Contents:   
SaaSFast

* [Get Started!](https://saasfast.docs.smorchestra.ai/)
* Lessons
  + [Ship in 5 minutes](https://saasfast.docs.smorchestra.ai/Shipin5minutes/)
  + [Static Page](https://saasfast.docs.smorchestra.ai/components/)
  + [User Authentication](https://saasfast.docs.smorchestra.ai/NextAuth/)
  + [API Handling](https://saasfast.docs.smorchestra.ai/api_call/)
  + [Private Pages](https://saasfast.docs.smorchestra.ai/Privatepage/)
  + [Stripe Subscriptions](https://saasfast.docs.smorchestra.ai/stripe/)
  + [Privacy Policy & Terms](https://saasfast.docs.smorchestra.ai/Privacypolicy/)
* Features
  + [SEO Tags](https://saasfast.docs.smorchestra.ai/seo/)
  + [Database](https://saasfast.docs.smorchestra.ai/Database_nexauth/)
  + [Email](https://saasfast.docs.smorchestra.ai/email/)
  + [Payments](https://saasfast.docs.smorchestra.ai/Payments_strip/)
  + [Google Authentication (OAuth)](https://saasfast.docs.smorchestra.ai/Goole_auth/)
  + [Magic Links](https://saasfast.docs.smorchestra.ai/MagicLinks/)
  + [Customer Support](https://saasfast.docs.smorchestra.ai/Customersupport/)
  + [Error Handling](https://saasfast.docs.smorchestra.ai/Error_handling/)
  + [Analytics](https://saasfast.docs.smorchestra.ai/Analytics/)
* Components
  + [Components](https://saasfast.docs.smorchestra.ai/Components_tab/)
  + [Header](https://saasfast.docs.smorchestra.ai/Components_header/)
  + [Hero Section](https://saasfast.docs.smorchestra.ai/hero_component/)
  + [Features Listicle](https://saasfast.docs.smorchestra.ai/component_FeaturesListicle/)
  + [Features Accordion](https://saasfast.docs.smorchestra.ai/FeaturesAccordion/)
  + [WithWithout](https://saasfast.docs.smorchestra.ai/component_WithWithout/)
  + [Call To Action (CTA)](https://saasfast.docs.smorchestra.ai/component_CTA/)
  + [Pricing Section](https://saasfast.docs.smorchestra.ai/component_Pricing/)
  + [FAQ Section](https://saasfast.docs.smorchestra.ai/component_FAQ/)
  + [Testimonial Small](https://saasfast.docs.smorchestra.ai/componen_TestimonialSmall/)
  + [Testimonial Single](https://saasfast.docs.smorchestra.ai/TestimonialSingle/)
  + [Testimonial Triple](https://saasfast.docs.smorchestra.ai/component_Testimonial_Triple/)
  + [Testimonial Grid](https://saasfast.docs.smorchestra.ai/componen_TestimonialGrid/)
  + [Footer](https://saasfast.docs.smorchestra.ai/componen_Footer/)
  + [Button Lead](https://saasfast.docs.smorchestra.ai/componen_ButtonLead/)
  + [Button Checkout](https://saasfast.docs.smorchestra.ai/componen_ButtonCheckout/)
  + [Button Sign-in](https://saasfast.docs.smorchestra.ai/componen_ButtonSign_in/)
  + [Button Gradient](https://saasfast.docs.smorchestra.ai/componen_ButtonGradient/)
  + [Tabs](https://saasfast.docs.smorchestra.ai/component_taps/)
  + [Modal](https://saasfast.docs.smorchestra.ai/componen_Modal/)
* [Deployment](https://saasfast.docs.smorchestra.ai/deployment/)
* [Extras](https://saasfast.docs.smorchestra.ai/Extras/)

Table of contents

* [Welcome to SaaSFast, Innovator](https://saasfast.docs.smorchestra.ai/#welcome-to-saasfast-innovator)

Let’s Get Started!

Welcome to SaaSFast, Innovator

This is your quick guide to getting the template up and running with ease.[¶](https://saasfast.docs.smorchestra.ai/#this-is-your-quick-guide-to-getting-the-template-up-and-running-with-ease)

Once you’re done, jump straight into this quick-start guide to launch your project in just 5 minutes.

It’s time to turn your idea into a real startup—fast ⚡️

Run the Local Server[¶](https://saasfast.docs.smorchestra.ai/#run-the-local-server)

Open your terminal and run the following commands step-by-step:

git clone https://github.com/Marc-Lou-Org/SaaSFast.git [YOUR\_APP\_NAME]

cd [YOUR\_APP\_NAME]

npm install

git remote remove origin

npm run dev

Rename the .env.local[¶](https://saasfast.docs.smorchestra.ai/#rename-the-envlocal)

mv .env.example .env.local

Then open this link in your browser:[¶](https://saasfast.docs.smorchestra.ai/#then-open-this-link-in-your-browser)

[http://localhost:3000](http://localhost:3000/) — and boom! Your site is live ✅

You may see some warnings in the console. Don’t worry—they’re harmless and won’t break the app.

SaaSFast Project Structure (NextJS)[¶](https://saasfast.docs.smorchestra.ai/#saasfast-project-structure-nextjs)

* /app → Your pages (Each folder + page.js = one page)
* /app/api → Your API routes (One file = one endpoint)
* /components → Reusable React components
* /libs → Helper functions and integrations (Stripe, Resend, auth, etc.)
* /models → Database schemas

config.js File[¶](https://saasfast.docs.smorchestra.ai/#configjs-file)

This is where your app settings live. Each key is documented so you know exactly what it does and how to use it. Take time to go through it—it’s the backbone of your app.

.env File[¶](https://saasfast.docs.smorchestra.ai/#env-file)

Rename .env.example to .env.local and update the NEXTAUTH\_SECRET value to something secure (at least 15 characters).

Here’s how it should look:

NEXTAUTH\_URL=http://localhost:3000

NEXTAUTH\_SECRET=your\_secure\_secret\_here

GOOGLE\_ID=

GOOGLE\_SECRET=

MONGODB\_URI=

RESEND\_API\_KEY=

STRIPE\_PUBLIC\_KEY=

STRIPE\_SECRET\_KEY=

STRIPE\_WEBHOOK\_SECRET=

* Lessons
* [Ship in 5 minutes](https://saasfast.docs.smorchestra.ai/Shipin5minutes/)

Launch Your Startup and Reach Customers in Just 5 Minutes ⚡️[¶](https://saasfast.docs.smorchestra.ai/Shipin5minutes/#launch-your-startup-and-reach-customers-in-just-5-minutes)

Now it’s time to build a clean landing page and optionally add a waitlist form to collect emails. Complete this guide and your project will be featured on our homepage ⭐️

Haven’t set things up yet? Clone the repo and start your local server first. Check the “Get Started” guide here → [Let’s Begin](https://saasfast.docs.smorchestra.ai/) Then, delete all the content in /app/page.js and paste the code below:

import { Suspense } from 'react'

import Header from "@/components/Header";

import Hero from "@/components/Hero";

import Problem from "@/components/Problem";

import FeaturesAccordion from "@/components/FeaturesAccordion";

import Pricing from "@/components/Pricing";

import FAQ from "@/components/FAQ";

import CTA from "@/components/CTA";

import Footer from "@/components/Footer";

export default function Home() {

return (

<>

<Suspense>

<Header />

</Suspense>

<main>

<Hero />

<Problem />

<FeaturesAccordion />

<Pricing />

<FAQ />

<CTA />

</main>

<Footer />

</>

);

}

✏️ Edit the Content to Match Your Project Idea[¶](https://saasfast.docs.smorchestra.ai/Shipin5minutes/#edit-the-content-to-match-your-project-idea)

Each component includes notes to help you write strong, marketing-friendly content—check the components folder for guidance. Congrats! You now have a professional landing page to show off your project 🎉

📨 (Optional) Collect Email Addresses[¶](https://saasfast.docs.smorchestra.ai/Shipin5minutes/#optional-collect-email-addresses)

Want to build a waitlist?

1. Set up your database.
2. Uncomment the code in /api/lead/route.js to start saving emails directly.

🚀 (Optional) Customize the Main CTA Button[¶](https://saasfast.docs.smorchestra.ai/Shipin5minutes/#optional-customize-the-main-cta-button)

You can also replace the default call-to-action (CTA) button in the **Hero**, **Pricing**, and **CTA** sections with this component:

import ButtonLead from "@/components/ButtonLead";

// For Hero & CTA sections:

<ButtonLead />

// For the Pricing section (full-width style):

<ButtonLead extraStyle="!max-w-none !w-full" />

Ready to Launch?[¶](https://saasfast.docs.smorchestra.ai/Shipin5minutes/#ready-to-launch)

Need help publishing? Follow the [Deployment Guide](https://saasfast.docs.smorchestra.ai/Shipin5minutes/) for step-by-step instructions.

* Static Page

🧩 SaaSFast Components[¶](https://saasfast.docs.smorchestra.ai/components/#saasfast-components)

SaaSFast comes with many built-in components to help you create SEO-friendly pages quickly and easily:

* Hero
* Pricing
* FAQ (Frequently Asked Questions)
* and more...

🛠 You’ll find these components inside the /components folder.  
🔍 Use /libs/seo.js to configure SEO tags and improve your site’s visibility in search results.

✅ Here’s a simple example of a landing page you can build quickly:

import { getSEOTags } from "@/libs/seo";

export const metadata = getSEOTags({ canonicalUrlRelative: "/" });

export default function Landing() {

return (

<>

<main

className="min-h-screen p-12 pb-24 text-center"

data-theme="dark"

>

<section className="max-w-xl mx-auto space-y-8">

<h1 className="text-3xl md:text-4xl font-extrabold">

Food recipes you&apos;ll love 🥦

</h1>

<p className="text-lg leading-relaxed text-base-content/80">

Our AI will generate recipes based on your preferences. New recipes

will be added every week!

</p>

<img

src="https://images.unsplash.com/photo-1518843875459-f738682238a6?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=3484&q=80"

alt="Vegetables"

width={500}

height={250}

className="rounded-lg mx-auto"

/>

<button className="btn btn-primary btn-wide">Get started</button>

</section>

</main>

</>

);

}

* User Authentication

NEXT-AUTH + MONGO-DBSUPABASE

🔐 Authentication with NextAuth[¶](https://saasfast.docs.smorchestra.ai/NextAuth/#authentication-with-nextauth)

**SaaSFast** uses **NextAuth** for user login.  
You can configure it in the file:  
/app/api/auth/[...nextauth]/route.js

Supported Login Methods:[¶](https://saasfast.docs.smorchestra.ai/NextAuth/#supported-login-methods)

* ✅ Magic Links
* ✅ Google OAuth

Once you've completed at least one of the guides above, you can enable user sign-in like this:

"use client"

import { signIn } from "next-auth/react";

import config from "@/config";

const SigninButton = () => {

return (

<button

className="btn btn-primary"

onClick={() => signIn(undefined, { callbackUrl: config.auth.callbackUrl })}

>

Login

</button>

);

};

export default SigninButton;

🌐 Working with API[¶](https://saasfast.docs.smorchestra.ai/api_call/#working-with-api)

NEXT-AUTH + MONGO-DBSUPABASE

* API Handling[¶](https://saasfast.docs.smorchestra.ai/api_call/#api-handling)

Each file named route.js inside the /app/api folder represents an API endpoint.  
You can use the helper file /libs/api.js (which is a pre-configured instance of axios with interceptors) to simplify your API requests. It automatically does the following:

* Displays error messages
* Redirects to the login page on a 401 Unauthorized error
* Automatically prefixes requests with /api

📌 Example: instead of /api/user/posts you can simply use /user/posts.

**Example:** user/posts/ ←

🔐 Securing Sensitive API Requests[¶](https://saasfast.docs.smorchestra.ai/api_call/#securing-sensitive-api-requests)

✅ **NextAuth** handles authentication automatically using cookies.

// /app/user-profile/page.js

"use client"

import { useState } from "react";

import apiClient from "@/libs/api";

const UserProfile = () => {

const [isLoading, setIsLoading] = useState(false);

const saveUser = async () => {

setIsLoading(true);

try {

const { data } = await apiClient.post("/user", {

email: "new@gmail.com",

});

console.log(data);

} catch (e) {

console.error(e?.message);

} finally {

setIsLoading(false);

}

};

return (

<button className="btn btn-primary" onClick={() => saveUser()}>

{isLoading && (

<span className="loading loading-spinner loading-sm"></span>

)}

Save

</button>

);

};

export default UserProfile;

🖥️ On the Backend[¶](https://saasfast.docs.smorchestra.ai/api_call/#on-the-backend)

On the backend, you can retrieve the session and use it to fetch the user’s data from the database. Make sure your database is configured first.

