PICTURE PERFECT
DESIGN DOCUMENT

KHEMRAJ GURJAR

CONTENTS

About

Back-end

- Language
- Authentication
- Microservices
- Database
- AWS services
- Go Libraries
- UML Diagrams

Front-end

- Libraries
- Mock UI Lay-outs

Testing and Automation

- Testing
- Testing libraries in golang
- Automation Testing tool
- Cloud Server

ABOUT

This is a website which has a collection of various movies and their information like rating,review and other details. It provides a great user experience and lets users know everything about the recent and old movies. It also lists the various cineplexes which show the movie nearby.

BACKEND

LANGUAGE : *We will use go-lang to design the back-end server of the website. Go is a programming language developed by Google for building modern software.It is an excellent choice in building fast and scalable apps. Some features of golang are:*

1. *Go compiles lightning-quick into fast native code.*
2. *Go has excellent support for concurrency with goroutines and channels.*
3. *Go has an interesting take on object-oriented programming with interfaces.*
4. *Exhaustive standard library containing everything you need to do server-side/network programming.*

AUTHENTICATION: *We will use Goth: Multiprovider authentication for go. Package goth provides a simple, clean, and idiomatic way to write authentication packages for Go web applications. We can get goth at "github.com/markbates/goth" .*
Gothic uses a CookieStore from the gorilla/sessions package to store session data.

MICROSERVICES:*Go Kit is a programming toolkit for building microservices in Go. It is a library which is designed to be imported to a binary package. In go kit we can find packages for Transport, logging , Metrics , Tracing , Service discovery , Client and Server.*

DATABASE : *MySQL*
MySQL is a very commonly used relational database management system. We have to maintain a database of movies and their contents and need a structured relational database which is highly scalable and quick so MySQL is perfect for our purpose.

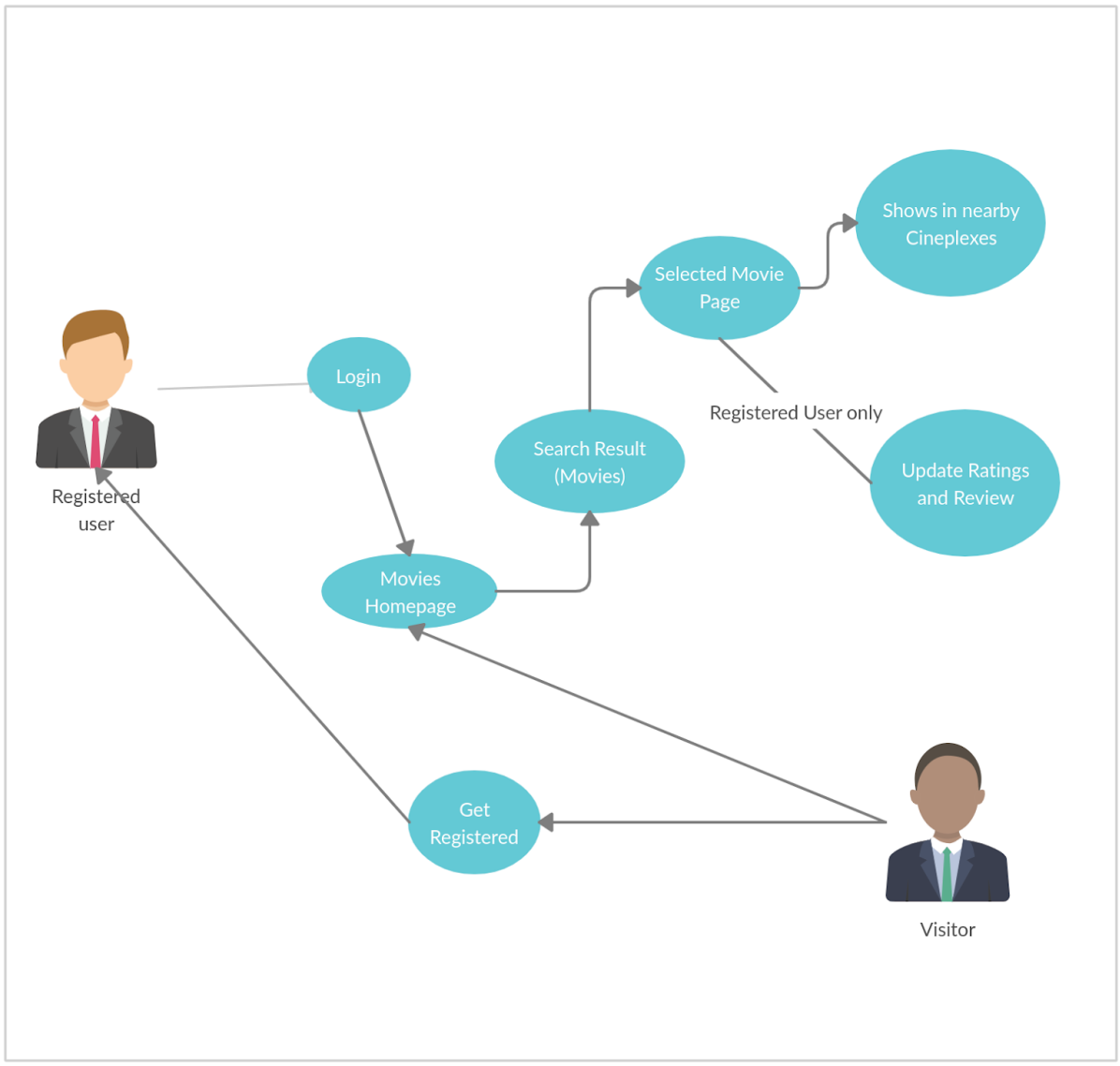
AWS SERVICES : *We will use AWS lambda, Amazon RDS (MySQL) , AWS CloudFront (Networking and content-delivery) and Amazon S3 (Storage).*

GO LIBRARIES:

