

# **ESTRELLA**

## **A PROJECT REPORT**

**Submitted by**

<b>320126551042</b>	<b>V. Nehaar</b>
<b>320126551040</b>	<b>M.Viswash</b>
<b>320126551001</b>	<b>B. Likitha</b>
<b>320126551048</b>	<b>P. Sadhika</b>
<b>320126551002</b>	<b>B. Akash</b>

**in fulfillment of the Mini Project of**

**DATABASE MANAGEMENT SYSTEMS**

**in**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
(AI & ML, DS)**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
(AI & ML, DS)**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

**(Affiliated to Andhra University)**

**SANGIVALASA, VISAKHAPATNAM - 531162**

**2020-2024**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
(AI & ML, DS)**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND  
SCIENCES**

**(Affiliated to Andhra University)**

**SANGIVALASA, VISAKHAPATNAM - 531162**

**2020-2024**



## **BONAFIDE CERTIFICATE**

This is to certify that this project report “ESTRELL-THE CINEMATIC PLATFORM” is the bonafide work of **M.Viswash (320126551040), V.Nehaar(320126551042),B.Likitha(320126551002),P.Sadihka(320126551048), B.Akash(320126551002)** of III/IV CSD carried out the project work undermy supervision.

**Mrs.G V Gayathri**  
**(ASSISTANT PROFESSOR)**  
COMPUTER SCIENCE  
ENGINEERING (AI&ML,DS)  
ANITS

**Dr.M.Ramakrishna Murty**  
**(HEAD OF THE DEPARTMENT)**  
COMPUTER SCIENCE  
ENGINEERING (AI&ML,DS)  
ANITS

## ACKNOWLEDGMENT

An endeavor over a long period can be successful with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them.

We owe our tributes to **Dr.M.Ramakrishna Murty**, Head of the Department, Computer Science Engineering (AI&ML,DS), ANITS, for his valuable support and guidance during the period of project implementation.

We wish to express our sincere thanks and also gratitude to our project guide **Mrs. G V Gayathri**, Assistant Professor, Department of Computer Science Engineering (AI&ML,DS), ANITS, for simulating discussions, in analyzing problems associated with our project work, and for guiding us throughout the project. Project meetings were highly informative.

We express our warm and sincere thanks for the encouragement, untiring guidance, and confidence they had shown in us. We are immensely indebted for their valuable guidance throughout our project.

We also thank all the staff members of, the Department of Computer Science Engineering (AI&ML,DS) for their valuable advice. We also thank supporting staff for providing resources as and when required.

**320126551042**

**V.Nehaar**

**320126551040**

**M.Viswash**

**320126551001**

**B.Likitha**

**320126551047**

**P.Sadhika**

**320126551002**

**B.Akash**

# TABLE OF CONTENTS:

	<b>Page no.</b>
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 Overall Description	
<b>2. SYSTEM ANALYSIS</b>	<b>2-3</b>
2.1 Software requirement specifications	2
2.1.1 Purpose	2
2.1.2 Scope	2
2.1.3 Objective	2
2.1.4 Existing System	2
2.1.5 Proposed System	2
2.1.6 Functional Requirements	3
2.1.7 Non-Functional Requirements	3
2.1.8 Software Requirements	3
2.1.9 Hardware Requirements	3
<b>3. DATABASE DESIGN</b>	<b>4-8</b>
3.1 Database	4
3.2 ER Diagram	4-6
3.2.1 Elements in ER Diagram	4-5
3.2.2 How to draw ER Diagram	5
3.2.3 ER Diagram for Project	6
3.3 Database Tables	7-8
<b>4. DATABASE NORMALIZATION</b>	<b>9-14</b>
4.1 Normalization	12-14
<b>5. IMPLEMENTATION</b>	<b>15-19</b>
5.1 Overview of Software Used	15-17
5.2 Coding	17-19
<b>6. TESTING</b>	<b>20-24</b>
6.1 Types of Testing	20-22

6.1.1 Unit Testing	20
6.1.2 Integration Testing	21
6.1.3 System Testing	21
6.1.4 Acceptance Test	22
6.2 Test Cases	22-24
6.2.1 Login	22
6.2.2 Movie Uploading	23
<b>7. RESULTS</b>	<b>25-34</b>
7.1 Input/Output Design	
<b>8. CONCLUSION AND FUTURE SCOPE</b>	<b>35</b>
8.1 Conclusion	35
8.2 Future Scope	35
<b>9. REFERENCES</b>	<b>36</b>

# 1. INTRODUCTION

This Estrella is system designed to overcome the problems faced by regular cinema streaming sites. Here we provide platform for the film makers, enthusiastic of sharing their content to world. Regular Cinema streaming sites does not allow to stream content by individual film makers. In order overcome this we have introduced Estrella.

## 1.1 Overall Description:

As mentioned earlier, this Virtual Estrella is a solution for problems faced due to Generic Cinema Streaming platforms. Problems. This system's access is granted to everyone and provides the users with a very friendly user interface.

This platform provides you with the excellent experience in watching cinema at users' will anywhere, anytime.

Estrella allows a user to experience film exploration via the Internet. Online streaming is also known as OTT(Over The Top). Estrella offers viewers almost every service traditionally available through a streaming platform including watch history, exclusive content, cinema, additionally we provide the ability to upload content and stream short-films with exceptional priority.

This system is a replacement for the existing systems like Netflix, Prime Video, Avanti cinema, so as to be able to watch instantly. Everything is available in the system and having a good Network is just enough for watching films.

The user just has to log in with their information like id and password and thus they can access their account and do whatever available in the system with their account. The system initially shows the home page showing recommendations. So if the user wants to make an upload then he can directly move to upload page and fill the details of the film and upload a Media file of type Moving Image, similar happens with the viewing.

## **2. SYSTEM ANALYSIS**

System Analysis is the description of a system into its component pieces to study how to troubleshoot for any errors and develop any new features to be added to the design.

### **2.1 Software Requirement Specifications:**

Software Requirement Specification is the starting point of software development activity.

As the system grew more complex it became evident that the goal of the entire system can be easily comprehended. Hence the needs for the requirements phase are used. The software project is initiated by the client's needs, the SRS is the means of translating the ideas of the minds of clients (the input) into a formal document. The purpose of the software requirement specifications is to reduce the communication gap between clients and developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

#### **2.1.1 Purpose:**

Estrella's purpose is to entertain people of their choice and support active individual filmmakers along.

#### **2.1.2 Scope:**

The scope of the system to provide any time watching experience with unmatched availability.

#### **2.1.3 Objective:**

The main objective of this system is to enhance the viewing or watching experience and effort to access a theatrical visit by creating a smart solution on the internet.

#### **2.1.4 Existing System:**

The existing streaming system is completely paid i.e., the user has to subscribe to the providers.

#### **2.1.5 Proposed System:**

The proposed system is a replacement for the existing streaming systems that are pay based. Access to Estrella is both for users and content creators, but they don't have to pay for any of the actions they perform, content they watch it is completely free of cost.

### **2.1.6 Functional Requirements:**

- Content creators and also users have the ability to log in.
- Both Content creator and users can create their own accounts.
- Creator can upload films.
- User has the ability to watch the films.
- Content has the ability to watch the films.
- Content creator can remove or add a new content for users.

### **2.1.7 Non-Functional Requirements:**

- User Interface should be compatible to load HTML, PHP, and CSS pages in the Front-End, and Back-End.
- Software Interface should support the OS and Database of the user.
- Communication Interface can support all Web Browsers except Internet Explorer.
- **Availability:** The application is available to all the intended users, all the time based on the Network Availability.
- **Maintainability:** Issues that have been solved can be deleted from the database to maintain less complexity.
- **Implementation:** This System can be easily implemented and has scope for making future changes easily since the system is developed by using the feature of Modularity.
- **Security:** Security is provided by JSON web tokens, and data is encrypted by SHA256.

### **2.1.8 Software Requirements:**

- HTML
- PHP
- CSS
- MYSQL
- Any browser except internet explorer.

### **2.1.9 Hardware Requirements:**

- Desktop Computers and Personal Mobile Devices
- Keyboard.
- Mouse.
- Minimum 4GB RAM.
- Pentium Processor and above.



### 3. SYSTEM DESIGN

Object Oriented Design is concerned with developing an object oriented model of a software system to implement the identified requirements. It is the process of defining the components, interfaces, objects, classes, Attributes and operations that will satisfy the requirements.

The designer's goal is how the outputs to be produced and in what format samples of output are also presented. The processing phases are handled through the program construction and testing.

The importance of software design can be stated in a single word "QUALITY". Design provides us with representations of software that can be accessed for quality. Design is the only way that can be able to accurately translate a customer's requirements into finished software product or system without design risk.

Object oriented design can yield the following benefits:

- **MAINTAINABILITY:** Through simplified mapping to the problem domain, which provides for less analysis effort, less complexity in system design, easier verification by the user.
- **REUSABILITY:** Of the design artifacts, which saves time and cost
- **PRODUCTIVITY:** Gains through direct mapping of features of Object Oriented Programming Languages.

#### 3.2 UML DESIGN:

##### 3.2.1 DATA FLOW DIAGRAM:

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system. It maps out the flow of the information for any process or system, how data is processed in terms of inputs and outputs. It uses defined symbols like rectangles, circles and arrows to show data inputs, outputs, storage points and the routes between each destination. They can be used to analyze an existing system or model a new one. A DFD can often visually "say" things that would be hard to explain in words and they work for both technical and non-technical.

There are four components in DFD: 1. External Entity

2. Process

3. Data Flow

4. Data Store

##### 1. External Entity:

It is an outside system that sends or receives data, communicating with the system. They are the sources destinations of the information entering and leaving the system. They might be an outside organization or person, a computer system or a business system. They are known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram. These are sources and destinations of the system's input and output.

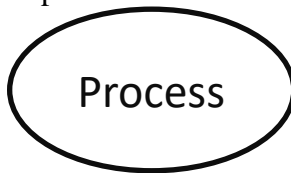
Representation:

Entity

## 2. Process:

It is just like a function that changes the data, producing an output. It might perform computations or sort data based on logic or direct the dataflow based on business rules.

Representation:



## 3.Data Flow:

A dataflow represents a package of information flowing between two objects in the data-flow diagram. Data flows are used to model the flow of information into the system, out of the system and between the elements within the system.

Representation:



## 4. Data Store:

These are the files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label.

Representation:



## DFD Levels:

A data flow diagram can drive into progressively more detail by using levels. DFD levels are numbered as 0, 1 or 2 and occasionally go to even level 3 or beyond. The necessary level of the detail depends on the scope of the task.

- **DFD Level 0:**

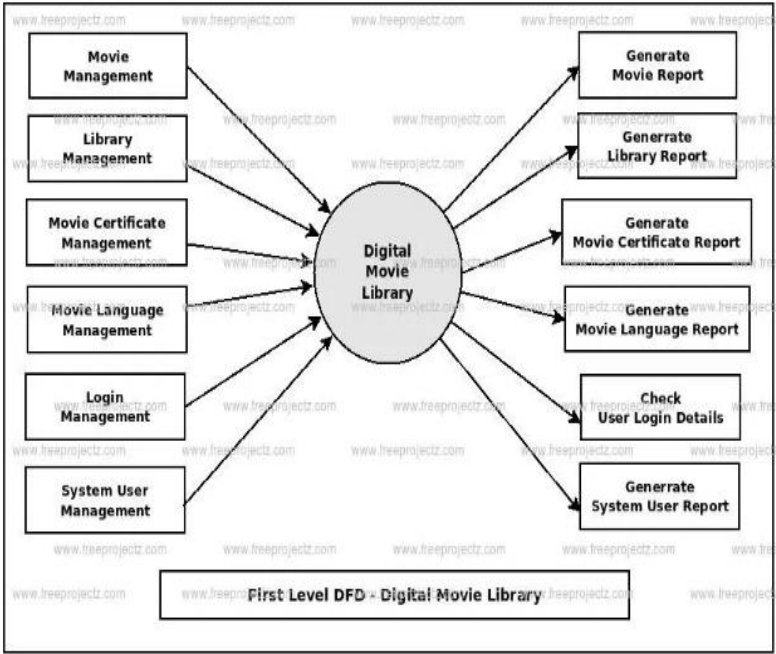
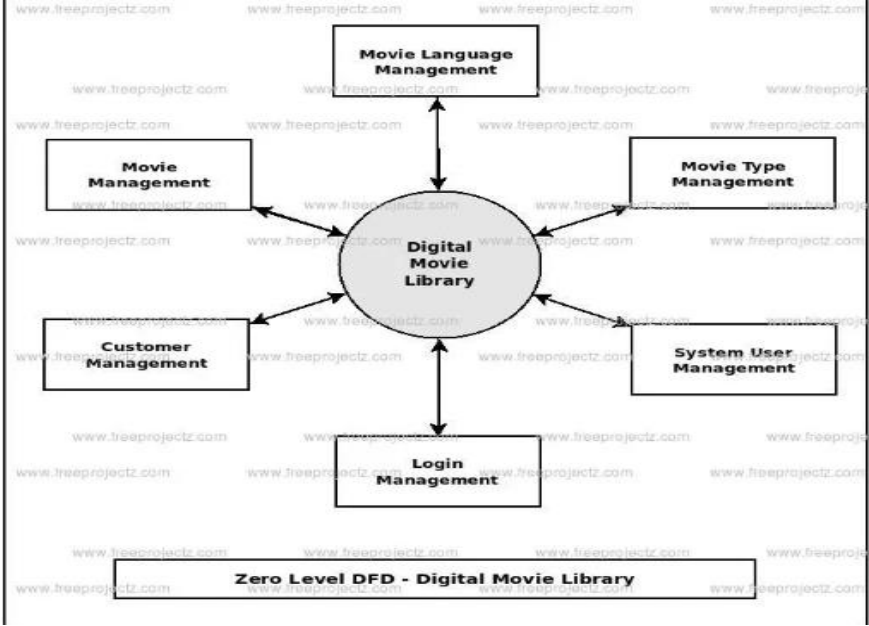
It is also called as context diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood.

