

IT DIVISION / SOFTWARE ENGINEERING

Project Kelompok 6
Website CineWave - Sebuah Website untuk Pemesanan Ticket
Bioskop



Kelompok 6
[2802398914] - [Christofle Tjhai]
[2802414426] - [Marcelinus Wijaya Oey]
[2802407446] - [Jason Benoit Adianto]

IT Division Bina Nusantara 2024

Introduksi

CineWave adalah sebuah **aplikasi web pemesanan tiket bioskop** online yang dirancang untuk memberikan kemudahan bagi pengguna dalam memilih dan memesan tiket film bioskop secara online. Dengan perkembangan teknologi dan perubahan gaya hidup masyarakat yang semakin mengarah pada transaksi digital, CineWave hadir sebagai solusi yang mempermudah pengalaman menonton film di bioskop. Aplikasi ini bertujuan untuk menyediakan platform yang tidak hanya mudah digunakan, tetapi juga efisien dan aman dalam memesan tiket bioskop.

Beberapa **prinsip design website CineWave**, yang didasarkan pada kenyamanan dan ease-of-use client, diantaranya :

Kemudahan Akses : Proses pemesanan tiket secara manual di bioskop seringkali menyita waktu dan tenaga. Masyarakat seringkali harus mengantre untuk membeli tiket, memilih tempat duduk, dan menunggu di lokasi yang mungkin penuh. Dengan CineWave, pengguna dapat memesan tiket kapan saja dan di mana saja tanpa harus datang langsung ke bioskop.

Kesesuaian : Dengan banyaknya bioskop yang menawarkan film-film terbaru, pengelolaan penjualan tiket yang manual sering kali menimbulkan Kesalahan Pemesanan atau Keterlambatan Update Ketersediaan Kursi. CineWave mengatasi masalah ini dengan menyediakan sistem yang selalu memastikan bahwa informasi mengenai ketersediaan kursi dan jadwal film selalu diperbarui secara real time

Kenyamanan : CineWave memberikan Desain User Interface dan Experience yang mudah dipahami dan nyaman bagi pengguna, sehingga meningkatkan kemudahan penggunaan dan insentif bagi para pengguna untuk menggunakan service CineWave.

Project : Website Pemesanan Tiket Bioskop

Website pemesanan tiket bioskop yang menggunakan tech-stack :

Frontend : HTML, JavaScript, TypeScript

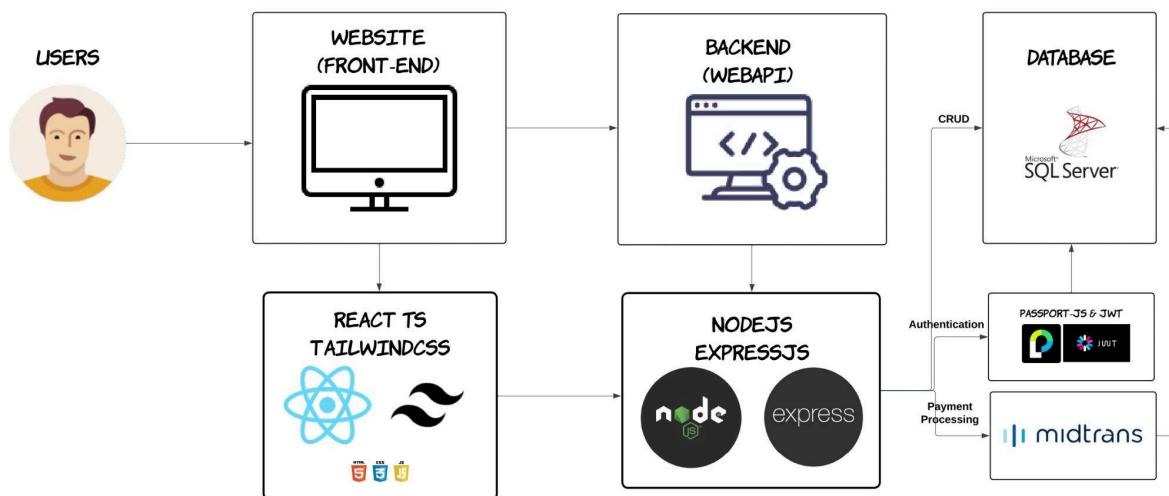
Framework Frontend : ReactJS + TypeScript, TailwindCSS

Backend : NodeJS (JavaScript)

Framework Backend : ExpressJS

Database : SQL Server

OS : Windows 11



Arsitektur aplikasi yang digunakan adalah **Client-Server Model**, di mana sistem dibagi menjadi dua komponen utama: Client dan Server. Dalam model ini, client (frontend aplikasi) akan berinteraksi dengan server (backend) melalui jaringan untuk mendapatkan data dan layanan yang diperlukan.

Client akan menggunakan tampilan website (frontend) sebagai user interface, dimana user interface tersebut akan menyajikan data yang diperoleh melalui response API backend terhadap request-request yang diberikan oleh frontend.

Detail Teknis dan Cara Kerja

1. Frontend (Client):

- **React TS dan Tailwind CSS:**

- **React TS:** Sebuah library JavaScript yang populer untuk membangun antarmuka pengguna yang dinamis. Dengan TypeScript, kita bisa menulis kode yang lebih terstruktur dan mudah dipelihara.
- **Tailwind CSS:** Sebuah framework CSS yang menyediakan kelas-kelas siap pakai untuk styling. Ini mempercepat proses styling dan membuat tampilan aplikasi menjadi lebih konsisten.

- **Cara Kerja:**

- Ketika pengguna berinteraksi dengan aplikasi (misalnya, mengklik tombol), frontend akan mengirimkan permintaan (request) ke backend.
- Request ini berisi informasi tentang tindakan yang ingin dilakukan pengguna.

2. Backend (Server):

- **Node.js dan Express.js:**

- **Node.js:** Sebuah runtime environment yang memungkinkan kita menjalankan JavaScript di luar browser.
- **Express.js:** Sebuah framework Node.js yang memudahkan dalam membangun server web dan API.
- **Passport.js:** Digunakan untuk menangani autentikasi pengguna.
- **Midtrans:** Sebuah gateway pembayaran yang digunakan untuk memproses transaksi pembayaran.

