console.warn('Could not access storage', e);
}
```

This way it fails gracefully instead of killing the script.

- **Firebase Auth State Handling:** If using Firebase Auth, often `onAuthStateChanged` is used to show/hide UI for logged-in users. Make sure that function is implemented correctly. For example, you might want to show the homepage content regardless of auth, but perhaps the code mistakenly only renders content after confirming auth state. If that listener never fires (or errors out), the content might never be unhidden. Ensure that any content reveal isn't gated behind a successful Firebase call. If it is, consider changing the logic to show the public content by default and only adjust UI if a user **is** logged in.

## Potential Routing or MIME Issues

Though the site uses multiple HTML pages (not a single-page app with client-side routing), consider what happens when navigation occurs: - Clicking “Get Started Free” on the home goes to **auth.html** (the sign-up page) <sup>13</sup>. - The “Apply Now” buttons on job listings likely lead to **apply.html** which then prompts sign-in <sup>5</sup>. - The navigation links (Home, Fast-Reply Jobs, Apply, Post Job, FAQ) go to their respective pages <sup>14</sup>. These should all load fresh pages from the server (since it's static pages, not an SPA). If some of those links were coded incorrectly (e.g., pointing to a wrong path or missing .html extension), users might hit a 404. A 404 page on GitHub Pages by default is a minimal page that might appear blank. Check that each link's `href` exactly matches the page file name. For example, ensure the “FAQ” link goes to `faq.html` (we saw FAQ link in how.html's nav) <sup>14</sup>. If it were missing the `.html`, it could be looking for a directory. - **MIME Type for CSS:** Similar to JS, if a CSS file is linked and not found or not served, the page might be unstyled (though not blank). If text was white on white background due to missing CSS, that could appear blank. But the content we saw (like “Jobs that reply.”) is plain text; if no CSS loaded, it would default to black text on white background, which *should* be visible. So CSS likely isn't the primary issue unless a style actively hides content.

## Identifying the Root Cause

From the above, the most likely cause of the blank page is an unhandled error in **auth-optimized.js** that prevents the page from finishing loading or displaying. The blank page symptom strongly suggests that the HTML did load (since no 404 error is shown and the browser address stayed on the correct URL), but something prevented visible rendering. Summarizing likely root causes to investigate:

- **Auth script crashing due to missing element or API** – e.g., running on the wrong page or missing Firebase includes, causing a JS exception that halts further execution. This would explain why the static content doesn't show (if it was intentionally hidden until script runs).
- **Auth script not loaded at all** (404) – leading to, perhaps, an overlay or some content not being injected. Check the Network panel for a 404 on `auth-optimized.js`. If found, correct the path or filename in the HTML. Ensure the file is in the repository (and not being ignored by Jekyll).
- **Firebase domain or config issue** – causing runtime errors (check console for Firebase errors). Even if not immediately blanking the page, these errors need resolution: add **guide-signal.com** to authorized domains in Firebase, and double-check the Firebase config snippet in the script (especially `authDomain` and API keys).
- **Body or content hidden by default** – and the script that was supposed to unveil it didn't run. Look for something like `<body class="js-disabled">` or a covering `<div id="spinner">` that never gets removed. Removing or modifying that in dev tools can quickly confirm if content is underneath.
- **DNS or GitHub Pages config** – less likely given the symptoms (partial content was indexed by search engines, meaning at least some bots saw it). But if any users get a blank due to caching, try accessing via another network or flush DNS. Ensure no mixed-content issues (all resources load over HTTPS).

## Guidance to Fix the Site

Based on the analysis, here are concrete steps to resolve the malfunction:

1. **Check Console for Errors:** Open the browser developer console on the blank page. Look for red error messages. These will often pinpoint the file and line of a script that failed. For example, an error referencing `auth-optimized.js:line 50` or a Firebase error code will guide you. Address whatever is reported:
2. If it says a function/variable is undefined, ensure the script order is correct (include dependencies first).
3. If it says a resource failed to load (404 or blocked), fix that path or upload the missing file.
4. If it's a runtime error (TypeError, etc.), jump to the script code at that line and inspect the logic.
5. **Fix Script Inclusion and Scope:** If `auth-optimized.js` is currently included on every page, consider limiting it to the pages that need it (auth and maybe apply). For now, you can add defensive coding:

```
const signInBtn = document.getElementById('signInButtonId');
if (signInBtn) {
  // attach event listener
}
```

This way, on pages without that element, the code inside doesn't run. Also ensure that the script is referenced by the correct filename. (Double-check the case sensitivity and that your repository's published files have the exact same name.)

