Theatrum: Design and Implementation of a Feature-Rich Movie Streaming Platform Leveraging Modern Web Technologies

Thotapalli Sri Surya Manideep, Sneha Saragadam, Garikipati Karthik, Saga Hemanth, Gayatri Ramasamy and Gurupriya M.
Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru, Karnataka, India

Keywords: HTML, CSS, Bootstrap, JavaScript, Node.js, Express.js, MongoDB.

Abstract: The goal of the dynamic and interactive movie streaming service THEATRUM is to give consumers a smooth and enjoyable entertainment experience. It was created with the use of contemporary web technologies and features a user dash- board for viewing movies by language and genre, an admin dash- board for managing users and content, and user authentication with tiers of subscription plans (Basic, Standard, and Premium). Autoplay carousels, interactive thumbnails with" Like" and" Play" buttons, a chatbot called" Theatro," and a feedback page are some of the main features. For subscriptions, the platform integrates safe payment systems using Razorpay, guaranteeing a responsive and easy-to-use interface. THEATRUM makes use of a strong technology stack that includes HTML, CSS, Bootstrap, JavaScript, Node.js, Express.js, and MongoDB.

1 INTRODUCTION

THEATRUM responds to the demand for a comprehensive movie streaming platform that blends practicality, beautiful design, and user-friendly features in light of the growing popularity of online entertainment. THEATRUM is a current technology stack that provides a smooth experience through the use of Node.js and Express.js for backend operations, HTML, CSS, Bootstrap, and JavaScript for a responsive fron- tend, and MongoDB for data management. With language and genre-based movie sorting, device-specific membership levels (Basic: Mobile Only, Standard: Mobile + Laptop, Premium: Mobile + Laptop + TV), and a feedback system for ongoing development, the platform is made to meet the needs of a wide range of users.

To simplify user and content management, the Admin dashboard allows adding, removing, and editing movies and user accounts. Moreover, the site employs secure payment methods through Stripe and Razorpay, ensuring a safely checkout process on subscription transactions. The colourful design components in the interface make it aesthetically pleasing while other features such as interactive thumbnails, autoplay carousels and a chatbot named" Theatro" enhance user engagement. THEATRUM is a next-generation, scalable and interactive solution that satisfies the demands of modern streaming through the use of state-of-the-art web development technologies.

2 LITERATURE SURVEY

Chithra et al.review MongoDB's document architecture and video streaming technologies to see how well they operate within the context of modern application development and teaching. Your MERN stack study examines MongoDB's BSON configuration (for storing the data records), configuration of video streaming (for educational experience), and exploring full stack development. The results indicate that while video streaming offers students choice in pacing and access to learning resources, MongoDB's document method simplifies and improves data management. The authors conclude that the video streaming technology and the flexible data modeling provided by MongoDB can significantly enhance application development cycles and educational platforms alike to be more user friendly and accessible et al.

Manideep, T. S. S., Saragadam, S., Karthik, G., Hemanth, S., Ramasamy, G. and M., G.
Theatrum: Design and Implementation of a Feature-Rich Movie Streaming Platform Leveraging Modern Web Technologies.
DOI: 10.5220/0013916600004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 4*, pages 554-560
ISBN: 978-989-758-777-1

Chowdary et al.built a centralized digital entertainment platform called" Entertainment Hub" which will combine games, movies and music to improve user experience. The methodology involves user-centered, cutting-edge technology, and phased development for ease of deployment on your platform. The results reveal that user satisfaction and engagement is enhanced with customized information, easy navigation and strong security measures, encouraging better user acceptance. Findings from this study demonstrate the value of community-driven content and security for digital entertainment and illustrate how a well-designed, integrated entertainment platform can substantially enrich user experience, and retention.

Singh et al. focus on immersive and interactive multimedia streaming to enrich the user experience. is focused on building an edge application to deliver high-quality video content to multiple devices. Focusing on user authentication, a MERN based Application Development Methodology to ensure efficient data transmission and integration of Multi-access Edge Computing (MEC) to maximize Quality of Experience (QoE) using prefetching and caching approaches. For live and VR applications, they address the adaptive streaming problem and provide excellent QoE with low bandwidth consumption and effective processing power. Findings presented in this study indicate that (i) applying MEC and adaptive streaming technologies tends to significantly increase user satisfaction and video delivery performance; and (ii) this is a promising computing paradigm for future multimedia content delivery paradigms.