- **Cara Kerja:**

- Backend menerima request dari frontend.
- Backend kemudian memproses request tersebut.
- Jika diperlukan, backend akan berinteraksi dengan database (SQL Server) untuk mengambil atau menyimpan data.
- Setelah proses selesai, backend akan mengirimkan respons kembali ke frontend.
- Respons ini bisa berupa data yang diminta, pesan error, atau informasi lainnya.

3. Database (SQL Server):

- **Fungsi:** Menyimpan data yang diperlukan oleh aplikasi, seperti data pengguna, produk, pesanan, dan lain-lain.
- **Interaksi:** Backend berinteraksi dengan database menggunakan query SQL untuk mengambil, memasukkan, memperbarui, atau menghapus data.

Alur Kerja Sederhana

1. Pengguna membuka aplikasi di browser.
2. Browser mengirimkan request ke server untuk memuat halaman awal.
3. Server memproses request, mengambil data yang diperlukan dari database, dan mengirimkan HTML, CSS, dan JavaScript ke browser.
4. Browser merender halaman dan menampilkannya kepada pengguna.
5. Ketika pengguna berinteraksi dengan elemen di halaman, browser mengirimkan request baru ke server.
6. Proses ini berulang terus-menerus.

Keuntungan Arsitektur ini

- **Skalabilitas:** Frontend dan backend dapat dikembangkan dan di-scale secara terpisah.
- **Maintainability:** Kode menjadi lebih terorganisir dan mudah dipelihara.
- **Reusability:** Komponen-komponen dapat digunakan kembali dalam proyek lain.
- **Performance:** Dengan pemisahan tugas, kinerja aplikasi dapat ditingkatkan.

Design software architecture ini memungkinkan pengembangan aplikasi web yang kompleks dengan memisahkan tampilan (frontend) dari logika bisnis dan data (backend). Setiap bagian memiliki tanggung jawab yang jelas, sehingga memudahkan dalam pengembangan dan pemeliharaan, dan juga meningkatkan skalabilitas aplikasi tersebut.

Perubahan pada satu bagian tidak akan berdampak signifikan pada bagian lainnya. Ketika jumlah pengguna meningkat, kita dapat dengan mudah menambahkan server baru untuk menangani beban tambahan, sesuai dengan kebutuhan pada saat itu.

Features :

Authentication (Login, Register, Logout)

Implementasi authentication melalui **PassportJS Local & JWT**.

PassportJS Local -> library yang memudahkan authentication.

JWT -> standar terbuka yang digunakan untuk mentransfer data secara aman antara dua pihak.

Terdapat beberapa API endpoint & page website yang terkunci sebelum user melakukan login dan mendapatkan token JWT.

Navbar (Home, Now Playing, Upcoming, Theaters, MyTicket, Payment History, FAQ)

Implementasi melalui **component ReactJS + TypeScript**

Navigasi antarpages dengan memanfaatkan **React Router**.

React Router -> library untuk mengelola navigasi di aplikasi ReactJS.

Search Bar (filtering movie berdasarkan keyword)

Implementasi searchbar melalui query ke database melalui **mssqlserver**

mssqlserver -> library untuk mengakses database melalui backend Node.

Fitur ini diimplementasikan dengan mem-fetch semua

Payment Processing (Midtrans)

Implementasi melalui library **midtrans-nodejs-client**

midtrans-nodejs-client -> library integrasi midtrans dengan NodeJS.

Simulasi payment processing dengan memanfaatkan midtrans sandbox untuk simulasi pembayaran.

Movie Details (Synopsis, Trailer, Poster, Sutradara, Producer, etc)

Implementasi dengan query GET ke database melalui **mssqlserver**

Menampilkan trailer movie sebagai embedded youtube link, poster movie sebagai link ke image CDN (cloudinary), dan juga berbagai informasi terkait film tersebut, seperti produser, sutradara, age rating, duration, dimension, release date, dll. (info disimpan pada table database).

Movie Seat Reservation

Implementasi dengan query POST ke database melalui **mssqlserver**

Backend akan melakukan post dan menambahkan instansi data yang terkait dengan fitur seat reservation kepada database.

Cancel Payment Button

Implementasi dengan query DELETE ke database melalui **mssqlserver**

Backend akan melakukan delete dan menghapus instansi data yang terkait dengan pembatalan pembayaran & booking seat reservation.

Movie Filtering (berdasarkan region theater)

Implementasi dengan query SELECT FROM ke database melalui **mssqlserver**, dimana kita menampilkan movies yang memenuhi kriteria. Movie filtering bekerja berdasarkan eksistensi showtime yang menuju kepada movie_id & theater_id yang sedang dituju.

Route Protection

Implementasi perlindungan route melalui authentication JWT.

Bertujuan untuk mencegah akses tidak sah ke data sensitif pengguna ataupun sistem.

