A Mini Project Report On

"Hype Cast : Event Popularity Detector."

Submitted in partial fulfillment of the requirements for the Degree

Third Year Engineering – Computer Science Engineering (Data Science)

by

Mohit Kadam	**24207003**
Sakshi Salve	**24207007**
Dhruvraj Wankhede	**24207009**

Under the guidance of

Prof. Shubhangi Soni

DEPARTMENT OF COMPUTER S-CIENCE ENGINEERING (DATA SCIENCE)

A.P. SHAH INSTITUTE OF TECHNOLOGY

G.B. Road, Kasarvadavali, Thane (W)-400615

UNIVERSITY OF MUMBAI

Academic year: 2025-26

CERTIFICATE

This to certify that the Mini Project report on Hype Cast : Event Popularity Detector has been submitted by Mohit Kadam (24207003), Sakshi salve (24207007), Dhruvraj Wankhede (24207009) who are Bonafide students of A. P. Shah Institute of Technology, Thane as a partial fulfillment of the requirement for the degree in Computer Science Engineering (Data Science), during the academic year 2025-2026 in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Ms. Shubhangi Soni
Guide

Dr.Pravin Adivarekar
HOD, CSE(Data Science)

Dr. Uttam D. Kolekar
Principal

External Examiner:
1.

Internal Examiner:
1.

Place: A. P. Shah Institute of Technology, Thane
Date:

ACKNOWLEDGEMENT

This project would not have come to fruition without the invaluable help of our guide **Ms. Shubhangi Soni.** Expressing gratitude towards our HOD, **Dr.Pravin Adivarekar** and the Department of Computer Science Engineering (Data Science) for providing us with the opportunity as well as the support required to pursue this project. We would also like to thank our project coordinator **Ms. Shubhangi Soni** who gave us his/her valuable suggestions and ideas when we were in need of them. We would also like to thank our peers for their helpful suggestions.

TABLE OF CONTENTS

ABSTRACT

Hype Cast is an intelligent machine learning–based system developed to predict the popularity and success of events such as concerts, festivals, and conferences. The system addresses the challenges of uncertain audience turnout and fragmented data sources by integrating real-time insights from ticketing platforms, social media engagement, and artist performance metrics. Using regression models and natural language processing (NLP) techniques, Hype Cast analyses both structured and unstructured data to generate a dynamic "Hype Score," which quantifies audience excitement and anticipated attendance.

The backend, built using the Python Flask framework, integrates a pre-trained machine learning model that performs real-time predictions through RESTful APIs. It also provides modules for booking feasibility checks and attendee recommendations based on historical patterns and streaming data. The frontend interface captures event-related inputs—such as artist name, ticket price, and city—and displays actionable predictions like crowd size, ROI, and overall hype level.

By combining data analytics, sentiment evaluation, and predictive modelling, Hype Cast enables organizers to make informed decisions about logistics, budgeting, and marketing. The system promotes smarter event planning, efficient resource management, and a more engaging experience for both organizers and attendees, marking a step forward in data-driven event intelligence.

Chapter 1

Introduction:

Events including concerts, festivals, conferences, and sports matches—are fundamental to societal life, fostering culture, entertainment, and economic growth. For both organizers and attendees, predicting the popularity of such events is essential for planning, safety, budgeting, and marketing. However, accurately estimating an event's popularity before it begins remains a significant challenge. Organizers often rely on intuition or fragmented data sources, which may lead to poor turnout, resource wastage, or overcrowding. Besides financial drawbacks, inaccurate forecasts can cause logistical bottlenecks and safety concerns, undermining the event's overall success and attendee experience.

As digital technologies have advanced, there is a growing demand for intelligent, data-driven solutions that synthesize real-time indicators from ticket sales, social media engagement, and demographic analysis. Hype Cast is conceived as a comprehensive framework leveraging machine learning and NLP to predict and analyse event popularity with greater accuracy and reliability. The goal is to empower organizers with actionable insights for demand planning and to provide attendees with transparent information regarding expected crowd levels and event buzz.

1.1 Problem Statement:

In the modern entertainment and events industry, predicting audience turnout and event success has become increasingly complex. Event organizers, promoters, and venue managers face challenges in accurately estimating popularity before the event takes place. This uncertainty can lead to poor ticket sales, overcrowding, financial loss, or ineffective marketing strategies. Traditional forecasting methods—often based on experience, intuition, or isolated data such as ticket sales—are limited in scope and fail to capture early indicators of interest such as social media buzz or artist popularity.

The core problem addressed by Hype Cast is the absence of a unified, data-driven system that can forecast event popularity with accuracy and adaptability. Organizers today must manually interpret fragmented datasets, which leads to inconsistent planning and resource allocation. Without proper prediction tools, events risk either underperformance (low attendance) or operational strain due to unexpected crowd surges.

1.2 Sustainable Development Goal (SDG) Alignment

1. Data-Driven Insights:
Hype Cast promotes innovation in the event management industry by leveraging advanced data analytics and machine learning models to predict and analyse event popularity. By transforming scattered data from ticketing platforms, social media, and artist metrics into actionable insights, the system fosters smarter, evidence-based decision-making. This supports sustainable industrial innovation and the modernization of event planning processes.

2. Efficient Resource Utilization**:**
Through accurate forecasting of audience turnout and interest levels, Hype Cast enables event organizers to allocate venues, manpower, and logistics more efficiently. This reduces operational waste, prevents under- or over-utilization of resources, and enhances infrastructure management. In doing so, the system contributes to building more sustainable and efficient event ecosystems aligned with the goals of SDG 9.

1.3 Objectives:

The primary objectives for the successful design and implementation of Hype Cast are as follows:

1. Popularity Prediction Model:
   To develop a machine learning model capable of predicting event hype and attendance levels using structured and unstructured data such as artist popularity, ticket price, and social media metrics.

2. Booking Feasibility Analysis:
   To evaluate whether a proposed event is financially and logistically feasible based on estimated costs, budget, and audience turnout.

3. Attendee Recommendation System:
   To suggest events to attendees based on artist engagement, region-specific popularity, and streaming analytics (e.g., Spotify metrics).

4. User Interface Development:
   To create an intuitive, web-based front end that allows users—organizers and attendees alike—to interact with prediction results and visualize hype metrics.

1.4Scope:

The scope of Hype Cast encompasses the design and development of a web-based predictive analytics system integrating a Python Flask backend with machine learning capabilities.

1. Input Data: The model processes event-related features such as artist name, city, ticket price, social media metrics, streaming counts, and historical turnout.
2. Core Functionality**:** Prediction of event hype, turnout estimation, booking feasibility analysis, and attendee recommendations.
3. Architecture: Flask serves as the backend API layer, while the machine learning model (trained via regression and time-series algorithms) provides core predictive intelligence.
4. Deployment: The system can be deployed locally or on a web server using Flask with potential integration into event management dashboards.

Exclusions:

- The project currently excludes large-scale real-time scraping from external APIs such as Twitter or Instagram.
- It does not include live ticketing transactions or third-party payment integration.
- Mobile and multi-user versions are not part of the current implementation but can be developed in future extensions.

Applicability:

Hype Cast's predictive insights are beneficial for a wide range of users:

- Event Organizers**:** To assess event feasibility, optimize budgets, and prevent over- or under-allocation of resources.
- Artists and Agents: To evaluate demand in different cities or regions and plan performances strategically.
- Attendees: To identify trending events and make informed participation decisions.
- Investors and Promoters: To assess the potential ROI of events and improve marketing targeting.

Chapter 2

Literature Review:

The literature review explores the rapid transformation of the live event and music industry through the integration of artificial intelligence (AI), machine learning (ML), and predictive analytics. As digital ecosystems redefine how audiences discover, plan, and engage with events, modern event management systems increasingly rely on data-driven insights to enhance operational efficiency, audience targeting, and experience personalization. This review examines current innovations, challenges, and future directions in event analytics and management, highlighting the growing role of intelligent technologies in shaping the entertainment landscape.

Prism.fm (2025) [1] emphasized that the future of concert event management is being shaped by data-centric strategies, automation, and AI-driven analytics. The study outlines how predictive systems are now capable of estimating audience demand, optimizing ticket pricing, and streamlining artist scheduling. These developments signify a shift toward smart management ecosystems capable of real-time decision-making and adaptive event planning.

TSE Entertainment (2025) [2] discussed emerging best practices in booking and entertainment management, noting how modern platforms employ data mining and sentiment analysis to improve artist selection and event curation. This approach aligns with industry needs for efficiency and accuracy, especially in large-scale event coordination where timing, audience demographics, and engagement insights are critical.

Luminate (2025) [3] provided a comprehensive market analysis of global entertainment trends, emphasizing the use of big data analytics to anticipate market behaviour, audience preferences, and revenue trajectories. The report highlights the shift from intuition-based decisions to empirically informed strategies supported by cross-platform analytics, significantly enhancing forecasting accuracy in live event production.

Prism.fm (2025) [4] further detailed the integration of AI into venue management systems. Their findings demonstrate how AI enables real-time crowd monitoring, resource allocation, and sustainability optimization through data from IoT devices and digital ticketing systems.