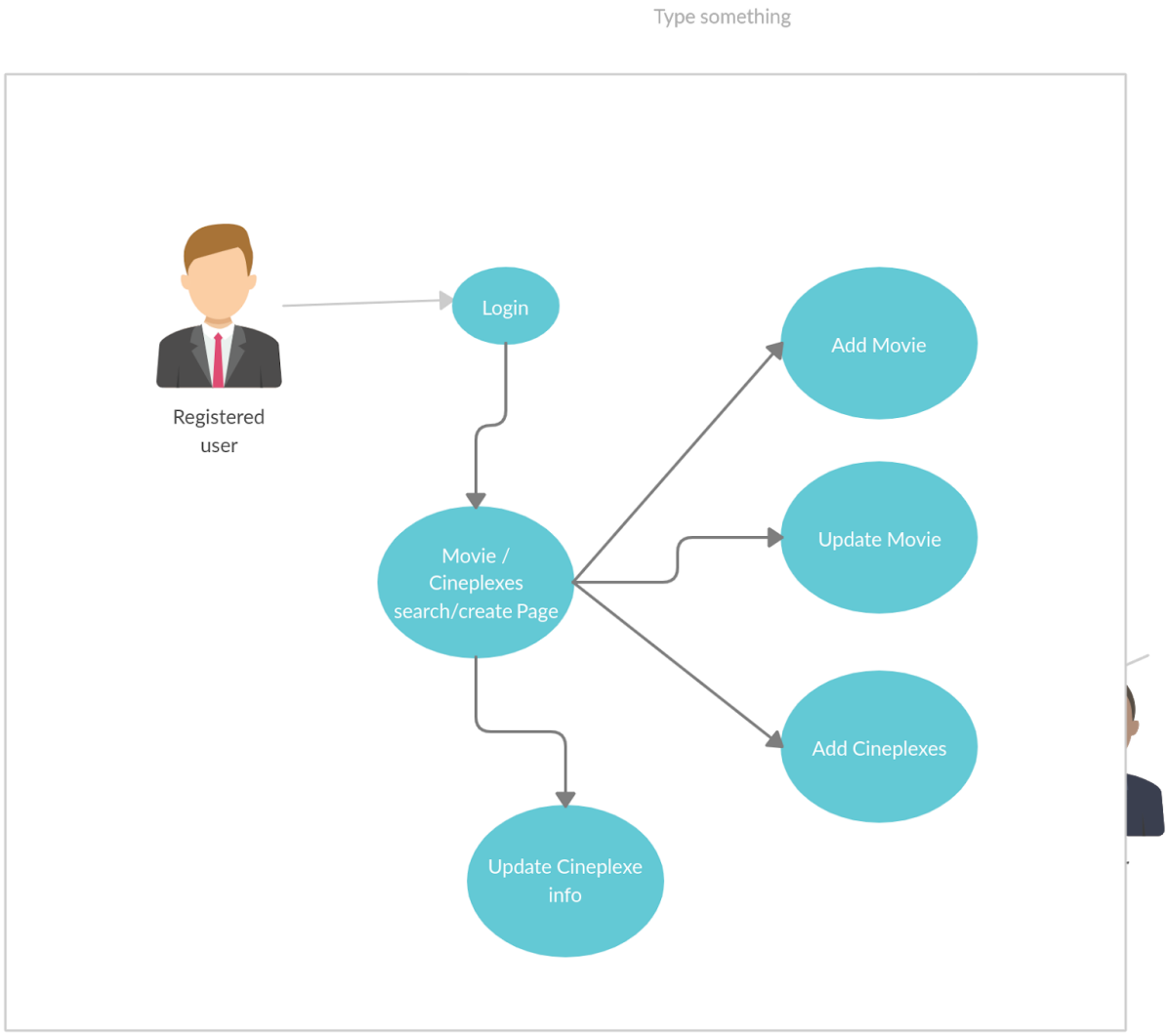
1.net/http, standard library HTTP package.
2.html/template, standard library HTML template engine.
3.database/sql, standard library interface with driver support for MySQL, Postgres, Oracle, MS SQL, BigQuery and most SQL databases.
4.markbates/goth, authentication for web apps.

SYSTEM DESIGN
(UML DIAGRAMS)

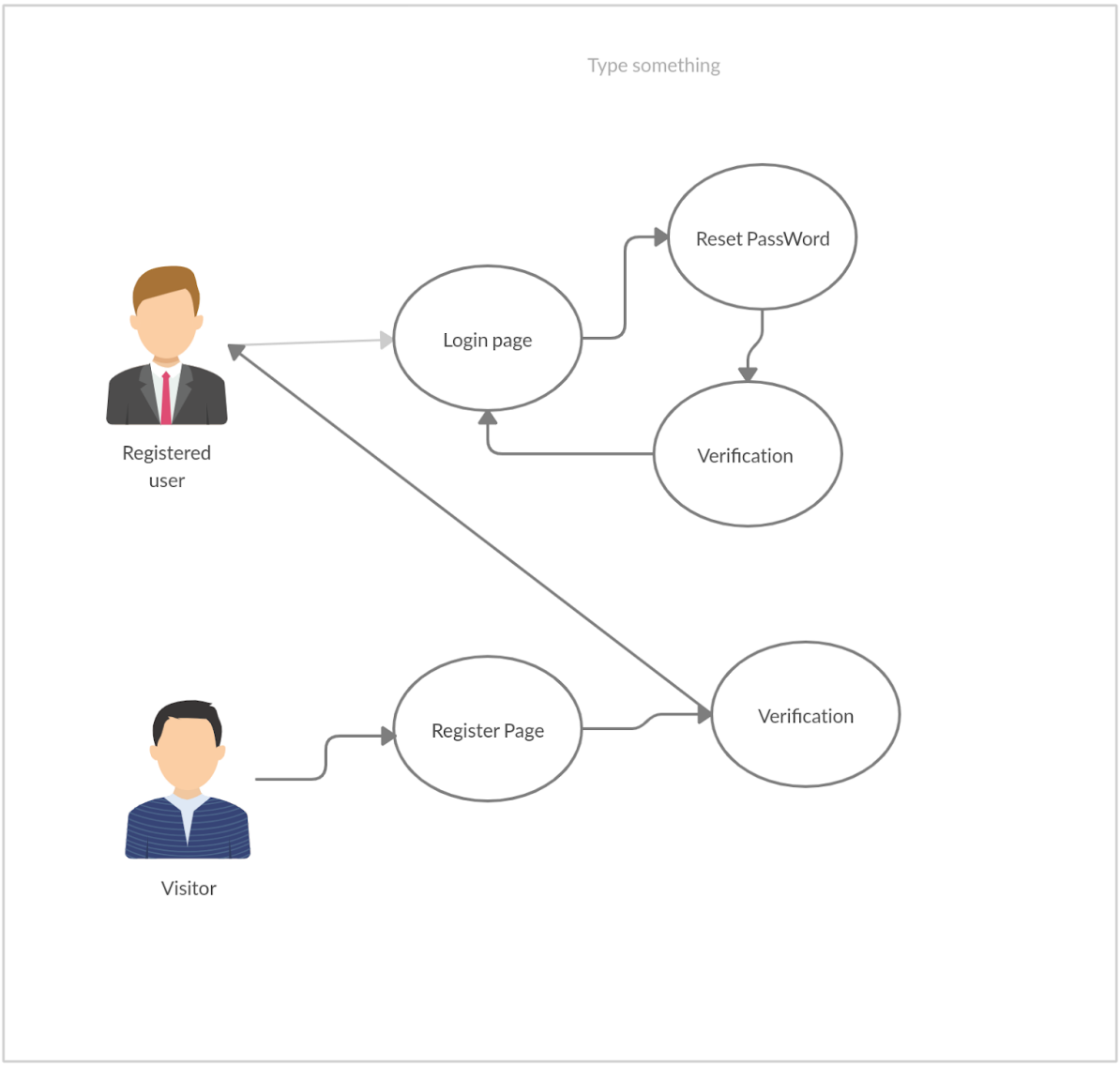
Use Case Diagram to illustrate user flow from one instance to another.



