**Filmy Villa: Designing an Online Ticket Reservation System**

**A PROJECT REPORT**

***Submitted by***

**Name of Student - Vansh Raturi**

**Registration No. - 12320821**

**Roll No. - 74**

**Section – K24KM**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF COMPUTER SCIENCE and ENGINEERING**



**Lovely Professional University, Punjab**

**CSE326 – Internet Programming Laboratory**

**DECEMBER 2024**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S NO.** | **TOPIC** | **PAGE NO.** |
| **1** | **Project Description** |  |
| **2** | **Project Objectives** |  |
| **3** | **Description[Main].html with Screenshots** |  |
| **4** | **Description[About].html page with Screenshots** |  |
| 5 | **Description[Events].html page with Screenshots** |  |
| 6 | **Description[Dabblet].css page with Screenshots** |  |
| 7 | **Description[Main].css with Screenshots** |  |
| 8 | **Description[Style].css with Screenshots** |  |
| 9 | **Description[Script].js with Screenshots** |  |
| 10 | **Output Screenshots** |  |
| 11 | **Github Repo Link** |  |
| 12 | **Conclusion** |  |
| 13 | **References** |  |

1.Project Description

This project focuses on developing a user-friendly, efficient movie ticket booking website that simplifies the process of reserving cinema tickets online. The website allows users to explore available movies, select seats.Aimed at enhancing the overall user experience, the platform incorporates features such as movie details, seat maps,offering a one-stop solution for cinema-goers.This website built using modern web technologies such as HTML5, CSS3, JavaScript.

2. Objective

**Simplify the Booking Process:** Develop an intuitive, user-friendly platform that allows users to reserve movie tickets online quickly and conveniently, reducing the need for in-person ticket purchases.

**Enhance User Experience:** Incorporate features such as detailed movie information, interactive seat maps, and streamlined navigation to provide a comprehensive and enjoyable booking experience for cinema-goers.

3.

**3. Description of main.html with screenshots**

* **Header Section:** A navigation bar with links to the Booking page and Help & Support page.
* **Main Section:** A dual-pane layout for user interaction:
  + **Login Panel:** Welcomes returning users with a prompt to log in to keep connected.
  + **Registration Panel:** Allows new users to create an account, including social media icons (for Gmail, Instagram, and Facebook) and an account registration form. The form collects the user's name, email, and password, with a submit button to complete registration.

The code is linked to an external stylesheet main.css for styling and Font Awesome icons for the social media icons.

**4. Description of about.html with screenshots**

This HTML code sets up a webpage for "Filmy Villa Event Technology," showcasing the company’s services in event management. The page includes a navigation menu that allows users to access sections like "About," "Our Work," "Our Team," "Careers," and "Contact Us."

The main content features a header with a brief introduction about the company, followed by detailed descriptions of various services offered, including Video, Audio, Lighting, Scenic Design, Creative Services, and Conventions. Each service is represented with an icon and a short explanation, highlighting the technology and expertise available.

The layout is designed to be user-friendly and visually appealing, with responsive design elements to ensure compatibility across different devices. The footer includes social media links for users to connect with the company online.

**5. Description about Events.html page with Screenshots**

This HTML code creates a simple and interactive seat booking interface for a movie theater. Users can select a movie from a dropdown menu, which displays the title and price of each film. The seat layout is visually represented, showing available, selected, and sold seats. The user can easily identify which seats are free and which ones are taken. As users select their seats, a summary updates dynamically to display the total number of seats selected and the overall price. Finally, there is a button to proceed to payment, allowing for a smooth booking experience.

6. **Description Dabblet.css page with Screenshots**

Responsive Design: Media queries ensure the layout adjusts seamlessly across different screen sizes, enhancing usability on mobile devices.

Navigation Menu: The menu is styled for clear visibility, with hover effects for better user interaction. A toggle feature is included for mobile responsiveness.

Banner Styles: Hero images are applied with gradient overlays to create a visually appealing header, and text styles are defined for clarity and emphasis.

Content Formatting: Sections are designed with appropriate spacing, colors, and font sizes to improve readability and organization, with a specific focus on the service offerings.

Footer Design: The footer is styled for clear social media links and copyright information, maintaining consistency with the overall theme.

