

**COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**COCHIN UNIVERSITY COLLEGE OF ENGINEERING**

**KUTTANADU, KERALA-688504**



**NOVEMBER 2023**

**PROJECT REPORT ON**

**APNA THEATER.com**

Submitted on partial fulfilment of the requirement for the award of the degree in Master  
of Computer Applications from Cochin University of Science and Technology

**Submitted by**

**VIKASH KUMAR DAS**

**38222166**

**ABHAY KUMAR**

**38222104**

**SHAMS TABREJ**

**38222156**

**SANDEEP S**

**38222154**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**2022-2024**

**COCHIN UNIVERSITY COLLEGE OF ENGINEERING  
KUTTANADU**



**CERTIFICATE**

This is to certify that the project report entitled “*Apna Theater.com*” is a Bonafide record on partial fulfilment for the degree of the **MASTER OF COMPUTER APPLICATIONS** to the **COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY** through **DEPARTMENT OF COMPUTER APPLICATIONS, COCHIN UNIVERSITY COLLEGE OF ENGINEERING KUTTANADU, and ALAPPUZHA** done by **VIKASH KUMAR DAS (Reg. NO: 38222166)** in the year 2023.

**Project Guide**

Mrs. RADHIKA B,  
Asst.Professor, MCA  
CUCEK

**Project Coordinator**

Mrs. RADHIKA B,  
Asst.Professor, MCA  
CUCEK

**Head of the Department**

Dr. JOSEPHKUTTY JACOB

**Examiner**

Place:

Date:

## DECLARATION

I hereby declare that the project entitled “*Apna Theater.com*” submitted to the  
DEPARTMENT OF COMPUTER APPLICATIONS, COCHIN UNIVERSITY  
COLLEGE

OF ENGINEERING KUTTANADU in the partial fulfilment of the requirements for the  
award of Degree in MASTER OF COMPUTER APPLICATIONS is a record of  
original work done by me under the guidance of Mrs. RADHIKA B, Assistant  
Professor of MCA department, during my period of study in COCHIN UNIVERSITY  
COLLEGE OF ENGINEERING KUTTANADU.

Place:

Date:

VIKASH KUMAR DAS

ABHAY KUMAR

SHAMS TABREJ

SANDEEP S

## ACKNOWLEDGEMENT

I am thankful to God almighty for the blessings in the successful completion of my minor project “**APNA THEATER.com**”. I would like to record my profound gratitude to **DR. JOSEPHKUTTY JACOB**, Principal and Head of the Department of MCA of COCHIN UNIVERSITY COLLEGE OF ENGINEERING, KUTTANADU who has deeply inspired me.

It's grateful to express my thanks to **Mrs. RADHIKA B**, Assistant Professor in MCA Department of COCHIN UNIVERSITY COLLEGE OF ENGINEERING KUTTANADU, my Project guide, for her effective guidance, constructive criticism and innovative and useful stream of suggestions that helped me to complete my project.

I am also grateful to my Project coordinator, **Mrs. RADHIKA B**, Assistant Professor of MCA department for providing various resources and moral support throughout the project development period. I am thankful to various resources that provide requirements for my projects, because requirements are back bone of every project. I am also thankful to my teachers, friends, family members, for their support and prayer for me to complete my project.

## **ABSTRACT**

The Apna Theater.com is an exciting and user-friendly movie streaming website that aims to provide a comprehensive collection of movies, TV shows, and Web Series across various genres. It offers an alternative to expensive streaming platforms, making entertainment accessible to a wider audience. The platform is designed to offer a seamless and enjoyable experience for movie enthusiasts, allowing them to watch their favorite content at their convenience.

ApnaTheatre.com is a place for all movieholics & cinephiles to enjoy. This movie streaming website allows users to watch their favorite's movies, TV shows, and Web series from multiple genres like Crime, Comedy, Drama, Documentaries, Horror, Anime, Romance, War, Action, Fantasy, Kids and many more!

## CONTENTS

<b>1.</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	About the Project .....	2
1.2	Objective & Scope of Project .....	2
1.3	Definition of Problem.....	3
<b>2.</b>	<b>SYSTEM ANALYSIS.....</b>	<b>4</b>
2.1	Existing System Disadvantages .....	5
2.2	Proposed system Advantages.....	6
2.3	Architecture .....	8
2.4	Module description .....	9
2.5	Feasibility Study .....	10
<b>3.</b>	<b>SYSTEM REQUIREMENTS AND SPECIFICATIONS.....</b>	<b>11</b>
3.1	Hardware Configurations .....	12
3.2	Software Configurations.....	12
3.3	Technology Used .....	13
3.4	Platform Used .....	13
<b>4.</b>	<b>SYSTEM DESIGN.....</b>	<b>14</b>
4.1	Table Design.....	15
4.2	Data Flow Diagram.....	18
4.3	Use Case Diagram.....	21
4.4	Class Diagram.....	22

<b>5.</b>	<b>SYSTEM IMPLEMENTATION, TESTING &amp; MAINTENANCE...23</b>
5.1	System Implementation..... 25
5.2	System Testing (Methodologies used for testing is required according to the project) ..... 26
5.3	Unit Testing..... 27
5.4	Integration Testing ..... 29
5.5	System Testing..... 32
5.6	Test Report ..... 34
5.7	System Maintenance ..... 36
<b>6.</b>	<b>CONCLUSION AND FUTURE ENHANCEMENT.....38</b>
6.1	Conclusion..... 39
6.2	Future Scope ..... 39
<b>7.</b>	<b>BIBLIOGRAPHY ..... 40</b>
<b>8.</b>	<b>SAMPLE CODE .....42</b>
<b>9.</b>	<b>SCREENSHOTS ..... 50</b>

## LIST OF FIGURES

1.	System Architecture.....	8
2.	DFD level 0.....	18
3.	DFD level 1.....	19
4.	DFD level 2.....	20
5.	Use Case Diagram .....	21
6.	Class Diagram.....	22
7.	User Signup Screenshot.....	51
8.	User Login Screenshot.....	51
9.	Home Page Screenshot .....	52
10.	Movies Screenshot .....	53
11.	Genres Screenshot.....	53
12.	Premium Screenshot .....	54
13.	Contact Us Screenshot .....	54



## LIST OF TABLES

1. User registration Details Table..... 14
2. Contact us message Table .....15

# **1. INTRODUCTION**

## **1.1 ABOUT THE PROJECT**

Apna Theatre.com project is to create a user-centric, cost-effective, and diverse movie streaming website. The project aims to provide a platform where movie enthusiasts can enjoy a wide range of content from various genres at an affordable price, making quality entertainment accessible to a broader audience.

Apna Theatre.com project encompasses the development and implementation of a robust online streaming platform. This platform will host a comprehensive collection of movies, TV shows, and web series, covering genres such as Crime, Comedy, Drama, Documentaries, Horror, Anime, Romance, War, Action, Fantasy, Kids, and more. The scope also includes creating an intuitive and user-friendly interface to enhance the overall viewing experience.

## **1.2 OBJECTIVE AND SCOPE OF THE PROJECT**

The primary objective of the Apna Theatre.com project is to establish a user-friendly and cost-effective movie streaming website that caters to the entertainment needs of a diverse audience.

The project aims to achieve the following key objectives:

- Affordability.
- User-Friendly Experience.
- User interface.

- User authentication and authorization.
- Convenience.
- Competitive Edge.

The scope of the Apna Theatre.com project encompasses various aspects related to the development and launch of the movie streaming website. The key elements within the project scope include:

- Website Development.
- Payment Integration.
- User Feedback Mechanism.
- Marketing and Promotion.