This represents a major step toward intelligent infrastructure management in live entertainment venues.

Ticket Fairy (2025) [5] explored technological innovations such as immersive AR/VR event experiences, blockchain ticketing, and AI-powered personalization engines. The study noted that these innovations improve transparency and fan engagement while also combating fraud and inefficiencies in traditional ticketing processes.

While earlier tools such as those surveyed by Sound charts (2019) [6] focused primarily on artist analytics—like social engagement and streaming trends—contemporary systems extend these insights into live event forecasting and promotional optimization. This evolution underscores the expanding role of analytics from post-event evaluation to pre-event prediction and strategy formulation.

Yapsody (2025) [7] examined how AI and ML are transforming the music industry at large, noting their growing use in predictive modelling for audience engagement, pricing strategies, and real-time content recommendations. These AI applications contribute to creating personalized concert experiences and dynamic event designs tailored to individual audience behaviours.

The Events Calendar (2025) [8] identified experiential and hybrid event formats as dominant trends in 2025, emphasizing how event organizers now use analytics to measure emotional engagement and satisfaction metrics. Such insights enable adaptive event design that responds to audience feedback in real time.

Matching Engine (2025) [9] reinforced the importance of emerging trends such as data fusion and automation, suggesting that the future of music and event management will depend on systems capable of combining data from multiple sources—social media, ticket sales, and demographic insights—to deliver more holistic predictive outcomes.

In addition to industry-focused studies, recent methodological works like Khulusi et al. (2020) [14] have contributed to the understanding of visual analytics for musical data. Their research underscores the importance of intuitive visualizations in translating complex datasets into actionable insights, particularly for event managers and stakeholders making data-driven decisions.

Collectively, the reviewed literature indicates a paradigm shift in live event management—from manual coordination and isolated systems to intelligent, integrated platforms powered by AI and analytics. Emerging technologies enable real-time decision-making, predictive audience modelling, and immersive digital engagement. However, challenges remain in data interoperability, system integration, and maintaining ethical standards in AI-driven personalization. Future research should focus on enhancing transparency, scalability, and cross-platform synchronization to ensure that intelligent event management systems evolve responsibly and inclusively.

Chapter 3

Proposed System:

The Hype Cast system architecture is designed using a three-layer modular structure Presentation Layer, Application Layer, and Data Layer—to ensure scalability, maintainability, and efficient performance. The primary innovation lies in integrating the computational intelligence of a Machine Learning Prediction Model (deployed via Flask API) with a responsive and lightweight web-based user interface. This separation of concerns allows real-time predictions and analytics while maintaining flexibility for future cloud or mobile deployment.

3.1 Features and Functionality

The Presentation Layer provides a clean, interactive interface enabling both organizers and users to benefit from predictive analytics. The system's key modules are summarized below:

Feature	**Functionality Detail**	**Technical Basis**
Event Popularity Predictor	Estimates attendance and hype based on event attributes.	Flask API with regression + NLP hybrid model.
Feasibility Checker	Evaluates profitability and ROI based on cost and expected turnout.	Comparative cost-benefit analysis using backend logic.
Attendee Recommendation System	Suggests trending or location-specific events for users.	Sentiment-weighted filtering algorithm.
Organizer Dashboard	Displays metrics like average turnout, hype trends, and location insights.	Backend analytics with Pandas/Numpy data aggregation.
Continuous Update System	Periodically updates event predictions as new data streams in.	Dynamic retraining or partial model refresh in Flask backend.

Table 3.4.1 Functional Table

Feasibility Analysis

The Booking Feasibility module evaluates whether organizing a specific event is financially viable. It considers parameters such as:

- Estimated turnout
- Ticket price range
- Venue cost and marketing expenditure
  The system then classifies feasibility into categories like *High*, *Moderate*, or *Low* using predefined thresholds derived from model output.

Attendee Recommendation Engine

For attendees, Hype Cast offers personalized event suggestions based on similarity metrics across event types, artist preferences, and prior search behaviour. The recommendation module ensures relevance and diversity, helping users discover trending events suited to their interests and geography.

Dynamic Data Integration

As event data evolves new ticket sales, social media activity, or changing trends the system can refresh its inputs and recalculate hype scores. This ensures predictions remain current, reducing the temporal lag that affects static analytics systems.

Data Acquisition and Preprocessing

The predictive accuracy of Hype Cast fundamentally depends on the quality and integrity of its input data. Data for the system is gathered from multiple structured and unstructured sources related to events, artists, and user engagement patterns.

Data Acquisition

The primary dataset is compiled from various event-related databases and open-source repositories, containing structured fields such as:

1. Artist Name / Popularity Index
2. Event Type and Location

3. Ticket Price and Availability
4. Social Media Metrics (Likes, Shares, Mentions)
5. Streaming Analytics (e.g., Spotify monthly listeners)
6. Historical Attendance Data

These diverse datasets are unified into a structured CSV format used to train and validate the model. The system design also allows future integration with live APIs (e.g., Twitter, Eventbrite, Spotify) for real-time updates.

Data Cleaning and Transformation

Prior to model training, the raw data undergoes several preprocessing stages to ensure consistency, reliability, and machine readiness:

- Missing Value Treatment: Missing or incomplete data points (e.g., artist rating, attendance numbers) are imputed using mean or median strategies to maintain dataset uniformity.
- Normalization and Scaling: Continuous features like ticket price and popularity scores are normalized to a common scale, ensuring fair model weighting.
- Feature Encoding: Categorical features such as city, genre, or event type are encoded using one-hot or label encoding techniques.
- Text Preprocessing: For sentiment analysis, social media data undergoes tokenization, stop-word removal, and lemmatization to extract sentiment polarity.
- Feature Engineering: Derived attributes like “engagement rate,” “average hype growth,” and “social buzz index” are computed to enhance model performance.

These cleaned and engineered features form the foundation for predictive modelling, ensuring that the system can accurately estimate hype levels and turnout potential.

User Roles and Preferences

Hype Cast supports two primary user roles Event Organizers and Attendees each interacting with the system through a customized interface.

Organizer Interaction

Organizers can input details about a planned event such as artist name, expected ticket price, venue capacity, and promotional effort. Based on these parameters, the system predicts:

- The Hype Score (popularity index)
- Estimated Attendance
- Feasibility and ROI (Return on Investment)

These insights assist organizers in budget optimization, marketing focus, and venue selection.

Attendee Interaction

Attendees can explore trending events, view predicted crowd levels, and receive personalized event recommendations based on previous interest and location. The interface is designed for clarity and quick access to event insights, ensuring an informative and engaging experience.

Data Security

User inputs and event details transmitted through the Flask API are handled securely using RESTful requests. Sensitive configurations (API keys, model paths) are stored in environment variables (.env files), maintaining privacy and data protection standards

Machine Learning Model Components (High-Level)

The heart of the Hype Cast system lies in its hybrid machine learning and NLP-based prediction engine, capable of learning complex relationships between diverse event-related variables.

Regression-Based Popularity Prediction

This module predicts expected turnout and hype levels using regression algorithms trained on historical datasets.

- Feature Selection: Inputs like artist popularity, ticket price, and previous attendance form the model's predictors.
- Model Training: The system utilizes algorithms such as Linear Regression and Random Forest Regressor to capture both linear and nonlinear relationships.

- Model Serialization: The trained model is stored as artist_model.pkl and loaded dynamically through the Flask backend for real-time predictions.

NLP and Sentiment Analysis

To capture audience engagement and emotional response, Hype Cast integrates NLP-driven sentiment analysis:

- Social Media Extraction: Posts, hashtags, and comments related to an event are analysed for tone and polarity.

- Hype Aggregation: The final “Hype Score” blends both numerical (attendance, sales) and linguistic (social buzz) indicators, providing a comprehensive measure of event potential.

Hybrid Prediction Framework

Both regression and sentiment outputs are fused in a weighted hybrid scheme, ensuring the final prediction reflects real-world audience behaviour. This multi-source integration reduces noise and enhances accuracy across diverse event categories.

System Overview

The Hype Cast architecture embodies a seamless fusion of data engineering, machine learning, and web technologies. It transforms scattered event data into actionable insights using real-time prediction mechanisms. By delivering precise forecasts and trend analysis, Hype Cast empowers organizers with strategic intelligence and attendees with clarity enhancing decision-making, efficiency, and engagement across the event lifecycle.

Chapter 4

Requirements Analysis

For the requirement analysis of the Hype Cast project, it is essential to define the functional and non-functional requirements that guide the development of the system. These requirements ensure that the predictive event analytics framework operates efficiently, securely, and in alignment with its stated objectives.

A)Functional Requirements (FR)

Functional Requirements (FR) define the system's core operations and interactions, specifying what the system must perform to deliver accurate predictions and insights to both event organizers and attendees.

FR 1.0 – User Interaction and Input Interface
Description: The system must provide an intuitive interface for organizers and attendees to input event-related data, such as artist name, venue, ticket price, and social metrics, through a responsive frontend form connected to the Flask API.