7. **Description of Main.css with Screenshots**

**Root Variables**: The :root selector sets a custom property --var-primary-color to a specific shade of blue (#2888E8FF), which is used throughout the styles for consistent theming.

**Font Sizing and Family**: The html tag sets the base font size to 62.5%, making it easier to use rem units for responsive design. The headers (h1, h2, h3, h4) utilize the 'Lucida Sans' font family, ensuring readability and a modern look.

**Header and Paragraph Styling**: Each header has defined sizes that establish a clear hierarchy, with the largest being h1 at 4.8rem. Paragraphs have a slightly larger font size (1.6rem) and letter-spacing for better legibility.

**Buttons and Input Fields**: Buttons and submit inputs are styled to have a clean look with a white border and a transparent background. The hover effects change the background color, providing visual feedback to users.

**Registration Page Layout**: The .registration-header class creates a full-screen background image of a movie theater, enhancing the thematic experience. The .container class centers elements and uses flexbox for layout.

**Navigation and Logo**: The logo and navigation links are styled for a cohesive and professional look, using white text against a red background.

**Main Content**: The main section is designed to be flexible, ensuring it adjusts to different screen sizes. The .registration class adds a background image and ensures content is centrally aligned.

**Form Styling**: The registration form has a structured grid layout for inputs and buttons. Each input has a light gray background, and the submit button stands out with the primary color defined earlier.

**Social Media Icons**: The social media icons are styled to be round with a border, making them visually distinct and easy to interact with.

**8. Description Style.css with Screenshots**

**Global Styles**:

* The universal selector (\*) sets all elements to use box-sizing: border-box, ensuring that padding and border are included in the element's total width and height.
* The body has a dark background color (#242333) and white text color, providing a modern aesthetic. It also uses Flexbox to center the content both vertically and horizontally, making the layout responsive and visually appealing.

**Movie Container**:

* The .movie-container class adds vertical spacing to elements within it. The dropdown selector in this container is styled with a white background and rounded corners, making it easy to interact with.

**3D Perspective Effect**:

* The .container class applies a perspective effect, which adds depth to the 3D layout of the seat selection interface.

**Seat Styling**:

* The .seat class represents individual seats in the theater. Each seat has a dark background and rounded corners. When selected, a seat changes to green, and sold seats are marked in white. The hover effect enlarges available seats slightly, indicating that they can be selected.

**Showcase of Seats**:

* The .showcase class presents information about the seats. It uses a flexible layout to arrange elements and includes styles for individual seat indicators. The seats in the showcase have a light background, making them stand out against the dark theme.

**Screen and Row Layout**:

* The .screen class represents the movie screen, featuring a white background and a rotated effect to simulate a theater view. The .row class utilizes Flexbox to align seats in a horizontal line.

**Text Styles**:

* Paragraphs with the class .text are used for additional information, with special spans to highlight selected text in a different color.

**Submit Button**:

* The submit button is styled to be prominent with a blue background and white text. It adjusts its size based on content while ensuring consistent appearance across various devices.

**9. Description Script.js with Screenshots**

**Element Selection**:

* The code begins by selecting important elements from the HTML document using document.querySelector and document.querySelectorAll. These include the main container for the seating layout, all available seats, the count of selected seats, the total price display, and the movie selection dropdown.

**Initial Setup**:

* The populateUI function is called at the start to populate the user interface with previously selected seats and the selected movie. This ensures that the user's previous selections are displayed if they return to the page.

**Ticket Price Management**:

* The initial ticket price is set based on the selected movie from the dropdown menu. The ticket price is extracted from the value of the movieSelect element.

**Local Storage Functions**:

* **setMovieData(movieIndex, moviePrice)**: This function saves the selected movie's index and price to local storage, allowing the information to persist even if the page is refreshed or reopened.
* **populateUI()**: This function retrieves the stored seat selections and movie index from local storage. If seats were previously selected, they are highlighted by adding the "selected" class. The movie dropdown is also set to the previously selected movie.

**Update Selected Count and Total**:

* **updateSelectedCount()**: This function calculates and updates the count of selected seats and the total cost. It selects all seats marked as "selected" and retrieves their indices to save them in local storage. The total price is calculated by multiplying the number of selected seats by the ticket price.

