PROJECT KELOMPOK 10

“Website pemesanan tiket Bioskop - Sebuah Website untuk Semua Pengguna”



Kelompok 10:

2702296142 - Kenzie Edernez  
2702229264 - Andhika Pratama Putra  
2702243110 - Yoseph Oktavianus Yusanto  
2702340156 - Kevin Pangsawira

IT Division Bina Nusantara

2024

**Introduksi**

Aplikasi ini dirancang untuk mempermudah pengguna membeli tiket bioskop secara cepat, efisien, dan tanpa antrian. Dengan aplikasi ini, pengguna dapat melihat jadwal film yang sedang tayang, memeriksa ketersediaan kursi secara real-time, dan memilih kursi sesuai preferensi. Aplikasi ini juga menyediakan informasi tentang film yang akan tayang di masa mendatang, sehingga memudahkan perencanaan kunjungan ke bioskop. Selain itu, tersedia berbagai metode pembayaran yang fleksibel untuk memberikan pengalaman transaksi yang nyaman. Dengan fitur-fitur ini, aplikasi ini menjadi solusi praktis bagi siapa saja yang ingin menikmati pengalaman menonton film di bioskop tanpa repot.

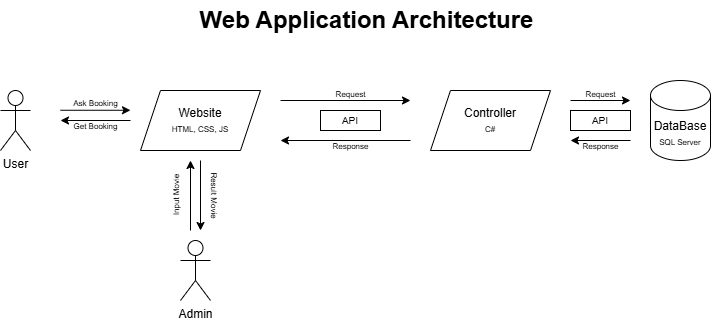
**PROJECT pemesanan tiket bioskop**

Backend: C# 6.0

Framework Backend: .NET 6  
 Frontend: HTML, CSS, JS

Framework Frontend : React  
 OS: Windows 11 Pro

Database: Microsoft SQL Server Management Studio



**User**

User adalah individu yang menggunakan aplikasi ini untuk berbagai tujuan terkait bioskop, seperti mengecek daftar film yang tersedia, melihat jadwal tayang di lokasi tertentu, dan memesan tiket bioskop. Selain itu, user juga dapat melakukan pembayaran tiket dan menerima konfirmasi berupa e-ticket setelah transaksi berhasil dilakukan. Semua interaksi user dilakukan melalui antarmuka website yang dirancang untuk memudahkan proses pencarian hingga pemesanan tiket.

**Admin**

Admin adalah pihak yang bertanggung jawab mengelola data di dalam sistem. Melalui dashboard website, admin dapat menginput film-film baru yang akan ditayangkan di bioskop, termasuk detail seperti judul, jadwal tayang, poster, sinopsis, dan harga tiket. Selain itu, admin juga dapat memperbarui informasi film yang ada, seperti mengubah jadwal tayang atau menambahkan slot tiket, serta menghapus film yang sudah tidak ditayangkan. Admin juga memiliki akses ke data transaksi yang tersimpan di database untuk memantau penjualan tiket dan melacak aktivitas di dalam sistem.

**Website**

Website adalah antarmuka pengguna atau user interface yang digunakan baik oleh user maupun admin untuk mengakses fitur-fitur yang tersedia. Bagi user, website menyediakan halaman untuk mencari film, memesan tiket, melakukan pembayaran, dan mengunduh e-ticket. Bagi admin, website menyediakan dashboard khusus untuk mengelola data film, jadwal tayang, serta melihat laporan transaksi. Website ini berfungsi sebagai jembatan utama antara user/admin dengan sistem backend.

**API**

API (Application Programming Interface) adalah komponen yang bertugas menghubungkan permintaan (*request*) dari website (frontend) ke controller di backend. API menerima data dari frontend, meneruskannya ke controller untuk diproses, dan berinteraksi dengan database untuk mengambil, menyimpan, atau memperbarui data. Setelah data diproses, API mengembalikan hasil dalam bentuk *response* yang ditampilkan di website kepada user atau admin. Dengan adanya API, proses komunikasi antara frontend dan backend menjadi lebih efisien dan terstruktur.