### **1.3 DEFINITION OF THE PROBLEM**

The entertainment industry has witnessed a surge in demand for online streaming services, offering a wide array of movies, TV shows, and web series. While major streaming platforms like Netflix, Amazon Prime, and Hotstar dominate the market, there are certain issues and challenges that the Apna Theatre.com project seeks to address:

- Cost Barrier.
- Limited Genre Options.
- User-Friendliness.

By identifying and addressing these issues, the Apna Theatre.com project aims to provide a solution that offers an affordable, diverse, and user-friendly streaming experience, creating a niche in the competitive entertainment landscape.

## **2. SYSTEM ANALYSIS**

## **2.1 EXISTING SYSTEM**

The existing system in the online streaming industry is primarily dominated by well-established platforms such as Netflix, Amazon Prime, and Hotstar. These platforms have set the benchmark for high-quality content and on-demand streaming services. However, several challenges and limitations persist within the existing system:

- Genre Limitations:
  - Issue: The existing platforms tend to focus on mainstream genres, potentially neglecting niche markets and specific content categories.
  - Impact: Viewers with unique preferences, such as those interested in anime, documentaries, or regional content, may find limited options on mainstream platforms.
- Market Dominance:
  - Issue: The streaming industry is highly competitive, with a few major players dominating the market.
  - Impact: New entrants face challenges in establishing a foothold and attracting a user base in the presence of well-established platforms.

## **Disadvantages**

- Competition with Free Platforms.
- Technical Infrastructure.
- Complex User Interface.
- Licensing and Copyright Challenges.

## **2.2 PROPOSED SYSTEM**

The proposed system for Apna Theatre.com aims to overcome the identified challenges and disadvantages of the existing streaming landscape. The key features and improvements are designed to create a unique and competitive platform in the industry.

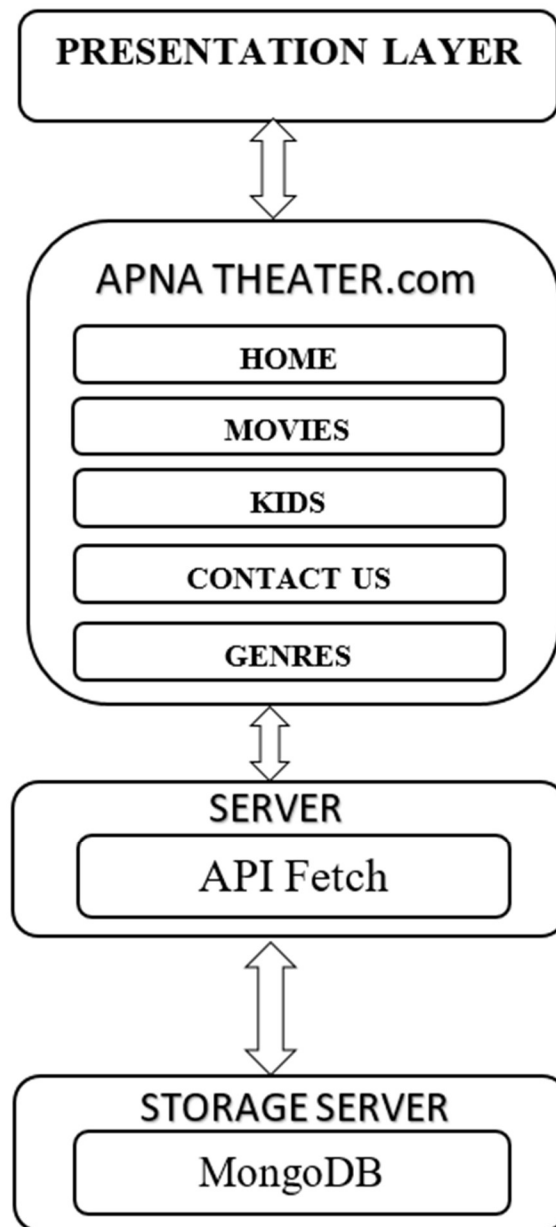
- **Cost-Effective Technical Infrastructure:**
  - **Objective:** Develop and maintain a cost-effective yet robust technical infrastructure for seamless streaming.
  - **Benefits:** Minimizing operational costs while ensuring a reliable and high-quality streaming experience for users.
- **Affordability and Flexible Subscription Models:**
  - **Objective:** Introduce cost-effective subscription plans, ensuring that users with various budgets can access premium content.
  - **Benefits:** Increased accessibility and a larger user base, addressing the affordability concern present in existing platforms.

## **Advantages**

- Affordability.
- Diverse Content Library.
- User-Friendly Interface.
- On-Demand and Flexible Viewing.
- Strategic Marketing and Branding.
- Cost-Effective Technical Infrastructure.
- Community Engagement and Feedback.
- Partnerships and Licensing Strategy.
- Cross-Platform Compatibility.
- Market Differentiation.
- User-Centric Approach.



## 2.3 SYSTEM ARCHITECTURE



## 2.4 MODULE DESCRIPTION

### 1. User Authentication and Authorization Module

- **Description:** This module manages user accounts, authentication, and authorization processes.
- **Functionality:** Handles user registration and login, verifies user credentials, and authorizes access based on subscription status.

### 2. Content Management Module:

- **Description:** The Content Management Module oversees the organization, storage and retrieval of the platform's content library.
- **Functionality:** Manages metadata for movies, TV shows, and web series, categorizes content, and ensures efficient retrieval for streaming.

### 3. Subscription and Payment Module:

- **Description:** This module facilitates user subscriptions and handles payment transactions securely.
- **Functionality:** Integrates with payment gateways, manages subscription plans, & ensures a seamless and secure payment process.

### 4. Feedback and Analytics Module:

- **Description:** Manages user feedback and gathers platform usage data for analysis.
- **Functionality:** Allows users to provide feedback, collects data on user interactions, and generates analytics reports for understanding user behavior and preferences.

### 5. Video Streaming and Playback Module:

- **Description:** Provides administrators with tools to monitor platform performance and manage user accounts.
- **Functionality:** Monitors server health, tracks user activity, and enables administrators to perform necessary maintenance tasks.

## **2.5 FEASIBILITY STUDY**

The feasibility study for Apna Theatre.com assesses the viability of the project from various perspectives, including technical, economic, operational, and scheduling considerations.

### **1. Operational Feasibility: -**

- Assessment: Operational feasibility assesses whether the project can be seamlessly integrated into existing operations and processes.
- Findings: The operational feasibility is favorable. The platform's design ensures ease of use for both users and administrators, and it aligns with industry best practices.

### **2. Technical Feasibility: -**

- Assessment: The technical feasibility evaluates whether the proposed system can be successfully implemented with the available technology and resources.
- Findings: The project is technically feasible. The required technologies for content streaming, user authentication, and database management are well-established and widely used.

### **3. Economic Feasibility: -**

- Assessment: The economic feasibility study examines whether the project is financially viable and justifiable within budget constraints.
- Findings: The economic feasibility is positive. The revenue model, which relies on affordable subscription plans, is designed to cover operational costs and provide a return on investment

### **3.SYSTEM REQUIREMENTS AND SPECIFICATIONS**

### **3.1 HARDWARE CONFIGURATIONS**

The selection of hardware is very important in the existence and proper working of any of the software. When selecting hardware, the size and capacity requirements are also important. This software is able to run with following hardware configuration

Processor - Intel Core x256 architecture

Processor Speed - 3.30GHz to 4.10GH

RAM - 4 GB

Hard Disk - 160 GB Keyboard – 101 keys

### **3.2 SOFTWARE CONFIGURATIONS**

One of the most difficult tasks is selecting software, once the system requirement is found out then we have to determine whether a particular software package fits for those system requirements. This section summarizes the application requirement.

