```
<meta name="hovercard-subject-tag" content="repository:850140199" data-turbo-transient>
```

```
<meta name="google-site-verification" content="Apib7-x98H0j5cPqHWwSMm6dNU4GmODRoqxLiDzdx9I">
```

```
<meta name="user-login" content="">



<meta name="viewport" content="width=device-width">



  <meta name="description" content="Movie Ticket Booking System. Contribute to sakshamhooda/CineSeat development by creating an accou

  <link rel="search" type="application/opensearchdescription+xml" href="/opensearch.xml" title="GitHub">

<link rel="fluid-icon" href="https://github.com/fluidicon.png" title="GitHub">
<meta property="fb:app_id" content="1401488693436528">
<meta name="apple-itunes-app" content="app-id=1477376905, app-argument=https://github.com/sakshamhooda/CineSeat/blob/main/README.md"

  <meta name="twitter:image:src" content="https://opengraph.githubassets.com/c1afbc9dd270292611eb5bec1ba83117d43318053004c5aad001591
```

```
  <meta name="hostname" content="github.com">



    <meta name="expected-hostname" content="github.com">
```

```
  <meta name="turbo-cache-control" content="no-cache" data-turbo-transient>
<meta data-hydrostats="publish">
```

```
<meta name="turbo-body-classes" content="logged-out env-production page-responsive">
```

```
<div class="position-relative js-header-wrapper ">
  <a href="#start-of-content" data-skip-target-assigned="false" class="px-2 py-4 color-bg-accent-emphasis color-fg-on-emphasis show-

  <span data-view-component="true" class="progress-pjax-loader Progress position-fixed width-full">
<span style="width: 0%;" data-view-component="true" class="Progress-item progress-pjax-loader-bar left-0 top-0 color-bg-accent-emphas
```

<react-partial partial-name="keyboard-shortcuts-dialog" data-ssr="false"

```
  <div hidden="hidden" data-view-component="true" class="js-stale-session-flash stale-session-flash flash flash-warn flash-full">

    <svg aria-hidden="true" height="16" viewBox="0 0 16 16" version="1.1" width="16" data-view-component="true" class="octicon octic
<path d="M6.457 1.047c.659-1.234 2.427-1.234 3.086 0l6.082 11.378A1.75 1.75 0 0 1 14.082 15H1.918a1.75 1.75 0 0 1-1.543-2.575Zm1.763
```

```
<button id="icon-button-4ade6542-568e-480e-815b-94cb12ba906e" aria-labelledby="tooltip-36addcc4-6bd6-459e-aaaf-6d86f352fd97" type="bu
<path d="M3.72 3.72a.75.75 0 0 1 1.06 0L8 6.94l3.22-3.22a.749.749 0 0 1 1.275.326.749.749 0 0 1-.215.734L9.06 8l3.22 3.22a.749.749 0
```

```
<div id="js-flash-container" class="flash-container" data-turbo-replace>
```

```
<include-fragment class="js-notification-shelf-include-fragment" data-base-src="https://github.com/notifications/beta/shelf"></inclu
```

```
  <div class="d-flex flex-nowrap flex-justify-end mb-3  px-3 px-lg-5" style="gap: 1rem;">

    <div class="flex-auto min-width-0 width-fit">
```

sakshamhooda (/sakshamhooda) / **CineSeat (/sakshamhooda/CineSeat)**

```
<span></span><span class="Label Label--secondary v-align-middle mr-1">Public</span>
```

```
    </div>

    <div id="repository-details-container" class="flex-shrink-0" data-turbo-replace style="max-width: 70%;">
        <ul class="pagehead-actions flex-shrink-0 d-none d-md-inline" style="padding: 2px 0;">
```


- Notifications (/login?return_to=%2Fsakshamhooda%2FCineSeat)
- Fork 0 (/login?return_to=%2Fsakshamhooda%2FCineSeat)
- Star 0


```
    </div>
  </div>

    <div id="responsive-meta-container" data-turbo-replace>
```

```
      <nav data-pjax="#js-repo-pjax-container" aligned-label="Repository" data-view-component="true" class="js-repo-nav js-sidenav-conta
```

