

Digital Signage SaaS - Complete Setup Guide

Complete File Checklist

You have **20** files to create. Check them off as you complete each one:

Root Directory Files

- `package.json` (File #1)
- `next.config.js` (File #2)
- `tailwind.config.js` (File #3)
- `postcss.config.js` (File #4)
- `.gitignore` (File #13)
- `README.md` (File #14)

app/ Folder

- `app/layout.js` (File #5)
- `app/globals.css` (File #6)
- `app/page.js` (File #12)

app/player/ Folder

- `app/player/page.js` (File #19)

lib/ Folder

- `lib/config.js` (File #7)
- `lib/api.js` (File #8)
- `lib/utils.js` (File #9)

components/ Folder

- `components/Modal.js` (File #10)
- `components/EmptyState.js` (File #11)
- `components/SetupWizard.js` (File #15)
- `components/AuthScreen.js` (File #16)
- `components/SuperAdminDashboard.js` (File #17)
- `components/CustomerDashboard.js` (File #18)

public/ Folder

- `public/sw.js` (File #20)
-

Step 1: Create Folder Structure

On your computer, create this exact folder structure:

```
digital-signage-saas/
├── app/
│   └── player/
├── components/
├── lib/
└── public/
```

How to do this:

On Windows:

1. Right-click on Desktop → New → Folder
2. Name it: `(digital-signage-saas)`
3. Open that folder
4. Right-click inside → New → Folder (name it `(app)`)
5. Right-click inside → New → Folder (name it `(components)`)
6. Right-click inside → New → Folder (name it `(lib)`)
7. Right-click inside → New → Folder (name it `(public)`)
8. Open the `(app)` folder
9. Right-click inside → New → Folder (name it `(player)`)

On Mac:

1. Open Finder
2. Go to Desktop
3. File → New Folder (name it `(digital-signage-saas)`)
4. Open that folder
5. Create folders: `(app)`, `(components)`, `(lib)`, `(public)`
6. Inside `(app)`, create folder: `(player)`

Step 2: Create Each File

For each file, you'll:

1. Create a new text file in the correct folder
2. Name it exactly as shown (including the extension)
3. Copy the content from the artifact
4. Save it

How to Create Files:

On Windows:

1. Right-click in the folder → New → Text Document
2. Name it with the correct extension (e.g., `package.json`)
3. Right-click the file → Edit with Notepad (or your text editor)
4. Copy content from artifact
5. Paste into file
6. Save (Ctrl+S)

On Mac:

1. OpenTextEdit
 2. Format → Make Plain Text
 3. Copy content from artifact
 4. Paste
 5. File → Save (save to correct folder with correct name)
-

File Creation Guide

ROOT DIRECTORY FILES

Create these files in the main `digital-signage-saas` folder:

File 1: `package.json`

Location: `digital-signage-saas/package.json` Content: Copy from Artifact #1

File 2: next.config.js

Location: `(digital-signage-saas/next.config.js)` Content: Copy from Artifact #2

File 3: tailwind.config.js

Location: `(digital-signage-saas/tailwind.config.js)` Content: Copy from Artifact #3

File 4: postcss.config.js

Location: `(digital-signage-saas/postcss.config.js)` Content: Copy from Artifact #4

File 13: .gitignore

Location: `(digital-signage-saas/.gitignore)` Content: Copy from Artifact #13 **Note:** On Windows, you may need to name it `(.gitignore.)` (with dot at end), then rename in Git Bash

File 14: README.md

Location: `(digital-signage-saas/README.md)` Content: Copy from Artifact #14

APP/ FOLDER FILES

Create these inside the `(app)` folder:

File 5: layout.js

Location: `(digital-signage-saas/app/layout.js)` Content: Copy from Artifact #5

File 6: globals.css

Location: `(digital-signage-saas/app/globals.css)` Content: Copy from Artifact #6

File 12: page.js

Location: `(digital-signage-saas/app/page.js)` Content: Copy from Artifact #12

APP/PLAYER/ FOLDER

Create this inside the `(app/player)` folder:

File 19: page.js

Location: `(digital-signage-saas/app/player/page.js)` Content: Copy from Artifact #19

LIB/ FOLDER FILES

Create these inside the `lib` folder:

File 7: config.js

Location: `(digital-signage-saas/lib/config.js)` Content: Copy from Artifact #7

IMPORTANT: After deploying your Cloudflare Worker, you MUST update these lines:

```
javascript
```

```
export const API_URL = 'https://your-actual-worker-url.workers.dev';
export const INSTANTDB_APP_ID = 'your-actual-instantdb-app-id';
```

File 8: api.js

Location: `(digital-signage-saas/lib/api.js)` Content: Copy from Artifact #8

File 9: utils.js

Location: `(digital-signage-saas/lib/utils.js)` Content: Copy from Artifact #9

COMPONENTS/ FOLDER FILES

Create these inside the `components` folder:

File 10: Modal.js

Location: `(digital-signage-saas/components/Modal.js)` Content: Copy from Artifact #10

File 11: EmptyState.js

Location: `(digital-signage-saas/components/EmptyState.js)` Content: Copy from Artifact #11

File 15: SetupWizard.js

Location: `(digital-signage-saas/components/SetupWizard.js)` Content: Copy from Artifact #15

File 16: AuthScreen.js

Location: `(digital-signage-saas/components/AuthScreen.js)` Content: Copy from Artifact #16

File 17: SuperAdminDashboard.js

Location: `(digital-signage-saas/components/SuperAdminDashboard.js)` Content: Copy from Artifact #17

File 18: CustomerDashboard.js

PUBLIC/ FOLDER FILES

Create this inside the `(public)` folder:

File 20: sw.js

Location: `(digital-signage-saas/public/sw.js)` Content: Copy from Artifact #20

Step 3: Upload to GitHub

Option A: Using GitHub Website (Easier for Beginners)

1. Go to GitHub.com

- Log in to your account

2. Create New Repository

- Click the "+" button (top right)
- Click "New repository"
- Repository name: `(digital-signage-saas)`
- Description: "Multi-tenant digital signage platform"
- Choose Public or Private
- **DO NOT** check "Add a README file" (we already have one)
- Click "Create repository"

3. Upload Your Files

- On the repository page, click "uploading an existing file"
- Drag and drop your entire `(digital-signage-saas)` folder
- **IMPORTANT:** Make sure you're uploading the contents, not the folder itself
- The files should appear in the root, like:

```
package.json
app/
components/
lib/
public/
```

- Add commit message: "Initial commit - complete SaaS platform"
- Click "Commit changes"

Option B: Using GitHub Desktop (Recommended)

1. Download GitHub Desktop

- Go to: <https://desktop.github.com/>
- Download and install

2. Sign In

- Open GitHub Desktop
- File → Options → Sign In
- Sign in with your GitHub account

3. Create Repository

- File → New Repository
- Name: `digital-signage-saas`
- Local Path: Choose where to save (e.g., Documents)
- Click "Create Repository"

4. Copy Your Files

- Open the repository folder (File → Show in Explorer/Finder)
- Copy ALL your files from your `digital-signage-saas` folder
- Paste them into this repository folder
- Your structure should look like:

```
package.json
app/layout.js
app/page.js
components/Modal.js
etc...
```

5. Commit and Push

- Go back to GitHub Desktop
- You'll see all files listed
- Summary: "Initial commit - complete platform"

- Click "Commit to main"
 - Click "Publish repository" (top right)
 - Choose Public or Private
 - Click "Publish repository"
-

Step 4: Deploy to Vercel

1. Go to Vercel.com

- Sign up or log in
- Use "Continue with GitHub" (easiest)

2. Import Project

- Click "Add New..." → "Project"
- You'll see your GitHub repositories
- Find `digital-signage-saas`
- Click "Import"

3. Configure Project

- Framework Preset: **Next.js** (should auto-detect)
- Root Directory: `/` (leave as default)
- Build Command: (leave as default)
- Output Directory: (leave as default)
- **DON'T add environment variables yet**
- Click "Deploy"

4. Wait for Deployment

- This takes 2-3 minutes
- You'll see build logs
- When complete, you'll get a URL like: `https://your-project.vercel.app`

5. Visit Your Site

- Click "Visit" or copy the URL
- You should see the Setup Wizard!