Operating system - Android, Windows, MacOS

Training and Testing - ANDROID PHONE, Windows laptop, MacBook

PLATFORM – VSCODE (editor), MongoDB, NodeJS

Backend (server-side scripting language) – JAVASCRIPT, NodeJS, Fetch API,

RESTful API, MYSQL

Frontend (client-side scripting language) – HTML, BOOTSTRAP, CSS

### **3.3 TECHNOLOGY USED**

- HTML and CSS Programming Language
- Bootstrap CSS framework
- NodeJS for server side scripting
- MongoDB used for storing data

### **3.4 PLATFORM USED**

- VS code for Windows 64 - bit
- MongoDB server for database server
- Windows, Android.

## **4. SYSTEM DESIGN**

## 4.1 TABLE DESIGN

### 1.Users Table:

- Table description:-for Sign-Up and potentially to keep track of user preferences.
- Primary key: User-id
- Foreign key- Nil
- Unique key- email

S.NO	FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	FIELD SIZE
1.	<u>User_id</u>	Varchar	To store User Id	15
2	Name	Varchar	To store Name	30
3.	Email	Varchar	To store Email	255
4	Username	Varchar	To store UserName	30
5.	Password	Varchar	To store encrypted Password	255



**2.Movies Table:**

- Table description:-for Genres and possibly the Home Page.
- Primary key: movie\_id
- Foreign key- Nil
- Unique key- Nil

S.NO	FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	FIELD SIZE
1	<u>movie_id</u>	Varchar	Unique identifier for each movie	15
2	title	Varchar	Title of the movie	50
3	Genre	Varchar	Genre of the movie	5
4	Release_date	Varchar	DATE	10
5	Director	Varchar	Director of the movie	50

Table: 4. Created Movie Table

**3.Genres Table:**

S.NO	FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	FIELD SIZE
1	<u>genre_id</u>	INT	Unique identifier for each genre	15
2	name	Varchar	Name of the genre	50

Table: 4.3 Created Genre Table

**4.User Preferences Table:**

S.N O	FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	FIELD SIZE
1	<u>preference_id</u>	Var	Unique identifier for each user preference	15
2	user_id	INT	References the user who has the preference	10
3	genre_id	INT	References the genre of the preference	10

Table: 4.4 Created User Preferences Table

## 4.2 DATA FLOW DIAGRAM

### LEVEL 0



Fig: 4.2 DFD Level 0

### LEVEL 1

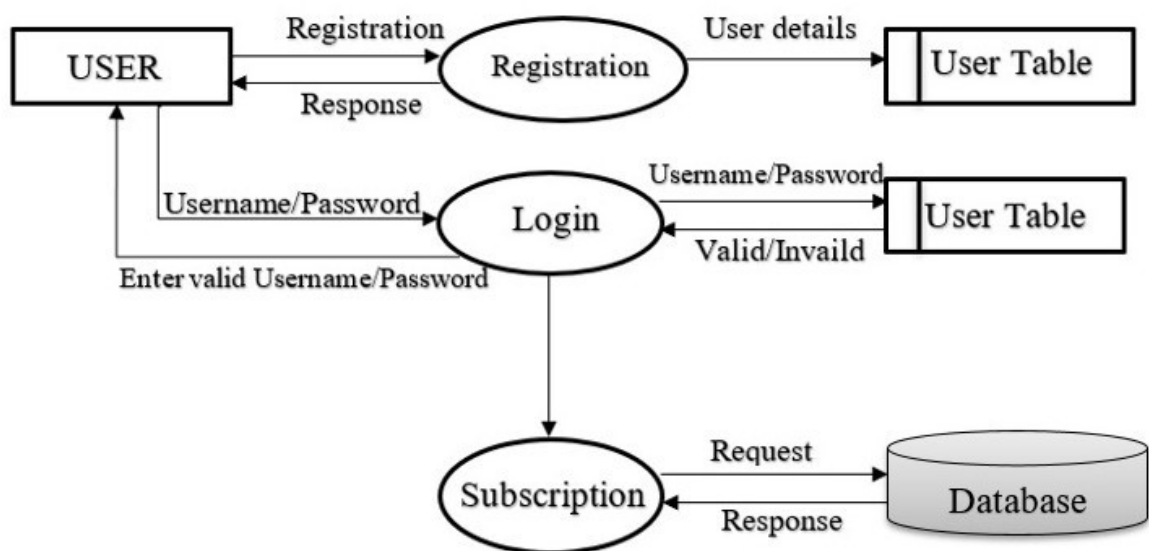
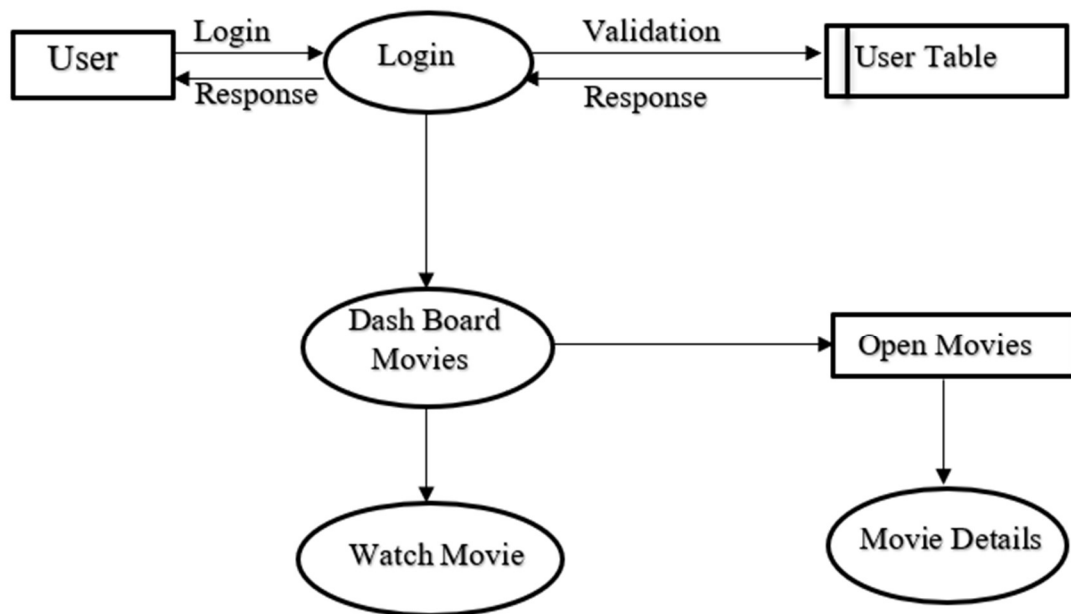


