Security Guidelines for VerifyLens Landing Page

Environment Variables

Current Setup

- All sensitive credentials are stored in `.env` file
- The `.env` file is properly excluded from version control via `.gitignore`
- A template file `.env.example` is provided for reference

Required Environment Variables

```
# Database URL (not currently used in lander, but available)
DATABASE_URL="postgresql://user:password@host:port/database"

# Stripe Configuration
STRIPE_SECRET_KEY="sk_test_..."
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY="pk_test_..."

# Main App URL (roblox-tool deployment)
NEXT_PUBLIC_APP_URL="https://www.verifylens.com"

# Lander URL (this site)
NEXT_PUBLIC_SITE_URL="https://site.verifylens.com"
```

Security Incident History

October 31, 2025 - GitHub Token Exposure Resolution

Issue: A GitHub Personal Access Token was accidentally committed in an earlier commit (c7700487d-d3dc9ae250a914f7f116dba4388d6d4) in the `.env` file.

Resolution Steps Taken:
1. ✅ Removed `.env` file from version control (commit fb16d07)
2. ✅ Confirmed `.env` is properly listed in `.gitignore`
3. ✅ Removed GitHub token from local `.env` file
4. ✅ GitHub secret scanning alert was acknowledged and allowed for push

Action Required:
⚠️ **IMPORTANT**: The GitHub token `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` that was exposed in the git history should be rotated immediately.

To rotate the token:
1. Go to GitHub Settings > Developer settings > Personal access tokens
2. Delete the exposed token
3. Generate a new token with the same permissions
4. Update any services using the old token

Best Practices

For Developers

1. **Never commit `.env` files** - Always use `.env.example` as a template
2. **Use placeholder values** in `.env.example` - Never put real credentials in example files
3. **Rotate credentials** immediately if they are exposed
4. **Use environment-specific files**:
   - `.env.local` for local development
   - `.env.production` for production (never commit this)
5. **Scan before committing**: Run `git diff --cached` to review changes before committing

For Deployment

1. **Use platform-specific secret management**:
   - Vercel: Use Environment Variables in project settings
   - AWS: Use AWS Secrets Manager or Parameter Store
   - Other platforms: Use their respective secret management tools
2. **Never expose secrets in client-side code**
3. **Use different credentials for different environments** (dev, staging, production)

Checking for Secrets

Run this command to check for potential secrets in your code:

```
# Check for common secret patterns
grep -r "ghp_\|ghu_\|github_pat_\|sk_test_\|sk_live_" \
  --include="*.tsx" --include="*.ts" --include="*.js" \
  --exclude-dir=node_modules --exclude-dir=.next
```

Git History Cleanup (If Needed)

If secrets are committed to git history, you have two options:

Option 1: GitHub Secret Scanning (Recommended for Public Repos)

GitHub will detect secrets and provide options to:
- Allow the secret (if it's already rotated)
- Remove the secret from history

Option 2: Manual History Rewrite (For Private Repos)

⚠️ **Warning**: This requires force-push and affects all collaborators

```
# Use git-filter-repo (recommended) or BFG Repo-Cleaner
git filter-repo --path .env --invert-paths
git push origin --force --all
```

Contact

For security concerns or to report vulnerabilities, contact the development team immediately.

Last Updated: October 31, 2025