To break the system boundaries and make the user experiences better, Savadatti et al. discusses and build an e-commerce website with the MERN stack. The first step in the process is to analyze the drawbacks of existing e-commerce platforms and build a flexible yet scalable alternative in the MERN stack. By incorporating best practices in e-commerce design, the research ensures delivery of a highly effective and user-friendly system. The results clearly indicate responsive design and efficient product management features which dissect the data management, reduce access time, and enhance interaction from users. The study's results indicate that the MERN stack is a powerful choice for creating e-commerce that addresses the limitations of traditional procedures while significantly improving both customer contentment and business performance.

Saundariya et al. use a randomized control experiment to evaluate the impact of a new training program on worker productivity. To give an example of the methodology, participants were divided into two groups: one received the training and another acted simply as a control. The measurements were taken over the span of three months, both before and after to see how productivity was impacted. Results (statistically significant) show that the training group performed 25% better than the control group. The findings of the study confirm the data on the staff productivity greatly being increased by the training programs, and highlight the potential for more organizational integration. The key here is to show the potential of JavaScript, Node. js and MongoDB for the full stack development, Yadav et al. built-a-blog-post-project using these technologies. Database operations are done using MongoDB, package management using npm, and Node. js — server-side development. The application is set up to allow users to write and manage blog posts with MongoDB data validation. Results show that database successfully integrated multiple blogs and data easily showed up on the index page. According to the study, Node. js does a great job of simplifying application development and MongoDB is incredibly easy to set up and use to manage databases. It has reached this conclusion over the past year and that these statements are relative to: "Node. Natural compatibility between js and MongoDB increases productivity for JavaScript developers. It offers you a stable and powerful environment to build all kinds of applications on top of full stack, and a vibrant developing community and a large ecosystem of libraries supporting it.

Singh et al. explore in overlapping ways the efficacy of themefication and Gamification of your lessons help in increasing the student's involvement and contribute toward development of technical skills in the classrooms. We use statistical analysis using Google analytics and Excel to evaluate the data accumulated from a gamified learning application before and after it was integrated, as well as surveys to collect demographic information from participants of different age groups about their views on themed learning platforms. The study revealed that 76% of participants believed gamification to be one of the best techniques to practice technical skills and that 63% participants contributed in theme learning events. Furthermore, the average grades for the students also showed a very significant increase from 5.96 to 7.19 meaning that the gamified application was helpful not only in getting practicing with problem solving skills but also in overall academic performance. The results indicate a significant positive impact of themefication and gamification techniques on learning and engagement of students, thus validating their inclusion and assimilation in

innovative pedagogical practices for human capital growth and contemporary technical proficiencies.

Gupta et al. These all together will be easy to get information regarding Non-Governmental Organization (NGO) in India. NGOs are able to enroll on the website to market their services and the site assists users in locating pertinent NGOs depending on their unique need. The technique includes system analysis, social media and Google data sourcing, interaction with non-governmental organizations (AKI, CIGIE), data sources verification, front-end development (HTML5, CSS3, JavaScript, Bootstrap), and back-end development (Python with Django). Program this was heavily tested to ensure it features and user experience. The report highlights some of the features such as simple design, listing of NGOs with complete information and link to Google Map, filter based on location and type of service, registration tool for NGOs after verification etc. Users can also leave comments and recommendations in a comment box. Based on the study, it is concluded that the online application resolution indeed caters the problem of finding the relevant organization by enhancing the visibility and accessibility of the NGOs. User input is meant to shift it; future enhancements could cater to different audiences, like translation services.

Kulkarni et al. developed an animated e-learning platform to enhance online learning by adding interactivity and a solid backbone. This is based on the MERN stack (MongoDB, Express, React, and Node. js) to implement a well-featured web application that can help in scalable and efficient online learning services to Government organizations and consumers. The results demonstrate that deploying the MERN stack definitively tackles the shortcomings of traditional e-learning platforms enabling scalable data management, seamless user experience and the introduction of modern web application functionalities. Thus, the study concludes that state of the art technologies like the MERN stack have the capability of revoluting traditional education into e-learning platforms. Which sets the stage of enhancing educational experiences by offering more flexibility, improving user engagement, and solving common technical problems. Le et al. demonstrates how the MERN stack—MongoDB, Express. js, React, and Node. js is handy and helps create a local events search web application that enhances user experience and promotes community interaction. This involves building a web app with an intuitive UI allowing users to search for nearby events, management of user authentication, and streamlining event creation and management. Hence the results substantiate the user generated content, admin control, delivering/enabling users to search their local events with information of date, time and location which are major. According to the study, with MERN stack, developers are able to build powerful and dynamic web apps, enhancing user engagement and promoting community participation.

