Ticket (#01): Establish user authentication for login

Description:

Making a working login system for the website

Acceptance Criteria:

1. User can sign up
    a. As user
2. User can login
    a. As user
    b. As admin

Assigned: Abdullah
PR: https://github.com/HamzaHassan02/CSCI4230U-Movie_Website/pull/2
Status: **DONE**

Follow-up Ticket (#02): Unit tests for login

Description:

Create unit tests for login
Ex. Structure

```
Project/
│
├── tests/
│   ├── unit/
│   │   ├── test_auth.py
```

Acceptance Criteria:

1. Test signup
2. Test successful user login
3. Test successful admin login
4. Test invalid password
5. Test not registered user

Assigned: Abdullah
PR:
Status: **DONE**

Ticket (#03): Get movie data from API

Description:

Get movie data from API

Acceptance Criteria:

1. Pull movie data from API
2. Replace dummy movie data with info from API

Assigned: Hamza
PR:
Status: **DONE**

Ticket (#04): Improve movie data presence

Description:

Update movie detail page to match design. Add movie data to the Booking Page so movie data and booking input can be extracted from the page.

*NOTE: This is a frontend feature.

Acceptance Criteria:

1. Movie page shows correct details
2. Movie Poster and Title appear in the Booking Page.
3. Improve page design.

Assigned: Hamza
PR:
Status: **DONE**

Ticket (#05): Store inputs from the booking page

Description:

Add logic to access input information. For now just extract inputs, save to a variable and then console log to test. Make the showtime button green when you select it.

Acceptance Criteria:

1. Console logs input on “Confirm Booking” click.
2. +Added Booking logic so now saves booking to databas
3. +Can view bookings in my bookings page

Assigned: Abdullah
PR:
Status: **DONE**

Follow-up Ticket (#06): Unit tests for booking

Description:

Write unit tests and if you have the time an e2e of the booking process.

Acceptance Criteria:

1. Test Successful booking
2. Test missing parameters (for each input)
3. Test edit booking
4. Test canceling booking
5. E2E test for successful booking

Assigned: Abdullah
PR:
Status: **DONE**

Ticket (#08): Add checkout logic for booking

Description:

When the user goes to book a movie, they are taken to a checkout page with the proper UI and functionality

Acceptance Criteria:

1. On click for Confirm Booking user should be taken to the checkout page
2. Check out takes the user's info and card details for successful checkout

Assigned: Abdullah
PR:
Status: **DONE**

Ticket (#07): Add frontend Admin Logic for movies

Description:

Admin should be able to remove movies listed in the home page and should be able to search for movies to add to the now playing list.

Acceptance Criteria:

1. Admin has a manage movie option in their dashboard to take them to new page
2. Search bar appears for admin with movie search logic
3. Movies and be added/removed from the current list of movies
4. Movies listed are in the database and loaded on the homepage for all users to see

Assigned: Hamza
PR:
Status: **DONE**

Ticket (#09): Add endpoints for admin to manage users

Description:

Add backend logic for admin’s access to users.

Acceptance Criteria:

1. Seed Admin into the database
2. Admin should be able to
    a. Get all users
    b. Delete users
    c. Edit user bookings
    d. Delete user bookings

Assigned: Abdullah
PR:
Status: **DONE**

Ticket (#10): Update logic for updating booking

Description:

Add proper update logic for updating the user's bookings

Acceptance Criteria:

1. When the user increases the seat quantity, they should be taken to the checkout page and pay for the additional ticket
2. When the user decreases the seat quantity, they are given a message instead: "X amount has been refunded," where X is the #ticketsDeducted*ticket price.

Assigned: Abdullah
PR:
Status: Ongoing

Ticket (#11): Add UI for Admin dashboard to manage users

Description:

The backend logic for managing users has been implemented now the frontend logic has to be added

Acceptance Criteria:

3. Admin can modify a user's booking
4. Admin can remove a user's booking
5. Admin can remove a user

Assigned: Abdullah
PR:
Status: **DONE**

Ticket (#12): Seed movies and add expiration to movies

Acceptance Criteria:

1. Some movies should be initially seeded into the database
2. Movies should only be able to be booked until the expiration date

Assigned: Hamza
PR:
Status: **DONE**

Ticket (#13): seat availability, new unit tests,

Acceptance Criteria:

1. Remove seat availability from show times
2. ~~When Admin deletes a user's bookings, it will tell the user that their booking was modified/deleted~~
3. Modify E2e tests so that it passes (excluding stripe checkout page)
4. Add new unit tests for the admin, where it tests:
    a. Delete user booking
    b. Edit user booking
    c. Delete user

Assigned: Abdullah
PR:
Status: **DONE**

Ticket: Have a working Docker file

Description:

Seeding is now done via seed.py instead app.py, so Docker does not create an admin user.

Acceptance Criteria:

1. Have seed.py be ran by docker
    a. Ensure seed.py doesn't run every launch, only initially
    b. NOTE: delete database.db before running `python seed.py` when testing
2. Update git workflows

Assigned: Abdullah
PR:
Status: **DONE**

Ticket: Create an AI chatbot for the homepage

Description:

An AI chatbot will be on the homepage and can recommend movies or tell you info about some movies

Acceptance Criteria:

Assigned: Hamza
PR:
Status: **DONE**

Template:

Ticket:

Description:

Acceptance Criteria:

Assigned:
PR:
Status: