
Software Requirements Specification

for

MovieBase

Version 1.0 approved

Prepared by

Group 21

Ashwin S B210508CS ashwin_b210508cs@nitc.ac.in

Abishek N B210454CS abishek_b210454cs@nitc.ac.in

Abijith R B210568CS abijith_b210568cs@nitc.ac.in

Johann B B210510CS johann_b210510cs@nitc.ac.in

Lehan D B210524CS lehan_b210524cs@nitc.ac.in

INSTRUCTORS: Prof. Abdul Nazeer & Prof. M Prabhu

COURSE: Database Management Systems

DATE: 21/10/2023

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 Definitions and Abbreviations	2
1.6 References	2
2. Overall Description	3
2.1 Product Overview	3
2.2 Product Functionality	4
2.3 User Classes and Characteristics	4
2.4 Design and Implementation Constraints	4
2.5 Assumptions and Dependencies	5
3. External Interface Requirements	6
3.1 User Interfaces	6
3.2 Hardware Interfaces	6
3.3 Software Interfaces	6
4. Functional Requirements	8
4.1 Functional specification	8
4.2 Use Case Model	9
5. Other Nonfunctional Requirements	16
5.1 Performance Requirements	16
5.2 Safety Requirements & Security Requirements	16
5.3 Software Quality Attributes	16
6. Appendix A - Data Dictionary	17
7. Appendix B - Group Log	18

1 Introduction

This project is a prototype for MovieBase, a web application created to enhance the exploration and management of cinematic content, encompassing features such as movie searching, viewing, rating, reviewing and more. This section provides a fundamental overview of the project's goals, highlighting the significance and utility of the software product, while also focusing on the target user base.

1.1 Purpose

This SRS (Software Requirements Specification) document is written to provide a detailed overview of the requirements and specifications for the development of a database application for movies, herein referred to as "MovieBase". It shall outline the essential functionalities, performance expectations and UI design, ensuring a shared understanding of the project's goals.

1.2 Document Conventions

This document is created strictly following the IEEE standards.

1.3 Intended Audience and Reading Suggestions

The document is intended to serve several groups of audience members.

1. The SRS will be referenced by the system designers to create the design of the application.
2. The developers and testers shall view it to ensure that the requirements have been met.
3. The admin and users who are the clients for the project shall review the SRS to get an understanding of the basic product functionalities.
4. The application maintenance staff can also review the document for future modifications to gain clarity on the current functionality.

The next section of the document, the Overall Description, gives an overview of the functionality of the product with the help of a high-level description diagram of the product. It describes the informal requirements, which is then used to establish a context for the detailed technical and functional requirements and the use case diagrams. The fifth section lays importance on the non-functional requirements of the product, including performance and safety and security requirements.

1.4 Product Scope

The MovieBase application is designed to address the needs of movie enthusiasts, offering a comprehensive platform for managing and exploring information about movies. Users can browse and contribute to an extensive movie database, creating an engaging and informative experience.

1.5 Definitions and Abbreviations

S. No	Abbreviation	Definition(s)
1	IEEE	Institute of Electrical and Electronics Engineers
2	Admin	Administrator of the database
3	SRS	Software Requirements Specifications
4	MovieBase	Movie Database
5	Database	Stores the details, ratings and reviews of movies and login credentials of users.
6	ISAN	International Standard Audiovisual Number

1.6 References

- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998
- Fundamentals of Database Systems by Ramez Elmasri
- [React Documentation](#).
- [NodeJS Documentation](#).
- [Express Documentation](#).

2 Overall Description

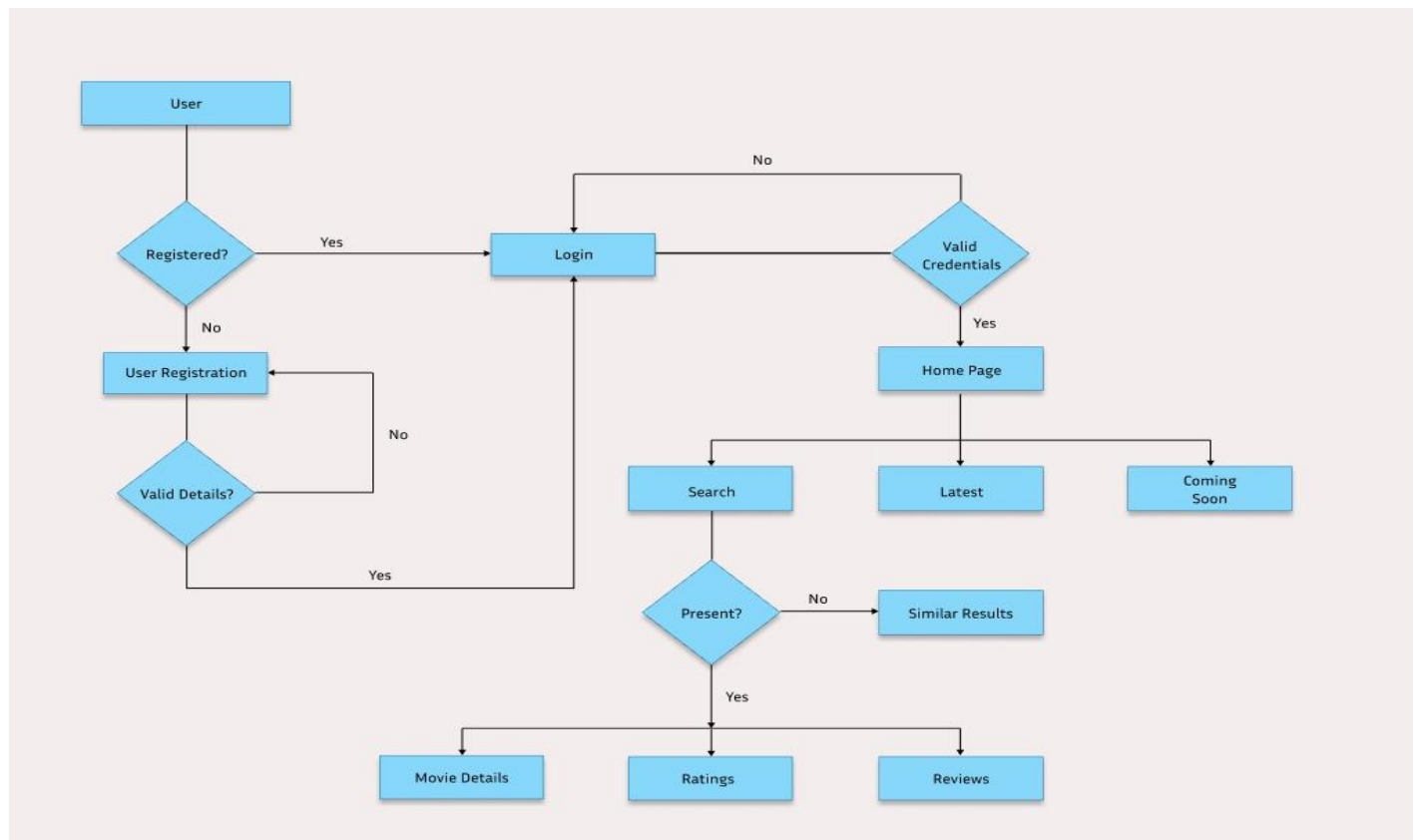
2.1 Product Overview

MovieBase will function as an interactive platform for avid film enthusiasts. It will offer real-time information on movies during searches. Registered users will have the ability to author reviews and provide ratings, while unregistered users can access and read them.

On the administrative front, the system will establish an administrator role with direct access to the database, enabling them to add, delete and update movie details and reviews. To facilitate these tasks, a dedicated login will be provided for administrators.

MovieBase will serve as a user interface for film enthusiasts. Real time information about available movies will be provided during searches. Registered users can write reviews and leave ratings, while unregistered users can read them.

Administratively, the system will designate an administrator role, who will have direct access to the database for adding, deleting and updating movie details. For this purpose, a dedicated login will be provided to the administrators.



2.2 Product Functionality

MovieBase will permit:

- Any user to view movie details and ratings and reviews submitted by other users.
- The registered users to login and access the functionalities.
- The registered users to:
 - . Search for a movie based on title/actor/director/genre.
 - . View details of upcoming movies, recently released movies, highest rated movies and popular movies
 - . View and edit the user's watchlist.
 - . View the user's search history.
 - . Submit ratings and reviews for a particular movie of their choice.
- The admins to:
 - . Login using valid credentials.
 - . Add/Delete/Update the database.
 - . Select movies as editor's choice.
 - . Verify/Unverify critics and proficient reviewers.

2.3 User Classes and Characteristics

Primarily MovieBase is built to support two types of accesses:

- **User Access:** login, search movie details and add reviews and ratings.
- **Admin Access:** add/delete/update movie records, delete inappropriate and spam user reviews, provide/remove verification status to users, display selected content in home screen.

2.4 Design and Implementation Constraints

Database must be updated on adding, modifying or deleting movies, ratings and reviews. User rating can be considered to be the average rating of all users. Official rating is a value determined by the admins.

Access to the website can be made by anyone with minimal internet connectivity as long as the system and OS requirements are met. Login is accepted by entering valid credentials.

2.5 Assumptions and Dependencies

The assumptions are:

- The data of all the movies and registered users must be stored in a database.
- The system will be available at all times.
- The system must have sufficient storage capacity and render fast access to the database.
- The users know English as the interface will be entirely in English.

The dependencies are:

- The hardware and software mentioned
- The admin should have a basic idea about the functioning of the system
- The database must be updated upon addition, modification and deletion of movie details, ratings and reviews.

3 Specific Requirements

3.1 User Interfaces

Upon opening the application, the user is greeted with a login page where they have the option to register or login.

If they choose to register they must enter a valid email, a valid username and a valid password and they are then given access to MovieBase.

If they choose to login, they must enter their username and password, which after being verified will let them access MovieBase.

An interface allowing users to search for movie records and to add reviews and ratings.

An interface allowing admins to manage the database content.

3.2 Hardware Interfaces

Processor: Pentium(R) Dual-core

CPU Hard Disk: 50GB

RAM: 2GB or more

3.3 Software Interfaces

Operating System	Windows version 7 and above. Implementation is focused on windows due to its large user base and user friendliness.
Database and Backend	Node JS, a server side tool based on Javascript (backend) and SQLite, a open-source relational database management system . Together, they are capable of delivering high quality and highly scalable products.

HTML, CSS and JS (React)	We primarily implement the frontend using HTML, CSS, and JavaScript, leveraging the React library. React is renowned for its ability to create exceptional and highly interactive user interfaces.
---------------------------------	--

4 Functional Requirements

4.1 Functional specification

MovieBase shall allow -

F1: Users to register into the website. The details of the users are added to the database on successful registration.

F2: Users and Admins login to the website using their valid credentials. And access the various functionality depending on the access level of the login.

F3: Users and Admins to search for records of Movies with respect to movie name, actors and genre.

F4: Users to add movies to their watchlist and edit the watchlist at any point of time.

F5: Users to see their view history.

F6: Users and admins to add reviews and ratings for movies of their choice. Also allow users to delete/edit their own reviews and allow admins to delete any review.

F7: Admins to add records of movies which include movie name, actors, description and genre.

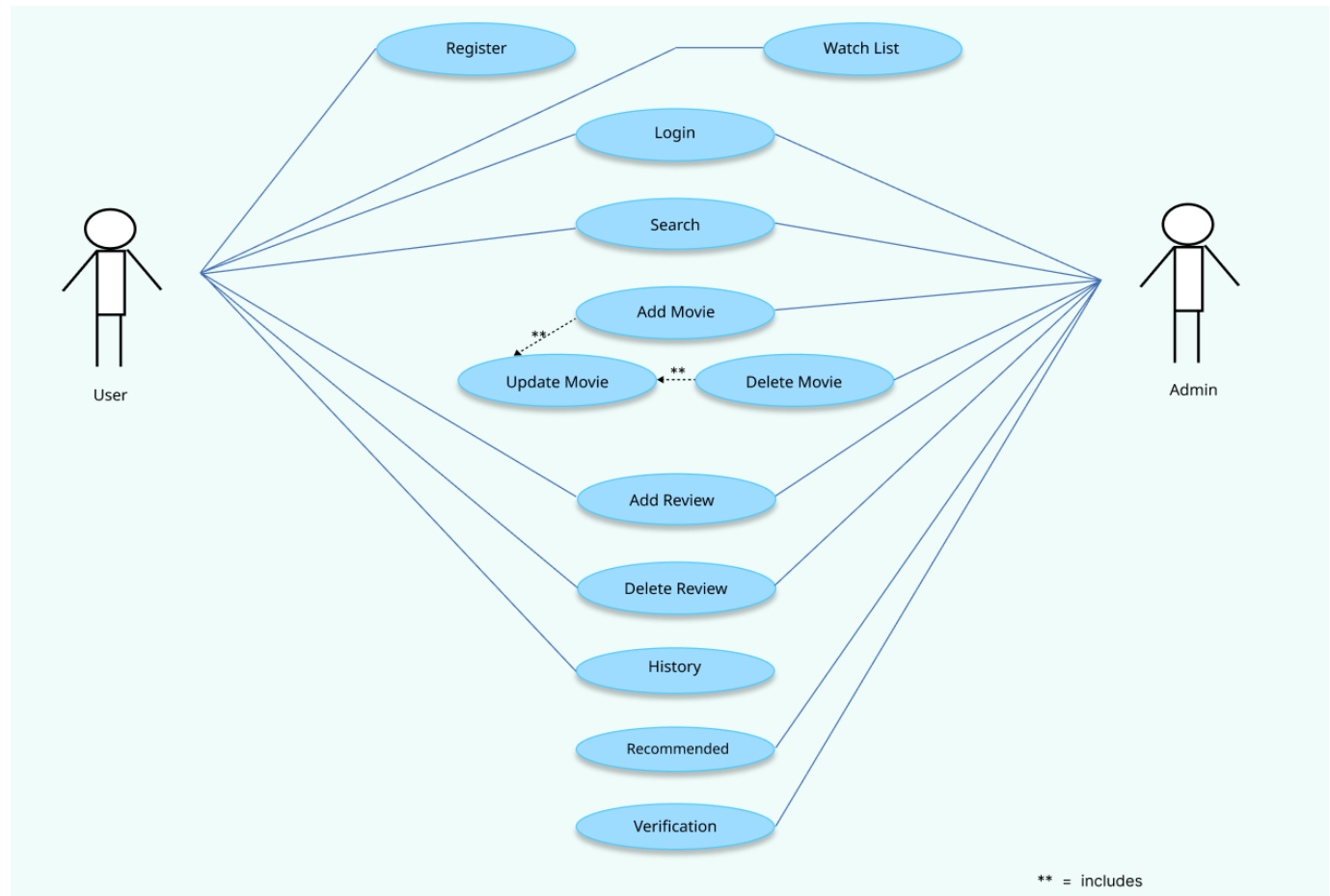
F8: Admins to update and delete details of movie records or a movie record as a whole.

F9: Admins to give and remove special access to users which sets them as verified reviewers, enabling the reviews of these users to be highlighted under the reviews of each movie.

F10: Admins to remove the reviews posted by a user if the review is deemed to be inappropriate or spam.

F11: Admins to add selected movies to the "Editor's choice" list of movies.

4.2 Use case model



4.2.1 Login - U1

Author: Ashwin Suresh Babu

Purpose: To log into the website.

Requirement Traceability: F2

Priority: High

Preconditions:

1. The user/admin must have valid login credentials.

Post conditions:

1. The user/admin successfully logs into the system.

Actors: Users, Admins

Extends: None

Flow of events:

1. Basic Flow:

- The user/admin tries to log into the website with their credentials.
- The user/admin logs in successfully.

2. Alternative Flow:

- User/admin tries to login with the wrong credentials.

- The website shows a message saying “invalid credentials”.
- The website redirects user/admin to the login page.

Includes: None

Notes/Issues: None

4.2.2 Register - U2

Author: Ashwin Suresh Babu

Purpose: To register a new user to the system.

Requirement traceability: F1

Priority: High

Preconditions: The user must not be a registered user.

Post conditions:

1. The user successfully completes the registration.
2. The user login credentials are saved in the system database.

Actors: User.

Extends: None.

Flow of events:

1. Basic Flow

- . A new user registers into the system.
- . The user login credentials are stored in the database.

2. Alternative flow

- . A registered user tries to register again.
- . The system shows a message showing that the userid is already registered.
- . The user is redirected to the login page.

3. Exceptions

- . Submitting invalid user details during registration.

Includes: None.

Notes/Issues: None.

4.2.3 Search - U3

Author: Ashwin Suresh Babu

Purpose: The user/admin can search for a movie on the basis of movie name, actors or genre.

Requirements traceability: F3

Priority: High

Preconditions:

1. The user/admin must be logged into the system.

Post conditions:

1. The movies related to the search will be retrieved.

Actors: User, librarian

Flow of events:

1. Basic flow

- The user/admin must be logged in with valid credentials.
- The user/admin enters the name of the movie/actor or genre.
- MovieBase will show the movies associated with the search query.

2. Alternative flow

- There are no movies with the given title/actor.

- MovieBase will show movies with similar titles.

Includes: None

Notes/issues: None

4.2.4 Add Review - U4

Author: Johann B Simon

Purpose: To add a review under a movie.