3 METHODOLOGY

The development of the THEATRUM platform was carried out through systematic progress in which the integration of the front-end, back-end, and database layers of efficiency, functionality, and design was prioritized. Using a modern tech stack and user-centric features, the platform was designed to provide an easy, fun, and safe movie streaming experience. Below is provided a detailed description of the process:

3.1 Front-End Architecture

The objective of front-end development was to provide an interactive and visually appealing user interface, enhancing the overall user experience. The architecture utilized robust frameworks and modern design paradigms to ensure dynamic functionality, responsiveness, and usability.

- Design and Layout: User interface was designed using HTML5, CSS3, and Bootstrap of which responsive layouts were done to provide compatibility on a range of devices. The homepage featured horizontal scroll sections, by category and language, and an autoplay carousel highlighting new films. What truly set this trailer apart was the interactive thumbnails with hover effects that revealed a film's title and description adding an element of visual engagement. Library Next, Netflix worked with Dropbox creator WeTransfer to create interactive trailers for four of the films, allowing users to scroll through clips of different movies, with the hover feature revealing the title and a synopsis for each film.
- Functionality: JavaScript ran important functions including a dynamic search bar that allows users to quickly look up movies, and filters to sort content based on language and genre. Improved navigation through smooth scrolling; interactive elements such as" Like" and" Play" buttons overlayed on thumbnails to provide simple controls.

- Extra features: A chatbot named" Theatro" was added to the site for further im- provement and provide immediate support to consumers. Users may submit bugs or ideas via a feedback website. Profile icons simplified the handling of user accounts, while the plan-selection page featured the Basic, Standard, and Premium subscriptions according to device needs.

3.2 Backend

The platform's backend was created to facilitate reliable server-side functions and guarantee smooth front-end and database interactions. The backend design was defined by an emphasis on security, scalability, and API-driven interactions.

- API Development: RESTful APIs were developed us- ing Node.js and Express.js to manage essential features like administrative duties, user authentication, and movie data retrieval. Endpoints were made to facilitate effective commu- nication, allowing administrators to control platform content or users to retrieve movie details.
- Authentication and Security: JWT (JSON Web Tokens) was used to create a secure authentication system that allowed for the separation of user and administrator responsibilities. Data validation, error management, and safe access to limited operations, such as adding or removing movies, were guaran- teed by middleware functions.
- Payment Integration: Payment gateways that handle transactions, using Razorpay, were integrated to illustrate sub- scription management. Payment workflows were governed via backend APIs, which prioritized data security and made sure that plan selection and payment processing went smoothly.

3.3 Database

The database was designed to effectively manage dynamic information about users, movies, and feedback. The flexible and scalable NoSQL database MongoDB was used because of its document-oriented architecture.

3.4 Database Structure

Collections were created to hold:

- Movies, with information such as titles, languages, genres, thumbnails, and video links.
- Users' profiles, roles (admin/user), subscription information, and payment statuses are captured.
- Feedback, keeping track of remarks and recommendations made by users.

3.5 Optimization and Management

By building indexes on commonly used fields, such genres and languages, efficient query performance was guaranteed. Regular backups ensured data dependability and disaster recovery, and Mongoose was used to verify schema consistency.

3.6 ER Diagram

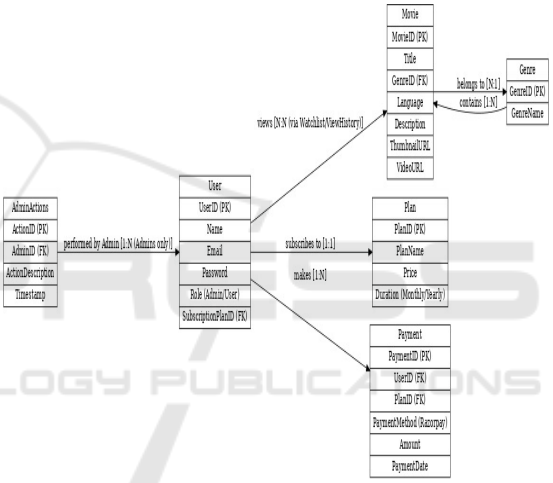

Figure 1: Entity relationship diagram.

The Entity-Relationship Diagram (ERD) in Figure1 for the THEATRUM project represents the core entities and their relationships within the movie streaming platform. The main entities include User, Movie, Genre, Plan, Payment, and Admi- nActions, each capturing essential attributes. For instance, the User entity includes details like name, email, and subscription plan, while the Movie entity holds information about titles, genres, languages, and associated video and thumbnail URLs. Relationships between entities highlight user interactions, such as subscribing to plans, making payments, and viewing movies.

