**Paso 0: Preparación del Proyecto (Build)**

Tu index.html intenta cargar directamente un archivo index.tsx:  
<script type="module" src="/index.tsx"></script>  
Además, también referencia un <script type="module" src="./bundle.js" defer></script>.

Los navegadores no pueden ejecutar archivos .tsx (TypeScript con JSX) directamente. Necesitan ser convertidos (transpilados y empaquetados) a JavaScript. El archivo bundle.js es probablemente el resultado de este proceso.

* **Asegúrate de tener un proceso de "build":**
  + Si tienes un archivo package.json con un script como "build": "comando\_para\_construir\_tu\_app", ese es el comando que se usará. Por ejemplo, podría ser vite build, react-scripts build, o si usas esbuild podría ser algo como esbuild index.tsx --bundle --outfile=bundle.js --jsx=react.
  + Este proceso de build debería generar el bundle.js final y cualquier otro archivo necesario en una carpeta de salida (comúnmente llamada dist, build, o a veces la raíz del proyecto si los archivos se generan ahí).
  + **Importante:** Para producción, tu index.html debería cargar el bundle.js (o como se llame tu archivo JavaScript final) y no el index.tsx directamente. Si tu proceso de build no actualiza el index.html automáticamente, puede que necesites un index.html específico para producción. Sin embargo, Vercel/Netlify suelen manejar esto bien si detectan un framework conocido o si tu script de build lo gestiona.

**Paso 1: Sube tu Código a un Repositorio Git**

Si aún no lo has hecho, crea un repositorio en GitHub (o GitLab, Bitbucket) y sube tu código.

git init

git add .

git commit -m "Initial commit"

git remote add origin TU\_URL\_DE\_REPOSITORIO\_GIT

git push -u origin main # o master

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487).Bash

**Paso 2: Regístrate en la Plataforma de Hosting**

* Ve a [Vercel](https://www.google.com/url?sa=E&q=https%3A%2F%2Fvercel.com) o [Netlify](https://www.google.com/url?sa=E&q=https%3A%2F%2Fnetlify.com).
* Regístrate, usualmente la opción más fácil es "Continue with GitHub" (o tu proveedor Git).

**Paso 3: Importa/Conecta tu Proyecto Git**

* Una vez dentro de tu dashboard de Vercel/Netlify, busca una opción como "Import Project", "New Project from Git", o "Add new site".
* Selecciona tu proveedor Git y autoriza el acceso.
* Elige el repositorio de tu proyecto Fashion Flex.

**Paso 4: Configura los Ajustes del Proyecto/Build**

La plataforma intentará autodetectar tu framework.

* **Framework Preset:** Puede que lo detecte como React, o puede que necesites elegir "Other" o una opción genérica si tu estructura no es estándar.
* **Build Command:**
  + Aquí es donde pones el comando que construye tu aplicación (ej: npm run build, yarn build, o el comando de esbuild si es lo que usas).
  + Si tu index.html y bundle.js (ya generado) están en la raíz y no necesitas un paso de build en el servidor de hosting (porque bundle.js ya está listo y correcto en tu repo), podrías dejar esto vacío o poner un comando simple como echo "No build required". Sin embargo, es más común que el build se ejecute en el servidor de hosting.
* **Output Directory / Publish Directory:**
  + Es la carpeta donde tu Build Command guarda los archivos listos para producción (ej: dist, build, o ./ si los archivos están en la raíz).
  + Vercel/Netlify servirán el index.html desde esta carpeta.
* **Variables de Entorno (¡MUY IMPORTANTE!):**
  + Aquí es donde configurarás tu API\_KEY para Gemini.
  + Busca la sección de "Environment Variables".
  + Añade una variable con el nombre API\_KEY y pega tu clave secreta como valor.
  + **NUNCA subas tu API\_KEY directamente en el código o en un archivo .env a tu repositorio Git público.** El hosting la inyectará de forma segura como process.env.API\_KEY.

**Paso 5: Desplegar**

* Haz clic en el botón "Deploy".
* La plataforma clonará tu repositorio, ejecutará el comando de build (si lo especificaste), y desplegará los archivos del directorio de salida.
* Te dará una URL como nombre-proyecto.vercel.app o nombre-proyecto.netlify.app. ¡Visítala para ver tu app en línea!

**Paso 6: Pruebas**

* Navega por tu sitio, prueba todas las funcionalidades, especialmente las que usan la Gemini API para asegurar que la API\_KEY está funcionando.
* Revisa la consola del navegador por errores.

**Paso 7: Conectar tu Dominio Personalizado (Cuando estés listo)**

* **Compra tu dominio:** Si aún no lo has hecho, compra thefashionflex.com en un registrador como Namecheap, Porkbun, Google Domains, etc.
* **En Vercel/Netlify:** Ve a la configuración de tu proyecto desplegado y busca la sección "Domains".
* **Añade tu dominio:** Ingresa thefashionflex.com.
* **Configura DNS:** La plataforma te dará instrucciones sobre qué registros DNS necesitas añadir o modificar en tu registrador de dominios. Usualmente son registros A, CNAME o ALIAS. Esto puede tardar un poco en propagarse (desde minutos hasta unas horas).

**Resumen Corto:**

1. **Código en Git.**
2. **Asegúrate de que tienes un script o comando para "construir" tu app** (convertir index.tsx a JS, generar bundle.js en una carpeta de salida).
3. **Regístrate en Vercel/Netlify,** conecta tu repo Git.
4. **Configura el comando de build, directorio de salida, y la variable de entorno API\_KEY.**
5. **Despliega.** ¡Prueba con la URL gratuita!
6. **Cuando quieras, compra thefashionflex.com y conéctalo** siguiendo las instrucciones de Vercel/Netlify.