International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 5, Sep-Oct-2025, ISSN (Online): 2395-566X

Movies for U

[1]Emmadi Uday, [2]Anand Jawdekar, [3]Komati Hema, [4]Lakshmikanth Vuyyuru, [5]Tadikamalla Vinod kumar
Department Of CSE
Parul Institute of Engineering And Technology, Vadodara, Gujarat, India

Abstract - This paper presents MoviesForYou, a web-based movie booking platform that integrates an AI-powered conversational chatbot for mood-based movie recommendations, automated seat suggestions, and a data analytics module for theater and revenue insights. The system allows users to describe their mood in natural language; the chatbot leverages language understanding and movie metadata to recommend titles that match the user's affective state and viewing prefer- ences. The analytics module computes weekly booking trends, genre popularity, theater performance, rating-based summaries, and revenue trend analysis. The platform intentionally supports offline payment workflows (seat-on-hold with time-limited reser- vation) rather than online payment. We describe the system architecture, implementation details (based on the provided project codebase), evaluation methodology, key results, and future directions. The paper includes flowchart and system- architecture placeholders, experimental and analytic outputs, and an APA-style reference list.

Keywords - movie recommendation, mood-based chatbot, conversational recommender, analytics, seat suggestion, booking system, offline payment.

INTRODUCTION

Online movie ticketing systems are a mature but con- tinually evolving domain. Modern platforms must integrate multiple concerns simultaneously: inventory management, showtime scheduling, real-time seat allocation, multi-screen theatre layouts, and user-facing discovery mechanisms that surface relevant films to a diverse audience. Beyond these traditional requirements, contemporary users expect person- alized and natural interactions — for example, conversational discovery, contextual suggestions, and rapid access to the best available seats. Incorporating intelligence into ticketing platforms therefore improves both discovery and conver- sion: recommender systems can reduce search friction while conversational interfaces (chatbots) can translate informal user inputs (such as mood descriptions) into actionable recommendations.

Background and Motivation

Recommender systems have been broadly studied in the literature and typically employ collaborative filtering, matrix factorization, content-based filtering, and hybrid approaches to infer user preferences and rank items (Koren, Bell, & Volinsky, 2009; Ricci, Rokach, & Shapira, 2015). These techniques are well suited for personalized long-term rec- ommendations when explicit or implicit user histories are available. However, practical production systems must also handle the cold-start problem and support ad-hoc, context- driven requests (for instance, a one-off request such as "I'm feeling nostalgic tonight"). Recent advances in natural language understanding and large language models (LLMs) enable systems that accept free-form descriptions (e.g., mood labels like "nostalgic", "energetic", or "melancholic") and map them to content recommendations with minimal histor- ical data (Goodfellow, Bengio, & Courville, 2016; He et al., 2017). This capability motivates the integration of conver- sational agents with classical recommendation engines: the chatbot can act as a bridge between unstructured user intent and structured movie metadata.

Problem Statement

Despite existing capabilities, a gap remains in systems that simultaneously satisfy (1) intuitive natural-language discovery (especially mood-based), (2) reliable, real-time booking and seat allocation, and (3) actionable analytics for theatre managers. Many platforms excel at one or two of these aspects but not all three together. Specifically, challenges include: (a) mapping ambiguous, affective user input to well-scored candidate movies; (b) preventing double- booking while offering a responsive seat-hold mechanism for offline payment models; (c) recommending contiguous and comfortable seats for groups while maximizing theatre uti- lization; and (d) producing analytics (weekly trends, revenue time series, genre demand) that are both correct and readily interpretable by non-technical stakeholders.

Objectives and Contributions

This work presents MoviesForYou, a prototype platform that addresses the challenges above by integrating an AI- driven conversational recommender with a robust booking subsystem and an analytics dashboard. The primary contri- butions of this paper are:

- Conversational mood-based recommendation pipeline: a practical architecture that converts free-form mood descriptions into ranked movie
- suggestions using a combination of semantic mapping and metadata-driven scoring.
- Seat suggestion algorithm (Movie Buddy): a heuris- tic and implementation for recommending optimal seats (or contiguous seat blocks) based on theatre geometry, central viewing preference, and current oc- cupancy constraints.
- Offline payment-friendly booking workflow: a seat- hold mechanism that supports time-limited reservations and optionally adjusts hold durations based on coarse distance/time heuristics without requiring online pay- ment.
- Analytics module: an aggregation and visualization layer that computes weekly booking trends, revenue trend analysis, genre popularity metrics, and theatre performance indicators to support data-driven decision making.
- Prototype and evaluation: implementation details (based on the supplied codebase), qualitative results from user trials of the conversational interface and seat suggestions, and illustrative analytics visualizations to demonstrate system utility.

Scope and Assumptions

The prototype assumes access to structured movie meta- data (title, genres, runtime, certification, rating, showtimes) and theatre layout data (seat coordinates, rows, and occu- pancy). For recommendation quality, the system leverages either a lightweight semantic mapping between mood terms and genre/keyword vectors or an LLM-based semantic scorer when available. The platform intentionally uses an offline payment model (seat-on-hold with a time-limited reserva- tion) to simplify payment compliance and focus this study on recommendation, seat allocation, and analytics. Privacy-preserving practices are assumed: mood inputs are processed with user consent and persisted only when explicitly opted in.

