

Below is the project folder structure for the `real-market` project, based on the setup using Next.js (App Router, no `src` folder, no Turbopack), Tailwind CSS v3, Appwrite SDK, AWS S3, and the specified domains (`real.harshafix.diy`, `blog.real.harshafix.diy`). The structure includes all files provided in the previous responses, organized to reflect the project's requirements and ensure clarity for development on VS Code with Fedora 41.

Project Folder Structure

text

... ⚡

```
real-market/
├── app/
│   ├── api/
│   │   ├── auth/
│   │   │   └── [...nextauth]/
│   │   │       └── route.ts
│   │   ├── blog-posts/
│   │   │   └── route.ts
│   │   ├── pages/
│   │   │   └── route.ts
│   │   ├── properties/
│   │   │   └── route.ts
│   │   └── upload/
│   │       └── route.ts
│   ├── blog/
│   │   ├── [slug]/
│   │   │   └── page.tsx
│   │   └── page.tsx
│   ├── properties/
│   │   ├── [id]/
│   │   │   └── page.tsx
│   │   └── [slug]/
│   │       └── page.tsx
│   ├── admin/
│   │   └── page.tsx
│   ├── globals.css
│   ├── layout.tsx
│   └── page.tsx
├── scripts/
│   └── init-appwrite.ts
├── .eslintrc.json
├── .gitignore
├── next.config.js
├── package.json
├── postcss.config.js
├── tailwind.config.js
├── tsconfig.json
├── README.md
└── .env.local
```

File Descriptions

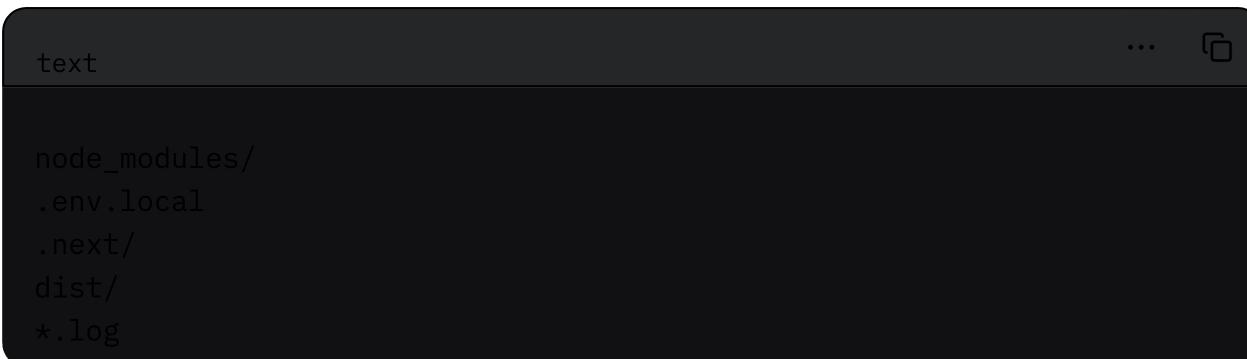
- **app/**: Contains Next.js App Router pages, API routes, and global styles.
 - **api/**: API routes for authentication, data operations, and file uploads.
 - auth/[...nextauth]/route.ts : Configures NextAuth for admin authentication with Appwrite.
 - blog-posts/route.ts : API to create blog posts in Appwrite.
 - pages/route.ts : API to create dynamic pages in Appwrite.
 - properties/route.ts : API to create property listings in Appwrite.
 - upload/route.ts : API to upload media to Appwrite's S3 bucket.
 - **blog/**: Blog-related pages.
 - [slug]/page.tsx : Dynamic route for individual blog posts.
 - page.tsx : Blog homepage listing all posts.
 - **properties/**: Property listing pages.
 - [id]/page.tsx : Dynamic route for individual property details.
 - **[slug]/page.tsx**: Dynamic route for custom pages (e.g., About, Contact).
 - **admin/page.tsx**: Admin dashboard for managing properties, pages, blog posts, and media.
 - globals.css : Global styles with Tailwind CSS v3 directives.
 - layout.tsx : Root layout with navigation and NextAuth session provider.
 - page.tsx : Homepage displaying property listings.
- **scripts/**:
- init-appwrite.ts : TypeScript script to initialize Appwrite (database, collections, storage bucket, authentication) using the SDK.

- **Root Files:**

- `.eslintrc.json` : ESLint configuration for TypeScript and Next.js.
- `.gitignore` : Ignores `node_modules`, `.env.local`, and build artifacts.
- `next.config.js` : Next.js configuration with blog subdomain rewrite.
- `package.json` : Project dependencies (Tailwind CSS v3, Appwrite, AWS SDK, etc.) and scripts.
- `postcss.config.js` : PostCSS configuration for Tailwind CSS v3 and Autoprefixer.
- `tailwind.config.js` : Tailwind CSS v3 configuration for App Router.
- `tsconfig.json` : TypeScript configuration for Next.js.
- `README.md` : Project documentation (optional, can be created manually).
- `.env.local` : Environment variables for Appwrite, AWS S3, and NextAuth (not committed).

Notes

- **Missing Files:**

- `.gitignore` and `README.md` are not explicitly provided but should be created:
 - `.gitignore` :

```
text ... ⌂

node_modules/
.env.local
.next/
dist/
*.log
```
- `README.md` : Add project setup instructions (e.g., `npm install`, `npx ts-node scripts/init-appwrite.ts`, `npm run dev`).
- `.env.local` is referenced but not included in the structure (as it's sensitive). Create it with the variables from the previous response.

- **Tailwind CSS v3:**

- Configured in `tailwind.config.js` and `postcss.config.js` to work with `app/**/*.{js,ts,jsx,tsx}`.
- Uses `tailwindcss@3.4.1` to avoid v4 issues.

- **Appwrite SDK:**

- The `scripts/init-appwrite.ts` script sets up the Appwrite backend programmatically.
- Replace `DATABASE_ID` in all files with the ID logged by the script after running `npx ts-node scripts/init-appwrite.ts`.

- **Directory Organization:**

- No `src` folder, as per your requirement.
- All pages and API routes are under `app/`, following Next.js App Router conventions.
- `scripts/` is used for the Appwrite initialization script to keep it separate from the main app.

- **Running the Project:**

- Install dependencies:

```
bash
```

```
npm install
```

- Initialize Appwrite:

```
bash
```

```
npx ts-node scripts/init-appwrite.ts
```

- Run the development server:

```
bash
```

```
npm run dev
```

- Access at <http://localhost:3000>.

Next Steps

- Create the folder structure as shown above.
- Copy the provided files into their respective directories.
- Create `.gitignore` and `.env.local` manually.
- Run the setup commands to install dependencies and initialize Appwrite.
- Test all routes (`/`, `/admin`, `/properties/[id]`, `/[slug]`, `/blog`, `/blog/[slug]`) and S3 uploads.
- Deploy to Vercel and configure domains (`real.harshafix.diy`, `blog.real.harshafix.diy`).

Let me know if you need the contents of `.gitignore`, `README.md`, or any additional files, or if you want help with a specific part of the setup!