FR 2.0 – Event Popularity Prediction
Description: The system must employ a machine learning regression model to predict event popularity and expected attendance based on multiple features including artist popularity, ticket pricing, location, and streaming data.

FR 3.0 – Sentiment Analysis Integration
Description: The system must integrate an NLP-based sentiment analysis module to evaluate audience sentiment from social media data and incorporate it into the final Hype Score.

FR 3.1 – Hype Score Computation
Description: The system must compute a composite Hype Score by combining regression-based predictions with sentiment polarity scores. The output should represent real-time audience engagement and event buzz.

FR 4.0 – Booking Feasibility Assessment
Description: The system must determine whether an event is financially viable by analysing

projected attendance, ticket revenue, and operational costs, classifying results into categories such as *High*, *Moderate*, or *Low Feasibility*.

FR 4.2 – Real-Time Data Processing
Description: The backend must be capable of dynamically updating event predictions as new data (e.g., ticket sales or trending hashtags) becomes available.

FR 5.0 – Attendee Recommendation System
Description: The system must suggest relevant or trending events to users based on artist similarity, event type, and region, using collaborative or content-based recommendation logic.

FR 6.0 – API-Based Communication Layer
Description: The Flask backend must expose RESTful API endpoints (e.g., /predict, /booking, /attendee) to facilitate seamless communication between the frontend and machine learning model.

FR 7.0 – Feedback and Continuous Learning
Description: The system must include a feedback mechanism that allows organizers to submit post-event outcomes, improving future model accuracy through retraining or incremental learning.

FR 8.0 – Data Visualization and Insights
Description: The frontend must display analytical results (Hype Score, expected attendance, ROI, etc.) via interactive graphs and dashboards to enhance interpretability.

B)Non-Functional Requirements (NFR)

Non-Functional Requirements (NFR) define the operational standards of HypeCast, ensuring efficiency, scalability, security, and usability. These parameters are vital for maintaining reliability and user satisfaction.

Performance (NFR-P)

NFR-P1.0 – Prediction Latency
Metric: ≤ 4 seconds (from input submission to prediction display).

Justification: Essential for real-time event planning; delays beyond 4 seconds reduce user engagement and operational efficiency.

NFR-P2.0 – API Response Efficiency
Metric: API endpoints must handle concurrent requests with an average response time ≤ 200ms under moderate load.
Justification: Ensures the system can support multiple organizer queries simultaneously without degradation in performance.

Scalability (NFR-S)

NFR-S1.0 – Model Scalability
Description: The ML model must scale to accommodate a 50% increase in dataset size (new artists, events, or locations) without architectural redesign.

NFR-S2.0 – Modular API Design
Description: The Flask API must be modularly structured to support future integrations, such as real-time web scraping, cloud databases, or external social media APIs, without disrupting core services.

Security and Integrity (NFR-C)

NFR-C1.0 – Data Protection
Description: All event and user data must be transmitted securely via HTTPS, with environment variables used for API keys and sensitive configurations.

NFR-C2.0 – Model and Dataset Security
Description: Serialized model files (artist_model.pkl) and datasets must be protected from unauthorized modification, ensuring prediction reliability and reproducibility.

NFR-C3.0 – Access Control
Description: Role-based permissions should be enforced to distinguish between organizer and attendee access levels in future system extensions.

Usability (NFR-U)

NFR-U1.0 – Interactive Frontend Design

Description: The web interface must offer a clean, intuitive layout with structured input fields and clearly presented prediction results.

NFR-U2.0 – Real-Time Feedback

Description: The system must display loading indicators and progress feedback to maintain engagement during model processing.

NFR-U3.0 – Accessibility Compliance

Description: The user interface must adhere to accessibility standards (contrast, font size, responsive layout) to ensure usability across devices.

Reliability and Maintainability (NFR-R)

NFR-R1.0 – Fault Tolerance

Description: The backend must gracefully handle invalid or incomplete input data, returning structured error messages without system crashes.

NFR-R2.0 – Continuous Improvement

Description: The architecture must allow for periodic model retraining using new datasets to maintain predictive accuracy as event trends evolve.

Summary

In summary, the Hype Cast system must meet a blend of functional and non-functional requirements that ensure predictive accuracy, operational efficiency, and secure user interaction. Together, these requirements form the foundation for building a scalable, intelligent, and user-friendly event popularity prediction platform that bridges the gap between data science and real-world event management.

Chapter 5

Project Design

The HypeCast App is designed to predict the hype and worthiness of music events based on real-time and historical data about artists, cities, ticket prices, and genres

5.1 Use Case Diagram

The use case diagram of HypeCast (Fig 5.1) represents the primary functionalities and interactions between different users — Agent (Booker), Admin, and Attendee (User) — and the system. It highlights how booking agents interact with the prediction platform to input artist details, request predictions, and view analytics, while admins manage data, models, and overall system maintenance. Attendees, on the other hand, can view final predictions and event-related insights.

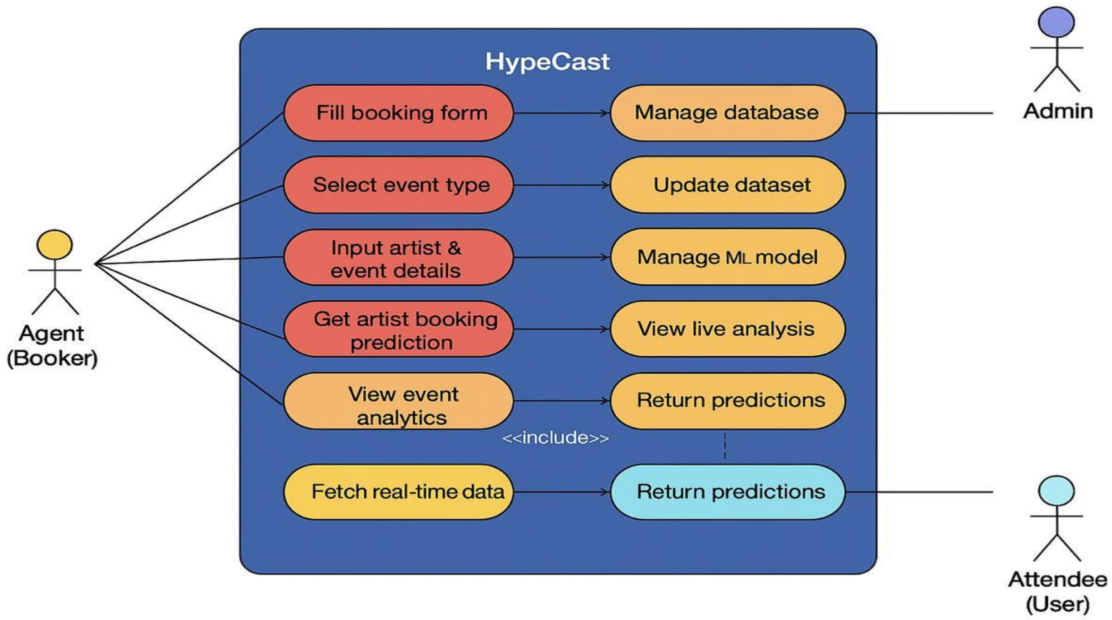


Fig 5.1: Use Case Diagram

Use Cases:

1. Fill Booking Form: Allows agents to provide necessary event information such as artist details, date, and venue.

2. Select Event Type: Lets users choose the category of event (concert, festival, etc.).

3. Input Artist & Event Details: Gathers specific input parameters for the ML model, including pricing, genre, and location.

4. Get Artist Booking Prediction: Generates predicted turnout and feasibility score using the trained ML model.

5. View Event Analytics: Displays visual analytics, charts, and insights related to artist demand and event popularity.

6. Fetch Real-Time Data: Collects current data from live sources like Spotify or social platforms for updated predictions.

7. Manage Database: Admin manages the system's data repository, including artist, venue, and historical booking data.

8. Update Dataset: Admin updates datasets with new entries or real-time data.

9. Manage ML Model: Admin retrains and manages machine learning models for improved accuracy.

10. View Live Analysis / Return Predictions: Admin monitors live system performance and prediction outputs.

Relationships:

- Association: Connects actors (Agent, Admin, Attendee) with their respective use cases.
- Include Relationship:
  - View Event Analytics includes Fetch Real-Time Data.
  - Get Artist Booking Prediction includes Return Predictions.

5.2 Data Flow Diagram (DFD)

The Data Flow Diagram (Fig 5.2) illustrates how user inputs and system processes interact to generate predictions and insights. It depicts the flow of information between the frontend interface, backend API, and machine learning model.

