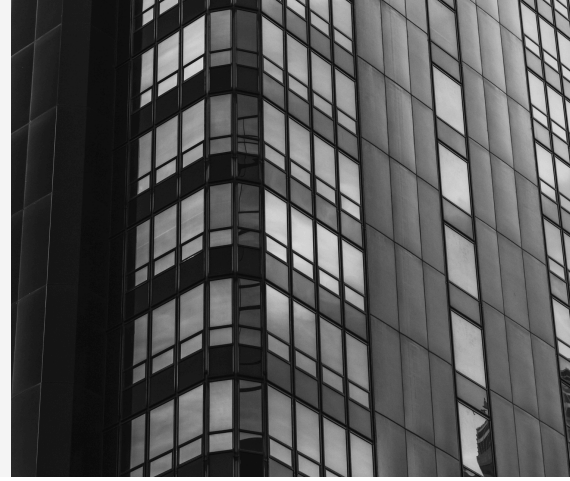


Phinaple Tree



Prepared for: II3160 Teknologi Sistem Terintegrasi Tugas Integrasi Layanan

Prepared by: Alfandito Rais Akbar 18222037

Important Link

Link Github	https://github.com/alfanditora/Phinaple-Tree
API Documentation	https://github.com/alfanditora/Phinaple-Tree/tree/main/backend
Website	https://phinaple.vercel.app

Micro Service

My Service

- **Platform e-commerce** yang memungkinkan pengguna untuk menelusuri dan membeli smartphone secara online.
- **Chatbot yang didukung oleh Gemini AI** untuk membantu pelanggan membuat keputusan yang tepat tentang pembelian mereka.

Friend Service

- Dikembangkan oleh Muhammad Faiz Attharahman
- **Gateway pembayaran** yang memungkinkan pengguna untuk menyelesaikan pembayaran dengan *cryptocurrency* (SOL atau USDT)

New Service

- Pengintegrasian kedua service menghasilkan **Phinaple Tree** yang menjadi platform e-commerce yang lengkap dan mudah dalam mencari *smartphone* yang diinginkan

About Products

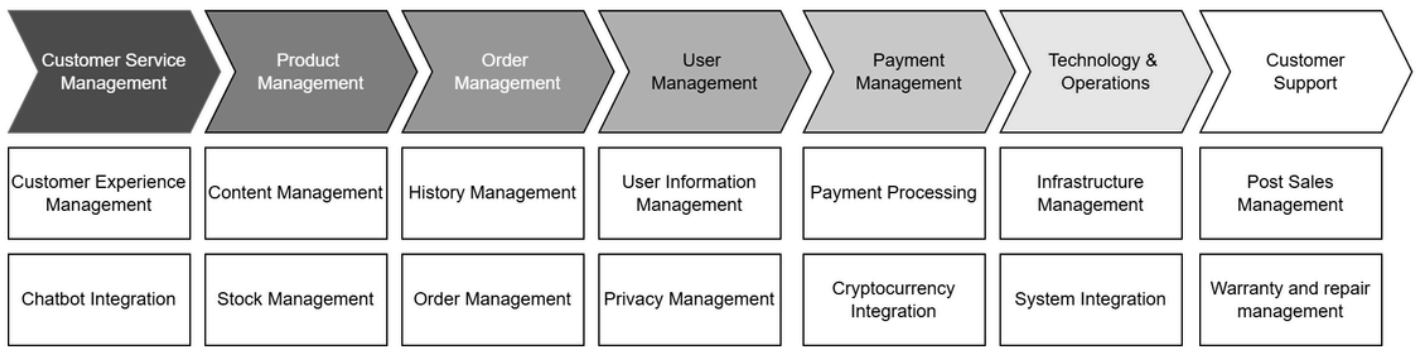
Proyek ini adalah platform e-commerce yang memungkinkan pengguna untuk menelusuri dan membeli *smartphone* secara online. Ini mengintegrasikan chatbot yang didukung oleh Gemini AI untuk membantu pelanggan dalam membuat keputusan yang tepat tentang pembelian mereka. Selain itu, ia menampilkan gateway pembayaran yang aman melalui SOLSTRA, solusi pembayaran khusus untuk pembayaran mata uang kripto (SOL atau USDT).