Code

```
          <svg aria-hidden="true" height="16" viewBox="0 0 16 16" version="1.1" width="16" data-view-component="true" class
<path d="M8 9.5a1.5 1.5 0 1 0 0-3 1.5 1.5 0 0 0 0 3Z"></path><path d="M8 0a8 8 0 1 1 0 16A8 8 0 0 1 8 0ZM1.5 8a6.5 6.5 0 1
```

<react-app app-name="react-code-view" initial-path="/sakshamhooda/CineSeat/blob/main/README.md" style="display: block; min-height: calc(100vh - 64px)" data-ssr="true" data-lazy="false" data-alternate="false"

main **CineSeat (/sakshamhooda/CineSeat/tree/main)** / **README.md**

(/sakshamhooda/CineSeat/commits/main/README.md)

177 lines (136 loc) · 6.52 KB

Preview Code Blame

CineSeat

Movie Ticket Booking System

For latest update, Please Click Here (link to updated repository) (https://www.github.com/sakshamhooda/CineSeat)

System Design

- **Frontend:** A user interface for customers to browse movies, view showtimes, and make reservations.
- **Backend:** A REST API that handles business logic, processes user requests, and interacts with the database.
- **Database:** A relational database to store information about movies, theaters, showtimes, users, and reservations.
- **Admin Dashboard:** An interface for theater administrators to manage movies, schedules, manage reservations manually and view booking statistics.

Architecture Diagram

High Leveel System Architecture of CineSeat (/sakshamhooda/CineSeat/blob/main/images/CineSeat_SystemArchitecture-2024-08-31-130640.png)

Database Schema

The database is designed to manage the following entities:

- **Users:** Stores user information such as username, email, and password.
- **Movies:** Contains details about the movies being shown, including title, genre, and duration.
- **Theaters:** Holds information about the theaters, including location and number of screens.
- **Showtimes:** Links movies to theaters at specific times.
- **Reservations:** Tracks bookings made by users, including seat selection and payment status.

ER Diagram

ER diagram for CineSeat's DB (/sakshamhooda/CineSeat/blob/main/images/CineSeat_DB-ER_diagram-2024-08-31-131511.png)

Database Schema Definition

```
CREATE TABLE users ( id INT PRIMARY KEY, username VARCHAR(50), email VARCHAR(100), password VARCHAR(100) );
CREATE TABLE movies ( id INT PRIMARY KEY, title VARCHAR(100), genre VARCHAR(50), duration INT ); nu
```

API Design

The Movie Booking System exposes the following RESTful APIs for interaction between the frontend and backend:

```
| Method | Endpoint          | Description                                          |
|--------|-------------------|------------------------------------------------------|
| GET    | /api/movies        | Retrieves a list of all movies                       |
| GET    | /api/movies/{id}   | Retrieves details of a specific movie by ID          |
| POST   | /api/reservations  | Creates a new reservation                            |
| GET    | /api/reservations  | Retrieves all reservations for the logged-in user    |
| POST   | /api/auth/login    | Authenticates a user and returns a JWT token         |
| POST   | /api/auth/register | Registers a new user                                 |
```

Frontend-Backend Interaction

The frontend interacts with the backend via these APIs by sending HTTP requests. For example, when a user selects a movie and makes a reservation, the frontend sends a POST request to the `/api/reservations` endpoint with the reservation details.

Component Design

The system is divided into the following components:

- **Frontend:** Built with React.js, responsible for rendering the user interface and making API calls to the backend.
- **Backend:** Developed using Node.js with Express, handling business logic, API endpoints, and communication with the database.
- **Authentication:** JWT-based authentication using Passport.js to secure the API endpoints.
- **Payment Gateway:** Integration with a payment gateway (e.g., Stripe) to handle payments securely.

