



**Computing Studies and Information Systems**

**CSIS 4495 002 WINTER 2025**

**Applied Research Project**

**Interactive Movie Trends Dashboard:  
Analyzing Film Industry Patterns and Personalizing User Experiences  
through Python-Based Visualization**

Submitted to: Prof. Padmapriya Arasanipalai Kandhadai

Submitted by: Lan Dinh

Student number: 300383107

April 13, 2025

## **ABSTRACT**

The "Movie Trends Dashboard" is a Python-based interactive web application designed to analyze film industry patterns and enhance user engagement through personalized cinematic experiences. Integrating IMDb, Netflix, and TMDb datasets, the dashboard employs Streamlit and Plotly to deliver four core modules: Gross Earnings Explorer for historical box office insights, Global Trends for streaming content analysis, Success Predictor for machine learning-driven movie outcome forecasts, and Profile & Movie List for user authentication, watchlist management, and social collaboration.

## TABLE OF CONTENTS

<b>A.</b>	<b><i>INTRODUCTION</i></b> .....	<b>4</b>
<b>B.</b>	<b><i>SUMMARY OF THE RESEARCH PROJECT</i></b> .....	<b>5</b>
<b>C.</b>	<b><i>CHANGES TO THE PROPOSAL</i></b> .....	<b>7</b>
1.	Change in Technology Stack: From Flask and Dash to Streamlit.....	7
2.	Addition of New Features: Success Predictor and Notification System .....	7
3.	Change in Data Storage: From Excel to CSV and JSON .....	8
4.	Expanded Feature Scope in Movie list.....	8
<b>D.</b>	<b><i>PROJECT COMPLETION TIMELINE</i></b> .....	<b>8</b>
<b>E.</b>	<b><i>IMPLEMENTED FEATURES</i></b> .....	<b>12</b>
1.	Implemented Feature 1: Welcome Page .....	13
2.	Implemented Feature 2: User Authentication .....	15
3.	Implemented Feature 3: Gross Earnings Explorer .....	17
4.	Implemented Feature 4: Explore Global Trends.....	20
5.	Implemented Feature 5: Success Predictor .....	23
6.	Implemented Feature 6: Profile & Movie List.....	26
7.	Implemented Feature 7: Notification Management .....	31
<b>F.</b>	<b><i>EVALUATION TECHNIQUE</i></b> .....	<b>31</b>
<b>G.</b>	<b><i>REFLECTION</i></b> .....	<b>32</b>
<b>H.</b>	<b><i>WORK DATE/HOURS LOGS</i></b> .....	<b>34</b>
<b>I.</b>	<b><i>CONCLUDING REMARKS</i></b> .....	<b>38</b>
<b>J.</b>	<b><i>REFERENCES</i></b> .....	<b>40</b>
<b>APPENDIX A: INTERVIEW QUESTIONS FOR MOVIES TRENDS AND INSIGHTS FOR INTERACTIVE DASHBOARDS.....</b>		<b>41</b>
<b>APPENDIX B: USER FEEDBACK SURVEY.....</b>		<b>47</b>
<b>APPENDIX C: INSTALLATION GUIDE .....</b>		<b>50</b>
<b>APPENDIX D: USER GUIDE.....</b>		<b>52</b>

## A. INTRODUCTION

The film industry represents a blend of art, commerce, and technology, generating vast datasets that reflect audience preferences, financial performance, and global distribution trends. With the rise of digital platforms and data analytics, there exists an opportunity to harness these datasets to uncover actionable insights and enhance user engagement. This research presents the "Movie Trends Dashboard," a Python-based interactive web application for analyzing movie data and personalizing cinematic experiences, spanning movie analytics, predictive modeling, and user interaction.

The primary questions this research seeks to address are: (1) What patterns in historical box office earnings and global streaming trends inform industry stakeholders? (2) Is movie success predictable using attributes such as runtime and genre? (3) How can a dashboard facilitate personalized movie tracking and user collaboration? These inquiries were shaped by interviews with a casual moviegoer and a data analyst (see Appendix A), who identified a need for tools that interpret complex film data and support peer connectivity. This work addresses deficiencies in platforms that integrate analytical rigor with accessible features, serving both industry analysts and general users.

Existing research, including Few (2022) on data visualization and Zhang et al. (2019) on machine learning for movie success, demonstrates methods for analyzing film data but often lacks multi-dataset synthesis or real-time interaction. Hayes (2019) documents Python's utility in data science, supporting its application here. This project bridges these gaps by combining IMDb and Netflix datasets with predictive and collaborative capabilities.

The research is grounded in several key assumptions that guide its approach and objectives. Firstly, it assumes that users prefer interactive dashboards for exploring movie data, as such interfaces can simplify complex datasets and make them more accessible. Secondly, it posits that Python-based frameworks like Streamlit are well-suited for building scalable and user-friendly dashboards. Additionally, the research assumes that integrating multiple

datasets, such as those from IMDB, Netflix, and Kaggle, will provide a more comprehensive and nuanced view of movie trends, enabling richer insights.

The potential benefits of this research are significant: it aims to enhance accessibility to movie data for non-technical users, empowering them to explore and analyze information. For industry professionals, the research could lead to improved decision-making by offering data-driven insights into trends and patterns.

## B. SUMMARY OF THE RESEARCH PROJECT

The "Movie Trends Dashboard" is a fully realized Python-based interactive web application that analyzes movie trends and enhances user engagement through a multifaceted platform. In its final form, the project integrates the IMDb Top 1000 Movies dataset for box office insights, the Netflix Movies and TV Shows dataset for global trends, and the TMDb API for real-time movie data, delivering a robust tool for both analytics and personalization.

The methodology encompasses a streamlined architecture and technology framework:

- **Architecture:** Raw datasets (CSV format) are cleaned and processed using Python, stored as structured files (imdb\_top\_1000.csv, netflix\_titles.csv), and supplemented by dynamic API calls (TMDb). Processed data is queried and aggregated to generate actionable insights, visualized through interactive charts and graphs, and presented via a single-page dashboard interface. User data (profiles, watchlists, notifications) is managed in JSON (users.json) for real-time updates.
- **Technology Framework:** Built entirely with Streamlit for both back-end logic (data handling, authentication) and front-end display (UI, interactivity), the dashboard leverages Plotly for dynamic visualizations. Python libraries—Pandas for data manipulation, NumPy for computations, and Requests for API integration—support the core functionality, while a pre-trained machine learning model (success\_predictor\_model.pkl) powers predictive analytics.

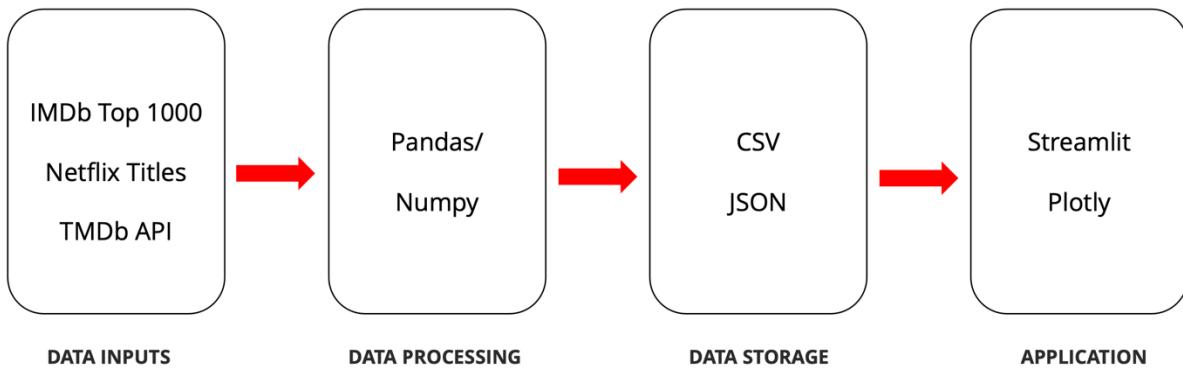


Figure 1. The full framework of the project

The final dashboard features four modules:

- **Gross Earnings Explorer:** Analyzes historical box office performance with interactive line charts (earnings over time), bar charts (top directors and actors by gross, with mean/median stats), and scatter plots (factors like runtime vs. gross). Filters include IMDb rating, Meta score, release year, and genre.
- **Global Trends:** Visualizes movies content with bar charts (duration by country), choropleth maps (title counts with genre breakdowns), and genre-based recommendations (exportable as CSV). Filters cover year, content type, rating category, and country.
- **Success Predictor:** Employs a pre-trained machine learning model (success\_predictor\_model.pkl) to classify movies as Blockbuster (Gross > \$100M, IMDb ≥ 8), Bust (Gross < \$20M, IMDb < 7), or Average, based on inputs like runtime, meta score, year, and genre.
- **Profile & Movielist:** Offers user authentication (stored in users.json), profile management (avatar, details), and a watchlist system. Users can add movies, mark them as watched, rate them (1–9 with emojis), share recommendations with custom messages, and manage notifications via a bell icon interface.

Hosted on Streamlit Cloud, the project is accessible to users via the link <https://landinh15-w25-4495-s2-land-implementationwelcome-xgrnok.streamlit.app>, with local deployment

also available by executing `streamlit run Welcome.py` to launch at `http://localhost:8501`. It serves as an analytical tool for industry insights and a personal resource for movie enthusiasts, validated through testing with credentials (Username: admin, Password: admin123).

## C. CHANGES TO THE PROPOSAL

This section outlines the modifications made to the initial research proposal titled "Developing a Python-Based Interactive Dashboard for Analyzing Movie Trends and Insights," submitted on January 26, 2025. These changes reflect adjustments in technology stack, features, data storage, scope, and timelines, were informed by technical evaluations, user needs, and project constraints, ensuring the final dashboard meets its objectives of accessibility and analytical depth.

### 1. Change in Technology Stack: From Flask and Dash to Streamlit

Initially, the proposal planned to use Flask for back-end development and either Dash or Streamlit for the front end, aiming for a lightweight framework. However, the final implementation adopted Streamlit exclusively for both back-end and front-end tasks. This shift was motivated by acknowledging Streamlit's ability to handle data processing, user authentication, and interactive visualizations within a single platform, reducing setup complexity development and meeting the project timeline effectively.

### 2. Addition of New Features: Success Predictor and Notification System

The original proposal emphasized dynamic charts (bar, scatter, heatmaps, line graphs) and basic interactivity (filters, search), without mention of predictive modeling or user notifications. In the final version, a Success Predictor module was introduced to classify movies as Blockbuster, Bust, or Average, alongside a notification system within the Movie list for sharing recommendations. These enhancements were driven by user feedback highlighting the value of predictive insights and social engagement, extending the dashboard's utility beyond static analytics to a more interactive and collaborative tool.

### **3. Change in Data Storage: From Excel to CSV and JSON**

The proposal suggested Microsoft Excel for its simplicity and accessibility, suitable for small-scale development. However, the final implementation utilized CSV files for static datasets and JSON for user data. This is because CSV offered faster read times with Pandas, while JSON's flexibility supported nested structures like watchlists and notifications. This shift ensured efficient data handling and compatibility with Streamlit's real-time updates, improving performance over Excel's manual approach.

### **4. Expanded Feature Scope in Movie list**

In the initial proposal, the Movie list was limited to search and filter functions, focusing solely on data exploration. The final version significantly broadened this scope to encompass user authentication, comprehensive watchlist management (adding, marking as watched, rating, and removing movies), sharing capabilities with custom messages, and notification handling. This expansion responded to user demands for personalization and collaboration, identified through interviews, transforming the dashboard into a dual-purpose platform that balances analytical insights with practical, user-centric features.

## **D. PROJECT COMPLETION TIMELINE**

This section presents the actual timeline for the completion of the project, spanning from the beginning to the end of the Winter 2025 term (January to April 2025). As an individual project, all responsibilities were undertaken by the author, encompassing research, design, development, testing, documentation, and presentation.

The project was executed over 13 weeks, from January 6, 2025, to April 13, 2025, across six distinct phases. Each phase includes specific milestones, deadlines, and deliverables, detailed below to reflect the actual work completed.

### **Phase 1: Research Planning and Proposal Development**

Duration: January 6, 2025 – January 19, 2025 (Weeks 1–2)

Milestones:

- Defined research objectives, including the dashboard's purpose (accessible movie analytics), target users (movie enthusiasts, analysts), and core features (visualizations, filters). Completed by January 8, 2025.
- Identified technology stack, initially considering Flask and Dash/Streamlit, later finalized as Streamlit. Completed by January 12, 2025.
- Drafted and revised the research proposal based on feedback, securing supervisor approval by January 19, 2025 (adjusted from January 26 due to early submission).