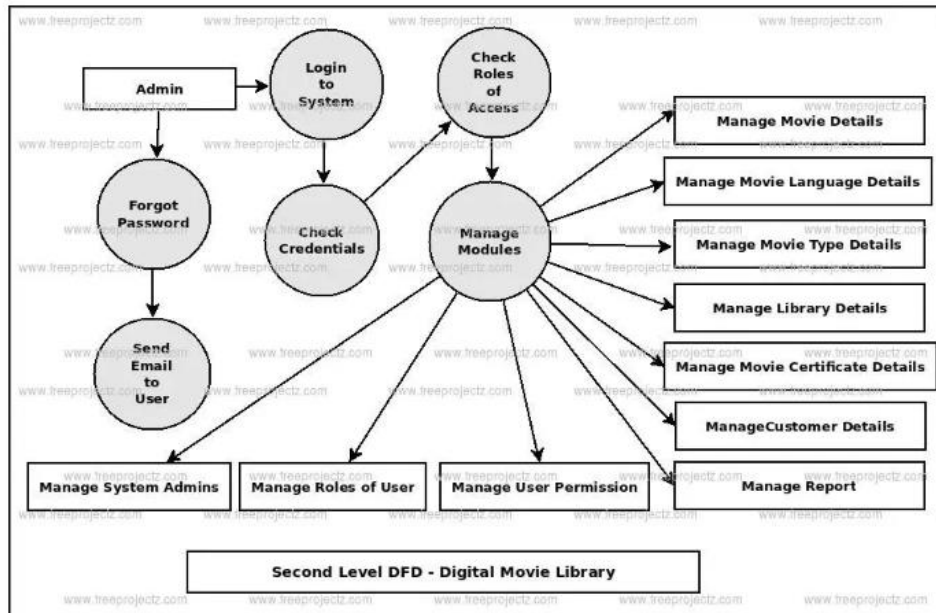
- **DFD Level 1:**

It provides a more detailed breakout of pieces of the Context Level Diagram. The main functions carried out by the system, break-down of the high-level process of the context diagram into its sub-process.

- **DFD Level 2:**

This goes one step deeper into parts of level 1. It may require more text to reach the necessary level of detail about the system's functioning.









### 3.2.2 Use Case Diagram:

Use Cases are used during Requirement Elicitation and Analysis Phase to represent the functionality of the system. The different roles that the people can fill, when they interact with a system are known as Actors. Use Case describes a function provided by the system that yields a visible result for an Actor.

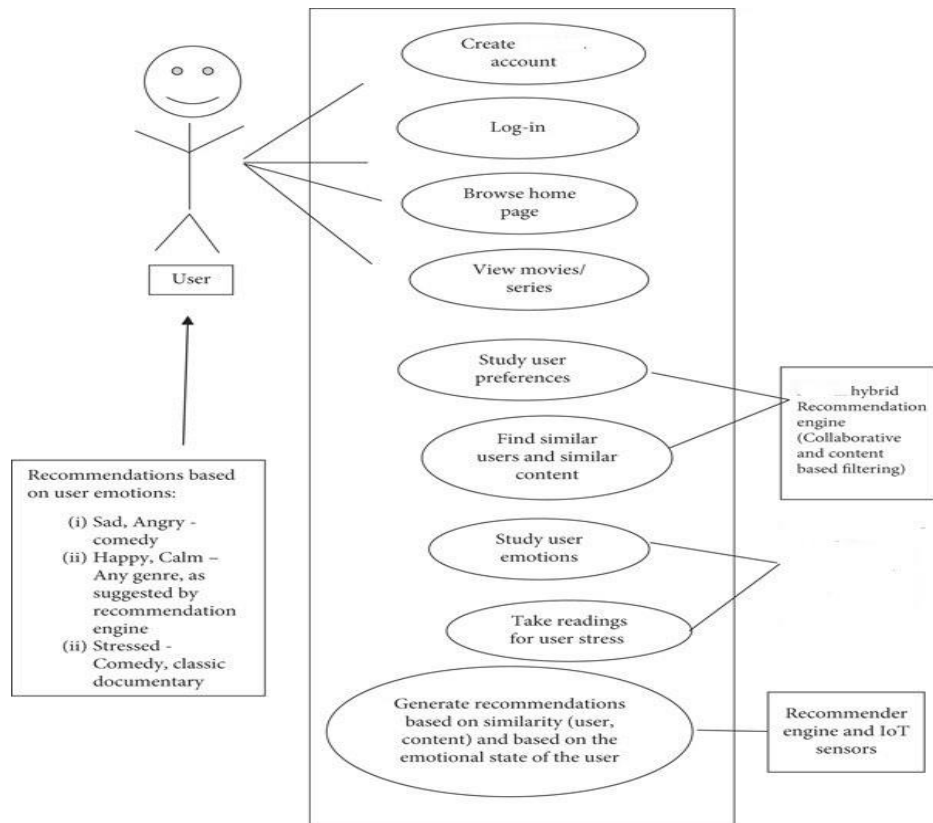
The identification of Actor and Use Case result in the definition of the boundary of the system, i.e., differentiating the tasks accomplished by the system and the tasks accomplished by its environment. The Actors are outside the boundary of the system, where as the Use Cases are inside the boundary of the system. Use Cases describe the behavior of the system, as seen from Actor's point of view.

A Use Case can participate in several Relationships as mentioned below:

Relationship	Function	Notation
Association	To indicate the communication between actors and uses cases.	
Extend	To indicate the insertion of additional behavior into a base use case.	 << extend >>
Include	To describes a behavior that is inserted explicitly into a base use case.	 << include >>
Use case or actor generalization	To indicate the communication between a general use case (actor) and a more specific use case (actor) that inherits and adds features to it.	

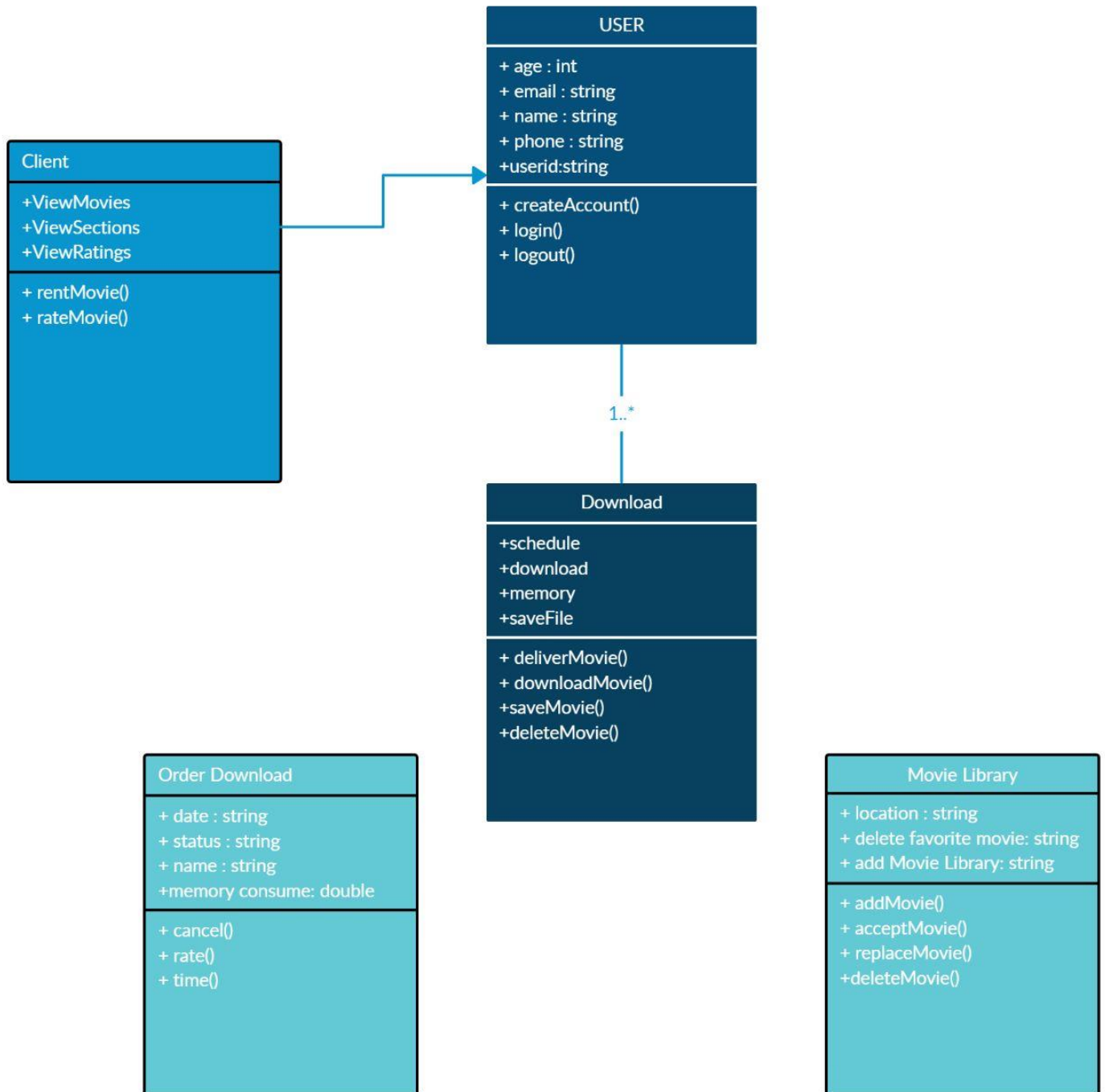
**Table 3.2.2.1 Relationships and their Notation in Use Case Diagrams**

## Use Case Diagrams for Project:



### **3.2.3 CLASS DIAGRAM:**

Class diagram model class structure and contents using design elements such as classes, packages and objects. Class diagram describes 3 perspectives when designing a system-Conceptual, Specification, Implementation. Classes are composed of three things: name, attributes and operations. Class diagrams also display relations such as containment, inheritance, associations etc. The association relationship is most common relationship in a class diagram. The association shows the relationship between instances of classes.



**Fig 3.2.3.1 Class Diagram for Estrella**



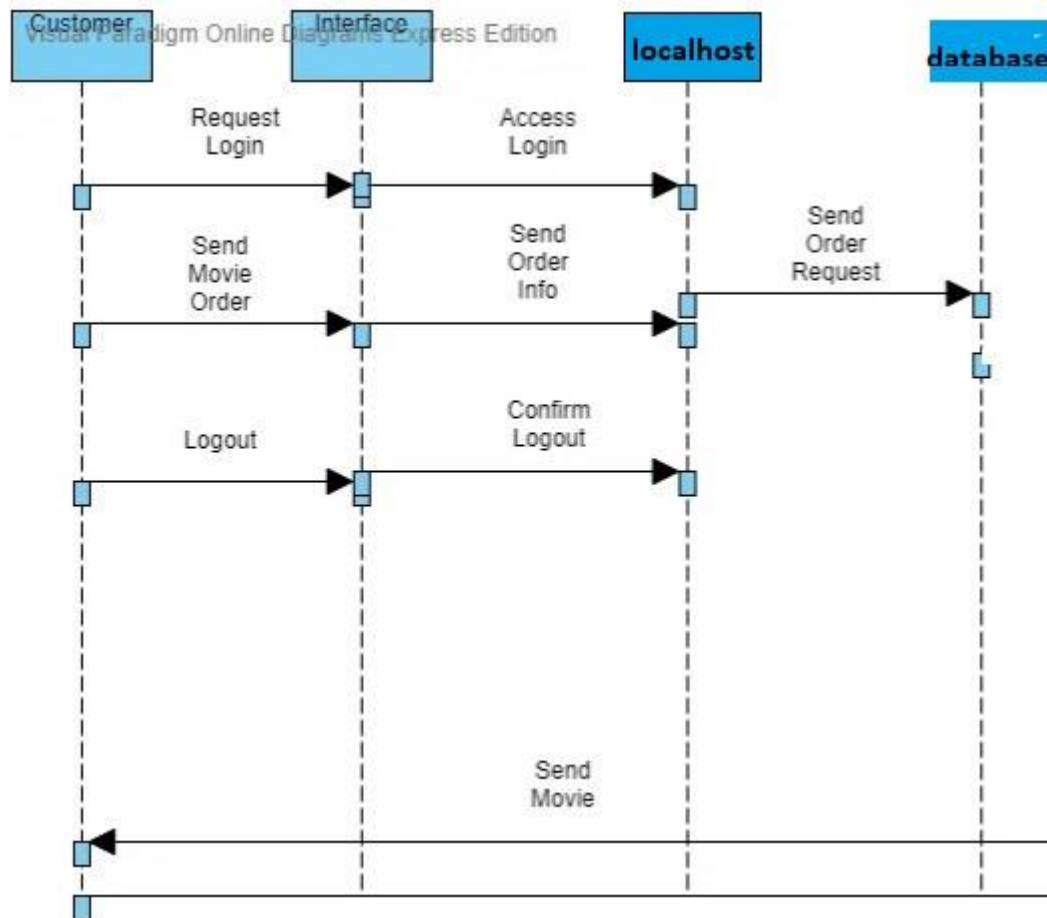
### 3.2.4 SEQUENCE DIAGRAM:

**Sequence Diagrams:** Sequence Diagrams display the time sequence of the objects participating in the interaction. This consists of the vertical dimension (time) and horizontal dimension (different objects).