Features

- **Katalog Produk yang Luas:** Jelajahi berbagai macam *smartphone* termasuk berbagai merek, spesifikasi, dan harga.
- **Gemini AI Chatbot:** Gunakan bantuan chatbot berbasis AI untuk membantu pelanggan menemukan *smartphone* yang tepat untuk kebutuhan mereka.
- **Integrasi Pembayaran SOLSTRA:** Proses pembayaran dengan mulus menggunakan mata uang kripto SOL atau USDT melalui gateway SOLSTRA yang aman.
- **Desain Responsif:** Situs web ini dioptimalkan untuk desktop dan perangkat seluler.
- **Inventaris Real-time:** Ketersediaan stok yang diperbarui untuk memastikan pelanggan mendapat informasi tentang ketersediaan produk.
- **Manajemen Pesanan:** Pelacakan dan pengelolaan pesanan yang mudah.

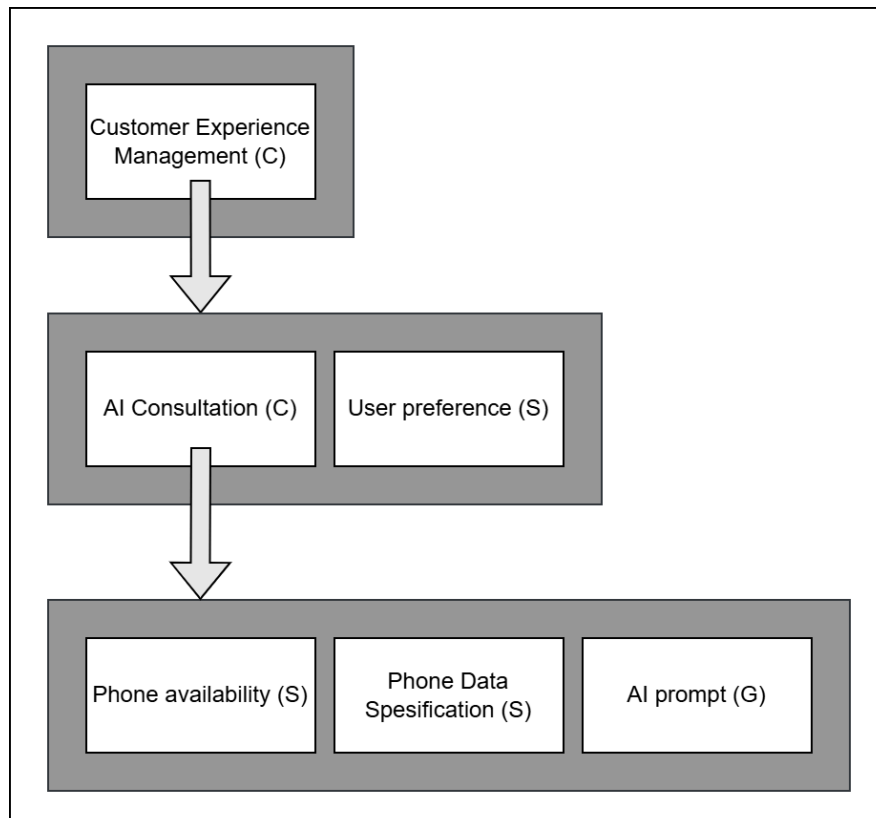
Business Capability Model

Phone-Store Chatbot



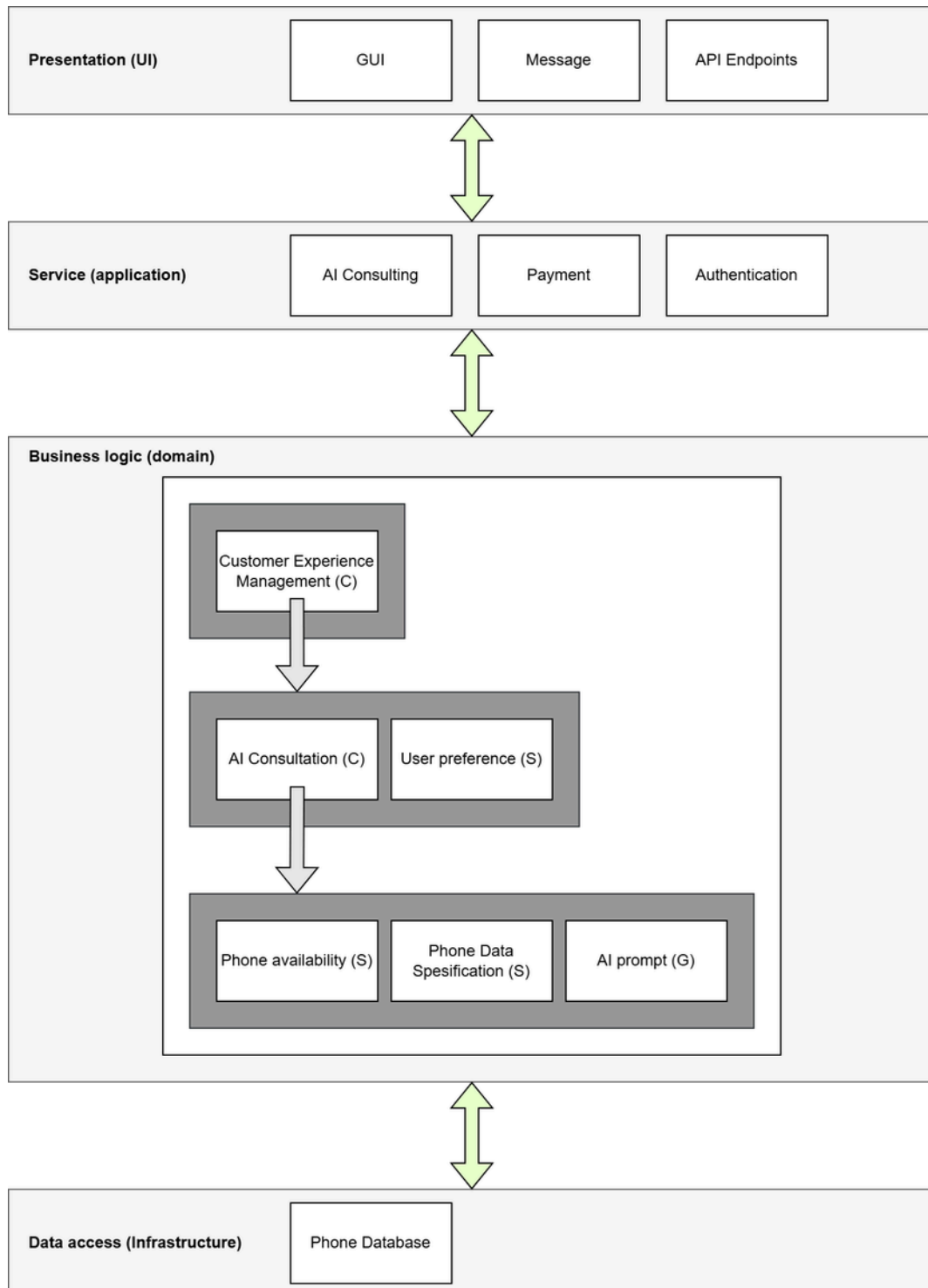
Capability mapping ini memberikan gambaran yang jelas tentang keterkaitan antara fungsi-fungsi inti, strategis, dan pendukung dalam proyek toko smartphone yang Anda buat. Setiap capability—baik itu Customer Service Management, Product Management, Order Management, User Management, Payment Management, Technology & Operations, maupun Customer Support—terhubung satu sama lain dan memainkan peran penting dalam menjaga keseimbangan antara operasional harian, pengembangan produk, pelayanan pelanggan, dan teknologi yang digunakan.