Deliverables:

- Approved research proposal, outlining scope, methodology, and objectives, submitted a proposal file.
- Initial requirements document, specifying features (e.g., gross earnings analysis, global trends) and datasets (IMDb, Netflix, TMDb API).

## Phase 2: Design and Prototype Development

Duration: January 20, 2025 – February 18, 2025 (Weeks 3–6)

Milestones:

- Created wireframes for dashboard layout, including sidebar navigation and module interfaces, finalized by January 24, 2025.
- Developed a basic prototype with Streamlit, loading IMDb and Netflix datasets and displaying simple visualizations (e.g., bar charts), completed by February 16, 2025.
- Iterated prototype based on initial testing, adding filter functionality by February 18, 2025.

Deliverables:

- Wireframes detailing layouts for Gross Earnings and Global Trends modules.
- Prototype (Welcome.py, early versions of GrossEarnings.py, and Global Trends.py), capable of loading data and rendering basic Plotly charts.

## **Phase 3: Implementation and Feature Development**

Duration: February 119, 2025 – March 9, 2025 (Weeks 7–9)

Milestones:

- Implemented core features, including Gross Earnings Explorer (line charts, bar charts, scatter plots with filters) and Global Trends (choropleth maps, duration charts), completed by March 2, 2025.
- Added advanced features: Success Predictor (machine learning model integration) and Movie list with authentication, watchlist, and notifications, finalized by March 9, 2025.
- Conducted interim testing to resolve bugs (e.g., API rate limits), ensuring feature stability by March 9, 2025.

Deliverables:

- Fully functional dashboard, comprising Welcome.py, GrossEarnings.py, GlobalTrend.py, SucessPredictor.py, MovieChecklist.py, and auth.py, hosted at <http://localhost:8501>.
- Midterm progress report detailing development milestones and challenges (e.g., JSON storage implementation).

## **Phase 4: Testing and Optimization**

Duration: March 10, 2025 – March 23, 2025 (Weeks 10–11)

Milestones:

- Conducted usability testing with peers and supervisor, simulating user scenarios (e.g., filtering data, sharing movies), completed by March 16, 2025.
- Optimized performance (e.g., faster CSV reads, streamlined JSON updates) and fixed UI issues (e.g., responsive layouts), finalized by March 23, 2025.

Deliverables:

- Usability product evaluation report summarizing feedback (e.g., intuitive navigation) and resolved issues (e.g., notification delays).

- Optimized dashboard, bug-free and responsive, ready for final deployment.

## **Phase 5: Documentation and Final Report Writing**

Duration: March 24, 2025 – April 11, 2025 (Week 12)

Milestones:

- Drafted technical documentation and user manual, covering setup, usage, and troubleshooting, completed by March 30, 2025.
- Finalized the project report, integrating proposal changes, methodology, and outcomes, submitted for review by April 6, 2025.

Deliverables:

- User instructions/guide providing step-by-step instructions (e.g., authentication, feature navigation).
- Final project report summarizing the project's scope and results.

## **Phase 6: Presentation and Submission**

Duration: April 7, 2025 – April 13, 2025 (Week 13)