**Object:** Object can be viewed as an entity at a particular point in time with a specific value and as a holder of identity that has different values over time.

**Actor:** An Actor represents a coherent set of roles that users of a system play when interacting with the use cases of the system.

**Message:** A Message is sending of a signal from one sender object to other receiver objects.



### 2.6 Activity Diagram:

An activity diagram shows the flow from activity to activity. An activity is a going non-atomic execution within a state machine. An activity results in some action, results in a change of state or return of a value.

Activity Diagram commonly contains:

- Activity states and action states
- Transitions
- Objects, it may contain nodes and constraint

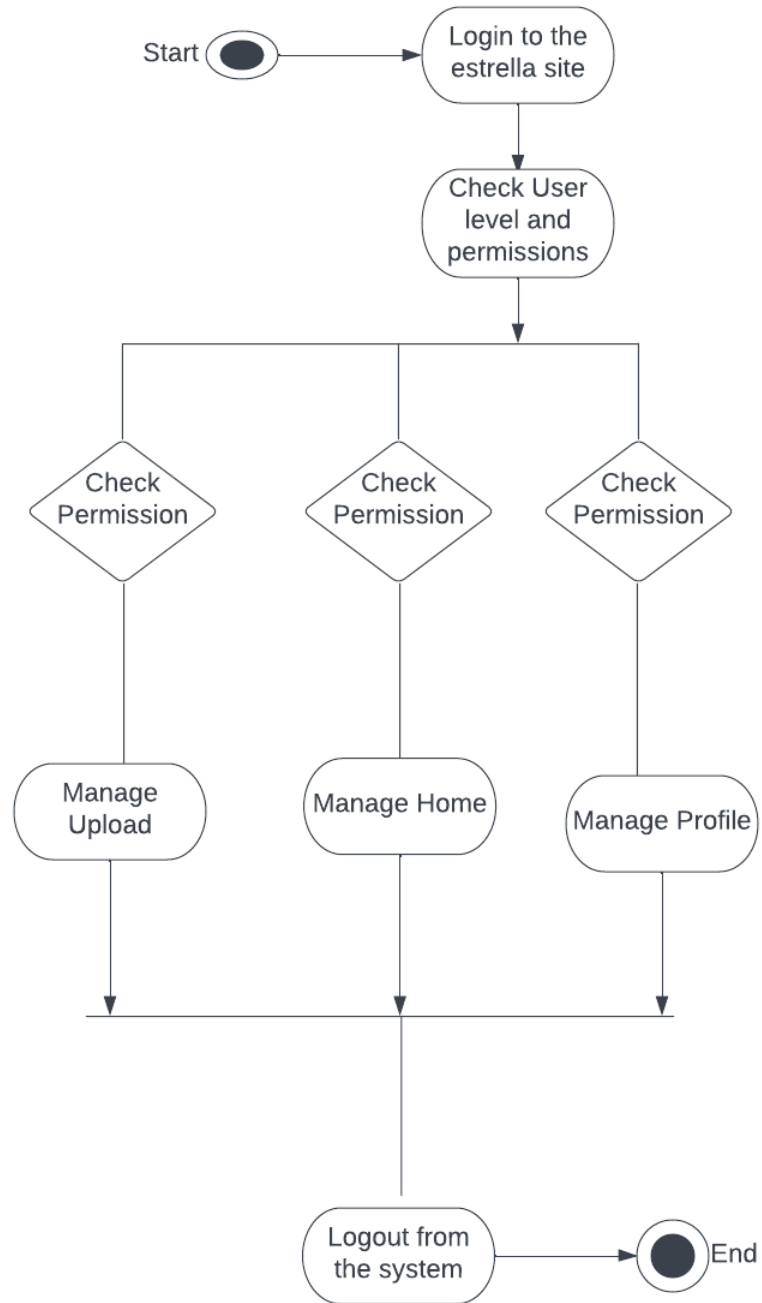
**Activity states and action states:** An executable atomic computation is called action state, which cannot be decomposed. Activity state is non-atomic, decomposable and takes some duration to execute.

**Transition:** It is a path from one state to the next state, represented as simple directed line.

**Branching:** When an alternate path exists, branching arises which is represented by open diamond. It has a incoming transition, two or more outgoing transitions.

**Forking and Joining:** The synchronization bar when split one flow into two or more flows is called fork. When two or more flows are combined at synchronization bar, the bar is called join.

**Swim Lanes:** Group work flow is called swim lanes. All groups are portioned by vertical solid lines. Each swim lane specifies locus of activities and has a unique name. Each swim lane is implemented by one or more classes. Transition may occur between objects across swim lanes.



## 3. DATABASE DESIGN

### 3.1 Database:

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

The database management system(DBMS) is the software that interacts with end users, applications, and the database itself to capture and analyze the data. The DBMS software additionally encompasses the core facilities provided to administer the database. The sum total of the database, the DBMS, and the associated applications can be referred to as a “database system”. Often the term “database” is also used to loosely refer to any of the DBMS, the database system, or an application associated with the database.

### 3.2 ER Diagram:

An Entity Relationship Diagram (ERD) is a visual representation of different entities within a system and how they relate to each other. ER modeling is a data modeling technique used in software engineering to produce a conceptual data model of an information system. Diagrams created using this ER-modeling technique are called Entity-Relationship Diagrams, ER diagrams, or ERDs. So you can say that Entity Relationship Diagrams illustrate the logical structure of databases.

ERDs show entities in a database and relationships between tables within that database. It is essential to have ER-Diagrams if you want to create a good database design. The diagrams help focus on how the database actually works.

ER modeling is one of the most cited papers in the computer software field. Currently, the ER model serves as the foundation of many system analysis and design methodologies, computer-aided software engineering (CASE) tools, and repository systems.

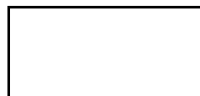
#### 3.2.1 Elements in ER diagram:

Entity relationship diagrams are used in software engineering during the planning stages of the software project. They help to identify different system elements and their relationships with each other. It is often used as the basis for data flow diagrams or DFDs as they are commonly known.

The basic elements in ER-Diagrams:

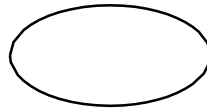
- **Entity:**

Entities are the “things” for which we want to store information. An entity is a person, place, thing, or event. An entity can be represented with Rectangles.



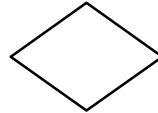
- **Attributes:**

Attributes are the data we want to collect for an entity. An attribute is a property, trait, or characteristic of an entity, relationship, or another attribute. Attributes are represented by Oval shapes.



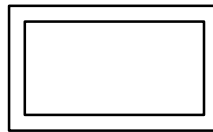
- **Relationships:**

Relationships describe the relations between entities. ERDs show entities in a database and relationships between tables within that database. It is essential to have ER-Diagrams if you want to create a good database design. The diagrams help focus on how the database actually works.



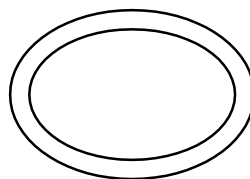
- **Weak Entity:**

A weak entity is an entity that depends on the existence of another entity. In more technical terms it can be defined as an entity that cannot be identified by its own attributes. It uses a foreign key combined with its attributes to form the primary key.



- **Multi-valued Attribute:**

If an attribute can have more than one value it is called a multi-value attribute. It is important to note that this is different from an attribute having its own attributes.

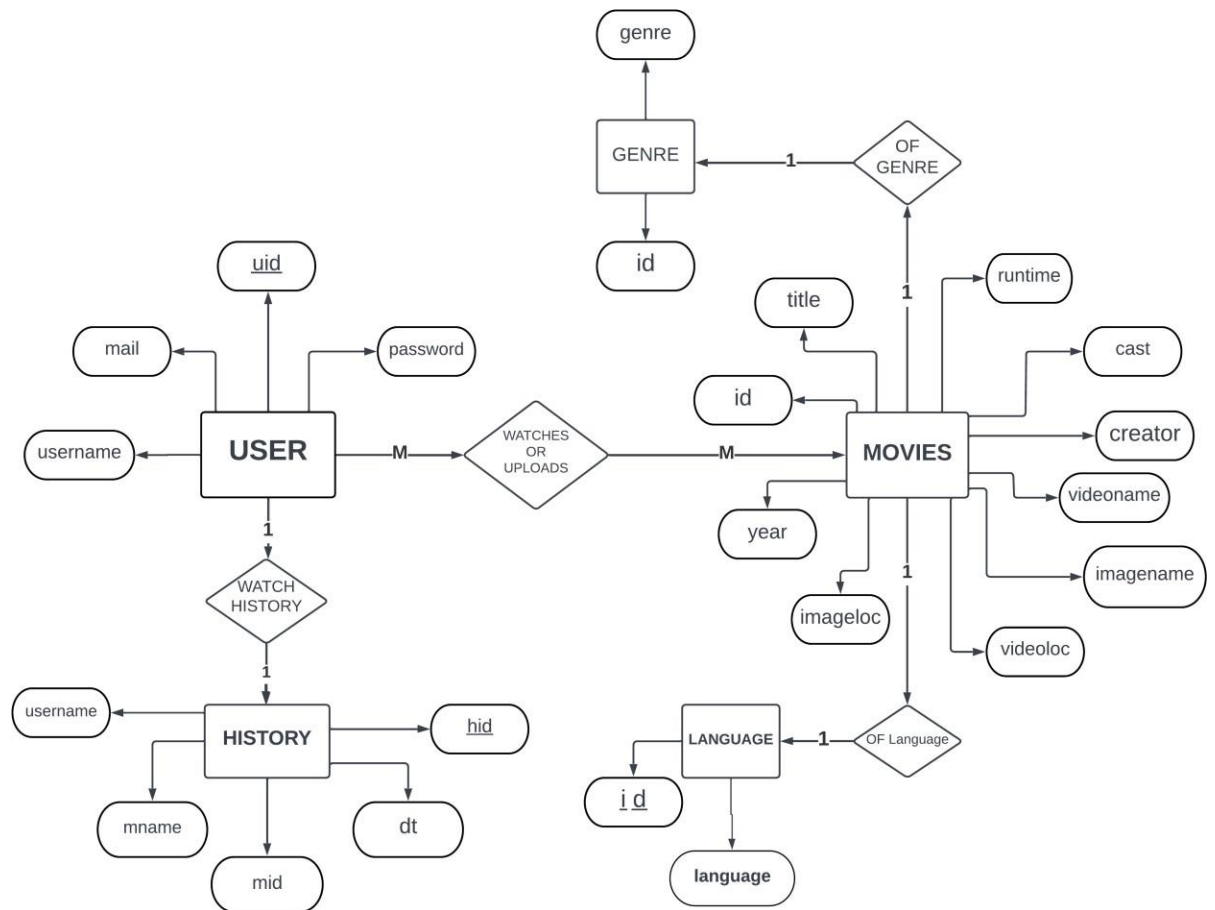


### 3.2.2 How to Draw ER Diagrams:

The below points show how to go about creating an ER diagram.

1. Identify all the entities in the system. An entity should appear only once in a particular diagram. Create rectangles for all entities and name them properly.
2. Identify relationships between entities. Connect them using a line and add a diamond in the middle describing the relationship.
3. Add attributes for entities. Give meaningful attribute names so they can be understood easily.

### 3.2.3 ER Diagram for Project:



**Fig 3.2.3.1 ER Diagram for ESTRELLA**

## Movies table:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	title	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3	runtime	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	4	storyline	text	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5	cast	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	6	year	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	7	creator	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	8	imagename	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	9	imageloc	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	10	videoname	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	11	videoloc	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
	<input type="checkbox"/>	Check all	With selected:  Browse  Change  Drop  Primary  Unique  Index  Spatial  Fulltext							

## User table:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	uid	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	name	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3	email	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4	password	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
	<input type="checkbox"/>	Check all	With selected:  Browse  Change  Drop  Primary  Unique  Index  Spatial  Fulltext							

## Language table:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	language	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More

## History table:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	hid	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	mid	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	3	mname	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4	dt	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5	username	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More

☐ Check all    With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

## Genre table:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	genre	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More





## 4. DATABASE NORMALIZATION

A database design method called normalisation avoids data duplication and gets rid of undesired traits like Insertion, Update, and Deletion Anomalies.

Using relationships, normalisation rules break up larger tables into smaller ones. SQL normalisation serves the dual purposes of removing unnecessary (repetitive) data and ensuring logical data storage. With the introduction of the First Normal Form, the relational model's creator Edgar Codd put out the notion of data normalisation, and he later expanded it with the Second and Third Normal Forms.

Later, he collaborated with Raymond F. Boyce to create the Boyce-Codd Normal Form theory.

- 1NF (First Normal Form)
- 2NF (Second Normal Form)
- 3NF (Third Normal Form)
- BCNF (Boyce-Codd Normal Form)
- 4NF (Fourth Normal Form)
- 5NF (Fifth Normal Form)

### 1NF :

A relation violates first normal form if it has composite or multi-valued attributes, or it is in first normal form if neither of these attributes are present.

If all of the attributes in a relation are solitary valued attributes, the connection is said to be in first normal form.

### 2NF:

- In the 2NF, relational must be in 1NF.
- In the second normal form, all non-key attributes are fully functional dependent on the primary key

### 3NF:

- A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.
- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

A relation is in third normal form if it holds atleast one of the following conditions for every non-trivial function dependency  $X \rightarrow Y$ .

