

FindMyNiche Frontend — findmyniche-site

Production-grade documentation for the **FindMyNiche frontend**, including Firebase Hosting, Firebase Auth, Firebase App Check, Cloud Run API integration, deployment pipelines, and debugging workflows.

1. Project Overview

This project is the **public web frontend** for FindMyNiche. It:

- Serves a Vite-built static site via **Firebase Hosting**
- Uses **Firebase Authentication (Anonymous)** to identify users
- Uses **Firebase App Check (reCAPTCHA v3)** to protect backend API access
- Calls a protected **Cloud Run backend API** (``)
- Displays backend status and authenticated user metadata

Live Domains

Domain	Purpose
https://findmynicheapp.com	Primary production domain
https://www.findmynicheapp.com	Redirect / alias
https://findmyniche-483917.web.app	Firebase Hosting default domain

2. Architecture Flow



Security Model

Layer	Purpose
Firebase Hosting	Secure static content delivery
Firebase Auth	Identifies user (anonymous or signed-in)
App Check	Prevents non-browser / bot / curl abuse
Cloud Run IAM	Restricts backend verification permissions

3. Tech Stack

Component	Technology
Framework	Vite + Vanilla JS / React-compatible
Hosting	Firebase Hosting
Auth	Firebase Auth (Anonymous)
App Protection	Firebase App Check (reCAPTCHA v3)
API	Google Cloud Run (FastAPI backend)
DNS	Custom domain mapped via Firebase

4. Repository Structure

```
findmyniche-site/
├─ public/
│   ├── logo.png
│   └─ favicon.ico
├─ src/
│   ├── firebase.js      # Firebase SDK + Auth + App Check
│   ├── api.js           # API client wrapper
│   └─ main.js           # UI logic
├─ index.html
├─ vite.config.js
├─ firebase.json
├─ .firebaserc
└─ package.json
```

5. Firebase Project Setup

Project

Project ID:

```
findmyniche-483917
```

Enabled Services

Enable in Firebase Console:

- Authentication
- App Check
- Hosting

6. Firebase Hosting Setup

Firebase CLI Install

```
npm install -g firebase-tools
```

Login

```
firebase login
```

Initialize Hosting

```
firebase init hosting
```

Recommended answers:

Prompt	Answer
Public directory	<code>dist</code>
SPA rewrite	Yes
GitHub Actions	Optional

firebase.json

```
{
  "hosting": {
    "public": "dist",
    "ignore": [
      "firebase.json",
      "**/*.*",
      "**/node_modules/**"
    ],
    "rewrites": [
      {
        "source": "**",
        "destination": "/index.html"
      }
    ]
  }
}
```

7. Firebase Authentication Setup

Enable Anonymous Auth

Firebase Console → Authentication → Sign-in Methods

Enable:

Anonymous

Why Anonymous Auth?

This:

- Assigns each visitor a unique UID
- Allows API personalization
- Prevents unauthenticated API abuse
- Avoids forcing sign-up

8. Firebase App Check Setup

Purpose

Prevents:

- cURL abuse
- Postman attacks
- Bots
- Unauthorized scripts

Only **real browsers running your frontend** can call the backend.

Register Web App

Firebase Console → App Check → Apps → Register Web App

Select:

reCAPTCHA v3

Save:

- **Site Key** → Used in frontend
 - (Secret key is stored in Firebase, not in frontend)
-

9. Frontend Firebase Code

src/firebase.js

```
import { initializeApp } from "firebase/app"
import { getAuth, signInAnonymously } from "firebase/auth"
import { initializeAppCheck, RecaptchaV3Provider, getToken } from "firebase/app-check"

const firebaseConfig = {
  apiKey: "AIzaSyDjXigFVx2q6o_HA160riKNtTxM00-9bY",
  authDomain: "findmyniche-483917.firebaseio.com",
  projectId: "findmyniche-483917",
  storageBucket: "findmyniche-483917.firebaseio.com",
  messagingSenderId: "747869728784",
  appId: "1:747869728784:web:8019548f37558dfabf9a95",
  measurementId: "G-G38N2XF8YC"
}
```

```

const app = initializeApp(firebaseConfig)

export const auth = getAuth(app)

// App Check
const appCheck = initializeAppCheck(app, {
  provider: new ReCaptchaV3Provider("YOUR_SITE_KEY_HERE"),
  isTokenAutoRefreshEnabled: true
})

export async function getAppCheckToken() {
  const { token } = await getToken(appCheck, false)
  return token
}

export async function getIdToken() {
  const userCred = await signInAnonymously(auth)
  return await userCred.user.getIdToken()
}