Paper Organization

The remainder of the paper is organized as follows. Section III details the system architecture and analytics components, including flowcharts and data flow descriptions. Section ?? describes implementation choices and integra- tion details referencing the project codebase. Section ?? presents results from prototype evaluation, sample analyt- ics visualizations, and a discussion of observed behavior. Section ?? highlights limitations, ethical considerations, and future work. Finally, Section ?? concludes with a summary of contributions and potential production extensions.

II. RELATED WORK

Foundational texts in machine learning and deep learn- ing provide the theoretical and practical basis for mod- ern recommender systems. Classical treatments describe how models can learn representations and generalize from data, covering supervised, unsupervised and representation- learning paradigms that are relevant for feature extraction and model design in recommendation tasks (Mitchell, 1997; Bishop, 2006; Goodfellow, Bengio, & Courville, 2016). These sources underpin choices about model capacity, reg- ularization, and evaluation that inform recommender devel- opment.

Collaborative Filtering and Matrix Factorization

Collaborative filtering (CF) remains a cornerstone of rec- ommender systems. Early neighborhood-based approaches (user-based and item-based) estimate ratings by finding simi- lar users or items; work by Sarwar et al. (2001) demonstrated the practicality and relative efficiency of item-based CF for large-scale systems. Matrix factorization methods later provided a compact latent-factor formulation that scales well and handles sparsity by mapping users and items into a shared low-dimensional space (Koren, Bell, & Volinsky, 2009). Matrix factorization also supports extensions such as regularization, bias terms, and implicit-feedback modeling, making it a robust baseline for many production recom- menders.

Neural and Deep Models for Recommendation

Deep learning has enabled more expressive recommender architectures. Neural Collaborative Filtering (NCF) replaces the simple inner product in matrix factorization with a train- able neural network that can learn complex, non-linear user–item interactions (He et al., 2017). More broadly, sequence models (RNNs, Transformers) and hybrid architectures com- bine content features, session context, and latent factors to capture richer user behavior. These approaches often yield improved accuracy on large datasets but require careful engineering for training, regularization, and inference latency (Goodfellow et al., 2016; Ge´ron, 2022).

Time-aware and Session-based Recommendation

User preferences and item popularities evolve over time; time-aware models explicitly model temporal dynamics to improve relevance (Xiang & Yang, 2009). Time-dependent techniques adjust biases or latent factors as a function of time and can incorporate seasonal effects or recency weight- ing. Session-based recommenders further address short-term intent (e.g., a single browsing session) rather than long- term profiles, which is important for systems where many interactions are one-off or where mood/context vary between sessions.

Conversational and Mood-Aware Recommenders

Conversational recommenders merge dialog systems with recommendation engines to accept free-form, incremental user input and refine suggestions interactively. Such sys- tems can alleviate the cold-start problem by eliciting prefer- ences through natural language instead of relying solely on historical logs. Mood-aware recommenders (for movies or music) specifically map affective user states (explicit mood labels or implicit sentiment) to content attributes (genres, keywords, tempo, valence). Prior surveys and handbooks on recommender systems discuss hybridization strategies (content + collaborative signals) and the potential benefits of context-aware inputs (Ricci, Rokach, & Shapira, 2015). Recent demonstrations show that large language models and semantic embeddings can translate unstructured mood descriptions into meaningful candidate lists, improving initial recommendation quality in conversational settings (Ge´ron, 2021; Mu¨ller & Guido, 2016).

Datasets, Benchmarks, and Evaluation

Practical recommender research relies on public datasets such as MovieLens, which provide ratings, timestamps, and metadata suitable for evaluating CF, matrix factorization, and temporal models (GroupLens Research, 2025). Benchmarks facilitate reproducible comparisons using metrics like Pre- cision@K, Recall@K, NDCG, and ranking-oriented losses. For conversational systems, user studies and human-in-the- loop evaluations are also common to capture subjective relevance and satisfaction.

Engineering, Systems, and Deployment Concerns

Building production-grade recommenders requires software-engineering rigor: modular code, testing, scalable data pipelines, and robust data storage (Pressman & Maxim, 2019; Sommerville, 2015). Database design and transaction semantics are critical for booking systems to avoid double-booking and maintain ACID guarantees for seat reservations (Silberschatz, Korth, & Sudarshan, 2019). Design patterns and architectural guidelines (Gamma et al.; Fowler, 2002) help structure services (e.g., separating model training from serving, using microservices for the chatbot and booking subsystems).

Gaps and Positioning of This Work

While extensive literature exists on CF, matrix factoriza- tion, neural recommenders, and conversational systems, there is less work that integrates mood-driven conversational rec- ommendation with transactional booking and operational an- alytics in a single end-to-end platform. This project positions itself at that intersection: it combines semantic mood inter- pretation (conversational front-end) with metadata-informed ranking, seat-allocation heuristics for real-time booking, and an analytics layer that supports theatre-level operational deci- sions. By integrating these components, MoviesForYou aims to demonstrate a practical pipeline from natural-language intent to completed, time-constrained offline reservations, while producing analytics that close the loop for theatre managers.