**Controller**

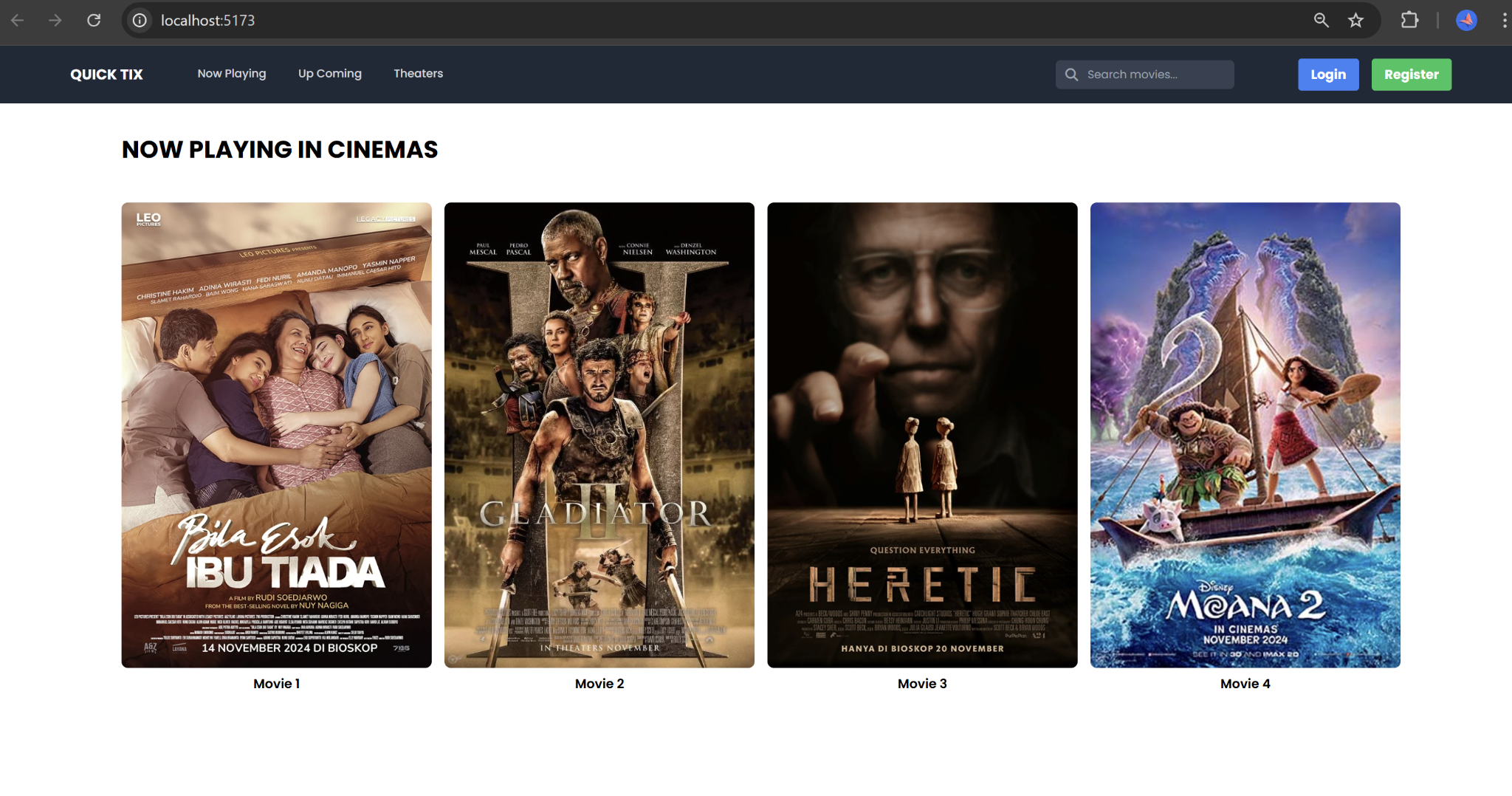
Controller bertanggung jawab untuk mengelola logika bisnis dalam aplikasi. Controller menerima permintaan (*request*) yang dikirimkan melalui API, memprosesnya, dan berinteraksi dengan database sesuai kebutuhan. Operasi yang dilakukan bisa berupa pengambilan data (*GET*), penambahan data baru (*POST*), pembaruan data (*PUT*), atau penghapusan data (*DELETE*). Controller juga mengatur validasi data, pengelolaan error, dan pengembalian hasil (*response*) ke API untuk diteruskan ke website.