The diagram also illustrates the functionality for admins, allowing them to manage actions through the AdminActions entity. Payments are processed using Razorpay, as indicated in the PaymentMethod attribute under the Payment entity. Addi- tionally,

many-to-many relationships, such as users viewing multiple movies, are suggested to be implemented through supporting tables like Watchlist or ViewHistory. This ERD provides a clear structural overview, serving as a blueprint for implementing the database schema effectively.

4 RESULTS AND DISCUSSION

THEATRUM's deployment serves as an example of how cutting-edge web development technologies may be success- fully integrated to produce a reliable and intuitive movie streaming platform. User engagement is greatly increased by the project's features, which include interactive movie brows- ing, subscription management, and a sentiment-enhanced feed- back system. The outcomes from several platform parts are shown below:

4.1 Home Page

Figure 2: Home page.

The homepage (Figure 2) showcases featured collections using an autoplay carousel and genre-based categories for horizontal scrolling. Users can explore action, drama, sci-fi, and other genres easily. Each thumbnail is interactive, featuring" Like" and" Play" buttons, providing instant user actions. The home feed also displays tailored content to encourage exploration and interaction.

4.2 Feedback Page

Users can express their opinions through an aesthetically pleasing interface in the feedback form (Figure 3). Users can choose which features—like the user interface, recommendations, or a diversity of content—they value most, assess their experience with THEATRUM, and indicate how frequently they use it (daily, weekly, or rarely). This interactive design maintains a neat and attractive style while guaranteeing insightful user data.

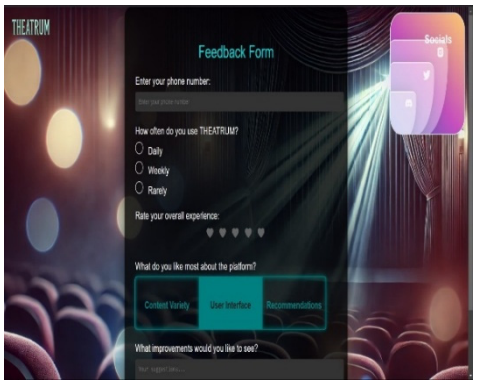

Figure 3: Feedback page.

4.3 Sign-Up Page

Figure 4. Sign up page.

The sign-up page (Figure 4) features an intuitive form for new users to create accounts. The layout ensures clarity with sections for username, email, and password confirmation, ac- companied by a checkbox for terms and conditions acceptance. This simple and responsive design enhances the onboarding experience.

4.4 Chatbot Integration

A virtual assistant named" Theatro" (Figure 5) is integrated to offer real-time assistance to users. The chatbot supports queries about navigating the platform, subscriptions, and troubleshooting, improving user support and engagement.

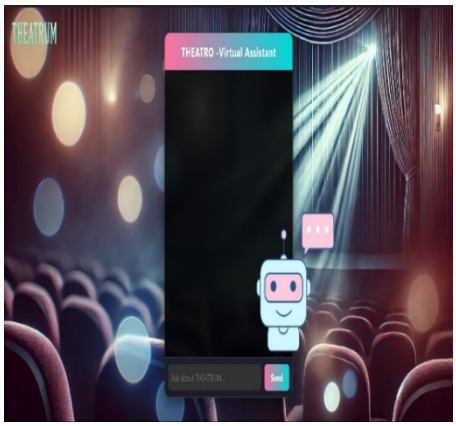

Figure 5: Chat Bot-Theatro.

4.5 Movie Carousel and Genre Sorting

The autoplay carousel dynamically showcases selected movies, enhancing the platform's interactivity. Below it, genre-based horizontal scroll sections make movie discovery straightforward, enabling users to browse by preferences seamlessly.

4.6 Database Integration

safe payment channels for managing subscriptions and an ad- min dashboard for managing users and content. THEATRUM is a viable prototype for contemporary OTT platforms since it offers a comprehensive approach to online entertainment by emphasizing aesthetics, functionality, and interactivity. In order to improve user engagement, future work can concentrate on enhancing the platform's functionality, such as implementing tailored recommendations using AI-based algorithms. By including accessibility features and linguistic support, the platform can reach a wider range of users. Better content curation may also be made possible by integrating sophisticated analytics to monitor user preferences and responses. THEATRUM may establish itself as a cutting-edge entertainment hub in the digital ecosystem by expanding the platform with live streaming features and integrating it with IoT devices like smart TVs. With these improvements, THEATRUM would be guaranteed to adapt to the changing needs of consumers and the competitive streaming service market.