III. SYSTEM STRUCTURE AND ANALYTICS

High-level Architecture

MoviesForYou adopts a modular, service-oriented archi- tecture that separates user-facing concerns (UI/UX) from business logic (booking, seat-allocation) and intelligence (chatbot, recommendation scoring). The principal compo- nents are:

- Frontend (Web UI): A single-page application that provides the primary user experience: browsing film

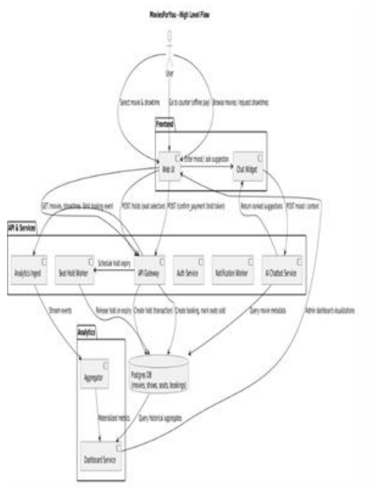
MoviesForYou - High Level Flow
User
Select movie & showtime
Go to counter (offline pay)
Browse movies / request showtimes
Frontend
Web UI
Enter mood / ask suggestion
Chat Widget
GET movies, showtimes, Send booking event
POST holds (seat selection)
POST /confirm payment (hold token)
Return ranked suggestions
POST mood / context
API & Services
Analytics Ingest
Seat Hold Worker
Schedule hold expiry
API Gateway
Auth Service
Notification Worker
AI Chatbot Service
Stream events
Release hold on expiry
Create hold (transaction)
Create booking, mark seats sold
Query movie metadata
Admin dashboard visualizations
Analytics
Aggregator
Postgres DB (movies, shows, seats, bookings)
Materialized metrics
Query historical aggregates
Dashboard Service

Fig. 1. High level Flow Chart

catalogues, interacting with the conversational assistant, selecting showtimes and seats, and viewing simple ana- lytics (personal booking history). The UI must provide immediate visual cues for seat availability and the current hold state to reduce user confusion.
- API & Server Layer: A stateless REST/GraphQL API implements booking workflows, seat-hold seman- tics, showtime queries, and authorization. This layer

- coordinates transactions with the database and queues background jobs for analytics aggregation.
- AI Chatbot Service: A dedicated microservice that accepts natural-language inputs, optionally computes mood/sentiment embeddings, and returns a ranked list
- of candidate movies. The service can operate in two modes: (1) lightweight — using keyword-to-genre mappings and local embeddings for minimal latency and cost; (2) LLM-assisted — invoking an external large model for nuanced interpretation when higher quality is required.
- Database (Transactional Store): A relational database (PostgreSQL recommended) stores canonical entities: movies, theaters, screens, seat layouts, showtimes,
- bookings, and user metadata. Transaction isolation and strict locking/higher-level reservation logic are required to prevent double-booking.
- Analytics Engine: An offline/nearline subsystem that ingests booking events and computes aggregated KPIs. This module exposes a reporting API and renders inter-
- active visualizations for administrative users.
- Auxiliary Services: Background workers (job queues) handle hold expirations, email/SMS notifications, nightly aggregation, and geolocation/travel-time estima- tion (if distance-based hold adjustment is used).

Data Flow and Transaction Semantics

User actions produce two classes of events: read-only queries (browse catalog, request suggestions) and state- changing commands (select seats, place a hold, confirm payment). Read-only queries can be safely routed to replicas or cached layers to reduce latency. State-changing commands must be handled by the primary transactional store and follow an explicit reservation protocol:

- Seat Hold (Thold): When a user selects seats, the API creates a hold record with a unique token, reserved seat IDs, and an expiry timestamp computed as Tnow + Δ, where Δ is the hold-duration. The hold is created within a serializable transaction to ensure seats are atomically marked as unavailable for other concurrent requests.
- Hold Extension / Confirmation: The frontend polls or receives push updates regarding hold status. If the user reaches the counter and confirms payment (offline), the backend transitions the hold to a final booking record and marks seats as sold. If the hold expires, a background worker releases the seats.
- Distance-based Adjustment: Optionally, the hold- duration Δ can be adjusted using coarse distance bands (e.g., within 5 km → extend by 5 minutes). Crucially, only coarse, non-identifying location infor- mation should be used to preserve privacy.

Seat Recommendation Logic (Movie Buddy)

The seat suggestion component ranks available seats using a composite utility score that balances viewing comfort, group adjacency, and occupancy optimization. Key design choices:

- Viewing Comfort: Model seats with geometric coor- dinates (row, column). Prefer seats minimizing lateral offset from the centerline and at row heights that align with screen center for typical auditorium geometry.
- Group Constraints: For multi-seat bookings, search for contiguous blocks whose mean utility is maximized subject to availability constraints.
- Utilization-aware Heuristics: To avoid leaving isolated single seats, the scorer can include a penalty for allo- cations that create inaccessible single vacancies.
- Explainability: The API returns both seat IDs and a concise explanation (e.g., “center row, best view”) to increase user trust.

Analytics Subsystem: Design and Metrics

The analytics subsystem is decoupled from the transac- tional booking flow to protect write throughput and ensure reliable reporting. Booking and seat-event records are emit- ted to a durable event channel (for example, a message queue or WAL stream) and consumed by the analytics pipeline, which produces both nearline summaries and heavier batch aggregates (materialized views) for reporting.[Figure 2]

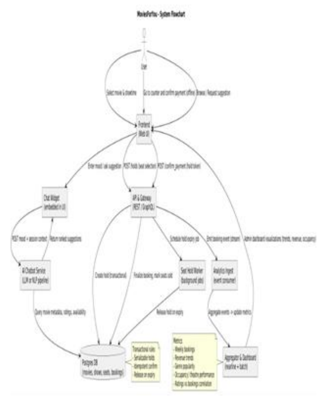
User
Frontend (Web UI)
Chat Widget (embedded in UI)
API & Gateway (REST / GraphQL)
AI Chatbot Service (LLM or NLP pipeline)
Seat Hold Worker (background jobs)
Analytics Ingest (event consumer)
Postgres DB (movies, shows, seats, bookings)
Aggregator & Dashboard (nearline + batch)
Metrics

