HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY OF INTERNATIONAL EDUCATION

CAPSTONE PROJECT

CREATING AN ONLINE MOVIE STREAMING WEB APPLICATION

Students:

Chu Nguyễn Hoàng Sơn 19119128

Nguyễn Quốc Hoàng 19110128

Instructor: PhD. Mai Anh Thơ

Ho Chi Minh City, December 2024

HOCHIMINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION
FACULTY OF INTERNATIONAL EDUCATION
fie.hcmute.edu.vn

SOCIALIST REPUBLIC OF VIETNAM

Independence - Freedom – Happiness

****

Ho Chi Minh City, December, 2024

CAPSTONE PROJECT OBJECTIVES

Students' names:

1. Chu Nguyễn Hoàng Sơn ID: 19119128 Class: 19110CLA2
2. Nguyễn Quốc Hoàng ID: 19110128 Class: 19110CLA1

Major: **Information Technology**

Instructor: **PhD. Mai Anh Thơ**

Date of assignment: 02/09/2024 Date of submission: 12/12/2024

1- Project title: **Creating an online movie streaming web application**

2- Initial materials provided by supervisor: **None**

3- Content of the project:

Theory:

- Nextjs
- Nodejs
- Expressjs
- PostgreSQL
- Cloudinary

Practice:

Creating an online movie streaming web application with a recommender system are the main features:

- Enable website admins to oversee accounts, orders, products, and categories efficiently.

- Empower website admins to perform CRUD operations on product and category information.
- Enable website admins to review and modify order statuses for new orders.
- Provide customers with the ability to search and view product details.
- Allow customers to comment, rate on movies and complete orders with payment options (Paypal/ VNPay).
- Notify users about orders to help customers keep in track of their order process and to help the admin know customer's actions on order

4- Final product:

An web application combined with a recommender system

HEAD OF INFORMATION TECHNOLOGY PROGRAM	**INSTRUCTOR**
(Sign with full name)	*(Sign with full name)*
	Mai Anh Thơ

HOCHIMINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION
FACULTY OF INTERNATIONAL EDUCATION
fie.hcmute.edu.vn

THE SOCIALIST REPUBLIC OF VIETNAM

Independence – Freedom– Happiness

****

INSTRUCTOR'S EVALUATION SHEET

Students' names:

1. Chu Nguyễn Hoàng Sơn ID: 19119128 Class: 19110CLA2
2. Nguyễn Quốc Hoàng ID: 19110128 Class: 19110CLA1

Major: **Information technology**

Project title: **Creating an online movie streaming web application**

Instructor: **PhD. Mai Anh Thơ**

EVALUATION:

1. Content of project:

.....

.....

.....

.....

.....

.....

2. Strengths:

.....

.....

.....

.....

.....

.....

3. Drawbacks:

.....

.....

.....

.....

.....

.....

4. Approval for oral defense? (Approved or denied)

.....

.....

5. Overall evaluation (Excellent, Good, Fair, Poor): .....

6. Mark: ……..... (in words: …….....……..)

Ho Chi Minh City, December, 2024

INSTRUCTOR

(Sign with full name)

Mai Anh Thơ

HOCHIMINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION
FACULTY OF INTERNATIONAL EDUCATION
fie.hcmute.edu.vn

THE SOCIALIST REPUBLIC OF VIETNAM

Independence – Freedom– Happiness

****

REVIEWER'S EVALUATION SHEET

Students' names:

1. Chu Nguyễn Hoàng Sơn ID: 19119128 Class: 19110CLA2
2. Nguyễn Quốc Hoàng ID: 19110128 Class: 19110CLA1

Major: **Information technology**

Project title: **Creating an online movie streaming web application**

Reviewer: **MSc. Nguyễn Đức Khoan**

EVALUATION:

1. Content and workload of the project:

.....

.....

.....

.....

.....

.....

2. Strengths:

.....

.....

.....

.....

.....

.....

3. Drawbacks:

.....

.....

.....

.....

.....

.....

4. Approval for oral defense? (Approved or denied)

.....

.....

5. Overall evaluation (Excellent, Good, Fair, Poor): .....

6. Mark: ……..... (in words: …….....……..)

Ho Chi Minh City, December, 2023

REVIEWER

(Sign with full name)

Mai Anh Thơ

ACKNOWLEDGEMENTS

We would like to express our appreciation to the instructor, PhD Mai Anh Tho, who worked closely with us to develop a project that enabled us to successfully finish our report. Throughout this course, you have constantly assisted us with our difficulties in the study process as well as provided us with excellent feedback and recommendations to help us grow as developers in the future. As a result, you have become our inspiration for this final project, and we would like to express our gratitude for the opportunity to learn from you on this course. Everyone on our team has various talents, and we have discovered our shortcomings in each other over this final assignment, but we've all managed to face this challenge as a team. As a result, everyone on our team is grateful for the opportunity to collaborate.

We now have all we need to learn about this subject and do well in it. Additionally, we want to thank our classmates for supplying us with a plethora of knowledge and pertinent facts that helped us to strengthen our thesis.

We finished the subject and report in a short measure of time, with restricted information and numerous different limitations regarding philosophies and skill with programming execution. As a consequence of this, there will undoubtedly be flaws in the process of creating the topic; consequently, we eagerly await your significant feedback so that we can increase the number of our experts and perform better the following time. This capstone project is the result of our tireless efforts throughout the semester. Because we are still novices, we are aware that our design has numerous shortcomings. We welcome any feedback as well as suggestions for improving our project. We appreciate it. Sincerely!

Ho Chi Minh City, December 2024

Chu Nguyễn Hoàng Sơn 19119128

Nguyễn Quốc Hoàng 19110128

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1. Reason for choosing the topic

More and more people enjoy watching movies online to relax instead of going outside nowadays, the idea of a movie watching and streaming application has become an interesting topic to develop. We are looking to create something for entertainment but still convenient.

Building a movie watching and streaming website for users lets you share a wide range of movies with others, without needing a physical space like a movie theater but still be able to spend time with friends and family.

It's a simple and modern way for people to enjoy films right from their devices, making it easy to watch anytime, anywhere.

1.2. Purpose of project

Provide users with convenient and instant access to a vast library of movies from various genres, languages, and time periods.

The application aims to deliver an enjoyable, personalized viewing experience by offering a user-friendly platform.

They can explore, discover, and watch films of their choice at any time, from anywhere.

1.3. Object & scope

Product scope:

- Product Goal: Build an online movie streaming platform similar to major platforms, providing an optimal user experience with rich content and ease of use.
- Core Functions: Focus on streaming, account management, payment processing, and copyright protection.
- Target Users: Movie watchers with internet access, using multiple platforms from mobile, desktop, to smart TV.
- Development Direction: Create a diverse content library, integrate AI to

suggest content based on user preferences, expand payment options, and support multiple device formats.

Project scope:

- Platform Development: Design and implement the backend (server, database) and frontend (user interface).
- Service Integration: Connect with payment services, security (DRM), and data analytics.
- Testing and Quality Assurance: Perform functional, security, and performance testing on different devices.
- Project Management: Plan the deployment in phases (MVP, additional feature development), manage the development team, and track progress.
- Support and Maintenance: Update the platform, provide technical support for users, and monitor the system to ensure continuous operation.

1.4. Expected results

The anticipated outcomes include the creation of a dynamic online movie platform where users experience seamless interactions, such as browsing, watching, commenting, and rating movies. With the integration of a robust recommendation system, personalized movie suggestions based on user preferences are expected to enhance user engagement and satisfaction. Streamlined features for administrators ensure efficient management of movie libraries, user accounts, and content moderation. This user-friendly platform aims to deliver an enjoyable, accessible experience tailored to movie enthusiasts on desktop devices.

CHAPTER 02: THEORY FUNDAMENTAL

2.1. System Architecture

In our front-end development, we utilize ReactJS with NextJS framework. At back-end, we harness the capabilities of NodeJS for API development. Our database solution is powered by PostgreSQL. The overall system architecture adheres to the Client-Server Model, strategically capitalizing on the inherent flexibility and performance benefits offered by these seamlessly integrated technologies.

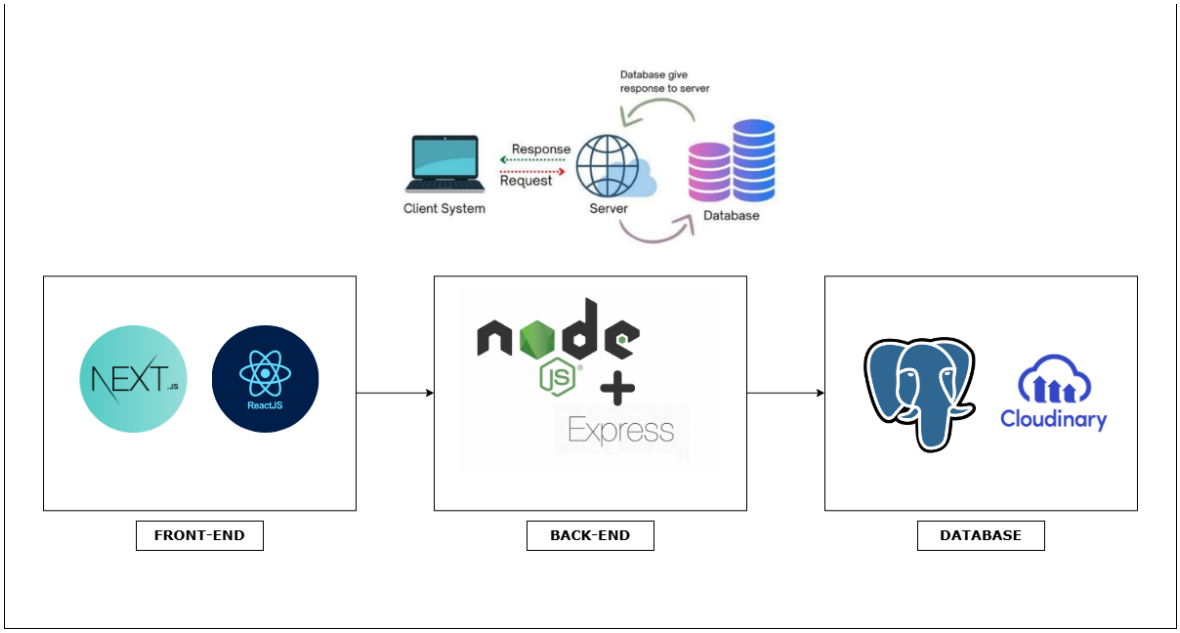


Figure 1. System Architecture

2.2. Libraries

2.2.1. Front- End

Table 1. Front-End Libraries

Library	Version	Description
nextui-org/*	2.x.x	A set of React components designed for building user interfaces with ease, part of the NextUI framework.
paypal/react-paypal-js	8.7.0	A React wrapper for integrating PayPal buttons and other functionalities.
radix-ui/react-slider	1.2.1	A customizable slider component for React apps.
react-aria/ssr	3.9.4	Tools for handling server-side rendering accessibility issues.
react-aria/visually-hidden	3.8.12	A utility for visually hidden elements (like screen-reader text).
tabler/icons-react	3.22.0	Icon components for React, based on Tabler's icon library.
canvas-confetti	1.9.3	A lightweight library to create confetti effects on a web canvas.
cloudinary	2.5.1	A library for managing image and video uploads and transformations via the Cloudinary API.
clsx	2.1.1	A utility for dynamically combining class names based on conditions.
embla-carousel-autoplay	8.3.0	A lightweight, customizable carousel slider library for React.
embla-carousel-react	8.3.0	A plugin for Embla Carousel that enables autoplay functionality.

framer-motion	11.1.1	A powerful animation library for React, useful for creating fluid and interactive UI animations.
glob	11.0.0	A utility to find files/directories using patterns (useful for working with file systems).
intl-messageformat	10.5.0	A library for formatting messages for internationalization
moment	2.30.1	A popular library for parsing, manipulating, and formatting dates and times.
next	14.2.13	The core Next.js library for building React-based web applications with server-side rendering and static site generation.
next-auth	4.24.8	An authentication library for Next.js supporting various providers (e.g., Google, GitHub, etc.).
next-cloudinary	6.16.0	A Next.js-specific Cloudinary utility for managing media in your app.
next-themes	0.2.1	A theme management library for handling light/dark mode in Next.js.
react	18.3.1	The core libraries for building React applications.
react-dom	18.3.1	The core libraries for building React applications
react-simple-star-rating	5.1.7	A simple star rating component for React apps.
react-toastify	10.0.6	A library for showing toast notifications in React apps.

rimraf	6.0.1	A utility for deleting files and directories (a cross-platform alternative to rm -rf).
tailwindcss	3.4.3	A utility-first CSS framework for creating custom designs efficiently.
typescript	5.0.4	A superset of JavaScript that adds static typing, helping catch errors early during development.
eslint	8.57.0	A linting tool to ensure consistent code style and catch common coding errors.

2.2.2. Back- End

Table 2. Back-End Libraries

Library	Version	Description
bcryptjs	2.4.3	A library for hashing passwords. It is used to securely store and verify user passwords.
cloudinary	2.5.1	Provides integration with the Cloudinary API to handle image and video uploads, storage, and transformations.
cookie-parser	1.4.7	Middleware for parsing cookies in HTTP requests.
cors	2.8.5	Middleware for enabling Cross-Origin Resource Sharing, allowing your server to handle requests from different origins.
cross-env	7.0.3	Allows you to set environment variables across different operating systems in a consistent way.

express	4.21.0	A popular web framework for Node.js, used to create APIs and handle HTTP requests and responses.
google-auth-library	9.14.2	A library for implementing Google authentication and working with Google APIs.
googleapis	144.0.0	Provides access to Google's APIs, such as Drive, Sheets, and Calendar.
jsonwebtoken	9.0.2	A library for generating and verifying JSON Web Tokens (JWTs), commonly used for authentication and secure data transfer.
moment	2.29.4	A library for manipulating and formatting dates and times.
moment-timezone	0.5.46	An extension of Moment.js for handling time zones.
multer	1.4.5-lts.1	A middleware for handling multipart/form-data, commonly used for uploading files.
nodemailer	6.9.15	A library for sending emails through Node.js.
pg	8.13.0	A library for interfacing with PostgreSQL databases, allowing you to query and manage the database.
ws	8.18.0	A WebSocket library for enabling real-time communication between the server and clients.

eslint	8.24.0	A tool to identify and fix coding issues, ensuring a consistent code style.
nodemon	3.1.7	A utility that automatically restarts your server when it detects file changes, streamlining the development process

2.3. Technology

2.3.1. Reactjs

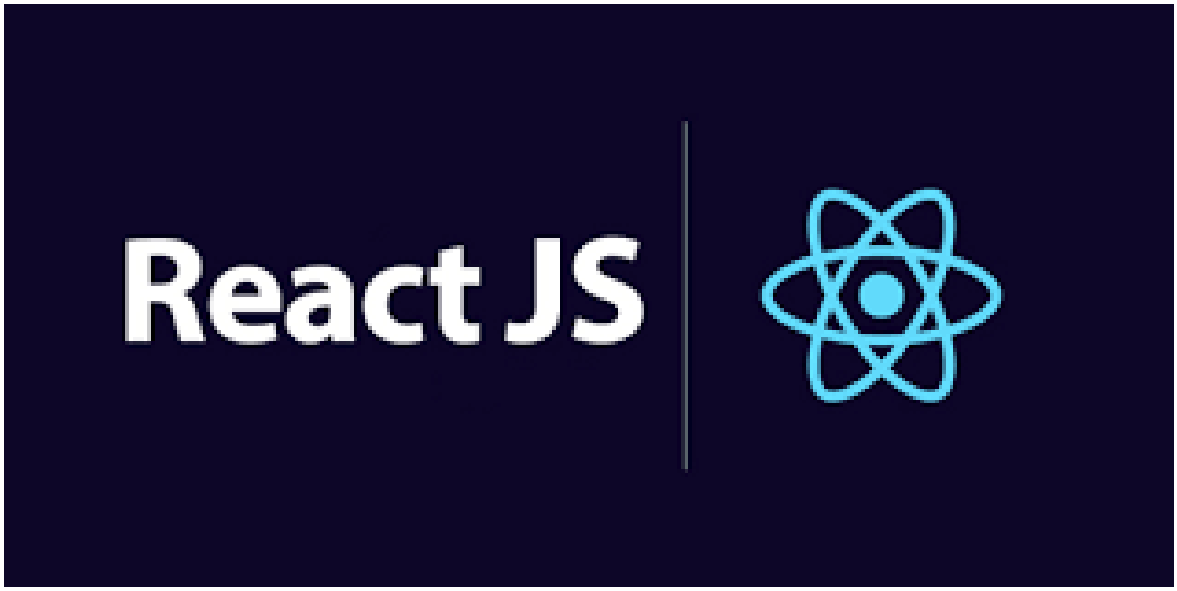


Figure 2: Reactjs

What is Reactjs?

React.js is an open-source JavaScript library for building user interfaces. Developed by Facebook, it's widely used for creating dynamic and efficient web applications. React.js employs a component-based architecture, allowing developers to build UIs as a collection of reusable and modular components. It utilizes a virtual DOM, enhancing performance by minimizing actual DOM manipulations. React.js supports a unidirectional data flow, making it easier to manage and update state changes. Its declarative syntax simplifies the creation of complex UIs, promoting code readability and maintainability. React.js is often used in conjunction with other libraries or frameworks and is known for its efficiency in handling real-time updates and rendering large datasets.

Why use Reactjs?

- Component-Based Architecture: React.js promotes code reusability and maintainability through a component-based approach, where UIs are built as a collection of modular components.

- Virtual DOM Efficiency: React's use of a virtual DOM minimizes actual DOM manipulations, ensuring efficient updates and optimizing overall performance.
- Declarative Syntax: The declarative nature of React simplifies the development of complex UIs, making code more readable and easier to understand.
- Unidirectional Data Flow: React supports unidirectional data flow, facilitating the management of state changes and enhancing predictability in application behavior.
- Community and Ecosystem: The strong community and extensive ecosystem surrounding React provide ample support, libraries, and tools for developers, contributing to the framework's robustness.
- Popularity and Adoption: React's widespread adoption by major companies ensures long-term viability and continuous improvement, making it a reliable choice for web application development.

How to apply Reactjs to build our project?

In our project, ReactJS is fundamental for front-end development, enabling a dynamic user interface.

- Front-end Construction: React's component-based architecture streamlines code organization, ensuring a maintainable codebase.
- Event Handling: React facilitates seamless responses to user interactions through functions responding to events like clicks or input changes.
- Route Management: Incorporating React Router enhances navigation, ensuring intuitive updates based on URL changes.
- API Interaction: React facilitates efficient data communication with the backend through technologies like Fetch API or Axios.

In conclusion, our strategy maximizes ReactJS, empowering the creation of a robust, interactive web application with a responsive UI, effective event handling, streamlined navigation, and seamless API interactions.

2.3.2. NextJS

Figure 3: NextJS

What is NextJS?

Next.js is a powerful open-source React framework designed for building fast, scalable, and SEO-friendly web applications. It simplifies web development by offering features like server-side rendering (SSR), static site generation (SSG), and API routes out of the box. With Next.js, developers can create dynamic and optimized websites that provide an excellent user experience. Its developer-friendly structure and robust community support make it a popular choice for modern web development.

Why use NextJS?

- Server-Side Rendering (SSR): Next.js allows rendering pages on the server, improving performance and ensuring content is SEO-friendly. This is especially useful for content-heavy or dynamic applications.
- Static Site Generation (SSG): With SSG, pages can be pre-rendered at build time, making them load faster and reducing server overhead.
- API Routes: Next.js enables creating backend API endpoints directly in the project, eliminating the need for a separate server.
- Routing Made Simple: The file-based routing system automatically generates routes based on the folder structure, simplifying navigation and reducing setup time.

- Performance Optimization: Built-in features like image optimization, code splitting, and automatic bundling ensure fast-loading and efficient applications.
- Flexibility: Developers can choose between SSR, SSG, or client-side rendering (CSR) for different parts of the application, adapting to specific project requirements.
- Built-in CSS and Sass Support: Next.js supports CSS modules and preprocessors like Sass, allowing for modular and maintainable styling.
- TypeScript Support: Seamless integration with TypeScript ensures type safety and better maintainability.
- Community and Ecosystem: As part of the React ecosystem, Next.js benefits from an active community, extensive documentation, and numerous third-party plugins.
- Open Source: Next.js is open source, making it freely available for developers to use, modify, and collaborate.

How to apply NextJS to build our project?

Next.js plays a pivotal role in streamlining the process of building modern web applications. Its powerful features, such as server-side rendering (SSR), static site generation (SSG), and dynamic routing, simplify the creation of performant, scalable, and SEO-friendly applications. With Next.js, developers can efficiently structure their projects, optimize page loading, and enhance user experience without the complexity of managing separate tools or configurations. By providing a unified framework, Next.js reduces development time while ensuring flexibility and maintainability in building robust web applications..

2.3.3. NodeJS

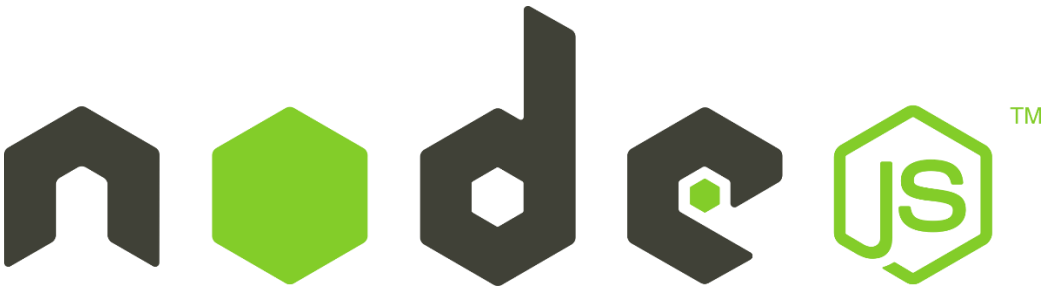


Figure 4: NodeJS

What is NodeJS?

Node.js is an open-source, cross-platform runtime environment for executing JavaScript code on the server side. Built on Chrome's V8 JavaScript engine, Node.js is designed to build scalable network applications. Its event-driven, non-blocking I/O model ensures high performance and makes it ideal for building real-time, data-intensive applications.

Why use NodeJS?

- High Performance: Node.js is built on the V8 engine, offering fast execution and high performance for server-side applications.
- Asynchronous and Event-Driven: The non-blocking I/O model enables handling multiple requests simultaneously, making it suitable for real-time applications.
- JavaScript Everywhere: With Node.js, developers can use JavaScript for both client-side and server-side, unifying the development stack.
- Scalability: Its lightweight architecture and support for microservices allow building scalable applications.
- Rich Ecosystem: The npm (Node Package Manager) provides access to thousands of open-source libraries, accelerating development.
- Real-Time Capabilities: Node.js excels in building real-time applications like chat apps and collaborative tools using WebSockets.
- API Development: Node.js simplifies the creation of RESTful APIs, allowing

seamless integration with frontend frameworks.

- Cross-Platform Support: Developers can use Node.js to build applications that run on multiple platforms, including Windows, macOS, and Linux.
- Community and Support: Node.js has an extensive and active community that provides plugins, tools, and documentation.
- Open Source: Node.js is freely available, fostering collaboration and innovation within the developer community.

How to apply NodeJS to build our project?

Node.js is an open-source runtime environment for executing JavaScript on the server side. Built on Chrome's V8 engine, it is ideal for creating scalable, real-time, and data-intensive applications with its event-driven, non-blocking I/O model.

How to Use Node.js in a Project:

- Building APIs: Develop robust RESTful APIs using frameworks like Express.js for efficient routing and middleware management.
- Database Connectivity: Connect to databases like MongoDB or PostgreSQL with libraries like Mongoose or Sequelize for seamless data handling.
- Security with JWT: Secure APIs with JSON Web Token (JWT) authentication, ensuring safe user access.

2.3.4. ExpressJS

Figure 5: ExpressJS

What is ExpressJS?

Express.js is a minimal, flexible, and powerful web application framework for Node.js. It provides a robust set of features for building web applications and APIs. By simplifying the process of handling HTTP requests, routing, and middleware, Express.js allows developers to create scalable and efficient server-side applications with ease.

Why use ExpressJS?

- Simplified Server Setup: Express.js abstracts the complexities of Node.js and provides an intuitive way to set up a server.
- Routing Made Easy: It offers a simple and powerful routing mechanism to define and handle different endpoints in your application.
- Middleware Support: Express.js supports middleware, enabling efficient request processing, authentication, logging, and error handling.
- Scalability: Its modular structure makes it suitable for building small applications and scaling them into complex systems.
- Integration with Databases: Express.js integrates seamlessly with databases like MongoDB, PostgreSQL, and PostgreSQL, providing flexibility for back-end development.
- Extensive Community and Ecosystem: With a rich ecosystem of plugins and tools, developers can extend Express.js functionality to meet project requirements.
- Lightweight: Its minimalist design ensures high performance and low overhead, making it ideal for real-time applications and APIs.
- Compatible with Front-End Frameworks: Express.js works seamlessly with modern front-end libraries and frameworks like React, Angular, and Vue.js.
- Customizability: Developers can customize every aspect of the application to suit their needs without being restricted by rigid conventions.
- Open Source: As an open-source framework, Express.js is free and backed by

a vibrant community.

2.3.5. Cloudinary

Figure 6: Cloudinary

What is Cloudinary?

Cloudinary is a cloud-based media management platform that provides storage, optimization, and delivery services for images and videos. It allows developers to upload, store, manage, transform, and deliver media content seamlessly using advanced cloud computing features. Cloudinary is widely used by developers and businesses for media hosting, real-time optimization, and fast delivery via Content Delivery Networks (CDNs).

Key features of Cloudinary include:

- Cloud-based storage for media assets (images, videos, audio, etc.).
- Media transformation (resize, crop, compress, and edit images and videos dynamically).
- Adaptive bitrate streaming for optimal video playback across different devices and network conditions.
- CDN-powered delivery for faster content access globally.
- Security features, including access control, encryption, and authentication.
- Analytics tools to monitor media usage and optimize performance.

Why use Cloudinary?

Cloudinary offers several compelling advantages that make it ideal for online media hosting and delivery, especially for an online movie streaming website:

- Efficient Video Storage & Delivery: Cloudinary provides scalable cloud storage for large video files, ensuring videos can be securely stored and accessed without impacting server performance.
- Optimized Streaming: With adaptive bitrate streaming, Cloudinary adjusts video quality in real-time based on a user's network speed and device type, minimizing buffering and improving user experience.
- Faster Loading Times: Cloudinary uses a global Content Delivery Network (CDN) to cache and deliver media content closer to users' locations, ensuring faster and smoother streaming experiences.
- Media Compression & Transformation: Videos and images can be resized, transcoded, and compressed on the fly without manual intervention, reducing file sizes while maintaining quality and performance.
- Ease of Integration: Cloudinary provides APIs and SDKs for most popular programming languages and frameworks, making integration straightforward for developers.
- Scalability: Cloudinary is built for scalability, allowing businesses to scale their media storage and delivery infrastructure as user demand increases.
- Secure Media Hosting: Cloudinary offers encryption, access control, and secure media access, ensuring that user-generated or proprietary media is protected.

How to apply Cloudinary to build our project?

To implement Cloudinary for an online movie streaming website, we would utilize its cloud-based media management capabilities to store, manage, and deliver streaming video content efficiently. Cloudinary provides secure, scalable, and optimized media storage solutions, allowing the website to store a vast catalog of videos, trailers, and other streaming content.

Using Cloudinary's advanced features, such as adaptive bitrate streaming, video compression, and CDN-powered delivery, we can ensure seamless playback experiences for users across different devices and network conditions. Cloudinary's cloud storage capabilities handle large video files, ensuring reliability and scalability as the movie streaming library expands. Its integration allows for faster delivery of video content by utilizing a global CDN, ensuring minimal buffering for users regardless of their location.

Integrating Cloudinary into the online movie streaming website provides a powerful, secure, and structured video hosting solution. This ensures that users can stream movies, TV shows, and other multimedia content without interruptions, with optimized video performance, reduced load times, and reliable playback quality. Cloudinary's analytics also allow website administrators to track usage patterns and make data-driven decisions to further optimize the user experience.

2.3.6. PostgreSQL

Figure 7: PostgreSQL

What is PostgreSQL?