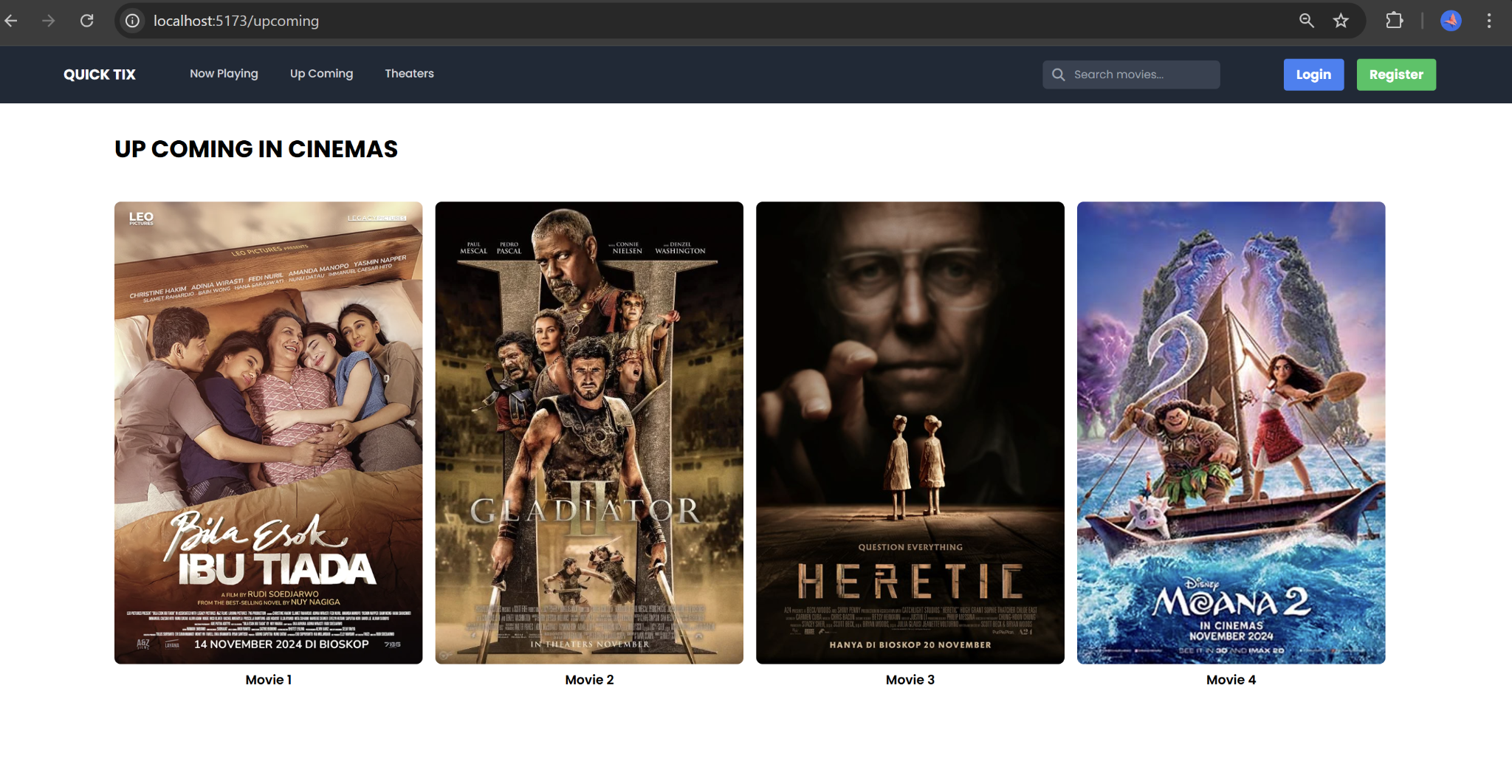
**Database**

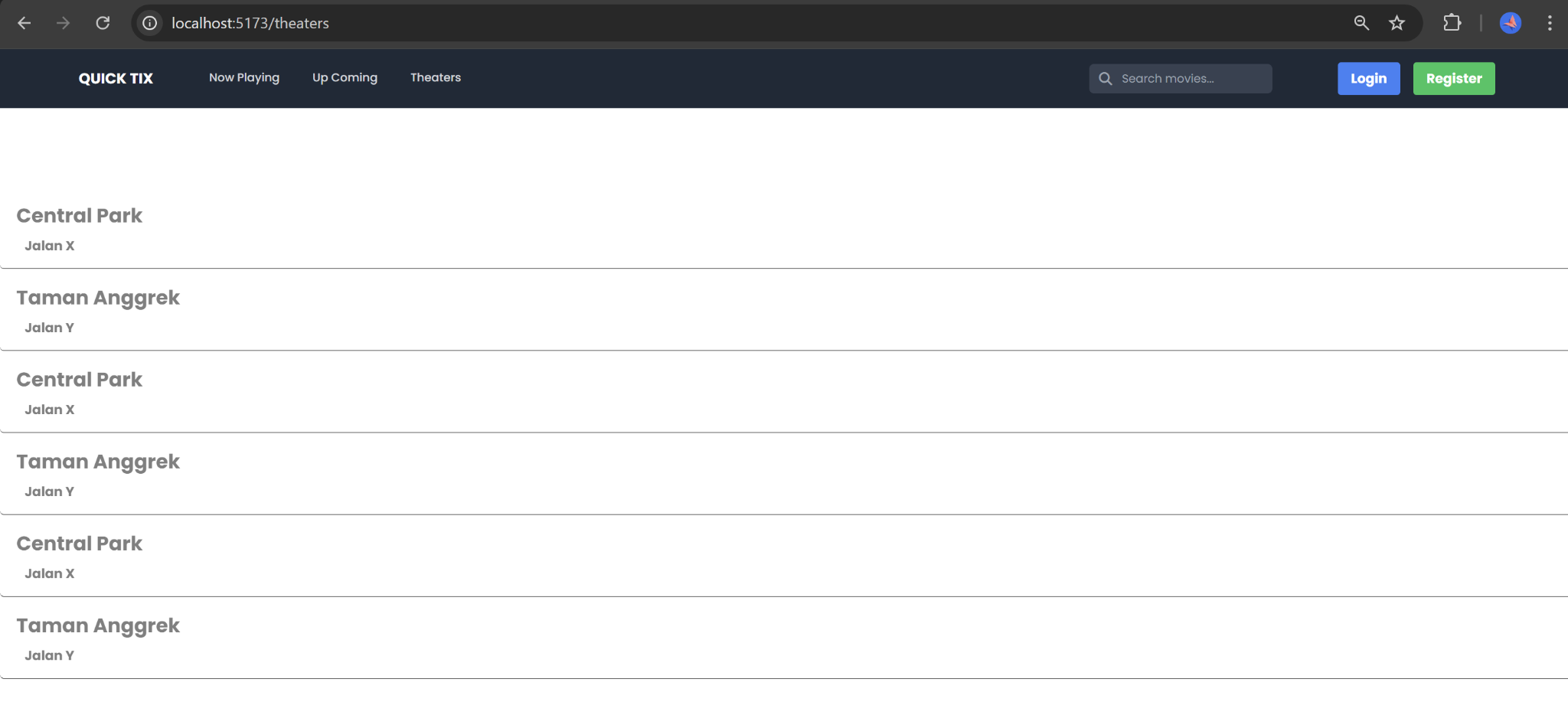
Database adalah tempat penyimpanan utama semua data dalam sistem. Data yang disimpan meliputi informasi user, seperti nama, riwayat pemesanan, dan detail pembayaran, serta data film yang mencakup judul, sinopsis, poster, jadwal tayang, dan harga tiket. Selain itu, database juga menyimpan catatan transaksi yang mencakup semua pemesanan tiket yang dilakukan oleh user. Admin dapat dengan mudah mengakses data di database melalui dashboard untuk memeriksa penjualan tiket atau memantau transaksi yang sedang berlangsung. Database menjadi komponen krusial untuk menjaga integritas dan ketersediaan data di seluruh sistem.