Here’s what your API file would look like:

// /app/api/user/route.js

import { NextResponse } from "next/server";

import { getServerSession } from "next-auth/next";

import { authOptions } from "@/libs/next-auth";

import connectMongo from "@/libs/mongoose";

import User from "@/models/User";

export async function POST(req) {

const session = await getServerSession(authOptions);

if (session) {

const { id } = session.user;

const body = await req.json();

if (!body.email) {

return NextResponse.json({ error: "Email is required" }, { status: 400 });

}

try {

await connectMongo();

const user = await User.findById(id);

if (!user) {

return NextResponse.json({ error: "User not found" }, { status: 404 });

}

user.email = body.email;

await user.save();

return NextResponse.json({ data: user }, { status: 200 });

} catch (e) {

console.error(e);

return NextResponse.json(

{ error: "Something went wrong" },

{ status: 500 }

);

}

} else {

// Not Signed in

return NextResponse.json({ error: "Not signed in" }, { status: 401 });

}

}

* Private Pages

NEXT-AUTH + MONGO-DBSUPABASE

🔒 Creating Private Pages After Authentication[¶](https://saasfast.docs.smorchestra.ai/Privatepage/#creating-private-pages-after-authentication)

Once the user is authenticated, you can create private pages such as a **dashboard**, **profile**, and more.

**Making Protected API Requests**[¶](https://saasfast.docs.smorchestra.ai/Privatepage/#making-protected-api-requests)

If you'd like to perform protected API calls, follow this detailed guide.

**Securing Routes with Layout Files**[**¶**](https://saasfast.docs.smorchestra.ai/Privatepage/#securing-routes-with-layout-files)

The layout.js file inside the /dashboard directory ensures that all child pages under this route are protected.  
If the user is not authenticated, they’ll be redirected automatically to the login page (as defined in the config.js auth settings).

**User Dashboard Example**[¶](https://saasfast.docs.smorchestra.ai/Privatepage/#user-dashboard-example)

Here’s a basic example of a user dashboard that displays private user data using a **server component**:

// /app/dashboard/page.js

import { getServerSession } from "next-auth";

import { authOptions } from "@/libs/next-auth";

import connectMongo from "@/libs/mongoose";

import User from "@/models/User";

export default async function Dashboard() {

await connectMongo();

const session = await getServerSession(authOptions);

const user = await User.findById(session.user.id);

return (

<>

<main className="min-h-screen p-8 pb-24">

<section className="max-w-xl mx-auto space-y-8">

<h1 className="text-3xl md:text-4xl font-extrabold">

User Dashboard

</h1>

<p>Welcome {user.name} 👋</p>

<p>Your email is {user.email}</p>

</section>

</main>

</>

);

}

* Stripe Subscriptions

Let's Start by Setting Up Stripe Checkout to Enable Subscriptions[¶](https://saasfast.docs.smorchestra.ai/stripe/#lets-start-by-setting-up-stripe-checkout-to-enable-subscriptions)

We’ll leave the job of granting user access to our custom webhook.

Prerequisites[¶](https://saasfast.docs.smorchestra.ai/stripe/#prerequisites)

You should already have: - A configured **Stripe account** - A ready **database**

🛠️ Setup[¶](https://saasfast.docs.smorchestra.ai/stripe/#setup)

1. In your Stripe dashboard, go to:  
   **[More +] > [Product Catalog] > [+ Add Product]**
2. Enter the product name and set a **monthly price** (or any pricing model you use), then click **[Save Product]**.
3. After saving the product, go to the **[Pricing]** section and **copy the Price ID** (it starts with price\_).
4. Add this ID to the first plan in the stripe.plans array inside your config.js file.

🧾 Page Setup[¶](https://saasfast.docs.smorchestra.ai/stripe/#page-setup)

Open the file /dashboard/page.js and paste the following code:

import ButtonAccount from "@/components/ButtonAccount";

import ButtonCheckout from "@/components/ButtonCheckout";

import config from "@/config";

export const dynamic = "force-dynamic";

export default async function Dashboard() {

return (

<main className="min-h-screen p-8 pb-24">

<section className="max-w-xl mx-auto space-y-8">

<ButtonAccount />

<h1 className="text-3xl md:text-4xl font-extrabold">

Subscribe to get access:

</h1>

<ButtonCheckout

mode="subscription"

priceId={config.stripe.plans[0].priceId}

/>

</section>

</main>

);

}

🧪 Payment Test[¶](https://saasfast.docs.smorchestra.ai/stripe/#payment-test)

After that, open this URL in your browser: [**http://localhost:3000/dashboard**](http://localhost:3000/dashboard)

* Log in, click the payment button, and try the test card: 4242 4242 4242 4242.

🔄 Access Handling[¶](https://saasfast.docs.smorchestra.ai/stripe/#access-handling)

The webhook at /api/webhook/stripe/route.js listens to Stripe events and handles granting or revoking access to the user accordingly.

* Check the hasAccess field in the User.js model (for NextAuth) or has\_access column in the profiles table (for Supabase).

✅ **Note:** You must have the Webhook running **locally** during development for this to work correctly.

🧩 Webhook Customization[¶](https://saasfast.docs.smorchestra.ai/stripe/#webhook-customization)

You can customize the logic inside /api/webhook/stripe/route.js to:

* Send **abandoned cart** emails
* Deduct user balance
* Trigger specific email or notification flows

🧾 Subscription Management[¶](https://saasfast.docs.smorchestra.ai/stripe/#subscription-management)

Users can manage their subscriptions (cancel, change payment method...) via the <ButtonAccount /> component.

* Privacy Policy & Terms

**📄 Privacy Policy & Terms of Service Made Easy**

Setting up Stripe or applying for Google OAuth requires you to have a Privacy Policy and Terms of Service. But writing legal docs from scratch? Not fun.

The **SaaSFast** boilerplate includes **built-in ChatGPT prompts** to help you generate these documents in minutes.

✅ Just head to these files in your project:

* /app/tos/page.js – Terms of Service
* /app/privacy-policy/page.js – Privacy Policy

Add your business details, feed the prompt to ChatGPT, and you're good to go.

No lawyer needed (for basic setups).

* Features
* [SEO Tags](https://saasfast.docs.smorchestra.ai/seo/)

⚙️ Setup[¶](https://saasfast.docs.smorchestra.ai/seo/#setup)

1️⃣ Open the config.js file and add the following values:  
appName, appDescription, and domainName.

🧠 These values will be used as default SEO meta tags.

2️⃣ The helper /libs/seo.js is used to add all essential SEO tags (with default values) to all pages via the main layout file /app/layout.js.

3️⃣ To add custom SEO tags to a specific page without rewriting everything, do the following:

📄 /app/terms/page.js

import { getSEOTags } from "@/libs/seo";

...

export const metadata = getSEOTags({

title: "Terms and Conditions | SaaSFast",

canonicalUrlRelative: "/tos",

});

export default async function TermsAndConditions() {

...

✅ It’s recommended to always define title and canonicalUrlRelative for each page.

🧩 When needed, add structured data to the page using the renderSchemaTags() function in /libs/seo.js.

🔍 This helps Google better understand your site and may enable rich results.

📘 Open the component for more details.

📄 Example: /app/page.js

import { renderSchemaTags } from "@/libs/seo";

export default function Page() {

return (

<>

{renderSchemaTags()}

<main className="flex min-h-screen flex-col items-center justify-center text-center gap-12 p-8">

<h1 className="text-3xl font-extrabold"> SaaSFast</h1>

...

</main>

</>

);

}

4️⃣ Add your root URL to siteUrl (e.g., https://yourdomain.com) inside the next-sitemap.config.js file in the root folder.

🗺️ This will generate sitemap.xml and robots.txt files for all your pages during build.

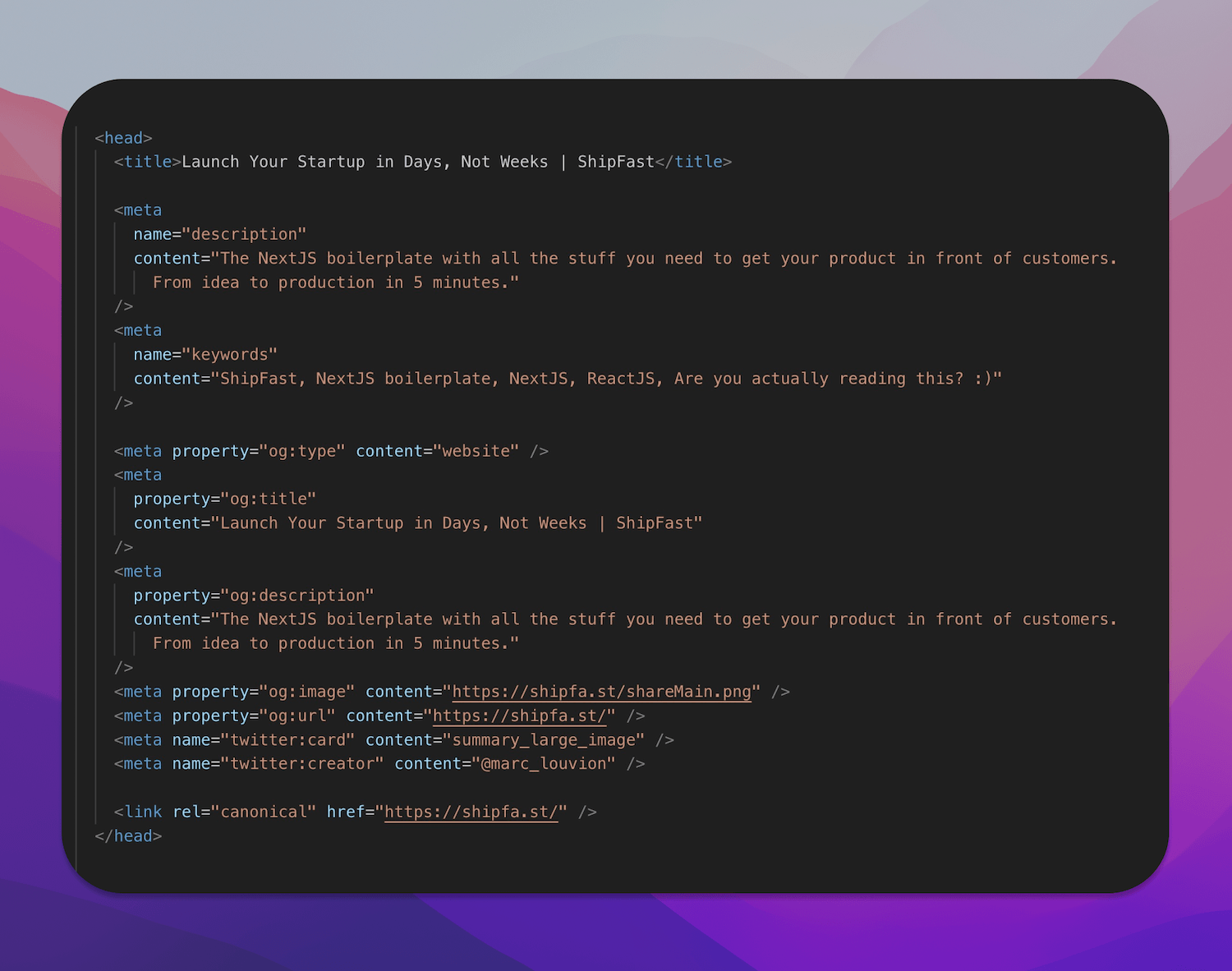
5️⃣ Verify your domain in **Google Search Console** to help with site indexing.

📝 Create a Blog in Minutes[¶](https://saasfast.docs.smorchestra.ai/seo/#create-a-blog-in-minutes)

📂 Inside the folder /app/blog/\_assets, you’ll find the content.js file which contains all your blog posts, authors, categories, and blog style.

✍️ Just add your content there and **SaaSFast** will automatically generate a blog for you.