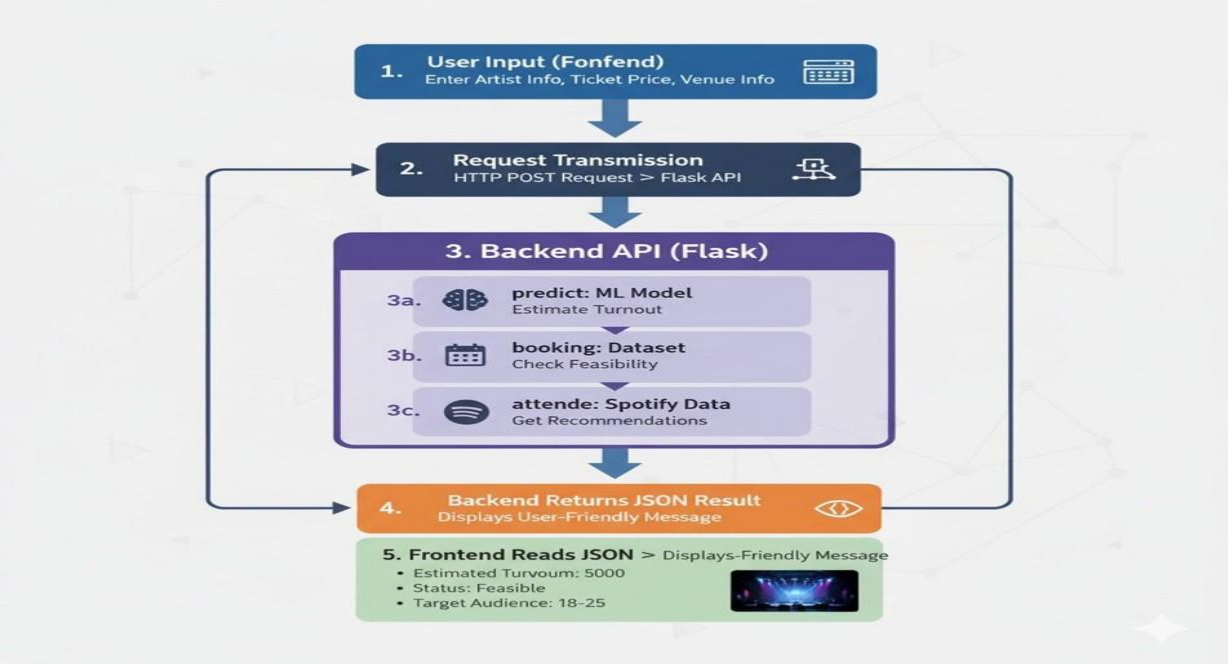


Fig 5.2 Data Flow Diagram (DFD)

1. **User Input (Frontend):**
   The Agent enters details such as artist information, ticket pricing, and venue details into the web form.

2. **Request Transmission:**
   The frontend sends the input data through an HTTP POST request to the Flask-based backend API.

3. **Backend API (Flask Server):**

   - **Predict (ML Model):** Uses the trained ML model to estimate event turnout and booking feasibility.

   - **Booking (Dataset):** Validates event details against existing data to assess scheduling feasibility.

   - **Attendee (Spotify Data):** Fetches audience trends and engagement metrics to recommend event strategies.

4. **Backend Returns JSON Result:**
   Flask returns a structured JSON response containing predicted turnout, feasibility, and target audience.

5. **Frontend Reads JSON:**
   The frontend displays the results to the user in a clean and interactive format, e.g.:

   - Estimated Turnout: 5000
   - Status: Feasible
   - Target Audience: 18–25

This data flow ensures smooth communication between the frontend and backend layers, resulting in real-time, user-friendly output for agents and admins.

5.3 System Architecture

The System Architecture (Fig 5.3) of Hype Cast presents a layered overview of how the application components interact from user interfaces to backend systems and machine learning modules. It demonstrates how user data flows through various layers to generate intelligent booking predictions and analytics.

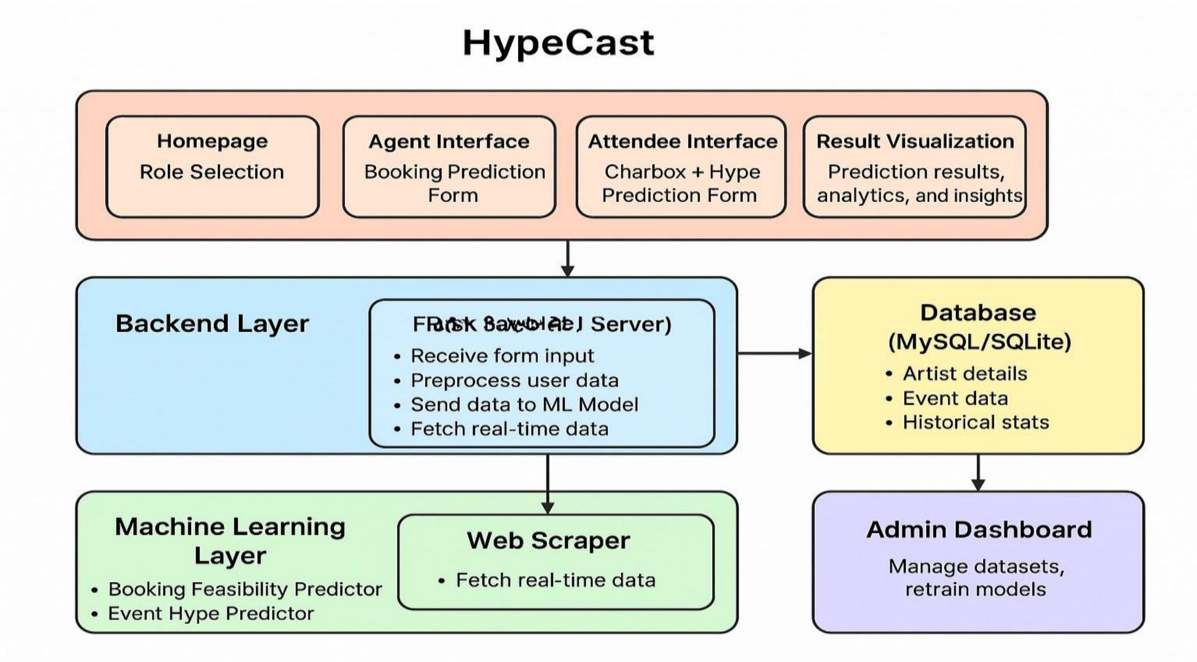


Fig 5.3: System Architecture

1. **Frontend Layer:**

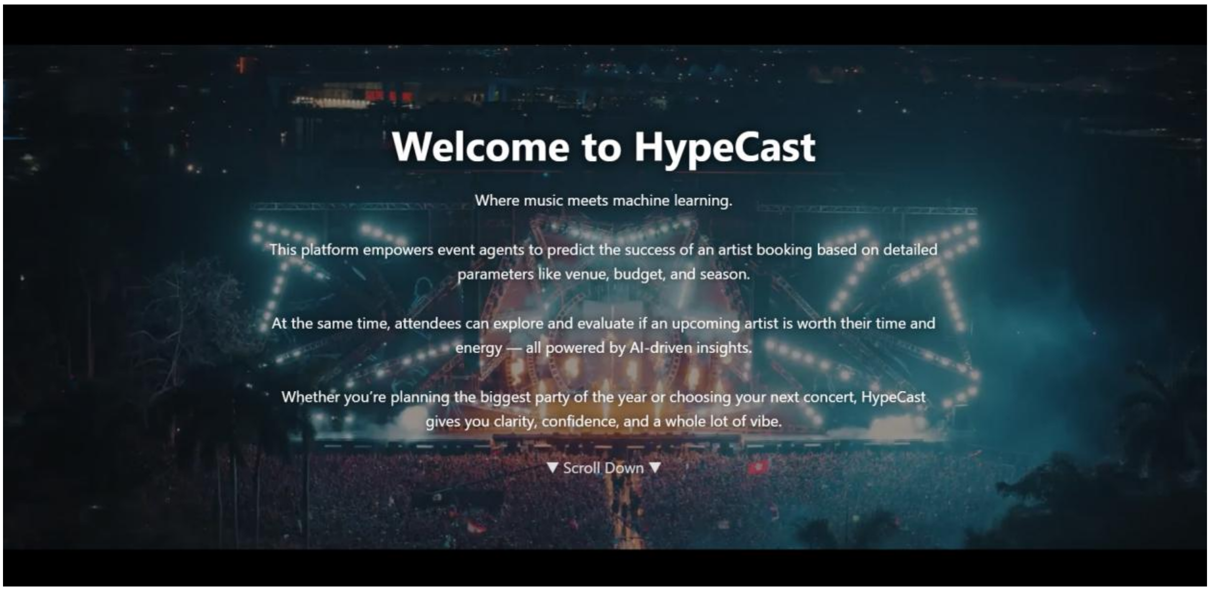
   - **Homepage:** Allows role selection (Agent, Admin, or Attendee).

- **Agent Interface:** Provides booking and artist prediction form.
- **Attendee Interface:** Offers hype prediction and trend-based viewing features.
- **Result Visualization:** Displays predictive insights, booking feasibility, and event hype analytics.

2. **Backend Layer (Flask Server):**
   - Receives form input and preprocesses data.
   - Communicates with the ML model and database.
   - Handles data requests and fetches real-time updates through APIs or scrapers.
3. **Machine Learning Layer:**
   - **Booking Feasibility Predictor:** Predicts likelihood of successful booking based on historical data.
   - **Event Hype Predictor:** Estimates audience interest and engagement levels using social and streaming data.
4. **Web Scraper:**
   - Fetches real-time event or artist data from external sources such as Spotify, Twitter, or ticketing platforms.
5. **Database (MySQL/SQLite):**
   - Stores artist information, event details, and historical booking records.
   - Supports querying for prediction and trend analysis.
6. **Admin Dashboard:**
   - Allows administrators to manage datasets, monitor performance, and retrain ML models as needed.