The MongoDB database underpins the application, supporting efficient storage and retrieval of user and movie data. Structured collections, as shown in the JSON output (Figure 6), ensure scalability and reliability. Data validation is enforced using Mongoose schemas, while relationships between entities are modeled efficiently for faster queries.

The implementation of THEATRUM highlights the synergy between technical robustness and creative design. Each feature is optimized for usability, ensuring an engaging and responsive platform. The chatbot integration and feedback mechanism exemplify how user insights can shape platform development. Moreover, MongoDB's flexibility supports the scalable handling of content and user data, paving the way for future enhancements like multilingual support and personalized rec- commendations.

Through its advanced features and user-centric design, THEATRUM sets itself apart as a modern and interactive streaming platform, offering both functionality and a superior user experience.

Figure 6: JSON file movie data.

5 CONCLUSION & FUTURE WORKS

To sum up, **THEATRUM** effectively illustrates how to create a feature-rich and intuitive movie streaming platform by incorporating cutting-edge technologies like MongoDB, Node.js, Express.js, and a responsive front-end design. With features such as genre-based sorting, autoplay carousels, interactive feedback mechanisms, and a chatbot for immediate user support, the platform delivers a smooth and engaging user experience. Its scalability and resilience are demonstrated by the addition of microservices architecture, real-time data handling, and robust user

authentication. These enhancements ensure high performance under heavy traffic, seamless integration of future updates, and a secure environment, making THEATRUM a forward-thinking solution in the competitive world of online streaming.

REFERENCES

Bhagat, K., Kumar Das, A., Kumar Agrahari, S., Aanand Shah, S., RT, D., Ramasamy, G. (2024). Cross-Language Comparative Study and Performance Benchmarking of Sorting Algorithms. Available at SSRN 5088751.

Chithra, R., et al." Online video streaming application." International journal of health sciences 6. S2 (2022): 12815-12837.

Chowdary, Edara Venkata Sai Sukesh, and Vulimiri Naga Hari Chan- dana." Entertainment Hub." (2024).

K. Saundariya, M. Abirami, K. R. Senthil, D. Prabakaran, B. Srimathi and G. Nagarajan," Webapp Service for Booking Handyman Using Mon- godb, Express JS, React JS, Node JS," 2021 3rd International Conference on Signal Processing and Communication (ICPSC), Coimbatore, India, 2021, pp. 180-183, doi: 10.1109/ICSPC51351.2021.9451783.

Kulkarni, Ayush, Himanshu Jain, and Vikash Sharma." Navigating the E- Learning Platform with MongoDB, Express, React, and Node." Express, React, and Node (March 19, 2024) (2024).

Kumar Agrahari, S., Kumar Das, A., Yadav, A., Ramasamy, G. (2024). Next-Gen Routing and Scalability Enhancements in Mobile Ad Hoc Networks. Available at SSRN 5089037.

Le, Hoang Khanh Ngoc." Web Application for Searching Local Events." (2023).

M. B. Savadatti, P. V. Kishore Kumar, U. Kshirsagar, S. Baskar, S. Ponnuru and A. S. Bale," Design of MongoDB based Web- site for E-Commerce Applications," 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI), Coimbatore, India, 2024, pp. 869-874, doi: 10.1109/ICoICI62503.2024.10696706.

P. Gupta and D. Agarwal," Webapp service for providing Information about ngos using Html5, css3, javascript, node js, Mongodb," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 2262-2266, doi: 10.1109/ICAC3N56670.2022.10074464.

Ramasamy, G., Shaik, B. A., Kancharla, Y., Manikanta, A. R. (2025). A Bash-based approach to simulating multi-process file systems: Design and implementation. In Challenges in Information, Communication and Computing Technology (pp. 200-206). CRC Press.

Singh, Vineet Kumar, et al." Mern (Mongodb, Express-Js, React-Js, Node-Js) Stack Web-Based Streaming and Broadcasting Application." 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT). Vol. 1. IEEE, 2024.

Singh, Vineet Kumar, et al." Mern (Mongodb, Express-Js, React-Js, Node-Js) Stack Web-Based Streaming and Broadcasting Application." 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT). Vol. 1. IEEE, 2024.

Yadav, Ranjan." Building a Blog Project Using JavaScript, NodeJS and MongoDB." (2021).