- X is a super key.
- Y is a prime attribute, i.e., each element of Y is part of some candidate key.

### 4NF:

- A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.
- For a dependency  $A \twoheadrightarrow B$ , if for a single value of A, multiple values of B exists, then the relation will be a multi-valued dependency.

## BCNF:

- BCNF is the advance version of 3NF. It is stricter than 3NF.
- A table is in BCNF if every functional dependency  $X \rightarrow Y$ , X is the super key of the table.
- For BCNF, the table should be in 3NF, and for every FD, LHS is super key.

## 5NF:

- A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.
- 5NF is satisfied when all the tables are broken into as many tables as possible in order to avoid redundancy.
- 5NF is also known as Project-join normal form (PJ/NF).

## 4.1 NORMALIZATION IN THE PROJECT:

ENTITIES : User, Movies, Genre, Language, History

As normalization can be derived from functional dependencies,

### 4.1.1 Deriving normal forms from functional dependencies:

- Functional Dependency for Users Relation:

Uid  $\rightarrow$  Uname

Uname  $\rightarrow$  email

Candidate keys-Uid,Uname,email

Super keys- Uname,uid,email

- Deriving normal form:
  1. No multivalued attribute so the relation is in 1NF
  2. No partial dependencies so the relation is in 2NF
  3. No transitive dependency so the relation is in 1NF
  4. All the determinant attributes are super keys so the relation is in BCNF

Therefore, the 'User relation is in BCNF'

- Functional Dependency for History Relation:

mid  $\rightarrow$  mname

Username  $\rightarrow$  dt

Candidate keys-Username,mid

Super keys- Username,mid,hid,dt

- Deriving normal form:
  1. No multivalued attribute so the relation is in 1NF
  2. No partial dependencies so the relation is in 2NF
  3. No transitive dependency so the relation is in 3NF
  4. All the determinant attributes are super keys so the relation is in BCNF

Therefore, the 'History relation is in BCNF'

- Functional Dependency for Language Relation:

id  $\rightarrow$  language

Candidate keys-id

Super keys- id

- Deriving normal form:
  1. No multivalued attribute so the relation is in 1NF
  2. No partial dependencies so the relation is in 2NF

3. No transitive dependency so the relation is in 3NF
4. All the determinant attributes are super keys so the relation is in BCNF

Therefore, the 'Language relation is in BCNF'

- Functional dependency for Genre relation:-

Genre relation:

Functional dependencies-

id  $\rightarrow$  genre

Super key-id

- Deriving normal form:
  1. No multivalued attribute so the relation is in 1NF
  2. No partial dependencies so the relation is in 2NF
  3. No trivial dependency so the relation is in 1NF
  4. All the determinant attributes are super keys so the relation is in BCNF
  5. Therefore, the 'Genre relation is in BCNF'

Movies relation:

Functional dependencies-

id  $\rightarrow$  title

Candidate key-id

Super key-id

1. Deriving normal form:
2. No multivalued attribute so the relation is in 1NF
3. No partial dependencies so the relation is in 2NF
4. No trivial dependency so the relation is in 1NF
5. All the determinant attributes are super keys so the relation is in BCNF
6. Therefore, the 'Movies relation is in BCNF'

As all the relations are in BCNF the project database is in BCNF.

RELATION NAME	ATTRIBUTES	CANDIDATE KEY	NORMALFORM
User	Uid Username Password email	Uid Uname email	BCNF
History	hid mid mname dt Username	Username mid	BCNF
Genre	id , genre	Id	BCNF
Languages	id,language	id	BCNF
Movies	id title runtime storyline cast year creator imagename imageloc videoname videoloc	id	BCNF

“THE PROJECT DATABASE IS IN BCNF”

## 5. IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the system for the users that it will work efficiently and effectively. The system will be implemented only after thorough testing and if it's found thorough according to the specification.

### 5.1 Overview of Software Used:

#### HTML :

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a webpage semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as 'IMG' and 'input' directly introduce content to the page. Other tags such as 'p' surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tag but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. The inclusion of CSS defines the look and layout of content.

#### CSS :

Stands for 'Cascading Style Sheet'. Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML. CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once. For example, a Web developer may want to increase the default text size from 10pt to 12pt for fifty pages of a Web site. If the pages all reference the same style sheet, the text size only needs to be changed on the style sheet and all the pages will show the larger text. While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used to define the cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects. CSS gives Web developers more exact control over how Web pages will look than HTML does. This is why most Web pages today incorporate cascading style sheets.

## **MYSQL:**

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons –

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

## **PHP:**

PHP means – **Personal Home Page**, but it now stands for the recursive backronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks.

A PHP file can also contain tags such as HTML and client side scripts such as JavaScript.

- HTML is an added advantage when learning PHP Language. You can even learn PHP without knowing HTML but it's recommended you at least know the basics of HTML.
- Database management systems DBMS for database powered applications.
- For more advanced topics such as interactive applications and web services, you will need JavaScript and XML.
- PHP is open source and free.
- Short learning curve compared to other languages such as JSP, ASP etc.
- Large community document
- Most web hosting servers support PHP by default unlike other languages such as ASP that need IIS. This makes PHP a cost-effective choice.
- PHP is regularly updated to keep abreast with the latest technology trends.
- Another benefit that you get with PHP is that it's a server-side scripting language; this means you only need to install it on the server and client computers requesting resources from the server do not need to have PHP installed; only a web browser would be enough.

- HP has built support for working hand in hand with MySQL; this doesn't mean you can't use PHP with other database management systems. You can still use PHP with
  - gres
  - racle
  - [SQL](#) Server
  - C etc.
- HP is cross-platform; this means you can deploy your application on a number of different operating systems such as windows, Linux, Mac OS, etc.
- In terms of market share, there are over 20 million websites and applications on the internet developed using PHP scripting language.

## 5.2 Coding:

### VIDEO.php

```
<?php
include("config.php");
session_start();
$check = $_SESSION['email'];
$select = " SELECT * FROM login WHERE email = '$check' "
$result = mysqli_query($conn, $select);
while($row = mysqli_fetch_assoc($result)){
    $user = $row['name'];
}
if($check == true){

}
else{
    header('location:login.php');
}
?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" href="images/est.png">
    <title>
        PLAYER
    </title>
    <link rel="stylesheet" href="style.css" class="css">
    <link rel="stylesheet" href="vapp.css">
    <!-- GOOGLE FONTS -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;300;400;600;700;900&display=
```



```

swap" rel="stylesheet">
<!-- OWL CAROUSEL -->
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.css"
integrity="sha512-
tS3S5qG0BlhnQROyJXvNjeEM4UpMXHrQfTGmbQ1gKmelCxlSEBUaxhRBj/EFTzpbP4RVSrp
EikbmdJobCvhE3g==" crossorigin="anonymous" />
<!-- BOX ICONS -->
<link href='https://unpkg.com/boxicons@2.0.7/css/boxicons.min.css' rel='stylesheet'>
</head>
<body>
<!-- NAV -->
<div class="nav-wrapper">
<div class="container">
<div class="nav">
<a href="index.php" class="logo">
<i class='bx bxs-star bx-tada main-color'></i> E<span class="main-
color">S</span>TRELLA<i class='bx bxs-movie-play bx-tada main-color'></i>
</a>
<ul class="nav-menu" id="nav-menu">
<li><a href="home.php">Home</a></li>
<li><a href="genre.php">Genre</a></li>
<li><a href="language.php">Language</a></li>
<li><a href="upload.php">Upload</a></li>
<li><a href="profile.php">Profile</a></li>
</ul>
<!-- 
<ul class="menu-box" id="menubox">
<li><a href="index.php" class="btn btn-hover"><span>Home</span></a></li>
<li><a href="genre.php" class="btn btn-hover"><span>Genre</span></a></li>
<li><a href="language.php" class="btn btn-hover"><span>Language</span></a></li>
<li><a href="upload.php" class="btn btn-hover"><span>Upload</span></a></li>
<li><a href="profile.php" class="btn btn-hover"><span>Profile</span></a></li>
</ul> -->
<!-- MOBILE MENU TOGGLE -->
<div class="hamburger-menu" id="hamburger-menu">
<div class="hamburger"></div>
</div>
</div>
</div>
</div>
<!-- <script>
let menubox = document.getElementById("menubox");
let menuicon = document.getElementById("menuicon");

menuicon.onclick = function(){
    menubox.classList.toggle("open-menu");

    if(menubox.classList.contains("open-menu")){
        menuicon.src = "images/close.png";
    }
    else{
        menuicon.src = "images/menu.png";
    }
}

```

```

    }
</script> -->
<!-- END NAV -->

<div class="btn">
    <div class="play"></div>
    <p>Play Video</p>

    <?php
        date_default_timezone_set("Asia/Kolkata");
        $dt=date("d/m/y h:i:sa");
        $x=$_GET["id"];
        $fetchVideos = mysqli_query($conn, "SELECT * FROM movies,language,genre
where movies.id=genre.id and movies.id=language.id and movies.id=$x");
        while($row = mysqli_fetch_assoc($fetchVideos)){
            $location = $row['videoloc'];
            $title = $row['title'];
            $genre = $row['genre'];
            $language = $row['language'];
            $storyline = $row['storyline'];
            $cast = $row['cast'];
            $year = $row['year'];
            $creator = $row['creator'];
        }
        echo "
            <h1>$title</h1>
            <p>$storyline</p>
            <p>Genre - $genre</p>
            <p>Language - $language</p>
            <p>Year - $year</p>
            <p>Starring &ensp;</p><p>$cast</p>
            <h2>Uploaded by $creator</h2>
        ";
        $query1="INSERT INTO history(mid,mname,username,dt)
VALUES('".$x."','".$title."','".$user."','".$dt."')";
        mysqli_query($conn,$query1);
    ?>
</div>
<div class="clip">
    <?php
        //$name = $row['name'];
        echo "<div >
            <video src='".$location.'" controls ></video>
            <h2>$title</h2>
        </div>";
    ?>

    <b class="close">Close</b>
</div>
<script>
    let btn=document.querySelector('.btn');
    let clip=document.querySelector('.clip');
    let close=document.querySelector('.close');
    let video = document.querySelector('video')
    btn.onclick = function(){

```

```

        btn.classList.add('active')
        clip.classList.add('active')
        video.play();
        video.currentTime=0;
    }
    close.onclick = function(){
        btn.classList.remove('active');
        clip.classList.remove('active');
        video.pause();
    }
}

</script>
</body>
</html>

```

## Vapp.css:

```

:root {
  --main-color: #BF40BF;
  --body-bg: #181616;
  --box-bg: #221f1f;
  --text-color: #ffffff;
  --nav-height: 60px;
  --space-top: 30px;
}

* {
  padding: 0;
  margin: 0;
  box-sizing: border-box;
}

html {
  font-size: 16px;
}

body {
  font-family: "Cairo", sans-serif;
  background-color: var(--body-bg);
  color: var(--text-color);
  padding-top: var(--nav-height);
}

a {
  text-decoration: none;
  color: unset;
}

img {
  max-width: 100%;
}

.main-color {
  color: var(--main-color);
}

```

```

.container {
  max-width: 1920px;
  padding: 0 40px;
  margin: auto;
}

.overlay {
  position: absolute;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  background-color: rgba(0, 0, 0, 0.5);
}

.nav-wrapper {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  z-index: 99;
  background-color: #000000;
}

.nav {
  display: flex;
  align-items: center;
  justify-content: space-between;
  color: var(--text-color);
  height: var(--nav-height);
}

.nav a {
  color: var(--text-color);
}

.logo {
  font-size: 2rem;
  font-weight: 900;
}

.nav-menu {
  list-style-type: none;
  display: flex;
  align-items: center;
  padding: 0 20px;
}

.nav-menu li ~ li {
  margin-left: 30px;
}

.nav-menu a {
  text-transform: uppercase;
}

```

## 6. TESTING

Testing is done to look for mistakes. The process of testing include looking for any potential flaws or weaknesses in a product. It offers a means of examining the operation of parts, subassemblies, assemblies, and/or a finished product.

It is the process of testing software to make sure that it satisfies user expectations and meets requirements without failing in an unacceptable way.

Software testing is a crucial component of software quality assurance and is the last step in the definition, design, and coding review process. The expense of software failures and the growing viability of software as a system are driving pressures for properly thought-out, thorough testing.

### Testing Objectives:

The following guidelines can be used as test objectives:

- Programs are tested by being run with the goal of identifying errors.
- A test case is effective if it has a strong probability of recognizing an unknown error.

### 6.1 Types of Testing

The various tiers of testing strategies that are used at various stages of software development to ensure that the system is error-free are:

#### 6.1.1 Unit Testing

Individual modules are subjected to unit testing as they are finished and ready for execution. It is primarily restricted by the demands of the designer. Unit testing is distinct from and should happen before all the other approaches, which including:

- Inform debugging:  
Debug information is a group of data produced by a C/C++ compiler or an assembly programme and used by a debugger to explain an application.
- Code debugging:  
Debugging is the process of iteratively running your code in a debugging environment, such as Visual Studio, to identify the precise place where a programming error occurred. After that, you are aware of the changes your code needs.

Each module can be tested using the following two strategies:

1. Black Box Testing
2. White Box Testing

### **Black Box Testing:**

This approach generates some test cases that fully execute all of the program's functional requirements as input conditions. The following categories of error have been identified using this testing:

- Wrong or absent functionalities.
- Interface mistakes.
- Inaccuracies in external database access or data structure and Error in performance.
- Errors in initialization and termination.
- Only the output is examined for accuracy in this testing.
- The data's logical flow is not verified.