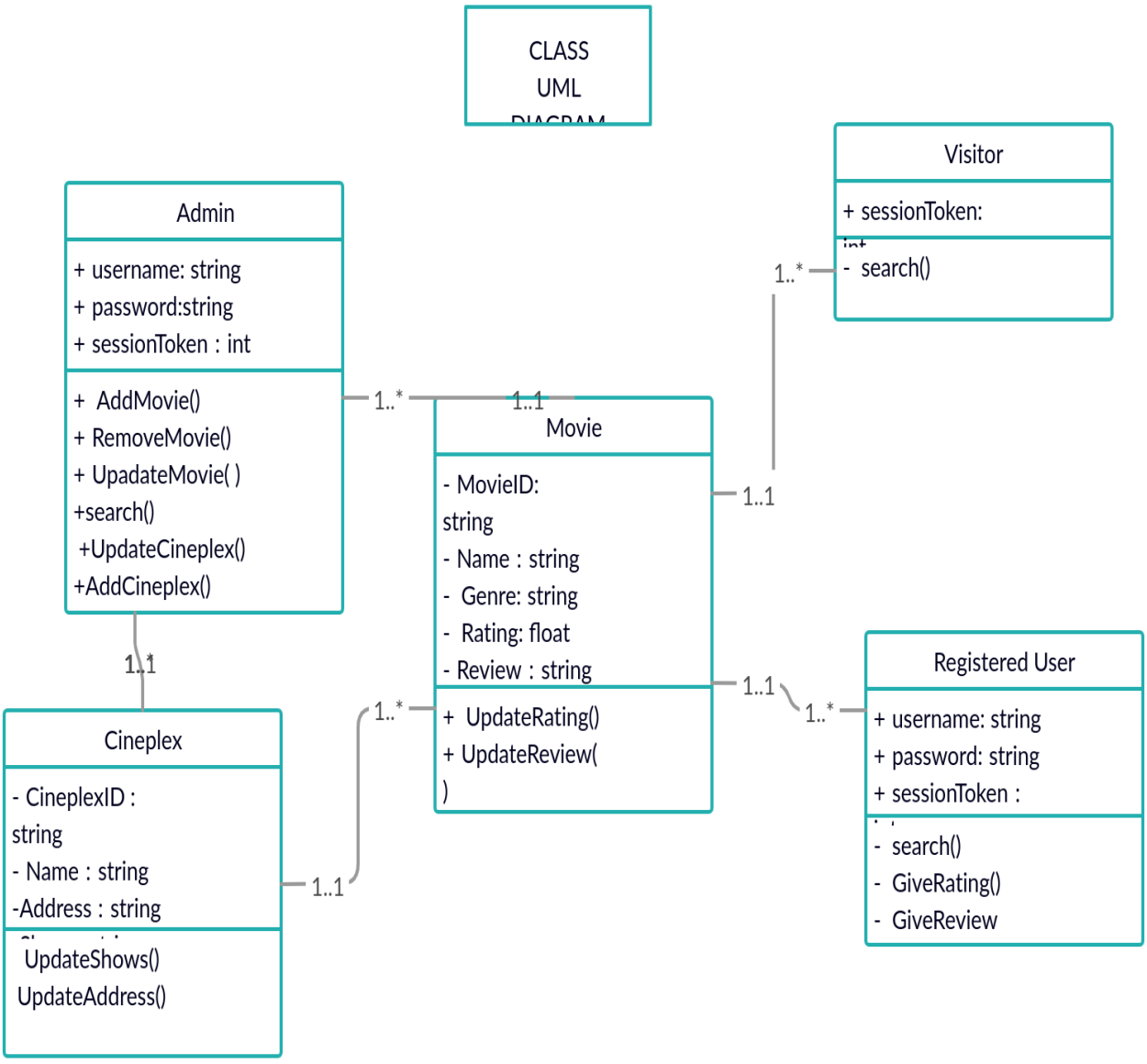
Use Case Diagram to illustrate Admin Operations.



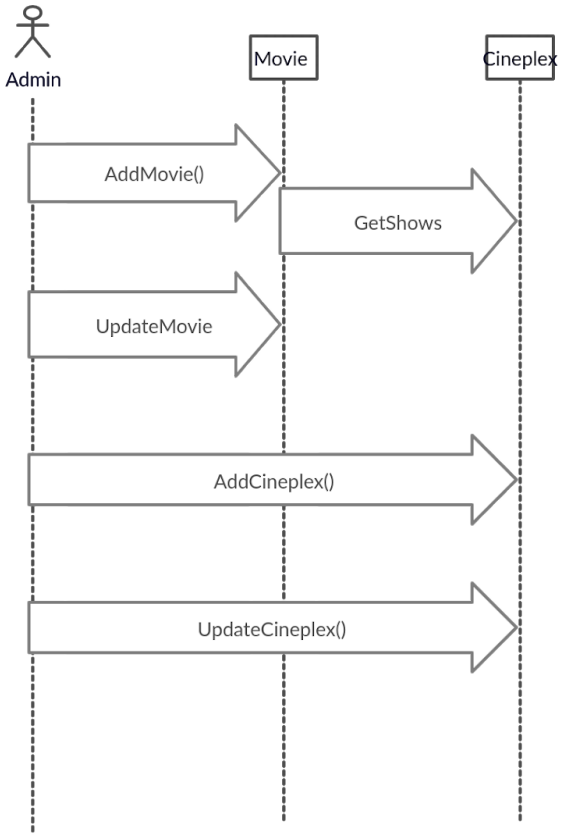
Use Case diagram to illustrate the Login/Authentication Process.

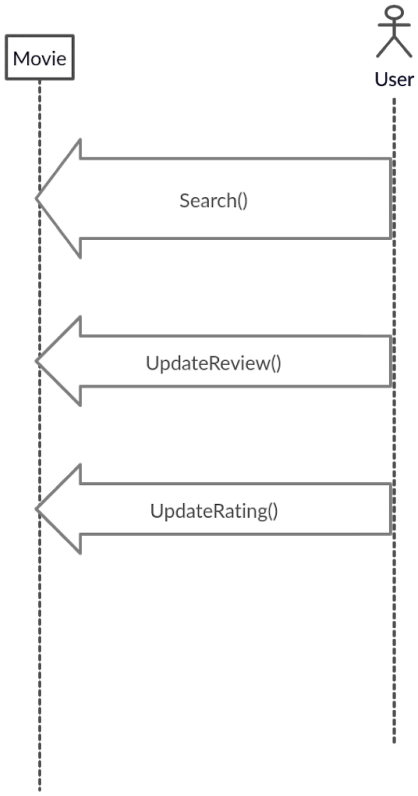


Class Diagram to illustrate various modules and relations.



Sequence Diagram to show various operations.





