

THE SaaS TECH STACK

5 tools, one codebase, ship a real product. What each one does and how to start.

1

NEXT.JS – frontend + backend

One codebase for your pages and your API routes. React-based, huge community.

```
npx create-next-app@latest my-saas
```

 Docs: nextjs.org

2

SUPABASE – database + auth + realtime

Postgres database, login (email, Google), file storage and realtime in one place. Start in the dashboard, create a project, copy the URL and anon key.

```
npm i @supabase/supabase-js @supabase/ssr
```

supabase.com

3

VERCEL – deploy on every push

Push your repo to GitHub, import it in Vercel, add your environment variables, done. Every push to main becomes a live deploy, every PR gets a preview link.

vercel.com

4

RESEND – transactional emails

Welcome emails, password resets, receipts. Verify your domain first (DNS records), then send with one API call.

```
npm i resend
```

resend.com

5

CLOUDFLARE R2 – file storage

S3-compatible object storage for images, uploads and exports, with no egress (download) fees. Create a bucket, then an API token.

developers.cloudflare.com/r2

START IN THIS ORDER

1. Create the Next.js app and push it to GitHub.
2. Create a Supabase project, add the URL + anon key to `.env.local`.
3. Add login (Supabase Auth) and your first table with Row Level Security turned on.
4. Deploy on Vercel and add the same env vars there.
5. Add Resend (verify your domain) for emails.
6. Add R2 when you need user uploads.

SUPABASE CLIENT

```
// lib/supabase.ts
import { createClient } from '@supabase/supabase-js'
export const supabase = createClient(
  process.env.NEXT_PUBLIC_SUPABASE_URL!,
  process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!
)
```

SEND AN EMAIL (RESEND)

```
// app/api/welcome/route.ts
import { Resend } from 'resend'
const resend = new Resend(process.env.RESEND_API_KEY)
await resend.emails.send({
  from: 'You <hello@yourdomain.com>',
  to: ['user@email.com'],
  subject: 'Welcome!',
  html: '<p>Glad you are here.</p>'
})
```

UPLOAD TO R2 (S3 API)

```
npm i @aws-sdk/client-s3
const s3 = new S3Client({
  region: 'auto',
  endpoint: `https://${ACCOUNT_ID}.r2.cloudflarestorage.com`,
  credentials: { accessKeyId, secretAccessKey }
})
await s3.send(new PutObjectCommand({ Bucket: 'uploads', Key: 'a.png', Body }))
```

BEFORE YOU LAUNCH

- ☐ Row Level Security on every table
- ☐ Secrets only in env vars, never in the repo
- ☐ Service-role / secret keys never in frontend code
- ☐ Custom domain on Vercel
- ☐ Email domain verified
- ☐ Check each service's current free-tier limits and pricing