Vipcomputers Pc builder website

**Definitief Plan: PC Builder met Flask + React zonder Accounts**

We bouwen een **applicatie** waarmee gebruikers een PC kunnen samenstellen en vervolgens **een bestelling kunnen plaatsen of een offerte kunnen aanvragen**.  
📧 **Beide acties sturen een e-mail naar de klant en worden opgeslagen in de backend voor beheer.**

**1. Functies van de Website**

✅ **PC Samenstellen** → Gebruikers kiezen onderdelen per categorie.  
✅ **Prijsberekening** → Live bijwerken van de totaalprijs.  
✅ **Bestellen of Offerte Aanvragen** →

* **Bestellen:** Klant ontvangt een bevestiging via e-mail.
* **Offerte aanvragen:** Klant ontvangt een offerte via e-mail.  
  ✅ **Backend Beheer van Offertes** → Offertes worden opgeslagen en kunnen bekeken worden, **maar niet aangepast of verwijderd**.  
  ✅ **Beheer van Onderdelen** → Admins kunnen **gemakkelijk nieuwe onderdelen toevoegen of verwijderen**.

**2. Architectuur en Technologieën**

**Frontend**

* **PC Builder UI** met categorieën en onderdelen.
* **Knoppen voor bestellen en offerte aanvragen**.
* **Beheerpagina voor onderdelen** (Toevoegen/verwijderen van onderdelen).
* **Beheerpagina voor offertes** (Bekijken van aanvragen).

**Backend**

* **Beheert onderdelen en offertes in een database**.
* **API-routes voor het ophalen, toevoegen en verwijderen van onderdelen**.
* **API-route voor het bekijken van offertes (maar niet aanpassen of verwijderen)**.
* **E-mailfunctionaliteit** voor bestellingen en offertes (Flask-Mail of SendGrid).

**Database (PostgreSQL of MongoDB)**

* **Opslag van onderdelen** (naam, categorie, prijs).
* **Opslag van offertes** (klantgegevens, gekozen onderdelen, prijs, aanvraagdatum).

**3. Stappenplan**

**🔹 Stap 1: Backend (Flask)**

1. **Flask installeren & configureren**.
2. **Database opzetten** met onderdelen en offertes.
3. **API-routes maken** voor:
   * Onderdelen ophalen, toevoegen en verwijderen.
   * Offertes opslaan en bekijken.
4. **E-mailfunctionaliteit toevoegen** voor bestellingen en offertes.

**🔹 Stap 2: Frontend (React)**

1. **PC Builder UI maken** met categorieën en prijsberekening.
2. **Bestel- en offerteknoppen toevoegen**.
3. **Beheerpagina maken voor onderdelen (toevoegen/verwijderen)**.
4. **Beheerpagina maken voor offertes (alleen bekijken)**.
5. **API-koppeling met Flask backend maken**.

**4. Backend Functionaliteiten**

1. **Onderdelen Beheer (Admin)**
   * **Toevoegen van een nieuw onderdeel** via een eenvoudige API-call.
   * **Verwijderen van een onderdeel** als deze niet meer beschikbaar is.
2. **Offerte- en Bestelbeheer**
   * **Nieuwe offerte/bestelling opslaan in de database**.
   * **E-mail naar de klant sturen met de offerte of bestelling**.
   * **Offertes kunnen in een aparte admin-pagina worden bekeken (niet verwijderd of aangepast).**

**5. Extra Zaken om Over Na te Denken**

* **Welke e-mailservice?** Flask-Mail (lokaal SMTP) of SendGrid/Mailgun (betrouwbaarder)?
* **Welke database?** PostgreSQL (beter voor gestructureerde data) of MongoDB (flexibeler voor variabele data)?
* **Hoe ziet het adminpaneel eruit?** Simpel webpaneel in React of via een API-dashboard zoals Postman?