

Cinema Reservation System

Objectives

- Implementing a full front end using React
 - Create multiple components with state handling
 - Implement routing
- Implementing a suitable backend for the project
 - Use php scripting to implement a full backend
- Using Database in Web Applications

Introduction

In this project, a customer requests creating an automated reservation system for a Cinema. The customer has defined his requirements for the project.

You are required to design and develop an automated system used in reservation process. The system is used by the Cinema management to keep track of movies' schedules, number of attendees and the registered users.

Requirements

1. Users:

1.1 User Types:

The Cinema Management provided you with the following description of each user type:

1- Site Administrator

The Site Administrator is an IT Engineer that manages the users' authorities. He also manages adding movies schedules (adding, deleting, etc..).

2- Customer

He/she is an end-user that had registered his personal data. He/she could reserve any available seats at any movie.

3- Guest

He/she is an unregistered user or didn't login. He/she could login, register.

1.2 User Data:

Each user should have the following **personal data**:

- 1- Username (Should be unique)
- 2- Password
- 3- First Name
- 4- Last Name

- 5- Birth Date
- 6- Email (Should be unique)
- 7- Type (admin/customer)

1.3 **User Features:**

Each user type has the following features:

- Site Administrator
 - Add a new movie
 - Add a new Movie with all its details.
 - Add a movie screening
 - For an existing movie, add a movie screening time.
- Customer
 - See the available movies
 - The customer could see available movie details
 - For each movie screening, the customer could see vacant seats for the movie
 - Reserve vacant seat/s in an available movie screening
 - The customer could select vacant seat/s only
 - Then the reservation is confirmed and a reservation ticket number (unique) is generated
 - Reservation should be done graphically where each seat is colour coded whether it is reserved or not.
 - Conflicts should be resolved by assigning each chair to only one customer, if two customers try reserving the same seat only one should get it.
 - Dynamically update the seat graph whenever a seat is reserved without needing to refresh the page.
- Guest
 - Register a new account
 - The guest could register a new account
 - Login an existing account
 - The guest could login to an existing account

2. **Screens:**

Each Screen should have the following **details:**

- 1- Screen Number (Unique)
- 2- Rows
- 3- Columns

3. **Movies**

Each movie should have the following **details:**

- 1- Movie Name
- 2- Genre
- 3- Screen where the movie is shown
- 4- Movie Length
- 5- List of date-time when movie is shown (Screening).

The main webpages that must exist:

- Website Homepage (The main page for the Cinema)
- Login Page (The user should enter his username & password)

- For each feature decide whether it is a stand-alone page or a function embedded in another page.

Implementation Guidelines:

- Implement a full front-end pages with React JSX
 - Use the steps of design we went over. Start with a mock website.
 - Use react router to route different webpages (Components you created)
 - Use css and styling to design an appealing UI to your website. (Using bootstrap will be a bonus).
 - Use Ajax to send requests to your back-end server (Optional, you can use any other library to create your requests.)
- Implement a full back-end php scripts
 - Implement an API for each needed request by your client side.
 - Implement a simple routing for each needed request.
 - Using an authentication library like OAuth will be a bonus.
- Implement a Database to store your data (MYSQL for example)
- Your frontend code must be completely separated from your backend. They should only communicate through post or get methods.

Teams

The teams should be 2-3 members in each team (not more or less). **Deadline:** Prepare and deliver a spreadsheet with the teams & names by Dec 6th, 2019.

Final Delivery

Deadline: Dec 24th, 2019 at the section time.

- You will be required to deliver a .rar file a Google drive form (link will be shared by email). The .rar file should contain
 - A .txt file named ID.txt with your names, sections and BNs.
 - A folder named frontend containing your React project
 - A folder named backend containing your php scripts
 - A sql file (or any script) that does the following
 - Create all tables
 - Adds one admin account
 - Adds some dummy screens and movies with schedules
 - A .txt file named login.txt containing the login information of that admin account.

Note: Complete Projects from the internet will get zero.