```

10. API Client Wrapper

src/api.js

```

import { getIdToken, getAppCheckToken } from "../firebase"

const API_BASE = "https://api.findmynicheapp.com"

export async function apiFetch(path) {
  const idToken = await getIdToken()
  const appCheckToken = await getAppCheckToken()

  const res = await fetch(`${API_BASE}${path}`, {
    headers: {
      "Authorization": `Bearer ${idToken}`,
      "X-Firebase-AppCheck": appCheckToken
    }
  })

  if (!res.ok) {
    throw new Error("Request failed")
  }

  return res.json()
}

```

11. UI Integration

Example API Call

```
import { apiFetch } from "../api"

async function loadMe() {
  try {
    const me = await apiFetch("/v1/me")
    document.getElementById("api-status").innerText = "ONLINE"
    document.getElementById("user-data").innerText = JSON.stringify(me,
null, 2)
  } catch {
    document.getElementById("api-status").innerText = "OFFLINE"
  }
}

loadMe()
```

12. Local Development

Install

```
npm install
```

Run Dev Server

```
npm run dev
```

App runs at:

```
http://localhost:5173
```

NOTE: App Check does NOT work locally unless you use debug tokens

13. Build for Production

```
npm run build
```

Output:

```
dist/
```

14. Deploy to Firebase Hosting

```
firebase deploy --only hosting
```

15. Custom Domain Setup

Firebase Console → Hosting → Add Custom Domain

Add:

```
findmynicheapp.com  
www.findmynicheapp.com
```

Update DNS:

Type	Name	Value
A	@	Firebase IP
CNAME	www	Firebase domain

16. CORS Requirements (Backend Dependency)

Your frontend WILL FAIL if backend CORS does not include:

```
allow_origins=[  
  "https://findmyniche-483917.web.app",  
  "https://findmynicheapp.com",  
]
```



```
"https://www.findmynicheapp.com",  
]
```

17. Common Errors & Fixes

API Status OFFLINE

Cause:

- Missing Auth token
- Missing App Check token
- CORS blocked

Fix:

- Verify headers sent:

```
Authorization: Bearer <token>  
X-Firebase-AppCheck: <token>
```

CORS Error in DevTools

Cause: Backend not allowing domain

Fix: Update backend CORS allow list

App Check Invalid Token

Cause:

- Wrong site key
- App not registered

Fix: Firebase Console → App Check → Verify site key

Works in Incognito Only

Cause: Browser cache holding old JS bundle

Fix: Hard reload:

Cmd + Shift + R

18. Production Security Model

Threat	Protection
Bot scraping	App Check
API abuse	Auth + App Check
Token replay	Short-lived tokens
Domain hijack	Firebase Hosting SSL

19. Deployment Flow

```
Developer
→ npm run build
→ firebase deploy
→ Firebase CDN
→ Users
```

20. System Relationship

Component	Repo
Frontend	findmyniche-site
Backend	fmn-public-api
Hosting	Firebase
API	Cloud Run
Security	Firebase Auth + App Check

21. Recommended Production Improvements

- Enable Firebase App Check enforcement
- Add Cloud Armor rate limiting
- Add Firebase Analytics
- Add Error Monitoring (Sentry)

- Add CI/CD GitHub Actions
-

22. Emergency Debug Commands

Check Hosting

```
firebase hosting:sites:list
```

Check API Reachability

```
curl https://api.findmynicheapp.com/health
```

Verify Tokens

DevTools → Network → Headers

23. Maintainer Notes

This frontend is:

- Production-hardened
- Bot-protected
- Secure-by-default
- Cloud-native

Any developer can safely extend this system by:

- Adding real Firebase login
 - Adding dashboards
 - Enabling paid features
-

24. Ownership

FindMyNiche © 2026

Maintained by:

Satta