Requirements traceability: F6

Priority: High

Preconditions:

1. The user should have valid credentials to log into the system.

Post conditions:

1. The review is added into the database.

Actors: Users, Admins

Flow of events:

1. Basic flow:

- The user logs into the system with valid credentials.
- The user searches for the movie to review.
- The user types and submits a review under the movie.
- The review is added into the database.

Includes: None

Notes/Issues: None

4.2.5 Edit Review - U5

Author: Johann B Simon

Purpose: To allow users/admins to view and edit the reviews posted by them.

Requirement Traceability: F6

Priority: High.

Precondition:

1. The user/admin must have a valid login credential .

Post Condition:

1. A particular review of the user is viewed and updated.
2. The edited reviews are updated in the database.

Actors: Users, Admins.

Flow of events:

1. Basic Flow:

- The user must be logged into the system.
- The review to be edited must belong to the user.
- After editing the review is updated in the database.

2. Alternate Flow:

- Logged in user does not have any valid reviews.
- No review is updated.

Includes: None

Notes/Issues: None

4.2.6 Delete review - U6

Author: Johann B Simon

Purpose: To allow users to delete their reviews and to allow admins to delete user's reviews.

Requirement traceability: F6

Priority: High

Precondition :

1. The user must have a valid login credential.
2. The user must be deleting their own review.
3. The admin may delete any review.

Post conditions:

1. The review will be deleted from the database.

Actors: Users, admins

Flow of events:

1. Basic Flow:

- The user/admin must be logged into the system.
- The user/admin must select the review to be deleted.
- The user/admin must select the delete option for the review.
- The particular review is deleted from the database.

Includes: None

Notes/Issues: None

4.2.7 View History - U7

Author: Abijith R

Purpose: The user can view previously visited movies.

Requirements Traceability: F5

Priority: Low

Preconditions: The user must be logged in with valid login credentials.

Post conditions:

1. The view history of the user will be retrieved.

Actors: Users, Admins.

Extends:

Flow of Events:

1. Basic Flow

- . User clicks the history button.
- . The view history of the user is displayed.

Includes: None.

Notes/Issues: None.

4.2.8 Add Movie - U8

Author: Abishek Nair

Purpose: To add movie records into the database.

Requirements Traceability: F7

Priority: High

Preconditions:

1. The admin must be logged into the system.

Post Conditions:

1. The database is updated after the addition of each movie record.

Actors: Admins

Extends: None

Flow of Events:

1. Basic Flow:

- The admin inputs the movie title, actor names, description and genre of the movie.
- The new movie record is added to the database and the corresponding change is reflected in the website.

2. Alternative Flow:

- The name of the movie trying to be added already exists.
- Display a message saying "Movie already exists, to make changes go to Edit Movie Details".
- Returns to "add movie" page.

Includes: None

Notes/Issues: None

4.2.9 Edit Movie - U9

Author: Abishek Nair

Purpose: To update the details of movie records from the database.

Requirement Traceability: F8

Priority: High

Preconditions:

1. Admin must be logged in.
2. The movie record to be edited must be valid.

Post Conditions:

1. The edited movie record is updated in the database.

Actors: Admins

Extends: None

Flow of events:

1. Basic Flow:

- The admin searches for the movie to be edited.
- The admin updates the fields that require editing.
- The changes are reflected in the database and ultimately in the website.

2. Alternative Flow:

- The movie records to be edited is invalid.
- Display a message saying "Couldn't Identify Record".
- **Redirect** to edit page.

Includes: None.

Notes/Issues: None.

4.2.10 Delete Movie - U10

Author: Abishek Nair

Purpose: To delete a movie record from the database.

Requirement traceability: F8

Priority: High

Preconditions:

1. The admin should have valid credentials to login to the system.
2. The admin must ensure that the movie to be deleted is not required.

Postconditions:

1. The movie must be deleted from the database.

Actors: Admins**Flow of events:****1. Basic flow:**

- The admin searches for the movie that needs to be deleted.
- The admin requests for the deletion of the movie.
- The movie is deleted from the database.

2. Alternative Flow

- The movie records to be deleted is invalid.
- Display a message saying "Couldn't Identify Record".
- Returns to "delete movie" page.