Fig. 2. System flow chart

Key metrics and their purpose

- Weekly booking patterns: Aggregate bookings by movie, theatre, and timeslot on a weekly basis. Use moving averages and week-over-week comparisons to reveal seasonality and identify peak demand windows for scheduling and staffing.
- Revenue dynamics: Compute daily and weekly cumu- lative revenue series, rolling totals, and growth rates per

- film and per screen. These metrics highlight short-lived spikes and long-tail performers for programming and promotional decisions [figure 3]. [h]

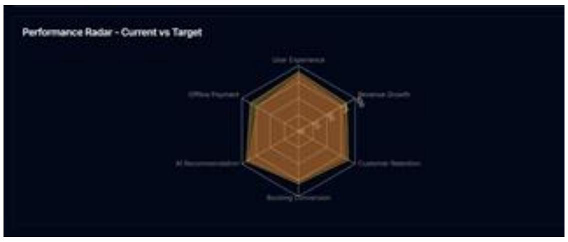

Fig. 3. Performance Radar - Current Vs Target

- Genre demand analysis: Summarize bookings and revenue by genre (apply proportional weighting for multi-

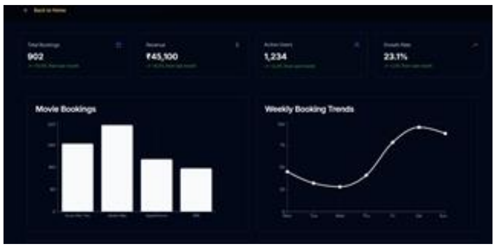

Fig. 4. Analysis Weekly Booking

- genre titles). This informs content acquisition, targeted marketing, and genre-focused scheduling[Figure 4].

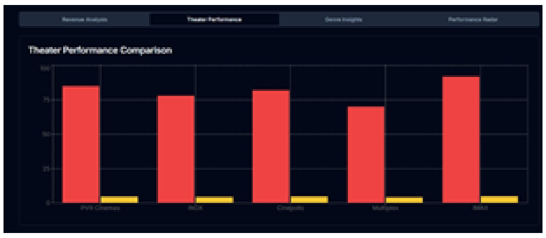

Fig. 5. Genre demand analysis

- Screen utilization and performance: Measure oc- cupancy rate (sold seats ÷ total seats), revenue per seat, and per-show profitability. Present heatmaps across showtimes to optimize screen allocation and frequency.
- Ratings vs attendance correlation: Calculate correla- tion statistics (e.g., Pearson/Spearman) between aggre-
- gated user/critic ratings and actual bookings to surface mismatches where high-rated films underperform or low-rated films attract large audiences.
- Customer cohorts and conversion metrics: Track repeat-customer rates, conversion after chatbot sugges-
- tion, time-to-purchase after recommendation, and coarse location-based cohorts (e.g., distance bands) to evaluate recommendation effectiveness and user behavior[figure 5].

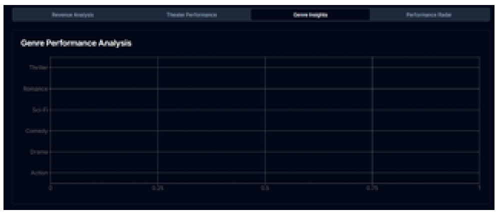

Fig. 6. Generic Performance

- Operational notes: Design the pipeline to support near-real-time operational views (updated every few minutes) and nightly batch reports. Store aggregates in indexed sum- mary tables or materialized views for fast dashboard queries, while retaining raw events for audit and deeper analysis. Ensure event consumers are idempotent and include retry logic so reprocessing does not corrupt aggregates.

Aggregation and Visualization Patterns

Aggregation is performed in two tiers:

- Nearline Aggregates: Updated every few minutes for operational dashboards (e.g., current occupancy, live hold counts).
- Batch Aggregates: Nightly or hourly jobs compute heavier aggregates (weekly trends, cohort analyses) and populate materialized views used by the reporting UI.
- Visualizations should support drill-down: e.g., clicking a high-level genre bar reveals movie-level trends and then theatre-level breakdowns. Common chart types include time- series line charts for revenue, stacked bars for genre compo- sition, heatmaps for occupancy across showtimes, and scatter plots for rating vs. bookings.

Scalability, Consistency, and Fault Tolerance

To scale while preserving booking integrity:

- Use database transactions with optimistic concurrency control for hold creation and finalization. Where ex- treme scale is required, move to a partitioned approach (per-show locking keys) or centralized reservation co- ordinator.
- Cache read-heavy catalogue queries in a CDN or in-memory cache (Redis) with short TTLs to reflect near-real-time seat changes.
- Implement idempotent APIs and durable job queues so that background workers can safely retry hold expira- tions and aggregate jobs without data corruption.

- Provide graceful degradation: if the AI Chatbot Service is slow or unavailable, fall back to a deterministic keyword-mapping recommender to maintain respon- siveness.

Privacy, Security, and Compliance
The system stores personal information and booking his- tories. Best practices include:

- Minimize retention of raw mood text; prefer ephemeral processing and store only derived preference signals when consent is granted.
- Use parameterized SQL and input validation to prevent injection; secure APIs with authentication and role- based authorization.
- Encrypt sensitive fields at rest where required by local law, and ensure secure transmission (TLS) between components.
- Provide users with explicit controls to delete their conversational logs and booking data in compliance with privacy regulations where applicable.