FRONT-END

LIBRARY USED: *React , Bootstrap .*

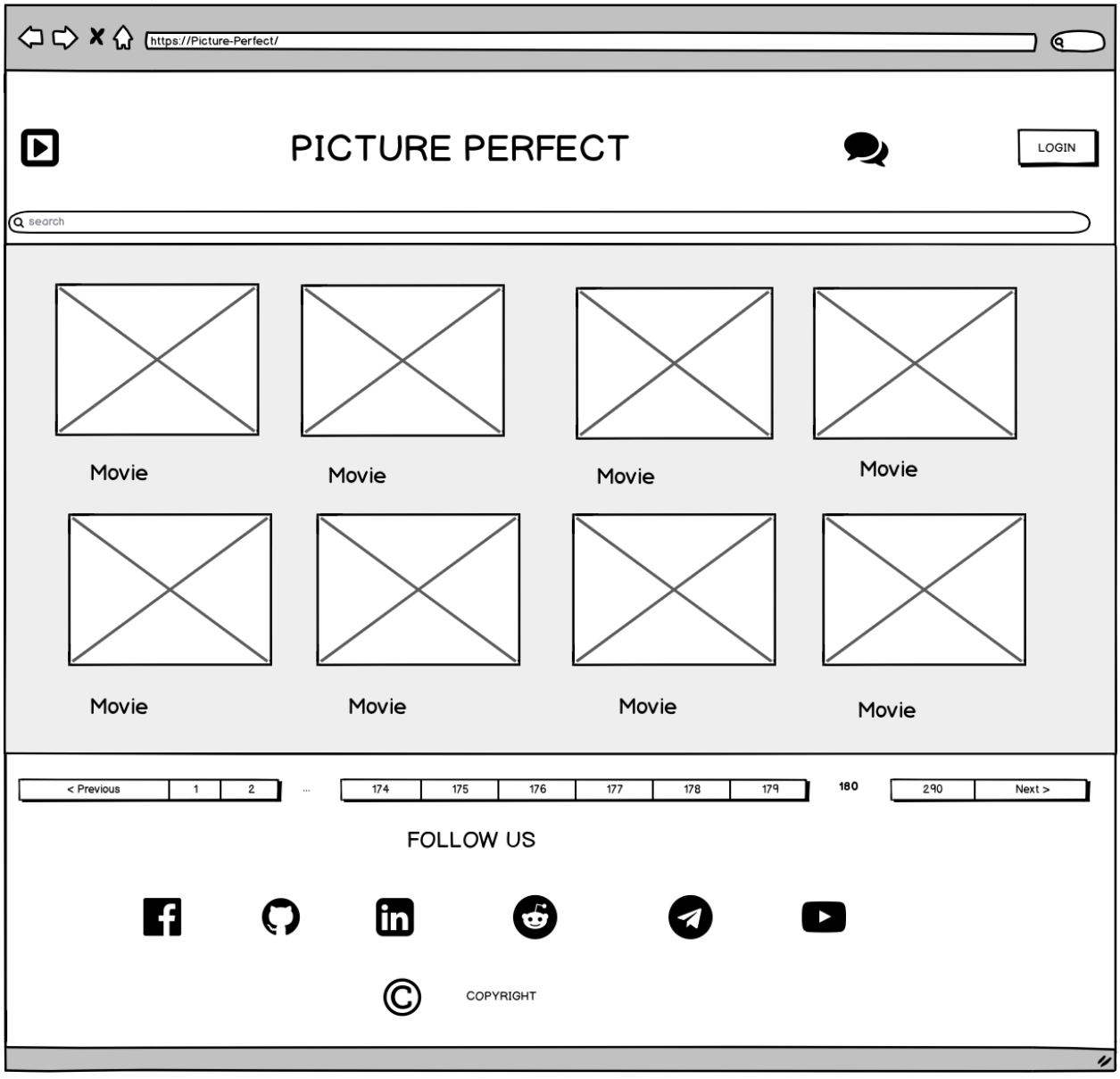
***React** allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple. It works only on user interfaces in the application.*

***Bootstrap** is a framework to help you design websites faster and easier. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, etc.*

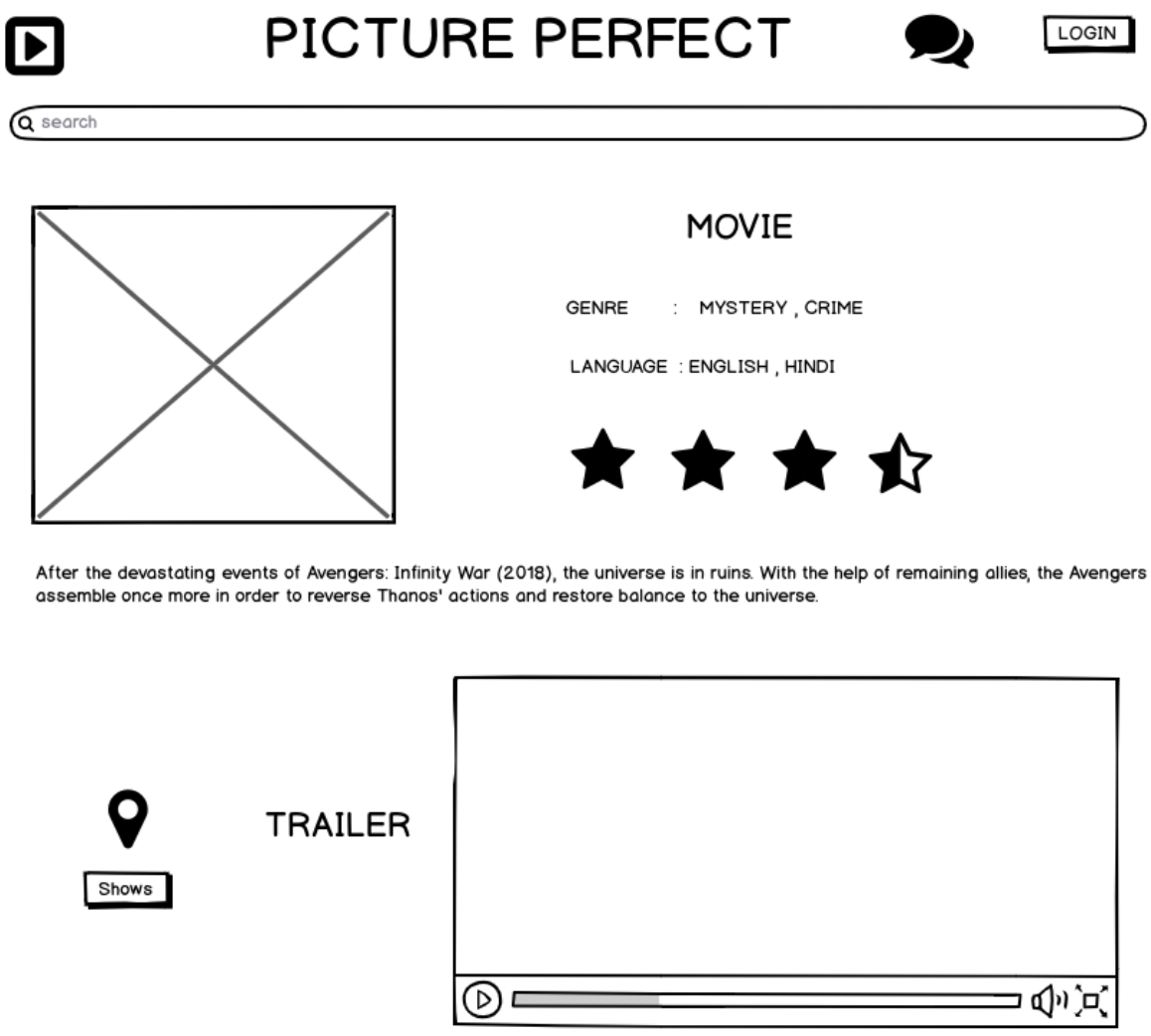
UX LAY-OUT MOCKS:

Here are various mock layouts of various web pages of our application that we are going to build :