Includes:**Notes/issues:** None

4.2.11 Verification - U11

Author: Lehan Divakaran**Purpose:** To add/remove verification from users so that their reviews get prioritized.**Requirement Traceability:** F9**Priority:** Low**Preconditions:**

1. The admin must have valid credentials to login to the system.
2. The user account must be valid.

Post conditions:

1. The user account gets verified.

Actors: Admin**Extends:** None**Flow of Events:****1. Basic Flow:**

- . The admin searches for a user account to add/remove verification.
- . The admin adds/removes verification from the user.
- . The user account is set as verified/unverified in the database.

Includes: None.**Notes/Issues:** None.

4.2.12 Edit Watchlist - U12

Author: Abijith R**Purpose:** For users to add and edit movies in their watchlist.**Requirement Traceability:** F4**Priority:** Medium**Preconditions:**

1. The user must be logged in to the website.

Post conditions:

1. The user adds and removes movies from their watchlist.
2. The user's watchlist gets updated.

Actors: User

Extends: None

Flow of Events:

1. Basic Flow

- The user searches for a movie.
- The user adds the movie in his watchlist or removes a movie from his watchlist.
- The watchlist of the user gets updated and the changes are made in the database.

Includes: None

Notes/Issues: None.

4.2.13 Recommended - U13

Author: Lehan Divakaran

Purpose: For Admin to select specific movies as editor's choice.

Requirement Traceability: F11

Priority: Low

Preconditions:

1. The admin must have valid credentials to login to the system.
2. The admin must select a valid movie present in the database.

Post conditions:

1. The movie is successfully added to the editor's choice list.

Actors: Admin

Extends: None.

Flow of Events:

1. Basic Flow

- The Admin tries to login with valid login credentials.
- The Admin successfully logs into the system.
- The Admin searches for a movie to add it to the editor's choice list.
- The database is updated and the movie is added to the list.

2. Alternative Flow

- The Admin is unable to add the movie due to authentication issue or invalid movie search.
- No change happens to the database.

Includes: None

Notes/Issue: None

5 Other Non-functional Requirements

5.1 Performance Requirements

The system must respond to a search movie request rather quickly, but may take a while when submitting/loading reviews. Overall, the performance must be fast and accurate. The system must be capable of handling a large amount of users and data.

5.2 Safety and Security Requirements

User login details must be saved securely. Only admins may add/modify movie details. In terms of the safety aspect, the system does not pose a threat to its users. To combat attacks by malware, backing up the database is advised.

5.3 Software Quality Attributes

Reliability-

The system must deliver precise results swiftly.

Maintainability-

Errors must be easy enough to solve by a maintenance developer.

Portability-

The system must be able to deploy on multiple devices.

Cost-effectiveness -

The system must provide excellent value for investment.

Appendix A - Data Dictionary

Database	A database is a structured repository of data, meticulously organized and indexed, facilitating data retrieval, management, and analysis, and serving as a cornerstone for data-driven applications and decision-making processes.
Admin	This group of individuals possesses elevated privilege levels on the website, enabling them to add, edit, and delete movie records in the database through the website's interface. Additionally, they serve as moderators for user comments and have the authority to grant special privileges to specific users. Furthermore, they have the capability to curate and showcase a selection of movies as recommendations for any user accessing the website.
User	Individuals who use MovieBase are granted the ability to access comprehensive movie record details and can actively contribute to the platform by adding or editing reviews and ratings for movies of their preference.
MovieBase	This platform allows registered users to access comprehensive movie information, including descriptions, actors, genres, and release dates. Furthermore, users have the capability to contribute by adding reviews and ratings to enhance the collective movie database.
React	React, a JavaScript library, is instrumental in the construction of single-page applications by enabling the creation of reusable UI components that streamline the development of dynamic and interactive user interfaces.
Node	Node.js, as an open-source server environment, empowers the execution of JavaScript on the server side.

Appendix B - Group Log

Meeting day	Timing
18/10/2023	08:00 PM - 10:00 PM
19/10/2023	07:30 PM - 10:00 PM
20/10/2023	09:00 AM- 11:00 AM
20/10/2023	07:00 PM - 8:30 PM

Name	Contribution
Ashwin S	Function requirements, specification and use case model. Appendix A, Appendix B. Use case #1, #2 and #3.
Abishek N	Specific requirements, user interfaces, hardware and software interfaces. Use case #8, #9 and #10.
Abijith R	Flowchart, Use case diagram, Other non-functional requirements. Use case #7, #12
Johann B	Product overview, product functionality, user classes and characteristics, design and implementation, assumptions and dependencies. Use case #4, #5 and #6.
Lehan D	Document purpose, conventions, intended audience and reading suggestions, product scope, references. Use case #11, #13.