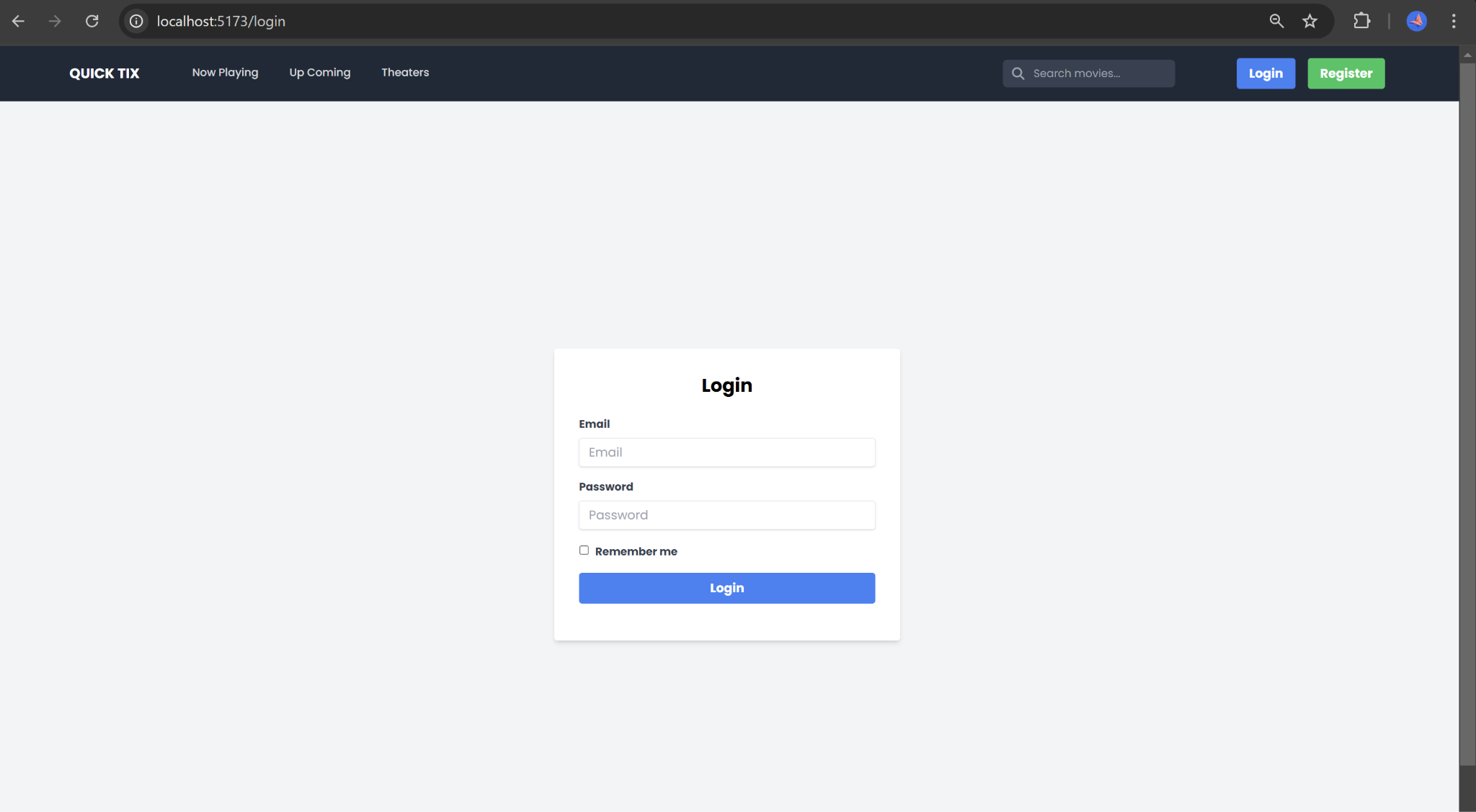
Dalam database architecture tersebut, dapat diliat bahwa semua tabel itu memiliki hubungannya masing masing. Hubungan one to many itu terdiri dari, table\_theaters->table\_seats, tabel\_users->tabel\_bookings, table\_movies->table\_schedules. Dan juga ada hubungan one to one yang dimana terdiri dari, table\_booking->table\_booking\_seats, table\_booking\_seats->seats, table\_bookings->table\_schedules, table\_schedules->table\_theaters.

