Cinema Management System

|  |  |  |  |
| --- | --- | --- | --- |
| name | Dep | ID | Section Number |
| Youssef Edress Mohamed | SC | 2022170509 | 6 |
| Mazen Alaa Saeed | SC | 2022170336 | 4 |
| Youssef Mahmoud Ibrahim | SC | 2022170528 | 6 |
| Mohamed Mahmoud Abd El Twab | SC | 2022170618 | 5 |
| Mohamed Hussien Ragab | SC | 2022170581 | 4 |
| Mazen Mohamed Abd El Aal | SC | 2022170337 | 4 |
| Mazen Hassan Youssef | CS | 2022170335 | 4 |

T.A / Ramadan Wasfi

Introduction :

1. Introducing our Movie Management System with simple interface for users to register with their emails while applying a strong password security policy to keep our users safe , after Registration the user is allowed to search and see films , movies and series , with the ability to react and comment with their thoughts , also having the ability to rate the films , actor and series .

User Requirements :

* Customer can search for movies.
* Customers can extend films menu.
* Customer can Register with their emails.
* Customer can Login with their Credentials.
* Customer can see their history .
* Customers can Manage Their Accounts
* Customers can Change their Password
* Customers can Verify their accounts
* Customers can Reset their passwords
* Customers can React on Films
* Customers can Comment on Films

User Requirements :

* Admin can Add Users
* Admin can Edit Users
* Admin can Delete Users
* Admin can Add Movies
* Admin can Edit Movies
* Admin can Delete Movies
* Admin can Add Schedule
* Admin can Edit Schedule
* Admin can Delete Schedule

Functional Requirements :

**Customer Functions**

**Functional Requirement Name: Register**

**- Description/Action: Customer creates an account using their email.**

**- Requirements/Inputs: Email, password, name.**

**- Source: Customer input via registration form.**

**- Pre-condition: Email is not already registered.**

**- Post-condition: Customer account is created.**

**- Output: Success message and verification email.**

**-----------------------------------------------------------**

**Functional Requirement Name: Login**

**- Description/Action: Customer logs in using credentials.**

**- Requirements/Inputs: Email, password.**

**- Source: Customer input via login form.**

**- Pre-condition: Account exists and is verified.**

**- Post-condition: Customer gains access to the system.**

**- Output: Dashboard view.**

**-----------------------------------------------------------**

**Functional Requirement Name: Search Movies**

**- Description/Action: Customer searches for movies by title, genre, or date.**

**- Requirements/Inputs: Search keywords/filters.**

**- Source: Customer input via search bar.**

**- Pre-condition: Customer is logged in.**

**- Post-condition: Matching movies are displayed.**

**- Output: Filtered movie list.**

**-----------------------------------------------------------**

**Functional Requirement Name: View History**

**- Description/Action: Customer views booking and interaction history.**

**- Requirements/Inputs: Customer ID.**

**- Source: Automated system retrieval from database.**

**- Pre-condition: Customer is logged in.**

**- Post-condition: History data is displayed.**

**- Output: List of past bookings, reviews, and comments.**

**-----------------------------------------------------------**

**Functional Requirement Name: Manage Account**

**- Description/Action: Customer updates account details (e.g., email, password).**

**- Requirements/Inputs: Updated fields (e.g., new password).**

**- Source: Customer input via account settings.**

**- Pre-condition: Customer is logged in.**

**- Post-condition: Account details are updated.**

**- Output: Confirmation message.**

**-----------------------------------------------------------**

**Functional Requirement Name: Review Movies**

**- Description/Action: Customer reacts to or comments on a movie.**

**- Requirements/Inputs: Movie ID, rating (e.g., 5 stars), comment text.**

**- Source: Customer input via movie details page.**

**- Pre-condition: Customer has booked the movie.**

**- Post-condition: Review is added to the database.**

**- Output: Published review on the movie page.**

**-----------------------------------------------------------**

**Security & Verification**

**Functional Requirement Name: Verify Account**

**- Description/Action: Customer verifies email via a confirmation link.**

**- Requirements/Inputs: Verification token from email.**

**- Source: Customer clicks email link.**

**- Pre-condition: Customer has registered but not verified.**

**- Post-condition: Account status is set to "verified".**

**- Output: Success message and full system access.**

**-----------------------------------------------------------**

**Functional Requirement Name: Reset Password**

**- Description/Action: Customer resets forgotten password via email.**

**- Requirements/Inputs: Registered email address.**

**- Source: Customer input via "Forgot Password" page.**

**- Pre-condition: Account exists and is verified.**

**- Post-condition: Temporary password is sent to email.**

**- Output: Password reset email.**

**-----------------------------------------------------------**

**User Roles:**

**- Admin**

**- Customer**

Non-Functional Requirements :

* Performance
  + Our System can deal with more than 1500 request at the same time
* Security
  + Our System is using strong security policy to keep our clients secure
  + Our System uses https instead of http traffic to keep our data encrypted
  + Our System affords Multi-Factor Authentication to help our users in case they forgot they passwords

Non-Functional Requirements :

* Reliability
  + Our system is integrated with cloud solutions so that we are able to be available by 99.9% all over the year
* Scalability
  + Our system is designed to be scalable all the time by using Forward Proxies which helps in load Balancing
* Usability
  + Our system is designed with simple User Interface to achieve a good user Experience while using the System