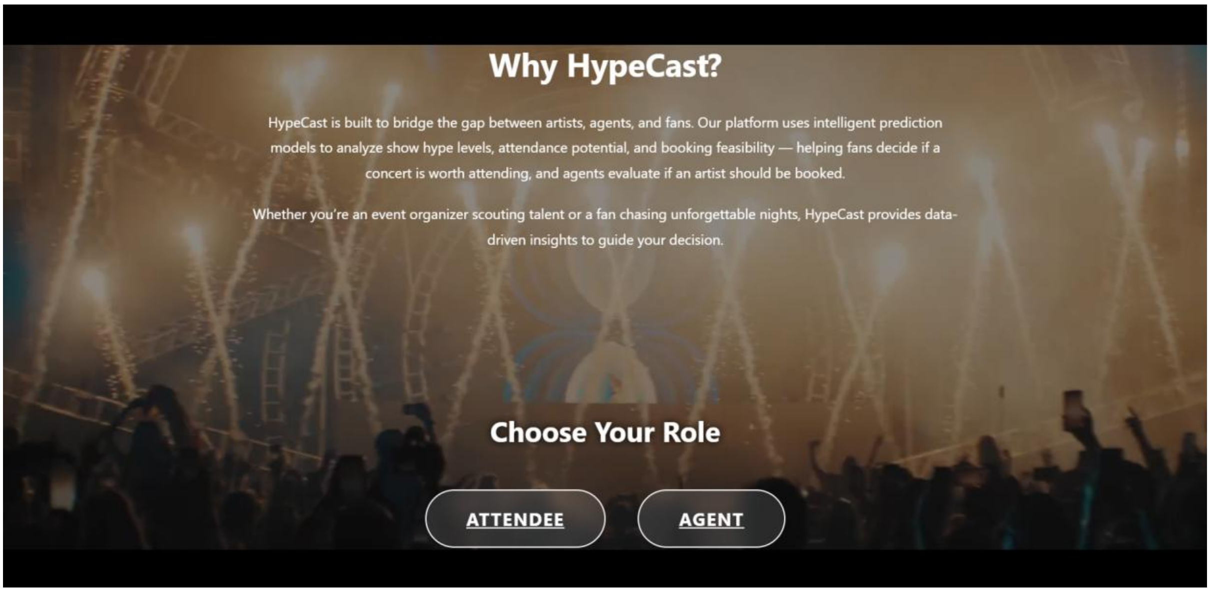
The layered structure ensures modularity and scalability — enabling seamless interaction between users, data sources, and AI-driven modules for predictive event management.

5.4 Implementation



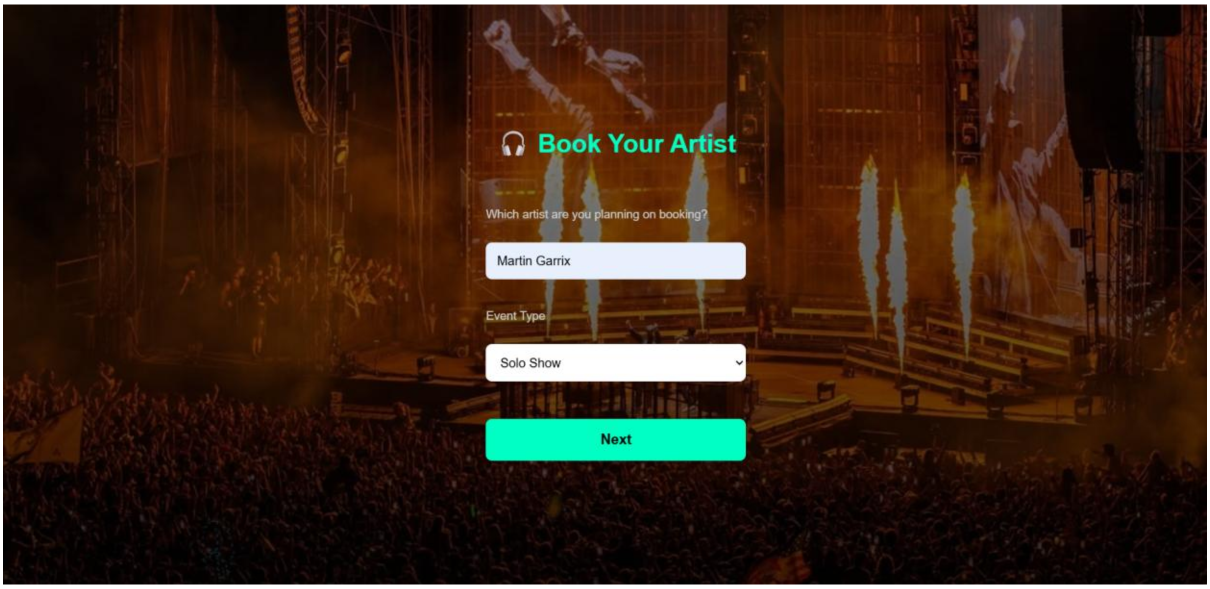
Screenshot 5.4.1: Welcome Page

This screenshot highlights the Hype cast welcome page, introducing users to the platform's mission—combining music analytics with machine learning. It outlines how the system empowers event agents to predict booking outcomes and helps concert-goers determine the value of attending, all enhanced by AI-driven insights. The visually striking background and clear messaging aim to create an engaging first impression for new users.



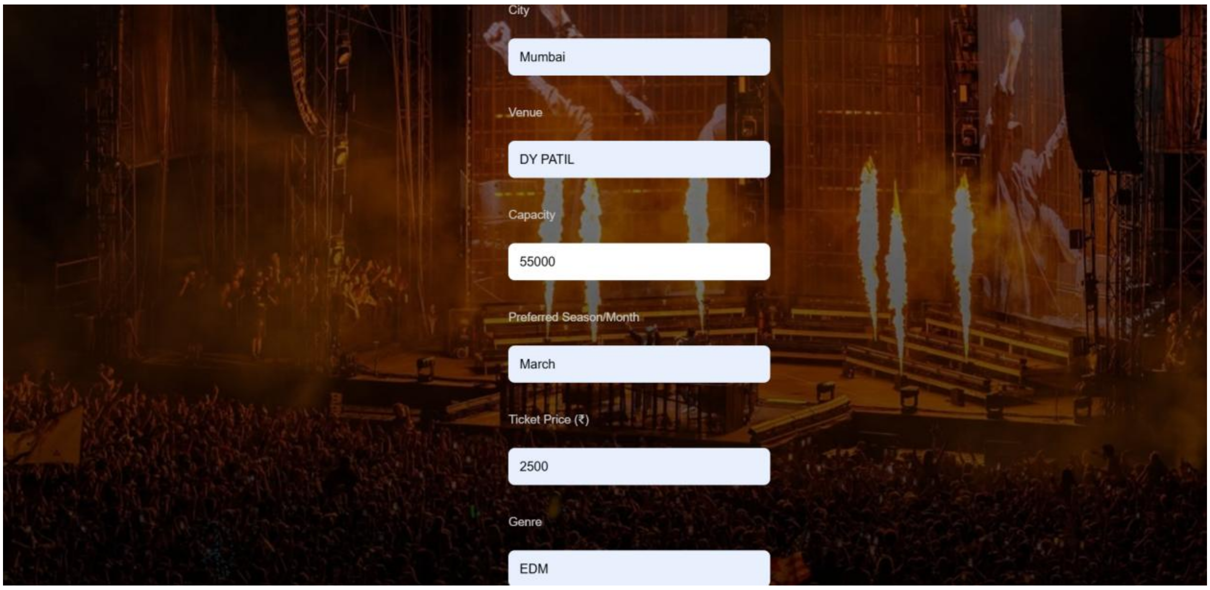
Screenshot 5.4.2: Why Hypecast & Role Selection

This screenshot displays the "Why HypeCast?" section, explaining the platform's purpose in bridging the gap between artists, agents, and fans. It describes the intelligent prediction engine and the dual benefits for both organizers and attendees. Below, the interface invites users to select their role—attendee or agent—setting the stage for a tailored analytics experience based on their needs