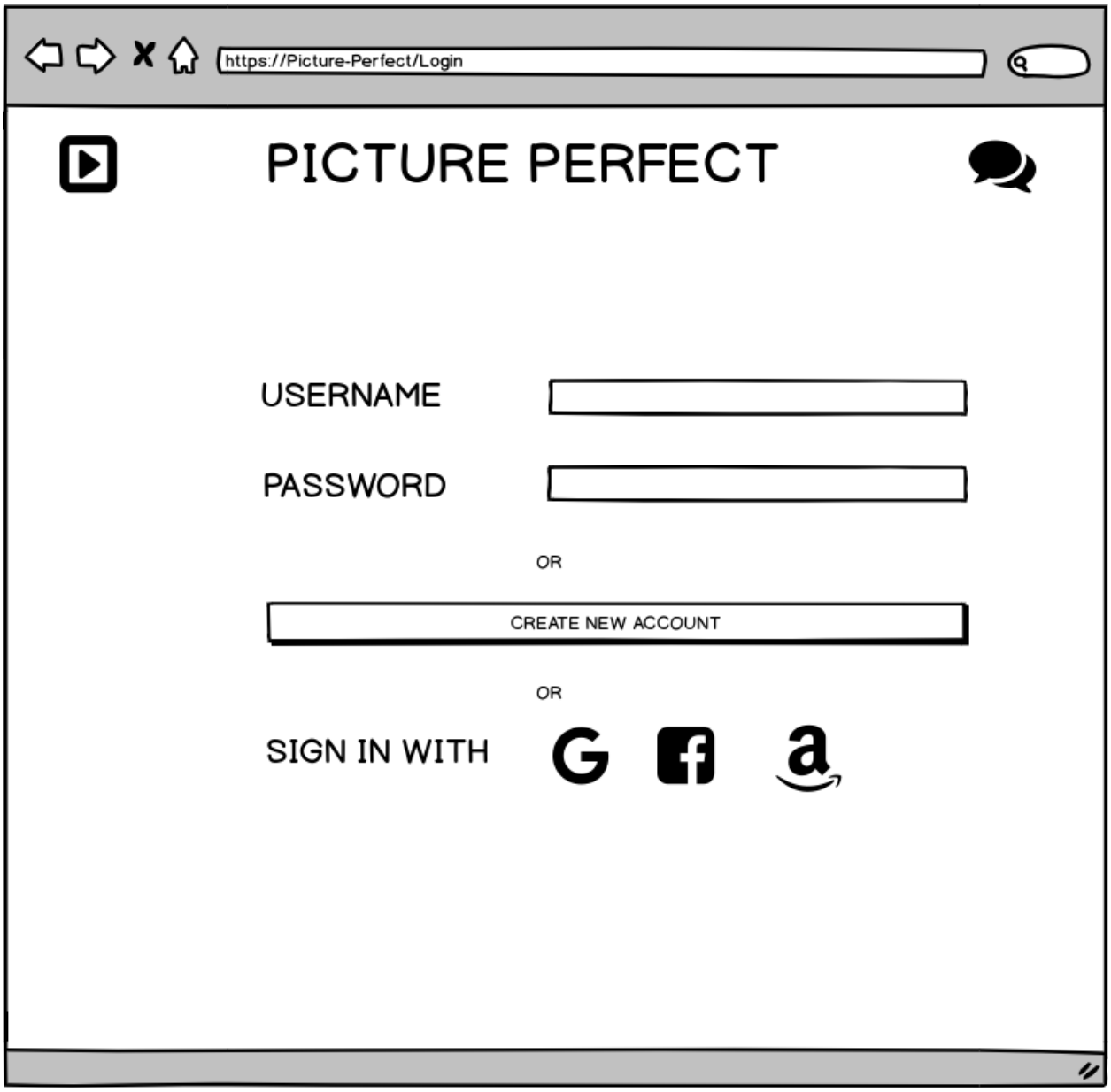
Home Page


https://Picture-Perfect/
PICTURE PERFECT
LOGIN
search
Movie
Movie
Movie
Movie
Movie
Movie
Movie
Movie
< Previous
1
2
...
174
175
176
177
178
179
180
290
Next >
FOLLOW US
COPYRIGHT

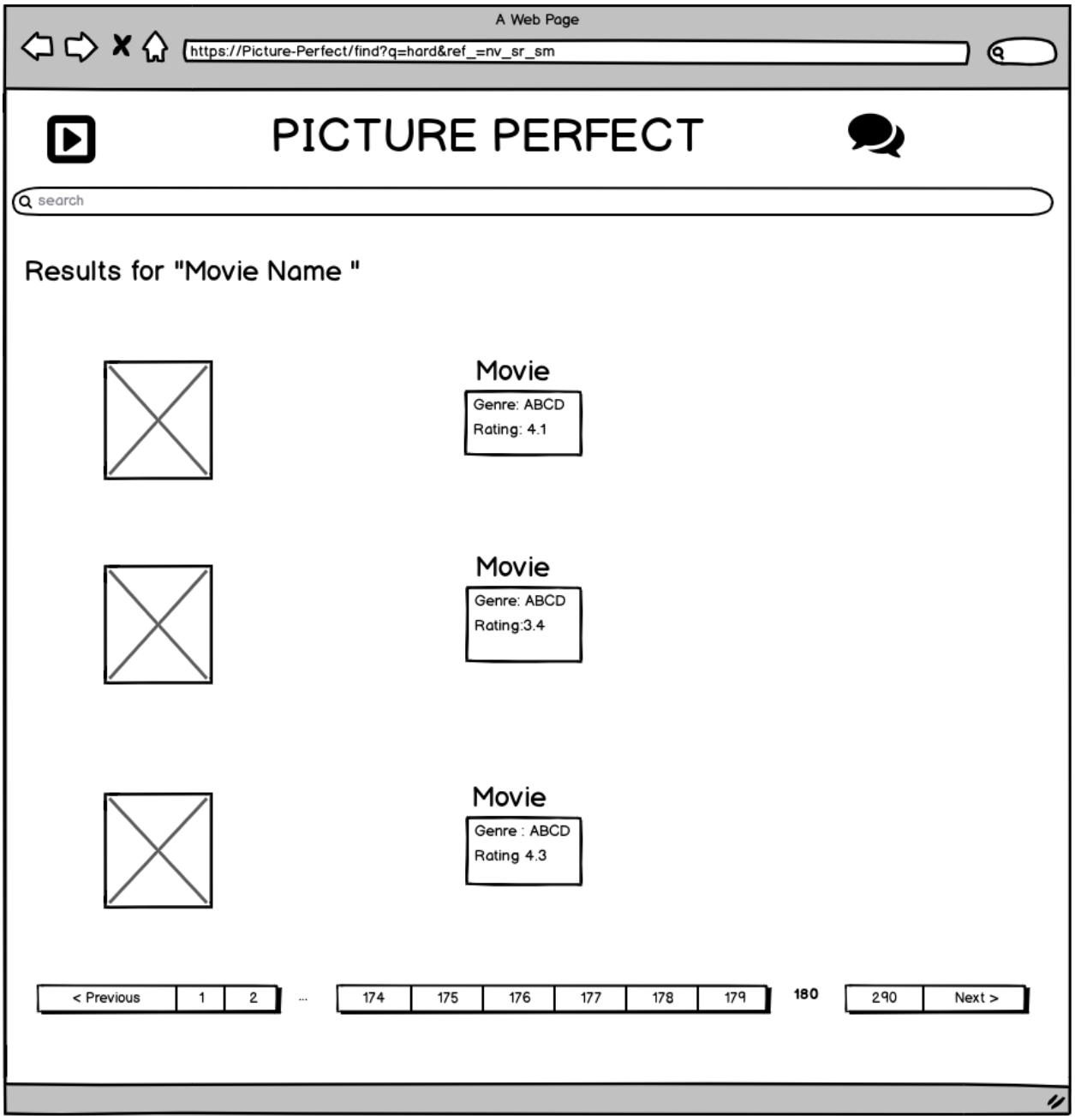
Movie Page : Have all details and media of a particular movie.



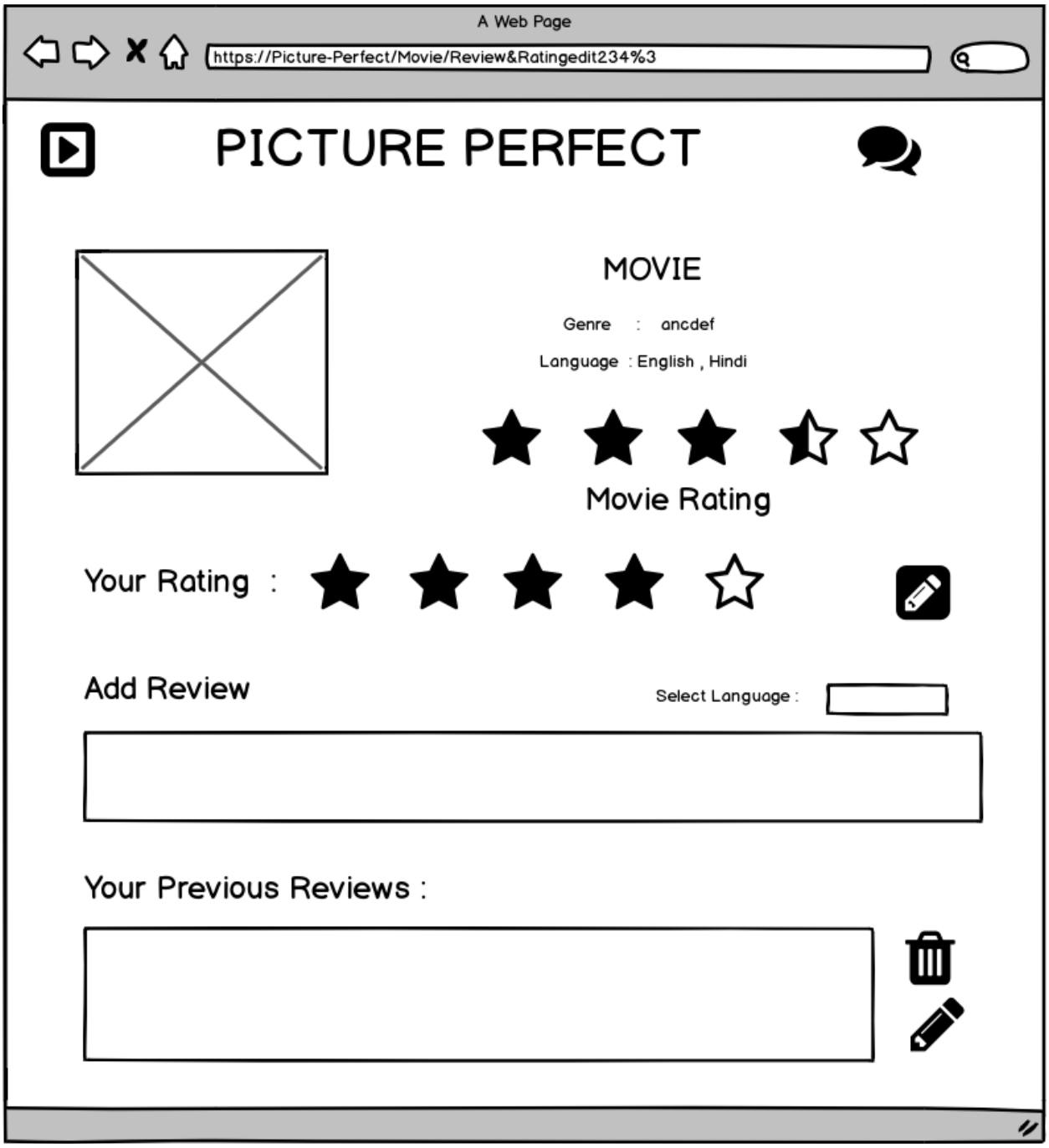
Login Page for User/Admin with various other login/sign in options.



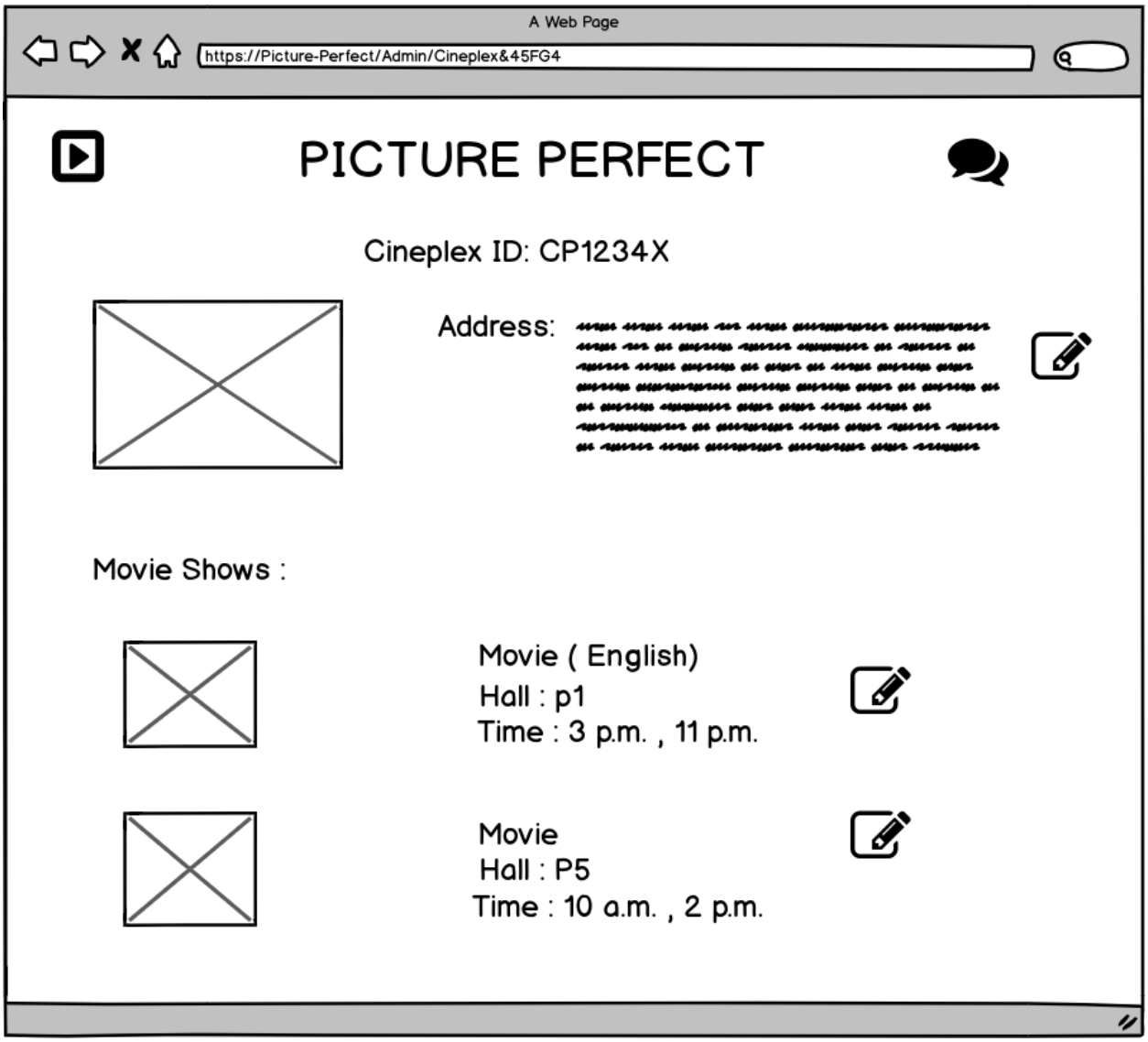
Search Result Page with related results.