6. **Verify Firebase Initialization:** In `auth-optimized.js` (or wherever you initialize Firebase), confirm that you call `firebase.initializeApp({...})` with the correct config **before** calling any auth or Firestore methods. If you split the Firebase config into a separate file, make sure it's being loaded. Also, add `console.log("Firebase init")` after initialization to confirm it runs. If using Firebase Auth, also add a listener for errors:

```
firebase.auth().onAuthStateChanged(user => {
  console.log("Auth state changed, user:", user);
}, error => {
  console.error("Auth state error:", error);
});
```

This might surface hidden issues.

7. **Add guide-signal.com to Firebase Auth Domains:** Go to Firebase console > Auth > Settings and add the custom domain. This prevents Firebase from blocking sign-in flows on that domain. Without this, sign-in attempts can fail silently or with console errors.

8. **Remove Temporary Hiding Styles:** If you find CSS that hides the body on load, disable it to see if content appears. For instance, if the HTML has `<body class="loading">`, remove that class in dev tools and see if the page content pops in. If it does, the issue is indeed that the script never removed the class. You can then focus on why the script didn't. As a quick fix, you could remove the hiding behavior until the underlying script issue is fixed, so at least users see the content. For example, delete or comment out the CSS rule that hides content on load.
9. **Ensure Index.html Content:** Open the repository and verify that **index.html** contains all the sections you expect on the homepage (hero text, any overview or "How it works" teaser, etc.). The search-indexed content "Where a Job Finds You. The fastest way to work where you belong..." <sup>11</sup> suggests the homepage had more content. If that was removed intentionally and moved elsewhere, the homepage might currently be very sparse (just a header and links) <sup>1</sup>. That by itself wouldn't be blank, but it could be easy to miss on a mobile screen (e.g., if a fixed header overlaps it). Make sure the homepage isn't essentially empty. Re-add key sections or at least a welcome message so users don't land on a bare page.
10. **Test on Multiple Devices:** After changes, test on desktop and mobile (especially Safari). Use an iPhone or Safari's emulator to verify that no Safari-specific errors occur (watch out for storage access and any ES6 features that might need polyfills).
11. **Monitor Form Integration:** The FAQ indicates that forms use Formspree and Stripe for payments <sup>15</sup>. These shouldn't cause the initial load to fail since they are likely simple form submissions or scripts loaded on demand. But ensure any Stripe script is loaded correctly over HTTPS and not blocking the UI.
12. **Redeploy and Clear Cache:** Push the fixes to GitHub Pages. Sometimes after a domain change or Pages update, older cached DNS or content might linger. On your devices, clear the cache or do a hard reload. Ensure you're getting the latest files. You can append a version query to the script URL (e.g., `auth-optimized.js?v=2`) to bust caches if needed during debugging.

By following the above steps, you should identify the exact cause of the blank page. In many cases, it boils down to a JavaScript error that halts execution before the page finishes rendering. Once that error is fixed (whether it's an asset path, a null reference, or a config issue), the site should load its content normally.

**In summary,** focus on the **auth-optimized.js** file and related Firebase setup, as that's the most likely source of the problem. Make sure the correct **index.html** is served, all assets load without errors, and that no code is inadvertently hiding the UI. With these corrections, GuideSignal's pages should display as intended, allowing users to see the job listings and use the sign-up/sign-in functionality.

1 11 **GuideSignal - Jobs that reply**

<https://guide-signal.com/>

2 14 **How GuideSignal Works - 3 Simple Steps**

<https://guide-signal.com/how.html>

3 **Jobs - GuideSignal**

<https://guide-signal.com/jobs.html>

4 **Post a Job - GuideSignal**

<https://guide-signal.com/post.html>

5 **Apply for Fast-Reply Job Matches - GuideSignal**

<https://guide-signal.com/apply.html>

6 7 12 13 **Sign In - GuideSignal**

<https://guide-signal.com/auth.html>

8 15 **Frequently Asked Questions - GuideSignal**

<https://guide-signal.com/faq.html>

9 **Terms of Service - GuideSignal**

<https://guide-signal.com/terms.html>

10 **GuideSignal - Jobs that reply**

<https://www.guide-signal.com/>