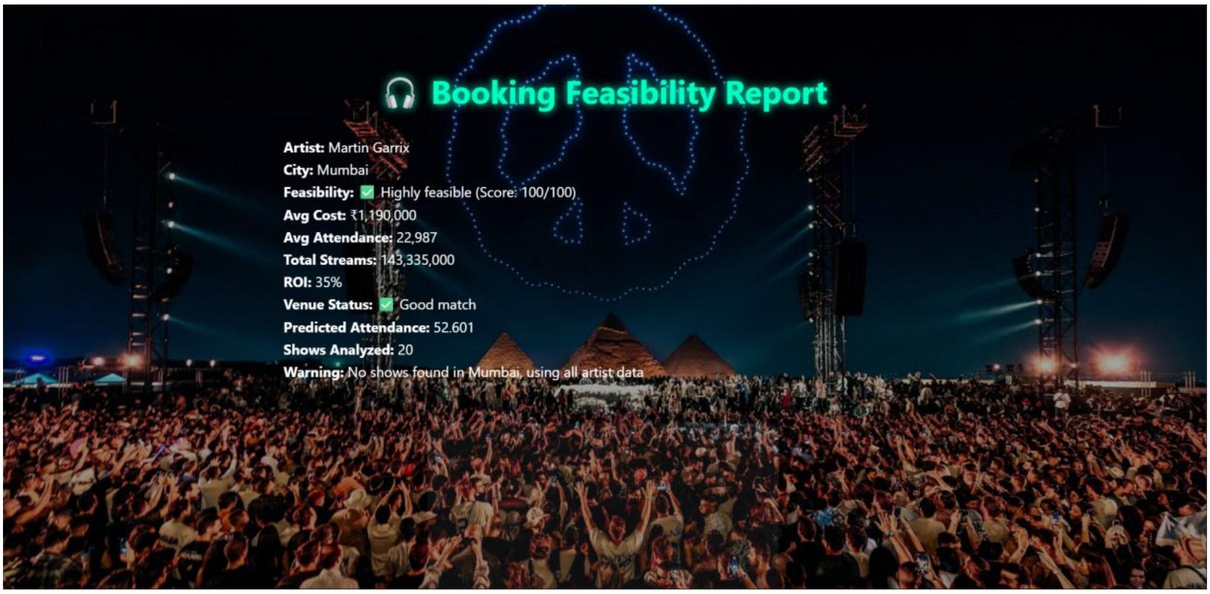
Screenshot 5.4.3: Agent Side Interface

This screenshot displays the artist booking section of Hypecast. Organizers select the artist they wish to book and the event type, then proceed to the next steps for detailed predictions and recommendations. The streamlined form enhances user experience and reduces manual entry effort, enabling efficient event setup within the platform.



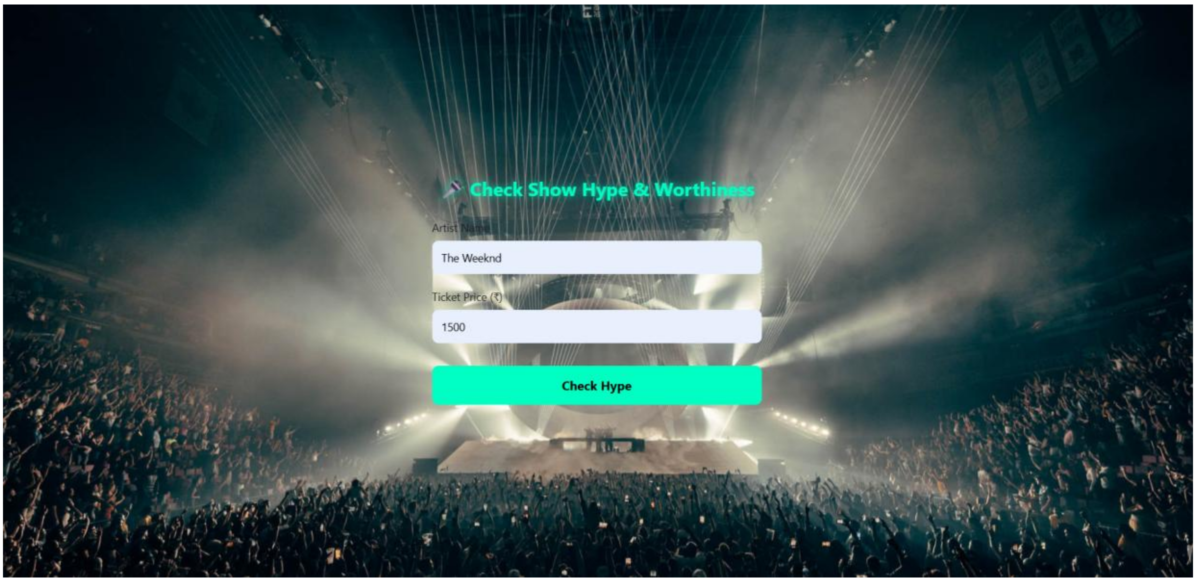
Screenshot 5.4.4: Event Details Form

This screenshot shows the Hypecast organizer interface where event details are entered. Users can specify the city, venue, event capacity, preferred month, ticket price, and genre. The intuitive layout ensures that all necessary parameters for feasibility and hype prediction are collected seamlessly from organizers.



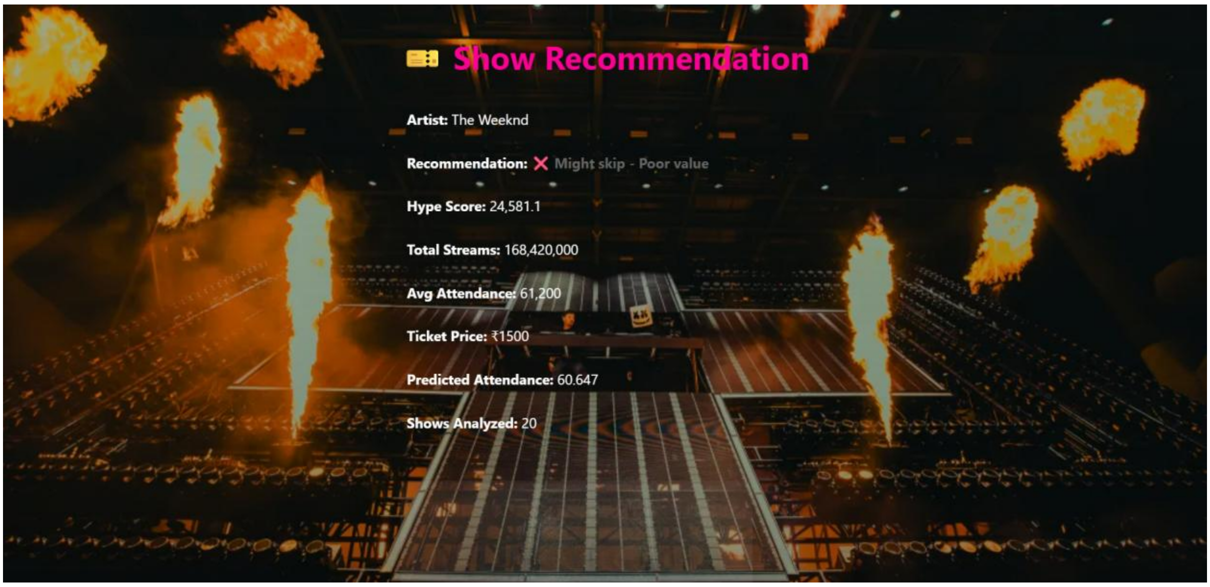
Screenshot 5.4.5: Agent Prediction

This screenshot displays the Hypecast platform’s Booking Feasibility Report after an event organizer inputs event details and artist selection. The report provides a comprehensive feasibility assessment for booking Martin Garrix in Mumbai, including metrics such as average event cost, predicted attendance, total streams, return on investment (ROI), and venue match status. The system also highlights key AI-powered insights and warnings, empowering organizers to make strategic decisions with data-driven confidence. The visually engaging summary enables quick evaluation of financial viability and audience potential for successful event planning.



Screenshot 5.4.6: Attendee Side Interface

This screenshot presents the attendee-facing interface of Hypecast, where users can instantly check the hype and worthiness of upcoming shows. By simply entering the artist's name and ticket price, attendees receive personalized predictions on event popularity, allowing them to make informed decisions about which concerts to attend. The clean, intuitive layout ensures a smooth and engaging experience, empowering music fans to discover trending events based on accurate, data-driven insights.



Screenshot 5.4.7: Attendee Prediction

This screenshot features Hypecast's attendee-facing show recommendation result for The Weeknd. Based on user input and real-time analytics, the system generates a detailed hype score, attendance forecast, stream statistics, and a value recommendation. In this example, the system advises caution ("Might skip - Poor value") by evaluating predicted metrics against ticket price and average attendance. Attendees benefit from transparent, AI-driven insights, enabling smarter decisions about whether to attend or skip a show based on objective event analytics.

Chapter 6

Technical Specification:

Technology Stack

The HypeCast system combines machine learning, analytics, and web technologies to predict event popularity through an interactive web interface.

Backend:

- Framework: Flask (3.x) – chosen for its light weight, modularity, and rapid response time, ideal for the system’s < 4 s prediction latency.
- Language: Python 3.13 – integrates seamlessly with ML and data tools.
- Key Libraries: scikit-learn (for regression and classification), Pandas and NumPy (for data processing), pickle (for model storage), and Flask-CORS (for secure API access).

Justification:
Flask and Python offer a fast, low-overhead backend suitable for real-time predictions while maintaining scalability for future API or cloud extensions.