Fig: 4.2 DFD Level 1

**LEVEL 1 USER****Fig: 4.2 DFD Level 1**

## LEVEL 1 ADMIN

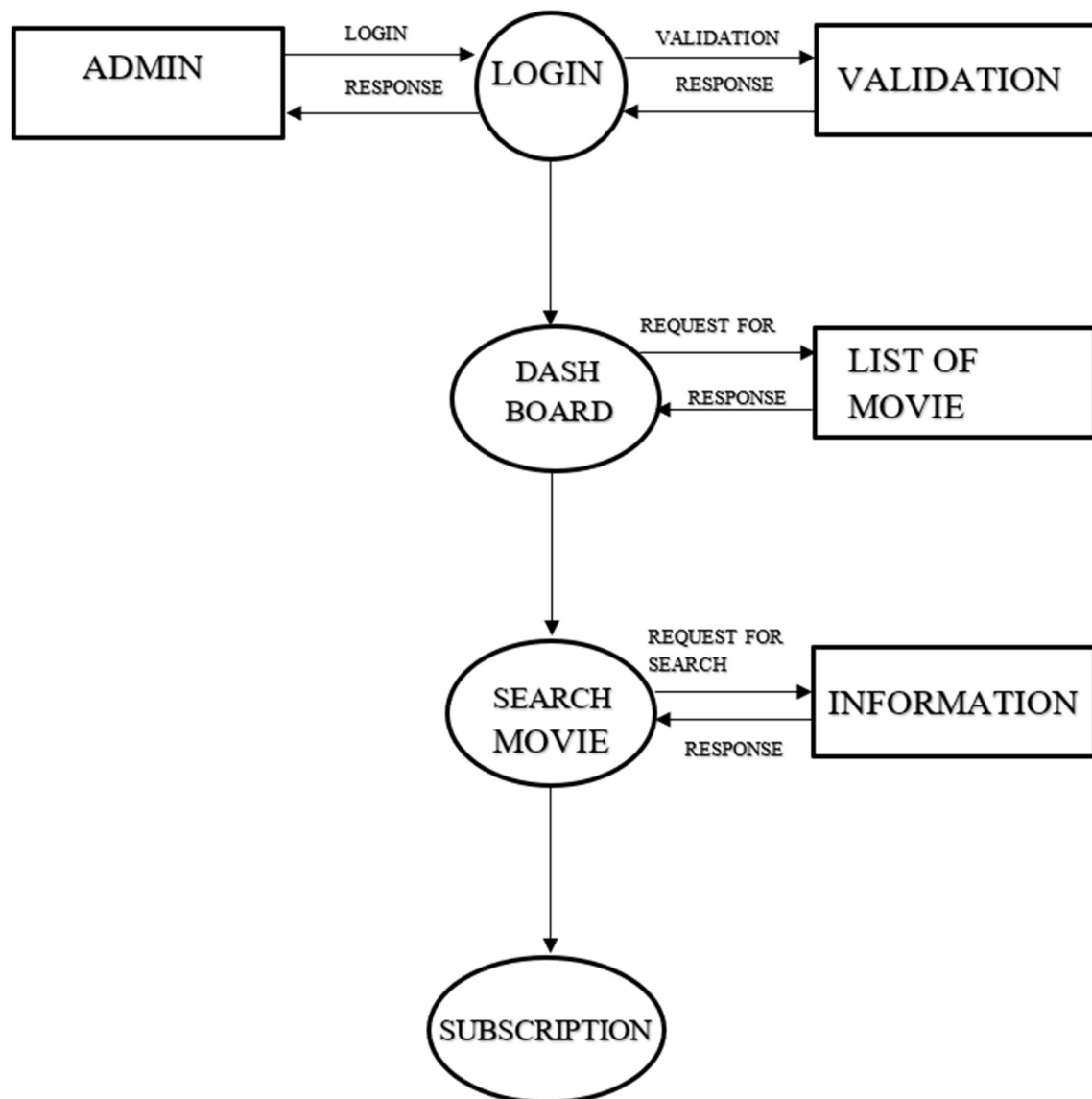


Fig: 4.2 DFD Level 2

### 4.3 USE CASE DIAGRAM

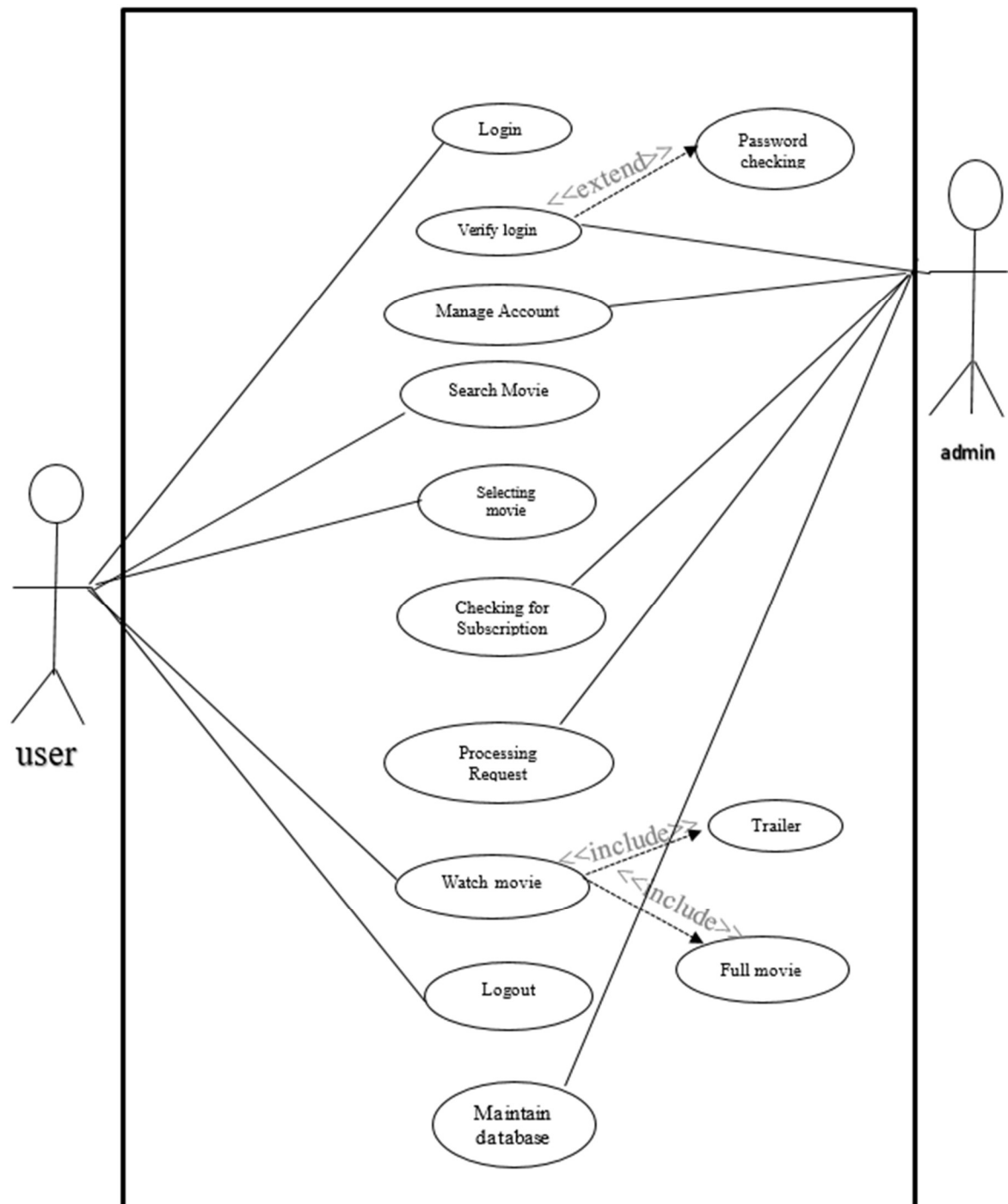


Fig: 4.5 Use Case Diagram

#### 4.4 CLASS DIAGRAM

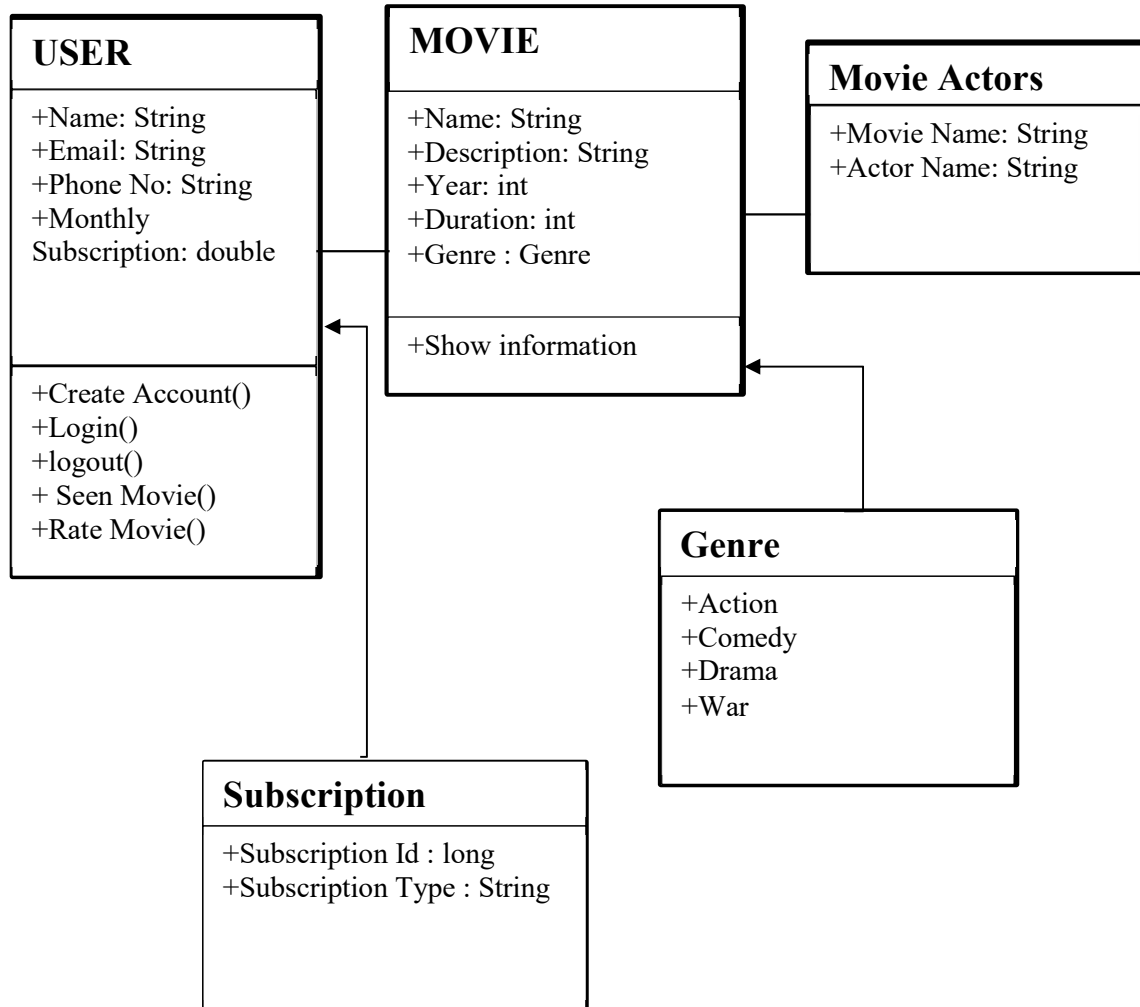


Fig: 4.7 CLASS Diagram

## **5.SYSTEM IMPLEMENTATION, TESTING & MAINTENANCE**



## **5.1 SYSTEM IMPLEMENTATION**

The implementation, testing, and maintenance phases are crucial steps in the development lifecycle of Apna Theatre.com. These phases ensure that the system is developed according to the design specifications, thoroughly tested for quality, and maintained to meet evolving requirements and address issues over time.

- **Frontend Development:**

- Implement the user interfaces based on the design specifications.
- Utilize the chosen frontend framework (e.g., React, Angular, or Vue.js) to create responsive and interactive views.

- **Backend Development:**

- Develop the backend using Node.js.
- Implement RESTful API endpoints for communication with the frontend.
- Set up routes, controllers, and middleware as per the design.

- **Database Implementation:**

- Create the database schema based on the table design.
- Implement database queries and interactions in the backend code.

- **External Service Integration:**

- Integrate with external APIs for payment gateways, content providers, and other services.
- Ensure secure and seamless communication with external services.

## **5.2 SYSTEM TESTING**

System testing is a comprehensive phase in the software development lifecycle where the entire system is evaluated to ensure that it functions as intended and meets the specified requirements. In the case of Apna Theatre.com, system testing involves verifying the integrated components, checking functionality, assessing performance, and ensuring security. Here's a breakdown of the key aspects of system testing:

- **Functional testing:**

- API Testing: Validate the functionality of API endpoints, ensuring they respond correctly to various requests.
- API Testing: Validate the functionality of API endpoints, ensuring they respond correctly to various requests.

- **User Acceptance Testing (UAT):**

- Real User Testing: Collaborate with actual users to conduct UAT and gather feedback.
- Usability Testing: Evaluate the overall user experience and interface intuitiveness.

- **Performance testing:**

- Load Testing: Evaluate the system's performance under expected load conditions.
- Stress Testing: Assess the system's behavior under extreme conditions to identify breaking points.
- Scalability Testing: Test the system's ability to scale horizontally or vertically.

- **Security testing:**

- Authentication and Authorization Testing: Ensure secure user authentication and proper authorization.
- Data Security Testing: Verify the security of data storage, transmission, and access controls.

- **Database Testing:**

- Data Integrity Testing: Confirm that data stored in the database is accurate and follows integrity constraints.
- Database Performance Testing: Evaluate the performance of database queries and transactions.

### **5.3 UNIT TESTING**

Unit testing is a fundamental phase in the software development process where individual units or components of a system are tested in isolation to ensure they function as designed. In the context of Apna Theatre.com, which is a movie streaming platform, unit testing would involve testing small, independent parts of the codebase. Here are key aspects of unit testing:

#### **1. Purpose of Unit Testing:**

- **Isolation:** Test each unit (function, method, or module) in isolation to identify and fix defects early in the development process.
- **Verification:** Verify that individual units perform as expected according to the specifications and requirements.
- **Debugging Aid:** Facilitate the identification and resolution of issues in a specific unit, making debugging more manageable.

#### **2. Unit Testing Process:**

- **Test Planning:** Define a unit testing strategy and plan, including which units will be tested, what test cases will be used, and the expected outcomes.
- **Automated Testing:** Leverage automated testing frameworks (e.g., Jest, Mocha, JUnit) to streamline the execution of unit tests.
- **Test Cases:** Create test cases for each unit, covering various scenarios, input ranges, and edge cases.

### **3. Focus Areas for Unit Testing in Apna Theatre.com:**

- **Backend Components:** Test individual backend components, such as API endpoints, database interactions, and business logic.
- **Frontend Components:** Test frontend components, including user interface elements, event handling, and data rendering.
- **Data Access Layer:** Verify that data access layer components (e.g., database queries) operate correctly.
- **Business Logic:** Test business logic units to ensure they produce the correct outputs for different inputs.
- **Integration Points:** If a unit interacts with external services or APIs, use mocks or stubs to isolate the unit from external dependencies.

## **5.4 INTEGRATION TESTING**

Integration testing is a crucial phase in the software development life cycle where individual components or modules are combined and tested as a group to ensure they work together as intended. In the context of Apna Theatre.com, which is a movie streaming platform, integration testing involves validating the interaction and collaboration between various subsystems. Here are key aspects of integration testing:

### **1. Purpose of Integration Testing:**

- **Verify Interactions:** Ensure that different components, modules, or services interact correctly with each other.
- **Detect Interface Issues:** Identify issues related to data exchange, communication protocols, and dependencies between integrated units.
- **Data Flow:** Validate the flow of data between different subsystems, including inputs, processing, and outputs.

### **2. Types of Integration Testing:**

- **Top-Down Integration Testing:** Start testing from the top level (user interface or outermost layer) and gradually integrate and test lower-level modules.
- **Bottom-Up Integration Testing:** Begin testing from the lower-level modules, gradually combining and testing higher-level modules.
- **Big Bang Integration Testing:** Combine all components simultaneously and test the system as a whole.

- **Incremental Integration Testing:** Integrate and test small units incrementally until the entire system is tested.

#### 4. Integration Testing Process:

- **Integration Test Plan:** Develop a comprehensive plan outlining the integration testing strategy, scope, and schedule.
- **Component Selection:** Choose the components or modules to be integrated based on the integration testing strategy.
- **Test Cases:** Create test cases that cover the interactions between integrated components, considering various scenarios and edge cases.
- **Integration Environment:** Set up a dedicated integration testing environment that mirrors the production environment as closely as possible.
- **Automated Testing:** Leverage automated testing tools to streamline the execution of integration tests, especially in complex systems.
- **Continuous Integration/Continuous Deployment (CI/CD):** Integrate integration tests into the CI/CD pipeline to ensure regular testing with code changes.

#### 5. Focus Areas for Integration Testing in Apna Theatre.com:

- **Backend-Frontend Integration:** Validate the interaction between the backend (server-side) and frontend (client-side) components.

- **Database Integration:** Test how the application interacts with the database, ensuring proper data storage and retrieval.
- **External Service Integration:** Validate the integration with external services, such as payment gateways, content delivery networks, and third-party APIs.
- **API Integration:** Ensure that different API endpoints work together seamlessly, providing the necessary data for various functionalities.
- **User Authentication and Authorization:** Test the integration of user authentication and authorization mechanisms across different components.
- **Error Handling and Logging:** Verify how errors are handled and logged when different components encounter issues during integration.

#### 6. Best Practices for Integration Testing:

- **Comprehensive Test Cases:** Develop comprehensive test cases that cover various integration scenarios.
- **Environment Isolation:** Isolate the integration testing environment from the production environment to prevent unintended impacts.
- **Test Data Management:** Use realistic and varied test data to simulate different scenarios and edge cases.
- **Dependency Mocking:** If certain components or services are not available during testing, use mocks or stubs to simulate their behavior.
- **Regression Testing:** Continuously perform regression testing to ensure that new integrations do not break existing functionality.



## **5.5 SYSTEM TESTING**

System testing is a comprehensive phase in the software development lifecycle where the entire system is tested as a whole to ensure that it meets specified requirements and functions as intended. In the context of Apna Theatre.com, a movie streaming platform, system testing involves validating the complete application, including its frontend, backend, and interactions with external services. Here are key aspects of system testing:

### **1. Purpose of System Testing:**

- **End-to-End Verification:** Verify that the entire system, including all integrated components, functions correctly from end to end.
- **Requirement Compliance:** Ensure that the system meets all specified requirements, including functional and non-functional aspects.
- **User Acceptance:** Assess the system from a user's perspective to ensure it aligns with user expectations.
- **Stability and Reliability:** Confirm the stability and reliability of the entire application under different scenarios and conditions.

### **2. Types of System Testing:**

- **Functional Testing:** Validate that all functional requirements are met, covering various user scenarios.
- **Performance Testing:** Assess the system's performance, including load testing, stress testing, and scalability testing.
- **Security Testing:** Evaluate the security aspects of the system, including authentication, authorization, and data security.

- **Usability Testing:** Assess the user interface and overall user experience to ensure usability and intuitiveness.
- **Compatibility Testing:** Verify that the application works correctly on different browsers, devices, and operating systems.
- **Regression Testing:** Confirm that new changes or features do not negatively impact existing functionality.

### 3. Focus Areas for System Testing in Apna Theatre.com:

- **User Interface and User Experience:** Validate the overall user interface, navigation, and user experience
- **Functionality and Features:** Confirm that all features, such as movie streaming, searching, and user profiles, work as intended.
- **Performance:** Assess the system's performance under various loads and usage patterns.
- **Security:** Verify the security measures in place, including secure authentication, authorization, and data protection.
- **Compatibility:** Test the application's compatibility with different browsers, devices, and operating systems.
- **User Workflows:** Validate end-to-end user workflows, including account creation, content browsing, and streaming

## **5.6 TEST REPORT**

A test report is a comprehensive document that summarizes the results, findings, and outcomes of the testing process. It serves as a crucial communication tool between different stakeholders, providing insights into the quality and readiness of the software being tested. In the context of Apna Theatre.com, a movie streaming platform, the test report should capture information from unit testing, integration testing, system testing, and any other relevant testing phases. Here's an outline of what a test report for Apna Theatre.com might include:

### **1. Introduction:**

- **Project Overview:** Briefly introduce the Apna Theatre.com project and its objectives.
- **Testing Objectives:** Outline the specific objectives of the testing phase, such as ensuring functionality, performance, and security.
- **Scope:** Define the scope of testing, including the components, modules, and functionalities covered.

### **2. Test Environment:**

- **Testing Tools:** Specify the testing tools and frameworks used, such as testing automation tools, performance testing tools, etc.
- **Testing Infrastructure:** Describe the testing environment, including hardware, software, and network configurations.

### **3. Performance Summary:**

- **Performance Metrics:** Report key performance metrics such as response times, throughput, and resource utilization.

- **Scalability:** Assess the scalability of the system by reporting its performance under different loads.
- **Stress Testing Results:** Summarize the performance under stress conditions and identify any bottlenecks.

#### 4. Conclusion:

- **Overall Assessment:** Provide an overall assessment of the software's readiness for release.
- **Readiness for Production:** Indicate whether the software is ready for production deployment or if further testing or refinements are needed.

#### 5. Appendices:

- **Detailed Test Results:** Include detailed test results, including test logs, screenshots, and any additional documentation.
- **Defect Details:** Provide a detailed breakdown of each identified defect, including steps to reproduce, expected behavior, and actual behavior.

#### 5. Sign-off:

- **Testing Team Sign-off:** Document the sign-off from the testing team, indicating their approval of the software's readiness.
- **Stakeholder Sign-off:** Obtain sign-off from relevant stakeholders, confirming their acceptance of the testing outcomes.

## **5.7 SYSTEM MAINTENANCE**

System maintenance is a crucial aspect of the software development life cycle that involves ongoing activities to support, enhance, and optimize the Apna Theatre.com movie streaming platform after its deployment. Maintenance ensures that the system remains reliable, secure, and up-to-date with evolving user needs and technological advancements. Here are key aspects of system maintenance for Apna Theatre.com:

### **1. Bug Fixes and Issue Resolution:**

- **Identify and Prioritize:** Continuously monitor and identify any bugs or issues reported by users or discovered during testing.
- **Prioritize Fixes:** Prioritize bug fixes based on severity, impact on users, and business priorities.
- **Issue Resolution:** Implement timely resolutions for identified issues to maintain a high-quality user experience.

### **2. Security Updates:**

- **Monitor Threats:** Stay informed about the latest security threats and vulnerabilities in the software ecosystem.
- **Security Patches:** Apply security patches promptly to address known vulnerabilities and ensure data protection.
- **Regular Audits:** Conduct regular security audits to proactively identify and address potential security risks.

### **3. Performance Optimization:**

- **Performance Monitoring:** Continuously monitor system performance to identify and address any performance bottlenecks.
- **Caching Strategies:** Implement caching mechanisms to optimize response times and reduce server load.

- **Database Optimization:** Optimize database queries, indexes, and data structures for improved performance.

#### 4. Documentation Updates:

- **System Documentation:** Keep system documentation up-to-date, including architecture diagrams, API documentation, and user guides.
- **Change Logs:** Maintain detailed change logs to track modifications, bug fixes, and enhancements made during maintenance.
- **Knowledge Transfer:** Ensure that knowledge about the system is transferred effectively within the development team.

#### 5. Monitoring and logging:

- **Real-time Monitoring:** Implement real-time monitoring tools to detect and respond to issues promptly.

#### 6. Future Roadmap:

- **Feature Roadmap:** Plan a roadmap for future features and improvements based on user feedback, industry trends, and business goals.
- **Technology Upgrades:** Stay informed about advancements in technology and plan for upgrades that align with the platform's long-term vision.
- **Adaptation to Industry Trends:** Adapt the platform to align with evolving trends in the movie streaming industry, such as new content formats or distribution models.

## **6. CONCLUSION**

## 6.1 CONCLUSION

In conclusion, the development and implementation of Apna Theatre.com represent a significant stride in providing a user-friendly and accessible movie streaming platform. This comprehensive project aims to offer an alternative to expensive streaming platforms, making entertainment available to a broader audience. Let's recap the key aspects and takeaways from the Apna Theatre project.com:

### 6.2. Project Overview:

- **Objective:** The primary goal of Apna Theatre.com is to provide a seamless and enjoyable experience for movie enthusiasts, offering a diverse collection of movies, TV shows, and web series across various genres.
- **Accessibility:** Positioned as an alternative to costly streaming platforms, Apna Theatre.com strives to make entertainment accessible to a wider audience.

## 6.2 FUTURE SCOPE

The future scope of Apna Theatre.com holds exciting possibilities for further development, expansion, and adaptation to emerging trends in the entertainment and streaming industry. Here are some key areas of future scope for Apna Theatre.com:

- Affordability and Pricing Models.
- Partnerships with Devices and Platforms.
- Mobile App Optimization.
- Social Media Marketing and Branding.
- Localized Content and Global Expansion.
- Increasing the Size and Participant of the application.



## **7. BIBLIOGRAPHY**

- [1] <https://www.mongodb.com/>
- [2] <https://getbootstrap.com/>
- [3] <https://www.w3schools.com/html/>
- [4] <https://www.w3schools.com/jquery/>
- [5] <https://fonts.google.com/>
- [6] <https://icons.getbootstrap.com/>
- [7] <https://getbootstrap.com/docs/4.0/components/carousel/>
- [8] <https://rapidapi.com/gox-ai-gox-ai-default/api/ott-details/>
- [9] <https://getbootstrap.com/docs/4.0/components/card/>

## **8. SAMPLE CODE**

index.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,initial-scale=1.0">
  <title>Welcome To Apna Theater.com</title>
  <link rel="shortcut icon" href="./Images/Logo/Title.jpeg"
type="image/x-icon">
  <script src="https://cdnjs.cloudflare.com
/ajax/libs/jquery/3.6.0/jquery.min.js" integrity=
"sha512ZOGReFYm4dnWc1Qt5NtvYSaNcOP+u1T9qYdvdihz0PPSiiqn/+WGUrMQ=="
ssorigin="anonymous" referrerpolicy="no-referrer">
</script>
  <script src="./landing-page.js"></script>
  <!-- Font -->
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Ubuntu&display=swap"
rel="stylesheet">

  <link rel="stylesheet" href="landpage.css">

</head>

<body id="stop-scrolling">
  <div class="preloader"></div>

  <div class="text-container">
    <div class="container-2">
      <a class="navbar-brand" href="home.html"></a>
      <div class="intro-container">
        <h1>WELCOME TO<br><span>Apna Theater.com</span></h1>
        <p>Stream your favourite movies online. </p>
      </div>
      <a href="#" class="button" onclick="myFunction()">Register</a>
    </div>
    <div class="bottom-text-container1">
    <div class="bottom-text-container">
      <p class="watch">Watch</p>
      <p class="chill">Chill</p>
      <p class="repeat">Repeat</p>
    </div>
  </div>
  </div>

  <div class="images-container">
    <marquee behavior="alternate" direction="left" scrollamount="8.4">
// marquee - it defines a scrolling area in the HTML document that moves
across the page any direction.
    <div class="group1">
      <div class="image">
        
      </div>
      <div class="image">

```

```

 // The alt attribute specifies an
alternate text for an area, if the image cannot be displayed.
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
<div class="image">
    
</div>
</div>
</marquee>

```

```

<marquee behavior="alternate" direction="right" scrollamount="8.7"
scrollldelay="77">
  <div class="group2">
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
    <div class="images2">
      
    </div>
  </div>

```

```

<div class="images2">
  
</div>
<div class="images2">
  
</div>

<div class="images2">
  
</div>

<div class="images2">
  
</div>
<div class="images2">
  
</div>
</div>
</marquee>
<marquee behavior="alternate" direction="left" scrollamount="9" >
  <div class="group3">
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
    <div class="images3">
      
    </div>
  </div>

```

```

<div class="images3">
    
</div>
<div class="images3">
    
</div>
<div class="images3">
    
</div>
<div class="images3">
    
</div>
<div class="images3">
    
</div>
<div class="images3">
    
</div>
<div class="images3">
    
</div>
</marquee>
</div>

<script>
    function myFunction() {
        location.replace('login.html')
    }
</script>

<script>
    function preloaderFadeOutInit(){
        $('.preloader').fadeOut('slow');// will first fade out the
loading animation
        $('body').attr('id','');// will fade out the whole DIV that
website.
    }
    // Window load function
    jQuery(window).on('load', function () { // makes sure the whole site is
loaded
        (function ($) {
            preloaderFadeOutInit();
        })(jQuery);
    });
</script>

</body>
</html>

```



index.js

```

const express = require('express');
const mongoose = require('mongoose');
const cors=require("cors");
const bodyParse=require('body-parser');

const app = express();
app.use(cors());
app.use(bodyParse.json());
async function main() {
  try {
    await
mongoose.connect('mongodb://127.0.0.1:27017/registrationForm');mongoose:
    console.log("Database connected");
  } catch (error) {
    console.error("Error connecting to the database:", error);
  }
}

main().catch((err) => console.log(err));

const UserSchema = new mongoose.Schema({
  Name: String,
  Email: String,
  Password: String
});

const User = mongoose.model("User", UserSchema);

app.use(express.json());