Step 5: Configure Services

Before the setup wizard will work, you need:

1. Deploy Cloudflare Worker

See the separate Cloudflare Worker artifact (Artifact #1 from earlier conversation)

1. Install Wrangler: `npm install -g wrangler`
2. Create project folder
3. Add worker code to `src/index.js`
4. Create `wrangler.toml` with your credentials
5. Deploy: `wrangler deploy`
6. Note your worker URL

2. Update Frontend Configuration

1. In your GitHub repository, edit `lib/config.js`
2. Change these lines:

```
javascript

export const API_URL = 'https://signage-api.YOUR-SUBDOMAIN.workers.dev';
export const INSTANTDB_APP_ID = 'your-actual-app-id-from-instantdb';
```

3. Commit the changes
4. Vercel will automatically redeploy (takes 1 minute)

3. Complete Setup Wizard

1. Visit your Vercel URL
2. You should now see the Setup Wizard
3. Click through Step 1 (service checklist)
4. Step 2: Create your super admin account
 - Enter your name
 - Enter email
 - Enter password (min 8 characters)

- Click "Complete Setup"
5. You're now logged in as super admin!
-

Step 6: Test Everything

Test Customer Signup

1. Open incognito/private window
2. Go to your Vercel URL
3. Click "Sign up"
4. Create a test customer account
5. Verify you can see the customer dashboard

Test Firestick Player

1. On Firestick, open Silk Browser
2. Go to: <https://your-project.vercel.app/player>
3. You should see a 6-character pairing code
4. In customer dashboard, click "Pair Screen"
5. Enter the code
6. Screen should update to "Waiting for content"

Test Full Flow

1. Upload a test image in Media tab
 2. Create a playlist in Playlists tab
 3. Add the image to the playlist
 4. Go to Screens tab
 5. Assign the playlist to your screen
 6. Screen should start displaying the image!
-

Troubleshooting

"Module not found" errors

- Make sure all files are in the correct folders
- Check file names match exactly (case-sensitive)
- Verify folder structure matches the guide

Setup wizard doesn't appear

- Check `lib/config.js` has correct API_URL
- Open browser console (F12) to see actual error
- Verify Cloudflare Worker is deployed and accessible

Vercel build fails

- Check the build logs for specific error
- Most common: Missing files or wrong folder structure
- Solution: Re-check file locations

Can't upload to GitHub

- Make sure you're uploading file contents, not the parent folder
- Try GitHub Desktop if website upload isn't working
- Check you have permissions on the repository

Success Checklist

- All 20 files created in correct locations
- Files uploaded to GitHub successfully
- Vercel deployment successful and live
- Cloudflare Worker deployed
- `lib/config.js` updated with correct URLs
- Setup wizard completed
- Super admin account created
- Can log in as super admin
- Test customer account created
- Firestick player shows pairing code
- Successfully paired a screen

- Uploaded media file
 - Created playlist
 - Assigned playlist to screen
 - Content playing on screen
-

Need Help?

If you get stuck:

1. **Check the error message carefully** - it usually tells you what's wrong
 2. **Verify file locations** - most issues are from files in wrong folders
 3. **Check the browser console** - Press F12 to see JavaScript errors
 4. **Review Vercel build logs** - shows exactly where build failed
 5. **Double-check configuration** - API_URL and InstantDB App ID must be correct
-

Next Steps After Setup

Once everything is working:

1. Add your custom domain in Vercel settings
2. Configure multiple Firesticks with the player URL
3. Test offline mode (disconnect Firestick WiFi)
4. Create more customer test accounts
5. Adjust customer limits from super admin dashboard
6. Plan your Stripe integration for paid tiers

Congratulations! You now have a fully functional SaaS platform! 