

Docker Build Fixes Summary

Issue Resolution

The dependabot Docker build was failing with the following error:

```
error TypeError: Cannot read properties of undefined (reading 'manifest')
Error: Command failed with exit code 1: yarn add prisma@6.13.0 -D --silent
```

Root Causes Identified

1. **Package Manager Conflicts:** Mixed yarn and npm lock files causing workspace conflicts
2. **Legacy Environment Variable Syntax:** Docker warnings about ENV format
3. **Prisma Configuration:** Incorrect binary targets and output paths for Alpine Linux
4. **Workspace Version Issues:** Missing version in workspace package.json

Fixes Applied

1. Dockerfile Improvements

Before:

```
# Used yarn with potential conflicts
COPY package.json yarn.lock* package-lock.json* pnpm-lock.yaml* ./
RUN yarn --frozen-lockfile

# Legacy ENV format
ENV NEXT_TELEMETRY_DISABLED 1
ENV NODE_ENV production
```

After:

```
# Simplified to use npm only
COPY app/package.json ./app/
RUN rm -f app/yarn.lock app/package-lock.json
RUN npm install --production=false

# Fixed ENV format
ENV NEXT_TELEMETRY_DISABLED=1
ENV NODE_ENV=production
```

2. Prisma Schema Updates

Before:

```
generator client {
  provider = "prisma-client-js"
  binaryTargets = ["native", "linux-musl-arm64-openssl-3.0.x"]
  output = "/home/ubuntu/ai_rfp_risk_scanner/app/node_modules/.prisma/client"
}
```

After:

```
generator client {
  provider = "prisma-client-js"
  binaryTargets = ["native", "linux-musl-openssl-3.0.x", "linux-musl-arm64-openssl-3.0.x"]
}
```

3. Package Configuration

Before:

```
{
  "name": "app",
  "private": true,
  ...
}
```

After:

```
{
  "name": "app",
  "version": "1.0.0",
  "private": true,
  ...
}
```

4. Lock File Management

- Removed conflicting `package-lock.json` files
- Added package-lock exclusion to `.dockerignore`
- Simplified to use npm in Docker for better reliability

Build Process Changes

Old Process (Problematic)

1. Copy all lock files (yarn.lock, package-lock.json)
2. Try to resolve workspace dependencies with yarn
3. Fail on manifest conflicts
4. Generate Prisma with wrong binary targets






New Process (Fixed)

1. Copy only package.json files
2. Remove any conflicting lock files
3. Install dependencies with npm (more reliable in Docker)

4. Generate Prisma with correct Alpine Linux binary targets
5. Build with proper environment variables

Verification Steps

The fixes address:

-  Package manager conflicts
-  Legacy ENV format warnings
-  Prisma binary target compatibility
-  Workspace version warnings
-  Lock file conflicts

Testing Recommendations

1. Test Docker build locally:

```
bash
```

```
docker build -t ai-rfp-scanner .
```

2. Verify Prisma generation:

```
bash
```

```
docker run -it ai-rfp-scanner npx prisma generate
```

3. Check environment variables:

```
bash
```

```
docker run ai-rfp-scanner env | grep -E "(NODE_ENV|NEXT_)"
```

Expected Results

The GitHub Actions dependabot build should now:

- Successfully install dependencies without conflicts
- Generate Prisma client correctly for Alpine Linux
- Build the Next.js application without warnings
- Create a working Docker image ready for deployment

These fixes maintain all existing functionality while resolving the Docker build pipeline issues.