Operational Considerations
Operationally, key SLAs and configuration knobs include:

- Hold-duration default (e.g., 10–15 minutes) config- urable per theatre or showtime.
- Max concurrent holds per user to prevent hoarding.
- Analytics freshness targets (e.g., nearline for opera- tions, nightly for trend analysis).
- Monitoring and alerts for anomalies such as sudden booking surges, hold expiry backlogs, or significant discrepancies between rating and booking signals.
- Collectively, these architectural and process choices pro- vide a robust foundation for a mood-driven, conversational movie booking system that also supplies actionable analytics to theatre managers and content curators.

IV. IMPLEMENTATION

Technologies and Libraries
The implementation of MoviesForYou follows a modu- lar full-stack web development approach, combining mod- ern frontend frameworks, scalable backend APIs, robust databases, and machine learning powered recommendation services. The technology stack is selected to ensure scalabil- ity, maintainability, and integration of analytics and AI-based features.

- Frontend: The user interface is developed as a Sin- gle Page Application (SPA) using React.js or Vue.js. Styling and responsive layouts are handled by Tail- windCSS or Bootstrap frameworks. For visualization of booking trends, revenue curves, and genre-based analytics, libraries such as Chart.js, D3.js, or Highcharts are employed to generate interactive charts and dash- boards[figure 6].
- Backend/API Layer: The core application logic is im- plemented using Node.js with Express.js or alternatively Python-based frameworks such as Django and Flask.
- Django, in particular, offers a built-in ORM, admin interface, and structured approach for rapid prototyping (Django Software Foundation, 2025). The backend is responsible for managing business rules such as seat reservation logic, hold timers, and interaction with the AI chatbot service.
- Database: PostgreSQL serves as the relational database engine for transactional booking data, showtimes, and theater layouts. It provides ACID properties critical
- for seat allocation and booking consistency. For per- formance optimization, materialized views and indexed queries are used for analytics queries. In larger-scale de- ployments, horizontal sharding or caching layers (e.g., Redis) may be added to support high traffic.
- AI Service: The mood-based recommendation module is powered by a language understanding system. This may include pretrained Large Language Models (LLMs)
- such as GPT variants, or lightweight intent recognition pipelines built with libraries like spaCy, NLTK, or Hug- gingFace Transformers. The AI maps natural-language mood descriptions (e.g., “nostalgic,” “thrilling”) into

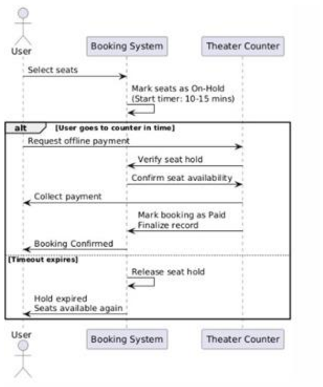

Fig. 7. Sequence Diagram

movie metadata (genres, ratings, availability) and pro- duces a ranked suggestion list. The AI component can be hosted as a microservice or integrated via serverless endpoints for scalability.

- Analytics & Visualization: Analytical insights are computed through batch aggregation jobs or SQL-based ETL pipelines. PostgreSQL queries are used to pro-

- duce metrics such as weekly booking volumes, revenue trends, and occupancy ratios. The frontend renders these insights in graphical dashboards using libraries like Chart.js or Recharts. For advanced analytics, integration with Python data science libraries (Pandas, NumPy, Matplotlib) or big-data frameworks (Apache Spark) can be considered.
- Deployment and Environment: The system can be containerized using Docker for reproducible deploy-
- ments. Orchestration using Kubernetes may be adopted in larger-scale scenarios. CI/CD pipelines ensure au- tomated testing and deployment. Static assets of the frontend can be hosted via CDN, while the backend and database can be deployed on cloud providers such as AWS, GCP, or Azure.
- Offline Payment Workflow: Unlike most commercial systems, the prototype enforces an offline payment
- model. The backend holds selected seats for a time- limited window (based on distance and travel estimates). This requires scheduled tasks (e.g., Celery in Django or Bull in Node.js) to release expired reservations and maintain booking integrity.

The combination of these technologies ensures that the platform not only provides a seamless booking experience for end-users but also integrates advanced analytics and conversational AI capabilities for enhanced discoverability and operational intelligence.

Chatbot Recommendation Pipeline

The conversational chatbot is the core innovation of the platform, enabling mood-aware and context-sensitive movie recommendations. The pipeline integrates Natural Language Processing (NLP), recommendation algorithms, and meta- data filtering. It consists of the following stages:

- Input Normalization: User queries are pre-processed through tokenization, lemmatization, and stop-word removal. Sentiment and mood terms are extracted using either lexicon-based methods or embedding similarity. For example, “I feel nostalgic and want something calm” maps to affective states associated with Drama or Romance genres.
- Context Enrichment: Session-level preferences are incorporated, including past searches, language pref- erences, or age restrictions. Contextual constraints im- prove personalization beyond a single utterance.
- Candidate Retrieval: The system queries the movie database for all films satisfying immediate constraints such as availability, showtimes, and parental ratings. This ensures that the candidate set only contains movies that can actually be booked.