Core Subdomain & Bounded Context

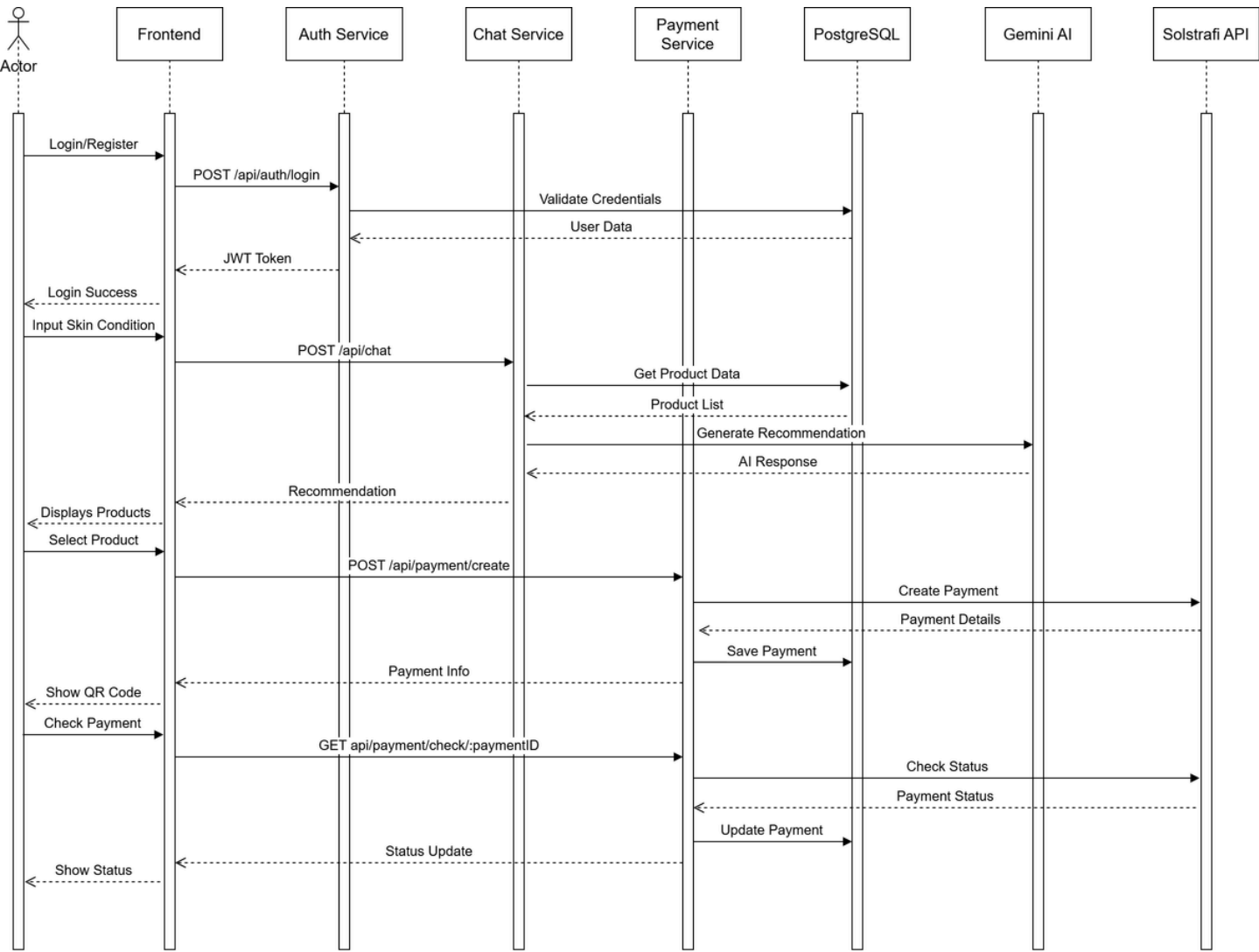


Customer Experience Management menjadi inti dari sistem, dengan AI Consultation sebagai penggerak utama untuk menghubungkan data preferensi pelanggan dan informasi produk. Alur ini memastikan bahwa pengalaman pelanggan terintegrasi dengan data yang relevan untuk menghasilkan keputusan yang mendukung kepuasan pelanggan.

Proposes Software Architecture



Phinaple Tree mengadopsi layer architecture yang terdiri dari empat layer utama diantaranya Presentation Layer yang menangani UI dan interaksi pengguna menggunakan React components, Service Layer yang berisi logika aplikasi untuk autentikasi, chat, dan pembayaran, Business Logic Layer yang mengatur aturan bisnis, serta Data Access Layer yang mengelola interaksi dengan database MongoDB dan external services (Gemini AI dan Solstrafi).



