**Name of the Project** – Multiplex Movie Ticket Booking

**Objectives/Vision**

Develop a web application that can see a list of movies and their availability on various multiplexes.

The application owner or admin can add multiplexes and corresponding movies. Movies can be selected based on the multiplexes and time slots.

Tickets are available with various options like gold, silver, and premium class.

**Functional Requirements**

1. Complete registration process: We need to develop user registration, login, and forgotten passwords. Reset password and forgot password should implement with the registered email id of the user.
2. Once the user provides the correct login credential, the user will lend into the user dashboard. The dashboard should have all necessary features available to the specific user.
3. User type varies from the user, admin, and application owner.
4. Every user will have their own dashboard and respective features that can be used for specific operations.
5. User Dashboard: Can see all movies and their availability in the corresponding multiplex.

Can see movies in the specific multiplex.

Can purchase a movie ticket and apply offer and discount available if any.

1. Discount: Discounts and offers can also be seen under the dashboard.
2. Proper cart should be implemented and all discounts and offers can be applied to the cart feature.
3. Admin dashboard: Can register as one of the multiplex owners and add movies to their respective multiplexes.
4. Application owner: Can provide some useful information on various multiplexes and users. Once the user login, there is some message regarding Covid guidelines that need to be followed when the user comes to see the movie.
5. Application owner can see the feedback coming from the user and multiplex owners.

**Non-Function Requirement:**

1. Build and Test Responsive and Interactive Webpages
2. Proper authentication should be implemented using OAuth 2.0 / JWT.
3. Application should have proper validation implementation with all required forms.
4. Appropriate cloud service should be incorporated / can be used for deployment, developing features based on project or use case.
5. Proper front-end unit testing should be part of the implementation.
6. Junit test coverage should also be part of development.
7. Implement proper CI and CD pipeline.
8. Deploy application on AWS cloud. (like Deploy Java Microservices on Amazon ECS using AWS Fargate).

**Tools and Technologies to be used**

- VCS: Gitlab

- Front End: HTML, CSS, JavaScript, Bootstrap and REACT.

- Backend: Java 8, Spring Boot.

- Database: MySQL/MongoDB

- Testing: Mocha/Chai, Junit.

-Deployment: AWS Cloud services