📚 For more details, check the blog section.



```

* Database

Setup[¶](https://saasfast.docs.smorchestra.ai/Database_nexauth/#setup)

1️⃣ **Create a new project and publish a cluster on MongoDB Atlas.**

2️⃣ **Run a local database for development setup:**  
- 💻 This allows you to work offline and speeds up development.

3️⃣ **Inside your MongoDB Atlas project:**  
- 🌐 **Click on [Network Access] then [+ Add IP Address].**  
- 🔑 **Enter:** 0.0.0.0/0 in the **[Access List Entry].**  
- 🌍 **Why:** This allows access from your machine and from deployed environments (like Vercel).

4️⃣ **Environment variable setup:**  
- 📝 If you haven’t already, rename .env.example to .env.local.  
- 🔗 **Then add your connection string to** MONGODB\_URI **inside** .env.local.

🛠️ Mongoose (Optional)[¶](https://saasfast.docs.smorchestra.ai/Database_nexauth/#mongoose-optional)

* 🗃️ **Why use Mongoose:**
* It simplifies working with MongoDB.
* It offers helpful features out of the box.
* 📂 **Model organization:**
* Define your models inside the /models folder.
* Add any new models there.
* ✅ **Frontend optimization:**
* The toJSON plugin has been added to all models to remove the \_id and \_\_v fields.
* If you set private: true on any field, it will be excluded from the response.
* For example, you can make the email field private so it won’t be sent to the frontend.
* Email

✉️ Mailgun Setup[¶](https://saasfast.docs.smorchestra.ai/email/#mailgun-setup)

📝 Introduction[¶](https://saasfast.docs.smorchestra.ai/email/#introduction)

Mailgun is not required, but you’ll need an email service to handle magic login links, abandoned cart emails, and more.

⚠️ Notes:[¶](https://saasfast.docs.smorchestra.ai/email/#notes)

* Mailgun silently removed the "pay-as-you-go" tier from the pricing page, but it still exists.
* 💡 Tip: Start with the $35 trial and cancel it.
* 🔄 Result: You’ll be downgraded to the free “pay-as-you-go” tier automatically.
* 💰 Cost: If you send fewer than 1,000 emails per month, it’s only $1/month.

⚙️ Setup[¶](https://saasfast.docs.smorchestra.ai/email/#setup)

1️⃣ Create a new Mailgun account.  
2️⃣ Under [Sending], go to [Domains] → [+ Add Domain]  
- 🌐 Recommended: Use a subdomain like mail.yourdomain.com.

3️⃣ If using the EU region:  
- Add the following in libs/mailgun.js:

url: "https://api.eu.mailgun.net/"

````

4️⃣ Complete all DNS verification steps

\* 🌍 If using a subdomain, make sure your DNS records are updated.

5️⃣ Add an additional DMARC record to improve deliverability:

TXT | \_dmarc.mail.yourdomain.com | v=DMARC1; p=none

6️⃣ SMTP Setup:

\* Go to \[Domain Settings] → \[SMTP Credentials] → \[Reset Password]

\* Choose \[Automatic] → \[Create Password]

\* Click \[Copy] in the bottom-right of the popup

\* 📂 In `.env.local`, set:

```bash

smtp://postmaster@[mail.yourdomain.com]:[copied\_password]@smtp.mailgun.org:587

7️⃣ API Key Setup:

* Under [Sending API Keys], click [Create sending key]
* Add to .env.local as MAILGUN\_API\_KEY

8️⃣ Update config.js:

mailgun: {

subdomain: "mg",

fromNoReply: `SaaSFast <noreply@mg.SaaSfa.st>`,

fromAdmin: `Marc at SaaSFast <marc@mg.SaaSfa.st>`,

supportEmail: "marc@mg.SaaSfa.st",

forwardRepliesTo: "marc.louvion@gmail.com",

}

9️⃣ TypeScript Config (optional): In types/config.ts, add:

mailgun: {

subdomain: string;

fromNoReply: string;

fromAdmin: string;

supportEmail?: string;

forwardRepliesTo?: string;

};

🗃️ Mailgun Library Setup[¶](https://saasfast.docs.smorchestra.ai/email/#mailgun-library-setup)

🔟 Create libs/mailgun.js with:

import config from "@/config";

const formData = require("form-data");

const Mailgun = require("mailgun.js");

const mailgun = new Mailgun(formData);

const mg = mailgun.client({

username: "api",

key: process.env.MAILGUN\_API\_KEY || "dummy",

});

export const sendEmail = async ({ to, subject, text, html, replyTo }) => {

const data = {

from: config.mailgun.fromAdmin,

to: [to],

subject,

text,

html,

...(replyTo && { "h:Reply-To": replyTo }),

};

await mg.messages.create(

(config.mailgun.subdomain ? `${config.mailgun.subdomain}.` : "") + config.domainName,

data

);

};

11️⃣ Update EmailProvider in libs/next-auth.js:

EmailProvider({

server: process.env.EMAIL\_SERVER,

from: config.mailgun.fromNoReply,

}),

12️⃣ Replace all config.resend with config.mailgun, and libs/resend with libs/mailgun across your codebase.

📧 Receiving Emails[¶](https://saasfast.docs.smorchestra.ai/email/#receiving-emails)

13️⃣ Go to [Receiving] → [Create Route]:

* Choose [Match Recipient]
* Add the email address you'll send from (e.g. name@mail.yourdomain.com)
* Forward to:

https://[your-domain].com/api/webhook/mailgun

* Click [Create Route]

14️⃣ Webhook Setup:

* Go to [Sending] → [Webhooks]
* Copy the HTTP Webhook Signing Key
* Add to .env.local as MAILGUN\_SIGNING\_KEY

15️⃣ Add a forwarding email (e.g., your Gmail) to mailgun.forwardRepliesTo in config.js.

🗂️ Create Mailgun Webhook Route[¶](https://saasfast.docs.smorchestra.ai/email/#create-mailgun-webhook-route)

16️⃣ Create app/api/webhook/mailgun/route.js:

import { NextResponse } from "next/server";

import { sendEmail } from "@/libs/mailgun";

import config from "@/config";

import crypto from "crypto";

export async function POST(req) {

try {

const formData = await req.formData();

const signingKey = process.env.MAILGUN\_SIGNING\_KEY;

const timestamp = formData.get("timestamp").toString();

const token = formData.get("token").toString();

const signature = formData.get("signature").toString();

const hash = crypto.createHmac("sha256", signingKey)

.update(timestamp + token)

.digest("hex");

if (hash !== signature) {

return NextResponse.json({ error: "Invalid signature" }, { status: 401 });

}

const sender = formData.get("From");

const subject = formData.get("Subject");

const html = formData.get("body-html");

if (config.mailgun.forwardRepliesTo && html && subject && sender) {

await sendEmail({

to: config.mailgun.forwardRepliesTo,

subject: `${config?.appName} | ${subject}`,

html: `<div><p><b>- Subject:</b> ${subject}</p><p><b>- From:</b> ${sender}</p><p><b>- Content:</b></p><div>${html}</div></div>`,

replyTo: sender,

});

}

return NextResponse.json({});

} catch (e) {

console.error(e?.message);

return NextResponse.json({ error: e?.message }, { status: 500 });

}

}

🔄 DNS Records Check[¶](https://saasfast.docs.smorchestra.ai/email/#dns-records-check)

Check DNS record validity using **MxToolbox** (enter the subdomain if applicable).

✉️ Sending Emails[¶](https://saasfast.docs.smorchestra.ai/email/#sending-emails)

1. **SMTP**
2. Used for sending magic login links.
3. **Mailgun API**
4. Use sendEmail() in libs/mailgun.js for everything else.

📥 Receiving Emails[¶](https://saasfast.docs.smorchestra.ai/email/#receiving-emails_1)

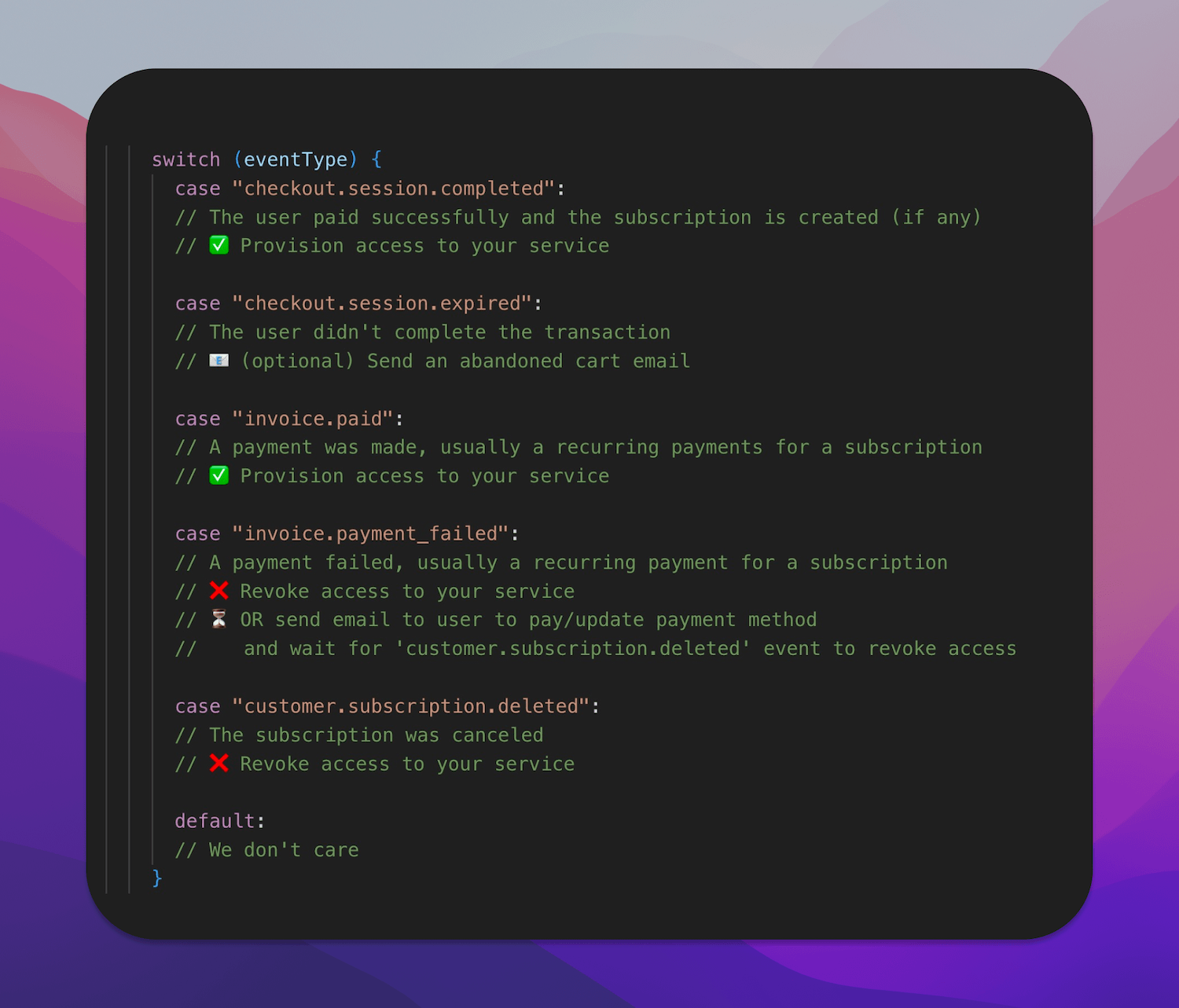
* 🚫 Mailgun does not store or forward emails by default.
* That’s why we created a route to handle emails sent to mailgun.supportEmail from config.js and redirect them to your API.
* 📧 Replies will automatically include the original sender in the reply-to field, so you can reply directly from your inbox.

✅ Spam Prevention Checklist[¶](https://saasfast.docs.smorchestra.ai/email/#spam-prevention-checklist)

📷 *There’s an image here*

* Payment

💳 Stripe Setup[¶](https://saasfast.docs.smorchestra.ai/Payments_strip/#stripe-setup)

[¶](https://saasfast.docs.smorchestra.ai/Payments_strip/#_1)

⚙️ Setup[¶](https://saasfast.docs.smorchestra.ai/Payments_strip/#setup)

1️⃣ **Create a new Stripe account and enable payments**  
- (I know, boring... see you soon!)

2️⃣ **Set your site information**  
- Go to **[Settings]** → **[Public Details]** and add your website URL.

3️⃣ **Brand customization**  
- Go to **[Settings]** → **[Branding]** and add your logo and colors.

4️⃣ **Enable customer email notifications**  
- Under **[Settings]** → **[Customer Emails]**, enable successful payment and refund email notifications.

5️⃣ **Enable the Customer Portal**  
- Under **[Settings]** → **[Customer Portal]**, activate the portal link (for later use).

6️⃣ **Fraud prevention**  
- In the search bar, type **"rules"** and click **[Fraud Prevention > Rules]**  
- ✅ Make sure the first 3DS rule is enabled  
- 💡 It’s also recommended to enable the second rule  
- 🚫 Ensure payments are blocked if CVC check fails

7️⃣ **Enable Test Mode**

8️⃣ **Create a new product**  
- Add a product and copy its **Price ID** (e.g. price\_1ju5GD464564)  
- Add it to config.stripe.plans[0].priceId in your config.js file

9️⃣ **Setup API keys**  
- Go to **[Developers]**, copy your **public and secret keys**  
- Add them to .env.local under STRIPE\_PUBLIC\_KEY and STRIPE\_SECRET\_KEY

🌐 Local Webhook[¶](https://saasfast.docs.smorchestra.ai/Payments_strip/#local-webhook)

🔟 **Setup Webhook locally**  
- Go to **[Developers]** → **[Webhook]** → **[Test in local environment]**  
- After signing in to Stripe CLI, run this command to forward webhook events:

```bash stripe listen --forward-to localhost:3000/api/webhook/stripe ````

* Copy the **Webhook Signing Secret** and add it to STRIPE\_WEBHOOK\_SECRET in .env.local

🚀 Going Live[¶](https://saasfast.docs.smorchestra.ai/Payments_strip/#going-live)

✅ Copy the webhook signing secret and add it to your .env.local as STRIPE\_WEBHOOK\_SECRET

⚙️ Production Setup Steps:[¶](https://saasfast.docs.smorchestra.ai/Payments_strip/#production-setup-steps)

1️⃣ **Turn off Test Mode** in the Stripe dashboard

2️⃣ Under **[Developers]**, copy your keys:

* STRIPE\_SECRET\_KEY
* STRIPE\_PUBLIC\_KEY 📥 Add them to your **production environment variables**

3️⃣ Under **[Developers]** → **[Webhook]**, click **[Add Endpoint]**:

* Set endpoint to: https://yourdomain.com/api/webhook/stripe
* ✅ Select event: checkout.session.completed (add more if needed)
* 🔐 Copy the signing secret and add it to STRIPE\_WEBHOOK\_SECRET in your **production environment**

📅 Optional Configuration[¶](https://saasfast.docs.smorchestra.ai/Payments_strip/#optional-configuration)

4️⃣ Under **[Balance]** → **[Manage Payouts]**, set a fixed payout date (e.g., 10th of every month)

5️⃣ Under **[Settings]** → **[Customer Emails]**, enable payment success and refund emails

💳 Creating a Checkout[¶](https://saasfast.docs.smorchestra.ai/Payments_strip/#creating-a-checkout)

From the frontend, use the <ButtonCheckout /> component to automatically generate a one-time or subscription checkout session.

🔄 Webhook & Subscription Handling[¶](https://saasfast.docs.smorchestra.ai/Payments_strip/#webhook-subscription-handling)

Your API listens to Stripe **webhook events** and updates the user’s access accordingly.

✅ It toggles the value of hasAccess (or has\_access in Supabase) to true or false based on whether the user should have access to paid resources.

💡 You can add your custom logic to the endpoint: /api/webhook/stripe Examples:

* Grant user credits
* Send an email with a paid ebook
* Any other logic (e.g. product unlock, discount code delivery, etc.)
* Google Authentication (OAuth)

⚙️ Setup[¶](https://saasfast.docs.smorchestra.ai/Goole_auth/#setup)

1️⃣ Add the following to your .env.local file (if not already done):[¶](https://saasfast.docs.smorchestra.ai/Goole_auth/#1-add-the-following-to-your-envlocal-file-if-not-already-done)

```bash NEXTAUTH\_URL=http://localhost:3000 NEXTAUTH\_SECRET= ````

NEXTAUTH\_SECRET is a random string used to encrypt JWT tokens. It must be at least **15 characters long**.

2️⃣ Create a new project in Google Cloud[¶](https://saasfast.docs.smorchestra.ai/Goole_auth/#2-create-a-new-project-in-google-cloud)

* Go to **APIs & Services** → **Credentials**
* Click on **[Configure Consent Screen]**
* Fill out the required information *(Tip: Use ChatGPT to generate /tos and /privacy-policy pages)*.
* Add userinfo.email and userinfo.profile to the **scopes**.
* Add yourself as a test user.
* Click **Submit**

3️⃣ Create OAuth credentials[¶](https://saasfast.docs.smorchestra.ai/Goole_auth/#3-create-oauth-credentials)

* Go to **Credentials** → **[+ Create Credentials]** → **[OAuth Client ID]**
* Choose **[Web Application]**
* Add the following to **Authorized JavaScript origins**:
* http://localhost:3000
* https://your-site.com
* And add the following to **Authorized redirect URIs**:
* http://localhost:3000/api/auth/callback/google
* https://your-site.com/api/auth/callback/google

📌 If you’re using a subdomain like www, make sure to include it as well.

* Click **[Create]**
* Copy:
* Client ID into GOOGLE\_ID
* Client Secret into GOOGLE\_SECRET and paste them into your .env.local

4️⃣ Publishing and verification[¶](https://saasfast.docs.smorchestra.ai/Goole_auth/#4-publishing-and-verification)

* Go to **[OAuth Consent Screen]** → click **[Publish App]** and submit your app for verification.
* Click **[Prepare for verification]** and complete the required fields.
* Google will send you an email—you need to respond to begin the process.
* Your **domain must be verified on Google Search Console**. You can do that now.

5️⃣ Test the login[¶](https://saasfast.docs.smorchestra.ai/Goole_auth/#5-test-the-login)

* You can now log in with Google on **localhost**.
* It will also work in production, but a **warning** will appear until verification is complete (can take a few days).

6️⃣ Saving users in your database[¶](https://saasfast.docs.smorchestra.ai/Goole_auth/#6-saving-users-in-your-database)

* The **MongoDB adapter** automatically saves new users when they log in via Google.
* ✅ Make sure your **database is configured first**.
* Magic Links

# **🔐 SaaSFast uses the NextAuth library for user authentication**[**¶**](https://saasfast.docs.smorchestra.ai/MagicLinks/#saasfast-uses-the-nextauth-library-for-user-authentication)

You can configure it in the file:  
/app/api/auth/[...nextauth]/route.js

## **⚙️ Setup**[**¶**](https://saasfast.docs.smorchestra.ai/MagicLinks/#setup)

1️⃣ If you haven’t already, add the following values to your .env.local file:  
💻 There is a code block here

🔑 NEXTAUTH\_SECRET is a random string used to encrypt JWT.  
Use any value that is **at least 15 characters long**.

2️⃣ If you haven’t set up email sending/receiving yet, do that first.

3️⃣ In the config.js file, add a from field to resend.fromNoReply  
📧 (Usually something like noreply@mail.yourdomain.com)

4️⃣ (Skip this if you are using Resend)  
🚫 Add a sending limit in Mailgun to prevent abuse.

5️⃣ The MongoDB adapter saves new users to the database when they click the login link received via email.  
🗃️ Make sure your database is properly configured first.

* Customer Support

⚙️ Setup[¶](https://saasfast.docs.smorchestra.ai/Customersupport/#setup)

1️⃣ Create an account on **Crisp**.

2️⃣ Create a new website and copy the **Website ID**.  
- You can find it under the **Integrations** menu, in the **HTML** section.  
- It's referred to as: CRISP\_WEBSITE\_ID

3️⃣ Paste the **Crisp Website ID** into crisp.id in the config.js file.

* Customer Support

Error Handling

⚠️ Graceful Handling of JavaScript and 404 Errors[¶](https://saasfast.docs.smorchestra.ai/Error_handling/#graceful-handling-of-javascript-and-404-errors)

* JavaScript errors are handled elegantly using the top-level component: /app/error.js
* 404 (Not Found) pages are also managed smoothly via the component: /app/not-found.js

🧩 What Happens When an Error Occurs[¶](https://saasfast.docs.smorchestra.ai/Error_handling/#what-happens-when-an-error-occurs)

* A clean, user-friendly error page is displayed, and users can reach out to support through the <ButtonSupport /> component.
* By default, the support button opens **Crisp live chat** (if crisp.id is configured in the config.js file).

📘 See the guide for setting up **Crisp**.

* If crisp.id is **not** set in config.js, the support button will instead open the user’s **email client** using a mailto: link, so they can contact your support address (resend.supportEmail in config.js).
* Analytics

⚙️ Setup[¶](https://saasfast.docs.smorchestra.ai/Analytics/#setup)

🎥 There’s a video walkthrough available.

1️⃣ Add a new site in **DataFast**.

🎁 As a **SaaSFast** member, you get **30% off** all plans.

2️⃣ Paste the **tracking script** from DataFast into your main layout.js file.

🧠 **DataFast** will track traffic and custom events, helping you identify which marketing channels convert best.

* Components

Components

🎨 All React Components Are Built with TailwindCSS[¶](https://saasfast.docs.smorchestra.ai/Components_tab/#all-react-components-are-built-with-tailwindcss)

🌗 Themes[¶](https://saasfast.docs.smorchestra.ai/Components_tab/#themes)

By default, **light and dark themes** are enabled. The system will automatically switch based on **user preferences**.

**To change or add themes:**[**¶**](https://saasfast.docs.smorchestra.ai/Components_tab/#to-change-or-add-themes)

1️⃣ Add or remove themes in the tailwind.config.js file under daisyui.themes 📌 *(Make sure to keep at least one theme)*

2️⃣ Add the **theme name** in the config.js file under both theme and main:

* **Path:** config.js

colors: {

theme: "YOUR\_THEME",

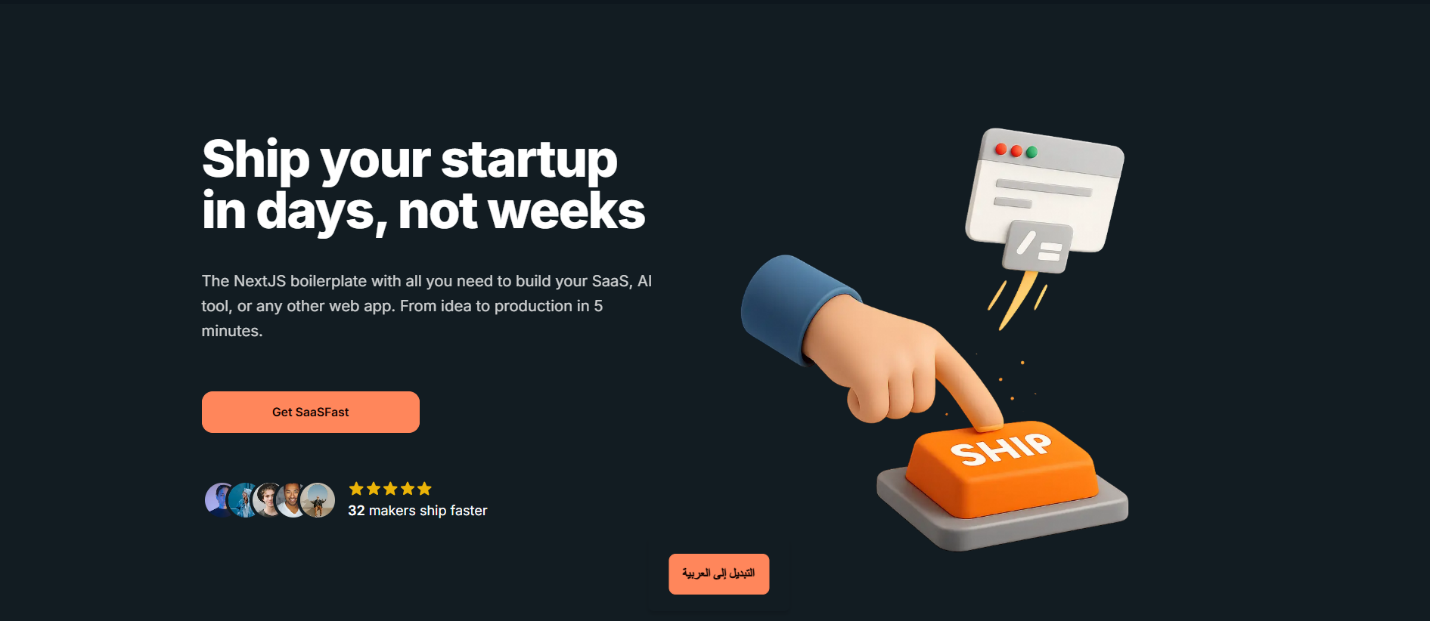