### **White Box Testing**

By defining the module's flow diagrams, test cases are created here for the logic of each module, and all cases are tested for logical decisions.

The following cases have made use of it to create test cases:

- Assure the completion of all separate pathways.
- Complete all loops while staying within their defined operating boundaries.
- Put internal data structures to use to verify their accuracy.

### **6.1.2 Integration Testing**

Integration testing guarantees that software and related systems function as a unit. It evaluates each module's interface to ensure that the modules function properly when combined. Developers often carry it out, particularly at the lower, module-to-module level. At the higher stages, testers participate.

### **6.1.3 System Testing**

Before even being delivered to the user, involves in-house testing of the complete system. The objective is to meet all the client's criteria while also satisfying the user. If a business has a testing agency, they run the test. Hand-generated toproduction-level test results are both acceptable.

To plan and organise tests, the following elements must be considered:

- Test data to be used;
- Inclusion of changes/fixes.

One common approach is graduated testing:

as system testing progresses and (hopefully) fewer and fewer defects are found, the code is frozen for testing for increasingly longer time periods.

### 6.1.4 Acceptance Test

It is a pre-delivery test in which the complete system is examined for flaws using real-world data at the client's location.

User Acceptance Test (UAT)

“**Beta Testing**”: Acceptance testing in the customer environment.

Traceability of requirements:

- Correlate requirements with test cases.
- At least one test case must pass for each criterion.
- Present requirements versus test cases in a matrix.

## 6.2 Test Cases

A test case is typically a collection of test data, a test program, and the expected outcomes. In software engineering, a test case typically includes a unique identifier, references to requirements from a design specification, preconditions, events, a series of steps (also known as actions to take), input, and output. It also validates one or more system requirements and generates a pass or fail result.

### Test Cases for the project:

In general, a test case is a set of test data and test programs and their expected results. A test case in software engineering normally consists of a unique identifier, requirement references from a design specification, pre-conditions, events, and a series of steps (also known as actions) to follow input, and output and it validates one or more system requirements and generates a pass or fail.

#### 6.2.1 Login

Test plan id: Pid1

Test case id: 101

Features to be tested: Login

Pre conditions: 1. Website must be running  
2. Valid Login Details

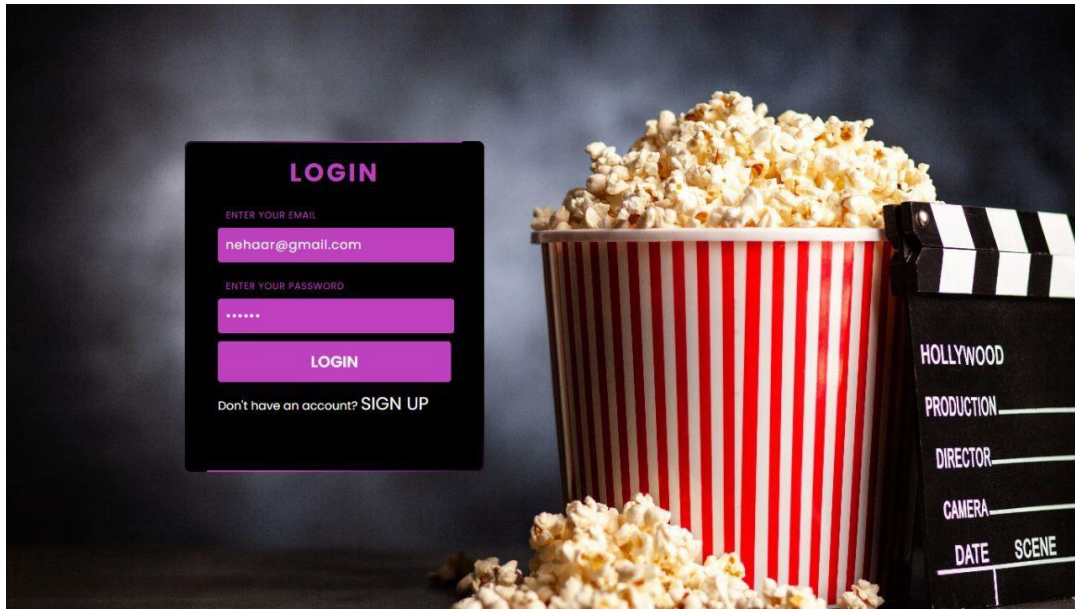
Test Script: 1. Verify Login Details  
2. Check Customer Credentials  
3. Accept and Update the Database if details are valid

Test Data: 1. Valid credentials that are filled  
2. Invalid credentials that are filled

Expected Results: 1. Successful login  
2. Un successful login

Test Status : Pass

**Successful login:**



### **6.2.2 Movie Uploading**

Test plan id: Pid2

Test case id: 102

Features to be tested: uploading a film

Pre conditions: 1. Website must be running  
2. Valid Login Details

Test Script: 1. Verify Login Details  
2. Check user Credentials  
3. Accept and Update the Database if details are valid

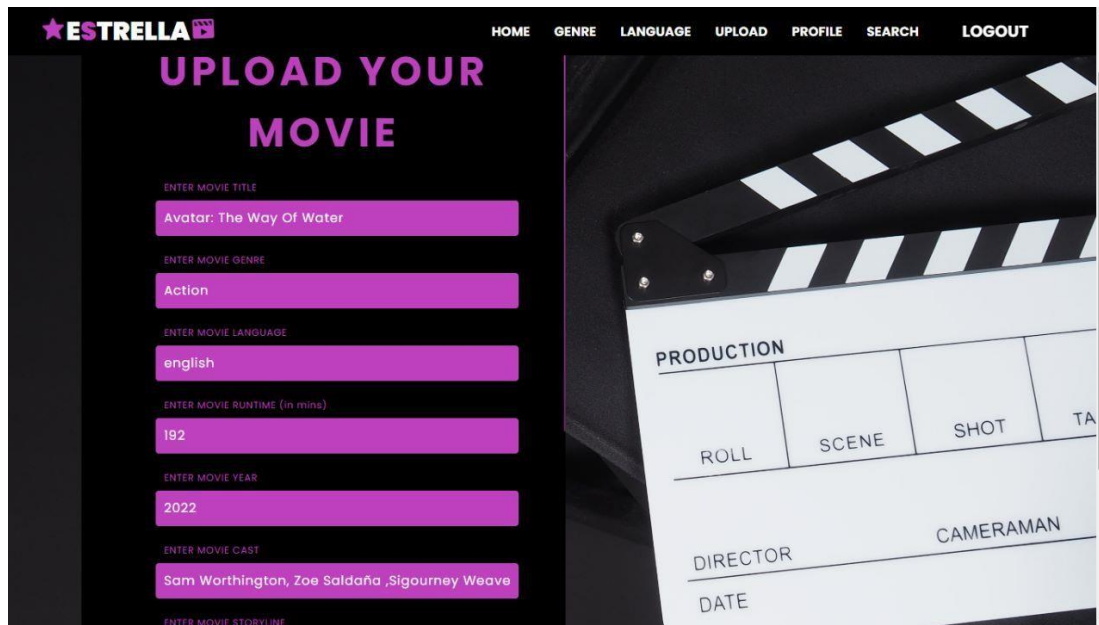
Test Data: 1. Valid credentials that are filled  
2. Invalid credentials that are filled

Expected Results: 1. upload the film as required

Test Status : Pass



Upload the film as required:



The screenshot shows a web application interface for uploading a movie. The header includes the logo '★ESTRELLA' and navigation links: HOME, GENRE, LANGUAGE, UPLOAD, PROFILE, SEARCH, and LOGOUT. The main heading is 'UPLOAD YOUR MOVIE'. The form contains several input fields with the following values:

- ENTER MOVIE TITLE: Avatar: The Way Of Water
- ENTER MOVIE GENRE: Action
- ENTER MOVIE LANGUAGE: english
- ENTER MOVIE RUNTIME (in mins): 192
- ENTER MOVIE YEAR: 2022
- ENTER MOVIE CAST: Sam Worthington, Zoe Saldana, Sigourney Weaver
- ENTER MOVIE STORYLINE: (empty)

The background of the form is a clapperboard with the following text:

PRODUCTION

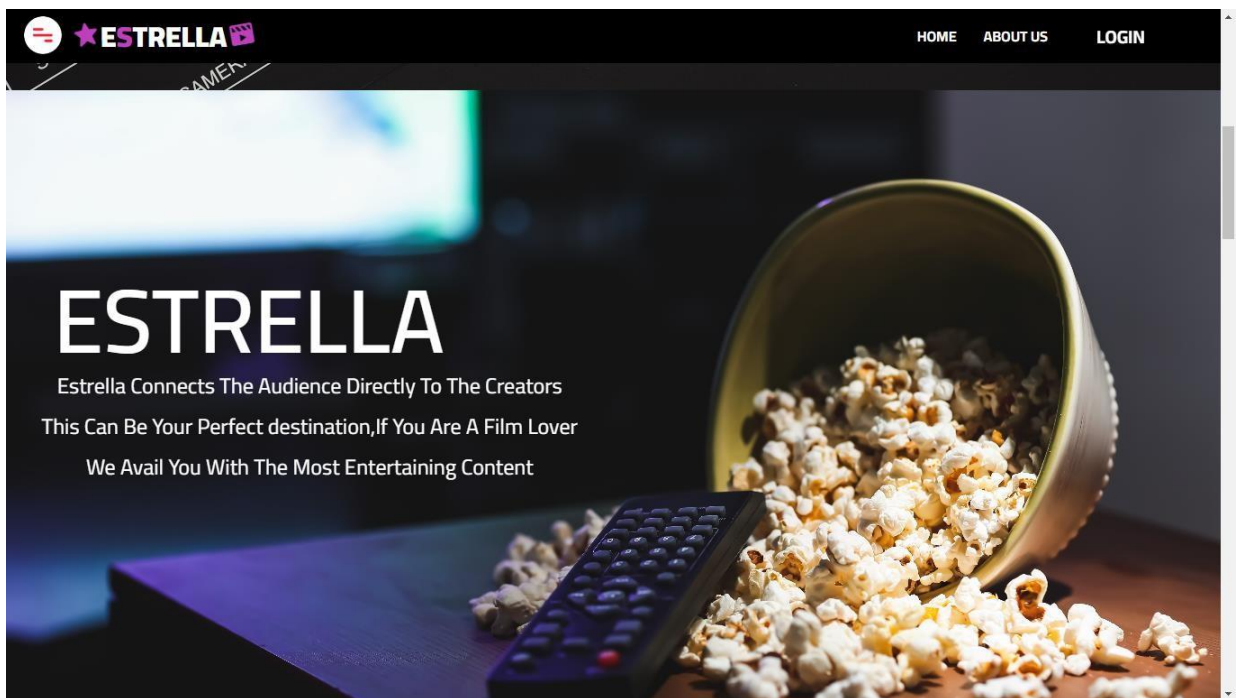
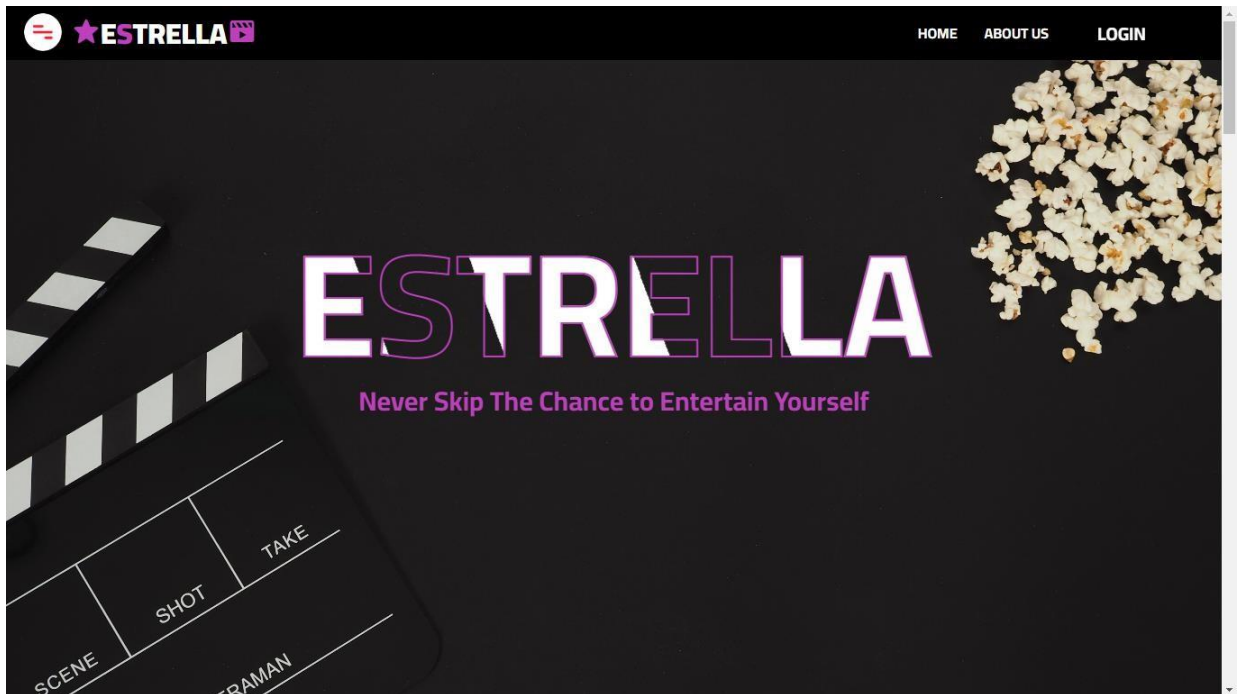
ROLL	SCENE	SHOT	TA
DIRECTOR		CAMERAMAN	
DATE			