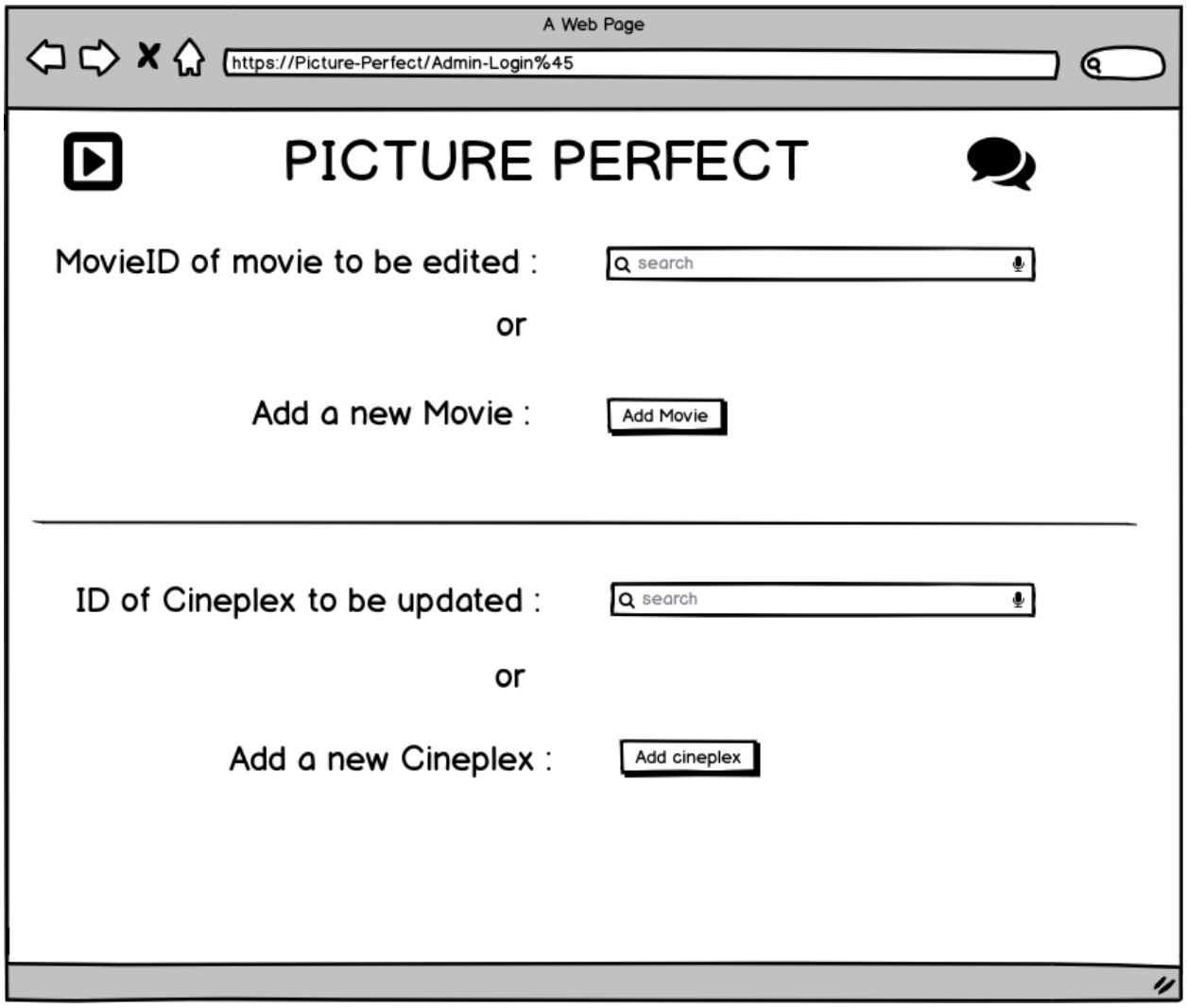
Review and Rating Page with some edit options for registered users.



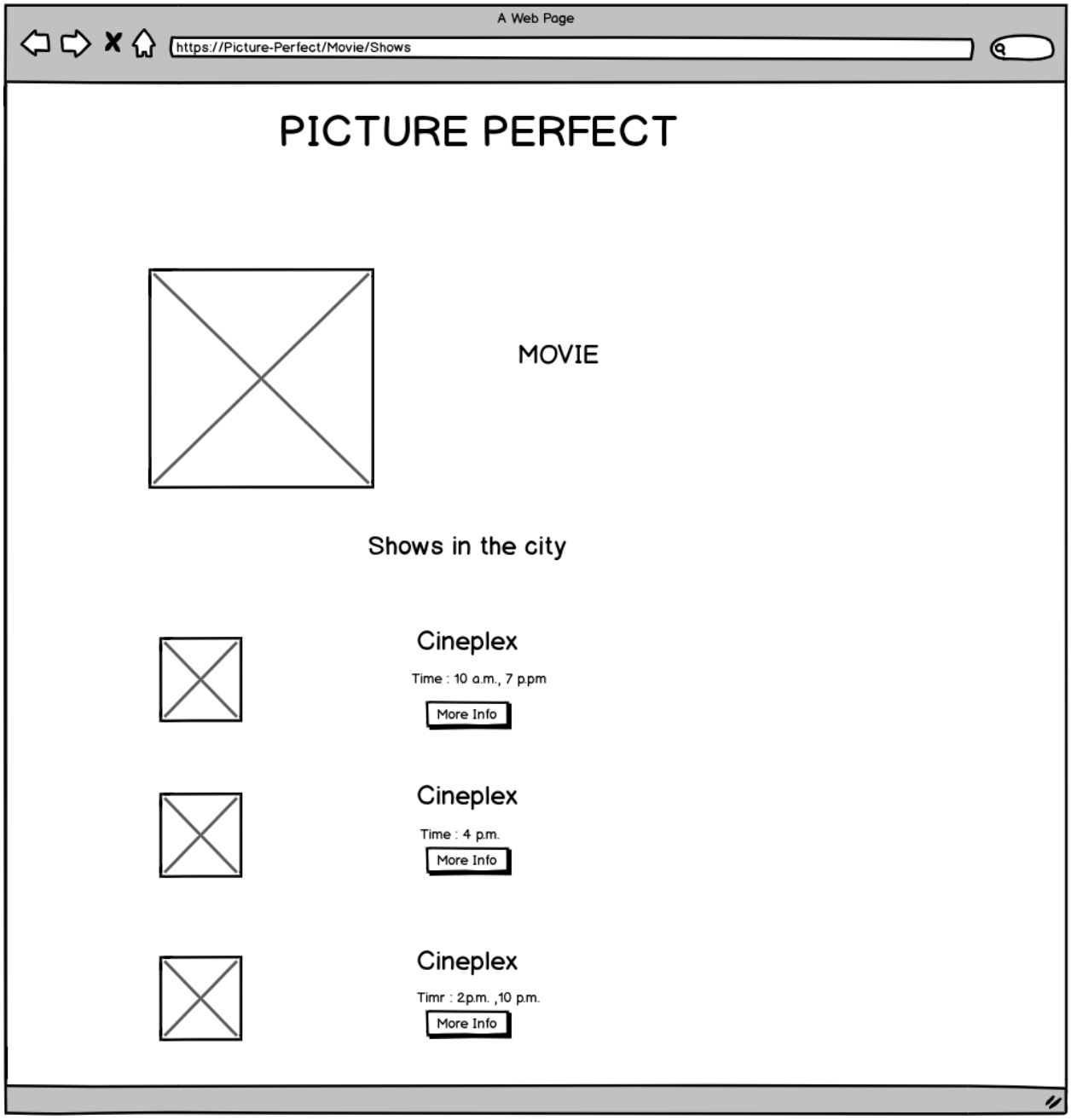
Cineplex Page (Admin only) with editable details.