main: themes[`[data-theme=YOUR\_THEME]`]["primary"],

}

🗂️ Here’s a list of over 20 available themes

🧩 Custom Components[¶](https://saasfast.docs.smorchestra.ai/Components_tab/#custom-components)

All custom components are located inside the /components folder in your **SaaSFast** project. Each component is **fully documented** in the code and this guide.

📸 Example: Here's the **Hero** component 

🧱 daisyUI Components[¶](https://saasfast.docs.smorchestra.ai/Components_tab/#daisyui-components)

**SaaSFast** uses **daisyUI** as its UI kit, which includes simple components like **buttons, inputs, tabs**, and more. 🔗 Explore all daisyUI components

**Example:**[**¶**](https://saasfast.docs.smorchestra.ai/Components_tab/#example)

Demo

To render this button, use:

className: "btn btn-primary"

🎞️ Animations[¶](https://saasfast.docs.smorchestra.ai/Components_tab/#animations)

You can explore some **built-in animations** in the config.tailwind.css file:

className: "animate-opacity" className: "animate-wiggle" className: "animate-appearFromRight" className: "animate-popup"

🔤 Custom Fonts[¶](https://saasfast.docs.smorchestra.ai/Components_tab/#custom-fonts)

1️⃣ Go to the Layout.js file in the /components folder

2️⃣ Import your preferred font from next/font/google:

import { Bricolage\_Grotesque } from "next/font/google";

3️⃣ Replace the current font with your custom one:

const font = Bricolage\_Grotesque({ subsets: ["latin"] });

* Header

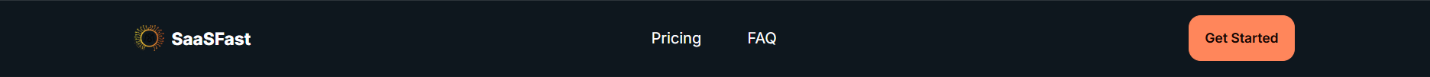
🧱 Responsive Header[¶](https://saasfast.docs.smorchestra.ai/Components_header/#responsive-header)

The header includes your **logo (on the left)**, **navigation links (centered)**, and a **Call-to-Action (CTA) button (on the right)**.

📱 **On mobile devices**, the links and CTA are **hidden** and accessed through a **burger menu**.

🖼️ Add your **custom logo** inside the /public folder. Make sure the **file name matches** the one used in the <Header /> component (currently logo.png).

* **Path:** page.js



// For both Arabic and English

import Header from "@/components/Header";

// For English only

import Header\_en from "@/components/Header\_en";

// For Arabic only

import Header\_ar from "@/components/Header\_ar";

🌐 Language Support[¶](https://saasfast.docs.smorchestra.ai/Components_header/#language-support)

There are 3 header components you can toggle between:

* <Header /> ➜ The default header that supports both languages.
* <Header\_ar /> ➜ A right-to-left (RTL) header for Arabic only.
* <Header\_en /> ➜ A left-to-right (LTR) header for English only.

🔁 You can switch between these headers based on the selected language through settings or the config.js file.

💡 Tips[¶](https://saasfast.docs.smorchestra.ai/Components_header/#tips)

* Unless you're **Nike** or **Apple**, people probably don’t know who you are. Keep your **brand name subtle and clean**.
* Always include a **"Pricing"** link in the header. No matter what you’re offering, **visitors will look for it**.
* Hero Section

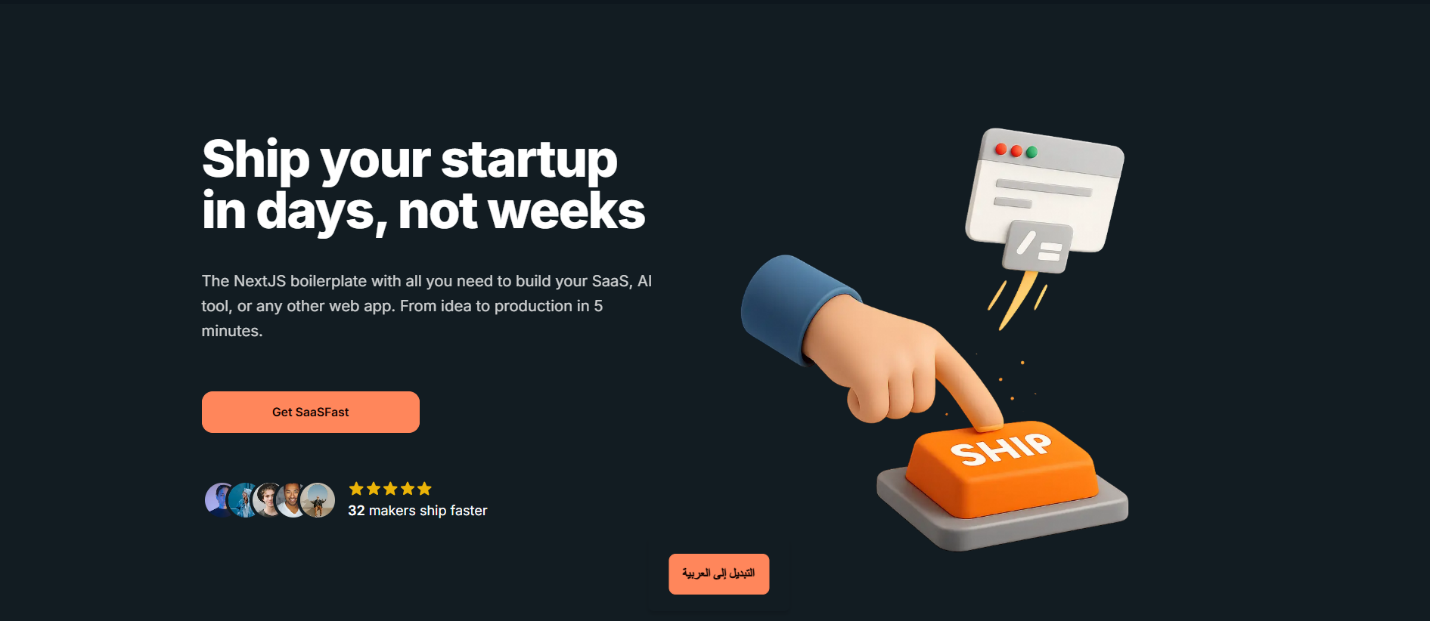
🎯 Beautiful Hero Section[¶](https://saasfast.docs.smorchestra.ai/hero_component/#beautiful-hero-section)

Includes:

* Main headline
* Supporting subheadline
* Image
* Social proof
* Call-to-Action (CTA) button

🗂️ There are two versions of this component:

* <Heroar /> ➜ Arabic version
* <Hero /> ➜ English version

📸 English version example: 

📸 Arabic version example: 

* **Path:** page.js

// English version

import Hero from "@/components/Hero";

// Arabic version

import Hero from "@/components/Heroar";

💡 Tips[¶](https://saasfast.docs.smorchestra.ai/hero_component/#tips)

* Your <h1> should answer this question in under 10 words: **"Why should a random visitor stay on your site for more than 10 seconds?"**
* Focus on the pain you relieve, the problem you solve, or the joy you bring.
* The supporting headline should explain how you deliver what the main headline promises.
* Your CTA button should start with an action verb *(e.g., Get, Discover, Learn, etc.).*
* **Social proof = trust = more conversions.**
* Offer your product to a few users for free in exchange for **honest testimonials**.
* The image should work like a **YouTube thumbnail** — it should be **crystal clear what your product offers at a glance**.
* 📋 Features List

This section includes:

* An icon
* A title
* A description for each feature

🖱️ Clicking on a feature will **reveal its description**.

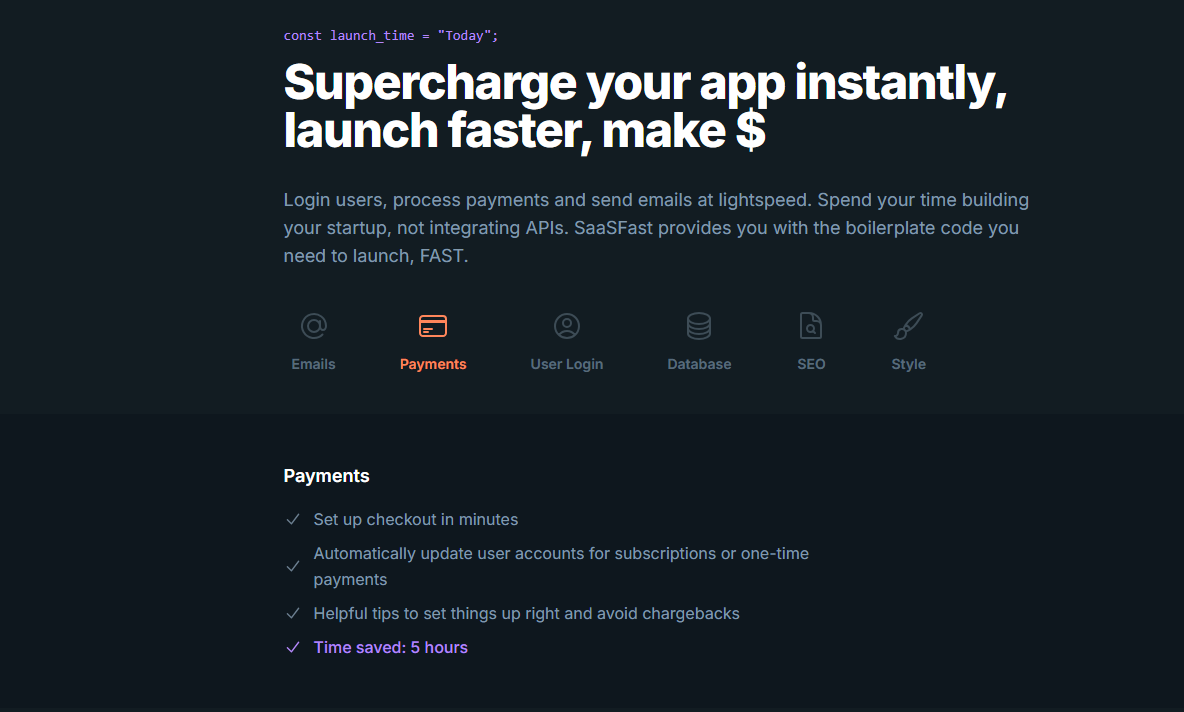
🔄 Features can **automatically rotate every 5 seconds** (optional).

🧩 This section is especially useful when showcasing **multiple features** in your app.

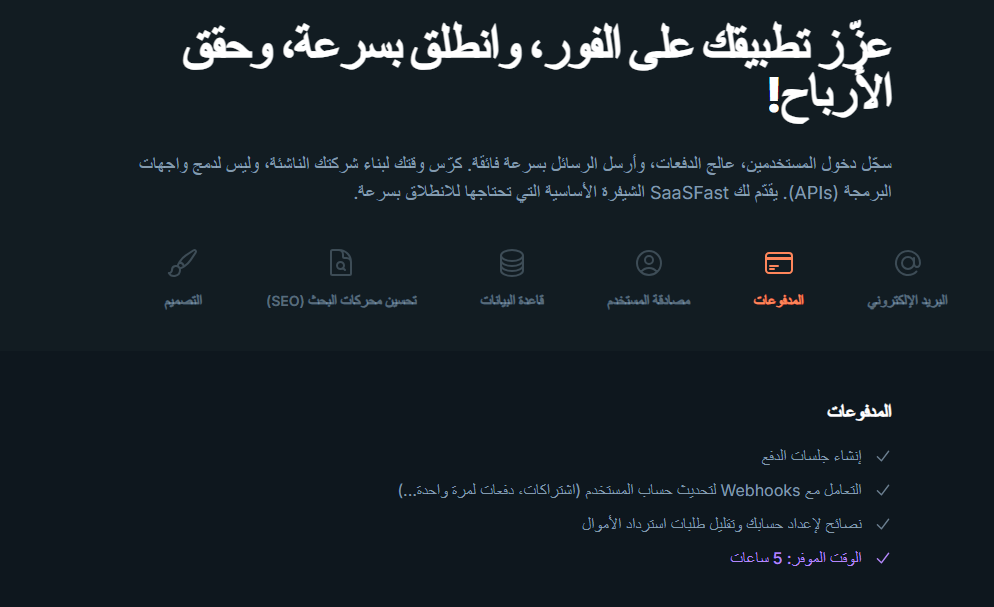
There are two components you can use based on the language:

* <FeaturesListicle /> ➜ English version
* <FeaturesListicle\_ar /> ➜ Arabic version

➜ English Version[¶](https://saasfast.docs.smorchestra.ai/component_FeaturesListicle/#english-version)



➜ Arabic Version[¶](https://saasfast.docs.smorchestra.ai/component_FeaturesListicle/#arabic-version)



* **Path:** page.js

// English version

import FeaturesListicle from "@/components/FeaturesListicle";

// Arabic version

import FeaturesListicle\_ar from "@/components/FeaturesListicle\_ar";

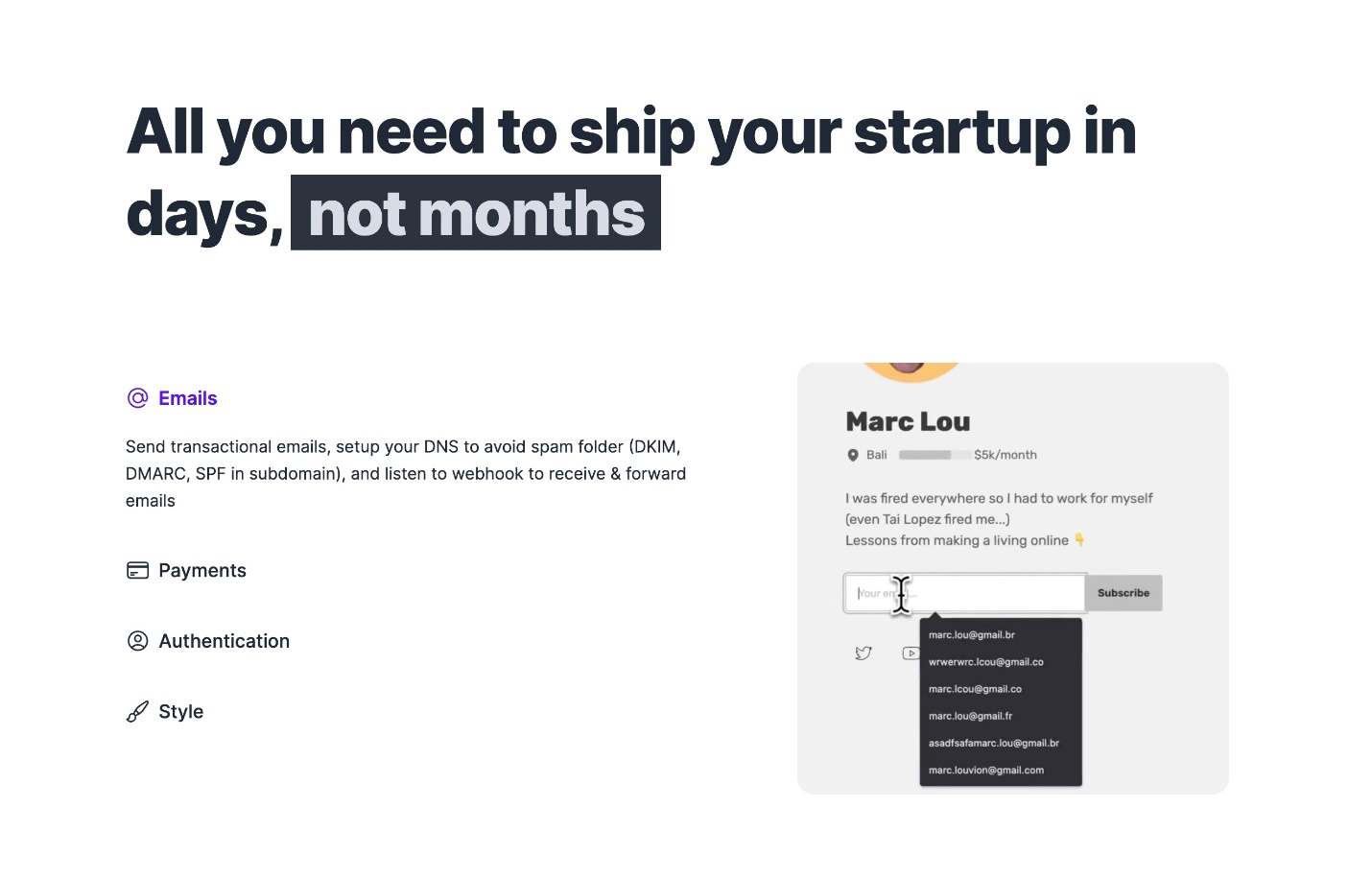
💡 Tips[¶](https://saasfast.docs.smorchestra.ai/component_FeaturesListicle/#tips)

* The **<h2> headline** should remind visitors of your product’s **core value**. **Why do they need it?**
* The **supporting headline** should explain **how** your product delivers what the main headline promises.
* Features Accordion

A component to display 2 to 5 features in an accordion. By default, the **first feature is selected**. When a feature is clicked, the others are closed.

🖱️ The **text is clickable** and displays the feature description when clicked. 🎥 The media could be a **video**, an **image**, or **nothing**. Videos are set to **autoplay** for the best UX.

🔧 A Features Accordion component for SaaSFast boilerplate[¶](https://saasfast.docs.smorchestra.ai/FeaturesAccordion/#a-features-accordion-component-for-saasfast-boilerplate)