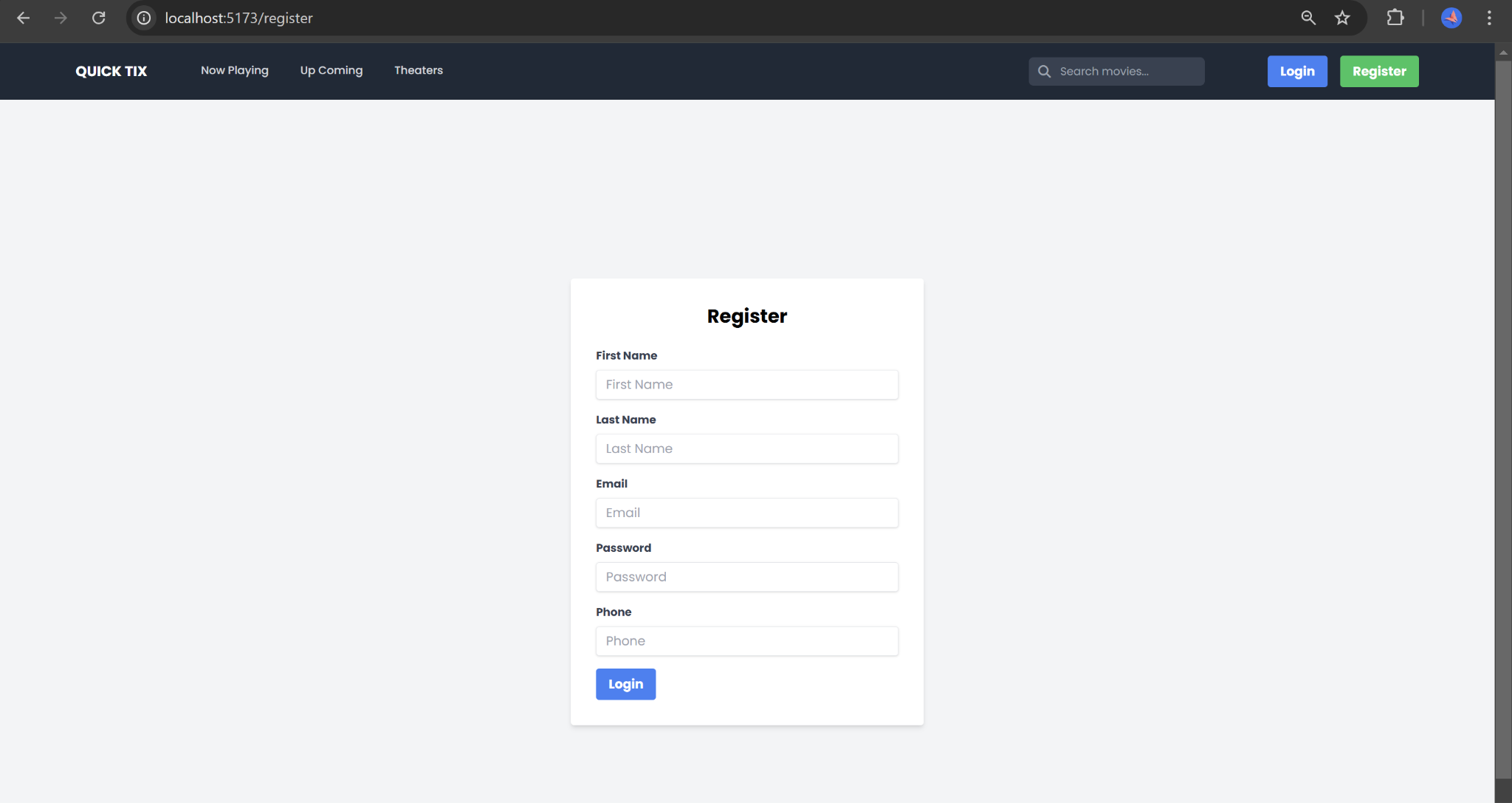
**User Interface**

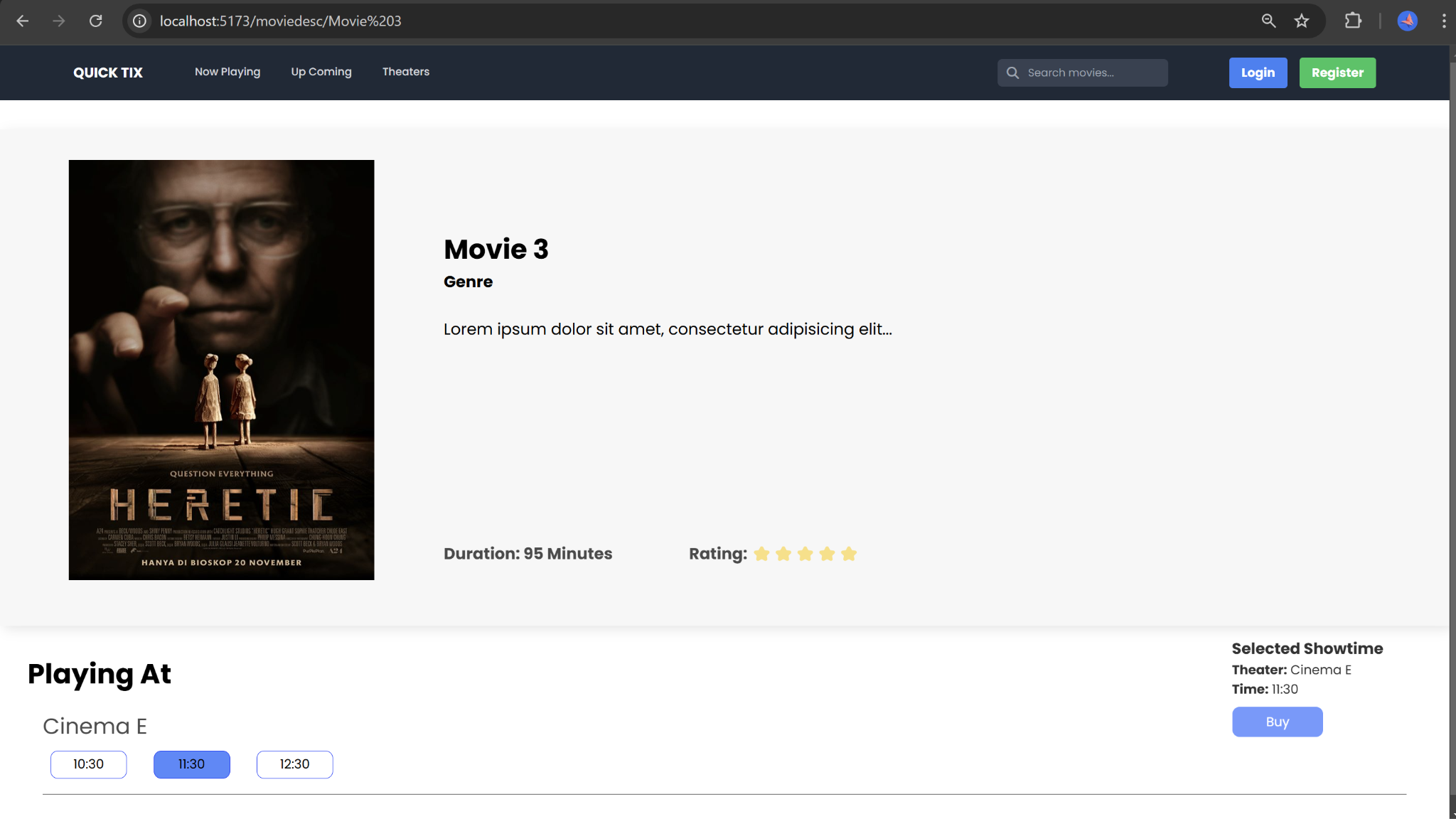




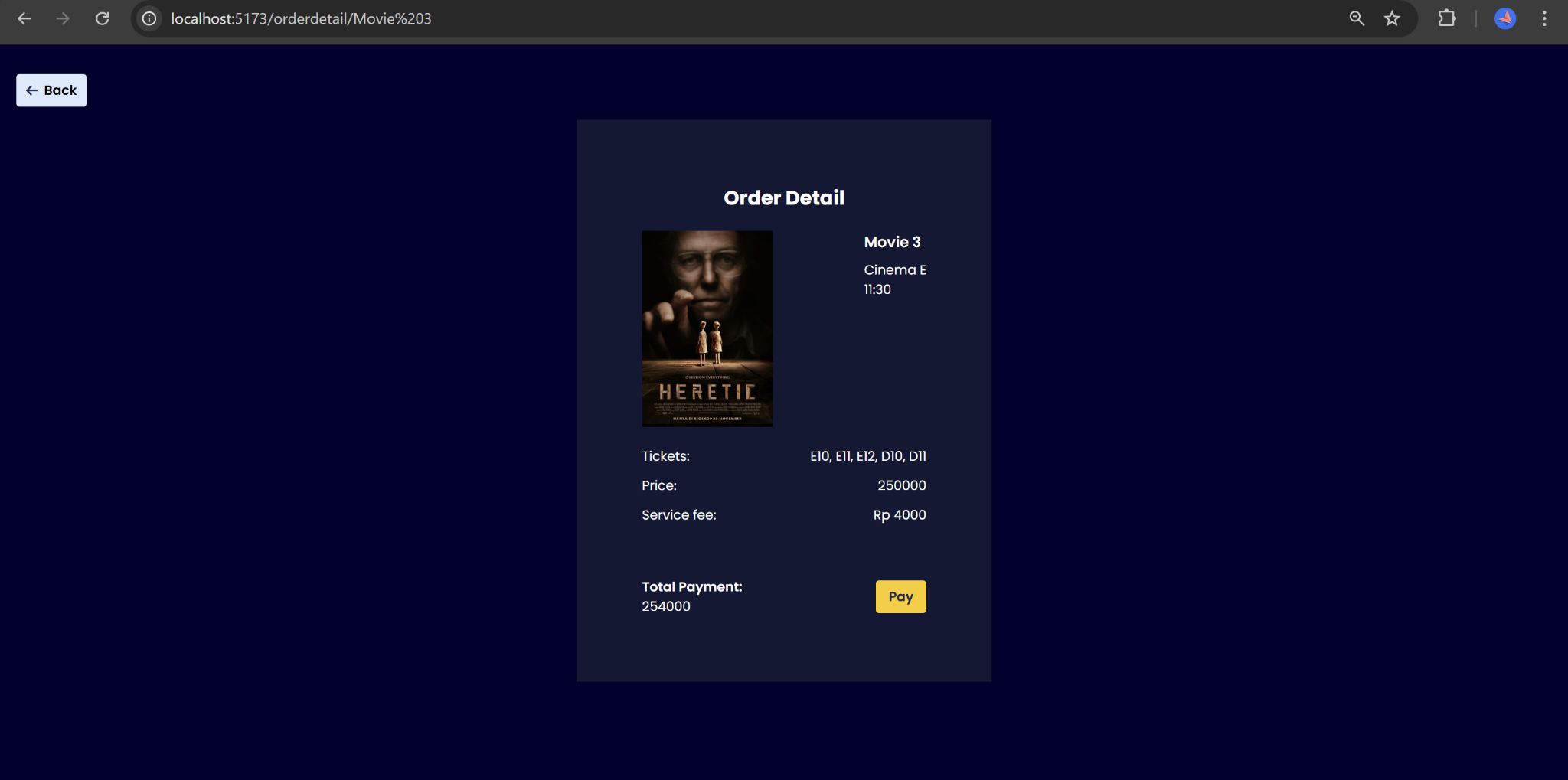




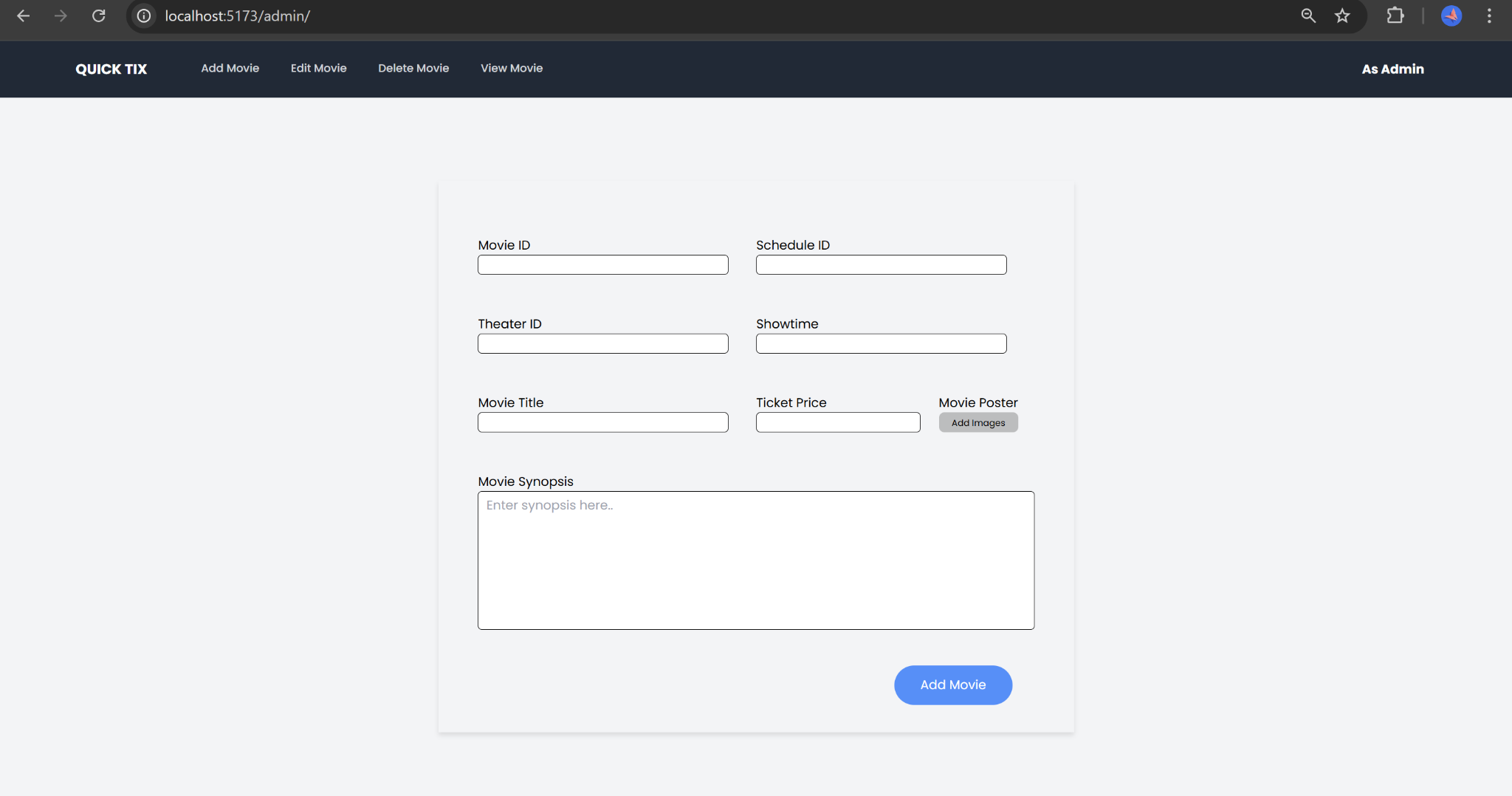


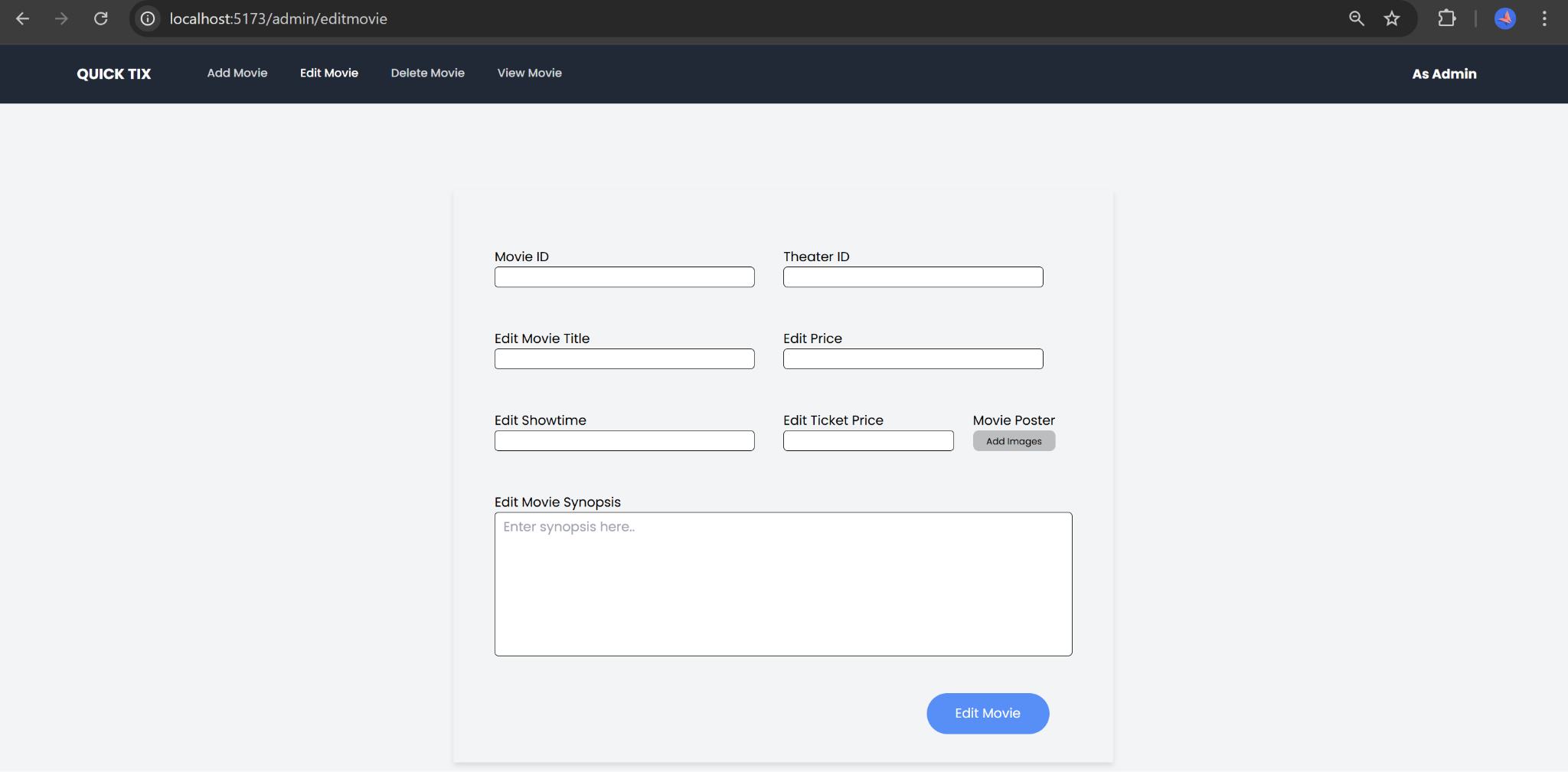


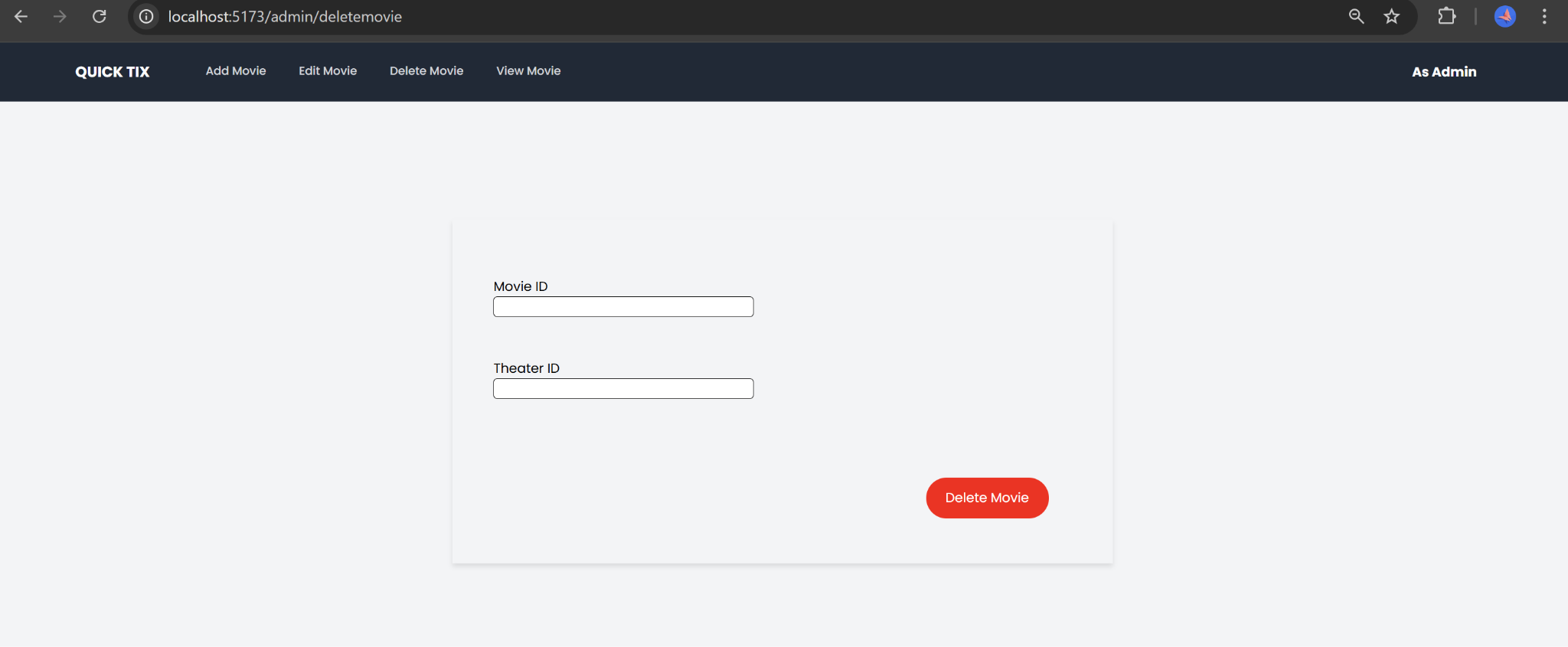


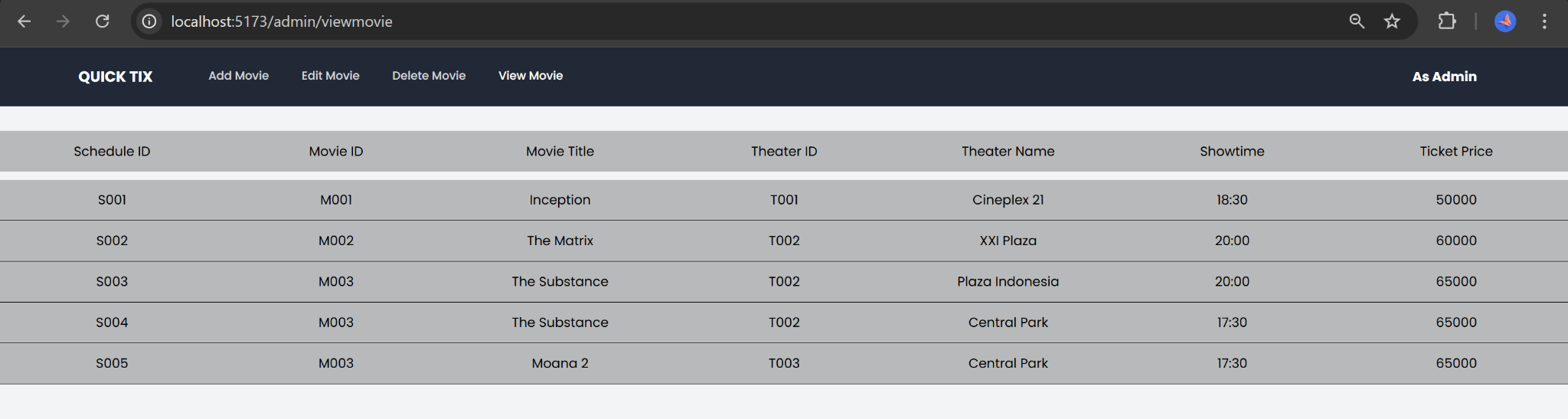












**Kontribusi**

| Nama | Role | Apa yang dikerjakan |
| --- | --- | --- |
| Kenzie Edernez | DataBase Administrator | * Mengerjakan semua bagian Documentation. * Mengerjakan pembuatan ERD. * Mengerjakan pembuatan WebApplicationArchictecture. * Mengerjakan pembuatan DataBase, dimana menggunakan SQL Server. * Mengerjakan pembuatan GitHub untuk melakukan proyek |
| Yoseph Oktavianus Yusanto | Front End Developer | 1 Nov 2024   * Membuat project react typescript menggunakan vite dan install library yang dibutuhkan, diantaranya: tailwind css, react router dom * Membuat komponen Navbar.tsx, NavbarAdmin.tsx, NavbarUser.tsx, dan NavbarGuest.tsx * Membuat halaman login dan register   2 Nov 2024   * Mengaplikasikan routing pada App.tsx * Membuat halaman Now Playing dan UpComing   5 Nov   * Membuat halaman choose seats * Membuat halaman order detail   6 Nov   * Menambahkan state untuk mentransfer data dari movie description ke choose seats dan kemudian ke order detail |