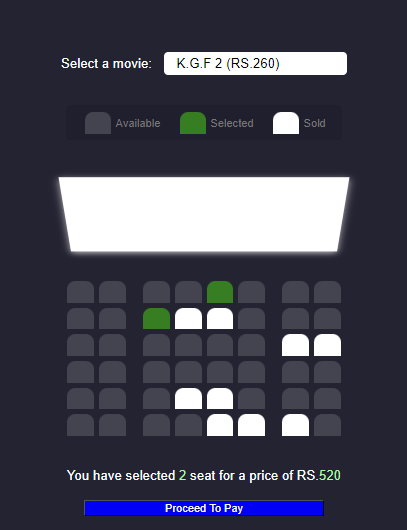
**Event Listeners**:

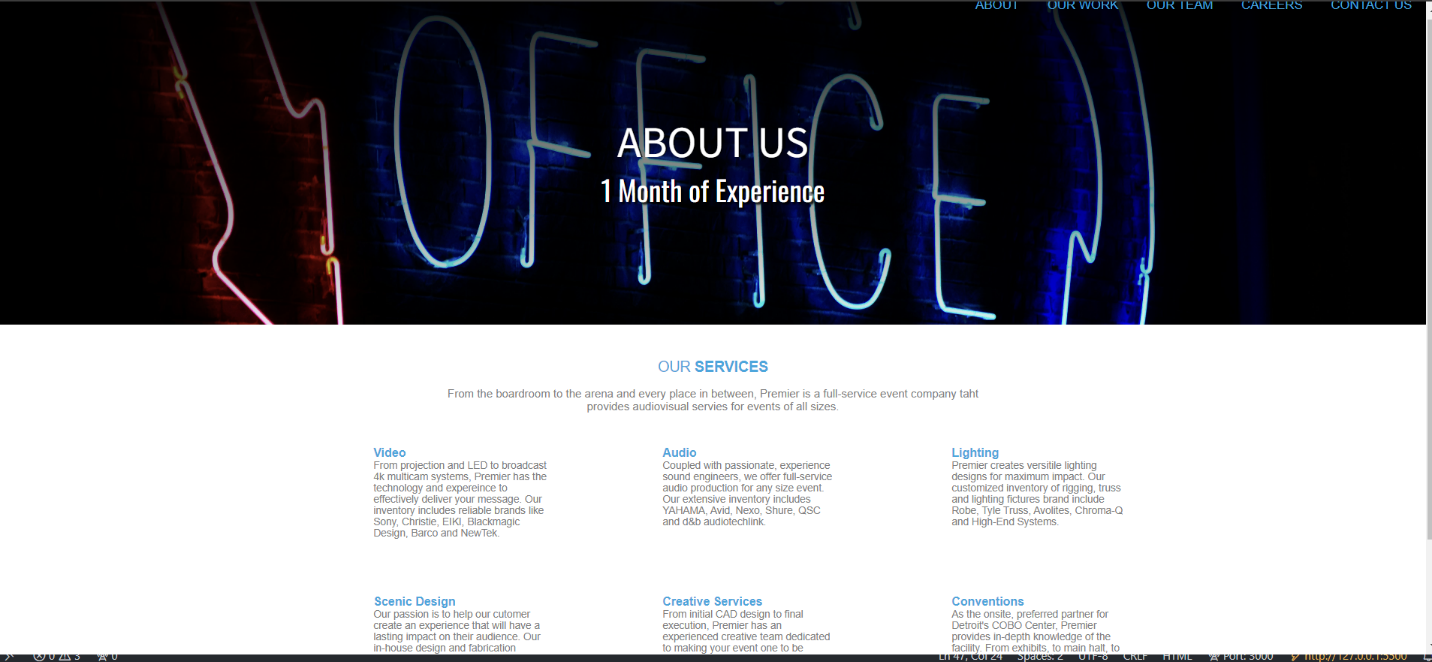
* An event listener is added to the movie selection dropdown (movieSelect). When a user selects a different movie, the ticket price is updated, and the selected movie data is saved.
* Another event listener is attached to the container that holds the seats. It detects clicks on seats that are not sold. When a seat is clicked, its selected state is toggled (either selected or deselected), and the count and total are updated accordingly.