Each component is designed to be modular and scalable, allowing for easy updates and maintenance.

Scalability and Performance

To ensure the system can handle increased traffic, the following strategies are planned:

- **Load Balancing:** Use of a load balancer to distribute incoming traffic across multiple backend servers.
- **Caching:** Implementation of caching mechanisms (e.g., Redis) to reduce database load and improve response times.
- **Database Optimization:** Indexing and query optimization in the database to handle large volumes of data efficiently.

Security Design

Security is a critical aspect of the Movie Booking System:

- **User Authentication:** JWT tokens are used for authenticating users, ensuring that only authorized users can access certain endpoints.
- **Authorization:** Role-based access control (RBAC) to restrict access to specific resources based on user roles (e.g., admin, user).
- **Data Security:** All sensitive data, including passwords and payment information, is encrypted. HTTPS is enforced for all communications.
- **Payment Security:** PCI-DSS compliance is ensured for handling payment data securely through the integrated payment gateway.

Deployment Strategy

The application will be deployed using the following strategy:

- **Hosting:** The backend and frontend will be deployed on a cloud provider (e.g., AWS, Heroku).
- **CI/CD Pipelines:** Continuous Integration and Continuous Deployment (CI/CD) pipelines will be set up using GitHub Actions or Jenkins to automate testing and deployment.
- **Monitoring and Logging:** Monitoring tools (e.g., Prometheus, Grafana) and logging solutions (e.g., ELK Stack) will be used to track system performance and diagnose issues post-deployment.

For more details on system design and implementation, please refer to the Detailed Documentation (/sakshamhooda/CineSeat/blob/main/docs/detailed_documentation.md).

```
</main>
```

```
      <footer class="footer pt-8 pb-6 f6 color-fg-muted p-responsive" role="contentinfo" >
```

```
<nav aria-label="Footer">
  <h3 class="sr-only" id="sr-footer-heading">Footer navigation</h3>

  <ul class="list-style-none d-flex flex-justify-center flex-wrap mb-2 mb-lg-0" aria-labelledby="sr-footer-heading">

      <li class="mx-2">
        <a data-analytics-event="{&quot;category&quot;:&quot;Footer&quot;,&quot;action&quot;:&quot;go to Terms&quot;,&quot;label&quot
      </li>

      <li class="mx-2">
        <a data-analytics-event="{&quot;category&quot;:&quot;Footer&quot;,&quot;action&quot;:&quot;go to privacy&quot;,&quot;label&qu
      </li>

      <li class="mx-2">
        <a data-analytics-event="{&quot;category&quot;:&quot;Footer&quot;,&quot;action&quot;:&quot;go to security&quot;,&quot;label&(
      </li>

      <li class="mx-2">
```

```
        <a data-analytics-event="{&quot;category&quot;:&quot;Footer&quot;,&quot;action&quot;:&quot;go to status&quot;,&quot;label&quo
      </li>

      <li class="mx-2">
        <a data-analytics-event="{&quot;category&quot;:&quot;Footer&quot;,&quot;action&quot;:&quot;go to docs&quot;,&quot;label&quot;
      </li>

      <li class="mx-2">
        <a data-analytics-event="{&quot;category&quot;:&quot;Footer&quot;,&quot;action&quot;:&quot;go to contact&quot;,&quot;label&qu
      </li>

      <li class="mr-3" >
```

```
<ghcc-consent id="ghcc" class="position-fixed bottom-0 left-0" style="z-index: 999999" data-initial-cookie-consent-allowed="" data-c
```

```
<template id="site-details-dialog">
```

```
<div class="Popover js-hovercard-content position-absolute" style="display: none; outline: none;">
```

```
<template id="snippet-clipboard-copy-button">
```

```
</div>

<div id="js-global-screen-reader-notice" class="sr-only mt-n1" aria-live="polite" aria-atomic="true" ></div>
<div id="js-global-screen-reader-notice-assertive" class="sr-only mt-n1" aria-live="assertive" aria-atomic="true"></div>
```