Project: ImagineHumans Academy

CI/CD Build & Firebase Integration – Issue Resolution

1. Build Status and Resolution Summary

Current Status

The application build process is now successful and generates a complete, optimised production output (the `.next` directory). The CI/CD pipeline is functional, and the Docker image can be deployed reliably.

Issue Encountered

The local production build (`npm run build`) was initially failing during the Docker image build and CI execution. Multiple issues were identified related to Firebase authentication, dependency versions, and environment configuration.

Root Causes

A. Firebase Service Account File Dependency

The application attempted to import a local Firebase service-account JSON file in:

```
src/lib/firebase-admin.ts
```

This caused build failures in CI/CD because:

- Local files are not available in the CI environment by default.
- Uploading service account JSON files to GitHub is blocked by secret-scanning and is not secure.

B. Firebase Functions Version Conflict

After installing the latest `firebase-functions`, the build failed with:

```
TypeError: Property 'region' does not exist
```

This occurred because:

- The codebase uses Firebase Functions v1 syntax (`functions.region()`).
- The installed package was a v2-compatible version, which no longer supports this syntax.

C. Firebase Permission Warnings During Build

During static page generation, the build logs showed warnings such as:

```
FirebaseError: Missing or insufficient permissions
```

This occurred when fetching Firestore data (e.g., courses, user counts).

2. Solutions To Be Implemented

A. Correct & Secure Implementation

The final and correct fix is:

Remove all imports of `service-account.json` from:

`src/lib/firebase-admin.ts`

- Firebase Admin authentication must rely on environment variables, not local files.
- The service account key is injected securely via environment variables (e.g. base64-encoded), ensuring:
  - No secrets are committed to the repository
  - CI/CD builds work consistently
  - Production deployments remain secure

B. Firebase Functions Compatibility Fix

- Downgraded `firebase-functions` to a v1-compatible version (v4.8.1).
- This resolved the `.region()` API compatibility issue and eliminated build-time errors.

C. Firebase Permission Warnings (No Action Required)

The Firebase permission warnings observed during build are expected and non-blocking.

Reason:

- Static generation runs without Firebase Admin credentials.
- Firestore access is restricted during build time.
- The application functions correctly at runtime when proper credentials are available.

These warnings do not affect deployment or runtime behaviour.

3. Future Risks and Mitigation

A. Dependency Instability (High Risk)

Risk:
The build currently depends on:

- `firebase-functions@4.8.1`
- `firebase-admin@13.6.0`
- Use of `--legacy-peer-deps`

Future dependency upgrades or clean installs may:

- Reintroduce version conflicts
- Break builds due to v1/v2 syntax mismatch

Mitigation

- Explicitly lock the `firebase-functions` version in `package.json`.
- Plan a future refactor of `./functions/src` to Firebase Functions v2 syntax for long-term stability.

B. Environment Variable Mismatch (Deployment Risk)

Risk:
Even though the build succeeds, deployments will fail if the required environment variables are missing or misconfigured during build time.

Critical variables include:

- `STRIPE_SECRET_KEY`
- `STRIPE_WEBHOOK_SECRET`
- `NEXT_PUBLIC_FIREBASE_API_KEY`
- Firebase service account credentials
- Other server-only keys

4. Summary

- The CI/CD pipeline is now stable and functional.
- The correct solution removes all file-based credential dependencies.
- `src/lib/firebase-admin.ts` should not import a service account file going forward.