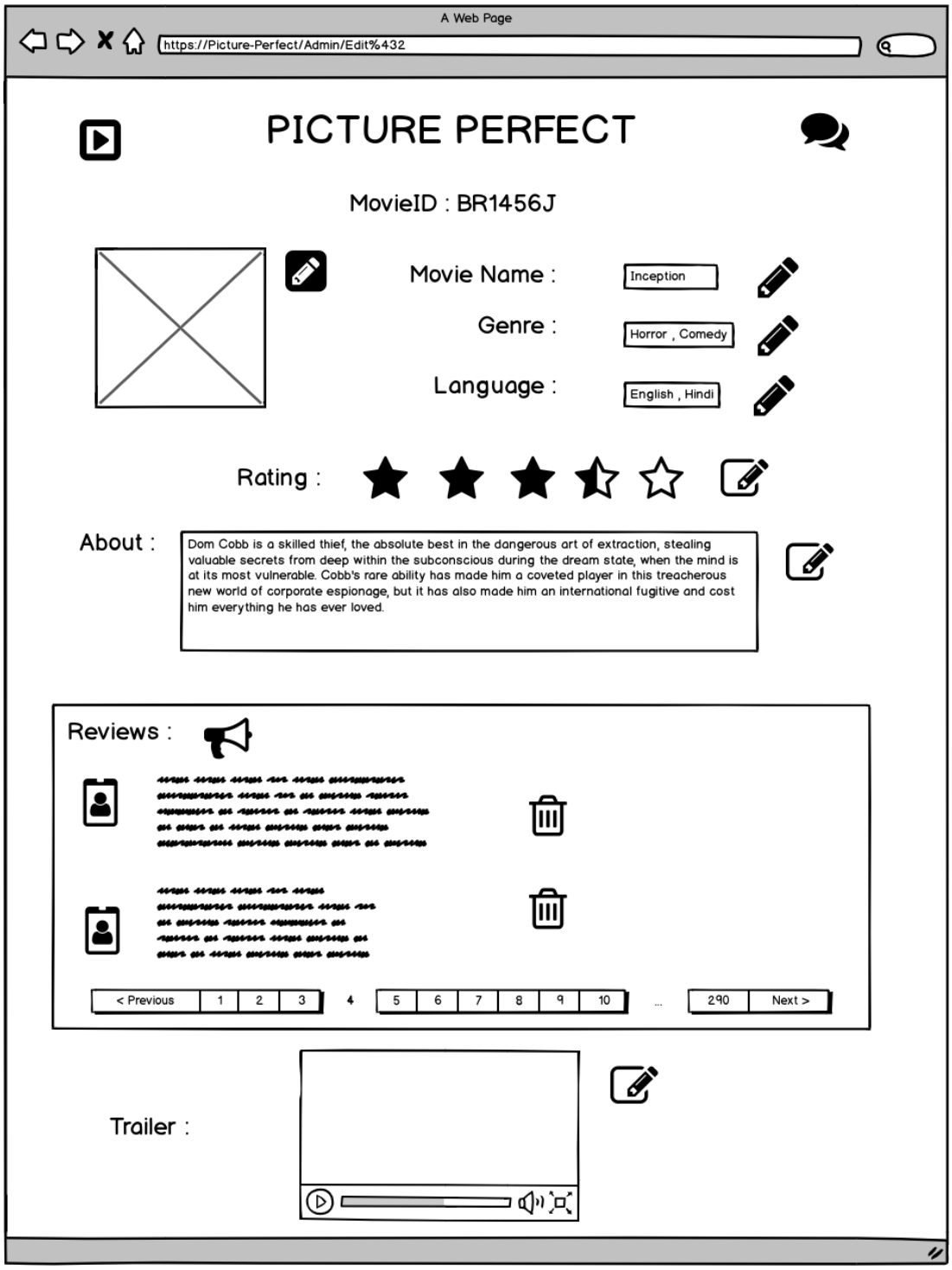
Admin Dashboard : For Editing/Adding Movie/Cineplex.



Shows Page for a particular movie in the city showing Cineplexs info.



Movie Page (Admin only) to update movie information



TESTING AND AUTOMATION

TESTING : *Some testing approaches we are going to use are :*

1.Functionality Testing : The first step of the testing phase is to make sure that the web application is functionally correct. Functionality testing checks the database connection, all links in the web pages, cookies, forms used for submitting and/or getting info from the user etc. This will be done alongside go server development. The built-in ***testing*** *package is provided by the Go's standard library.*

2.Interface Testing : This web page test checks whether or not all interaction between the app server and the web server runs smoothly. Not only the communication process needs to be tested, but the displaying of error messages as well. Also, this test is used to determine whether the interruptions by the server and/or by the user are handled properly.

3.Compatible Testing : We need to run the browser compatibility test to check if the app is displayed correctly in different browsers. Another test to run is the mobile browser compatibility which is supposed to confirm that the app displays correctly across mobile browsers.

4.Performance Testing : Performance testing includes testing under different internet speeds as well as under normal and peak loads.

5.Security Testing : This type of testing is done to check if our app is secure from any malicious programs and its data and integration are protected. We will test some programs on our application to verify its security.

LIBRARIES FOR TESTING IN GOLANG

*1.**testing** : This package contains all testing functions needed in go.*
*2.**`net/http/httptest`** :`It` is a library for testing web applications.*

AUTOMATION TESTING TOOL

***Selenium:** It provides extensions to emulate user interaction with browsers, a distribution server for scaling browser allocation, and the infrastructure for implementations of the W3C WebDriver specification that lets you write interchangeable code for all major web browsers.*

CLOUD SERVER: *AWS EC2 using AWS CodeDeploy.*