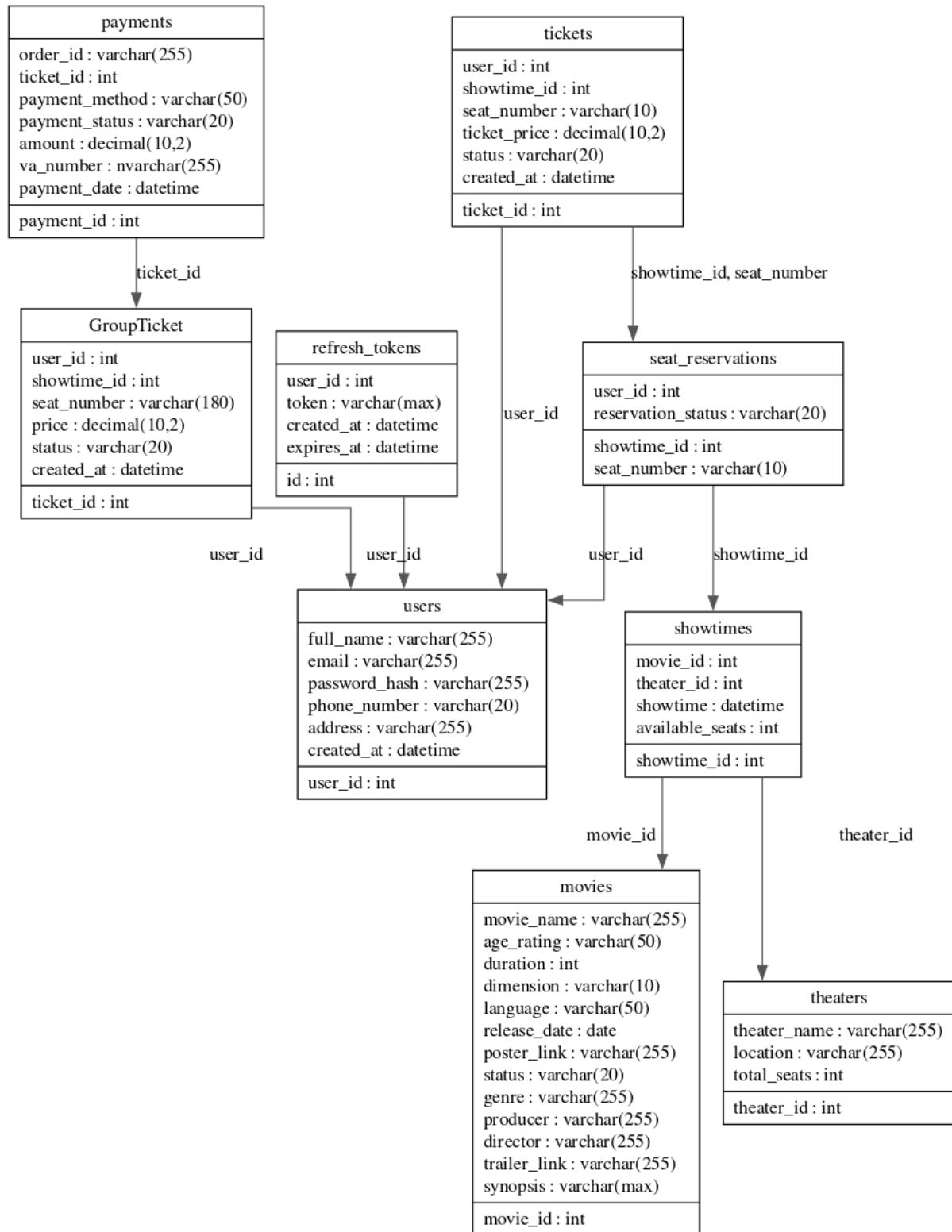
Payment History Page

Implementasi dengan query GET melalui **mssqlserver** ke database, untuk mendapatkan informasi relevan terkait dengan suatu pembayaran yang terdaftar ke midtrans.

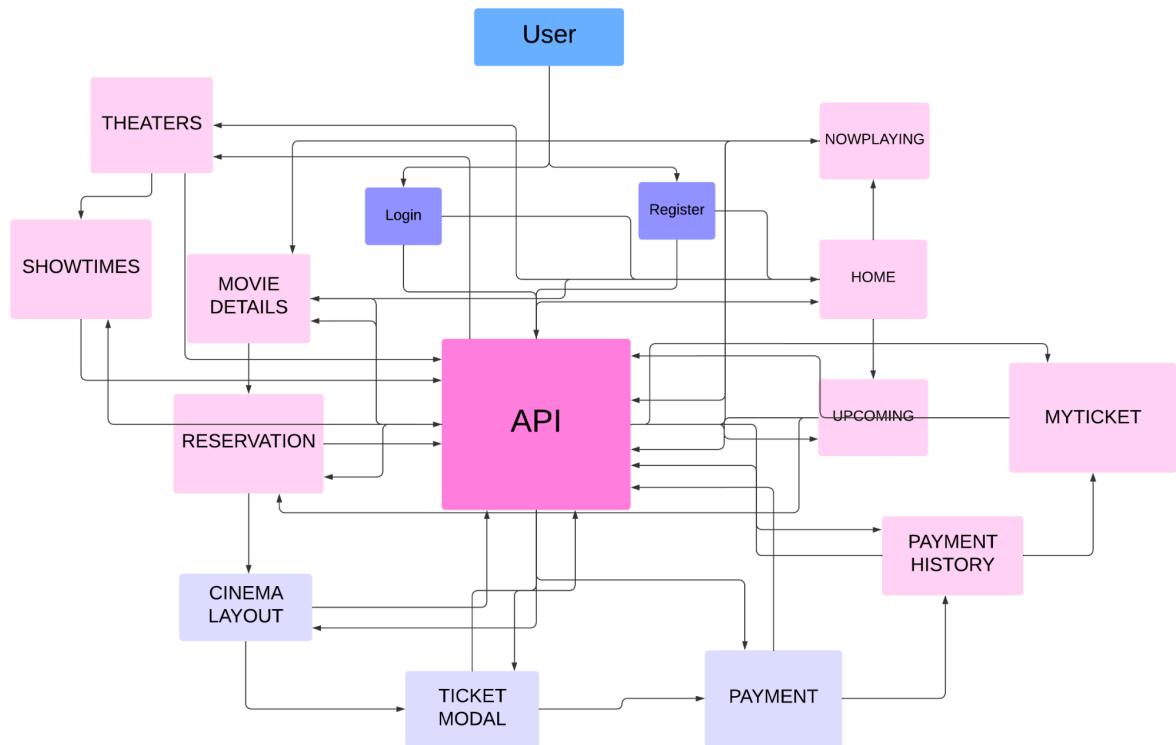
My Ticket Page

Implementasi dengan query GET melalui **mssqlserver** ke database, untuk mendapatkan riwayat booking ticket suatu user, berdasarkan user_id dari akun yang meminta request tersebut.

