Ticket Booking – Report

Student ID: 21f1005885

Course: Modern Application Development – II Project

Term: May-2023

Introduction:

Book For You- is a web application that is more similar to other ticket booking platforms like Book my show. It's a multiuser web application. By creating an account on this, users can see all the movies in their current location and also see the available theatres. By selecting a movie or theatre users are able to see the list of shows from there they can book a ticket by typing the number of the ticket.

Users can Download or cancel the existing bookings that they booked.

Technical Details:

- **Libraries:**
  - Flask, Flask SQL alchemy, Flask Security, Flask_cors,Flask_jwt_extended,Celery,Flask Mail, Flask Restful – To handle the backend
  - Vue CLI- is used for the frontend development
  - Redis -is used for caching as well as scheduling celery
  - Celery -is used for backend async jobs like sending daily reminders, monthly reminders, exporting csv
- **Basic Endpoints:**
  - **/** - Home page of the website.
  - **/registration** – User registration page. By providing basic information about the user, they can create an account on this page.
  - **/user/home** – User's home page. Here they can see all the Theatres Movies and Shows.
  - **/user/home/admin** - Admin Page to manage Movies, Theatres, and Shows
  - **/user/theatre**– List of available theatres
  - **User/movies** – List of movies currently available for booking
  - **User/booking** – List of bookings done by the user.
  - **User/admin/create/movie or theatre or show**– Admin to add a new movie or theatre or Show
- DB Schema Design:
  - **User**
    - Full name – User's Full name (Required)
    - User_name – User's unique name (Required and Unique)
    - Password – User's login password (Required and should be greater than or equal to 7)
    - E-mail – User's E-mail ID
    - Admin – Admin privilages.
  - **Movies**:
    - Movie Id –movie Id – (primary key auto increments)
    - Title –Movie title – (required and minimum 2 characters)
    - tags – Movie genre – (Required and minimum 2 characters)
    - Image –Movie cover image – (Optional)
    - Rating – User ratings
    - Rating_count - Total rating count.
    - Language – Movie's language.
  - **Venue**
    - Venue_id – Venue ID (Required and Primary Key)
    - Venue_name – Name of the venue (Required)
    - place – Place or the city of the venue(Required)
    - location – Landmark for the venue (Required)
    - Capacity – Number of seats available(Required)
  - **Shows**:
    - show_id – show id (Required)
    - venue – venue id which it's linked to (required foreginkey)
    - movie–Movie id which it's linked to (required foreginkey)
    - show_name – Title of the show (Required)
    - show_date– show date (Required)
    - show_time – Timing of the show (Required)
    - price – Price for each ticket (Required)

- **Booking**
  - booking_id –Booking ID (Required and Primary Key)
  - Venue – venue ID which it's linked to (required foreign key)
  - user –User ID which it's linked to (required foreign key)
  - movie –Movie ID which it's linked to (required foreign key)
  - show –show ID which it's linked to (required foreign key)
  - booking_count – Number of tickets booked (required foreign key)
  - Total – Total amount for the booking

API Design:

- **User**
  - get /api/user/{user_name} –to get the user details
  - put /api/user/{user_name} – to edit the user details
  - Delete /api/user/{user_name} –to remove a user
  - post /api/user–create a new user
- **Venue**
  - get/api/venue/{venue_name} –to get the venue details
  - put/api/venue/{venue_name} – to edit the Venue details
  - delete/api/venue/{venue_name} –to remove a venue
  - post/api/venue –create a new Venue
- **Movie**
  - get /api/venue/{movie _name} –to get the Movie details
  - put /api/venue/{movie _name} – to edit the Movie details
  - delete/api/venue/{movie _name} –to remove a Movie
  - post /api/venue - create a new Movie
- **Show**
  - get /api/venue/{ show_name}–to get the show details
  - put /api/venue/{ show_name}–– to edit the show details
  - delete/api/venue/{ show_name}– –to remove a show
  - post /api/venue–create a new show
- **Booking**
  - get/api/venue/{ booking_ID} –to get the user Booking
  - put /api/venue/{ booking_ID}– to edit the user Booking
  - delete /api/venue/{ booking_ID}–to remove a Booking
  - post /api/venue–create a new Booking

Architecture And Features:

For the frontend, we have used vue-cli due to which our frontend is stored in the frontend folder of the application. In the front-end inside src components contain all our frontend contain all our components and we have not created views separately. The Router folder contains all our implementation of routing while the store folder contains the vuex file.

Our backend apis are inside resource.py. We also have a requirement.txt and readme file Token authorization for the user login, Basic CRUD functionality for movies and theatres, Ability to search movies and theatres, Images can be uploaded to Movies as a cover picture, Exports option for Bookings, sending daily reminder if the user hasn't Booked, Sending Monthly engagement report to the user, Caching has been done to speed up the loading.

Following are some actions users/admins can do on the application.

- Admins are having the privilege to Create, Edit, and Delete movies, Theatres, and Shows.
- To book a ticket users can search for their interested movies or theatre and book the ticket based on available shows.
- Users have the control to cancel the ticket that they have booked.
- Users can add their ratings to their favourite movies.
- User can download their tickets as a pdf.
- Admin can download the movie and show activity details.

Video:

https://drive.google.com/file/d/1k-YT4VY0LBAJ74TREhRxWiI2zz04VMus/view?usp=sharing