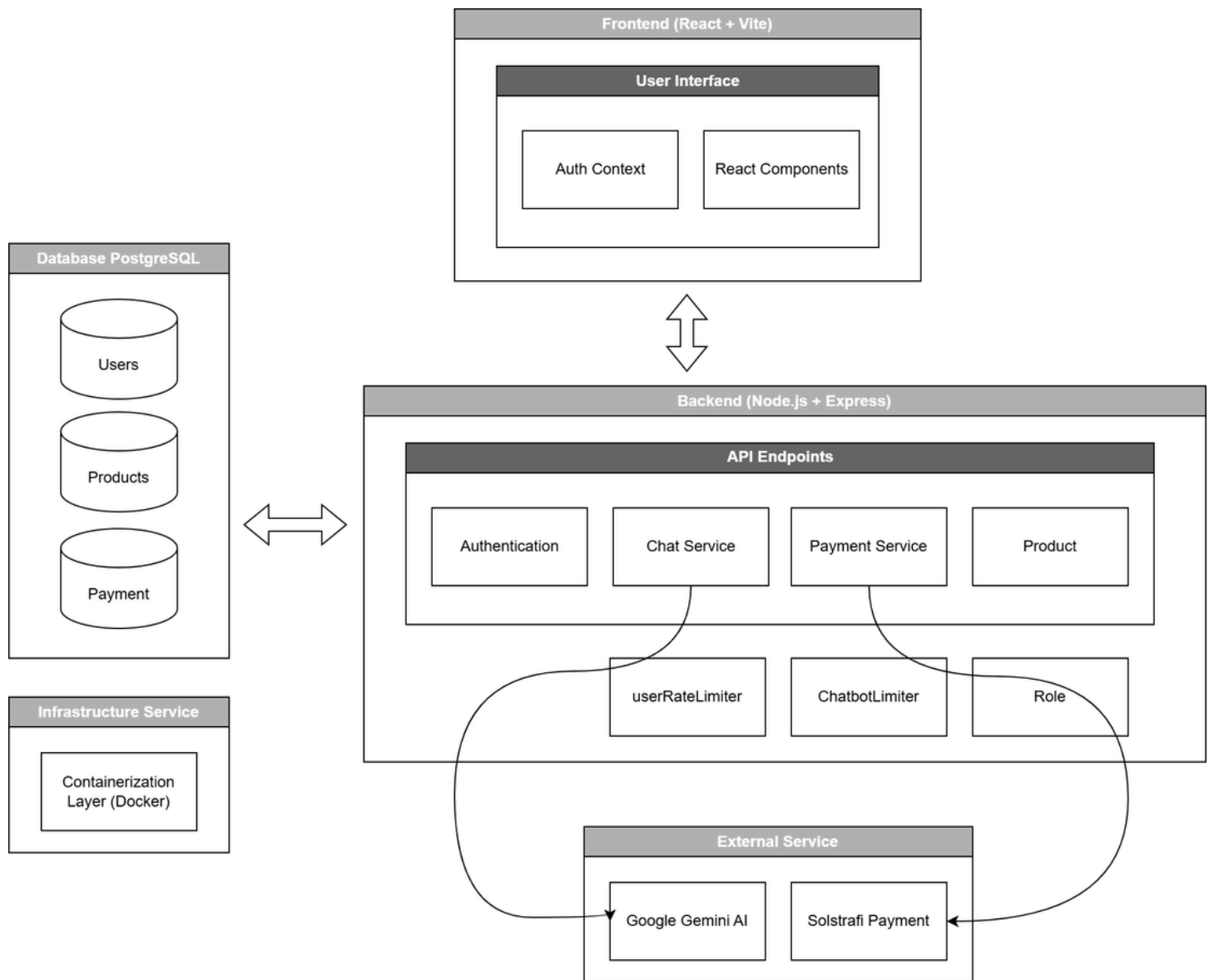
Sequence diagram dalam Phinaple Tree dimulai dari autentikasi user menggunakan JWT, dilanjutkan dengan proses chat dimana user menginputkan preverensi mereka yang kemudian diproses oleh Gemini AI untuk mendapatkan rekomendasi produk dari database, dan diakhiri dengan proses pembayaran menggunakan Solana Pay melalui Solstrafi API yang menghasilkan QR code untuk transaksi dan sistem monitoring status pembayaran, dengan setiap langkah proses ini terdokumentasi dalam database postgresQL untuk tracking dan history

Technology Architecture

Pada software yang diajukan akan di buat menjadi web-based dimana pelanggan dapat mengakses secara langsung lewat web yang nantinya dibuat. Pada hal ini teknologi yang digunakan untuk membangun web ini adalah sebagai berikut:

1. **React.js:** Framework ini digunakan untuk membangun interface dari web dan merupakan library yang bersifat open source sehingga memudahkan dalam Pembangunan sistem nantinya. Framework ini juga mendukung pembuatan UI yang responsif dan interaktif sehingga dapat memberikan pengalaman pengguna yang baik.
2. **PostgreSQL:** Database relasional ini dipilih karena cocok untuk sistem yang membutuhkan penyimpanan data yang terstruktur. Dalam hal ini, data smartphone yang mana memiliki spesifikasi tertentu dapat dengan mudah dibuat dengan PostgreSQL.
3. **Node.js:** Node.js digunakan untuk menghubungkan interface dengan AI dan database. Framework ini ideal untuk digunakan karena memungkinkan komunikasi yang cepat, serta dapat diintegrasikan dengan baik untuk permintaan HTTP.
4. **Express.js:** framework yang bekerja pada runtime environment Node.js untuk mengelola server. Alasannya karena framework ini ringan dan didesain untuk menyederhanakan dan mempercepat pengembangan website.
5. **Gemini AI API:** AI yang dapat digunakan pada web service dengan API key. Digunakan untuk membuat chatbot yang berguna dalam merekomendasikan smartphone dari database sesuai dengan preverensi user.
6. **Docker:** Teknologi kontainerisasi yang berfungsi dalam memudahkan pengembangan dengan *environment* yang sudah disiapkan.
7. **Solstra Payments:** Sebuah layanan pembayaran berbasis Solana blockchain yang dikembangkan oleh Muhammad Faiz Atharrahman (18222063). Integrasi ini memungkinkan pengguna untuk membayar produk skincare yang direkomendasikan menggunakan cryptocurrency SOL atau USDT melalui Solana blockchain.

System Architecture



Frontend (React + Vite): Bagian frontend menggunakan React dan Vite untuk membangun antarmuka pengguna. Lapisan ini mencakup:

- **Auth Context:** Untuk menangani otentikasi pengguna secara global di seluruh aplikasi.
- **React Components:** Merupakan bagian modular untuk membangun antarmuka dan logika tampilan.

Lapisan ini bertanggung jawab untuk memberikan pengalaman pengguna yang intuitif dan mengelola komunikasi dengan backend.

Backend (Node.js + Express): Backend menggunakan Node.js dan Express untuk menyediakan berbagai **API Endpoints**. Endpoint ini melayani beberapa layanan utama:

- **Authentication:** Mengelola proses otentikasi dan otorisasi pengguna.
- **Chat Service:** Menangani fitur obrolan atau konsultasi.
- **Payment Service:** Mengelola alur transaksi pembayaran.
- **Product:** Menyediakan informasi terkait produk, seperti spesifikasi, harga, dan ketersediaan.

Backend juga dilengkapi dengan beberapa middleware atau layanan tambahan:

- **userRateLimiter:** Membatasi jumlah permintaan dari pengguna untuk mencegah penyalahgunaan.
- **ChatbotLimiter:** Mengontrol penggunaan chatbot untuk efisiensi sumber daya.
- **Role:** Menangani kontrol akses berbasis peran untuk pengguna.

Database (PostgreSQL): Database menggunakan PostgreSQL untuk penyimpanan data terstruktur dengan skema utama:

- **Users:** Menyimpan informasi pengguna.
- **Products:** Menyimpan detail produk yang tersedia di sistem.
- **Payment:** Menyimpan catatan transaksi pembayaran.

PostgreSQL dipilih untuk mendukung kemampuan kueri yang kompleks dan integritas data.

Infrastructure Service: Sistem menggunakan **Docker** sebagai lapisan kontainerisasi untuk mempermudah pengelolaan lingkungan pengembangan dan produksi. Docker memastikan konsistensi sistem di berbagai platform.

External Service: Sistem terintegrasi dengan dua layanan eksternal:

- **Google Gemini AI:** Digunakan untuk fitur kecerdasan buatan, seperti analisis data atau chatbot.
- **Solstrafi Payment:** Digunakan untuk memproses pembayaran dengan cryptocurrency

Development Plan

Development Environment

Frontend	React JS Vite + Tailwind CSS
Backend	Express JS + Node JS
Database	PostgreSQL
Container	Docker
Version Control	Git
Deployment	Vercel (Frontend) + Backend (Railway)
API Testing	Postman
Code Editor	Visual Studio Code

Development Step

Initializes Project	<ul style="list-style-type: none">• Setup React JS Vite + Tailwind CSS• Setup PostgreSQL• Setup Node JS + Express JS• Membuat github repository
Backend Development	<ul style="list-style-type: none">• Authentication• Integrasi Gemini AI API• Model database• Integrasi Solstra API
Frontend Development	<ul style="list-style-type: none">• Implementasi UI• Responsive
System Testing	<ul style="list-style-type: none">• Testing flow• Testing API
Deployment	<ul style="list-style-type: none">• Setup Docker• Build docker image• Deployment



Phinapple Tree
THANK YOU