PostgreSQL is a powerful, advanced, and widely used open-source relational database management system (RDBMS). It is known for its stability, scalability, and robustness, making it an ideal choice for businesses of all sizes. PostgreSQL is developed and supported by a global community of developers and offers comprehensive SQL compliance along with advanced features:

- PostgreSQL is a database management system.
- PostgreSQL databases are relational.
- PostgreSQL is Open Source and free to use.
- The PostgreSQL Database Server is fast, reliable, scalable, and easy to integrate.
- PostgreSQL supports advanced data types and indexing mechanisms.
- It works across multiple platforms, supporting both client-server architecture and embedded systems.
- A variety of contributed tools and extensions are available for PostgreSQL.

Why use PostgreSQL?

PostgreSQL is one of the most popular choices for database management for many compelling reasons:

- PostgreSQL is free and distributed under an open-source license, meaning

it can be used without licensing costs.

- PostgreSQL supports advanced SQL features that make it a powerful database tool for complex queries.
- PostgreSQL uses standards-compliant SQL, making it an effective and versatile database for modern development.
- It integrates seamlessly with programming languages such as Python, Java, C++, PHP, Ruby, and more.
- PostgreSQL is highly efficient when working with large and complex datasets.
- It has strong support for advanced data types like JSON, XML, spatial data, arrays, and user-defined types.
- PostgreSQL scales well even when managing databases with millions of rows, and it supports advanced indexing options for better query performance.
- PostgreSQL's extensibility allows users to add custom features, operators, or data types to the system as required.

How to apply PostgreSQL to build our project?

To implement PostgreSQL for an online movie streaming website, we would design a relational database schema to store vital business information such as movie details, user accounts, subscriptions, viewing history, and payment transactions. PostgreSQL tables would include entities such as movies, genres, users, subscriptions, watchlists, viewing history, and payment information.

Using PostgreSQL's advanced querying features, we can effectively and efficiently retrieve and manipulate data to ensure seamless and fast user experiences on the streaming platform. Its ability to maintain data integrity through ACID compliance ensures that user accounts, subscription plans, payment transactions, and user viewing activity are always accurate and consistent. Additionally, PostgreSQL’s

scalability allows it to handle a growing number of users and an expanding catalog of movies.

Integrating PostgreSQL with the online movie streaming website provides a robust, secure, and structured backend system. This ensures user data is well-organized, accessible, and manageable, delivering a strong foundation for building features such as personalized recommendations, user profiles, seamless payment options, and a reliable streaming experience.

CHAPTER 03: SURVEY AND SYSTEM REQUIREMENT

3.1. Surveys

3.1.1. General background

- The past decade has witnessed a remarkable surge in the popularity of online movie streaming platforms. This shift in consumer behavior is evident in the increasing number of movies distributed by platforms like Netflix, Hulu, Prime Video, and Disney+. The convenience, affordability, and vast content libraries offered by these platforms have made them the preferred choice for many viewers.
- The growing demand for on-demand content has compelled traditional media companies to adapt to the evolving landscape and invest in their own streaming services. This has led to a proliferation of platforms, each offering unique content libraries and features to cater to diverse viewer preferences.
- The growth of online movie streaming platforms has also had a profound impact on the film industry. With a growing number of platforms competing for viewers' attention, there has been an increased emphasis on producing original content. This has created new opportunities for filmmakers and content creators, allowing them to reach wider audiences and experiment with innovative storytelling formats. Additionally, the rise of streaming platforms has led to a shift in the traditional theatrical release model, with some films being released simultaneously on streaming services and in theaters.
- As technology continues to advance, we can expect further innovation and growth in the realm of online movie streaming, offering viewers an even wider range of content and viewing experiences.

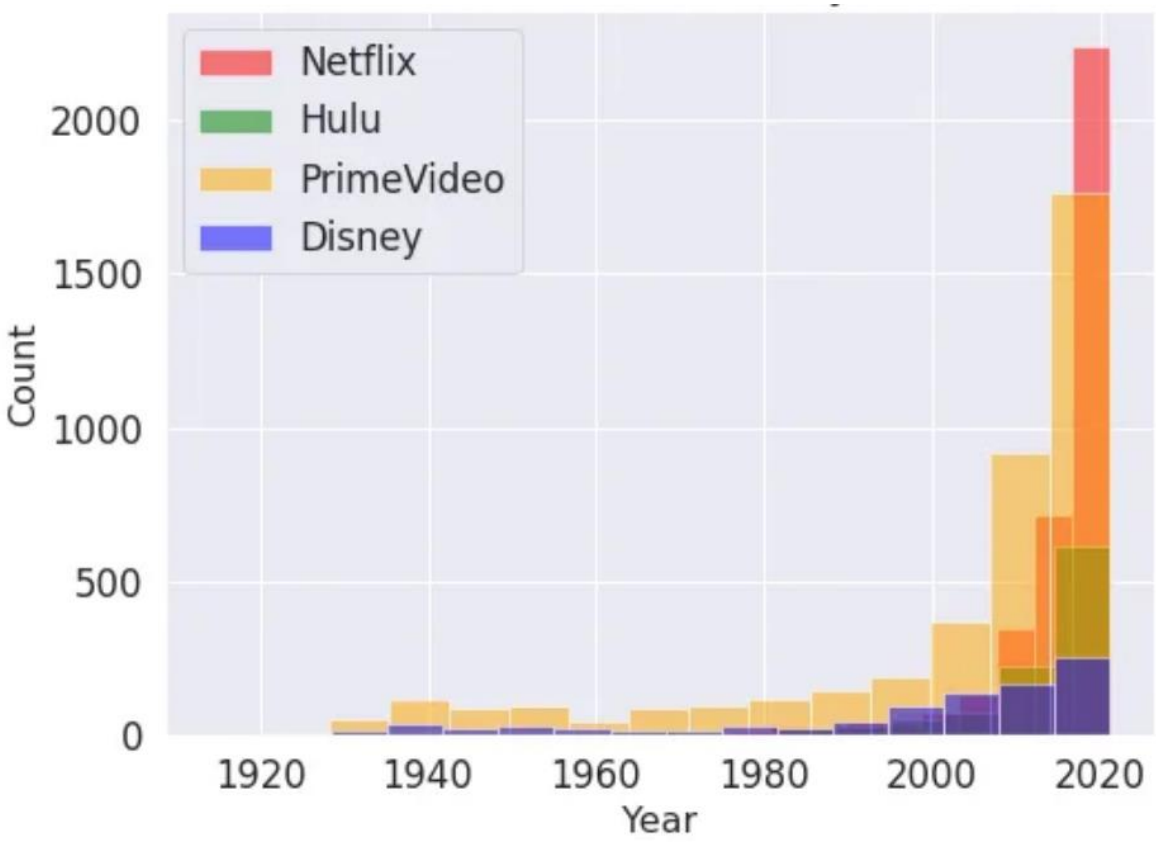


Figure 8: Year of Movie Distribution by Platforms

3.1.2. Surveys on popular movie streaming platforms

3.1.2.1. Netflix

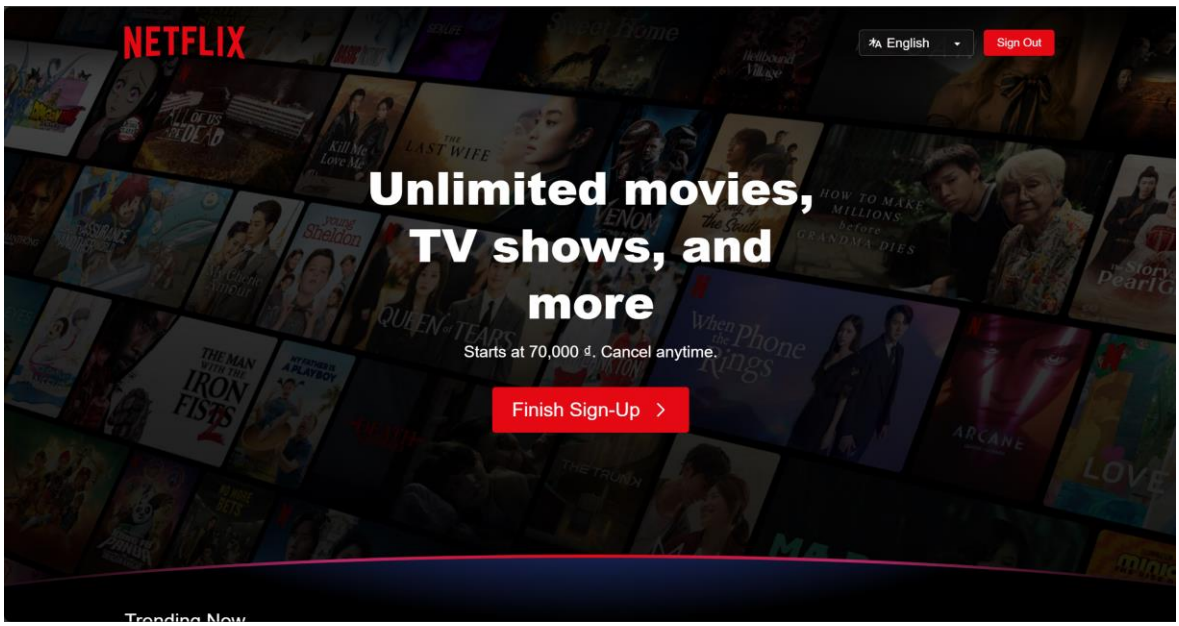


Figure 9: Netflix website

Table 3. Advantages and Disadvantages of Netflix

Advantages and Disadvantages of Netflix	
Advantages	✓ Users can watch content anytime and anywhere, as long as they have an internet connection and a compatible device. ✓ Netflix's intuitive interface makes it easy to browse and discover new shows and movies. ✓ The algorithm suggests content based on viewing habits, helping users find shows and movies that match their tastes. ✓ Users can create individual profiles under one account, ensuring personalized recommendations and watchlists for family members. ✓ Allows users to download content for offline viewing, which is particularly helpful during travel or in areas with poor connectivity. ✓ No interruptions from advertisements, providing a seamless entertainment experience. ✓ Features like Kids' Profiles and maturity level filters ensure a safe viewing experience for children. ✓ Access to diverse content from different cultures and languages, broadening entertainment options.
Disadvantages	- Users can watch content anytime and anywhere, as long as they have an internet connection and a compatible device. - Netflix's intuitive interface makes it easy to browse and discover new shows and movies. - The algorithm suggests content based on viewing

	habits, helping users find shows and movies that match their tastes. - Users can create individual profiles under one account, ensuring personalized recommendations and watchlists for family members. - Allows users to download content for offline viewing, which is particularly helpful during travel or in areas with poor connectivity. - No interruptions from advertisements, providing a seamless entertainment experience. - Features like Kids' Profiles and maturity level filters ensure a safe viewing experience for children. - Access to diverse content from different cultures and languages, broadening entertainment options.

3.1.2.2. Hulu

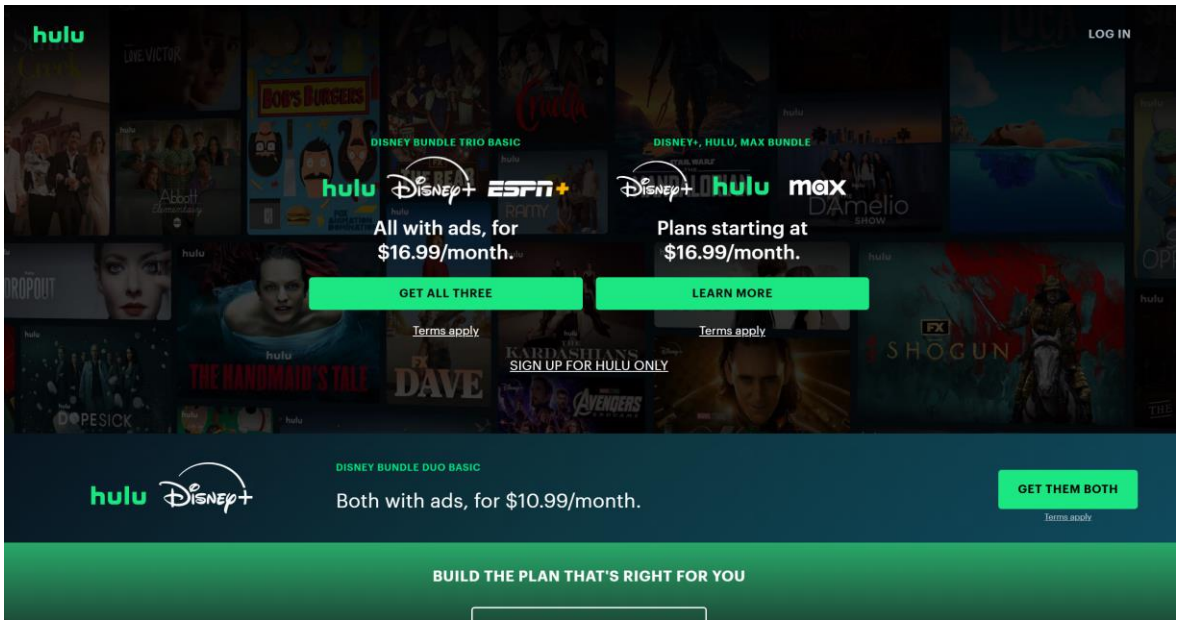


Figure 10: Hulu website

Table 4. Advantages and Disadvantages of Hulu

Advantages and Disadvantages of Hulu	
Advantages	✓ Hulu provides budget-friendly plans, including an ad-supported tier, making it accessible to more users. ✓ The Hulu + Live TV option lets users stream live sports, news, and other programming, combining traditional TV with streaming. ✓ Hulu features exclusive shows like The Handmaid's Tale and Only Murders in the Building, as well as licensed content from major networks. ✓ Allows up to six profiles per account, ensuring tailored recommendations and watchlists for different users. ✓ Offers Kids Profiles, filtering content to be age-appropriate for younger viewers. ✓ Content can be downloaded on mobile devices for offline streaming, handy for travel or areas with poor internet connectivity. ✓ Hulu offers bundles with Disney+ and ESPN+ at a discounted rate, providing diverse entertainment options.
Disadvantages	- Subscription fees may rise over time, and premium plans can be costly, especially for families needing multiple simultaneous streams. - Shows and movies can be removed from the platform due to licensing agreements, disappointing users who might not have finished watching. - Requires a stable and high-speed internet connection

	for smooth streaming, which can be a limitation in some areas. - With recent efforts to limit account sharing, users might face restrictions if they share their subscription with others. - The extensive library can lead to indecision, making it hard to settle on something to watch. - Content availability varies by location due to licensing restrictions, which might frustrate users looking for specific titles. - Binge-watching culture can lead to excessive screen time, impacting health and productivity. - Not all content is of high quality; some Netflix Originals receive criticism for being mediocre or rushed.

3.2. Functional requirements

3.2.1. Authorization

Table 5. Authorization

No.	Actor	Description
1	Guest	Users do not have any account.
2	Customer	Users have already had a customer account on this application.
3	Admin	Users have already had an admin account on this application.

3.2.2. System's Operations

Table 6. System's Operations

No.	Function	Description

1	Register/Login	Customers create a new account/ login to website and application.
2	Forgot Password	Customers forgot password and can reset password with email.
3	Edit profile	Customers can edit their profile information
4	Active email	Customers can active their email
5	View account information	Customers can view their account information
6	View movie list	Customers can view movie list
7	View subscription plans	Customers can view subscription plans
8	Pay with VNPay, PayPal	Customers can pay with VNPay or PayPal
9	View transaction history	Customers can view their transaction history
10	View policy	Customers can view policy
11	Watch movie	Customers can watch movie
12	Comment and rating	Customers can comment and rating
13	Filter movie	Customers can filter movie
14	Manage favourite list	Customers can manage favourite list
15	Manage user account	Admin can manage user account
16	Manage movies	Admin can manage movies
17	Manage episodes	Admin can manage episodes
18	Manage categories	Admin can manage categories
19	Manage subscriptions	Admin can manage subscriptions
20	View statistics	Admin can view website statistics

3.3. Non-Functional requirements

- User-friendly interface for easy navigation.

- Swift data processing to ensure efficient system performance.
- Strong security measures to protect customer's personal information.
- Easy updates and future expansions through a user-friendly system.
- Maintain data integrity to ensure customer trust and satisfaction.

3.4. Modeling requirements

Table 7. Modeling requirement

Actor	Requirement
Guest	- Login - Register - View policy
Customer	- Log in/out - Register - Edit profile - Reset password with email - Active email - View movie list - View subscription plans - Pay with VNPay, PayPal - View transaction history - View policy - View account info - Watch movie - Comment and rating - Filter movie - Manage favourite list
Admin	- Manage user’s account - Manage movies

	- Manage episodes - Manage categories - Manage subscriptions - View statistic

CHAPTER 4: SYSTEM ANALYST AND DESIGN

4.1. USECASE DIAGRAMS

Figure 11: System Diagram

4.2. Usecase specification

4.2.1. Login

Table 8: Login use case specification

<table>
<tr><td>Name</td><td>Log In</td></tr>
<tr><td>Brief Description</td><td>The way user website logins</td></tr>
<tr><td>Actor(s)</td><td>All user login into system</td></tr>
<tr><td colspan="2">Flow of Events</td></tr>
<tr><td colspan="2">Basic Flow</td></tr>
<tr><td colspan="2">This use case starts when the user wants to access the system
1. User select “Login” function on the system
2. The system displays the Login page
3. User enter the correct username and password then click "Log in"
4. The system will return the user to Home page.</td></tr>
<tr><td colspan="2">Alternate Flows</td></tr>
<tr><td>Title</td><td>Description</td></tr>
<tr><td>Enter the wrong login information</td><td>1. If the user enters an invalid password, the system will announce the username or password is invalid and ask the user to re-enter the username or password.</td></tr>
<tr><td colspan="2">Pre-Conditions</td></tr>
<tr><td>Title</td><td>Description</td></tr>
<tr><td></td><td>The user already has an account on the system</td></tr>
<tr><td colspan="2">Post-Conditions</td></tr>
</table>

Title	Description
Success	Log in successfully, the user accesses the system.
Failure	Login failed; the user cannot log in.
Extension Points	
None	

4.2.2. Register

Table 9: Register use case specification

Name	Register
Brief Description	The way customer creates personal account
Actor(s)	User
Flow of Events	
Basic Flow	
This use case starts when the user chooses the function to register 1. User chooses the "Register" function in the system. 2. The system displays a Register page with personal information and website information as username, password, email, real name, and phone number. 3. Customer fills out the above information and accept the contracts and policies then presses the "Register" button. 4. The notification system has successfully registered.	
Alternate Flows	
Title	**Description**

Fill out the information missing	1. The system requires re-fill and full information in the "Register" page.
Pre-Conditions	
Title	**Description**
Failed	Account has already existed in the system
Post-Conditions	
Title	**Description**
Success	Register successfully, user create account successfully.
Failure	Register failed; user created account failure.
Extension Points	

4.2.3. Forgot password

Table 10: Forgot password use case specification

Name	Forgot password
Brief Description	The way user can change password by sending system users' email
Actor(s)	All kinds of users
Flow of Events	
Basic Flow	
This use case starts when the user chooses the function forgot password 1. User chooses the "Forgot password" function in the system.	

2. The system will need user to input their email for sending reset account token 3. Turn back to our website for input the token. 4. If token is valid, will let the user re-password. 5. The system will notify when the user is re-password successfully.

Alternate Flows	
Title	**Description**
Fill out the email not correct with account	1. Users will not see the token to re-password their account.
Pre-Conditions	
Title	**Description**
Failed	The token is not valid
Post-Conditions	
Title	**Description**
Success	Reset password successfully
Failure	Reset password failed
Extension Points	

4.2.4. Change profile information

Table 11: Change profile use case specification

Name	Change profile information
Brief Description	User changes personal profile
Actor(s)	Customer
Flow of Events	
Basic Flow	
This use case starts when the user wants to edit profile 1. User fill in input tags 2. User change avatar by selecting his/her image 3. Click Edit button	
Alternate Flows	
Title	**Description**
Fill out the information missing	1. The system requires re-fill empty input at the "Profile page".
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in
Post-Conditions	
Title	**Description**
Success	Edit successfully

Failure	Edit failed
Extension Points	

4.2.5. Verify email

Table 12: Change profile use case specification

Name	Verify email
Brief Description	User want to verify his/her email
Actor(s)	Customer
Flow of Events	
Basic Flow	
This use case starts when the user chooses the function Verify email 1. Go to "Profile page" and click send Token 2. The token will send in user's email 3. Get the token and type in the token input 4. If the token is valid, system will notify when the email verified successfully	
Alternate Flows	
Title	**Description**
Fill out the wrong token	1. The system requires correct the token or resend token to verify email.
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in

Post-Conditions	
Title	**Description**
Success	Verified successfully
Failure	Verified failed
Extension Points	

4.2.5. Add to favorite

Table 13: Add to favorite use case specification

Name	Add to favorite
Brief Description	User add a movie/episode to favorite
Actor(s)	Customer
Flow of Events	
Basic Flow	
This use case starts when the user chooses the function Add to favorite 1. Select a movie/episode you want 2. Click “Add to favorite” button	
Alternate Flows	
Title	**Description**
None	None
Pre-Conditions	
Title	**Description**

None	None
Post-Conditions	
Title	**Description**
Success	Add successfully
Failure	Add failed
Extension Points	

4.2.6. Remove from favorite

Table 14: Remove from favorite use case specification

Name	Remove from favorite
Brief Description	User remove a movie/episode from favorite
Actor(s)	Customer
Flow of Events	
Basic Flow	
This use case starts when the user chooses the function Remove to favorite 1. Click the Remove to favorite button 2. System notify removed successfully	
Alternate Flows	
Title	**Description**
Pre-Conditions	

Title	Description
Logged in	Must be logged in
Post-Conditions	
Title	**Description**
Success	Remove successfully
Failure	Remove failed
Extension Points	

4.2.7. Filter movies

Table 15: Filter movies use case specification

Name	Filter movies
Brief Description	User filter movies
Actor(s)	Customer
Flow of Events	
Basic Flow	
This use case starts when the user wants to filter the movie categories 1. Select filter option 2. Click on 'Filter' button	
Alternate Flows	
Title	**Description**

Pre-Conditions	
Title	**Description**
Post-Conditions	
Title	**Description**
Success	Filter successfully
Failure	Cannot filter
Extension Points	

4.2.8. View detail movie

Table 16: View detail movie use case specification

Name	View detail movie
Brief Description	User view information of movie
Actor(s)	Customer
Flow of Events	
Basic Flow	
This use case starts when the user selects movie to view detail 1. Select product to view 2. Click watch or choose the episode to view detail movie	
Alternate Flows	

Title	Description
The movie is not having any episode	System will show “Coming Soon”
Pre-Conditions	
Title	**Description**
Post-Conditions	
Title	**Description**
Success	View successfully
Failure	Cannot view
Extension Points	

4.2.9. Comment in detail movie

Table 17: Comment in detail movie use case specification

Name	Comment in detail movie
Brief Description	User comment information of movie
Actor(s)	Customer
Flow of Events	
Basic Flow	
This use case starts when the user selects movie to view detail then comment	

1. Select product to view 2. Click watch or choose the episode to view detail movie 3. Comment and rating in the movie/episode	
Alternate Flows	
Title	**Description**
Missing rating or comment	System will notify to rating and comment before submit comment
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in
Post-Conditions	
Title	**Description**
Success	Comment successfully
Failure	Cannot comment
Extension Points	

4.2.10. Edit comment in detail movie

Table 18: Comment in detail movie use case specification

Name	Edit comment in detail movie
Brief Description	User edit comment information of movie
Actor(s)	Customer

Flow of Events	
Basic Flow	
This use case starts when the user selects movie to view detail then edit comment 1. Select product to view 2. Click watch or choose the episode to view detail movie 3. Edit comment in the movie/episode	
Alternate Flows	
Title	**Description**
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in
Post-Conditions	
Title	**Description**
Success	Edit comment successfully
Failure	Cannot edit comment
Extension Points	

4.2.11. Checkout VNPAY

Table 19: Checkout VNPAY use case specification

Name	Checkout VNPAY
Brief Description	Process for user to checkout

Actor(s)	Customer
Flow of Events	
Basic Flow	
This use case starts when the user checkout 1. Select subscription in the “Subscription page” 2. View the subscription bill 3. Select payment method VNPAY 4. Redirect to VNPAY to select options to checkout 5. Fill VNPAY personal information or scan QR code 6. Confirm checkout 7. Return “Invoice page” to show transaction result	
Alternate Flows	
Title	**Description**
Not have enough money	System shows not having enough money
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in
VNPAY account	Having VNPAY account
Post-Conditions	
Title	**Description**
Success	Checkout successfully

Failure	Checkout failed
Extension Points	

4.2.12. Checkout PAYPAL

Table 20: Checkout PAYPAL use case specification

Name	Checkout PAYPAL
Brief Description	Process for user to checkout
Actor(s)	Customer
Flow of Events	
Basic Flow	
This use case starts when the user checkout 1. Select subscription in the “Subscription page” 2. View the subscription bill 3. Select payment method PAYPAL 4. Show popup PAYPAL to fill information 5. Confirm checkout 6. Return “Subscription page” to show transaction result	
Alternate Flows	
Title	**Description**
Not have enough money	System shows not having enough money
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in

PAYPAL account	Having PAYPAL account
Post-Conditions	
Title	**Description**
Success	Checkout successfully
Failure	Checkout failed
Extension Points	

4.2.13. Create account

Table 21: Create account use case specification

Name	Create account
Brief Description	Admin want to create account
Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when admin wants to create an account 1. Get into "Admin user page" 2. Fill information 3. Click "Create" button	
Alternate Flows	
Title	**Description**
Missing information	When admin fill missing information

Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	
Title	**Description**
Success	Create successfully
Failure	Create failed
Extension Points	

4.2.14. Disable account

Table 22: Disable account use case specification

Name	Disable account
Brief Description	Admin will disable account (cannot use anymore)
Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when the admin disables account (can understand as banning account) 1. Move to Admin user page 2. Click to view detail of user account 3. Click “Disable account”	

Alternate Flows	
Title	**Description**
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	
Title	**Description**
Success	Disable successfully
Failure	Disable failed
Extension Points	

4.2.15. Active account

Table 23: Active account use case specification

Name	Active account
Brief Description	Admin will active account (can continue to use)
Actor(s)	Admin
Flow of Events	
Basic Flow	

This use case starts when the admin actives account (can understand as banning account) 1. Move to Admin user page 2. Click to view detail of user account 3. Click "Active account"	
Alternate Flows	
Title	**Description**
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	
Title	**Description**
Success	Active successfully
Failure	Active failed
Extension Points	

4.2.16. Create subscription

Table 24: Create subscription use case specification

Name	Create subscription
Brief Description	Admin create subscription

Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when the admin creates subscription 1. Move to Admin subscription page 2. Fill subscription information 3. Click “Create ticket” button	
Alternate Flows	
Title	**Description**
Missing information	System notifies to fill all fields
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	
Title	**Description**
Success	Create successfully
Failure	Create failed
Extension Points	

4.2.17. Modify subscription

Table 25: Modify subscription use case specification

Name	Modify subscription
Brief Description	Admin modify subscription
Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when the admin modifies subscription 1. Move to Admin subscription page 2. Select subscription to view detail 3. Modify field 4. Click “Modify ticket” button	
Alternate Flows	
Title	**Description**
Missing information	System notifies to fill all fields
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	
Title	**Description**
Success	Modify successfully

Failure	Modify failed
Extension Points	

4.2.18. Delete subscription

Table 26: Delete subscription use case specification

Name	Delete subscription
Brief Description	Admin delete subscription
Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when the admin deletes subscription 1. Move to Admin subscription page 2. Select subscription to view detail 3. Click “Delete ticket” button	
Alternate Flows	
Title	**Description**
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	

Title	Description
Success	Delete successfully
Failure	Delete failed
Extension Points	

4.2.19. Create movie

Table 27: Create movie use case specification

Name	Create movie
Brief Description	Admin create movie
Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when the admin creates movie 4. Move to Admin movie page 5. Fill movie information 6. Click “Create movie” button	
Alternate Flows	
Title	**Description**
Missing information	System notifies to fill all fields
Pre-Conditions	

Title	Description
Logged in	Must be logged in as admin
Post-Conditions	
Title	**Description**
Success	Create successfully
Failure	Create failed
Extension Points	

4.2.20. Modify movie

Table 28: Modify movie use case specification

Name	Modify movie
Brief Description	Admin modify movie
Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when the admin movies subscription 5. Move to Admin movie page 6. Select movie to view detail 7. Modify field 8. Click “Modify movie” button	
Alternate Flows	

Title	Description
Missing information	System notifies to fill all fields
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	
Title	**Description**
Success	Modify successfully
Failure	Modify failed
Extension Points	

4.2.21. Create episode

Table 29: Create episode use case specification

Name	Create episode
Brief Description	Admin create episode
Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when the admin creates episode 1. Move to Admin episode page	

2. Fill episode information 3. Click “Create episode” button	
Alternate Flows	
Title	**Description**
Missing information	System notifies to fill all fields
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	
Title	**Description**
Success	Create successfully
Failure	Create failed
Extension Points	

4.2.22. Delete episode

Table 30: Delete episode use case specification

Name	Delete episode
Brief Description	Admin delete episode
Actor(s)	Admin
Flow of Events	

Basic Flow	
This use case starts when the admin deletes episode 1. Move to Admin episode page 2. Select episode to view detail 3. Click “Delete episode” button	
Alternate Flows	
Title	**Description**
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	
Title	**Description**
Success	Delete successfully
Failure	Delete failed
Extension Points	

4.2.23. Create category

Table 31: Create category use case specification

Name	Create category
Brief Description	Admin create category

Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when the admin creates category 7. Move to Admin category page 8. Fill category information 9. Click “Create category” button	
Alternate Flows	
Title	**Description**
Missing information	System notifies to fill all fields
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	
Title	**Description**
Success	Create successfully
Failure	Create failed
Extension Points	

4.2.24. Modify category

Table 32: Modify category use case specification

Name	Modify category
Brief Description	Admin modify category
Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when the admin modifies category 9. Move to Admin category page 10. Select category to view detail 11. Modify field 12. Click “Modify category” button	
Alternate Flows	
Title	**Description**
Missing information	System notifies to fill all fields
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	
Title	**Description**
Success	Modify successfully

Failure	Modify failed
Extension Points	

4.2.25. Delete category

Table 33: Delete category use case specification

Name	Delete category
Brief Description	Admin delete category
Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when the admin deletes category 4. Move to Admin category page 5. Select category to view detail 6. Click “Delete category” button	
Alternate Flows	
Title	**Description**
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin
Post-Conditions	

Title	Description
Success	Delete successfully
Failure	Delete failed
Extension Points	

4.2.26. Filter statistics

Table 34: Filter statistics use case specification

Name	Filter statistics
Brief Description	Admin filter statistics
Actor(s)	Admin
Flow of Events	
Basic Flow	
This use case starts when the admin wants to filter the statistics 1. Select filter option 2. Click on ‘Filter’ button	
Alternate Flows	
Title	**Description**
Pre-Conditions	
Title	**Description**
Logged in	Must be logged in as admin

Post-Conditions	
Title	**Description**
Success	Filter successfully
Failure	Cannot filter
Extension Points	

4.3. Sequence diagrams

4.3.1. Login

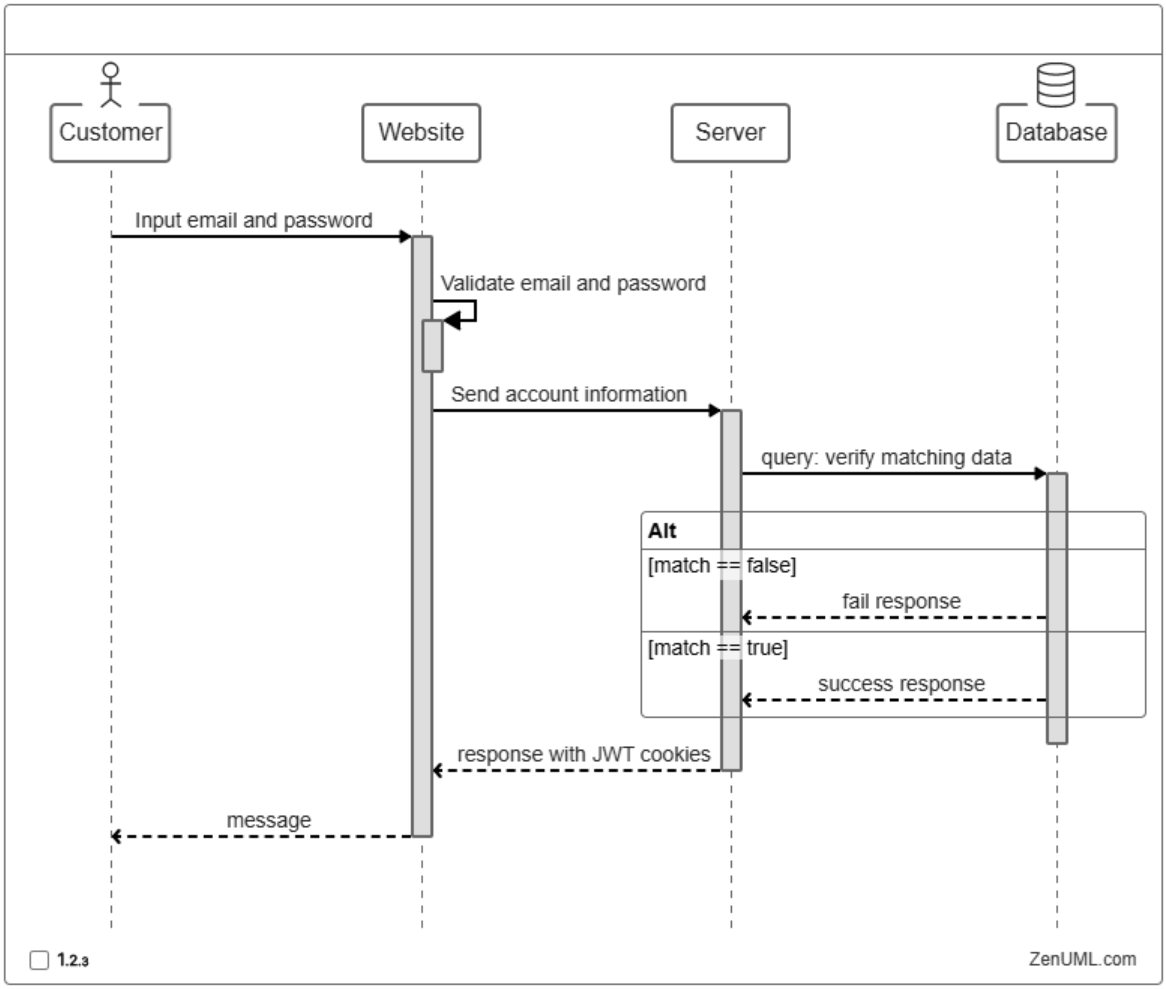


Figure 12: Login sequence

4.3.2. Register

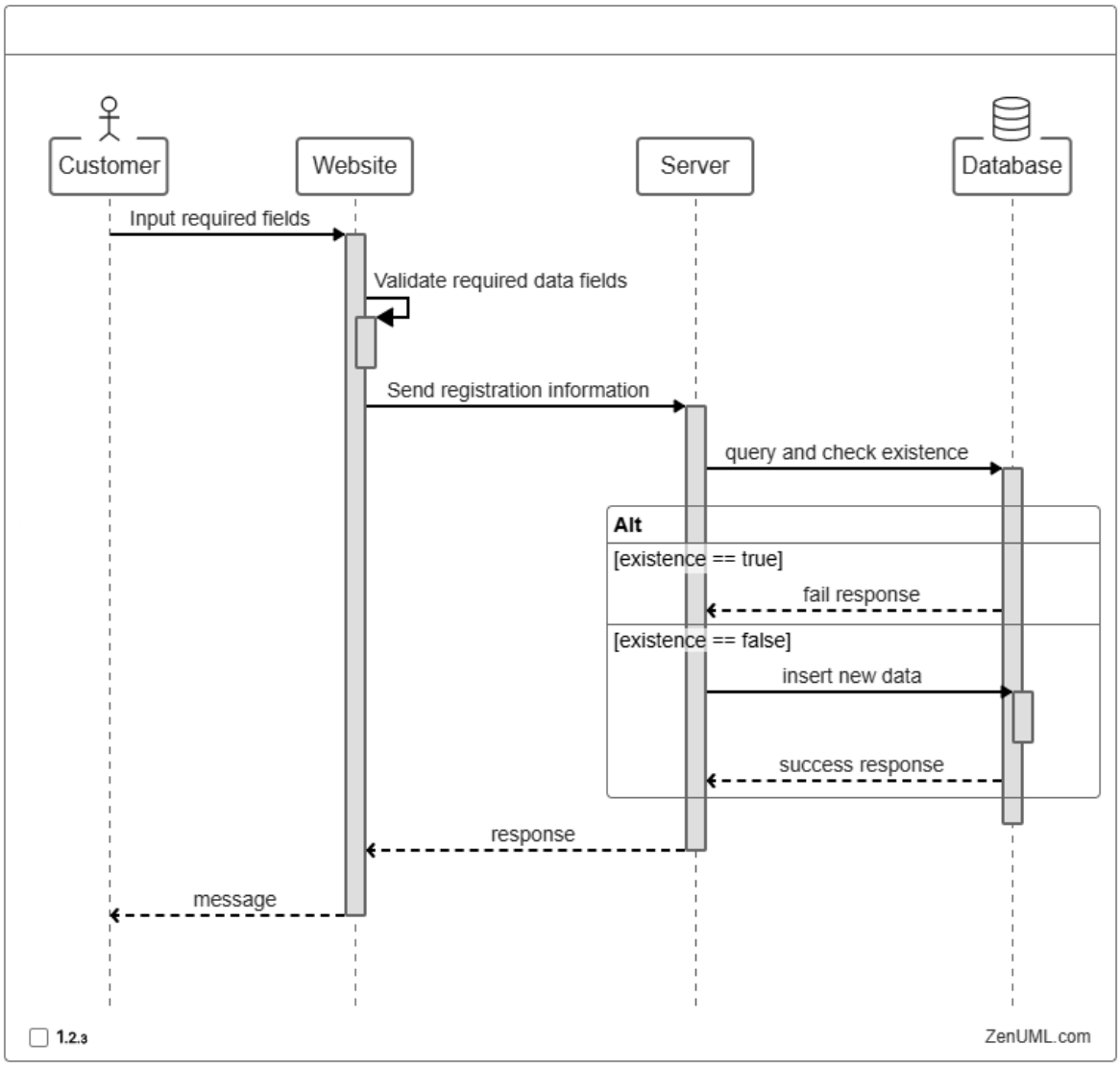


Figure 13: Register sequence

4.3.3. Reset password

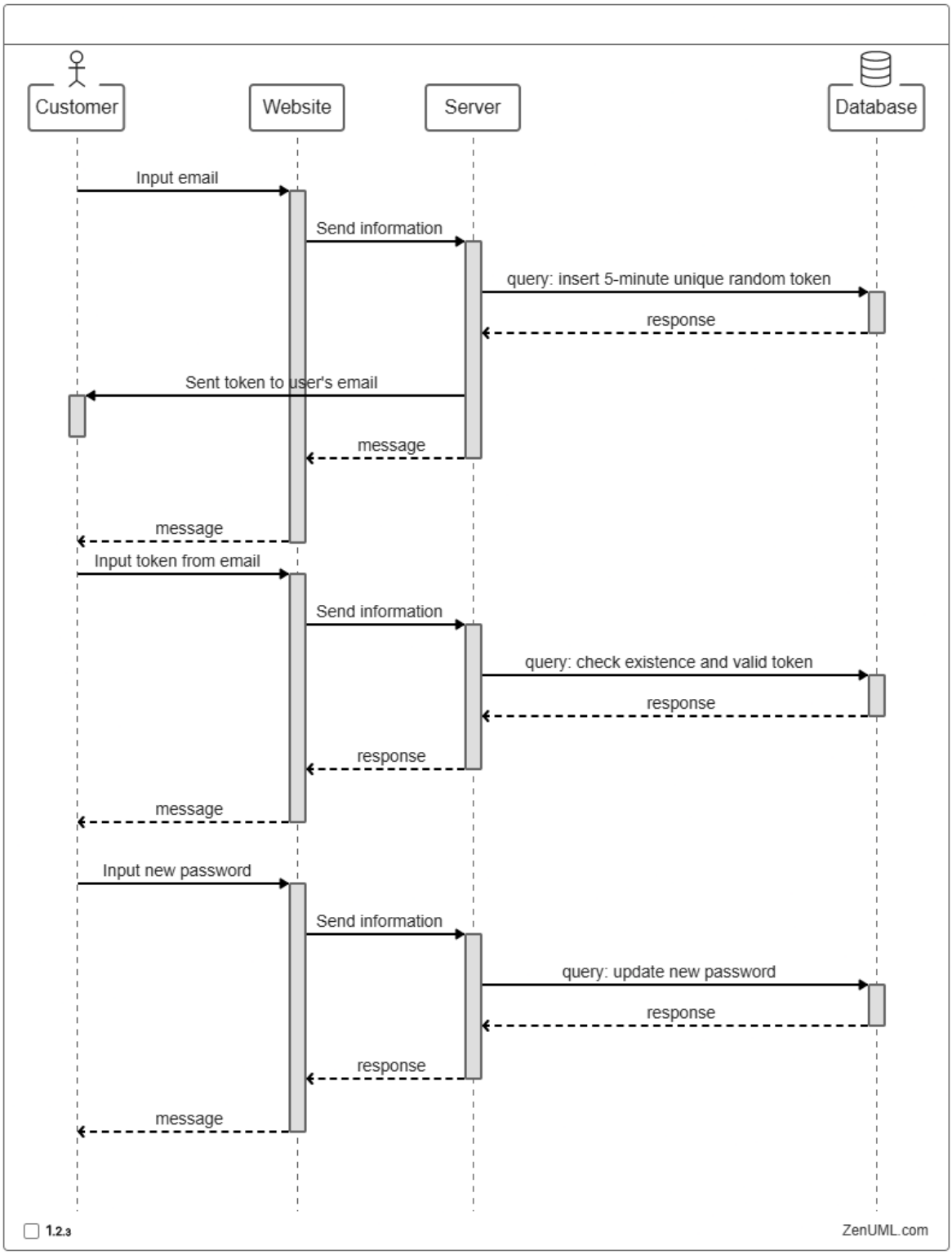


Figure 14: Reset password sequence

4.3.4. Change profile information

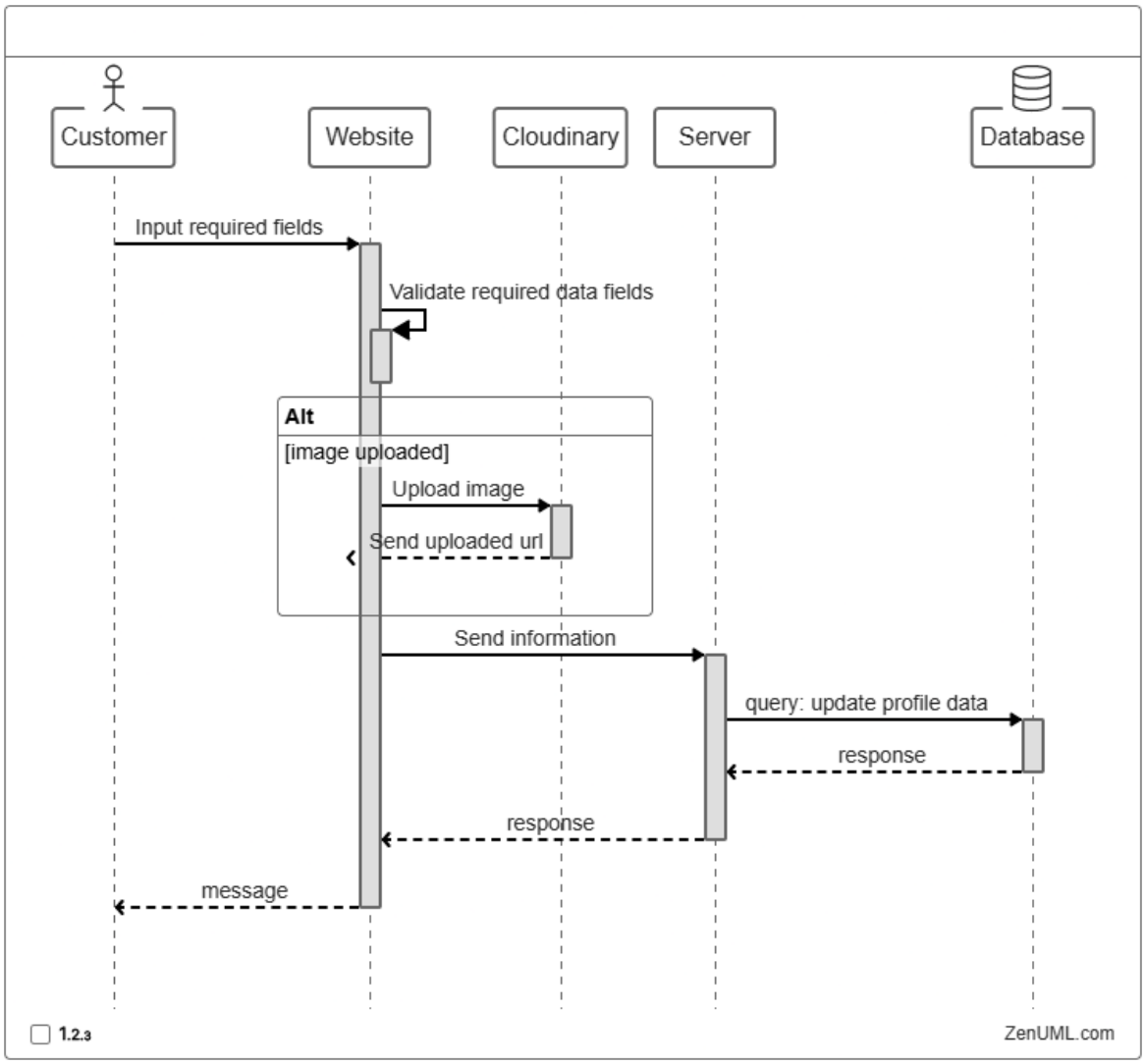


Figure 15: Change profile information sequence

4.3.5. View movie information

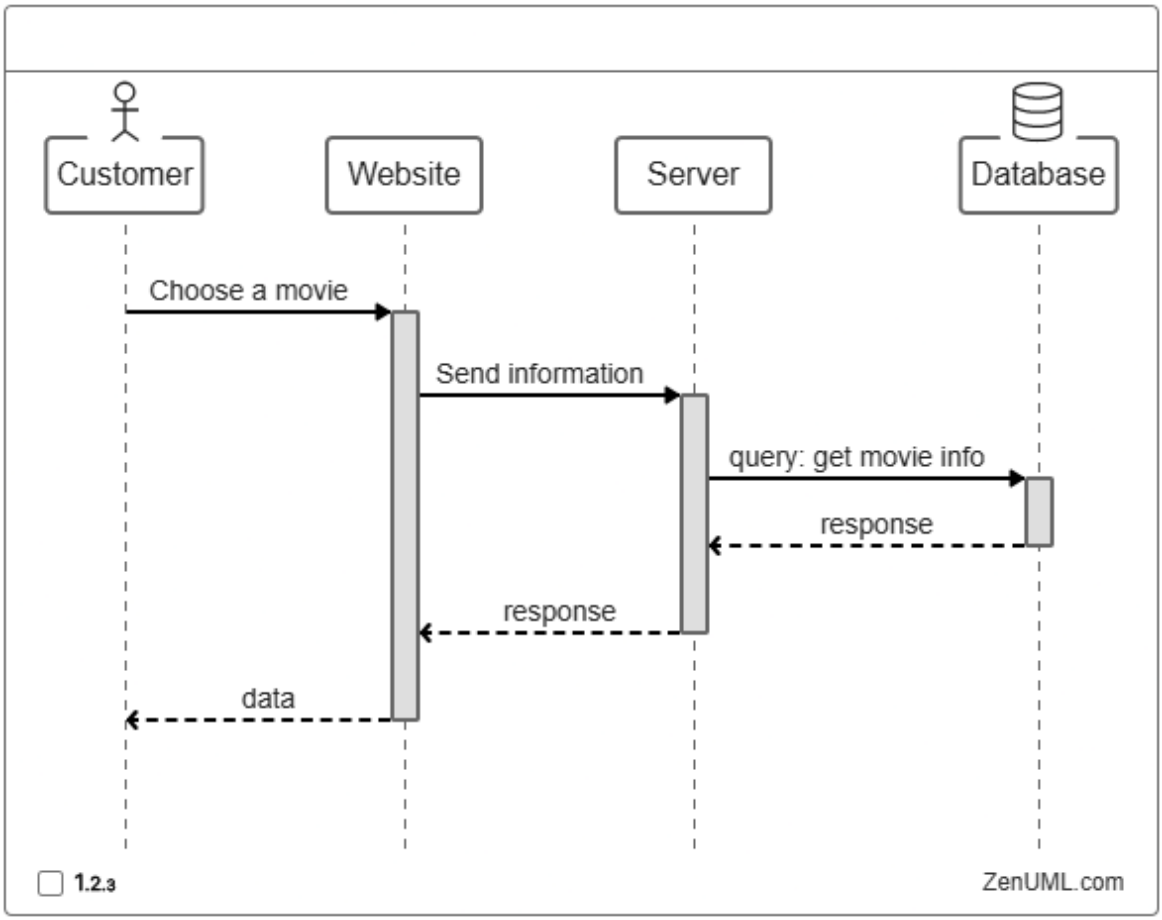


Figure 16: View movie information sequence

4.3.6. View episode information

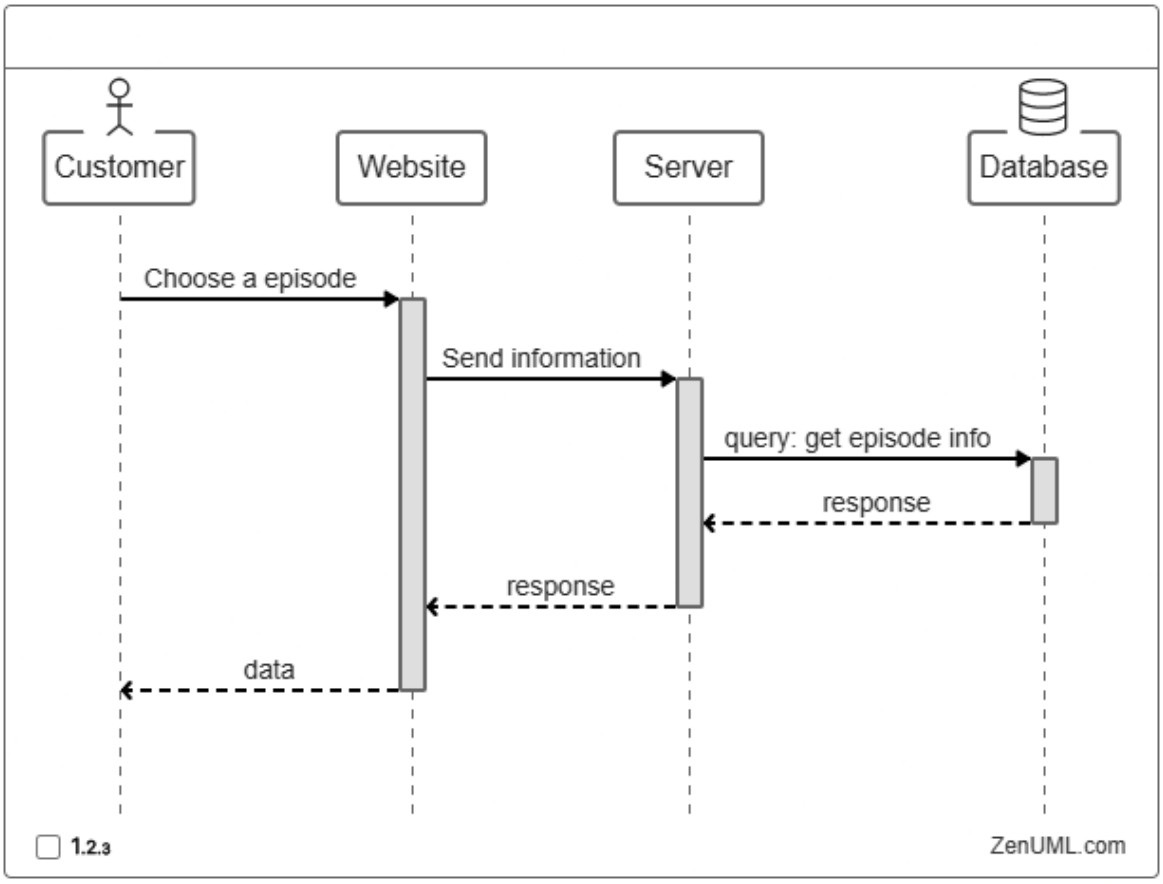


Figure 17: View episode information sequence

4.3.7. Filter movies

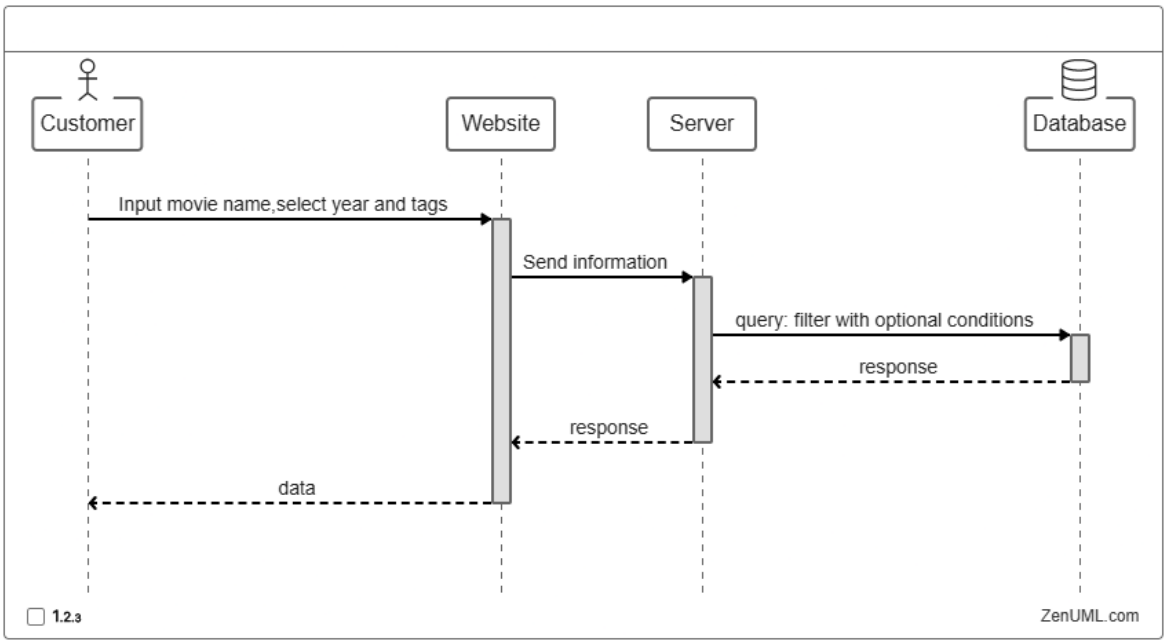


Figure 18: Filter movies sequence

4.3.8. Favourite list management

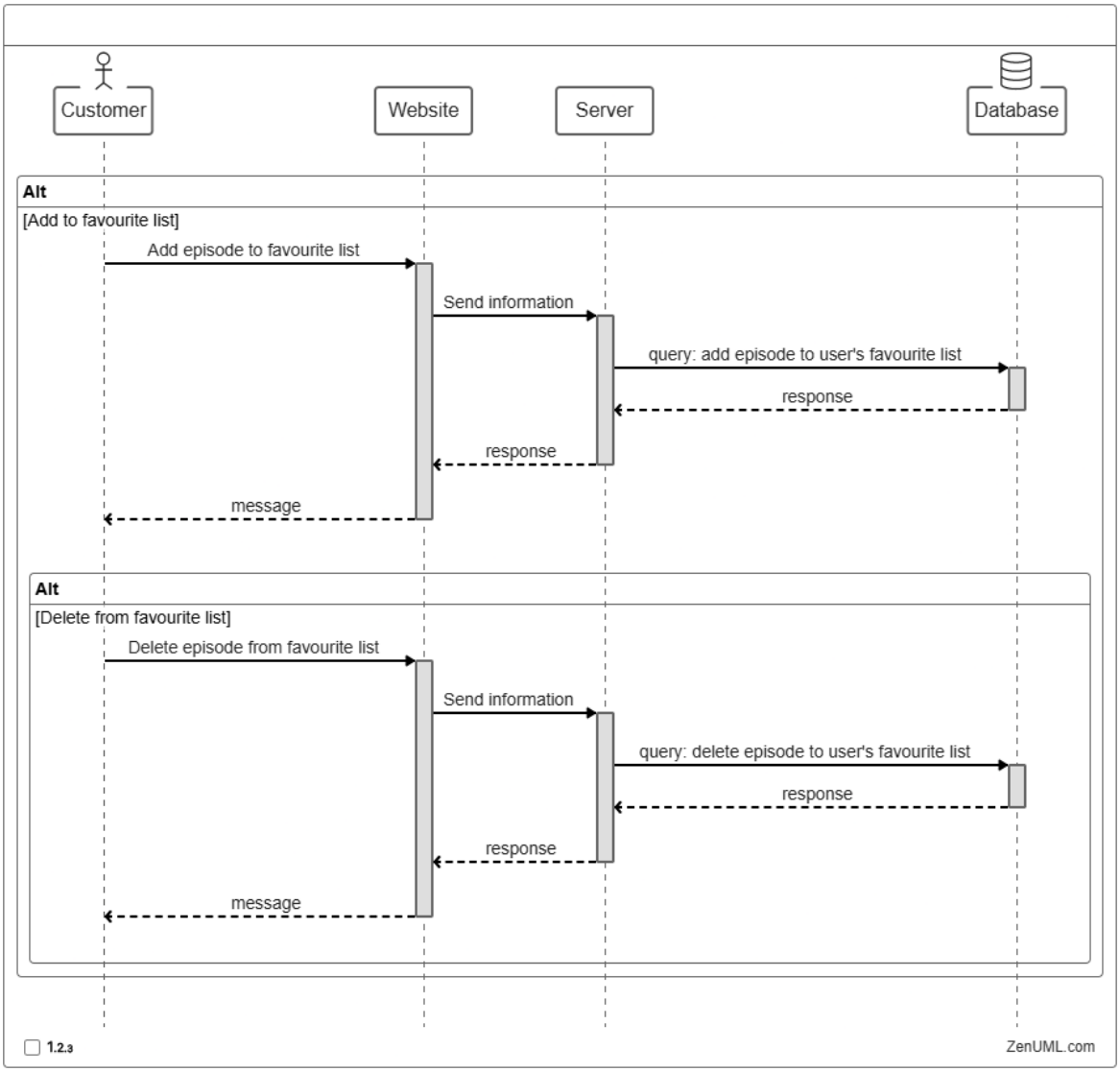


Figure 19: Favourite list management sequence

4.3.9. Manage comment

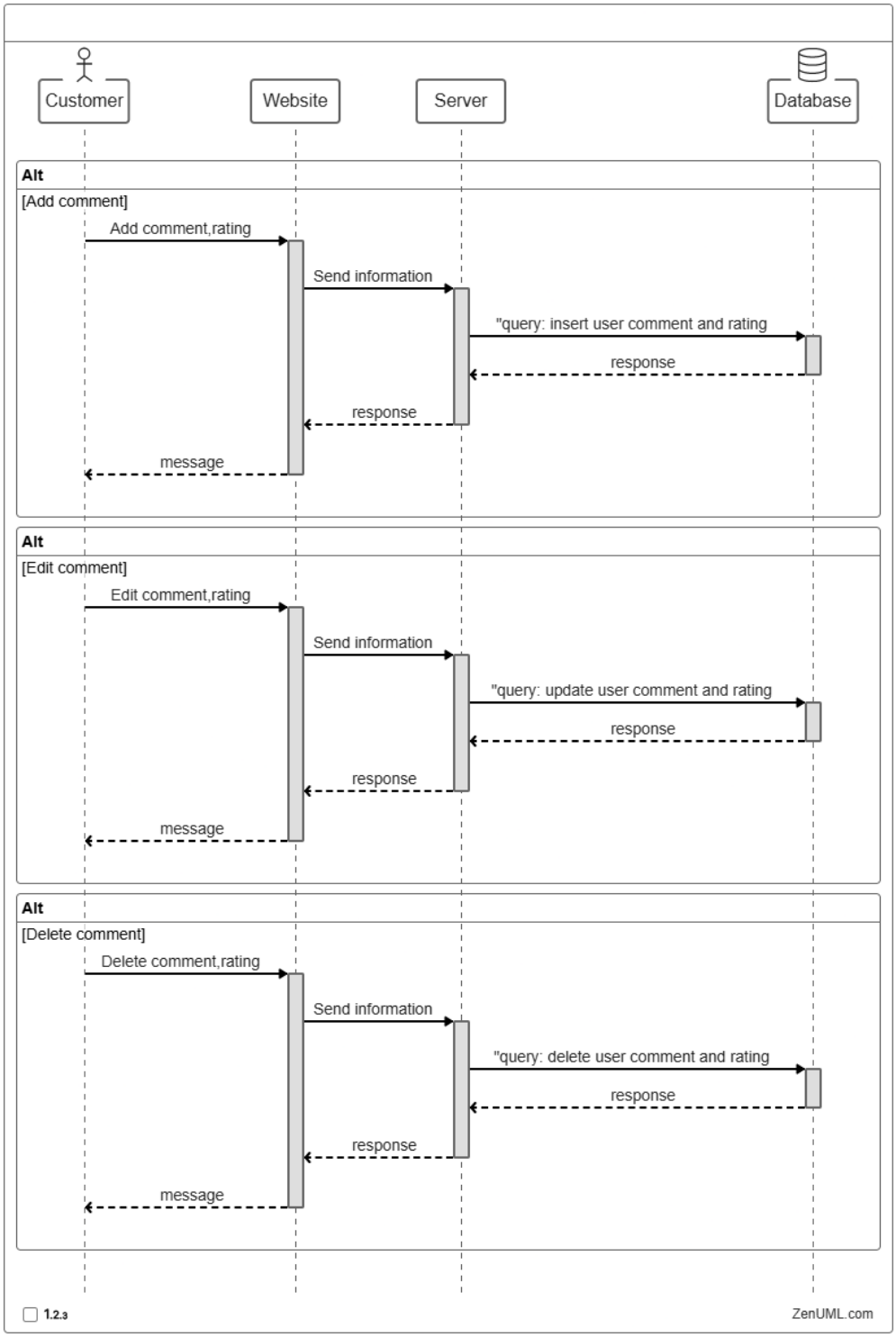


Figure 20: Manage comment diagrams

4.3.10. Active email

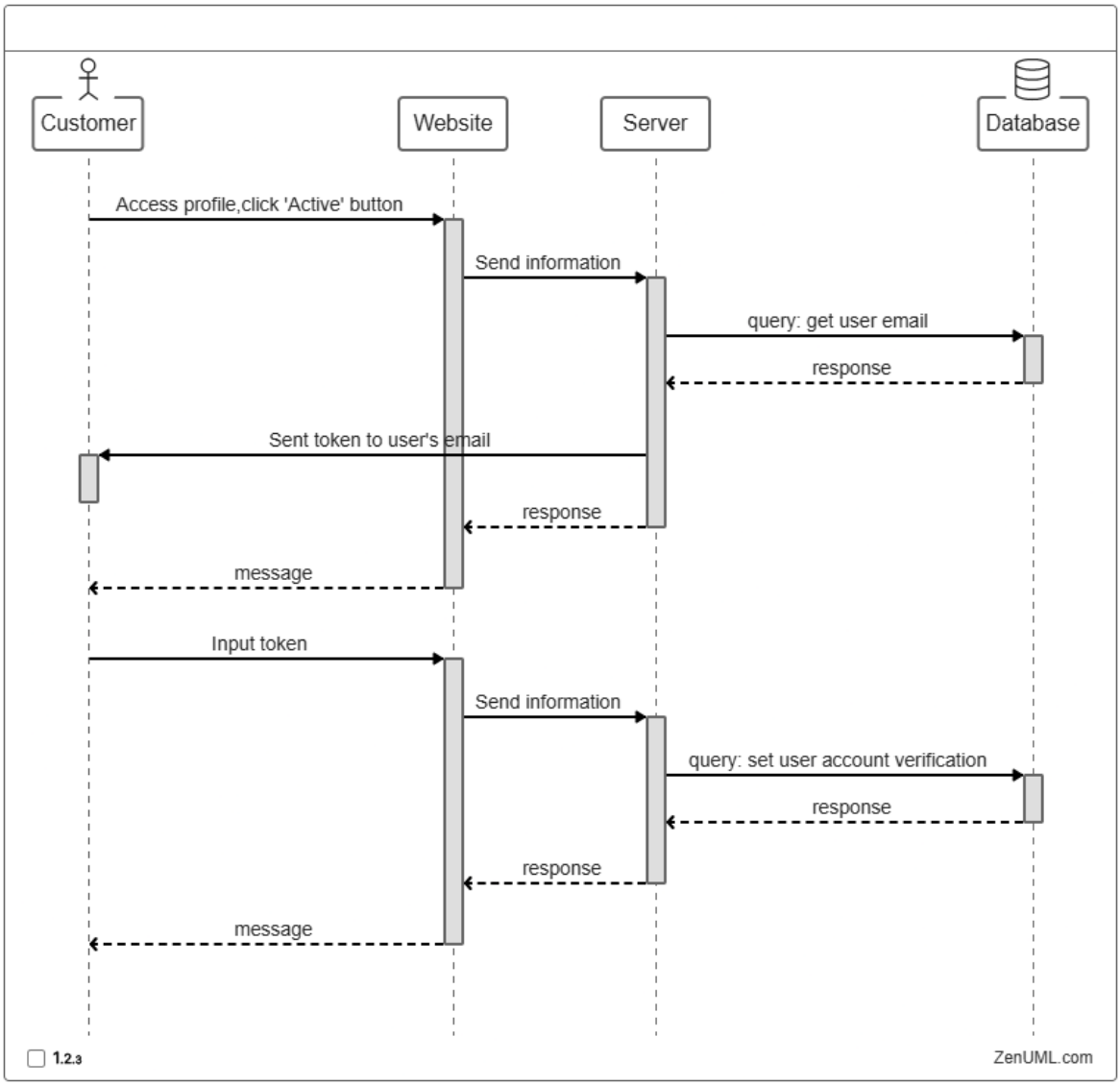


Figure 21: Active email sequence

4.3.11. Admin user management

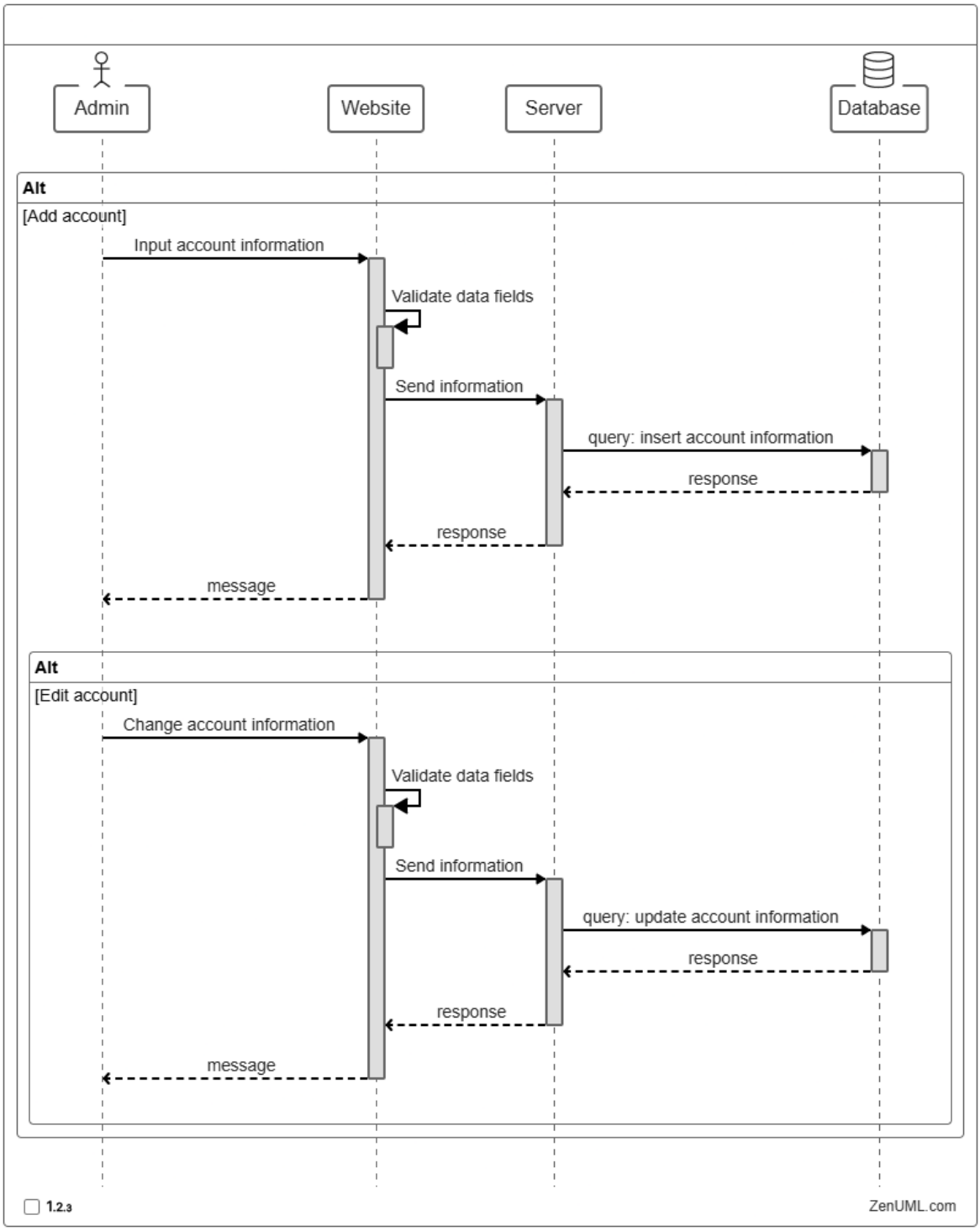


Figure 22: Admin user management sequence

4.3.12. Admin movie management

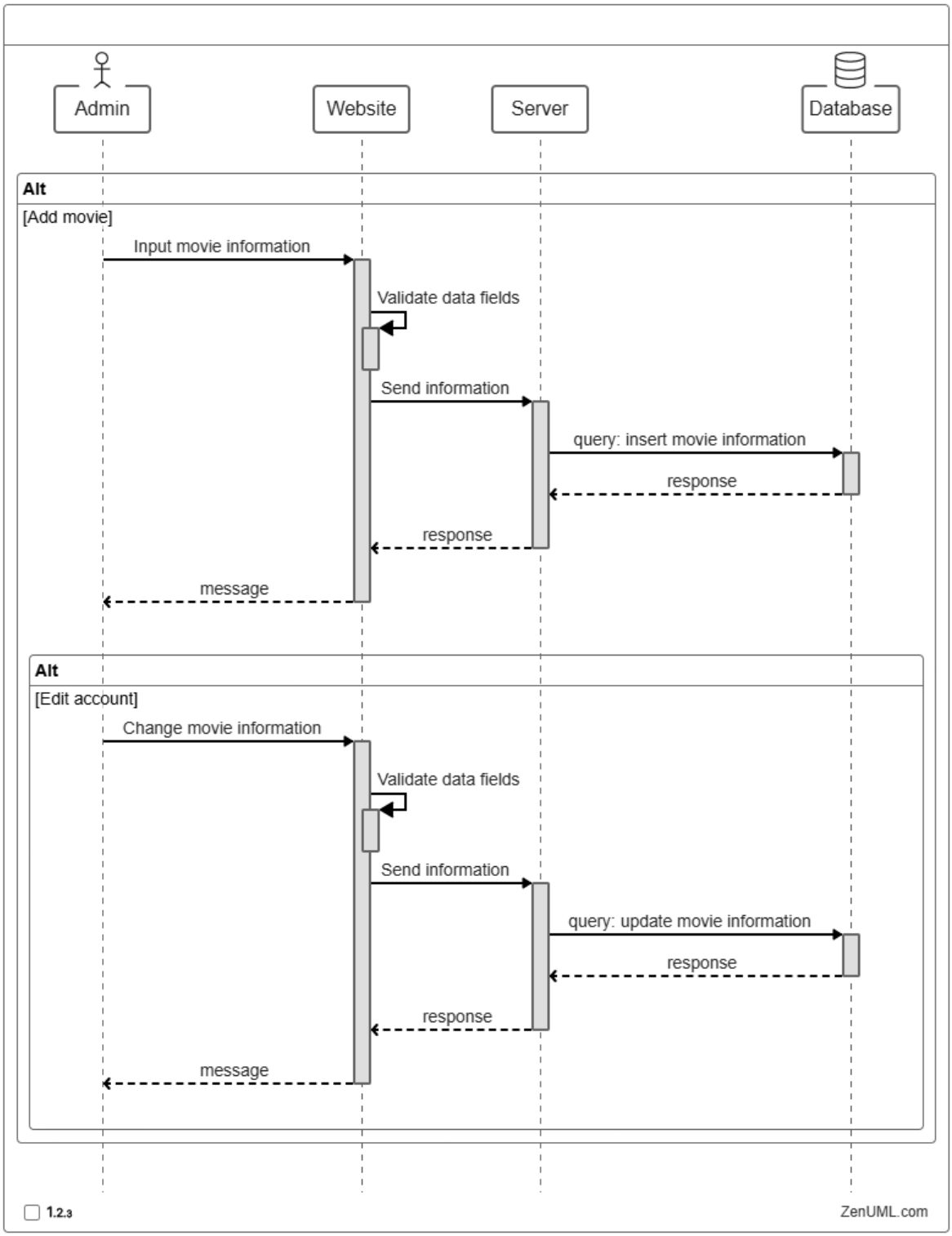


Figure 23: Admin movie management

4.3.13. Admin episode management

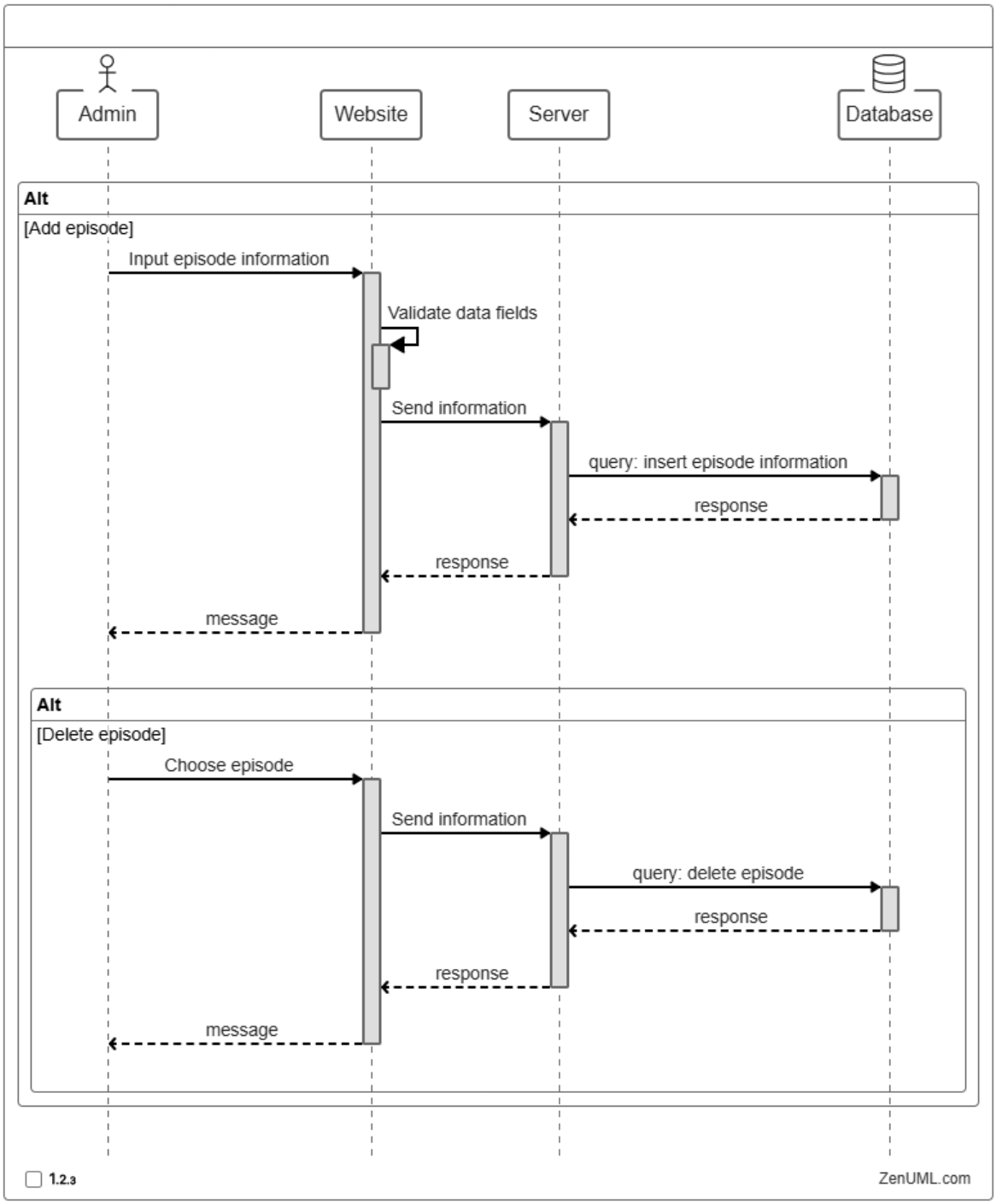


Figure 24: Admin episode management

4.3.14. Admin categories management

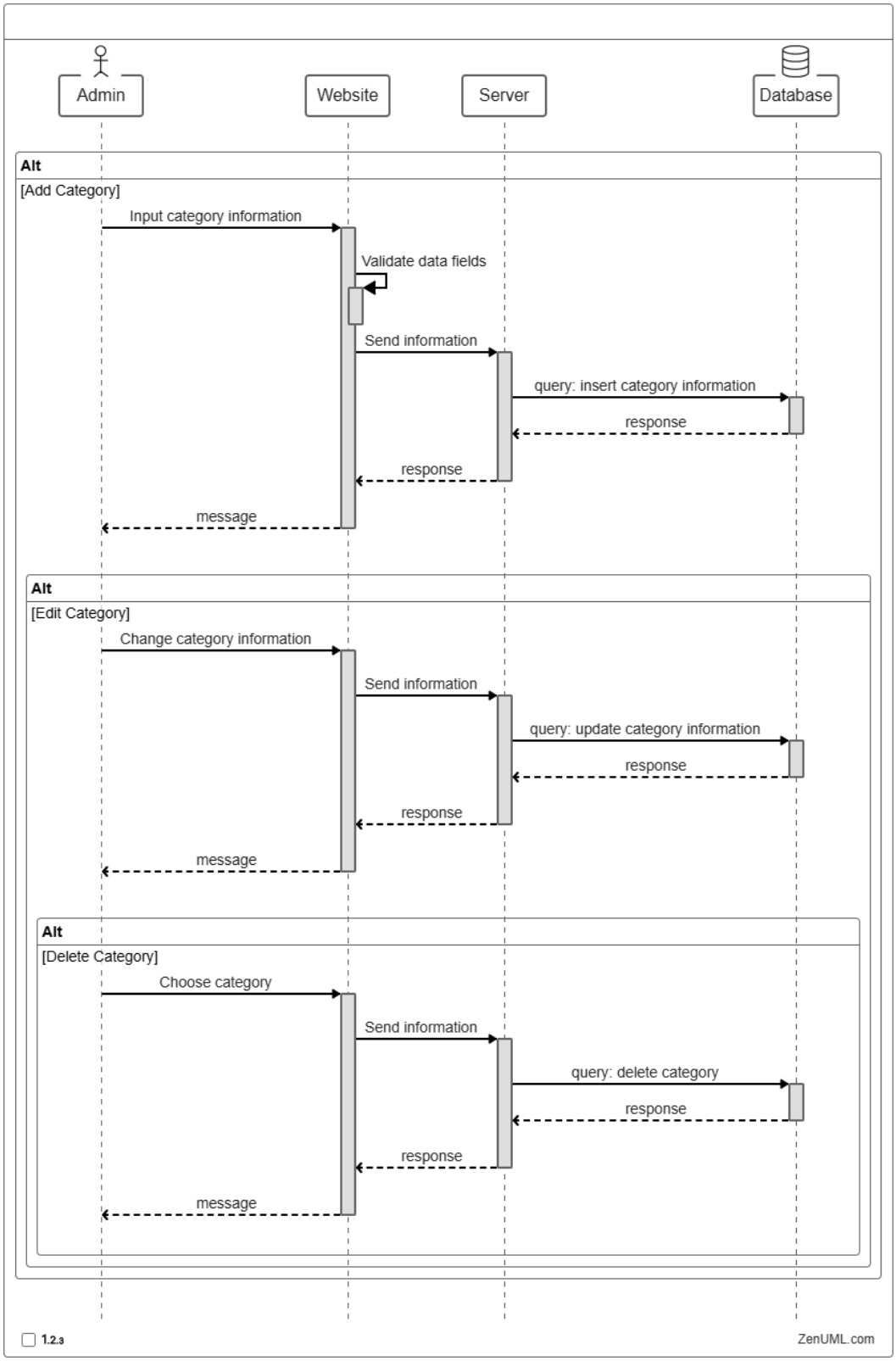


Figure 25: Admin categories management sequence

4.4. Class Diagram

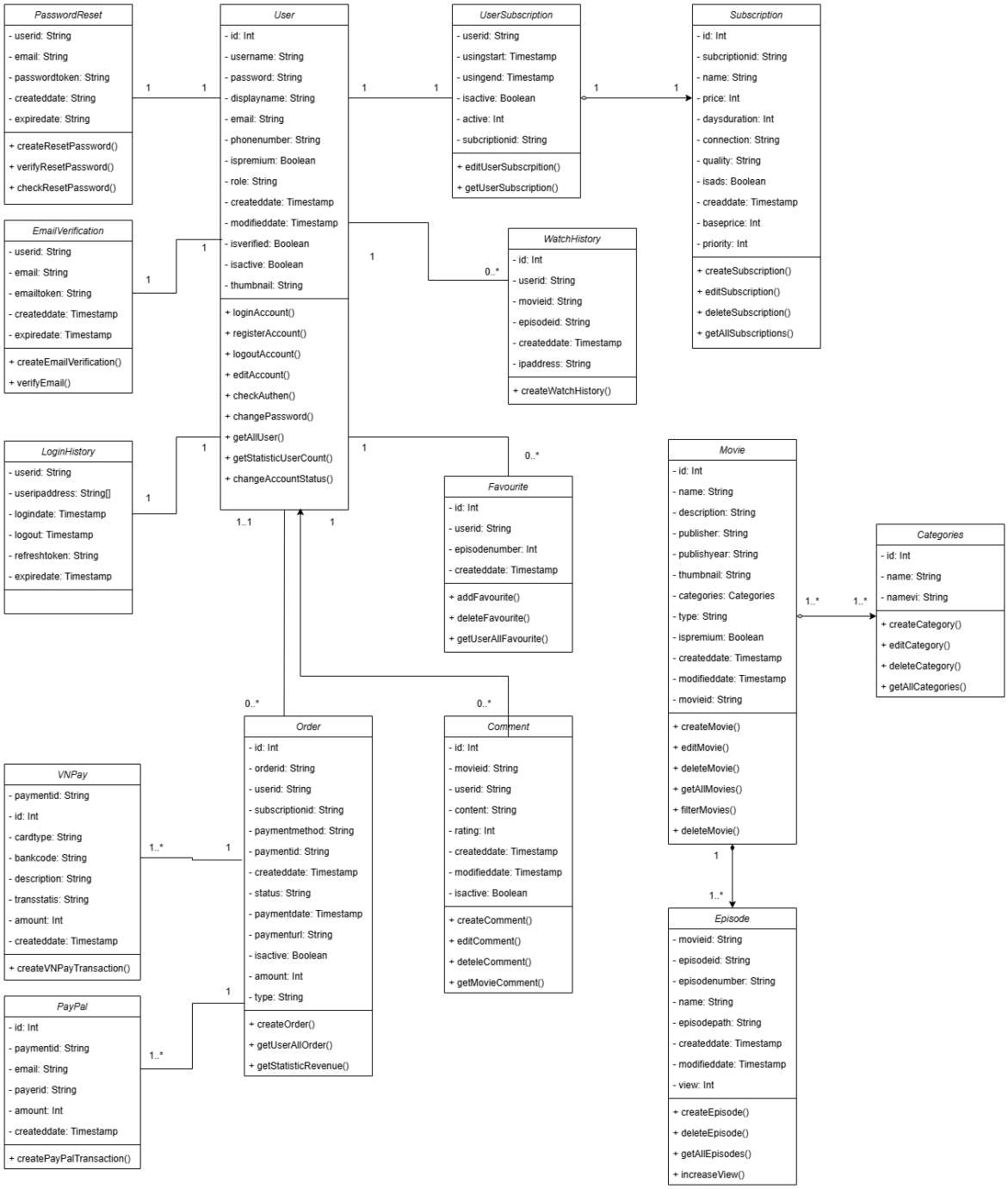


Figure 26: Class Diagram

4.5. Database

4.5.1. ERD

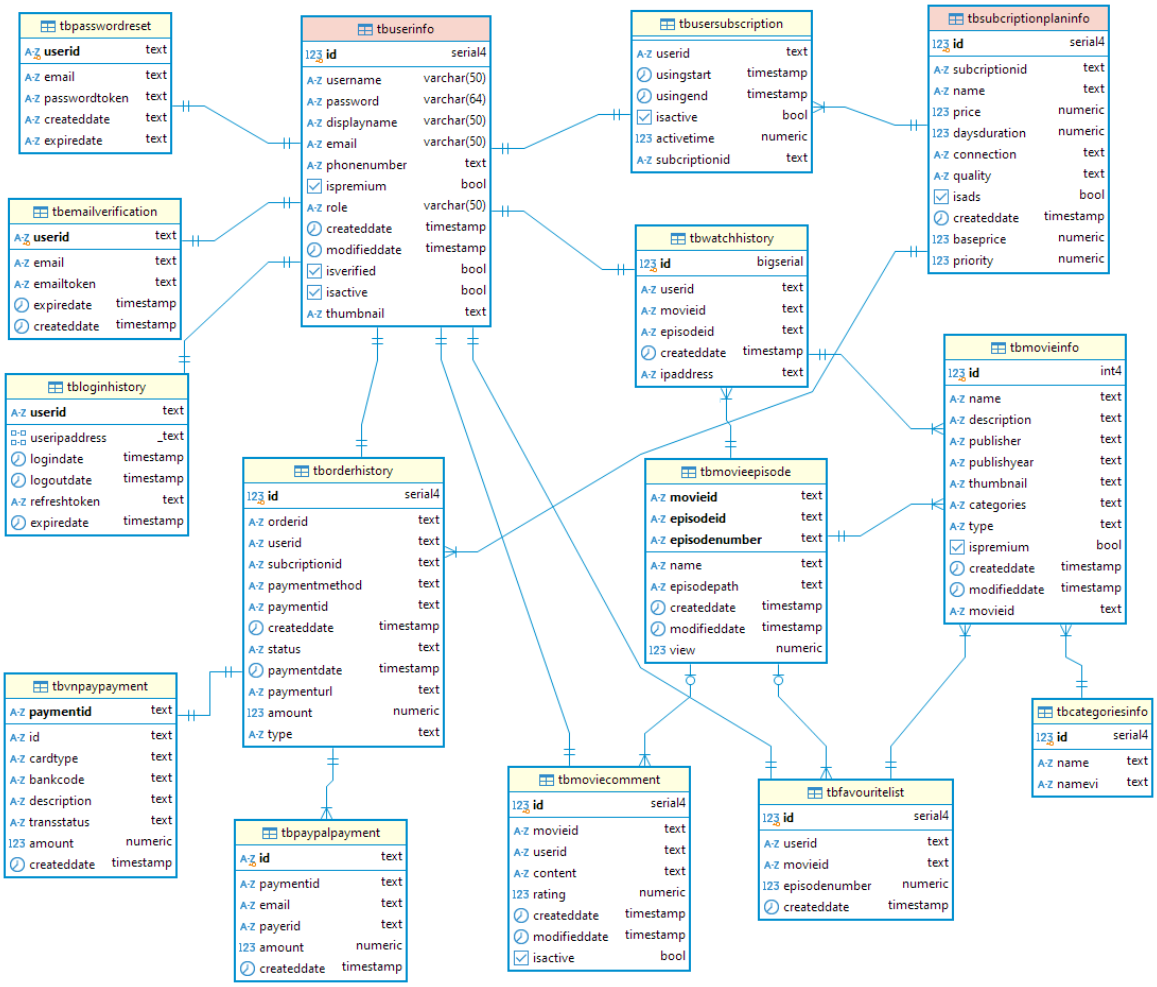


Figure 27: ERD

4.5.2. Database table details

4.5.3.1. tbuserinfo table

Ord	Attribute	Type	Domain	Meaning	Note
1	id	serial4	Primary key	Unique identifier for the user account.	
3	username	varchar(50)		Username of the account.	
4	password	varchar(64)		Encrypted password of the account.	
5	displayname	varchar(50)		Display name visible to other users.	
6	email	varchar(50)		Email address associated with the account.	
7	phonenumber	text		Contact phone number of the user.	
8	ispremium	bool		Indicates whether the user has a premium subscription (true/false).	
9	role	varchar(50)		Role of the user (e.g., admin, moderator, user).	
10	createddate	timestamp		Date and time the account was created.	
14	modifieddate	timestamp		Date and time the account was last modified.	

15	isverified	bool		Indicates whether the account is verified (true/false).	
16	isactive	bool		Indicates whether the account is active (true/false).	
17	thumbnail	text		Avatar of account	

Table 35: tbuserinfo table

4.5.2.2. tbcategoriesinfo table

Ord	Attribute	Type	Domain	Meaning	Note
1	id	serial4	Primary key	Unique identifier for the record.	
2	name	text		Name of the entity in English or general format.	
3	namevi	text		Name of the entity in Vietnamese.	

Table 36: tbcategoriesinfo table

4.5.2.3. tbemailverification table

Ord	Attribute	Type	Domain	Meaning	Note
1	userid	text	Primary key	Unique identifier for the user.	
2	email	text		Email address associated with the user account.	
3	emailtoken	text		Token used for email verification purposes.	
4	expiredate	timestamp		Expiration date and time for the email token.	
5	createddate	timestamp		Date and time the record was created.	

Table 37: tbemailverification table

4.5.2.4. tbfavouritelist table

Ord	Attribute	Type	Domain	Meaning	Note
1	id	serial4	Primary key	Unique identifier for the favorite list entry.	
2	userid	text		Unique identifier for the user.	
3	movieid	text		Unique identifier for the movie.	
4	episodenumber	text		Episode number associated with the movie (if applicable).	
5	createddate	timestamp		Date and time the record was created.	

Table 38: tbfavouritelist table

4.5.2.5. tbloginhistory table

Ord	Attribute	Type	Domain	Meaning	Note
1	userid	text	Primary key	Unique identifier for the user.	
2	useripaddress	_text		List of IP addresses used by the user.	
3	logindate	timestamp		Date and time the user logged in.	
4	logoutdate	timestamp		Date and time the user logged out.	
5	refreshtoken	text		Token used to refresh the user session.	
6	expiredate	timestamp		Expiration date and time for the refresh token.	

Table 39: tbloginhistory table

4.5.2.6. tbmoviecomment table

Ord	Attribute	Type	Domain	Meaning	Note
1	id	serial4	Primary key	Unique identifier for the movie comment.	
2	movieid	text		Unique identifier for the movie being commented on.	
3	userid	text		Unique identifier for the user who made the comment.	
4	content	text		Text content of the user's comment.	
5	rating	numeric		Numeric rating given by the user (e.g., 1-5 scale).	
6	createddate	timestamp		Date and time the comment was created.	
7	modifieddate	timestamp		Date and time the comment was last modified.	
8	isactive	bool		Indicates whether the comment is active (true/false).	

Table 40: tbmoviecomment table

4.5.2.7. tbmovieepisode table

Ord	Attribute	Type	Domain	Meaning	Note
1	movieid	text	Primary key	Unique identifier for the movie.	
2	episodeid	text		Unique identifier for the episode.	
3	name	text		Name of the episode.	
4	episodenumber	text		Number of the episode within the movie.	
5	episodepath	text		Path or URL to access the episode content.	
6	createddate	timestamp		Date and time the episode was created.	
7	modifieddate	timestamp		Date and time the episode was last modified.	
8	view	numeric		View count of the episode.	

Table 41: tbmovieepisode table

4.5.2.8. tbmovieinfo table

Ord	Attribute	Type	Domain	Meaning	Note
1	id	int4	Primary key	Unique identifier for the movie information.	
2	name	text		Name of the movie.	
3	description	text		Description or synopsis of the movie.	
4	publisher	text		Name of the publisher or production company.	
5	publishyear	text		Year the movie was published.	
6	thumbnail	text		URL or path to the thumbnail image of the movie.	
7	categories	text		Categories or genres the movie belongs to.	
8	type	text		Type of movie (e.g., series, feature film).	
9	ispremium	bool		Indicates if the movie requires a premium subscription (true/false).	
11	createddate	timestamp		Date and time the movie record was created.	

12	modifieddate	timestamp		Date and time the movie record was last modified.	
13	movieid	text		Unique identifier for the movie.	

Table 42: tbmovieinfo table

4.5.2.9. tborderhistory table

Ord	Attribute	Type	Domain	Meaning	Note
1	id	serial4	Primary key	Unique identifier for the payment record.	
2	orderid	text		Unique identifier for the order associated with the payment.	
3	userid	text		Unique identifier for the user making the payment.	
4	subcriptionid	text		Unique identifier for the subscription tied to the payment.	
6	paymentmethod	text		Method of payment used (e.g., credit card, PayPal).	
7	paymentid	text		Unique identifier for the payment transaction.	
8	createddate	timestamp		Date and time the payment record was created.	
9	status	text		Status of the payment (e.g., pending, completed).	
10	paymentdate	timestamp		Date and time the payment was made.	

11	paymenturl	text		URL for the payment gateway or receipt.	
12	amount	numeric		Total amount of the payment.	
13	type	text		Type of payment (e.g., subscription, one-time)	

Table 43: tborderhistory table

4.5.2.10. tbpasswordreset table

Ord	Attribute	Type	Domain	Meaning	Note
1	userid	text	Primary key	Unique identifier for the user.	
2	email	text		Email address associated with the user account.	
3	passwordtoken	text		Token used for resetting the user's password.	
4	createddate	timestamp		Date and time the password token was created.	
5	expiredate	timestamp		Expiration date and time for the password token.	

Table 44: tbpasswordreset table

4.5.2.11. tbpaypalpayment table

Ord	Attribute	Type	Domain	Meaning	Note
1	id	text	Primary key	Unique identifier for the payment record.	
2	paymentid	text		Unique identifier for the payment transaction.	
3	email	text		Email address associated with the payer.	
4	payerid	text		Unique identifier for the payer.	
5	amount	numeric		Amount paid in the transaction.	
6	createddate	timestamp		Date and time the payment record was created.	

Table 45: tbpaypalpayment table

4.5.2.12. tbsubcriptionplaninfo table

Ord	Attribute	Type	Domain	Meaning	Note
1	id	serial4	Primary key	Unique identifier for the subscription plan.	
2	subcriptionid	text		Unique identifier for the subscription plan.	
3	name	text		Name of the subscription plan.	
4	price	numeric		Price of the subscription plan.	
5	daysduration	numeric		Duration of the subscription plan in days.	
6	connection	text		Number of connections allowed in the plan.	
7	quality	text		Quality of the content in the plan	
9	isads	bool		Indicates whether ads are included in the plan (true/false).	
10	createddate	timestamp		Date and time the subscription plan was created.	

11	baseprice	numeric		Base price of the subscription plan.	
12	priority	numeric		Priority level of the subscription plan (higher value indicates higher priority).	

Table 46: tbsubcriptionplaninfo table

4.5.2.13. tbusersubscription table

Ord	Attribute	Type	Domain	Meaning	Note
1	userid	text		Unique identifier for the user.	
2	usingstart	timestamp		Start date and time when the user started using the service.	
3	usingend	timestamp		End date and time when the user stopped using the service.	
5	isactive	bool		Indicates whether the user's subscription is currently active (true/false).	
6	activetime	numeric		Amount of time the user has been active (in minutes, hours, etc.).	
7	subcriptionid	text		Unique identifier for the subscription the user is using.	

Table 47: tbusersubscription table

4.5.2.14. tbvnpaypayment table

Ord	Attribute	Type	Domain	Meaning	Note
1	id	text	Primary key	Unique identifier for the payment transaction.	
2	paymentid	text		Unique identifier for the payment.	
3	cardtype	text		Type of card used for the payment (e.g., credit, debit).	

4	bankcode	text		Code of the bank where the payment was processed.	
5	description	text		Description or additional details of the payment.	
6	transstatus	text		Status of the transaction (e.g., successful, failed).	
7	amount	numeric		Amount of the payment transaction.	
8	createddate	timestamp		Date and time when the payment was created.	

Table 48: tbvnpaypayment table

4.5.2.15. tbwatchhistory table

Ord	Attribute	Type	Domain	Meaning	Note
1	id	bigserial	Primary key	Unique identifier for the watch history record.	
2	userid	text		Unique identifier for the user who watched the content.	
3	movieid	text		Unique identifier for the movie that was watched.	
4	episodeid	text		Unique identifier for the episode (if applicable) that was watched.	
5	createddate	timestamp		Date and time when the watch history record was created.	
6	ipaddress	text		IP address of the user who watched the content.	

Table 49: tbwatchhistory table

4.6. Application interfaces

4.6.1. Guest

4.6.1.1. Register page

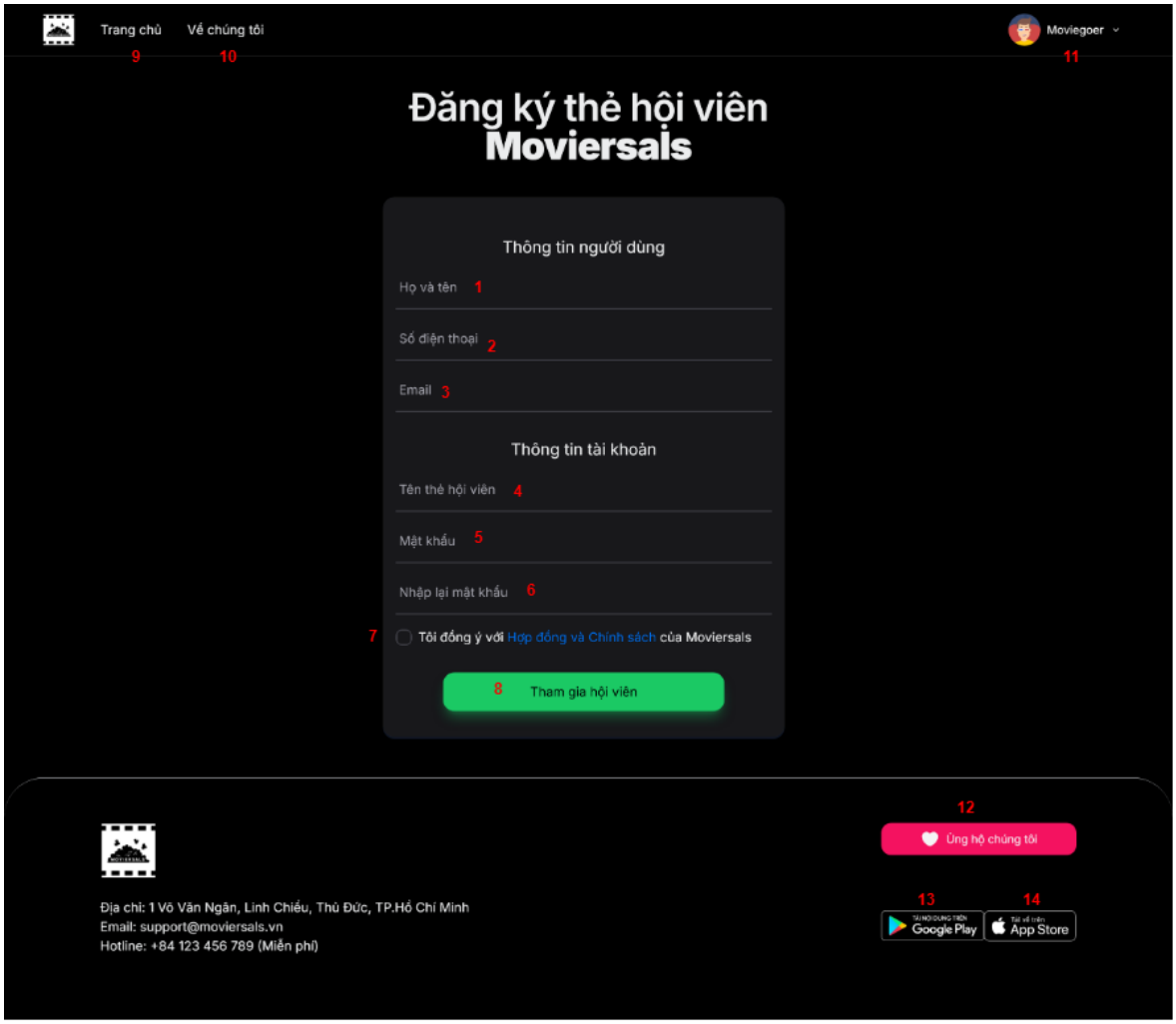


Figure 28: Register Page

Table 50: Register Page Object

No	Name	Format	Action
1	Fullname	Textbox	For guest to input fullname

2	Phone number	Textbox	For guest to input phone number
3	Email	Textbox	For guest to input email
4	Username	Textbox	For guest to input username
5	Password	Textbox	For guest to input password
6	Repeat Password	Textbox	For guest to repeat password
7	Confirm Policy	Checkbox	For guest to confirm policy of the website
8	Register	Button	For guest to register account
9	Homepage	Text	For guest to redirect to homepage
10	About us	Text	For guest to redirect to about us page

11	Username	Text	Show current user's fullname
12	Donate	Button	For user to donate
13	Google Play	Button	For user to download android mobile app (not developed yet)
14	Appstore	Button	For user to download ios mobile app (not developed yet)

4.6.1.2. Reset password page

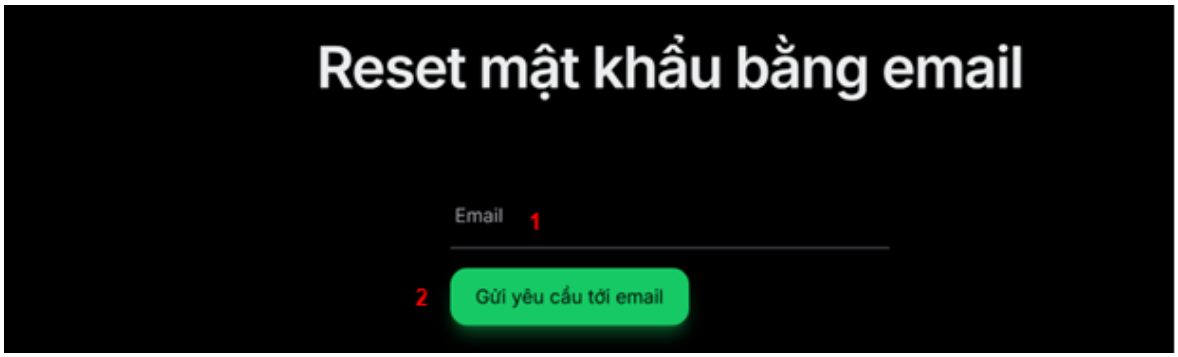


Figure 29: Reset password page

Table 51: Reset Password Page Object

No	Name	Format	Action
1	Email	Textbox	For guest to enter email

2	Send email	Button	For guest to send code to email

4.6.1.3. Login page

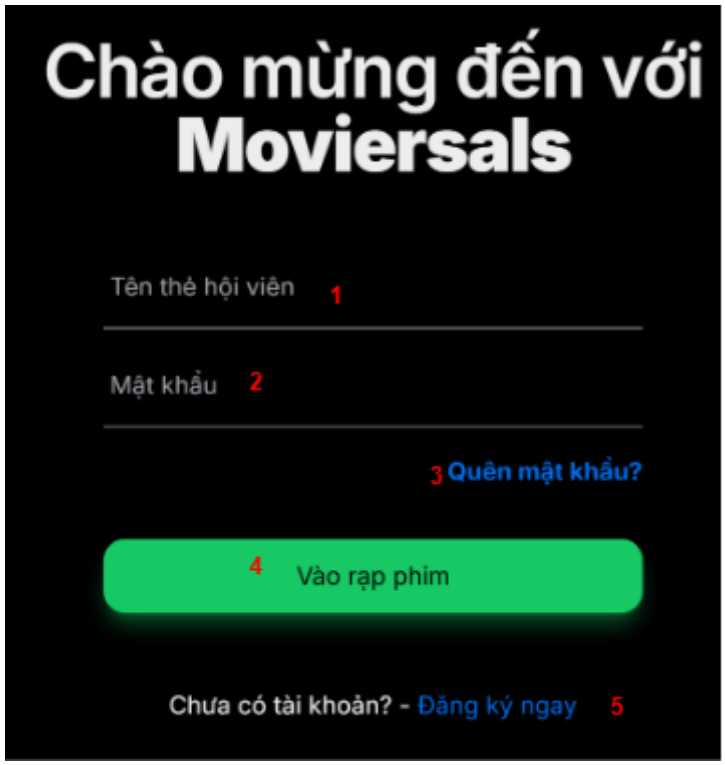


Figure 30: Login Page

Table 52. Login Page Object

No	Name	Format	Action
1	username	Textbox	For guest to input username
2	Password	Textbox	For guest to input password

3	Forgot Password	Text	For guest to redirect to forgot password page
4	Login	Button	For guest to login
5	Register	Text	For guest to redirect to register page

4.6.1.4. Policy page

Figure 31. Policy Page

Table 53. Policy Page Objects

No	Name	Format	Action
1	Home	Button	For customer to return to home page

2	Username	Text	For customers to input username
3	Password	Text	For customers to input password
4	Forgot password	Button	For customer to reset password page
5	Sign in	Button	For customer to send login request to the system
6	Create account	Button	For customer to go to the register page

4.6.2. Customer

4.6.2.1. Customer’s Homepage

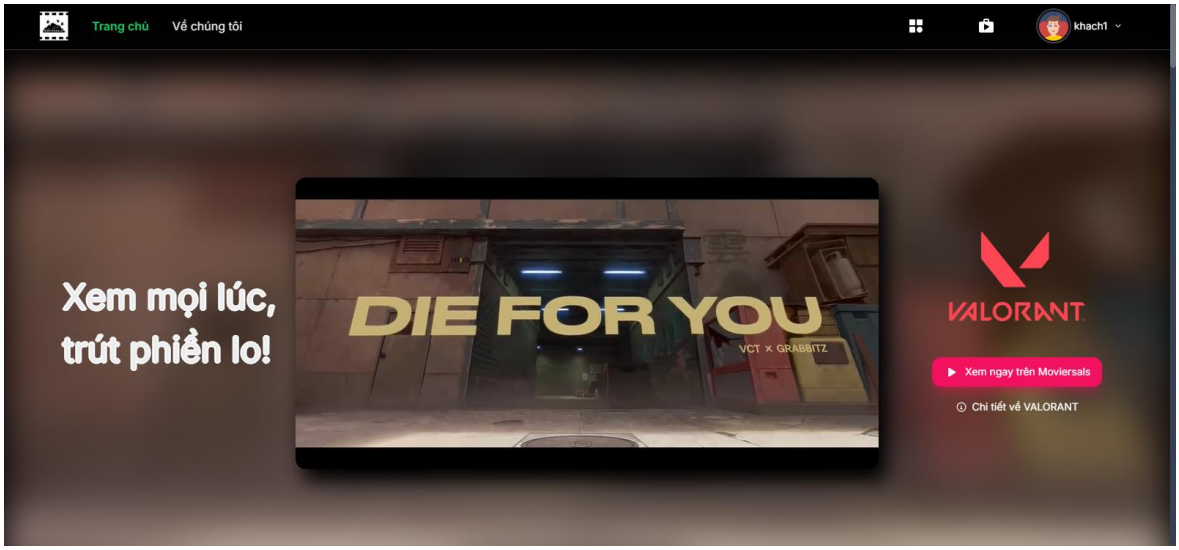


Figure 32: Customer's Homepage

Table 54: Customer's Homepage Object

No	Name	Format	Action
1	Translation	Button	For customer to open translation popup
2	Customer avatar	Image	Display customer avatar image
3	Customer name	Text	Display customer avatar name
4	Profile	Button	For customer to view the customer profile
5	Logout	Button	For customers to send a logout request to the system
6	Shop logo	Image	Display shop logo
7	Search	Text	For customer to input anything for search product
8	Search	Button	For customer to send search request to the system

9	Invoice	Button	For customer to go to the customer invoice page
10	Shop	Button	For customer to go to the customer shop page
11	Categories	Button	Display all category of product
12	Banner	Image	Display shop banner

4.6.2.2. Profile Page

Figure 33. Profile Page

Table 55: Profile Page Object

No	Name	Format	Action
1	User's Username	Textbox	Display user's username
2	User's Subscription	Textbox	Display user's subscription

3	End date	Textbox	Display user’s subscription end date
4	Email Status	Textbox	Display user’s email status
5	Send Code	Button	For user to send request to get email confirmation code
6	User’s Fullname	Textbox	Display user’s fullname
7	User’s Email	Textbox	Display user’s email
8	User’s Phone Number	Textbox	Display user’s phone number
9	Choose file	Button	For user to choose image file
10	Confirm change	Button	For user to confirm profile editing

4.6.2.3. Subscription Page

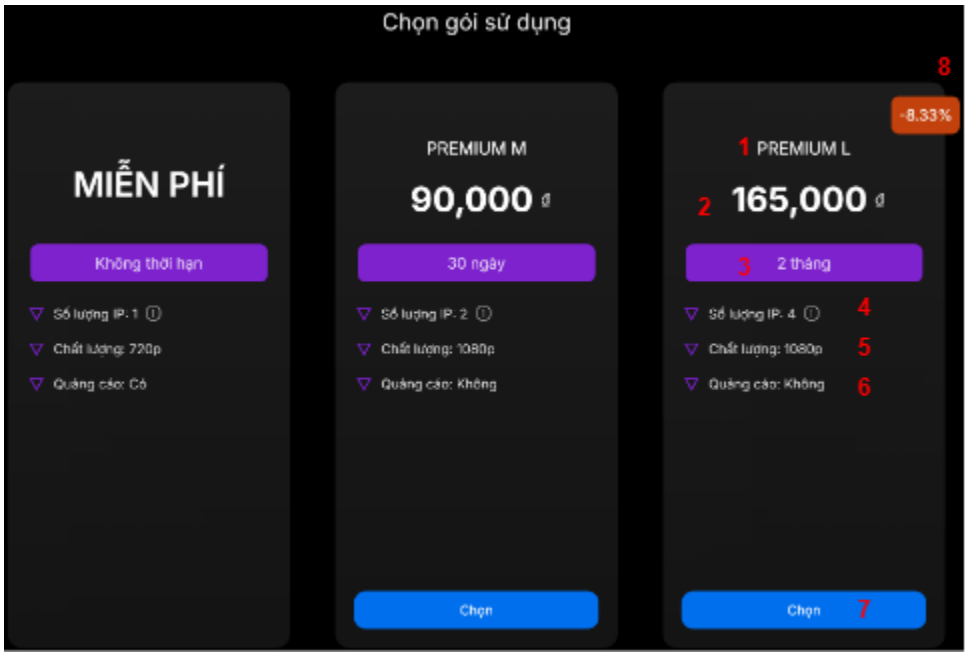


Figure 34: Subscription Page 1

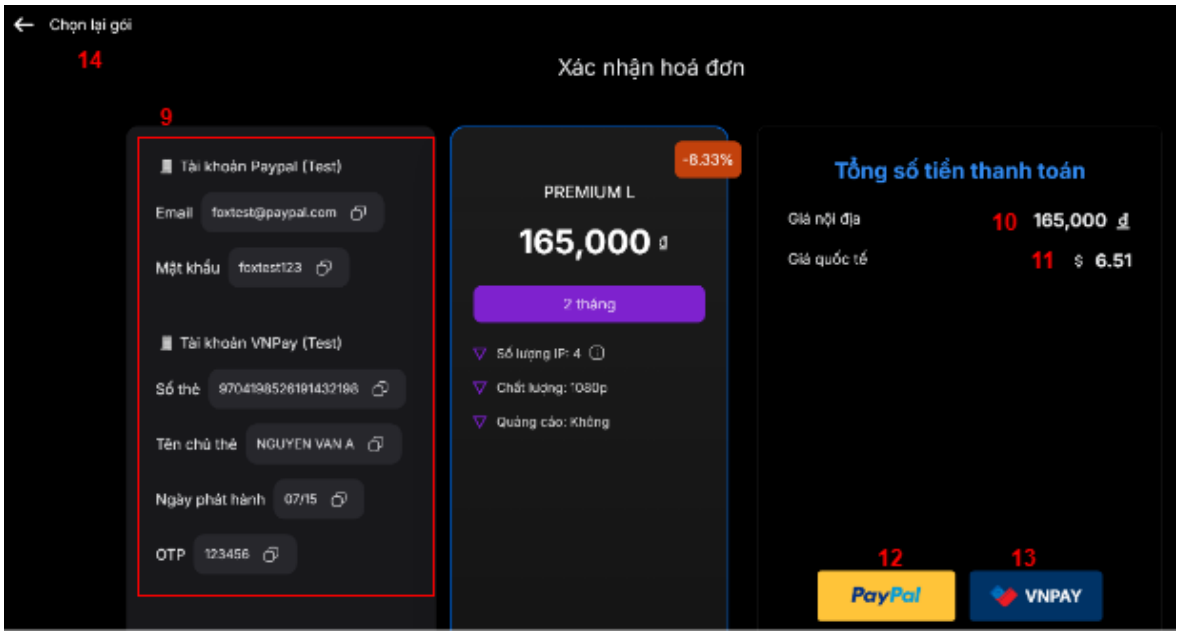


Figure 35: Subscription Page 2

Table 56: Subscription Page Object

No	Name	Format	Action
1	Subscription Name	Text	Display subscription name
2	Subscription Fee	Text	Display subscription fee
3	Subscription Length	Text	Display subscription length
4	Subscription Allow IP Number	Text	Display subscription allow IP number
5	Subscription Max Resolution	Text	Display subscription max resolution
6	Subscription Ad Status	Text	Display subscription ad status
7	Choose Subscription	Button	For user to choose subscription
8	Subscription Sale	Text	Display subscription sale

9	Payment Test Account Information	Multi Textbox	Display payment test account information
10	Subscription Fee (VND)	Text	Display subscription fee in VND
11	Subscription Fee (USD)	Text	Display subscription fee in USD
12	PayPal	Button	For customer to choose PayPal payment method
13	VNPay	Button	For customer to choose VNPay payment method
14	Back To Subscription Selection	Text	For user to back to subscription selection page

4.6.2.5. Order History Page

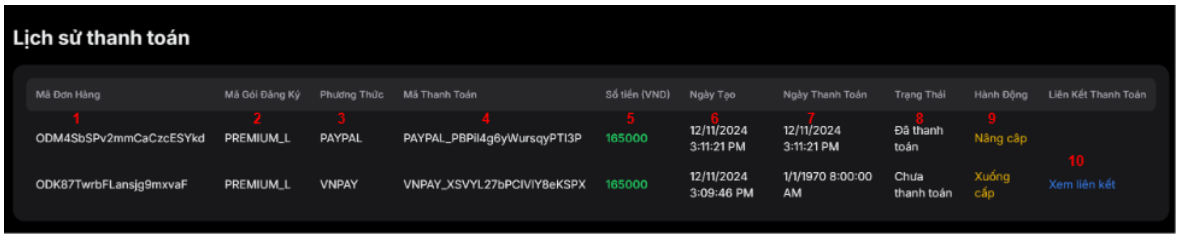


Figure 36: Order History Page

Table 57: Order History Page Objects

No	Name	Format	Action

1	Order ID	Text	Display order id
2	Subscription Name	Text	Display subscription name
3	Payment Method	Text	Display payment method
4	Payment ID	Text	Display payment id
5	Fee	Text	Display fee
6	Created Date	Text	Display created date
7	Paid Date	Text	Display paid date
8	Status	Text	Display status
9	Action	Text	Display action

10	Payment Link	Text	For user to redirect to payment link

4.6.2.6. Categories Page

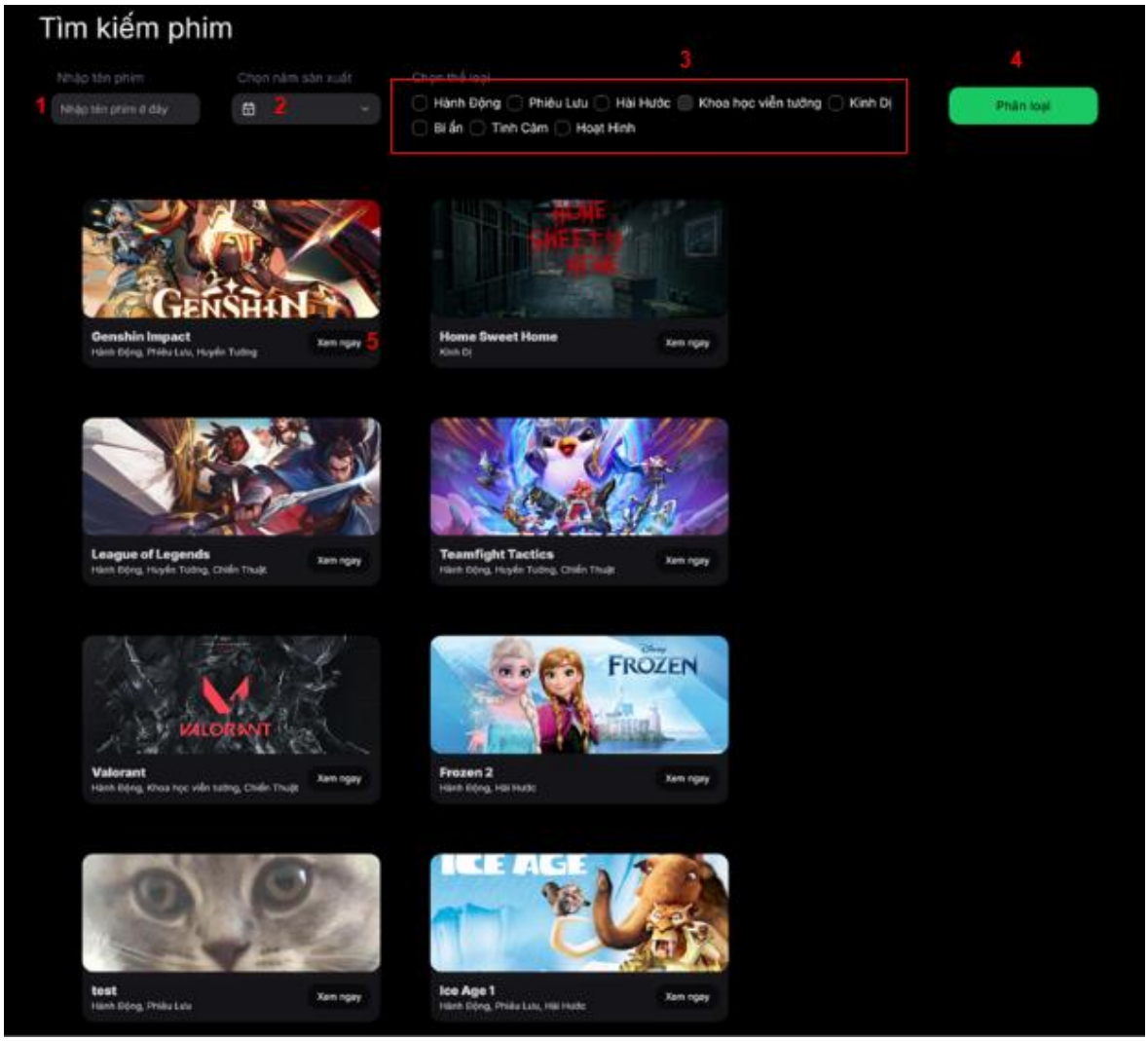


Figure 37: Categories Page

Table 58: Categories Page Objects

No	Name	Format	Action

1	Filter Movie Name	Textbox	For user to input movie name
2	Filter Movie Produced Year	Dropdown	For user to choose movie produced year
3	Filter Movie Categories	Checkbox	For user to choose movie categories
4	Filter	Button	For user to filter movie
5	Watch Now	Button	For user to redirect to movie detail page

4.6.2.7. Movie Detail Page

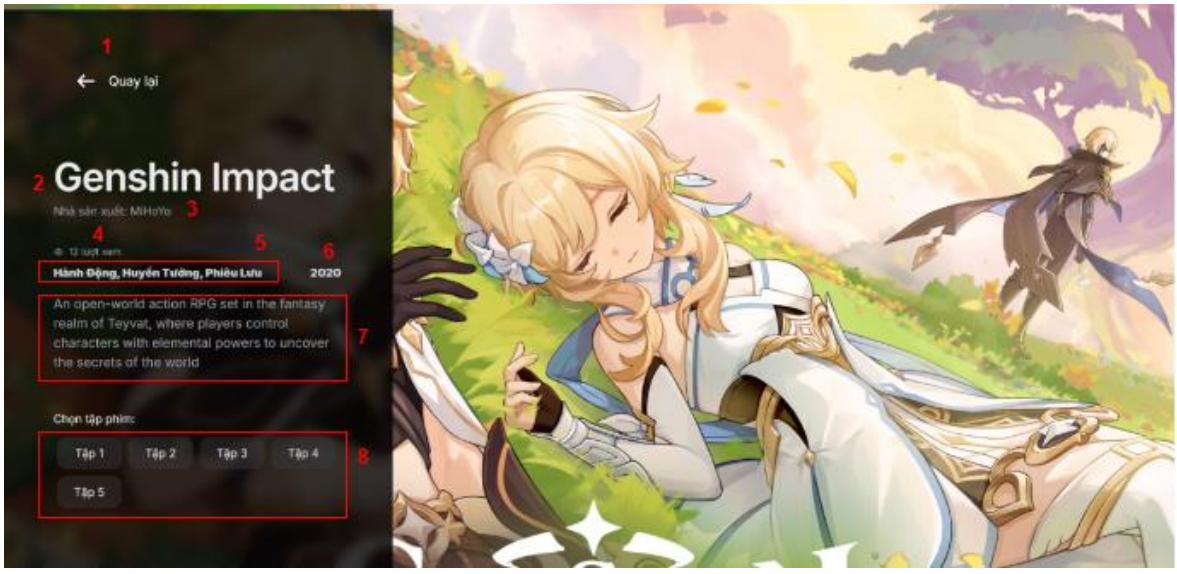


Figure 38: Movie Detail Page

Table 59: Movie Detail Page Objects

No	Name	Format	Action
1	Back	Text	For user to go back to previous page
2	Movie Name	Text	Display movie name
3	Movie Producer	Text	Display movie producer
4	Movie Views	Text	Display movie views
5	Movie Categories	Text	Display movie categories
6	Movie Published Date	Text	Display movie published date
7	Movie Description	Text	Display movie description
8	Movie Episodes	Text	For user to select episode to watch

4.6.2.8. Movie Episode Page

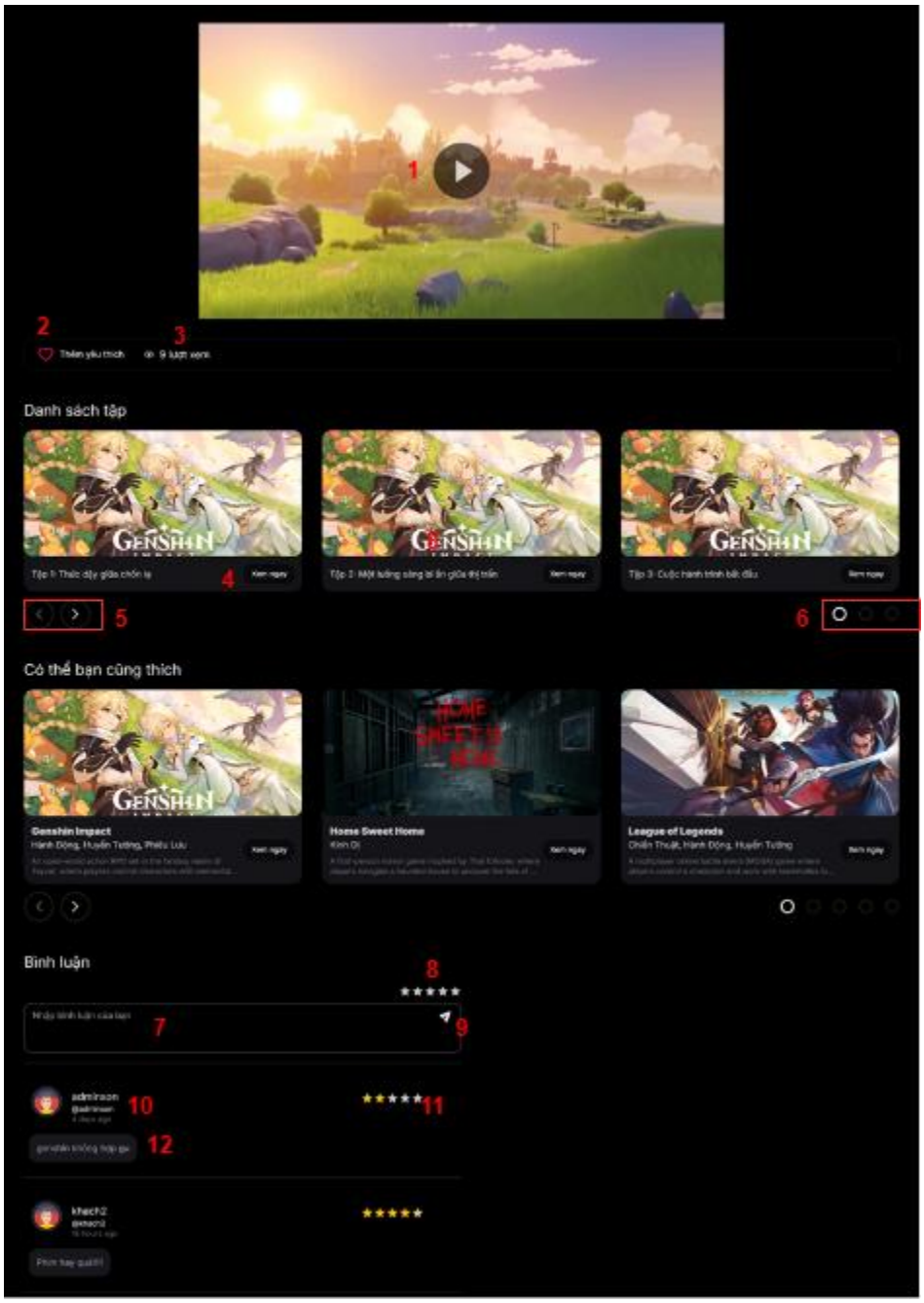


Figure 39: Movie Detail Page

Table 60: Movie Episode Page Objects

No	Name	Format	Action

1	Play	Button	For user to play video
2	Favourite	Button	For user to add movie to favourite
3	Views	Text	Display episode views
4	Watch Now	Button	For user to watch other episode
5	Navigate	Button	For user to scroll episodes
6	Pagination	Button	For user to scroll episodes
7	Comment	Text	For user to input comment
8	Rating	Slider	For user to select rating
9	Submit	Text	For user to submit comment/rating

10	Username	Text	Display username who commented
11	Rating	Slider	Display rating of the comment
12	Comment	Text	Display comment

4.6.2.9. Favourite Page

Figure 40: Favourite Page

Table 61: Favourite Page Objects

No	Name	Format	Action
1	Favourite	Button	For user to remove movie from favourite
2	Movie Name	Text	Display movie name

3	Added Date	Text	Display movie added to favourite date

4.6.3. Admin

4.6.3.1. Admin Page

Figure 41: Admin Page

Table 62: Admin Page Objects

No	Name	Format	Action
1	Manage User	Button	For admin to redirect to admin user page
2	Manage Subscription	Button	For admin to redirect to admin subscription page
3	Manage Movie	Button	For admin to redirect to admin movie page

4	Manage Categories	Button	For admin to redirect to admin categories page
5	View Statistic	Button	For admin to redirect to statistic page

4.6.3.2. Admin User Page

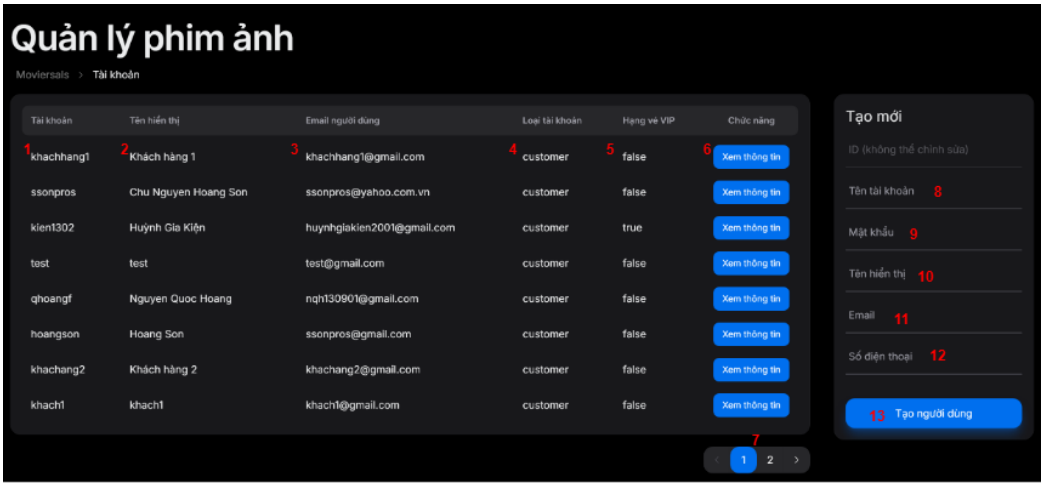


Figure 42: Admin User Page

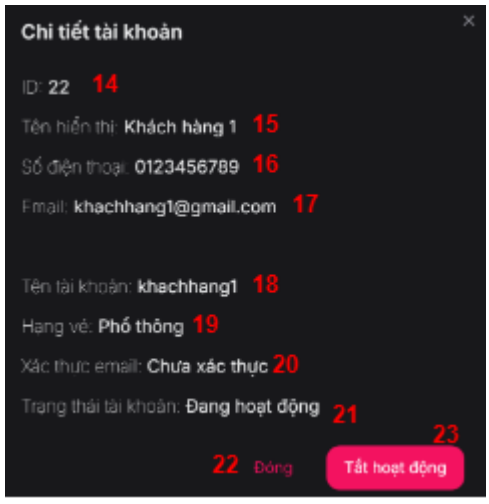


Figure 43: User Detail Panel

Table 63: Admin User Page Objects

No	Name	Format	Action
1	Username	Text	Displays username
2	Fullname	Text	Displays user fullname
3	Email	Text	Displays user email
4	User Type	Text	Displays user type
5	VIP	Text	Displays user VIP status
6	Function	Button	For admin to open user information panel
7	Navigate	Button	For admin to navigate to other pages
8	Username	Textbox	For admin to input new account username

9	Password	Textbox	For admin to input new account password
10	Fullname	Textbox	For admin to input new account fullname
11	Email	Textbox	For admin to input new account email
12	Phone Number	Textbox	For admin to input new account phone number
13	Create User	Button	For admin to create new account
14	User ID	Text	Display user id
15	Fullname	Text	Display user fullname
16	Phone Number	Text	Display user phone number
17	Email	Text	Display user email

18	Username	Text	Display user username
19	Subscription	Text	Display user subscription plan
20	Email Status	Text	Display user email status
21	Status	Text	Display user account status
22	Close	Button	For admin to close panel
23	Deactivate	Button	For admin to deactive user account

4.6.3.3. Admin Subscription Page

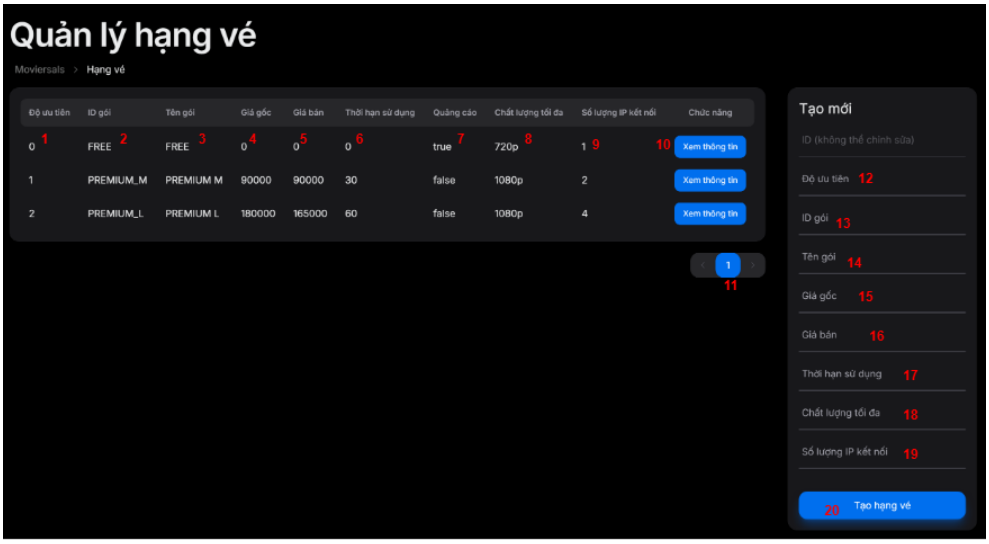


Figure 44: Admin Subscription Page

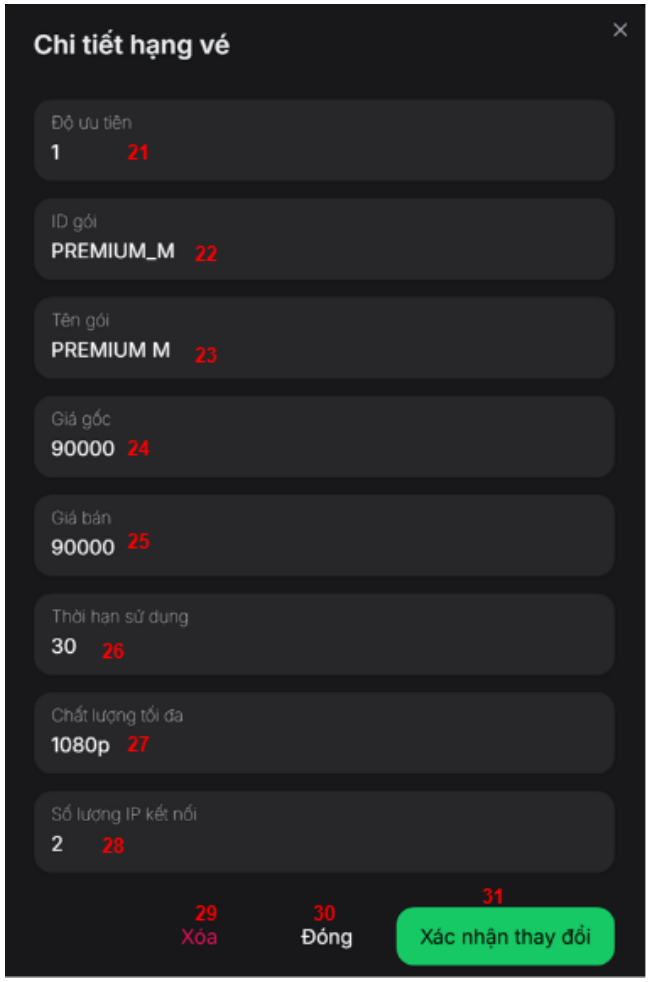


Figure 45: Subscription Detail Panel

Table 64: Admin Subscription Page

No	Name	Format	Action
1	Priority	Text	Display subscription priority
2	Subscription ID	Text	Display subscription id

3	Subscription Name	Text	Display subscription name
4	Base Price	Text	Display subscription base price
5	Sell Price	Text	Display subscription sell price
6	Expiration Date	Text	Display subscription expiration date
7	Ad Status	Text	Display subscription ad status
8	Max Quality	Text	Display subscription max quality
9	Max IP Connection	Text	Display subscription max ip connection
10	View Information	Button	For admin to open subscription information panel
11	Navigate	Button	For admin to navigate to other pages

12	Priority	Textbox	For admin to input new subscription priority
13	Subscription ID	Textbox	For admin to input new subscription priority
14	Subscription Name	Textbox	For admin to input new subscription priority
15	Base Price	Textbox	For admin to input new subscription priority
16	Sell Price	Textbox	For admin to input new subscription priority
17	Expiration Date	Textbox	For admin to input new subscription priority
18	Ad Status	Textbox	For admin to input new subscription priority
19	Max Quality	Textbox	For admin to input new subscription priority
20	Create Subscription	Button	For admin to create new subscription

21	Priority	Text	Display subscription priority
22	Subscription ID	Text	Display subscription id
23	Subscription Name	Text	Display subscription name
24	Base Price	Text	Display subscription base price
25	Sell Price	Text	Display subscription sell price
26	Expiration Date	Text	Display subscription expiration date
27	Max Quality	Text	Display subscription max quality
28	Max IP Connection	Text	Display subscription max ip connection
29	Delete	Button	For admin to delete subscription plan

30	Close	Button	For admin to close panel
31	Confirm Change	Button	For admin to confirm change

4.6.3.3. Admin Movie Page

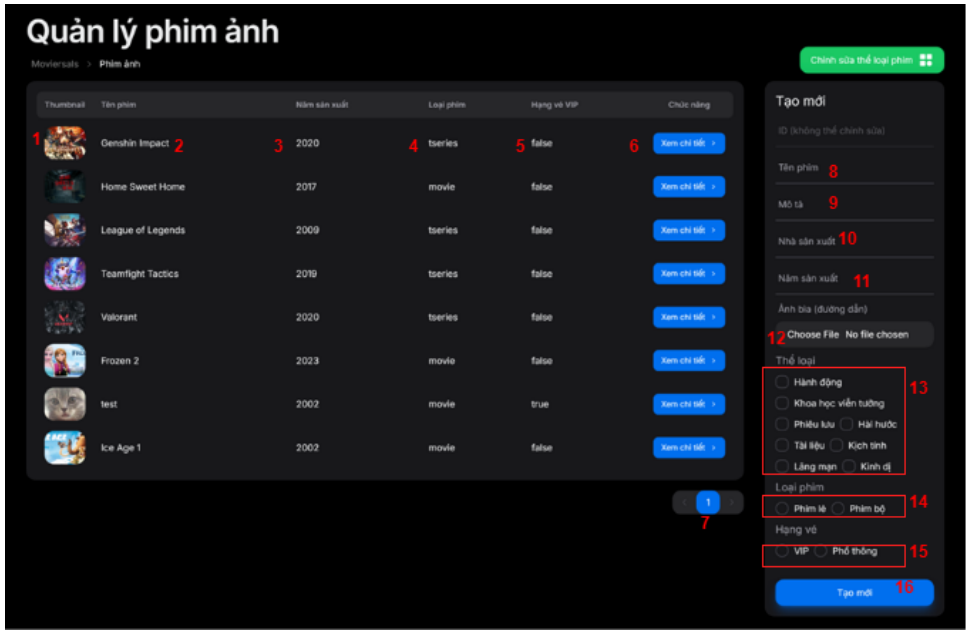


Figure 46: Admin Movie Page

Table 65. Admin Movie Page

No	Name	Format	Action
1	Thumbnail	Image	Display movie thumbnail

2	Movie Name	Text	Displays movie name
3	Published Year	Text	Displays movie published year
4	Movie Type	Text	Displays movie type
5	VIP	Text	Displays movie VIP status
6	View	Text	For admin to redirect to admin episode page
7	Navigate	Text	For admin to navigate to other pages
8	Movie Name	Textbox	For admin to input movie name
9	Description	Textbox	For admin to input movie description
10	Producer	Textbox	For admin to input movie producer

11	Published Year	Textbox	For admin to input movie published year
12	Choose Thumbnail	Button	For admin to choose movie thumbnail
13	Categories	Checkbox	For admin to choose movie categories
14	Movie Type	Radio Button	For admin to choose movie type
15	Subscription	Radio Button	For admin to input movie description
16	Create New	Button	For admin to create new movie
17	Manage Categories	Button	For admin to redirect to admin categories page

4.6.3.4. Admin Movie Detail Page

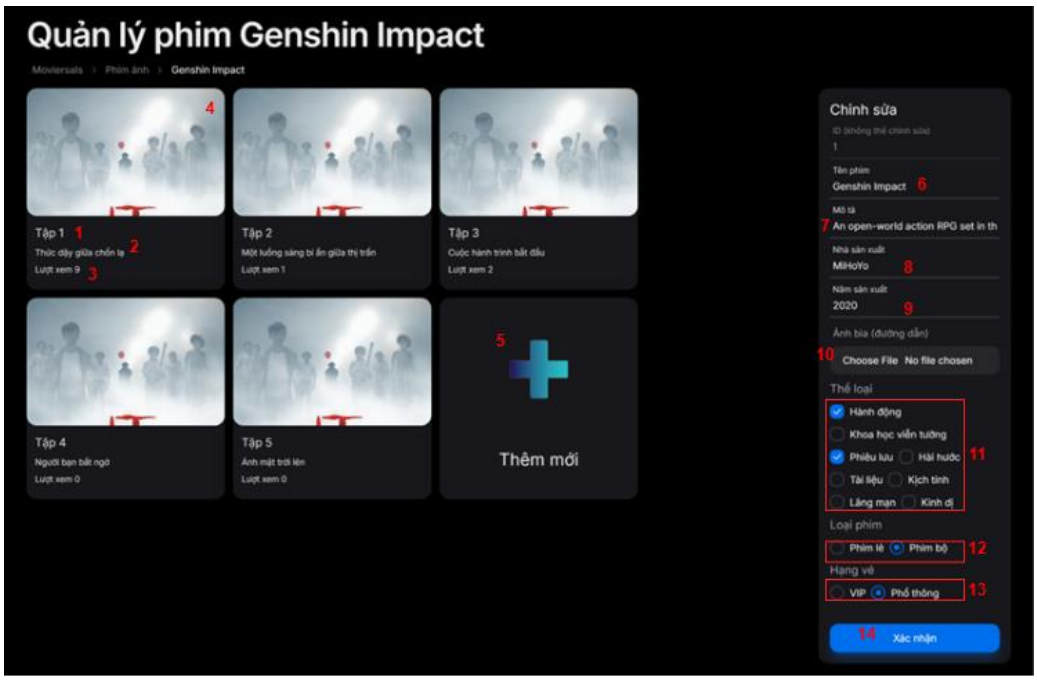


Figure 47: Admin Movie Detail Page

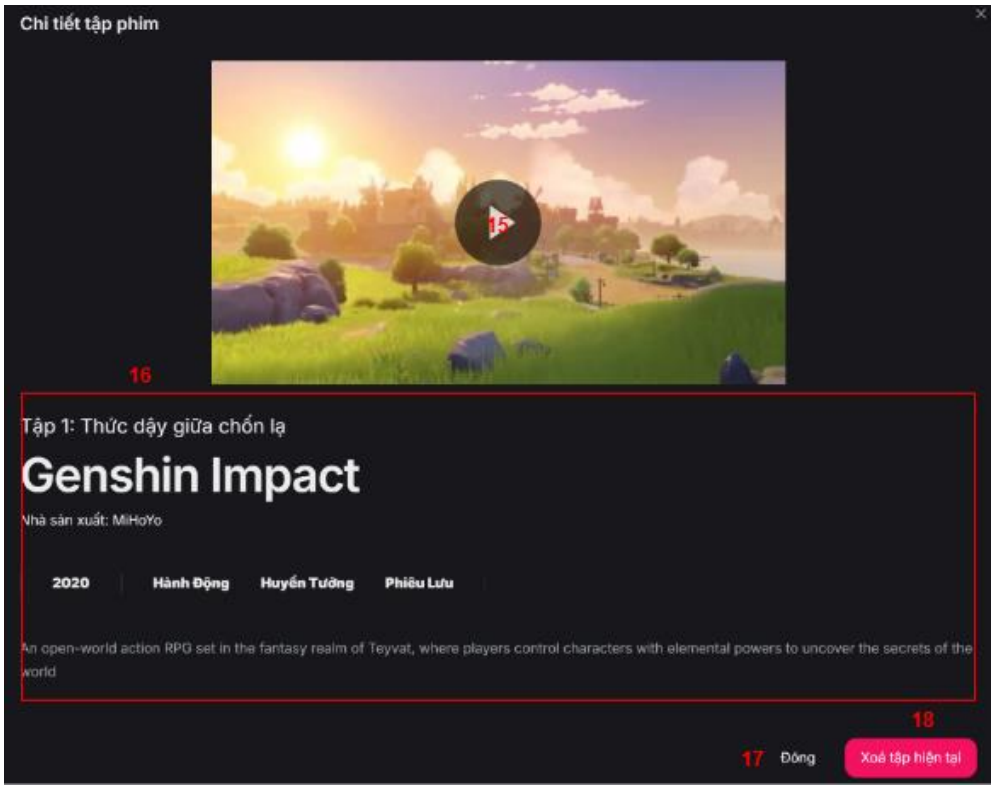


Figure 48: Admin Episode Detail Panel

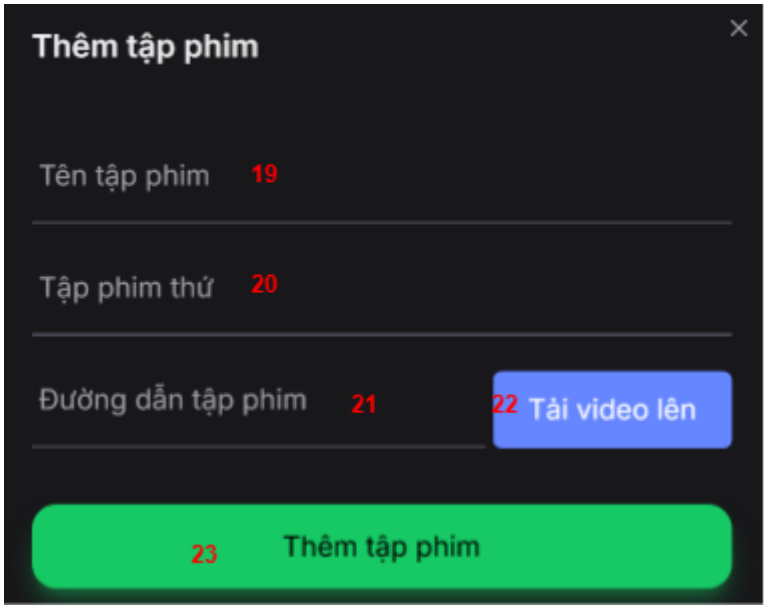


Figure 49: Admin Add Episode Panel

Table 66: Admin Movie Detail Page

No	Name	Format	Action
1	Episode	Text	Display episode
2	Episode Name	Text	Display episode name
3	Episode Views	Text	Display episode views
4	Episode Thumbnail	Image	Display episode thumbnail and for admin to open episode detail panel

5	Add Episode	Button	For admin to open add episode panel
6	Movie Name	Textbox	For admin to input movie name
7	Description	Textbox	For admin to input movie description
8	Producer	Textbox	For admin to input movie producer
9	Published Year	Textbox	For admin to input movie published year
10	Choose Thumbnail	Button	For admin to choose movie thumbnail
11	Categories	Checkbox	For admin to choose movie categories
12	Movie Type	Radio Button	For admin to choose movie type
13	Subscription	Radio Button	For admin to input movie description

14	Confirm	Button	For admin to edit movie information
15	Play	Button	For admin to play episode
16	Episode Information	Text	Display episode information
17	Close	Button	For admin to close admin episode detail panel
18	Delete Episode	Button	For admin to delete episode
19	Episode Name	Textbox	For admin to input episode name
20	Episode Number	Textbox	For admin to input episode number
21	Episode Path	Textbox	For admin to input episode path
22	Upload Episode	Button	For admin to upload episode

23	Add Episode	Button	For admin to add episode

4.6.3.5. Admin Categories Page

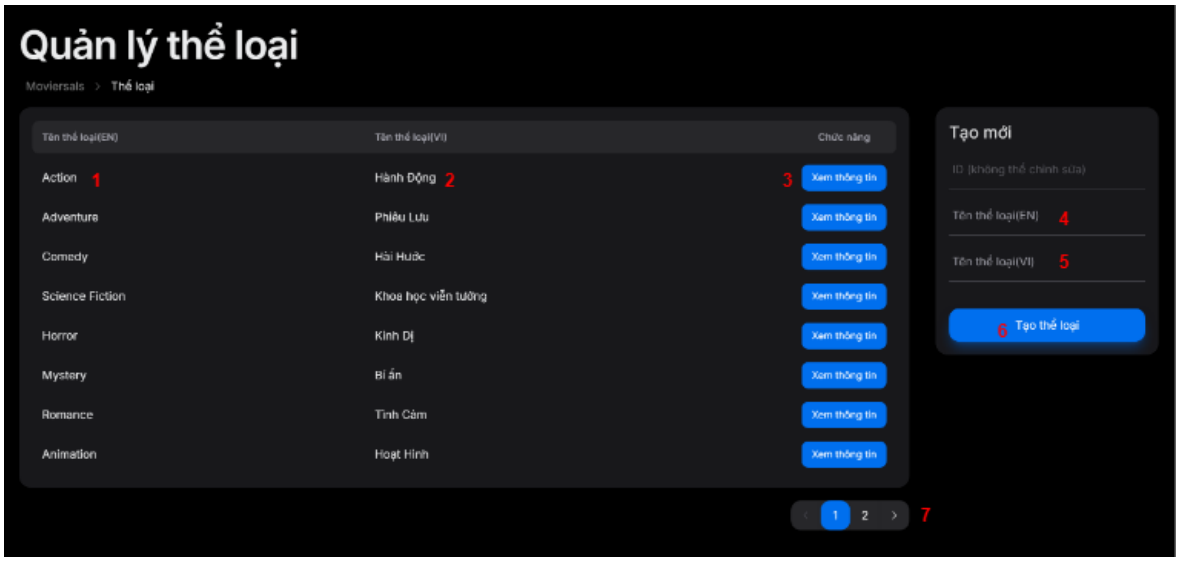


Figure 50: Admin Categories Page

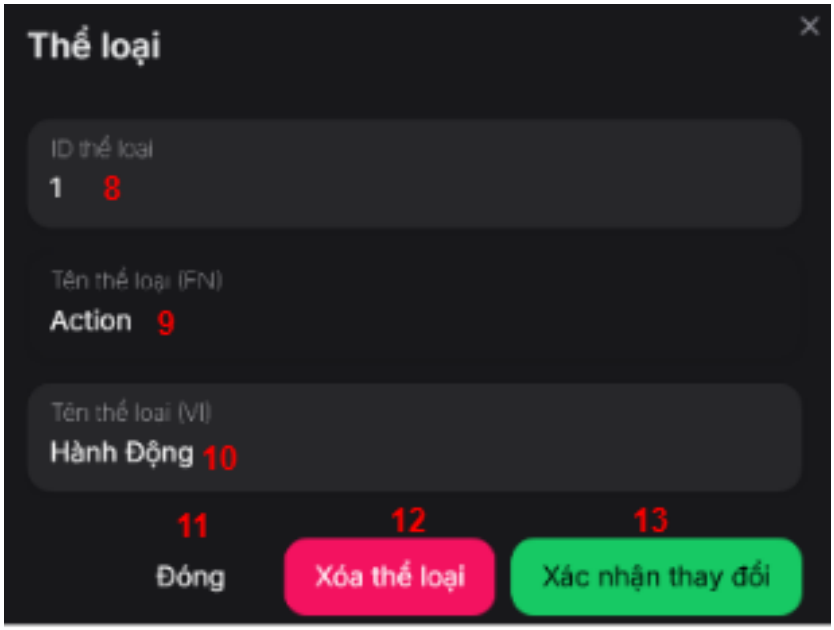


Figure 51: Admin Category Detail Panel

Table 67: Admin Categories Page Objects

No	Name	Format	Action
1	Category name (ENG)	Text	Displays images of the product
2	Category name (VI)	Text	Displays name of the product
3	View Information	Button	Open admin category detail panel
4	Category name (ENG)	Textbox	For user to input new category name (ENG)
5	Category name (VI)	Textbox	For user to input new category name (VI)
6	Create Category	Button	For admin to create new category
7	Navigate	Button	For admin to navigate to other pages
8	Category ID	Textbox	Display category id

9	Category Name (ENG)	Text	Display category name (ENG)
10	Category Name (VI)	Textbox	For admin to edit category name (VI)
11	Close	Button	For admin to close categories detail panel
12	Delete Category	Button	For admin to delete categories
13	Confirm Change	Button	For admin to confirm category change

4.6.3.6. Admin Statistic Page

Figure 52: Admin Statistic Page

Table 68: Admin Statistic Page Objects

No	Name	Format	Action
1	Start Date	Datetime Picker	For admin to select start date to filter
2	End Date	Datetime Picker	For admin to select end date to filter
3	View By Filtered Date	Button	For admin to filter revenue chart by start and end date
4	Revenue Chart	Chart	Display total
5	Total Revenue	Text	Display total revenue
6	Total Customer	Text	Display total customer
7	Total Order	Text	Display total order
8	Views Chart	Chart	Display views chart

CHAPTER 05: IMPLEMENTATION AND TESTING

5.1. Implementation

5.1.1. Environment

First, our backend is development by PostgreSQL database, we use PostgreSQL to store data.

Figure 53: PostgreSQL Database

Also, using Supabase which is similar to PostgresSQL to store data for deployed production website.

Figure 54: Supabase Service

And use Cloudinary to store all about thumbnails and movies

Figure 55. Cloudinary

After selecting the database for our project we choose VSCode is an extremely popular and very positive IDE for developers to create projects because of the convenient of VSCode which can add extensions of the programming language to help them run on IDE easily. Moreover, VSCode has the extension visual studio live

share extension can let we join the project together on the host IDE and we can code or debug together, we can log in that extension via GitHub.

Figure 56: Code Environment

5.1.2. Technology

First, the base technology we use to code around our project is ReactJS, that base on JavaScript programming language using Functional Component to create function for our project instead of Class Component.

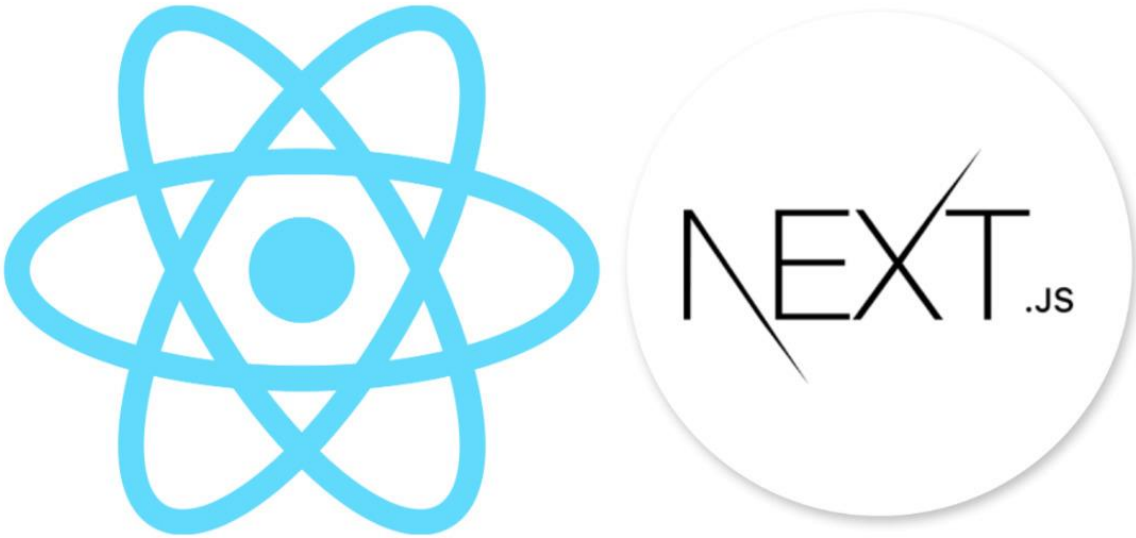


Figure 57: ReactJS & NextJS

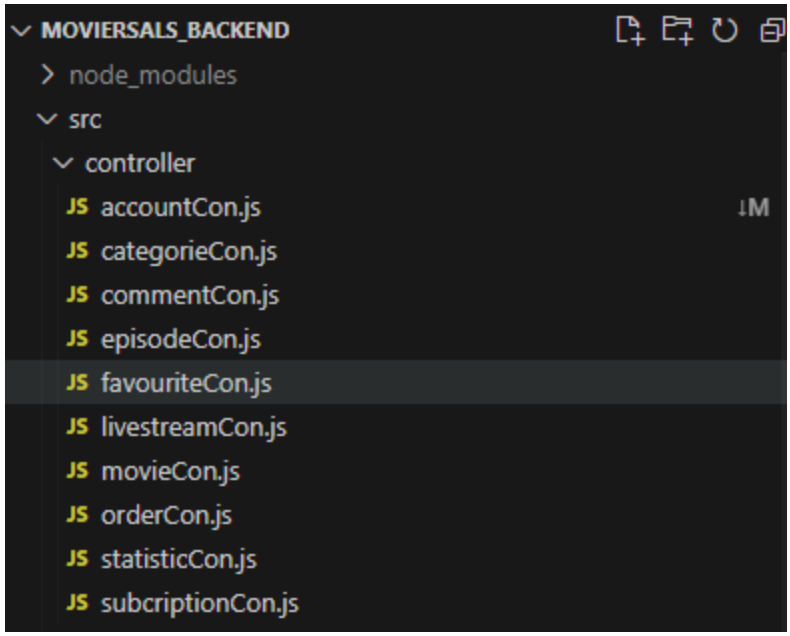
About Back-end development, we use Sping Boot framework of JAVA to write API, access to database and run handle events interacting with Google and Firebase.

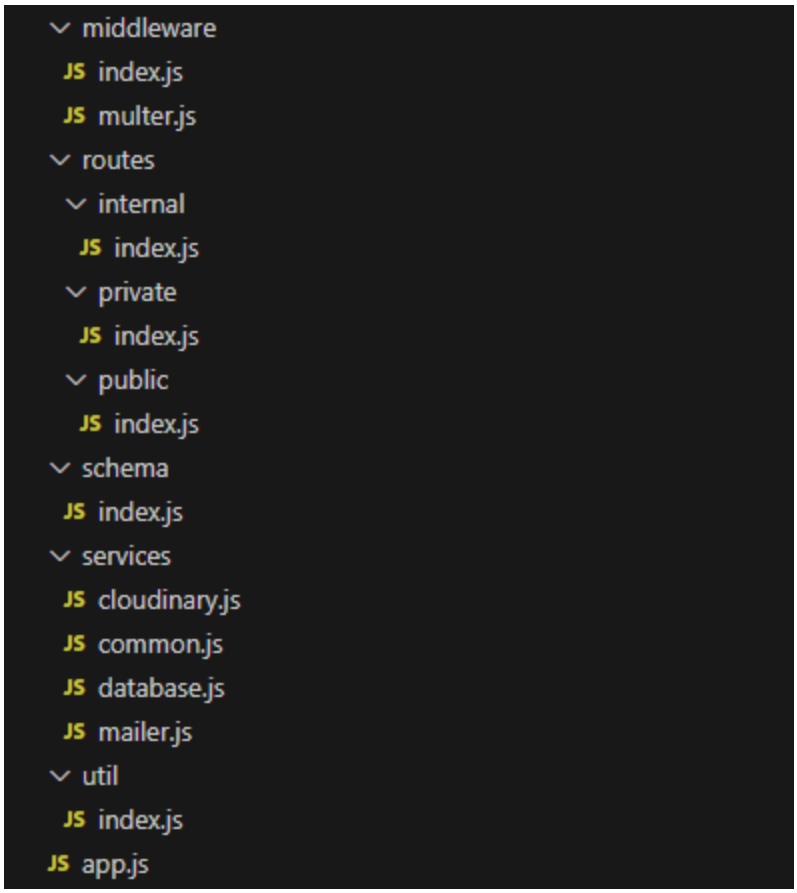
Figure 58: NodeJS & ExpressJS

5.1.3. Project file structure

5.1.3.1. Back-end file structure

- This is our Back-end file structure.


MOVIERSALS_BACKEND
node_modules
src
controller
accountCon.js
categorieCon.js
commentCon.js
episodeCon.js
favouriteCon.js
livestreamCon.js
movieCon.js
orderCon.js
statisticCon.js
subcriptionCon.js


middleware
index.js
multer.js
routes
internal
index.js
private
index.js
public
index.js
schema
index.js
services
cloudinary.js
common.js
database.js
mailer.js
util
index.js
app.js

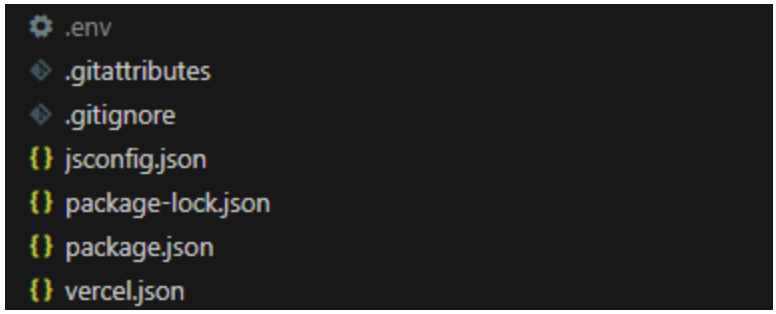
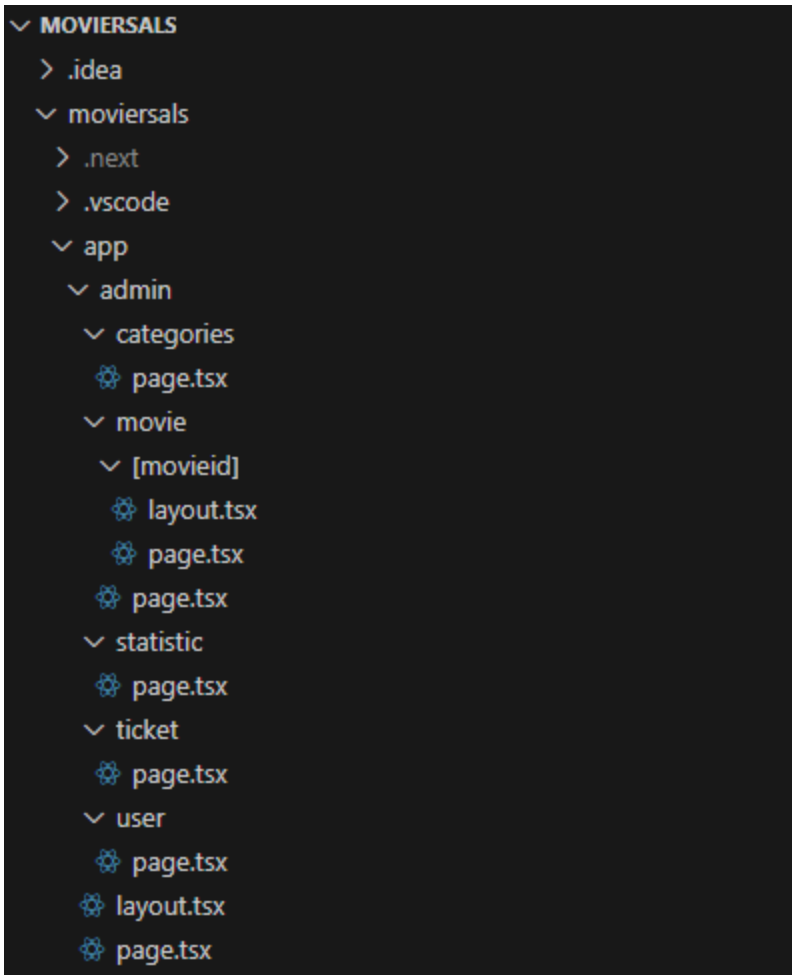
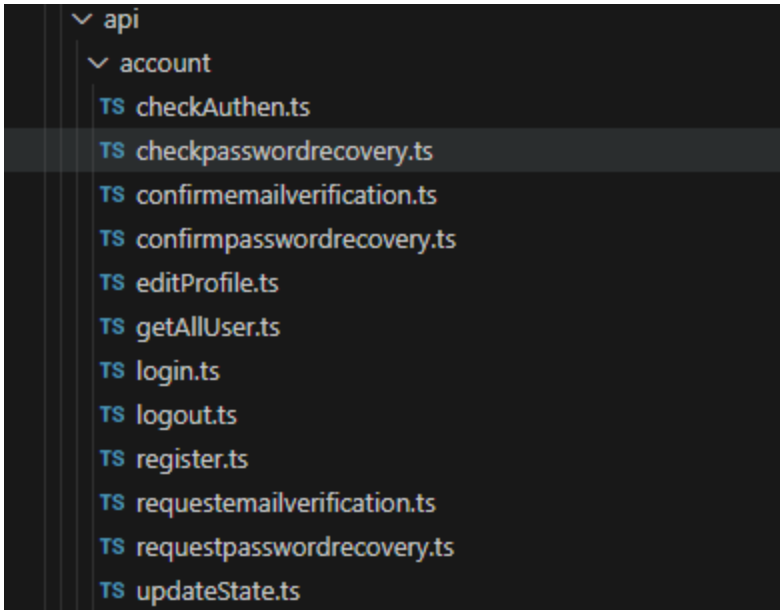


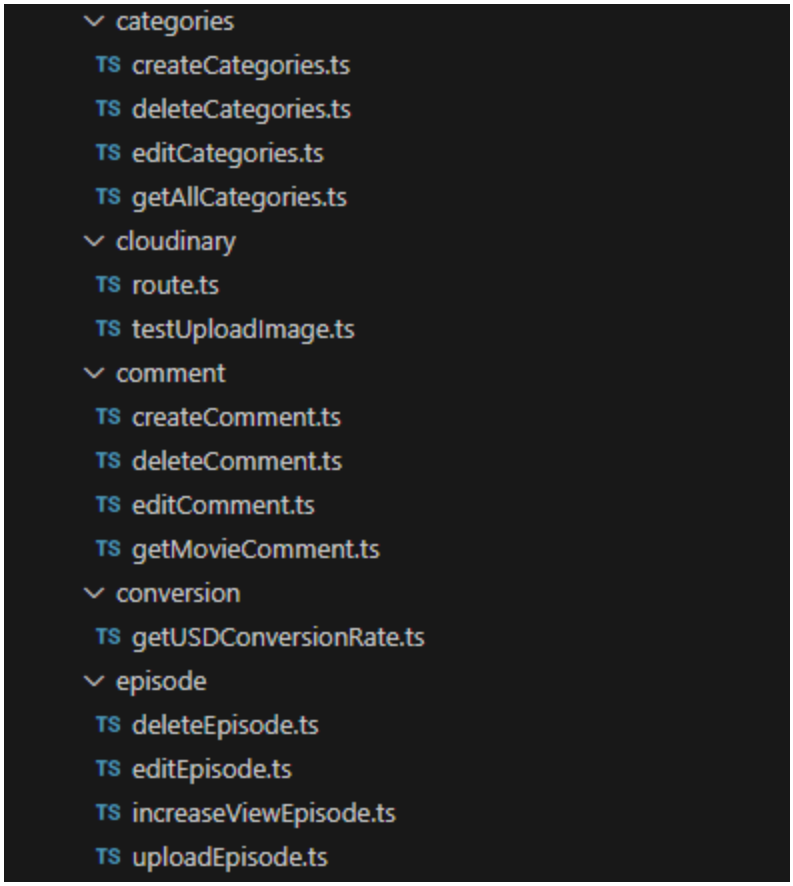
Figure 59. Back-end file structure

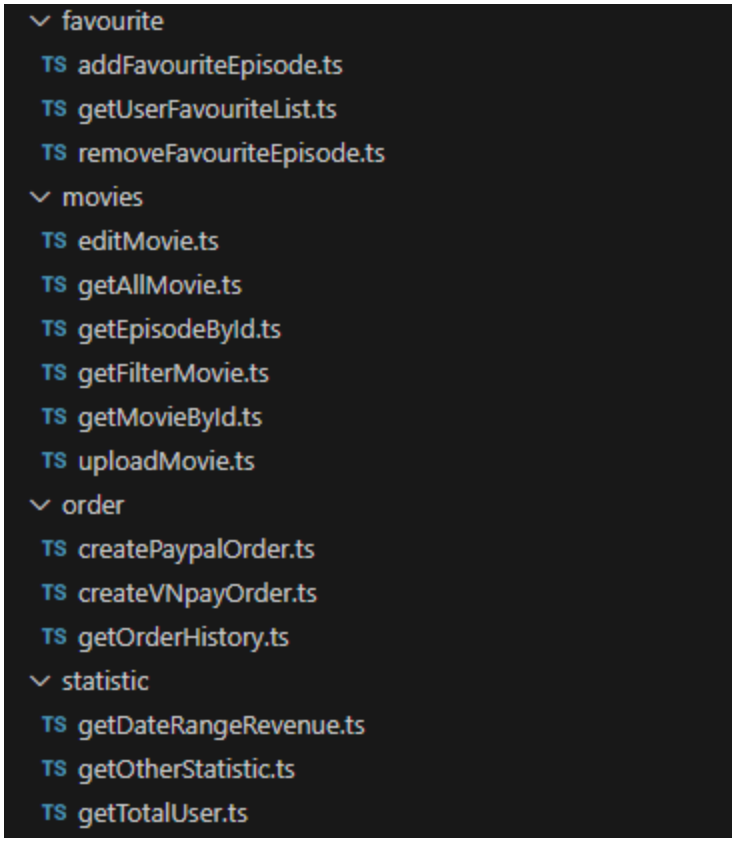
5.1.3.2. Front-end file structure

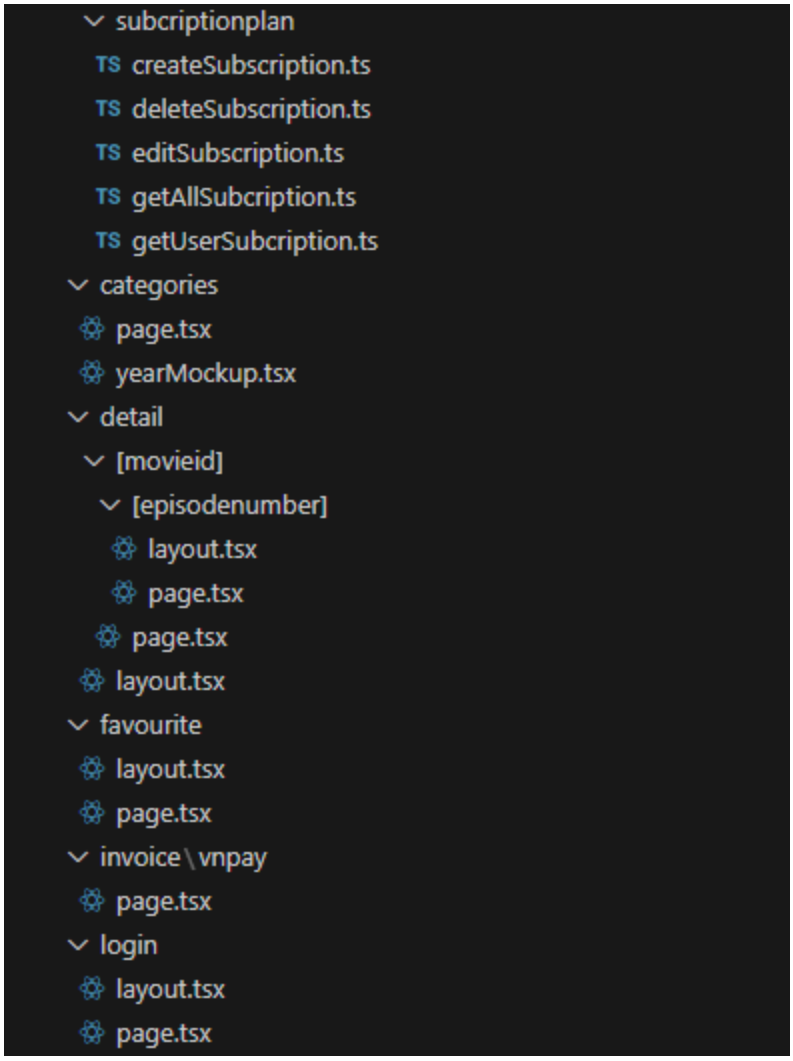
- This is our Front-end file structure.