| Andhika Pratama Putra | BackEnd Developer | * Membuat codingan untuk mengakses backend terhadap database * Membuat endpoint untuk mengakses data data yang tersimpan didalam database * Membuat Role based access control dimana hanya user yang sudah login yang bisa melakukan transaksi dan hanya admin yang bisa menambahkan beberapa data seperti movie, theater, seat, dan beberapa hal lainnya |
| Kevin Pangsawira | FrontEnd Developer | * Membuat halaman   MovieDesc(untuk melihat detail film),  Theaters(untuk melihat daftar bioskop yang tersedia) untuk user.   * Membuat halaman MyTicket(untuk melihat riwayat pembelian tiket) kepada user yang sudah login * Membuat halaman   AddMovie,  DeleteMovie, EditMovie, ViewMovie yang di-desain untuk admin untuk memudahkan dalam method GET, POST, PUT, DELETE.   * Membuat komponen   NavbarAdmin,  MallCard (sebagai card yang menampung data-data mengenai Bioskop Mall) , MovieInfo(sebagai card yang menampung data-data mengenai sebuah film) dan TIcketInfo(sebagai card yang menampung data-data mengenai tiket yang dibeli termasuk id, tanggal, lokas).   * Membuat state di komponen MovieCard supaya bisa mengirim data yang dibutuhkan dari film NowPlaying/Upcoming ke MovieDesc untuk menampilkan data film secara detail. |