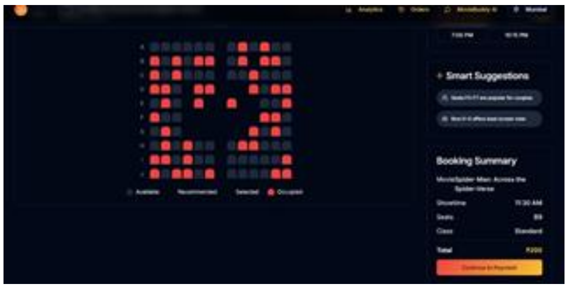

Fig. 8. Seat Suggestion

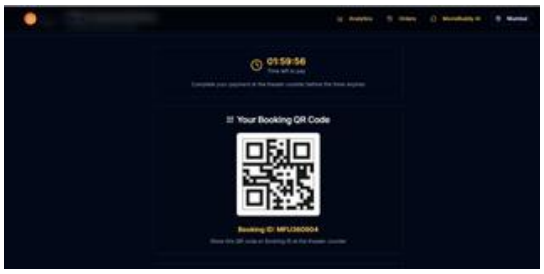

Fig. 9. Booking Time Limit

erences. The desirability of a seat is computed based on proximity, centrality, and adjacency [Figure 8]:

Scoring and Ranking: Each candidate is scored
based on a weighted combination of multiple sig-

Score(s) = w1

· 1 − |xs−xcenter | +w2

· 1 − |ys−yopt|

nals: Score(m) = α · ContentSim(m, u) + β · Popularity(m) + γ · MoodMatch(m, u)
where:

- ContentSim(m, u) measures similarity between movie m and user profile u (e.g., genres, key- words).
- Popularity(m) captures booking counts and user ratings.
- MoodMatch(m, u) quantifies the alignment be- tween the detected mood of the user and genre- affective embeddings.
- Parameters α, β, γ are tuned empirically to balance
- + w3 · AdjacencyBonus(s)
- where:
- xs, ys are the coordinates of seat s.
- xcenter is the central column of the screen.
- yopt is the optimal row (e.g., mid-hall row for acoustics).
- AdjacencyBonus(s) gives additional weight to seats that are contiguous in group bookings.

- w1, w2, w3 are tunable weights reflecting user preferences.
- For group bookings, the algorithm searches for contiguous seat blocks that maximize the average desirability score:
- Σ

content similarity, popularity, and mood alignment.
Response Generation: The top-N ranked items are

GroupScore(G) =
|G| s∈G Score(s) (1)

formatted as natural-language responses by the chatbot (e.g., "Based on your mood, you may enjoy Movie A or Movie B, both available this evening.").
Technically, the mood-to-genre mapping can be imple- mented through a dictionary-based affective lexicon, or em- beddings trained with models such as Word2Vec, BERT, or sentence transformers (Ge´ron, 2022). For higher accuracy, a lightweight LLM can dynamically infer mood-contextual recommendations at runtime.

Seat Suggestion (Movie Buddy)
The "Movie Buddy" module provides personalized seat recommendations to optimize user comfort and group pref- where G is the set of seats in the group. The top-K groups are then suggested to the user.

Booking and Offline Payment Handling
Unlike typical online booking systems, MoviesForYou adopts an offline payment workflow. The booking pipeline operates as follows:

- Seat Hold: When a user selects one or more seats, they are marked as "on hold" in the database. A timer Thold is assigned to each seat to prevent double booking.
- Payment Confirmation: If the user completes the payment at the physical counter before Thold expires, the status transitions from held to confirmed.
- Timeout Release: If Thold elapses without confirma- tion, the hold is automatically released and the seats return to the pool of available inventory.
- Distance-Aware Adjustment: Optionally, the hold duration can be personalized based on user proximity to the theater:

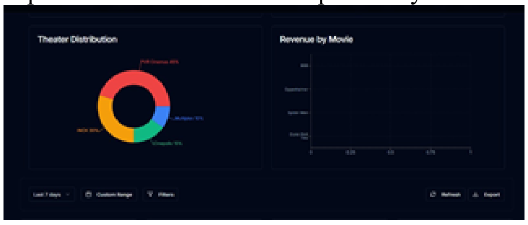

Fig. 10. Theater distribution and revenue analysis dashboard.

Booking Timer and QR Code
Figure 11 illustrates the unique booking timer mechanism. After seat selection, users receive a limited time (here set to just under 2 hours) to complete payment at a theater counter due to the offline-only payment setting. Upon suc- cessful booking, a QR code and unique Booking ID (e.g., MFU360904) are generated for ticket redemption, providing both security and convenience.

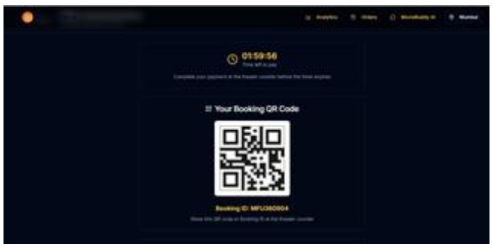

Fig. 11. Booking timer and QR code for ticket redemption.

Theater Performance Comparison
Figure 12 shows a bar chart comparing different theater chains (PVR, INOX, Cinepolis, Multiplex, IMAX). Red bars illustrate bookings/revenue, validating the analytics capabil- ities of the system.

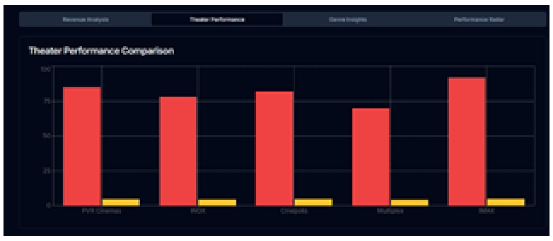

Fig. 12. Comparison of theater performance across chains.