// Server-side code
app.post('/api/register', async (req, res) => {
  try {
    let user = new User();
    user.Name = req.body.name;
    user.Email = req.body.email;
    user.Password = req.body.password;
    await user.save();
    res.json("success");
  } catch (error) {
    console.error(error);
    res.status(500).json({ message: 'Internal Server Error' });
  }
});

app.listen(8082, () => {
  console.log("Server running");
});

app.use(express.json());

```

node.js

```
const axios = require('axios');

const options = {
  method: 'GET',
  url: 'https://movie-database-imdb.p.rapidapi.com/movie/',
  params: {name: 'troy'},
  headers: {
    'X-RapidAPI-Key':
'468d3c1b73msh30231e95448655ep19380ejsn32f7b12e3d62',
    'X-RapidAPI-Host': 'movie-database-imdb.p.rapidapi.com'
  }
};

try {
  const response = await axios.request(options);
  console.log(response.data);
} catch (error) {
  console.error(error);
}
```

## **9. SCREENSHOTS**

**Web Series**

**Sign Up**

Enter your name

Enter your Email

Enter your password

**Register**

Need help?

Already a Member? Sign in here.

This page is protected by Google reCAPTCHA to ensure you're not a bot. [Learn more.](#)

Mirzapur IMDb - 8.4

Sacred Games IMDb - 8.7

Game of Thrones IMDb - 9.3

Kota Factory IMDb - 9.0

Inside Edge IMDb - 8.9

Pitchers IMDb - 8.1

Choose Who You Become IMDb - 7.4

Selection Day IMDb - 7.4

Scam 1993: The Big Deal IMDb - 7.3

Agne - Welcome to your Dark Side IMDb - 8.4

Panchayat IMDb - 8.7

Ashtam IMDb - 7.4

**Web Series**

**Sign In**

Email

Password

**Sign In**

Need help?

New to Moviezz Web? Sign up now.

This page is protected by Google reCAPTCHA to ensure you're not a bot. [Learn more.](#)

Mirzapur IMDb - 8.4

Sacred Games IMDb - 8.7

Game of Thrones IMDb - 9.3

Kota Factory IMDb - 9.0

Inside Edge IMDb - 8.9

Pitchers IMDb - 8.1

Choose Who You Become IMDb - 7.4

Selection Day IMDb - 7.4

Scam 1993: The Big Deal IMDb - 7.3

Agne - Welcome to your Dark Side IMDb - 8.4

Panchayat IMDb - 8.7

Ashtam IMDb - 7.4

localhost:27017

Documents registrationForm...

**registrationForm.users**

4 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: ( field: 'value' ) or [Generate query](#)

**ADD DATA** **EXPORT DATA**


1 - 4 of 4

<pre> _id: ObjectId('653a201f5f51e188id9b6c75') Name: "Vikash Kumar Das" Email: "vikashdas64@gmail.com" Password: "Vikash1011@" __v: 0 </pre>
<pre> _id: ObjectId('657569c9478aa54a71f966bb') Name: "Bikesh Patel" Email: "bikeshatilapura@gmail.com" Password: "BikeshBoka1011" __v: 0 </pre>
<pre> _id: ObjectId('6575a53a478aa54a71f966bd') Name: "SANDEEP " Email: "sandeep@gmail.com" Password: "Sandeep1011@" __v: 0 </pre>

[Home](#)
[Movies](#)
[Web Series](#)
[Kids](#)
[TV](#)
[Premium](#)
[Contact Us](#)
[Genres](#)
[Logout](#)

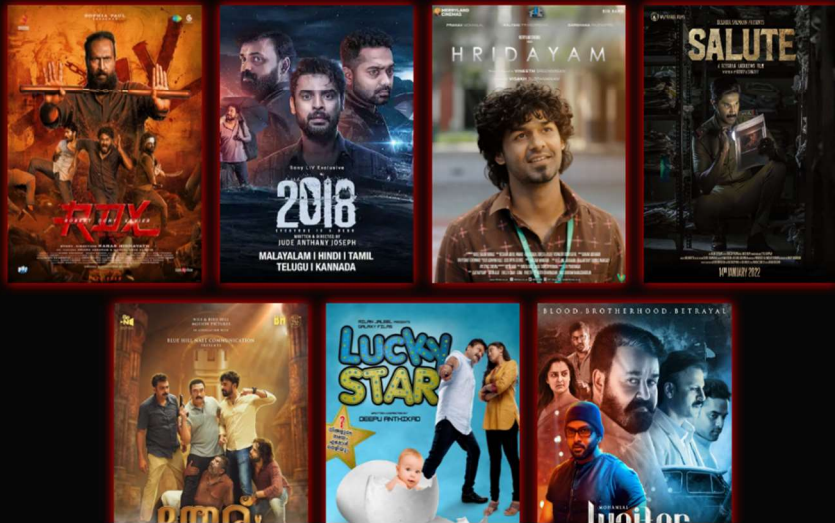
Welcome To Apna Theater.com®

Stream Latest Movies Now!

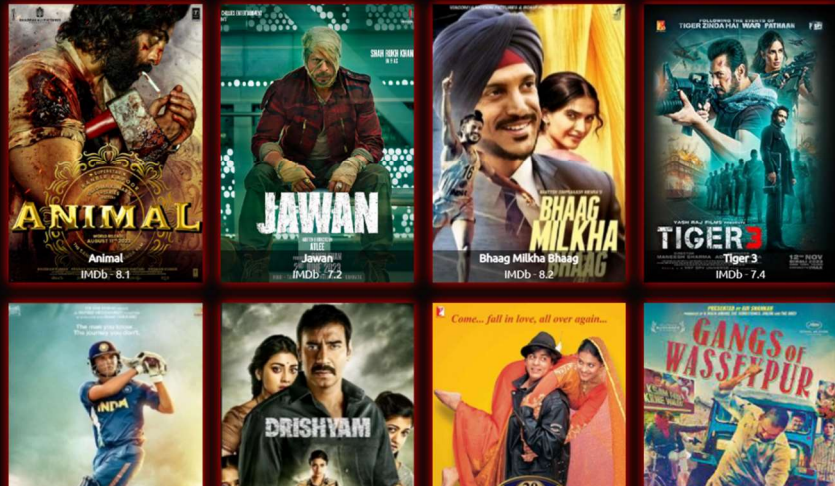


**Movies**

**Mollywood**



**Bollywood**







► Watch all you want. Advert-free.  
► Recommendations just for you.  
► Change or cancel your plan anytime.

### Basic

₹149 per month

- Video Quality: Good
- Resolution: 480p
- Devices you can use to watch:
  - Mobile
  - Tablet
  - TV
  - Computer

[Get Started](#)

### Premium

₹209 per month

- Video Quality: Best
- Resolution: 4K+HDR
- Devices you can use to watch:
  - Mobile
  - Tablet
  - TV
  - Computer

[Get Started](#)

### Standard

₹399 per month

- Video Quality: Better
- Resolution: 1080p
- Devices you can use to watch:
  - Mobile
  - Tablet
  - TV
  - Computer

[Get Started](#)

## Contact Us

Have Questions? We have answers ( may be )

### Get in touch

First Name	Last Name
Email	Phone Number
Subject	
Message	

[SEND MESSAGE](#) [Reset](#)

### Our Address

LRS  
Pulinkunnoo  
Alappuzha,  
Kerala - 688504

vikashdas649@gmail.com  
+91-84058 74061

### Follow Us

[f](#) [t](#) [in](#) [ig](#) [yt](#)

Watch! Chill! Repeat!

### Services

Enjoy Latest Movies  
Watch Web-Series  
Everything for Kids  
Coming soon  
Get Premium Subscription  
FAQ

### Try Our App

[Download on the App Store](#) [GET IT ON Google Play](#)

### Contact Us

Feel free to contact us.

[Contact](#)