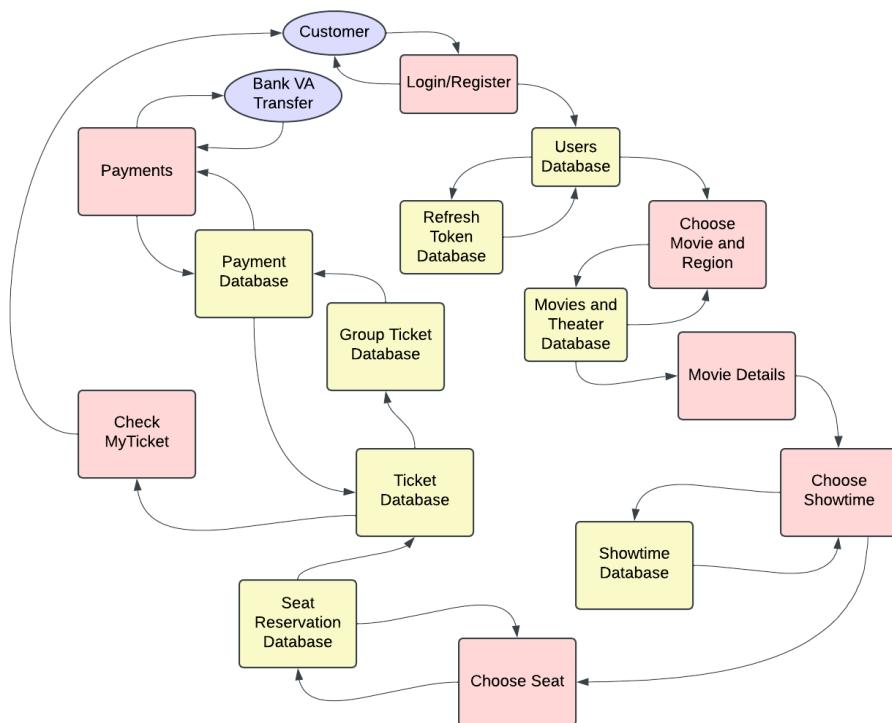
ERD



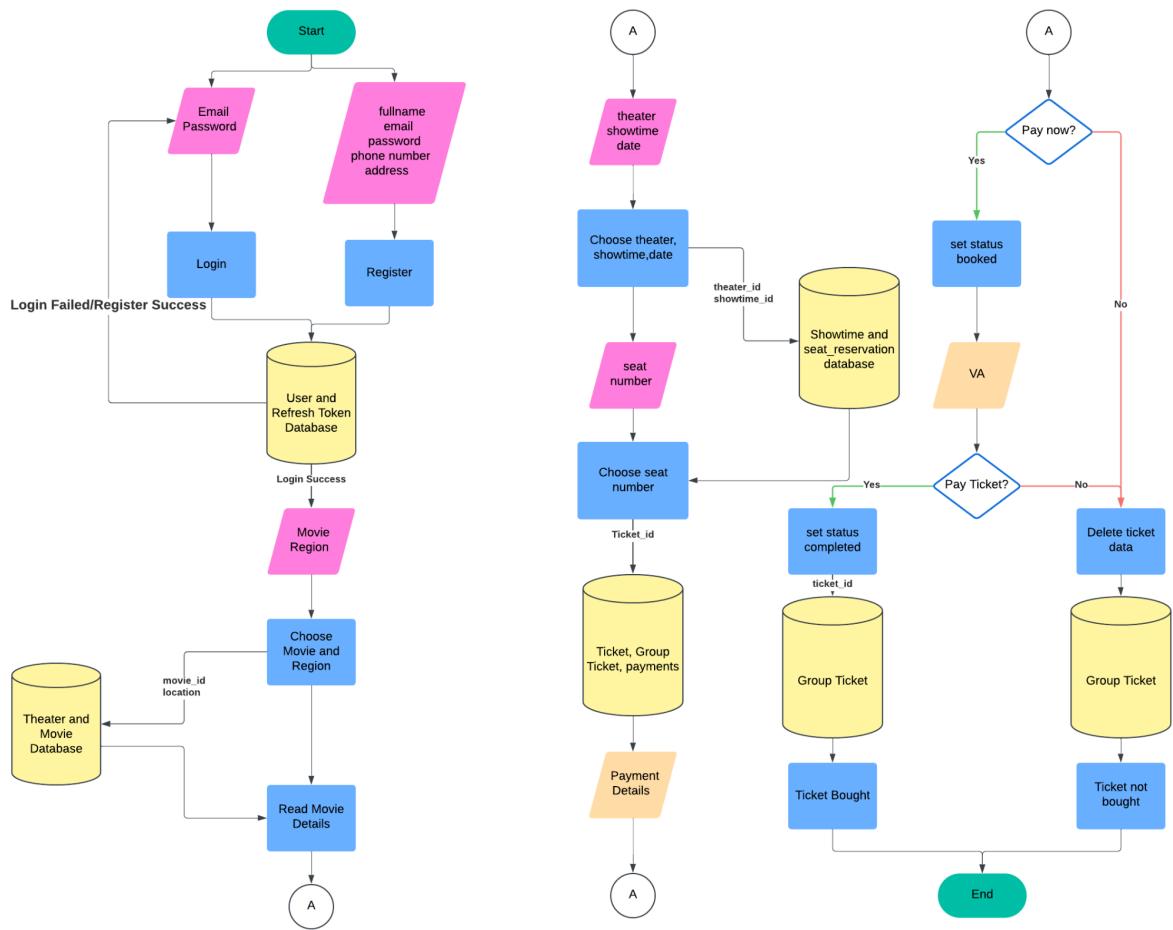
Concept Map



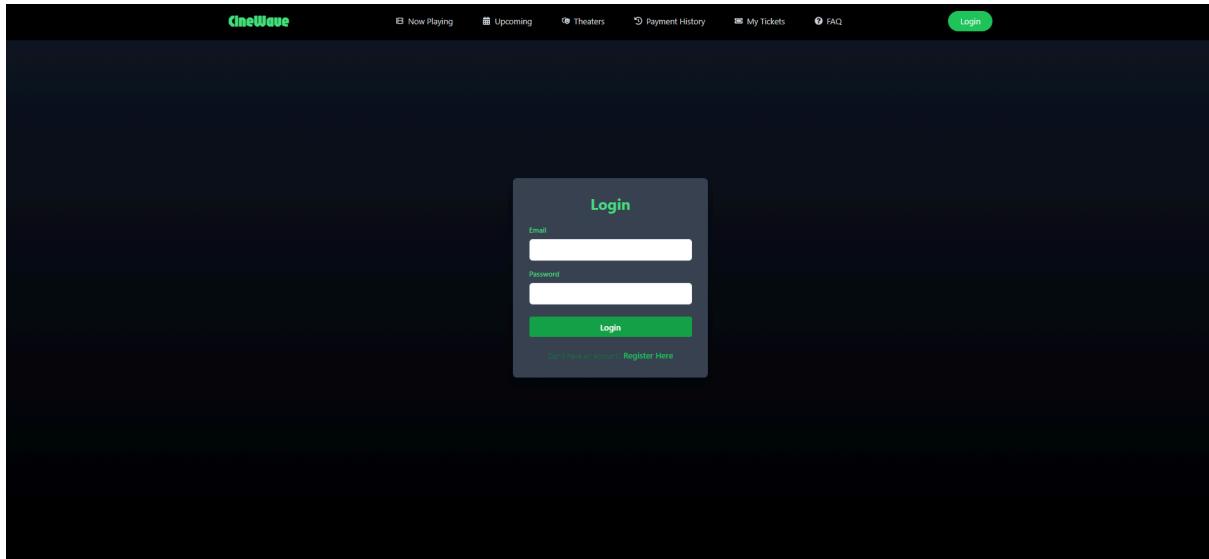