**Timeline**

|  | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 1 | 2 | 3 | 4 | 5 | 6 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | T | W | T | F | S | S | M | T | W | T | F | S | S | M | T | W | T | F |
| Setup GitHub for code |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Make Database |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Membuat project react typescript |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Membuat navbar untuk guest, admin, dan user |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Membuat halaman login dan register |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Mengaplikasikan routing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Membuat halaman now playing dan up coming |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Membuat halaman choose seats dan order detail |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Menambahkan state untuk transfer data dari movie description sampai order detail |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Membuatkan codingan untuk mengakses ke database |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Membuat endpoint dan set up access control |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Membuat halaman AddMovie, EditMovie, DeleteMovie, ViewMoviedl |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Membuat komponen MallCard, MovieInfo, TicketInfo |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Membuat state di komponen MovieCard |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Membuat halaman MovieDesc, Theaters, MyTicket |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Kesimpulan**

Aplikasi yang kami kembangkan memiliki fitur menarik dan berfungsi dengan baik pada sisi frontend maupun backend. Pada backend, kami telah menerapkan sistem otorisasi yang memastikan pengguna hanya dapat mengakses aplikasi jika login dan password benar, serta mewajibkan login untuk melakukan transaksi pembelian tiket film. Pengguna hanya memiliki akses untuk login, registrasi, dan pemesanan tiket, sedangkan admin memiliki akses khusus untuk menambahkan daftar film yang akan tayang atau sedang tayang ke dalam aplikasi. Dengan pembagian peran yang jelas antara pengguna dan admin, aplikasi ini dirancang untuk memberikan pengalaman yang aman, efisien, dan terorganisasi dengan baik.