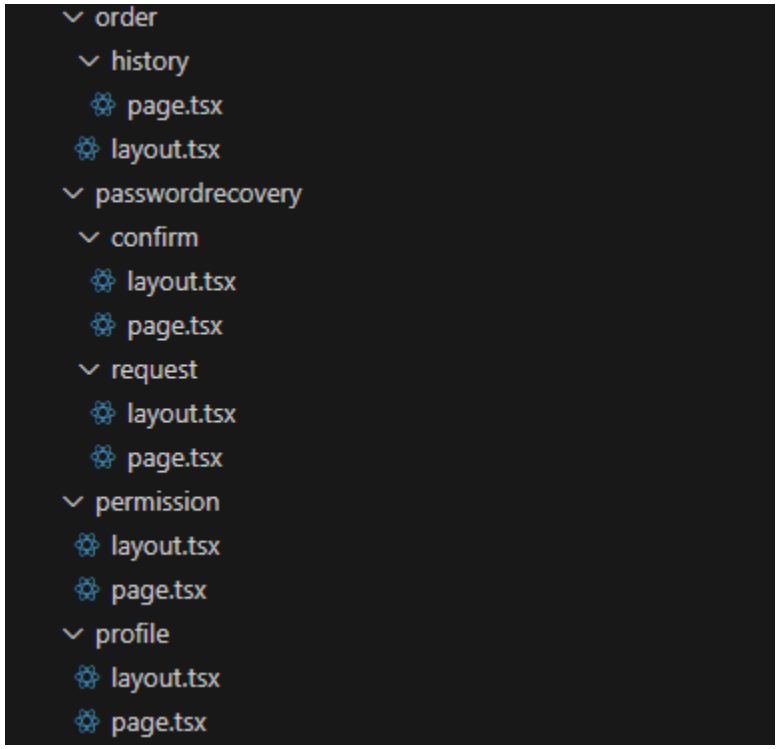


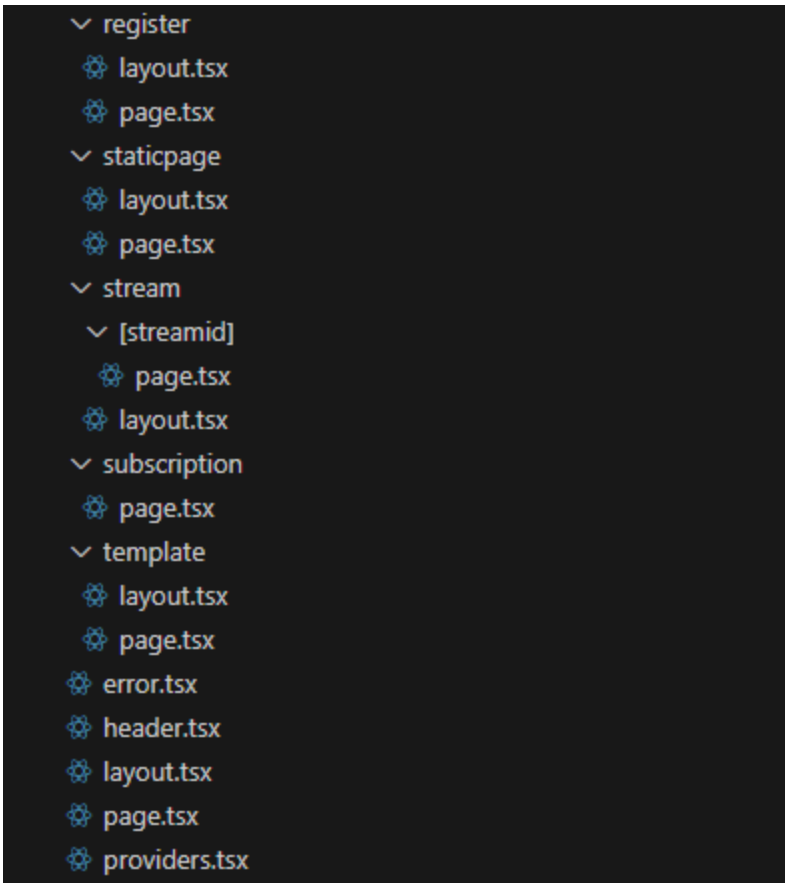


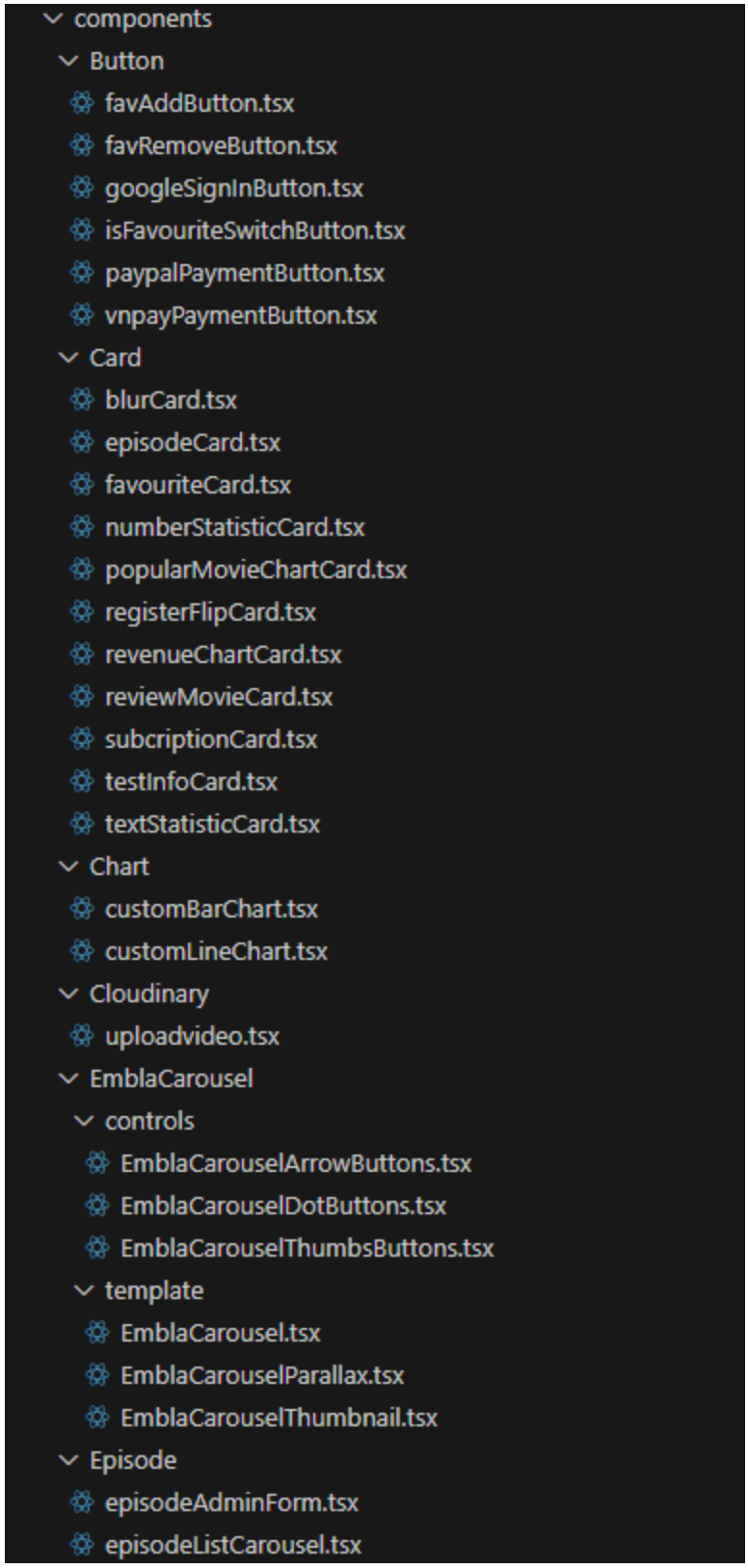

categories
createCategories.ts
deleteCategories.ts
editCategories.ts
getAllCategories.ts
cloudinary
route.ts
testUploadImage.ts
comment
createComment.ts
deleteComment.ts
editComment.ts
getMovieComment.ts
conversion
getUSDConversionRate.ts
episode
deleteEpisode.ts
editEpisode.ts
increaseViewEpisode.ts
uploadEpisode.ts

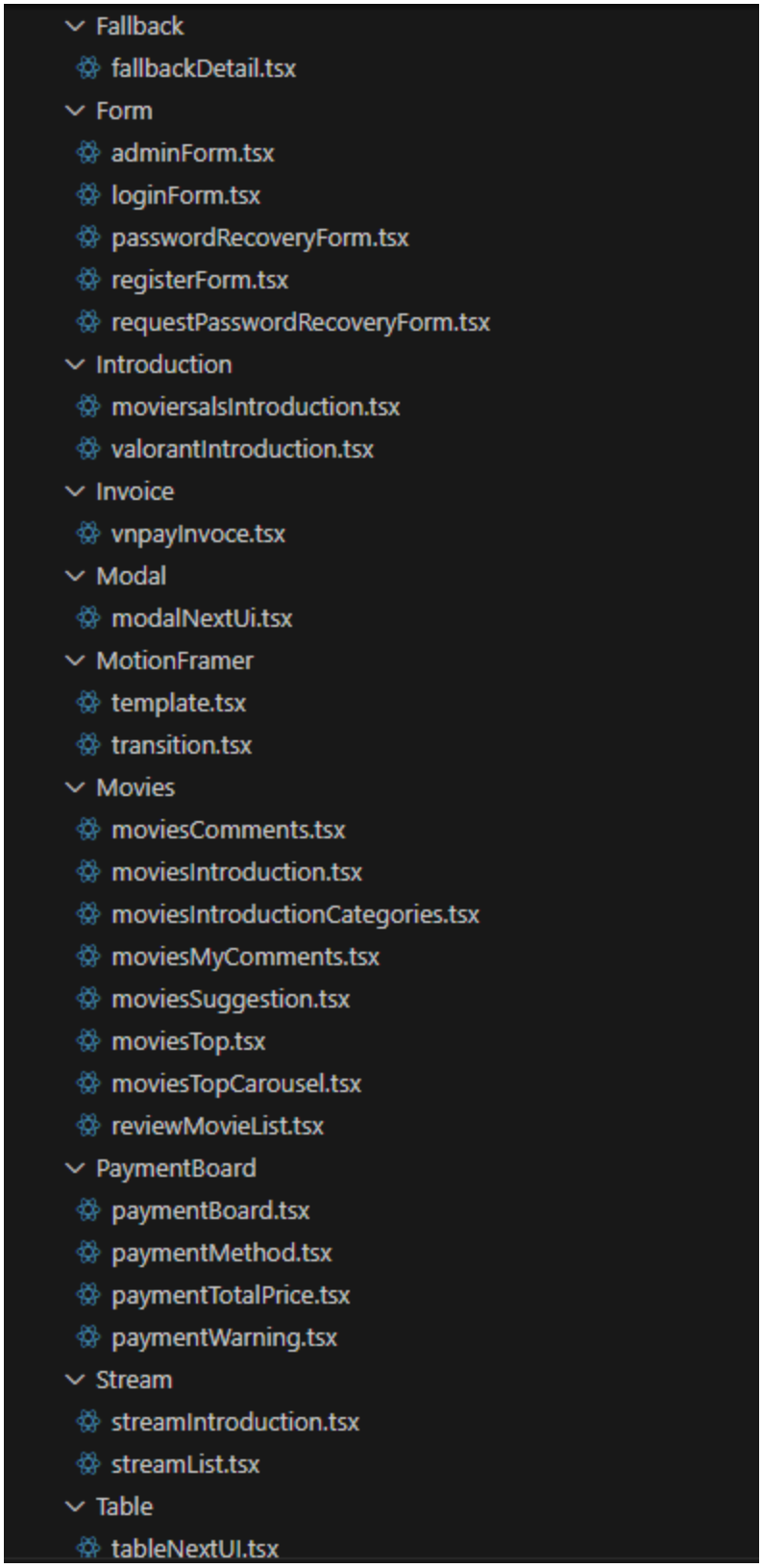

favourite
addFavouriteEpisode.ts
getUserFavouriteList.ts
removeFavouriteEpisode.ts
movies
editMovie.ts
getAllMovie.ts
getEpisodeById.ts
getFilterMovie.ts
getMovieById.ts
uploadMovie.ts
order
createPaypalOrder.ts
createVNpayOrder.ts
getOrderHistory.ts
statistic
getDateRangeRevenue.ts
getOtherStatistic.ts
getTotalUser.ts

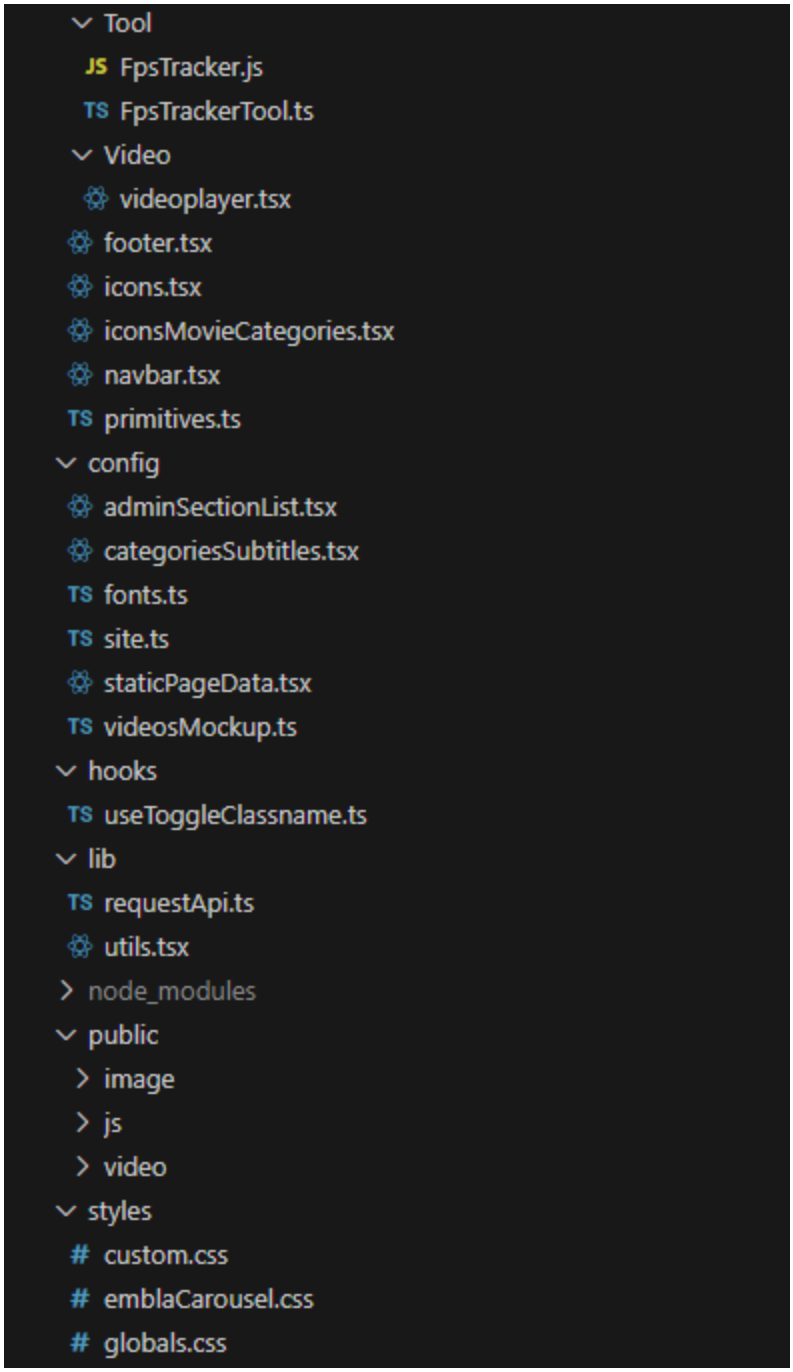

subcriptionplan
createSubscription.ts
deleteSubscription.ts
editSubscription.ts
getAllSubcription.ts
getUserSubcription.ts
categories
page.tsx
yearMockup.tsx
detail
[movieid]
[episodenumber]
layout.tsx
page.tsx
page.tsx
layout.tsx
favourite
layout.tsx
page.tsx
invoice\vnpay
page.tsx
login
layout.tsx
page.tsx


order
history
page.tsx
layout.tsx
passwordrecovery
confirm
layout.tsx
page.tsx
request
layout.tsx
page.tsx
permission
layout.tsx
page.tsx
profile
layout.tsx
page.tsx


register
layout.tsx
page.tsx
staticpage
layout.tsx
page.tsx
stream
[streamid]
page.tsx
layout.tsx
subscription
page.tsx
template
layout.tsx
page.tsx
error.tsx
header.tsx
layout.tsx
page.tsx
providers.tsx


components
Button
favAddButton.tsx
favRemoveButton.tsx
googleSignInButton.tsx
isFavouriteSwitchButton.tsx
paypalPaymentButton.tsx
vnpayPaymentButton.tsx
Card
blurCard.tsx
episodeCard.tsx
favouriteCard.tsx
numberStatisticCard.tsx
popularMovieChartCard.tsx
registerFlipCard.tsx
revenueChartCard.tsx
reviewMovieCard.tsx
subcriptionCard.tsx
testInfoCard.tsx
textStatisticCard.tsx
Chart
customBarChart.tsx
customLineChart.tsx
Cloudinary
uploadvideo.tsx
EmblaCarousel
controls
EmblaCarouselArrowButtons.tsx
EmblaCarouselDotButtons.tsx
EmblaCarouselThumbsButtons.tsx
template
EmblaCarousel.tsx
EmblaCarouselParallax.tsx
EmblaCarouselThumbnail.tsx
Episode
episodeAdminForm.tsx
episodeListCarousel.tsx


Fallback
fallbackDetail.tsx
Form
adminForm.tsx
loginForm.tsx
passwordRecoveryForm.tsx
registerForm.tsx
requestPasswordRecoveryForm.tsx
Introduction
moviersalsIntroduction.tsx
valorantIntroduction.tsx
Invoice
vnpayInvoce.tsx
Modal
modalNextUi.tsx
MotionFramer
template.tsx
transition.tsx
Movies
moviesComments.tsx
moviesIntroduction.tsx
moviesIntroductionCategories.tsx
moviesMyComments.tsx
moviesSuggestion.tsx
moviesTop.tsx
moviesTopCarousel.tsx
reviewMovieList.tsx
PaymentBoard
paymentBoard.tsx
paymentMethod.tsx
paymentTotalPrice.tsx
paymentWarning.tsx
Stream
streamIntroduction.tsx
streamList.tsx
Table
tableNextUI.tsx


Tool
FpsTracker.js
FpsTrackerTool.ts
Video
videoplayer.tsx
footer.tsx
icons.tsx
iconsMovieCategories.tsx
navbar.tsx
primitives.ts
config
adminSectionList.tsx
categoriesSubtitles.tsx
fonts.ts
site.ts
staticPageData.tsx
videosMockup.ts
hooks
useToggleClassname.ts
lib
requestApi.ts
utils.tsx
node_modules
public
image
js
video
styles
custom.css
emblaCarousel.css
globals.css

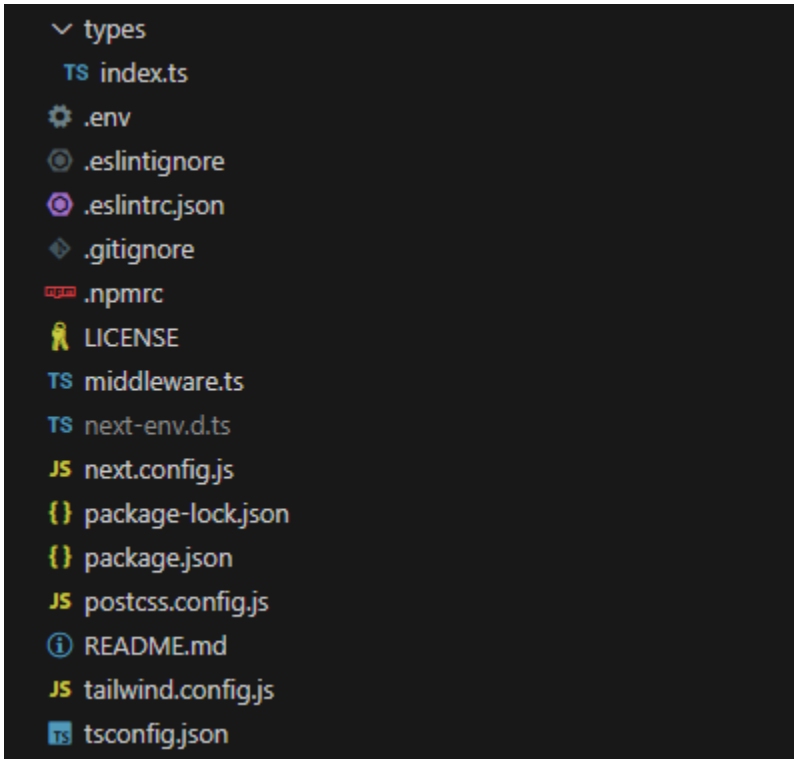


Figure 60: Front-end file structure

5.2. Deploymentation

We deploy Front-end, Back-end on Vercel, database on Supabase, images and videos on Cloudinary.

Link website and account information:

Table 69. Production test accounts

Link website	Username	Password	Role
https://moviersals.vercel.app/	admin	admin	admin
	khach1	123456	customer

5.3. Testing

Table 70. Testcases

ID	Description	Test steps	Test data	Expected Result	Actual Result	Status
TC_01	Login with a valid account	1. Enter email and password 2.Click submit	Email: khach1 Password: 123456	Notification: “Login successfully” and direct to Home Page	Notification: “Login successfully” and direct to Home Page	Pass
TC_02	Log in with an account that doesn't exist	1. Enter email and password 2.Click submit	Email: khach123 Password: 123456	Notification:” Error; account does not exist”	Notification: ” Error; account does not exist”	Pass
TC_03	Log in with an account that is the wrong password	1.Enter email and password 2.Click submit	Email: khach1 Password: 12345678	Notification:” Login Error; Password is incorrect”	Notification: ” Login Error; Password is incorrect”	Pass
TC_06	Register with valid information	1.Access website 2.Click on Account button on the top right 2.Click register button	Email: khach2 User name: khach2 Password: khach2	Register successfully and direct to Login Page	Register successfully and direct to Login Page	Pass

		3.Enter email 4.Enter username 5.Enter password 6.Click register				
TC_09	Customer add movie to favorite	1.Login by customer 2.Click a heart icon below movie		-Icon heart filled with pink color and the page favorite will add the movie into	-Icon fill pink and movie has added to favorite page.	Pass
TC_10	View Favorite Page	1.Login by customer 2.Select Wishlist in header 3.Show the list customer has added before		The list showed	The list showed	Pass

CHAPTER 06: CONCLUSION

6.1. Achievements

6.1.1. Knowledge and skills

- We gained a lot of insight into ReactJS, NextJS, NodeJS, ExpressJS, PostgreSQL, ... and learned how to build front-end websites, back-end websites, write API, build databases and connect them. We could also understand and apply other practical features into our projects such as deployment, ... Moreover, we could also improve our ability to develop and maintain a basic system, create and enhance business logic to complete the application .
- Teamwork skills had a big leap of improvement. Additionally, time and assignment management had a strong raise after doing this project because we had to balance the time of working on this project and the time at work or at school.
- Making an online movie streaming web application helped us a lot to practice our knowledge in real cases, gain more experience and also create a solution for our society.

6.1.2. Product

- In the realm of online streaming platforms, our product focuses on a dual-platform solution integrated. Key functionalities include:
- Administrative Control:
    - Empower website administrators to manage user accounts seamlessly.
    - Provide CRUD capabilities for administrators to handle crucial information related to movies, episodes and categories.
    - Enable administrators to manage user accounts.

- Customer-Centric Features:
    - Facilitate movie filter and detailed viewing for customers.
    - Allow customers to engage through comments and ratings.
    - Streamline the checkout process with two payment methods—pay later and online payment—and enable customers to track their orders.
- Enhanced User Interaction:
    - Enable customers to select movie resolution.
    - Support user account functionalities, including login, account creation, password management, and information editing.
- Role-Based Authorization: Incorporate distinct roles for customers and administrators, fostering seamless interaction and bolstering data management and security.
- User-Friendly Interface: Deliver a user-friendly interface across the website, prioritizing client comfort during service usage.
- Flexible Payment Methods: Offer a variety of payment methods, providing clients with diverse options for subscription plans within our system.

This comprehensive approach ensures a robust and user-centric online streaming experience, emphasizing versatility, security, and customer satisfaction.

6.2. Strengths and drawbacks

6.2.1. Strengths

- User-friendly and easy-to-see interface, simple operations and functions make users feel comfortable while using the web application.
- One key strength of a responsive movie streaming website is its ability to adapt seamlessly to various screen sizes and devices, ensuring a consistent user experience across desktops, tablets, and smartphones. This flexibility enhances accessibility, allowing users to enjoy content on their preferred

device without sacrificing quality or functionality. A responsive design also improves navigation, loading speed, and overall usability, making the platform more appealing and engaging for a diverse audience.

6.2.2. Drawbacks

- An online movie streaming web application without subtitle support significantly limits accessibility and user experience. It becomes challenging for non-native speakers to enjoy content in a foreign language, potentially alienating a global audience. Additionally, individuals with hearing impairments are excluded, reducing inclusivity. Subtitles also enhance comprehension in noisy environments or when watching at lower volumes, making their absence a notable drawback that can deter users and diminish overall engagement with the platform.

6.3. Future improvement

With the aforementioned restriction, the topic "Creating an online movie streaming web application" must have the following development guidelines in order to maximize customer happiness in the future:

- Enhance the user interface more.
- Include more user payment options, ...
- Add AI function like movie recommendation, support chatbot,..
- Develop a mobile app (on IOS, Android)

With the aforementioned development guidelines, the team is confident that the application's features will improve users' attempts to find themselves in the future when it enters the user market.

REFERENCES