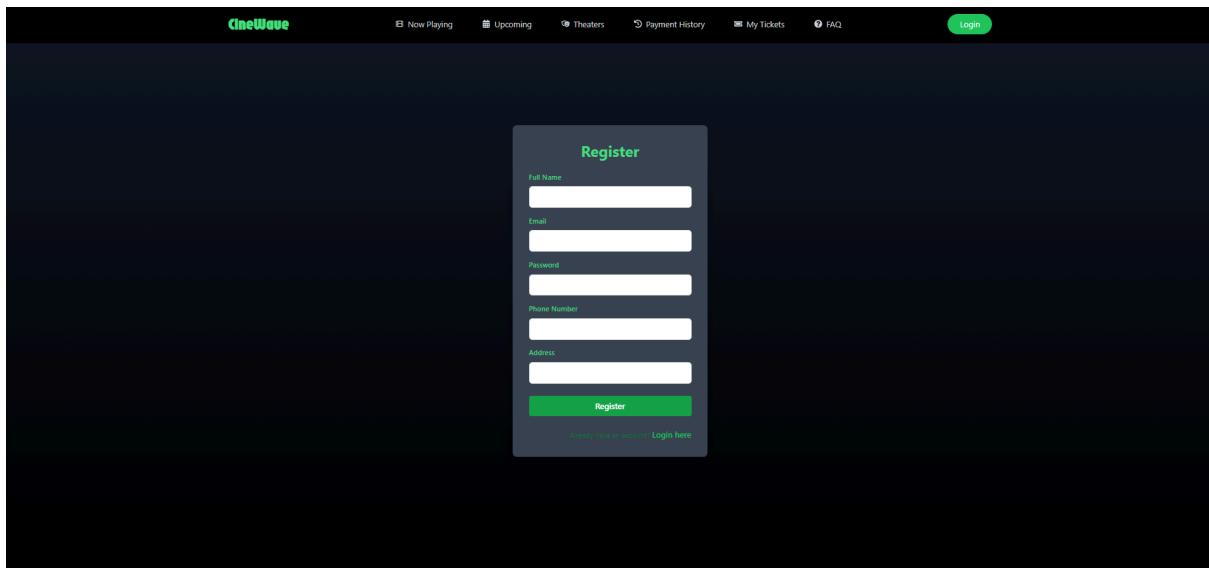
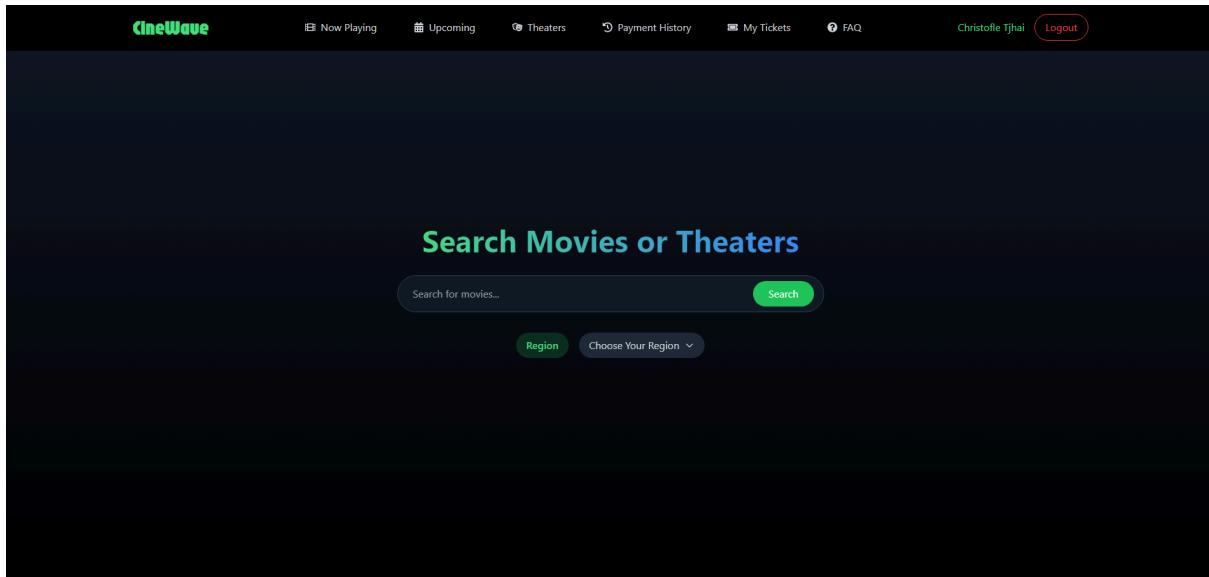
Data Flow Diagram (DFD)

