Diyar Property Management System - GitHub Push Instructions

Overview

This document provides instructions to push your Diyar Property Management System frontend code to your GitHub repository.

Repository Information

- **Repository URL**: https://github.com/chiragushah/Diyar-Property-Management
- **Repository Status**: ✅ Verified - Public repository with proper permissions

Method 1: Manual Push from Your Local Machine

Prerequisites

1. Git installed on your machine
2. Your personal access token: `github_pat_11BYC42EA0io3BMwQbFsrG_drRRjTvXN1QbZMvJA0t4L8hXOKwyZJIEmzrrNaWXnE`

Steps

1. **Download/Clone the current workspace** (if not already on your machine)
2. **Navigate to the project directory** `bash cd /path/to/diyar-property-management`

3. **Initialize Git (if not already done)**
   `bash git init git config user.name "chiragushah" git config user.email "your-email@example.com"`

4. **Add the remote repository**
   `bash git remote add origin https://github.com/chiragushah/Diyar-Property-Management.git`

5. **Add all files to Git**
   `bash git add .`

6. **Commit the files**
   `bash git commit -m "Initial commit: Diyar Property Management System"`

7. **Create main branch and push**
   `bash git branch -M main git push -u origin main`

When prompted for authentication:
- Username: `chiragushah`
- Password: `github_pat_11BYC42EA0io3BMwQbFsrG_drRRjTvXN1QbZMvJA0t4L8hXOKwyZJIEmzrrNaWXnE8DCC`

Method 2: Using GitHub CLI

If you have GitHub CLI installed:

```
gh auth login --with-token < echo
"github_pat_11BYC42EA0io3BMwQbFsrG_drRRjTvXN1QbZMvJA0t4L8hXOKwyZJIEmzrrNaWXnE8D
gh repo clone chiragushah/Diyar-Property-Management
# Copy your files to the cloned directory
cd Diyar-Property-Management
git add .
git commit -m "Add Diyar Property Management System"
git push
```

Method 3: Using Git with Token in URL

```
git remote add origin https://
chiragushah:github_pat_11BYC42EA0io3BMwQbFsrG_drRRjTvXN1QbZMvJA0t4L8hXOKwyZJIEm
chiragushah/Diyar-Property-Management.git
git push -u origin main
```

Project Structure to Upload

The following key files and directories should be pushed to GitHub:

```
diyar-property-management/
├── README.md                    # Project documentation
├── package.json                 # Dependencies and scripts
├── package-lock.json           # Lock file
├── tsconfig.json               # TypeScript configuration
├── vite.config.ts              # Vite build configuration
├── tailwind.config.js          # Tailwind CSS configuration
├── postcss.config.js           # PostCSS configuration
├── index.html                  # Main HTML file
├── src/                        # Source code
│   ├── App.tsx                 # Main App component
│   ├── main.tsx               # Entry point
│   ├── index.css              # Global styles
│   ├── components/            # React components
│   ├── pages/                 # Page components
│   ├── api/                   # API functions
│   ├── lib/                   # Utilities and configurations
│   └── stores/                # State management
├── public/                     # Public assets
├── dist/                       # Build output (optional)
└── supabase/                   # Supabase configurations
    ├── functions/              # Edge functions
    └── migrations/             # Database migrations
```

Files to Exclude (.gitignore)

Create a `.gitignore` file with the following content:

```
# Dependencies
node_modules/
.pnp
.pnp.js

# Production builds
/dist
/build

# Environment variables
.env
.env.local
.env.development.local
.env.test.local
.env.production.local

# Logs
npm-debug.log*
yarn-debug.log*
yarn-error.log*
pnpm-debug.log*
lerna-debug.log*

# Runtime data
pids
*.pid
*.seed
*.pid.lock

# Coverage directory used by tools like istanbul
coverage/
*.lcov

# nyc test coverage
.nyc_output
```

```
# Grunt intermediate storage
.grunt

# Bower dependency directory
bower_components

# node-waf configuration
.lock-wscript

# Compiled binary addons
build/Release

# Dependency directories
jspm_packages/

# TypeScript cache
*.tsbuildinfo

# Optional npm cache directory
.npm

# Optional eslint cache
.eslintcache

# Optional stylelint cache
.stylelintcache

# Microbundle cache
.rpt2_cache/
.rts2_cache_cjs/
.rts2_cache_es/
.rts2_cache_umd/

# Optional REPL history
```

```
.node_repl_history

# Output of 'npm pack'
*.tgz

# Yarn Integrity file
.yarn-integrity

# parcel-bundler cache
.cache
.parcel-cache

# Next.js build output
.next

# Nuxt.js build / generate output
.nuxt
dist

# Storybook build outputs
.out
.storybook-out

# Temporary folders
tmp/
temp/

# IDE
.vscode/
.idea/
*.swp
*.swo

# OS
.DS_Store
```

```
.DS_Store?
._*
.Spotlight-V100
.Trashes
ehthumbs.db
Thumbs.db

# Local environment files
.env*.local
```

Troubleshooting

If you encounter permission issues:

1. **Check Token Permissions**: Ensure your personal access token has `repo` scope
2. **Verify Repository Access**: Make sure you have write access to the repository
3. **Try SSH**: Set up SSH keys as an alternative to HTTPS

Next Steps After Successful Push

1. **Verify the upload**: Check https://github.com/chiragushah/Diyar-Property-Management
2. **Set up GitHub Pages** (if desired) for automatic deployment
3. **Configure branch protection rules** (optional)
4. **Add collaborators** (if needed)

Support

If you continue to experience issues, you can:
1. Try pushing from a different network
2. Contact GitHub support
3. Use GitHub Desktop application as an alternative

Note: Your personal access token should be kept secure and not shared publicly.