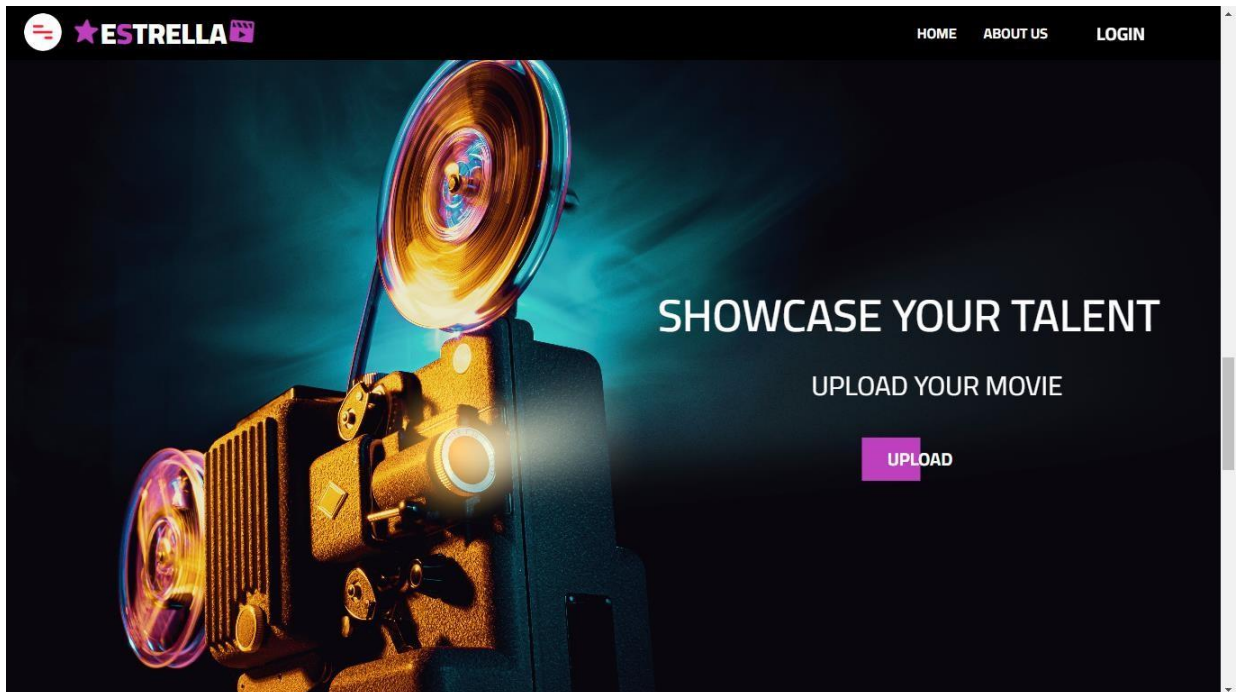
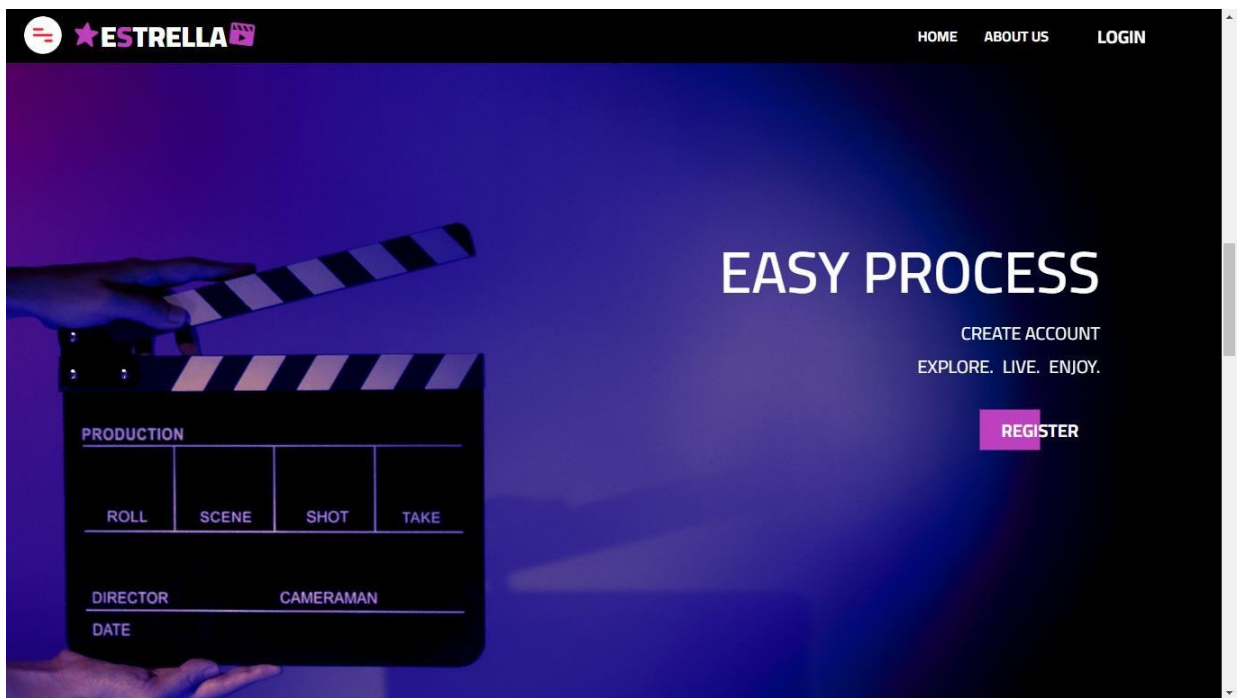
localhost says  
Upload successfully

OK

## 7. RESULTS

### 7.1 Landing Page





ENTER THE WORLD OF

# ESTRELLA

ENTER

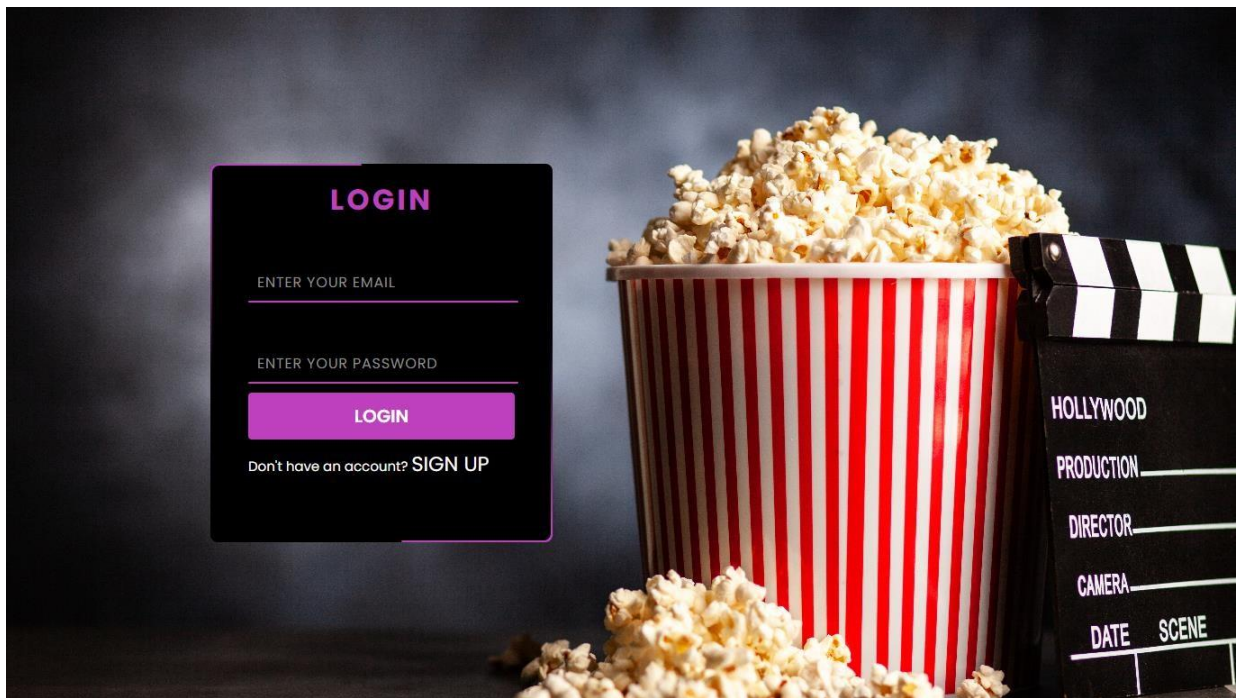
## **ESTRELLA**

Estrella means a Star that Shines over period lighting up opportunities for independent film-makers around the globe. This is a platform provided for independent to share their work

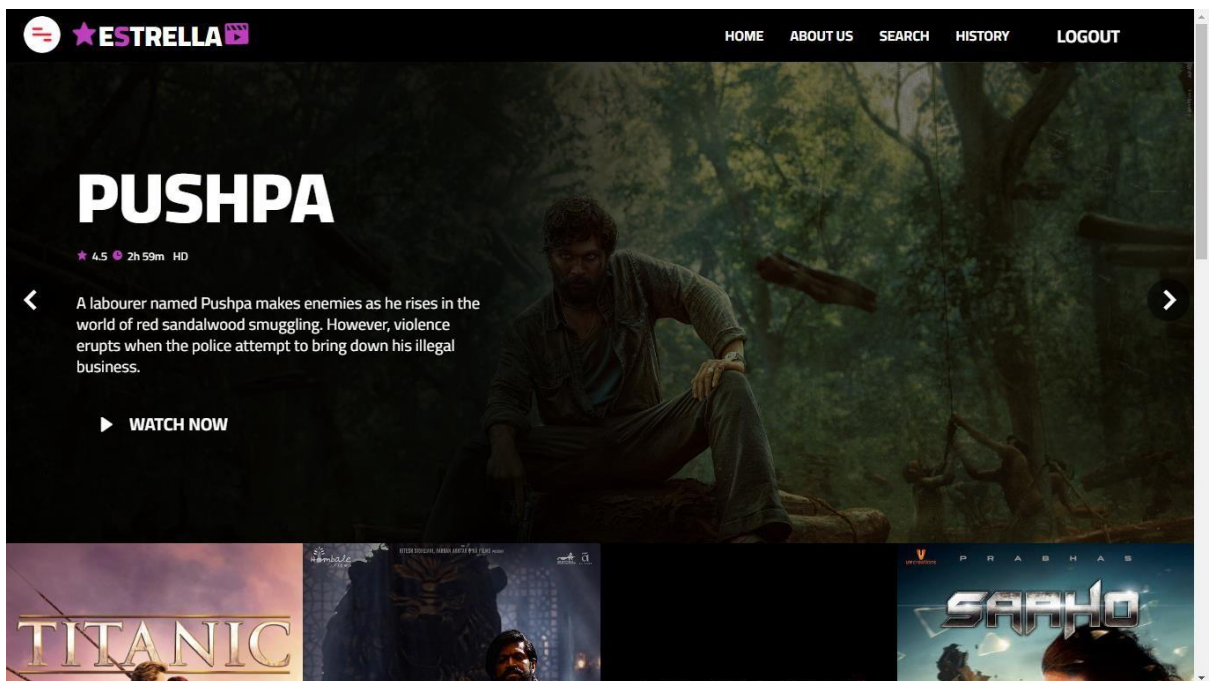
This platform is going to be the most loved by the film-makers and viewers, we intend to keep no creative restrictions over the maker's work and let our viewers enjoy cinema to its extent.

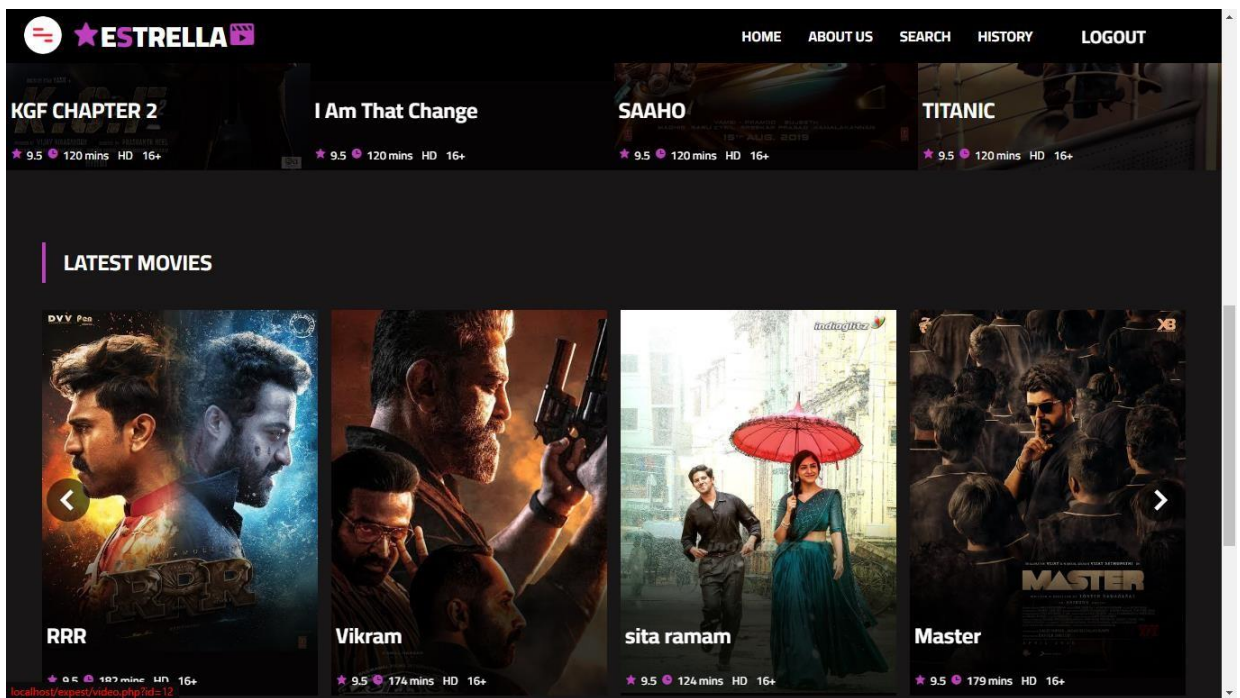


## 7.2 Login

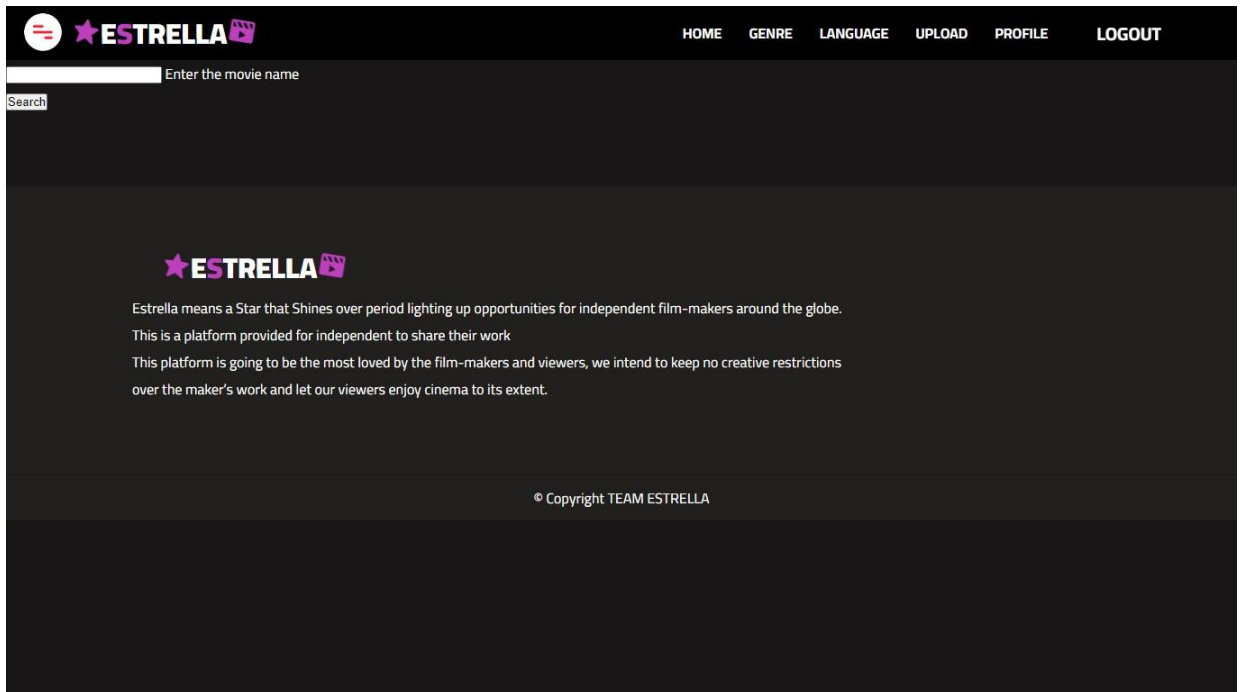


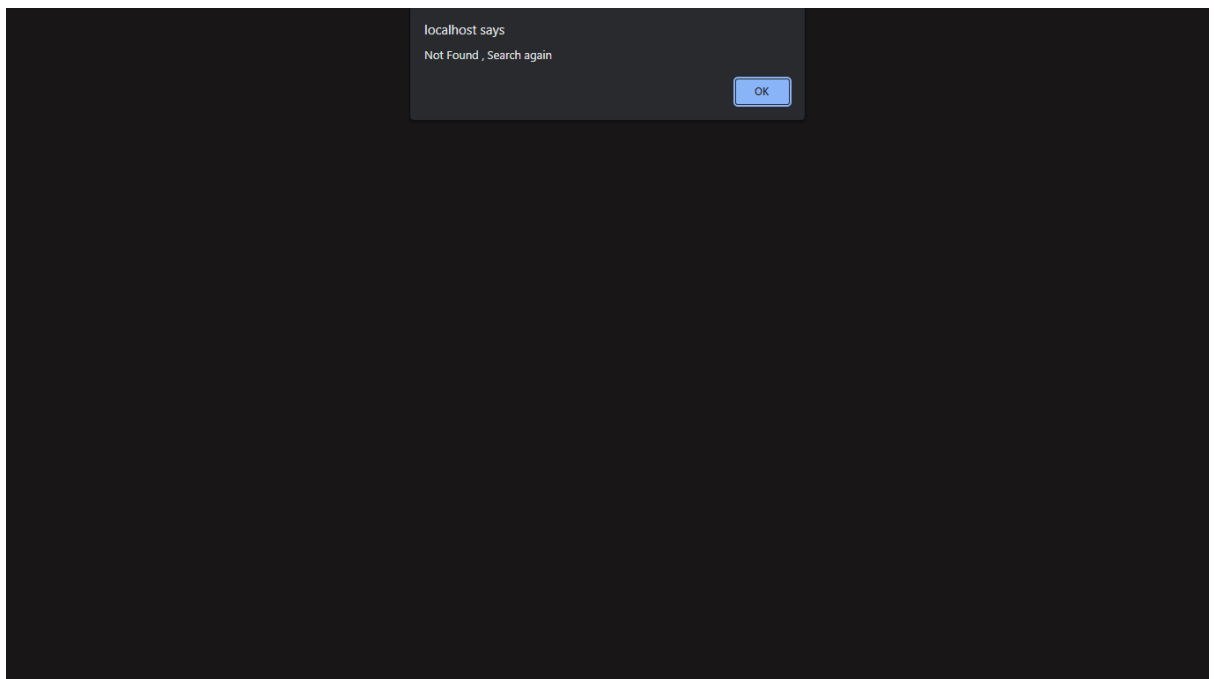
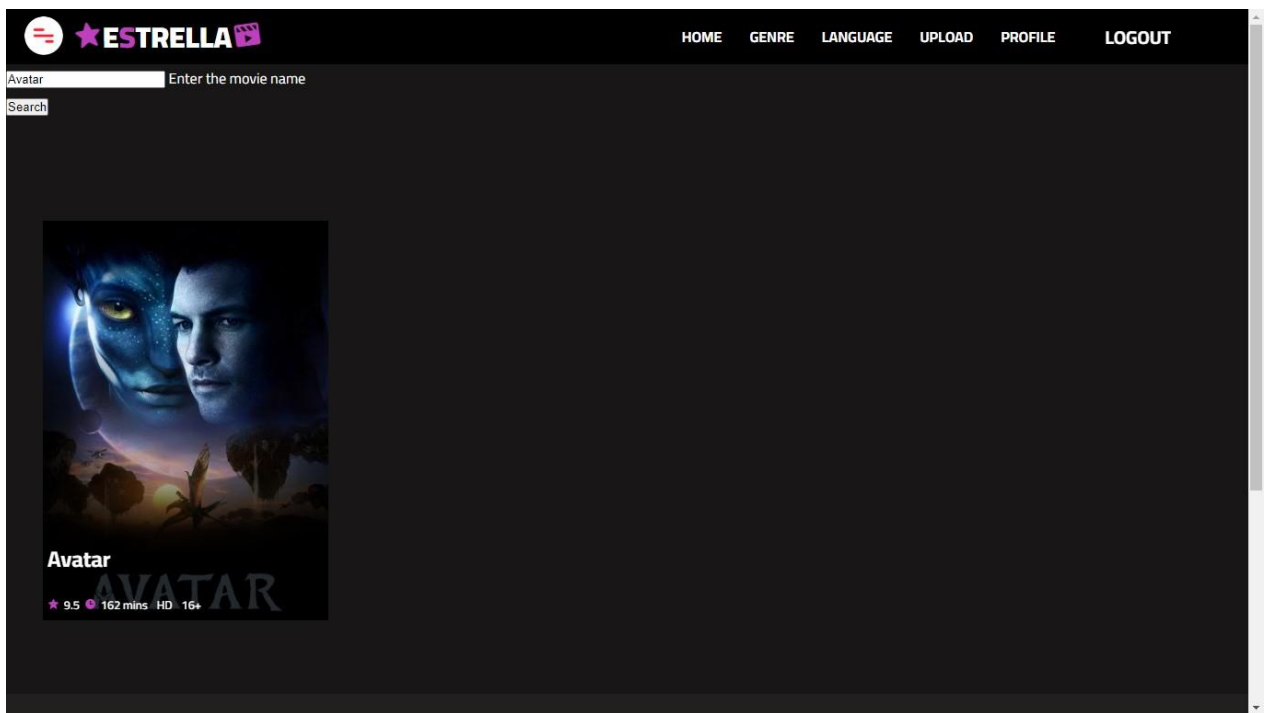
## 7.3 Home



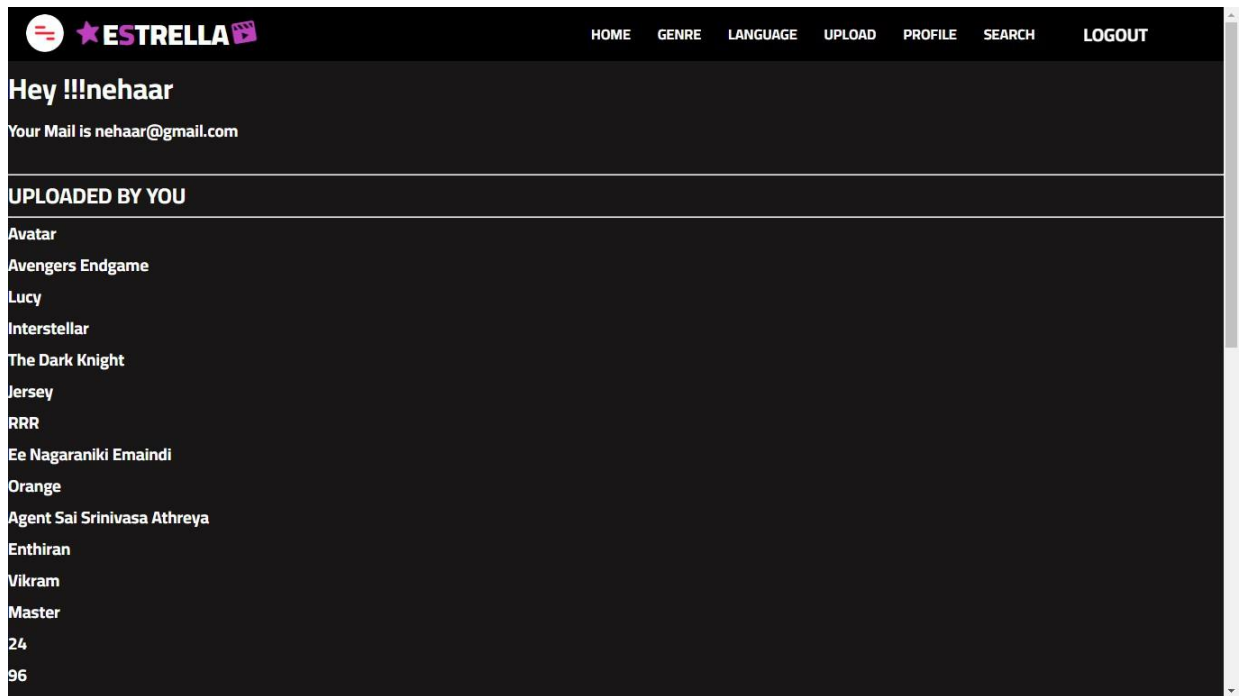


## 7.4 Search

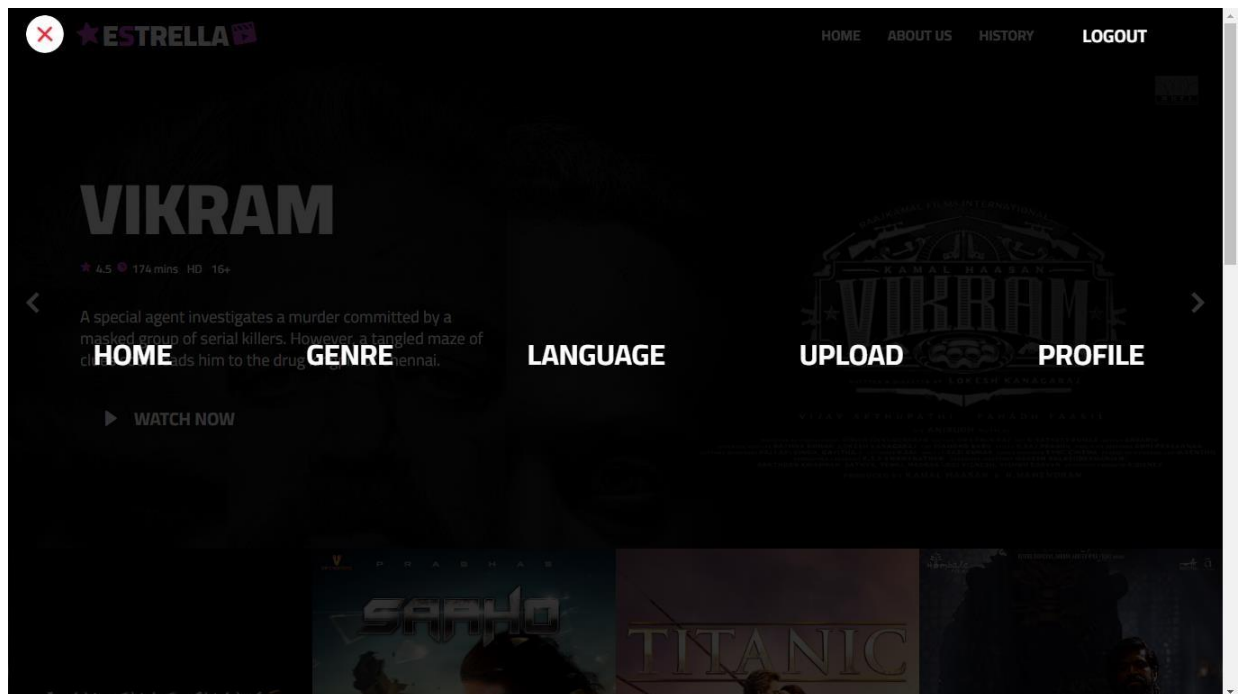




## 7.5 Profile

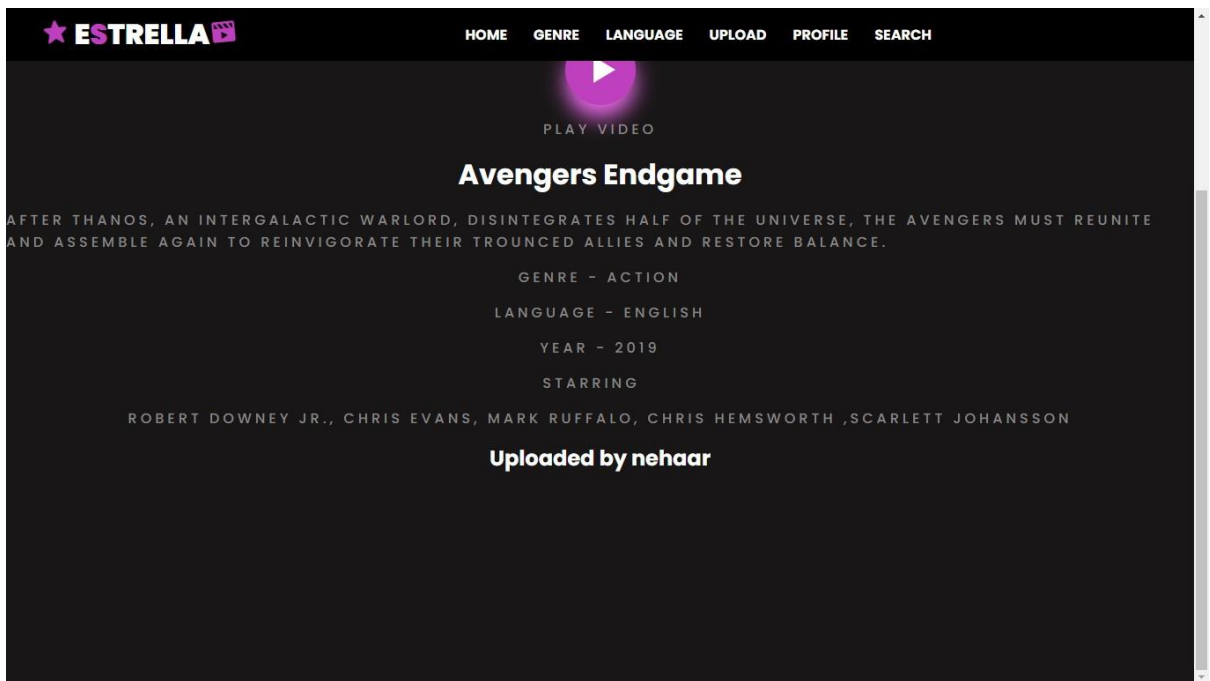
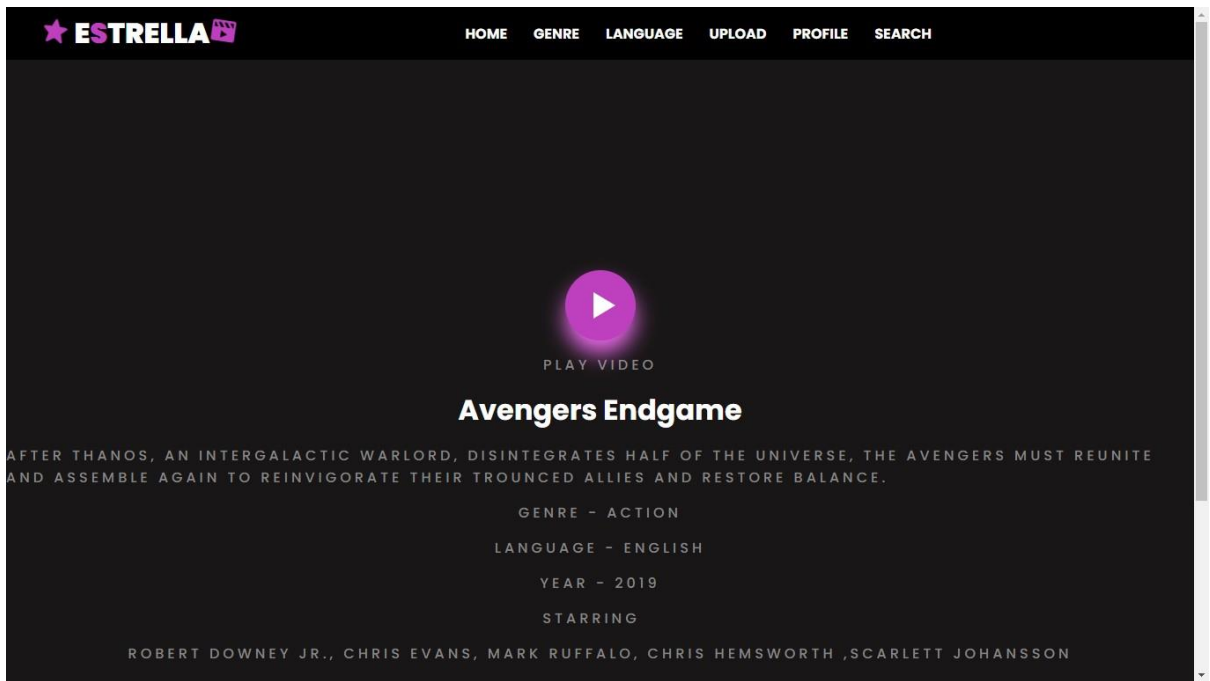


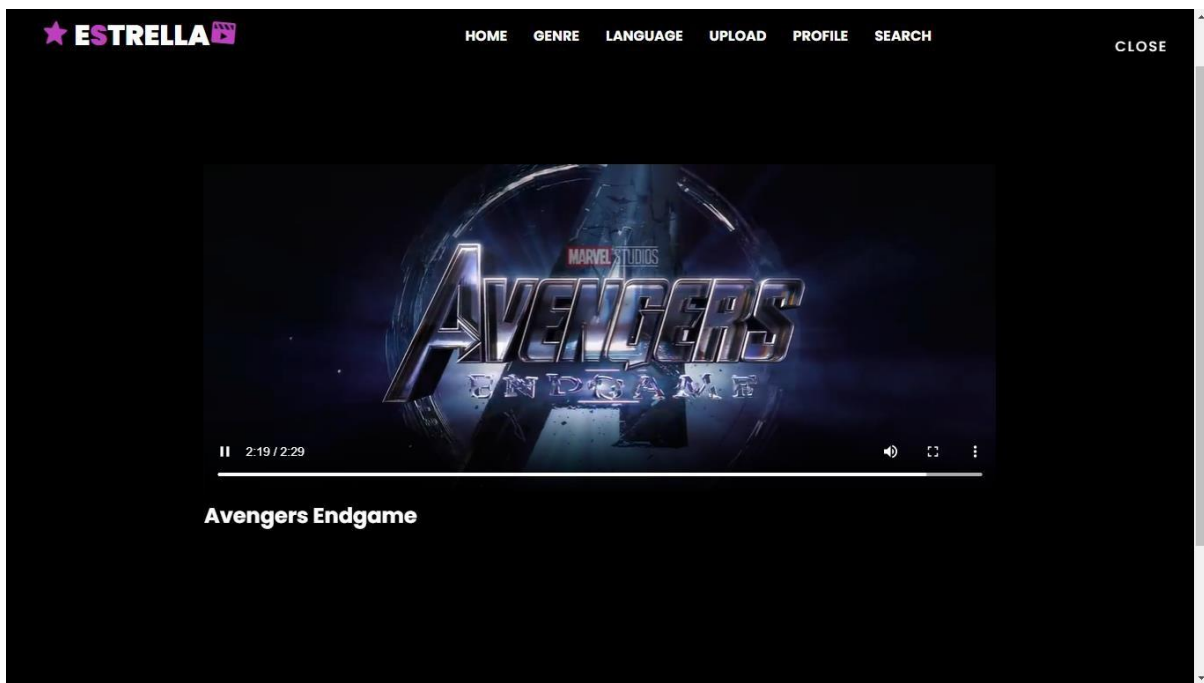
## 7.6 Hamburger Menu



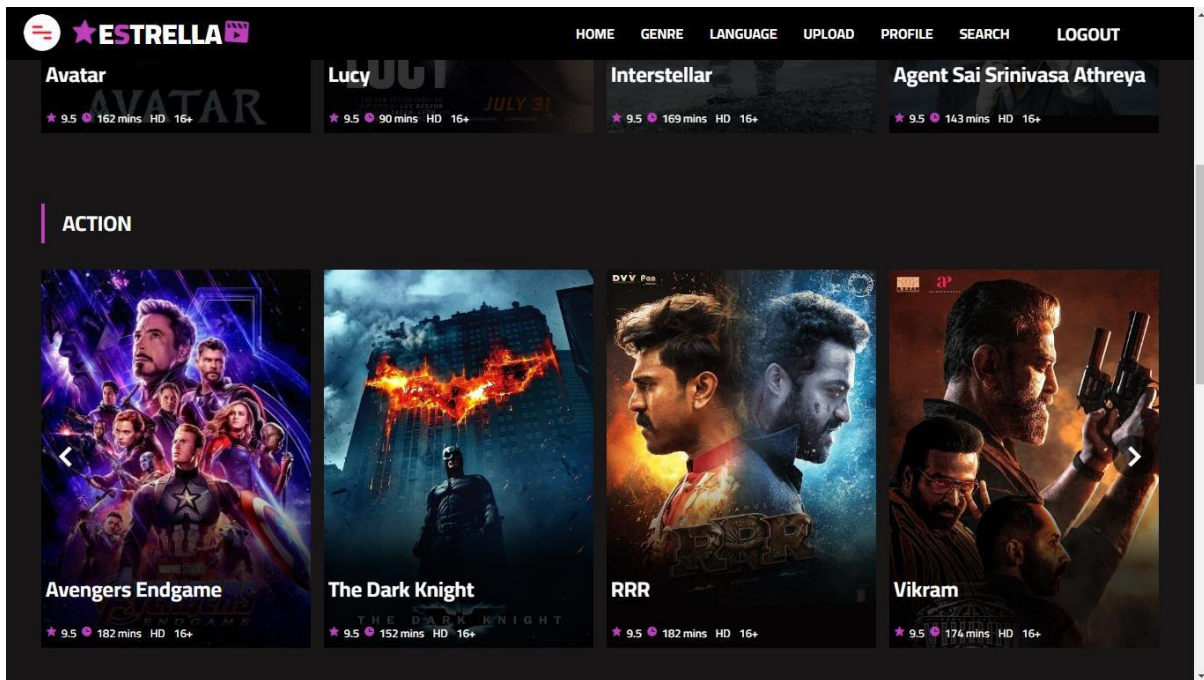


## 7.7 Video Player







## 7.8 Genre



## 7.9 Language


HOME GENRE LANGUAGE UPLOAD PROFILE SEARCH LOGOUT

ENGLISH



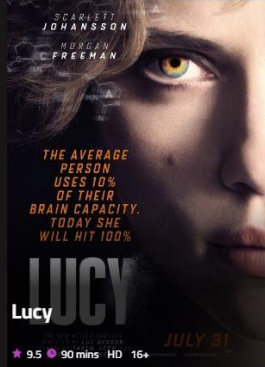
**Avatar**

★ 9.5 162 mins HD 16+




**Avengers Endgame**

★ 9.5 182 mins HD 16+



**Lucy**

★ 9.5 90 mins HD 16+



**Interstellar**

★ 9.5 169 mins HD 16+

TELUGU

## **8. CONCLUSION AND FUTURE SCOPE**

### **8.1 Conclusion:**

This proposed system “Estrella” offers all functionalities of the existing Streaming services and it aims to make the availability of services exist on the internet for free of cost and at your fingertips.

### **8.2 Future Scope:**

In the future, we would try to enhance the responsiveness of the system while watching the content. And we can develop the interface to be much more friendly to the user.

## 9. REFERENCES

### Web Sites:

1. <https://www.w3schools.com/html/>
2. <https://www.tutorialspoint.com/javascript/>
3. <https://www.geeksforgeeks.org/web-technology/>
4. <https://www.javatpoint.com/php-json-example>

### Textbooks:

1. Steven Holzner, “HTML Black Book: The Programmer’s Complete HTML Reference Book”.
2. Robin Nixon, “Learning PHP, MySQL, and JavaScript”, 4 th edition.