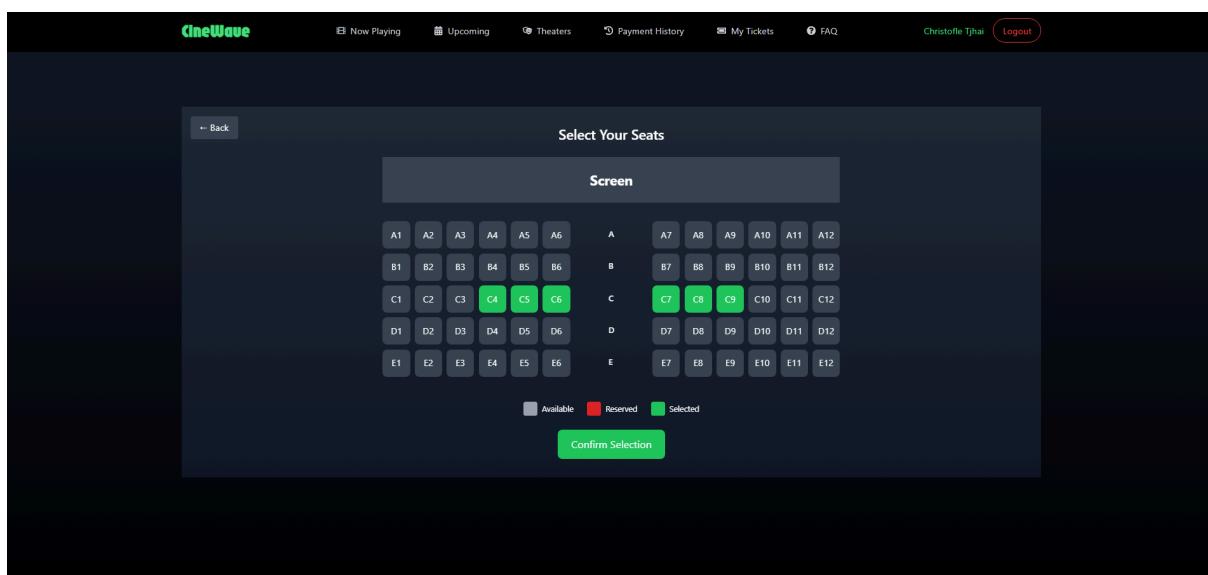
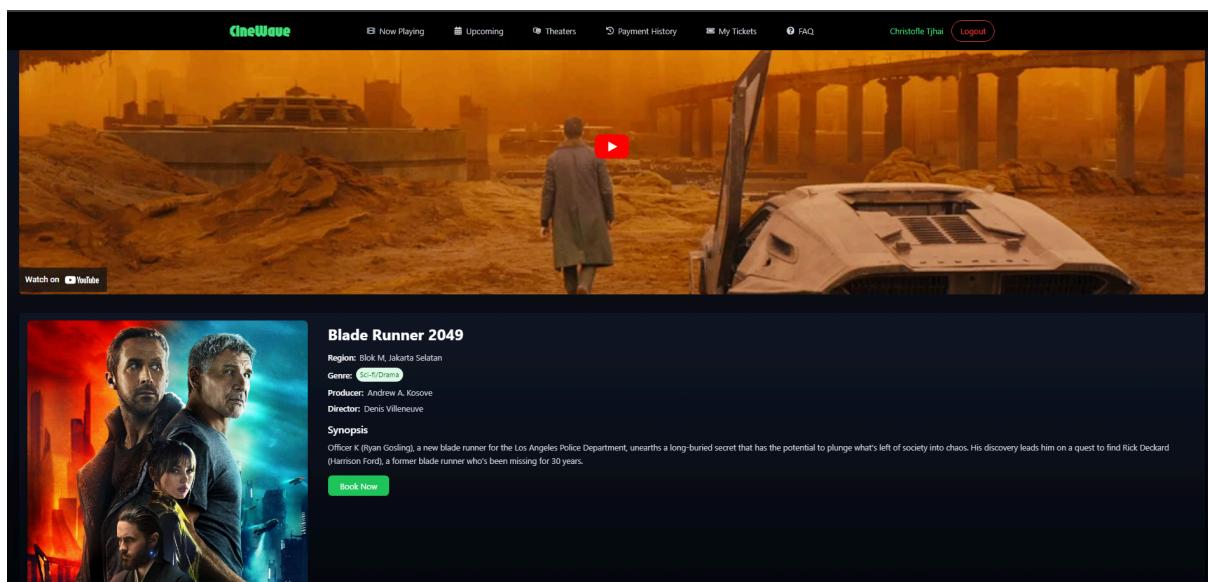
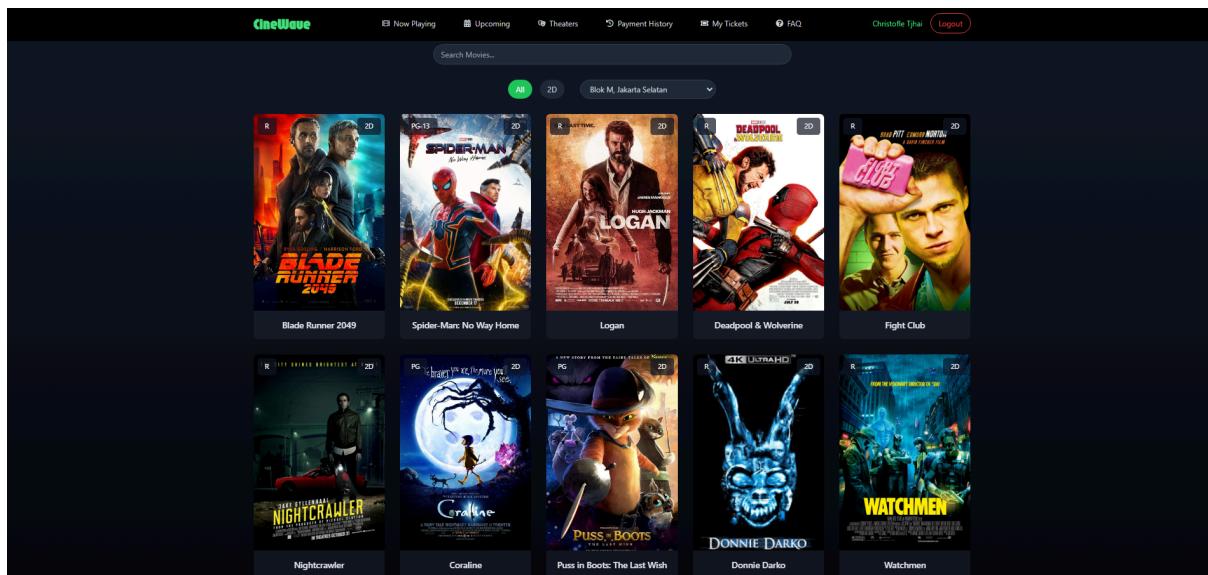