Dashboard Analytics: Bookings, Revenue, and Trends
Figure 13 presents a comprehensive dashboard snapshot with total bookings, revenue, user growth, and trends. Bar charts visualize bookings across movies (e.g., Dune 2, Spider-Man, Oppenheimer, RRR), while trend lines confirm peak activity on weekends.

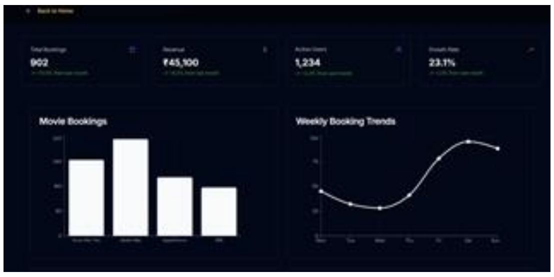

Fig. 13. Dashboard analytics showing bookings, revenue, and weekly trends.

Performance Radar
Figure 14 uses a radar chart to assess KPIs like user experience, AI recommendations, customer retention, and booking conversion. It highlights improvement areas and overall balance in platform performance.

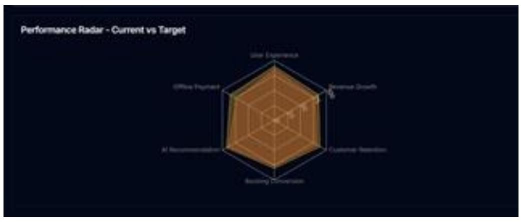

Fig. 14. Radar chart showing KPI performance benchmarks.

Booking Page: Details, Theater Proximity, and Payment Info
Figure 15 shows the booking confirmation screen in- cluding movie details, nearby theaters, and offline payment instructions. Clear steps increase user trust and transparency

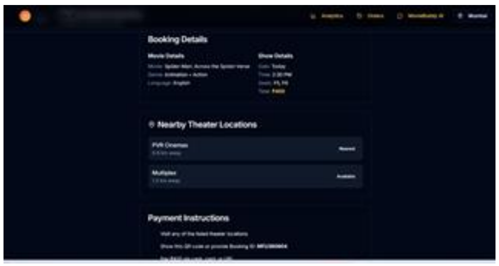

Fig. 15. Booking confirmation with theater proximity and payment info.

Seat Recommendation and Smart Suggestions
Figure 16 shows the interactive seat map with availability, recommendations, and “Smart Suggestions” (e.g., optimal viewing rows), demonstrating personalized seating intelli- gence.

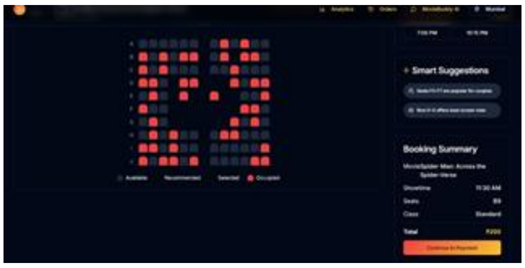

Fig. 16. Seat recommendation and smart suggestions in real-time.

Revenue Trend Analysis
Figure 17 visualizes financial growth over days, confirm- ing the effect of AI-driven recommendations on revenue increase.

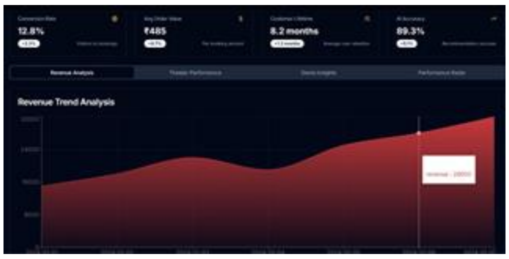

Fig. 17. Revenue trend analysis over multiple days.

Genre Performance Analysis
Figure 18 presents genre popularity analysis, correlating mood preferences with genre success. Though partial, it emphasizes extensibility.

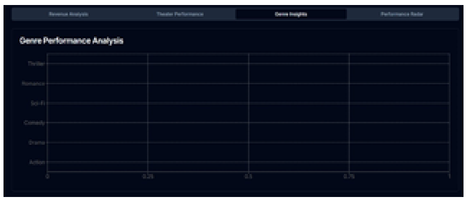

Fig. 18. Genre performance analytics for mood-genre correlation.

Main Page and AI Chatbot
Figure 19 shows the MovieBuddy AI chatbot that provides mood-based suggestions and remembers preferences. Quick access to genres and search functions improves engagement.

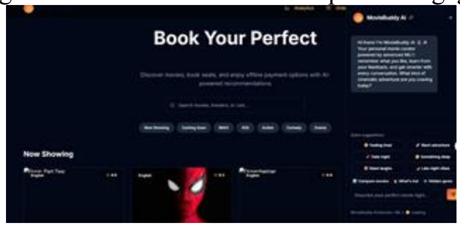

Fig. 19. Main page with MovieBuddy AI chatbot for mood-based recommendations.

VI. CONCLUSION

This paper presented MoviesForYou, a web-based movie booking platform enhanced with an AI-powered conversa- tional chatbot, automated seat recommendations, and an inte- grated analytics subsystem. The system successfully demon- strates how mood-based natural language inputs can be transformed into personalized movie suggestions, improving user experience beyond traditional keyword search.