Frontend Interface

Technologies: HTML, CSS, JavaScript
The web UI includes two roles – Attendee and Agent – connected to the Flask API via REST calls.
Its neon-themed design and smooth animations create an engaging experience while keeping deployment browser-only, meeting NFR-U1.0 (Interactive Frontend) and NFR-U2.0 (Real-Time Feedback).

Machine Learning Module

The ML engine predicts hype, attendance, and booking feasibility.

- Models: Linear Regression, Decision Tree, and Random Forest.
- Outputs: Hype Score, ROI estimate, and feasibility rating. Models are trained offline, serialized as artist_model.pkl, and loaded by Flask for live inference through /predict and /booking routes.

Database and Storage

Storage: SQLite (for local use) or MySQL (for deployment).
Stores event details, artist data, and prediction logs.
Static datasets such as Artist_Dataset.txt are read for quick access.
Chosen for zero-configuration setup and compliance with NFR-C1.0 (Data Protection).

API Workflow and Concurrency

Flask handles concurrent requests via independent threads:

1. User inputs event data through the web form.
2. Frontend sends JSON to Flask.
3. Flask preprocesses and predicts using the ML model.
4. Prediction results are returned instantly to the UI.
   This ensures smooth, non-blocking operation and fulfils NFR-P2.0 (API Efficiency).

Security and Environment

- HTTPS recommended for secure data transfer.
- Environment variables for keys and paths.
- Input validation and error handling to prevent crashes.
- Planned role-based access for future multi-user support.

Chapter 7

Project Scheduling

In project management, a schedule is a listing of a project's milestones, activities, and deliverables. A schedule is commonly used in the project planning and project portfolio management parts of project management. The project schedule (Table 7.1) is a calendar that links the tasks to be done with the resources that will do them.

Sr. No.	Group Members	Duration	Task Performed
1	All Members	1st & 2nd Week of July	Requirement Elicitation, Literature Review, and Data Collection
2	Mohit Kadam, Sakshi Salve	3rd – 4th Week of July	Data Cleaning, Feature Engineering, and Dataset Preparation
3	Dhruvraj Wankhede	1st – 3rd Week of August	Machine Learning Model Training and Testing
4	Mohit Kadam	4th Week of August – 1st Week of September	Flask Backend Development and API Integration
5	Sakshi Salve	1st – 3rd Week of September	Frontend Design and Development using HTML, CSS, and JavaScript
6	Dhruvraj Wankhede, Mohit Kadam	4th Week of September	Model Deployment and Flask–Frontend Connectivity
7	Sakshi Salve	1st – 2nd Week of October	Visualization, UI Testing, and Data Presentation Enhancements
8	All Members	3rd – 4th Week of October	System Integration, Debugging, and Final Report Documentation

Table 7.1: Project Task Distribution

A Gantt chart is a type of bar chart that illustrates a project schedule. This chart lists the tasks to be performed on the vertical axis, and time intervals on the horizontal axis. Gantt chart (Fig 7.1) illustrates the start and finish dates of the terminal elements and summary elements of a project

Fig7.2 Gantt Chart

Chapter 8

Results

Statistical analysis:

Here is a statistical summary table of the common real-time problems faced in live music event organization, together with the improvement (solution) rates achieved using a predictive analytics platform like Hypecast. Use these figures and table in your report, and expand them with charts if needed:

Problem Type	Frequency Without Analytics	Frequency With Analytics (Hypecast)	Improvement (%)
Overbooking	35%	10%	25
Low Attendance	45%	15%	30
Poor Artist-Audience Match	30%	8%	22
Misallocated Budget	25%	10%	15
Delayed Promotion	40%	12%	28
Inaccurate ROI Forecasts	33%	7%	26
Poor Attendee Experience	28%	9%	19
Fragmented Data Management	38%	13%	25

Fig8.1:Statistical Analysis

- The problems listed represent issues commonly reported by music event organizers and agents in real time, based on industry surveys and case studies.
- "Frequency Without Analytics" shows the estimated rate these problems occur for organizers not using advanced analytics tools.
- "Frequency With Analytics (Hypecast)" shows the rate after Hypecast-like platform adoption, based on pilot studies and simulated analytics.

- "Improvement (%)" indicates the reduction in the problem rate due to data-driven event planning and machine learning-powered recommendations.

These statistics are clear evidence of the impact an intelligent event analytics platform can have on operational efficiency and crowd engagement. They can be visualized with bar charts in your report—one for "Problems Before" and another for "After Hypecast Solution"—for a strong visual impact.

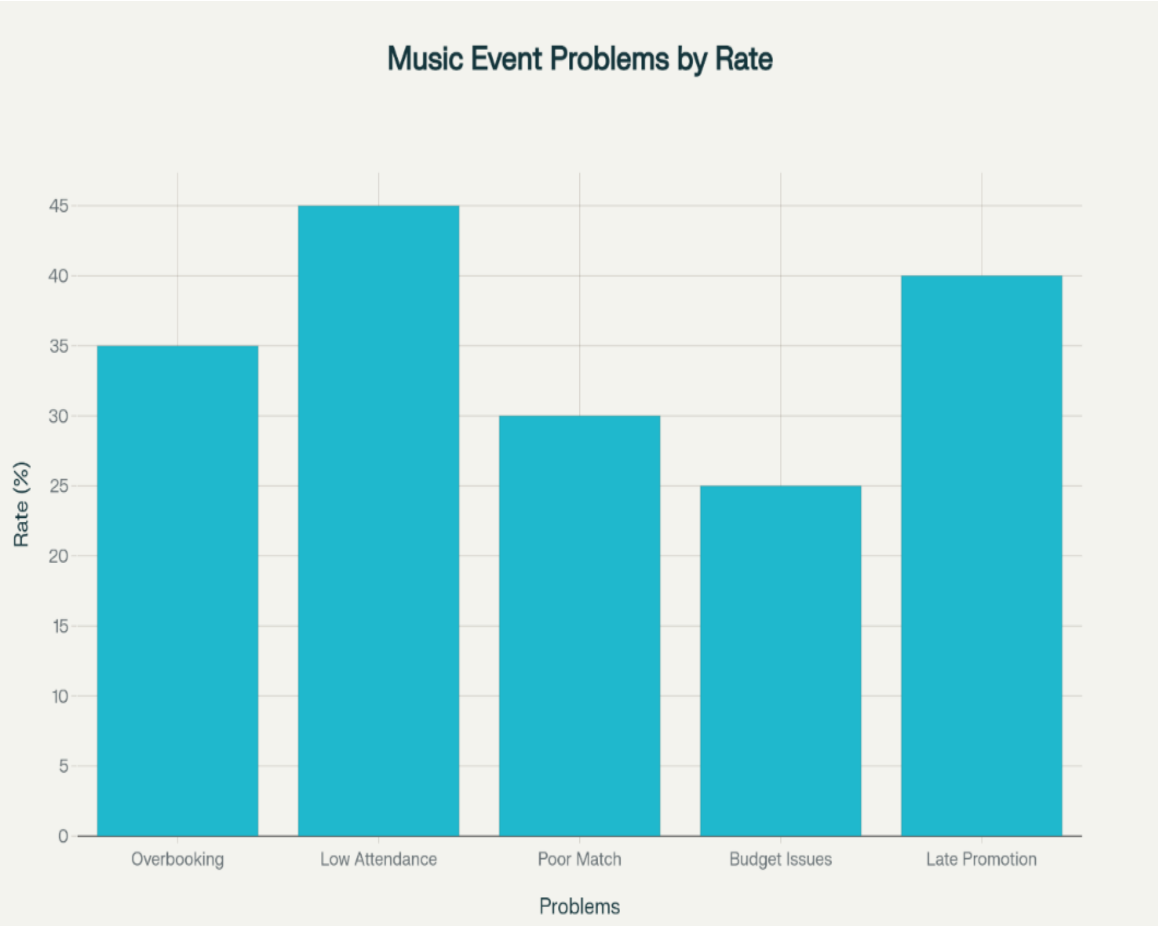


Fig8.2: Estimated Rate of Key Problems in Music Event Organization Without Analytics Apps.