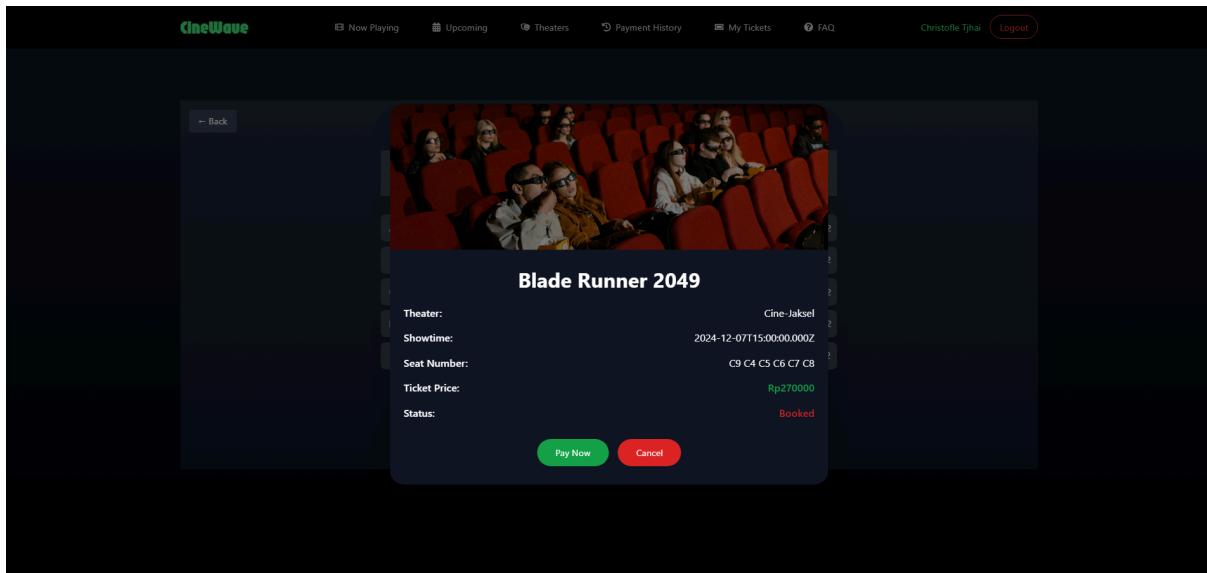
Flowchart



UI (User Interface) Application







Payment History						
Refresh All Statuses						
Order ID	Seat Number	Payment Method	Status	Amount	VA Number	Payment Date
BoATcnopqJmPaiB3	C5 C6 C4	bank_transfer	settlement	Rp 135000	70785204897128913836252	2024-12-05 10:34:40.733
dozQzd7VtaPMFKp5	A4 A5 A6	bank_transfer	settlement	Rp 135000	7078501596536294216461	2024-12-05 10:34:40.743
ZumyuUeaV5sop6	D7 D8 D9	bank_transfer	settlement	Rp 135000	70785686178120358905666	2024-12-05 10:34:40.730

My Tickets						
 Blade Runner 2049 Cine-Jaksel 2024-12-07T11:15:00+00:00Z Seat: C5 Rp 45000	 Blade Runner 2049 Cine-Jaksel 2024-12-07T11:15:00+00:00Z Seat: C6 Rp 45000	 Blade Runner 2049 Cine-Jaksel 2024-12-07T11:15:00+00:00Z Seat: C4 Rp 45000				
 Spider-Man: No Way Home Cine-Jaksel 2024-12-07T11:20:00+00:00Z Seat: A4 Rp 45000	 Spider-Man: No Way Home Cine-Jaksel 2024-12-07T11:20:00+00:00Z Seat: A5 Rp 45000	 Spider-Man: No Way Home Cine-Jaksel 2024-12-07T11:20:00+00:00Z Seat: A6 Rp 45000				
 Deadpool & Wolverine Cine-Jaksel 2024-12-04T09:00:00+00:00Z Seat: D7 Rp 45000	 Deadpool & Wolverine Cine-Jaksel 2024-12-04T09:00:00+00:00Z Seat: D8 Rp 45000	 Deadpool & Wolverine Cine-Jaksel 2024-12-04T09:00:00+00:00Z Seat: D9 Rp 45000				

SDLC : Hybrid -> Waterfall & Agile.

Waterfall :

Kami mengembangkan projek berdasarkan urutan tahapan yang sequential, mulai dari perencanaan hingga implementasi. Kami menekankan perencanaan dan dokumentasi yang detail pada proses development. Selain karakteristik SDLC Waterfall tersebut, kami juga menerapkan SDLC Agile, dimana kami menerapkan prioritisasi fitur dan fleksibilitas dalam mengubah rencana.