The inclusion of a seat recommendation engine, designed with heuristic scoring and adjacency considerations, en- hances the booking flow by offering context-aware and group-friendly seat allocations. The offline payment work- flow, with time-limited seat holds and QR-based booking verification, provides a practical alternative for deployment in regions with limited digital payment adoption.

The analytics module further extends the platform by generating actionable insights for theater managers, including booking trends, revenue trajectories, genre popularity, and performance comparisons across cinema chains. These re- sults validate the platform's ability to act as both a consumer- facing booking system and a decision-support tool for stake- holders.

Overall, the MoviesForYou prototype highlights the syn- ergy of conversational AI, recommendation pipelines, and analytics in modern entertainment platforms. Future exten- sions may include integrating online payments, real-time dynamic pricing, and advanced AI techniques (e.g., rein- forcement learning for recommendations) to further optimize both customer satisfaction and operational efficiency.

Acknowledgement
We sincerely thank our Project Guide, ANAND JAWDEKAR , for his valuable guidance and encouragement throughout this work. We are also grateful to Prof.Dr. Shailendra Kumar Mishra (HoD) and our project co- ordinators for their consistent support. Finally, we extend our appreciation to Dr. Vipul Vekeriya (Principal) for providing the facilities and opportunities that enabled the successful completion of this project.MoviesForYou project. Special thanks are extended to the academic department for providing the necessary resources and technical environment to design, implement, and evaluate the system.

The authors also acknowledge the use of open-source libraries, frameworks, and tools that enabled the successful integration of artificial intelligence, analytics, and web tech- nologies within this platform. Finally, heartfelt appreciation is given to family and friends for their constant support and motivation during the preparation of this research work.

REFERENCES

1. [Shivanand, S., & colleagues. (2020). Chatbot with music and movie recommendation based on mood. International Journal of Engineer- ing Research & Technology (IJERT), NCAIT 2020. Retrieved from https://ijert.org
2. [Nicoomanesh, A. (2025, March 4). Building local LLM-powered movie recommendation chatbot with Gemma 2 and LangChain.
Medium. Retrieved from https://medium.com
3. Filmgrail Blog. (2023, December). Cinema ticketing system analytics for smarter marketing. Filmgrail. Retrieved from https://filmgrail.com
4. Konstantinovsky, T. (2024, September 2). How to choose the best seat in a movie theater. Stackademic Blog. Retrieved from https://blog.stackademic.com
5. Highcharts Blog. (2016, March). Visualizing and comparing the highest-grossing films. Highcharts. Retrieved from https://www.highcharts.com/blog
6. ReelMind Blog. (2024–2025). Movie booking: AI for entertainment ticketing. ReelMind. Retrieved from https://reelmind.ai
7. Goodfellow, I., Bengio, Y., & Courville, A., "Deep Learning", MIT Press, 2016. Retrieved https://www.deeplearningbook.org.
8. Bishop, C. M., "Pattern Recognition and Machine Learning", Springer, 2006.
9. Mitchell, T. M., "Machine Learning", McGraw-Hill, 1997.
10. Ge´ron, A., "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.)", O'Reilly Media, 2022.
11. Mu¨ller, A. C., & Guido, S., "Introduction to Machine Learning with Python", O'Reilly Media, 2016.
Chollet, F., "Deep Learning with Python (2nd ed.)", Manning Publi- cations, 2021.
12. Koren, Y., Bell, R., & Volinsky, C., "Matrix Factorization Techniques for Recommender Systems", IEEE Computer, vol. 42, no. 8, pp. 30– 37, 2009. doi:10.1109/MC.2009.263.
13. Ricci, F., Rokach, L., & Shapira, B. (Eds.), "Recommender Systems Handbook (2nd ed.)", Springer, 2015.
14. Django Software Foundation, "Django documentation", 2025. Re- trieved from https://docs.djangoproject.com.

15. [TensorFlow, “TensorFlow: Open-source machine learning framework”, 2025. Retrieved from https://www.tensorflow.org.
16. GroupLens Research, “MovieLens dataset”, 2025. Retrieved from https://grouplens.org/datasets/movielens/.
17. Pressman, R. S., & Maxim, B. R., “Software Engineering: A Practi- tioner’s Approach (9th ed.)”, McGraw-Hill, 2019.
18. Sommerville, I., “Software Engineering (10th ed.)”, Pearson, 2015.
19. Silberschatz, A., Korth, H. F., & Sudarshan, S., “Database System Concepts (7th ed.)”, McGraw-Hill, 2019.
20. Gamma, E., Helm, R., Johnson, R., & Vlissides, J., “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1994.
21. Fowler, M., “Patterns of Enterprise Application Architecture”, Addison-Wesley, 2002.
22. Ge´ron, A., “Building Machine Learning Powered Applications: Going from Idea to Product”, O’Reilly Media, 2021.
23. Xiang, L., & Yang, Q., “Time-dependent models in collaborative filtering recommendation”, in Proceedings of the 2009 SIAM Inter- national Conference on Data Mining, pp. 485–496, SIAM, 2009. doi:10.1137/1.9781611972795.42.
He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S.,
24. “Neural Collaborative Filtering”, in Proceedings of the 26th Inter- national Conference on World Wide Web, pp. 173–182, ACM, 2017. doi:10.1145/3038912.3052569.
Sarwar, B., Karypis, G., Konstan, J., & Riedl, J., “Item-based col- laborative filtering recommendation algorithms”, in Proceedings of the 10th International Conference on World Wide Web, pp. 285–295, ACM, 2001. doi:10.1145/371920.372071.