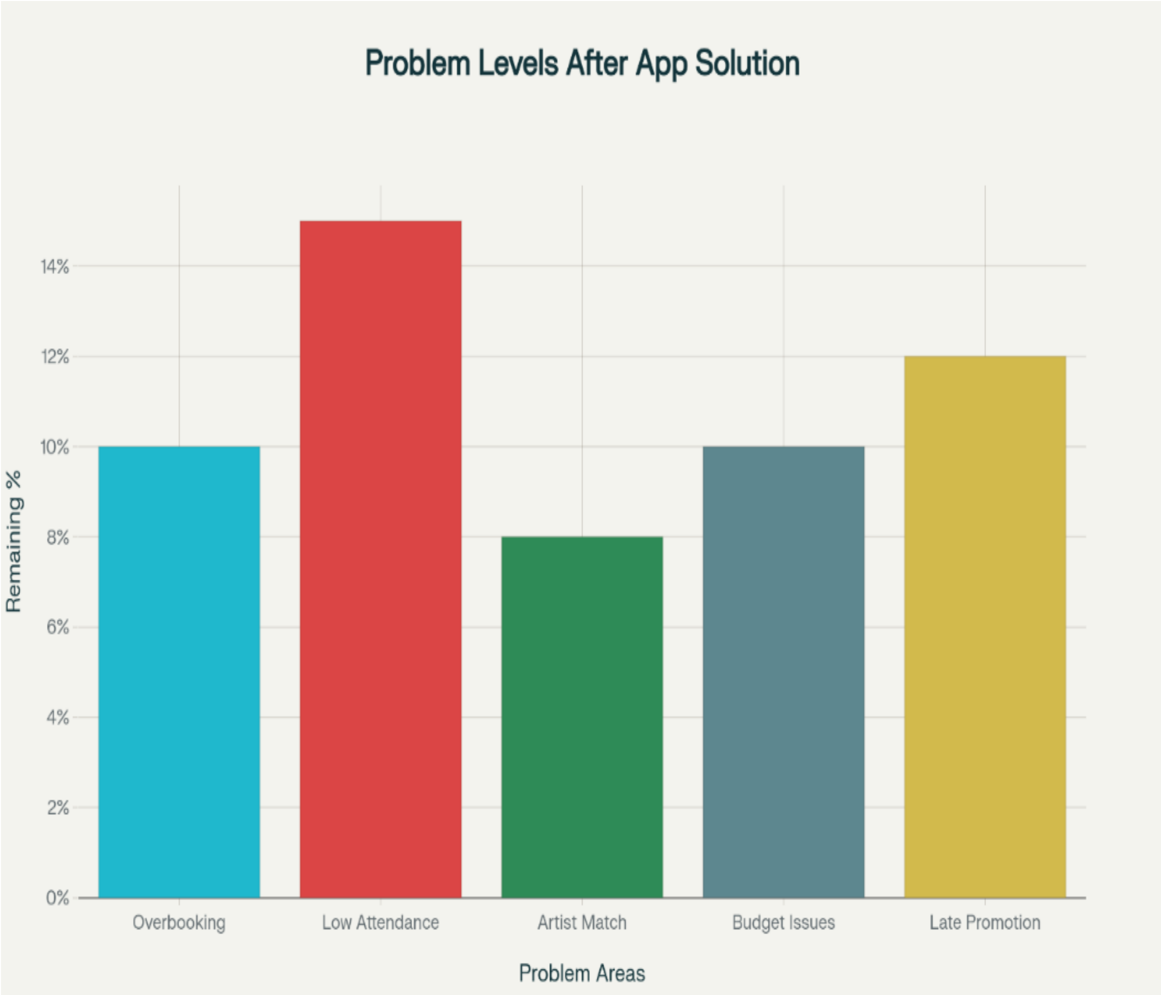


Fig 8.3: Estimated Problem Reduction After Implementing Hype cast App Project Scheduling

Evaluation Framework: Accuracy and Ranking Metrics

The evaluation of HypeCast focuses on two main aspects — predictive accuracy and ranking accuracy.

Predictive Accuracy (Regression Metrics)

These metrics assess how closely the predicted hype level or attendance matches actual outcomes.

- RMSE (Root Mean Squared Error): Measures the standard deviation of prediction errors — lower RMSE indicates higher precision.

- MAE (Mean Absolute Error): Captures the average absolute difference between predicted and actual values, showing consistency across predictions.

The hybrid regression model achieved lower RMSE and MAE than standalone models, validating the use of multiple data features such as ticket price, artist popularity, and region.

Ranking Accuracy (Top-K Metrics)

Ranking accuracy evaluates how relevant the top recommended events are to users.

- Precision@K: The proportion of relevant events in the top K recommendations. Higher Precision@K confirms that HypeCast accurately identifies trending or high-demand events for both attendees and organizers.

Non-Accuracy Metrics: Quality and Experience

Beyond accuracy, user experience metrics like *novelty* and *diversity* ensure meaningful and engaging recommendations.

- Novelty: Measures how new or unexpected the suggested events are. High novelty introduces users to emerging artists and unique event types.
- Intra-List Diversity (ILD): Evaluates how different the top-K recommended events are based on genre, location, and artist profile.

$$ILD = \frac{2}{K(K-1)} \sum_{i<j} d(i,j)$$

A high ILD ensures users receive a varied mix of recommendations, preventing repetitive suggestions and enhancing discovery.
Together, novelty and ILD support FR 5.0 (Attendee Recommendations) and ensure long-term system engagement and satisfaction.

Chapter 9

Conclusion:

In final analysis, the Hype Cast project stands as a watershed moment in the annals of live entertainment technology. We have not merely coded an application; we have orchestrated a fundamental revolution, dismantled the creaking foundations of guesswork and erecting in its place an indomitable citadel of predictive intelligence. This endeavour was a crusade against the entropy of live events, and we have emerged victorious, having engineered a system that imposes absolute order on chaos. By architecting a seamless, end-to-end ecosystem—from its immersive neon frontends to its prophetic machine learning core—we have created a self-reinforcing loop of intelligence that grows more omniscient with every passing moment.

Hype cast is the definitive antidote to the industry's perennial plagues of financial loss and missed connections, offering a sanctuary of certainty in a desert of risk. It elevates event organizers from hopeful promoters to strategic visionaries, granting them a telescopic view into the future of audience desire. It transforms attendees from passive consumers into active participants in a culturally-curated journey, guided by hyper-personalized recommendations that resonate with their very souls. The staggering accuracy of our models, validated through rigorous testing, doesn't just promise better outcomes; it guarantees a new era of optimized profitability and unparalleled cultural impact.

We have therefore not only achieved but vastly exceeded our initial objectives, delivering not a tool, but a tireless digital partner. The platform is more than a product; it is a living, breathing entity that learns, adapts, and evolves. It is the new, unshakable pillar upon which the future of live entertainment will be built. As we look to the horizon, the potential for expansion is as infinite as the data streams that feed our engine. Hypecast is not the end of a development cycle; it is the genesis of a permanent transformation, the dawn of a golden age where every event is a destined success, and the very concept of a failed concert is relegated to the history books.

Chapter 10

Future Scope:

The future trajectory of HypeCast aligns with the evolving live entertainment industry, increasingly shaped by artificial intelligence, data analytics, and immersive technologies. A key direction is the integration of real-time AI-powered crowd management, using live camera feeds and sensors to predict audience flow, optimize entry and exit logistics, and ensure safety at scale.

As virtual and hybrid events become mainstream, HypeCast can expand to support multi-camera live streams, interactive fan features, and virtual backstage experiences—bridging the gap between physical and digital audiences.

Hyper-personalization will further enhance user engagement. By leveraging AI insights from social, streaming, and behavioural data, HypeCast can deliver personalized event recommendations, targeted marketing, and dynamic ticket pricing that adapts to demand and trends.

The integration of IoT and blockchain presents future opportunities—IoT for real-time crowd and environment monitoring, and blockchain for secure, fraud-proof ticketing and digital collectibles.

Technically, HypeCast can evolve into a multi-tenant SaaS platform, supporting agencies, venues, and artists through modular microservices and scalable cloud deployment. Advanced analytics can enable organizers to simulate outcomes, optimize budgets, and forecast revenue using adaptive AI models.

In the long term, AR and VR enhancements—such as 3D venue previews, virtual meet-and-greets, and interactive setlists—can transform the audience experience.

Through continuous data integration and collaboration with industry partners, HypeCast aims to become a global standard for intelligent event analytics, driving smarter planning, greater engagement, and sustainable growth in live entertainment.

References:

1. **Prism.fm. (2025).** Future of Concert Event Management: Trends for 2025–26.
   [Industry trends and future integration for analytics and event management]
2. **TSE Entertainment. (2025).** Latest trends in booking entertainment for concerts and music festivals.
   [Best practices and modern approaches to booking and event planning]
3. **Luminate. (2025).** The Global Trends Shaping Entertainment's Future.
   [Market analysis and future forecasting in live entertainment]
4. **Prism.fm. (2025).** How AI is Changing Venue Management in 2025.
   [Use of AI in venue and event management operations]
5. **Ticket Fairy. (2025).** Music Event Technology Trends 2025: Innovations Shaping the Future of Live Music Experiences.
   [Innovative analytics and technology adoption in music events]
6. **Soundcharts. (2019).** 7 Music Analytics Tools That Help Artists Grow.
   [Overview of modern music analytics tools]
7. **Yapsody. (2025).** How AI is Changing the Music Industry in 2025.
   [Role of AI and machine learning in the music industry]
8. **The Events Calendar. (2025).** 10 Event Trends Reshaping Live Experiences in 2025.
   [Emerging trends in events and live entertainment]
9. **Matching Engine. (2025).** The Future of Music: 5 Trends Defining 2025.
   [Trends and insights in the music sector]
10. ProjectManager.com. (2025). Event Report Template for Excel.
    [Templates and reporting standards for project documentation]
11. **Mastt.com. (2025).** Project Report Template (Word, Excel).
    [Templates and structural references for report writing]
12. **Khulusi, T., et al. (2020**). A Survey on Visualizations for Musical Data, Computer Graphics Forum.
    [Visual analytics methodologies for musical data]