KONTRIBUSI :

Nama	Role	Apa yang dikerjakan
Christofle Tjhai	Fullstack Developer	<ul style="list-style-type: none">• Mengoordinasikan tahap-tahap pengerajan secara linear.• Men-design proses kerja website, software architecture & design database (ERD).• Menentukan teknologi dan library-library yang akan digunakan dalam projek.• Menginisialisasikan git repository frontend & backend dengan boilerplate code.• Mengoordinasikan git workflow (mengatur branch, resolve conflicts, maintain repository).• Melakukan testing pada integrasi komponen-komponen frontend & api backend.• Melakukan debugging untuk penyelesaian permasalahan pada frontend & backend projek.• Mengimplementasikan API untuk CRUD ke table database [movies, seat_reservation, showtimes, theaters, tickets].• Mengoptimalkan logic API & logic fetching (frontend) untuk mencegah fetching data redundant dan menjaga

		<p>maintainability project secara menyeluruh.</p> <ul style="list-style-type: none"> • Mengerjakan komponen frontend movie details & api backend untuk movie details. • Mengisikan dummy data pada database untuk deployment (karena tidak dideploy, hanya untuk schema)
Marcelinus Wijaya Oey	Fullstack Developer	<ul style="list-style-type: none"> • Mengimplementasikan authentication users (PassportJS & JWT) pada backend aplikasi. • Mengimplementasikan API untuk CRUD ke table database [users, groupon, payments]. • Mengembangkan API payments untuk disesuaikan dengan platform payment processing (Midtrans). • Mengerjakan komponen frontend untuk logic authentication. • Mengerjakan komponen frontend untuk processing payments & payment history. • Mengerjakan komponen frontend untuk ticket history. • Menambahkan table GroupTicket ke database sebagai denormalisasi untuk mengoptimalkan pemesanan banyak tiket.
Jason Benoit Adianto	Frontend Developer	<ul style="list-style-type: none"> • Men-design Data Flow Diagram berdasarkan ERD & proses kerja website yang sudah ditentukan. • Men-design UI/UX tampilan website secara menyeluruh. • Mengerjakan komponen frontend homepage. • Mengerjakan komponen frontend navigation bar.

- Mengerjakan komponen frontend page nowplaying & upcoming movies.
- Mengerjakan komponen frontend untuk seat reservation & payment confirmation.
- Mengerjakan komponen frontend theaters (theaters list).
- Mengerjakan komponen frontend untuk page FAQ.

BACKLOG / TIMELINE

Kesimpulan :

CineWave menawarkan platform pemesanan tiket bioskop online yang lengkap. Fitur-fitur seperti pemilihan kursi, pembayaran yang aman, dan tampilan yang user-friendly membuat CineWave menjadi pilihan yang tepat bagi para pencinta film.

Pilihan Teknologi Frontend

- **ReactJS dan TypeScript:** Kombinasi ini memberikan fleksibilitas dalam membangun antarmuka pengguna yang dinamis dan terstruktur. TypeScript membantu mencegah kesalahan pemrograman dan meningkatkan maintainability kode, sementara React memungkinkan pengembangan komponen yang reusable. Selain **efisiensi dalam pengembangan**, React dan TypeScript juga berkontribusi pada performa aplikasi. Virtual DOM yang menjadi ciri khas React memungkinkan pembaruan tampilan secara efisien hanya pada bagian yang berubah, sehingga mengurangi reflow dan repaint yang tidak perlu. TypeScript dengan sistem tipenya yang kuat membantu mencegah kesalahan runtime yang dapat menyebabkan penurunan performa.
- **Tailwind CSS:** Framework CSS ini mempercepat proses styling dengan menyediakan kelas-kelas yang siap pakai, sehingga tampilan aplikasi menjadi lebih konsisten dan modern.

Pilihan Teknologi Backend

- **Node.js dan Express.js:** Kombinasi ini memungkinkan pengembangan server-side yang efisien dan scalable. Node.js memungkinkan JavaScript berjalan di server, sementara Express.js menyediakan framework yang ringan dan mudah digunakan untuk membangun API.

Elaborasi mengapa memilih ExpressJS & NodeJS :

Terkait dengan kebutuhan pada projek, pemesanan tiket perlu ditindaklanjuti dengan processing pembayaran. Pada awal projek, kami berfikir untuk melakukan pendekatan pada tech-stack @ Bina Nusantara dengan menggunakan C# AspNET. Tetapi, saya telah melakukan research, dan majoritas library payment processing yang tersedia untuk

C# ASP.NET, seperti Paypal SDK, Stripe .NET, PayU .NET, ataupun Razorpay .NET, diantara semua SDK tersebut, tidak memberikan support untuk metode pembayaran yang sering digunakan di negara Indonesia, seperti Bank BCA, ShopeePay, DANA, ataupun GoPay. Walaupun untuk sekarang kami terbatas oleh deadline waktu dan hanya men-support payment via virtual account BCA, design aplikasi mendukung skalabilitas penambahan metode pembayaran lain, seperti e-wallet.

Selain itu, ExpressJS, sebagai framework backend yang tidak terlalu memaksakan struktur tertentu (non-opinionated), memberikan kebebasan yang sangat besar bagi pengembang, terutama bagi mereka yang baru memulai. **Fleksibilitas** ini memungkinkan kami, sebagai developer dengan pengalaman yang minim, untuk membangun sistem backend sesuai dengan kebutuhan spesifik proyek kami, tanpa terikat pada konvensi yang terlalu ketat, seperti yang terjadi di teknologi backend lainnya (opinionated).

- **SQL Server:** Sebagai database, SQL Server dipilih karena kemampuannya dalam mengelola data yang terstruktur dengan baik, mendukung transaksi yang kompleks, dan memiliki performa yang handal.

Kami **memutuskan** untuk memisahkan folder backend dan frontend untuk mempermudah pemeliharaan, memungkinkan pengembangan yang terpisah, dan memberikan fleksibilitas, di mana backend dan frontend dapat menggunakan framework atau alat berbeda sesuai kebutuhan. Selain itu, pemisahan ini mendukung skalabilitas, mempermudah pengujian independen seperti postman, serta memungkinkan build dan deployment yang terpisah. Pemisahan backend dan frontend mendukung skalabilitas dengan memungkinkan keduanya ditingkatkan secara independen sesuai kebutuhan. Backend dapat diskalakan menggunakan server yang lebih kuat untuk menangani lonjakan permintaan, sementara frontend dapat memanfaatkan **CDN** seperti cloudinary untuk mempercepat distribusi file statis ke pengguna.