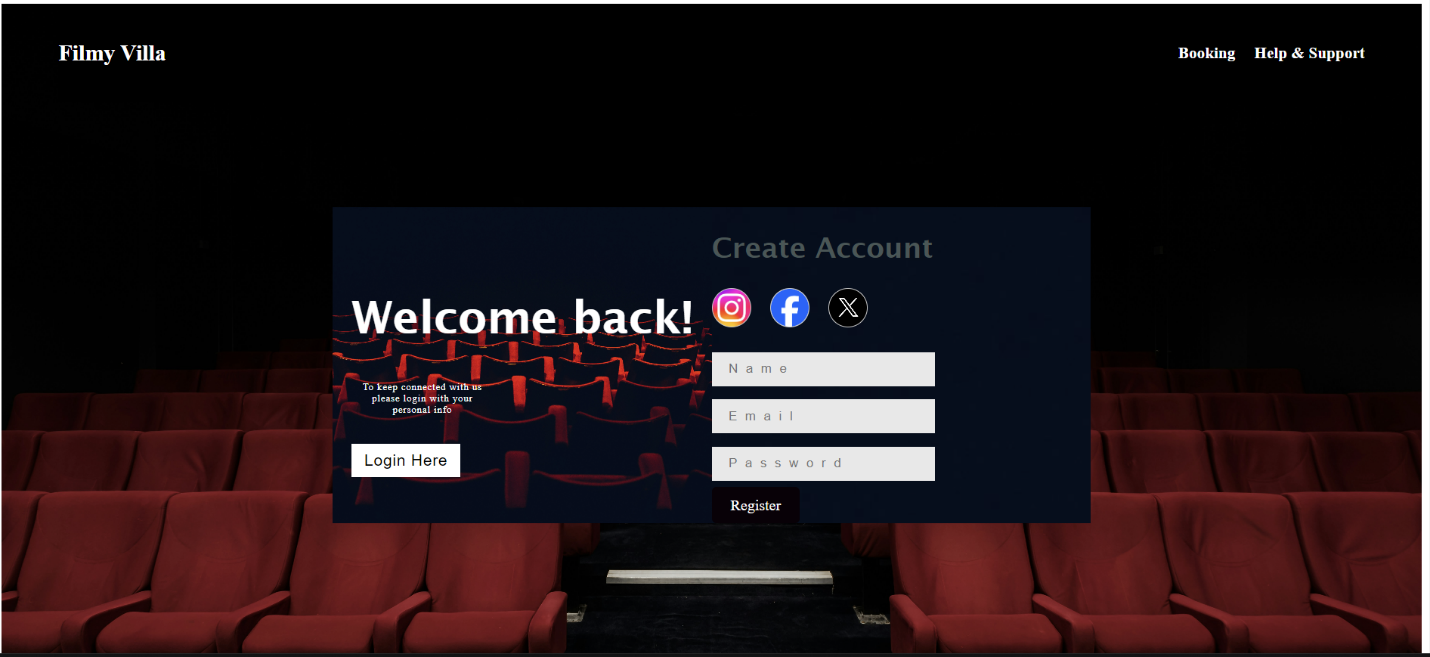
**Initial Count and Total Display**:

* The last line of the code calls the updateSelectedCount() function to display the initial count of selected seats and the total price when the page loads.

10. **Output Screenshots**







11. Git Hub Repo Link:

12. Conclusion

In today's Web development, a good page design is essential. A bad design will lead to the loss of visitors and that can lead to a loss of business. In general, a good page layout has to satisfy the basic elements of a good page design. This includes color contrast, text organization, font selection, style of a page, page size, graphics used, and consistency. In order to create a well-designed page for a specific audience. The developer needs to organized and analyze the users' statistics and the background of the users. Although it can be hard to come up with a design that is well suited to all of the users, there will be a design that is appropriate for most of the audience. The better the page design, the more hits a page will get. That implies an increase in accessibility and a possible increase in business.

13. References

1) https://www.w3schools.com/tags/tag\_form.asp

2) https://www.geeksforgeeks.org/css-tutorial/