**📄 File:** page.js

import FeaturesAccordion from "@/components/FeaturesAccordion";

💡 Tips[¶](https://saasfast.docs.smorchestra.ai/FeaturesAccordion/#tips)

* Keep the **title short and simple** *(less than 5 words)*.
* WithWithout

🔍 This section explains how your product is better than competitors[¶](https://saasfast.docs.smorchestra.ai/component_WithWithout/#this-section-explains-how-your-product-is-better-than-competitors)

It is useful if you are in a competitive market and challenge the status quo.  
Make sure to mention the name of your competitors for customers to understand the context.

[¶](https://saasfast.docs.smorchestra.ai/component_WithWithout/#_1)

🧩 A WithWithout component for SaaSFast boilerplate[¶](https://saasfast.docs.smorchestra.ai/component_WithWithout/#a-withwithout-component-for-saasfast-boilerplate)

**page.js**

import WithWithout from "@/components/WithWithout";

💡 Tips[¶](https://saasfast.docs.smorchestra.ai/component_WithWithout/#tips)

* Match each line of the **'With'** column with the **'Without'** column
* Call To Action ( CTA)

**Full-Screen CTA Section**:

📣 Full-Screen CTA Section[¶](https://saasfast.docs.smorchestra.ai/component_CTA/#full-screen-cta-section)

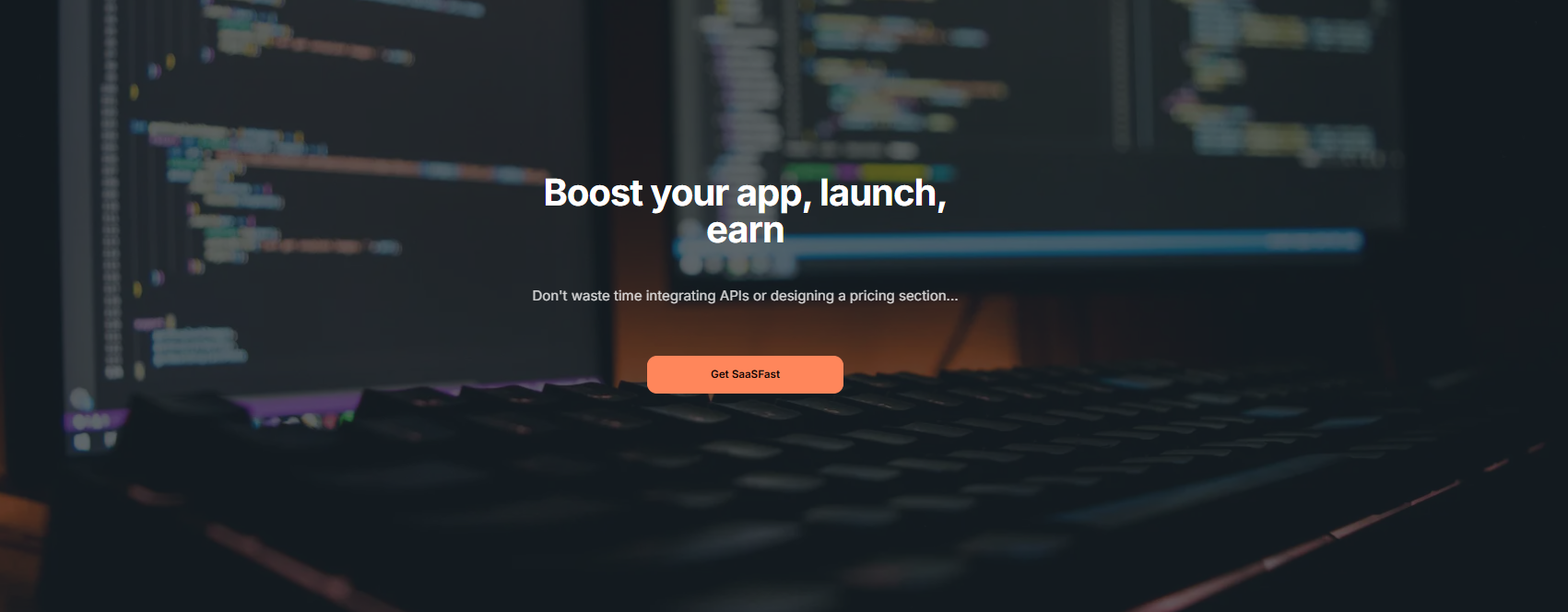
Includes:

* A main headline
* A supporting subheadline
* A background image
* A call-to-action (CTA) button

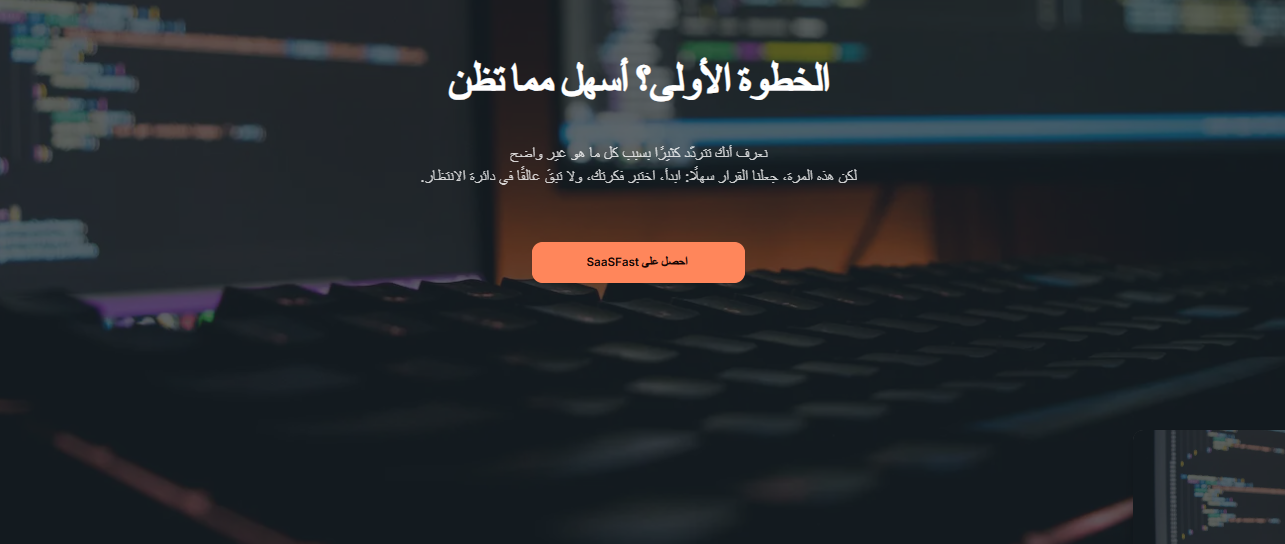
🧩 There are two components depending on the language:

* <CTA /> ➜ English version
* <CTA\_ar /> ➜ Arabic version

➜ English Version[¶](https://saasfast.docs.smorchestra.ai/component_CTA/#english-version)



➜ Arabic Version[¶](https://saasfast.docs.smorchestra.ai/component_CTA/#arabic-version)



* **Path:** page.js

// English version

import CTA from "@/components/CTA\_ar";

// Arabic version

import FeaturesListicle\_ar from "@/components/FeaturesListicle\_ar";

💡 Tips[¶](https://saasfast.docs.smorchestra.ai/component_CTA/#tips)

* The **headline should highlight the core value** of your product or service.
* Ask yourself: **Why should a random visitor sign up?**
* **What pain does your product help solve?**
* Your **CTA button** should always start with a verb *(e.g., Get, Discover, Learn, etc.)*
* **Pricing Section**:

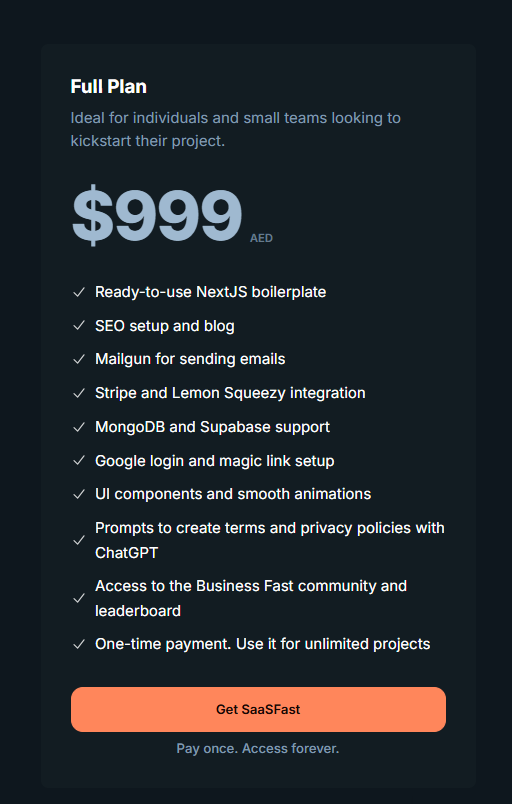
Includes the ability to **highlight a specific plan**.

🧩 There are two components available depending on the selected language:

* <Pricing /> ➜ Arabic version
* <Pricing\_en /> ➜ English version

📂 To add a new plan, simply **add it to the config.js file** — it will automatically appear in the pricing section.

➜ English Version[¶](https://saasfast.docs.smorchestra.ai/component_Pricing/#english-version)



➜ Arabic Version[¶](https://saasfast.docs.smorchestra.ai/component_Pricing/#arabic-version)



// English

import Pricing from "@/components/Pricing";

// Arabic

import Pricing\_en from "@/components/Pricing\_en";

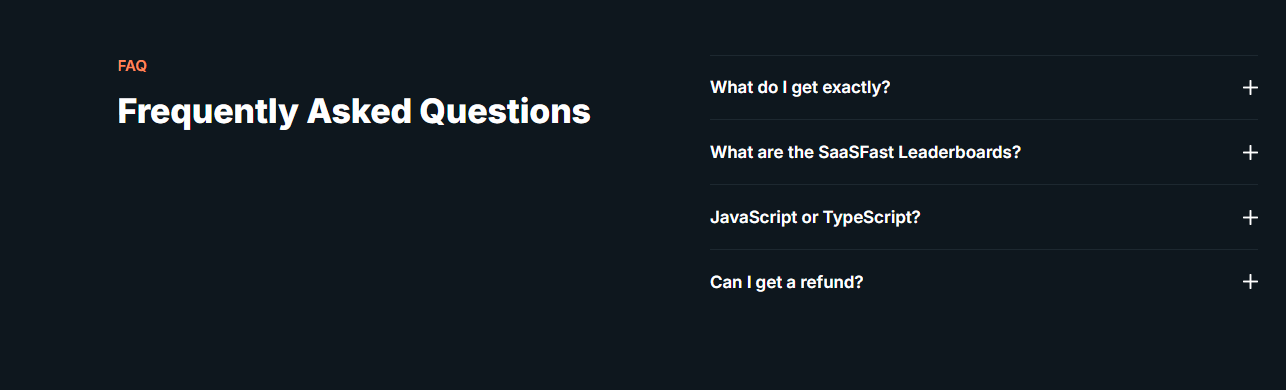
* FAQ Section

A list of **questions and answers** to address the most frequently asked queries.

🧩 Two components are available depending on the language:

* <FAQ /> # ➜ English
* <FAQ\_ar /> # ➜ Arabic

➜ English Version[¶](https://saasfast.docs.smorchestra.ai/component_FAQ/#english-version)



➜ Arabic Version[¶](https://saasfast.docs.smorchestra.ai/component_FAQ/#arabic-version)



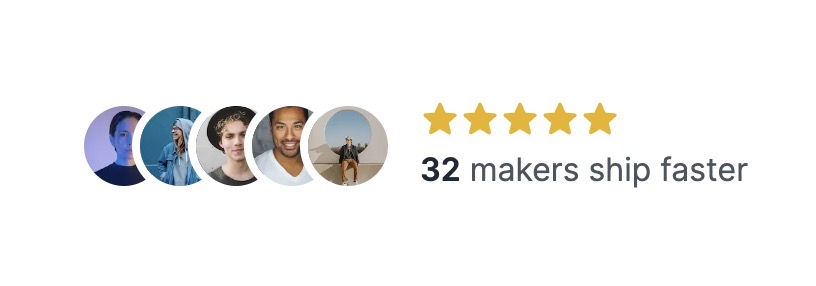
💡 Tips[¶](https://saasfast.docs.smorchestra.ai/component_FAQ/#tips)

* Try to address **any potential objections** your visitors might have.
* For example, if you're selling a **course**, consider **adding a question about your refund policy**.
* Testimonial Small

🌟 Compact Testimonial Display[¶](https://saasfast.docs.smorchestra.ai/componen_TestimonialSmall/#compact-testimonial-display)

A compact testimonial section that includes:

* 5 customer photos
* Along with a **rating**



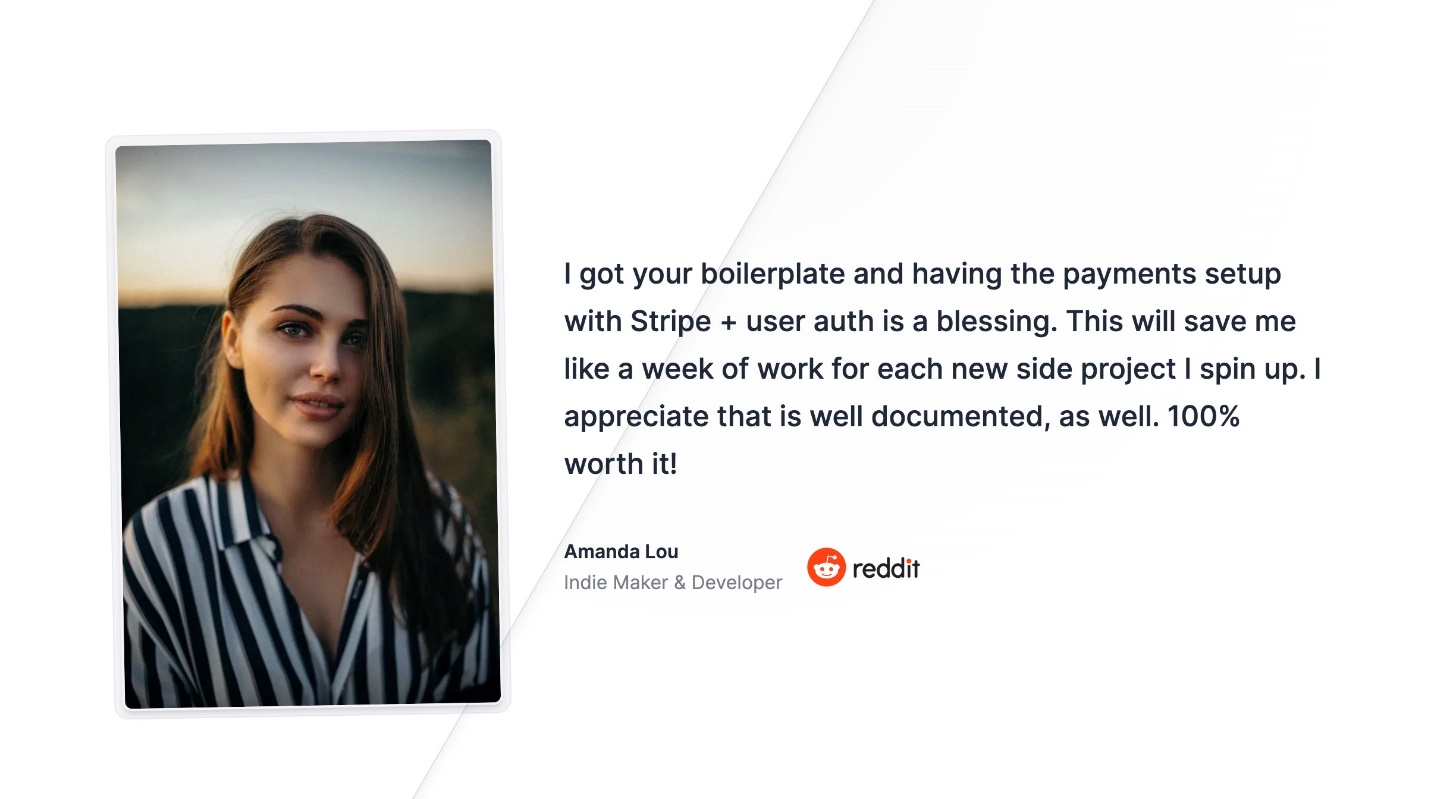
--------------------------------------------------------------------------------------------

* Testimonial Single

💬 Testimonial Single[¶](https://saasfast.docs.smorchestra.ai/TestimonialSingle/#testimonial-single)

A beautiful & detailed testimonial to show love from your customer.

🧪 A Testimonial Single component for SaaSFast boilerplate[¶](https://saasfast.docs.smorchestra.ai/TestimonialSingle/#a-testimonial-single-component-for-saasfast-boilerplate)



**📄 File:** page.js

import Testimonials1 from "@/components/Testimonials1";

💡 Tips[¶](https://saasfast.docs.smorchestra.ai/TestimonialSingle/#tips)

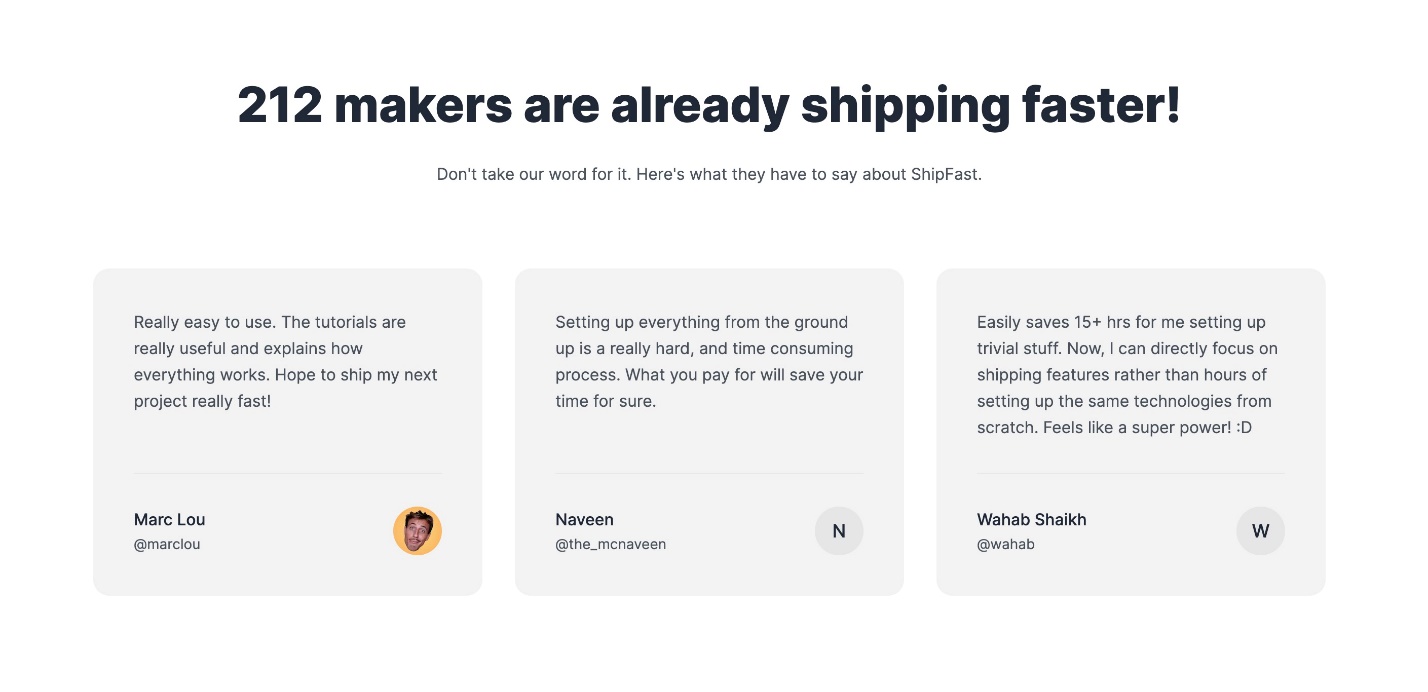
* Don't ask users *if* they like your product. Instead, ask **how it changed in their lives** after using it.
* Highlight the **most important part** of the testimonial to avoid clutter.

---------------------------------------------------------------------------------------------------

* Testimonial Triple

**3 simple testimonials** to show love from your customers.

🧪 A Testimonial Triple component for SaaSFast boilerplate[¶](https://saasfast.docs.smorchestra.ai/component_Testimonial_Triple/#a-testimonial-triple-component-for-saasfast-boilerplate)



**📄 File:** page.js

import Testimonials3 from "@/components/Testimonials3";

💡 Tips[¶](https://saasfast.docs.smorchestra.ai/component_Testimonial_Triple/#tips)

* Don't ask users *if* they like your product. Instead, ask **how it changed their lives** after using it.
* Highlight the **most important part** of the testimonial to avoid clutter.
* Testimonial Grid

🧾 Simple Testimonial Grid[¶](https://saasfast.docs.smorchestra.ai/componen_TestimonialGrid/#simple-testimonial-grid)

11 simple testimonials that reflect genuine appreciation and support from your customers.

📸 Under this, there's an image and code example.

💡 Tips[¶](https://saasfast.docs.smorchestra.ai/componen_TestimonialGrid/#tips)

* Don’t ask users *if* they liked your product — ask them *how* it changed their life after using it.
* Focus on the most impactful part of the testimonial to avoid clutter.
* Footer

🦶 Simple Footer[¶](https://saasfast.docs.smorchestra.ai/componen_Footer/#simple-footer)

A minimal footer that includes:

* **Logo**
* **Slogan**
* **Links**

🧩 There are 3 components you can use based on language:

* <Footer /> ➜ Footer that supports **both Arabic and English**
* <Footer\_ar /> ➜ Footer for **Arabic only**
* <Footer\_en /> ➜ Footer for **English only**

📸 

// For both Arabic and English

import Footer from "@/components/Footer";

// English only

import Footer\_en from "@/components/Footer\_en";

// Arabic only

import Footer\_ar from "@/components/Footer\_ar";

* Button Lead

✉️ Email Collection[¶](https://saasfast.docs.smorchestra.ai/componen_ButtonLead/#email-collection)

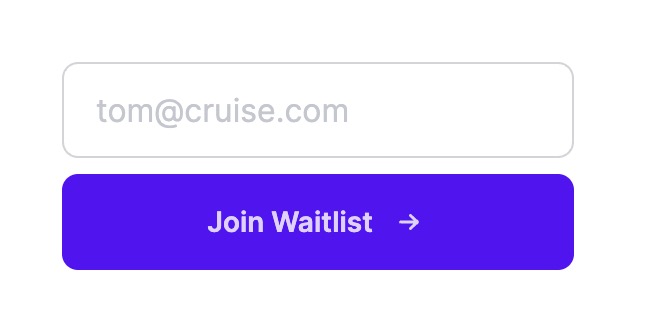
Collect email addresses and automatically store them in your database.

✅ Perfect for:

* A **waitlist** (if your product isn’t ready yet)
* Or a **popup form** for lead generation

🔁 It triggers the API route: /api/lead/route.js to save the lead in the database.

📦 This process **requires a database**.



import ButtonLead from "@/components/ButtonLead";

* Button Checkout

Here is a smooth and natural English translation of the section:

🧾 Start a Stripe Checkout Session[¶](https://saasfast.docs.smorchestra.ai/componen_ButtonCheckout/#start-a-stripe-checkout-session)

Launch a **Stripe Checkout** session — Perfect for **one-time payments** or **recurring subscriptions**.

⚙️ Setup[¶](https://saasfast.docs.smorchestra.ai/componen_ButtonCheckout/#setup)

* Use mode="payment" for one-time payments
* Or mode="subscription" for recurring payments like a **monthly plan**

🔐 Security[¶](https://saasfast.docs.smorchestra.ai/componen_ButtonCheckout/#security)

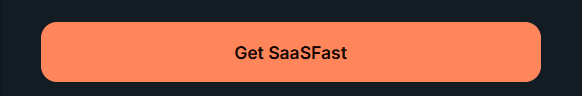
By default, users must **be logged in to complete the payment** to help prevent issues like fraudulent payments with fake emails.

📂 You can change this behavior in the API route: /api/stripe/create-checkout/route.js

🧩 Usage[¶](https://saasfast.docs.smorchestra.ai/componen_ButtonCheckout/#usage)

This component is used to handle:

* **Stripe subscriptions**
* **One-time payments**

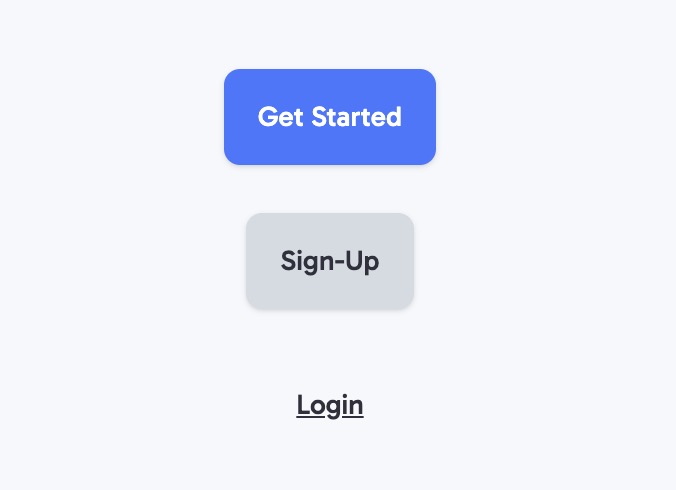


* Button Sign-in

🔐 Simple Sign-In / Sign-Up Button[¶](https://saasfast.docs.smorchestra.ai/componen_ButtonSign_in/#simple-sign-in-sign-up-button)

A simple button that lets users **sign in or create an account** using our supported providers (**Google** and **Magic Links**).

🔁 After signing in, users are **automatically redirected** to the callbackUrl defined in config.auth.callbackUrl — usually a **private dashboard or account page**.



import ButtonSignin from "@/components/ButtonSignin";

Button Gradient

🎨 Beautiful Animated Gradient Button[¶](https://saasfast.docs.smorchestra.ai/componen_ButtonGradient/#beautiful-animated-gradient-button)

A visually appealing button with an **animated gradient background**. Perfect for **AI tools**!



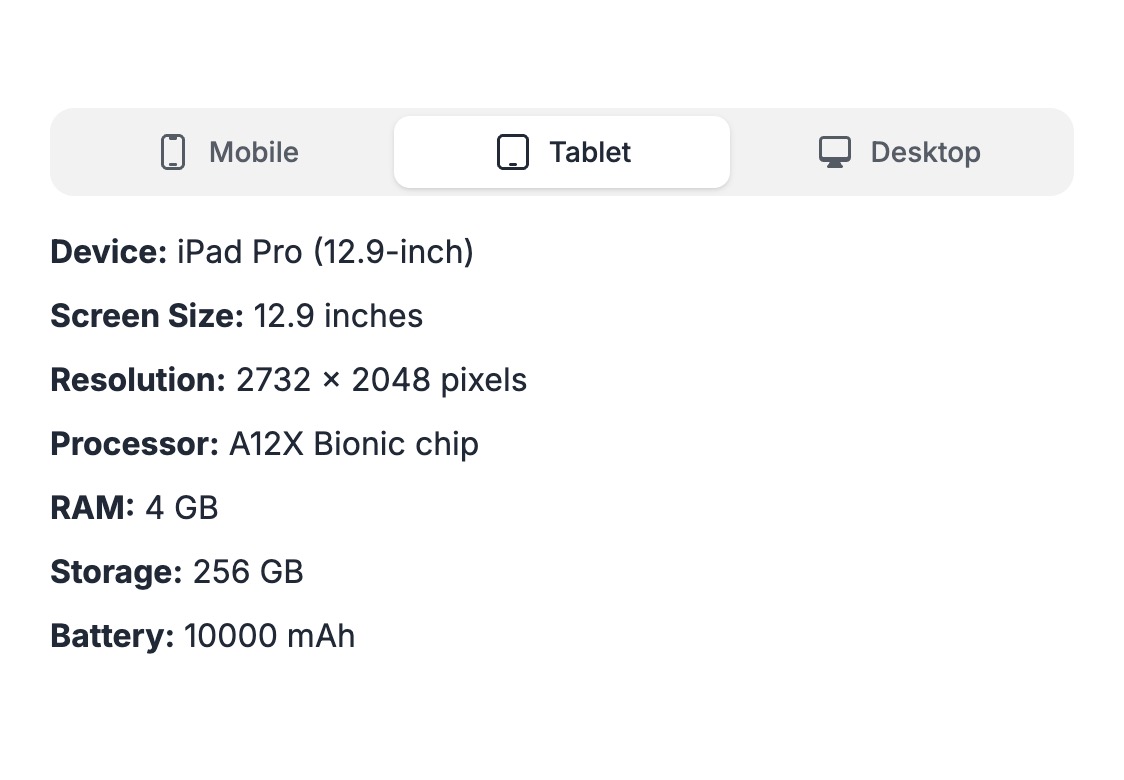
import ButtonGradient from "@/components/ButtonGradient";

Tabs

🧩 Components: Tabs[¶](https://saasfast.docs.smorchestra.ai/component_taps/#components-tabs)

A simple way to organize content into separate sections.

🧪 A Tabs component for ShipFast boilerplate[¶](https://saasfast.docs.smorchestra.ai/component_taps/#a-tabs-component-for-shipfast-boilerplate)



**📄 File:** page.js

import Tabs from "@/components/Tabs";

Modal

🪟 Animated Modal An animated popup window — perfect for:

Product explainers

Lead generation

Forms

And more

* **Path:** page.js
* import Modal from "@/components/Modal";
* export default function Component() {
* const [isModalOpen, setIsModalOpen] = useState(false);
* return (
* <Modal isOpen={isModalOpen} setIsOpen={setIsModalOpen} />
* );

}

* Deployment

⚙️ Deployment Setup

🚀 Time to Launch!

* **Commit your code** and push it to a **private GitHub repository**.
* Use your **preferred hosting provider** (I recommend Vercel) to create a new project based on that GitHub repo.
* You can deploy your app anywhere that supports **Next.js** like: **Vercel, Netlify, Render**, etc.
* Make sure to set up the correct **.env file for your deployment environment**, since .env.local isn’t pushed with the code.
* Ensure that **NEXTAUTH\_URL matches your domain name** so that authentication features work properly.
* Change **NEXTAUTH\_SECRET** to a secure value (minimum 15 characters).
* Using Stripe? Follow the **production checklist** (scroll down the page for it).
* *(Optional)* Track your traffic and identify your best marketing channels using **DataFast**. As a **SaaSFast** member, you get **30% off all plans** ❤️
* Extras

Setup[¶](https://saasfast.docs.smorchestra.ai/Extras/#setup)

* Add your own **logo** inside the /app folder and name it icon.png.
* Not using the .png extension? Update any components using icon.png, like the **Header** component.

🧠 Creating Icons[¶](https://saasfast.docs.smorchestra.ai/Extras/#creating-icons)

* Generate your **site favicon** using the [free Favicon Generator](https://favicon.io/). After downloading the zip file, add the following two files to the /app folder:
* apple-touch-icon.png → rename to apple-icon.png
* favicon.ico

🧠 **Next.js** will automatically include these in the <head /> tag of your HTML page. 📚 Learn more about [metadata file conventions](https://nextjs.org/docs/app/building-your-application/optimizing/metadata#file-conventions).

🖼️ Logo and Name[¶](https://saasfast.docs.smorchestra.ai/Extras/#logo-and-name)

* Create a **rectangle logo that includes your name**, similar to the example image below.
* Name it logoAndName.png and place it in the /public folder. 🔐 It will be used automatically on login pages by **NextAuth** (see the /libs/next-auth.js file).

📸 Social Media Sharing Images[¶](https://saasfast.docs.smorchestra.ai/Extras/#social-media-sharing-images)

* Create **two images sized 1200×660** for social sharing:
* opengraph-image.png
* twitter-image.png
* Add them to the /app folder. 🔁 **Next.js** will automatically reference them in the <head />.