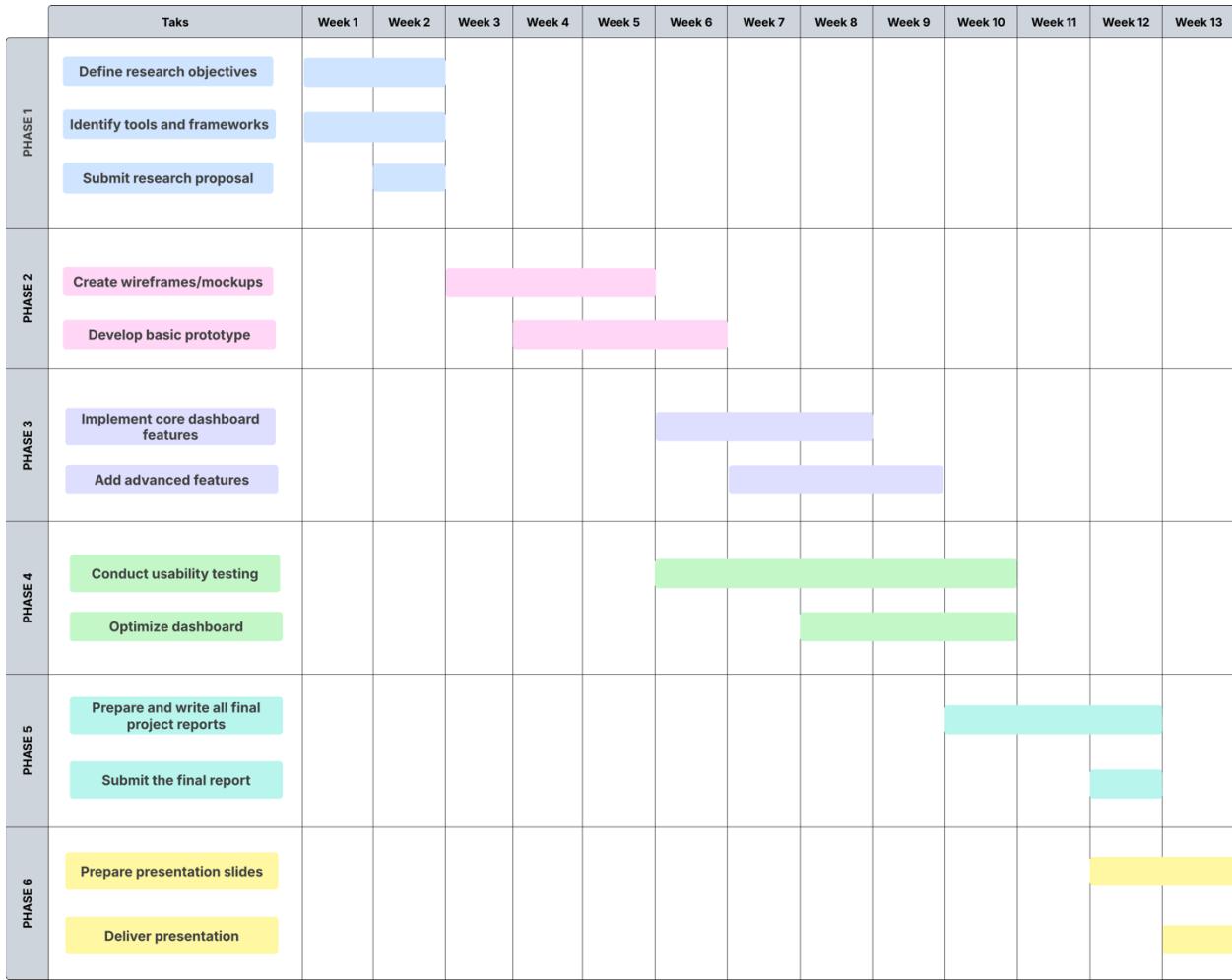
Milestones:

- Prepared and rehearsed a professional presentation, highlighting features and outcomes, finalized by April 10, 2025.
- Submitted all deliverables and presented the project to the class and supervisor on April 13, 2025.

Deliverables:

- Presentation slides covering novelty, utility, and demo highlights.
- Final deliverables package, including all reports, code, and datasets, uploaded to GitHub.

The Figure 2 below shows the project management of the research project in Gannt Chart form



*Figure 2- Gantt chart for the research project*

## E. IMPLEMENTED FEATURES

This section details the design and implementation of the four core features of project. Each feature—Gross Earnings Explorer, Explore Global Trends, Success Predictor, and Movie Checklist & Profile—is described with its objectives, technical implementation, and visual evidence through application and code screenshots.

## 1. Implemented Feature 1: Welcome Page

The Welcome Page serves as the entry point to the Movie Trends Dashboard, providing an interface to navigate its analytical and personal features. Designed to be visually appealing and user-friendly, it sets the tone for the application's accessibility across diverse audiences.

Implemented in **Welcome.py**, the Welcome Page rendered a centered layout and a sidebar for navigation. The sidebar offers options to access Gross Earnings Explorer, Explore Global Trends, Success Predictor, and Profile. Users are prompted to authenticate via the sidebar. If users do not log in or register, access is limited to the Welcome and Gross Earnings dashboards only. To unlock the full range of functionalities, users must complete the authentication process.

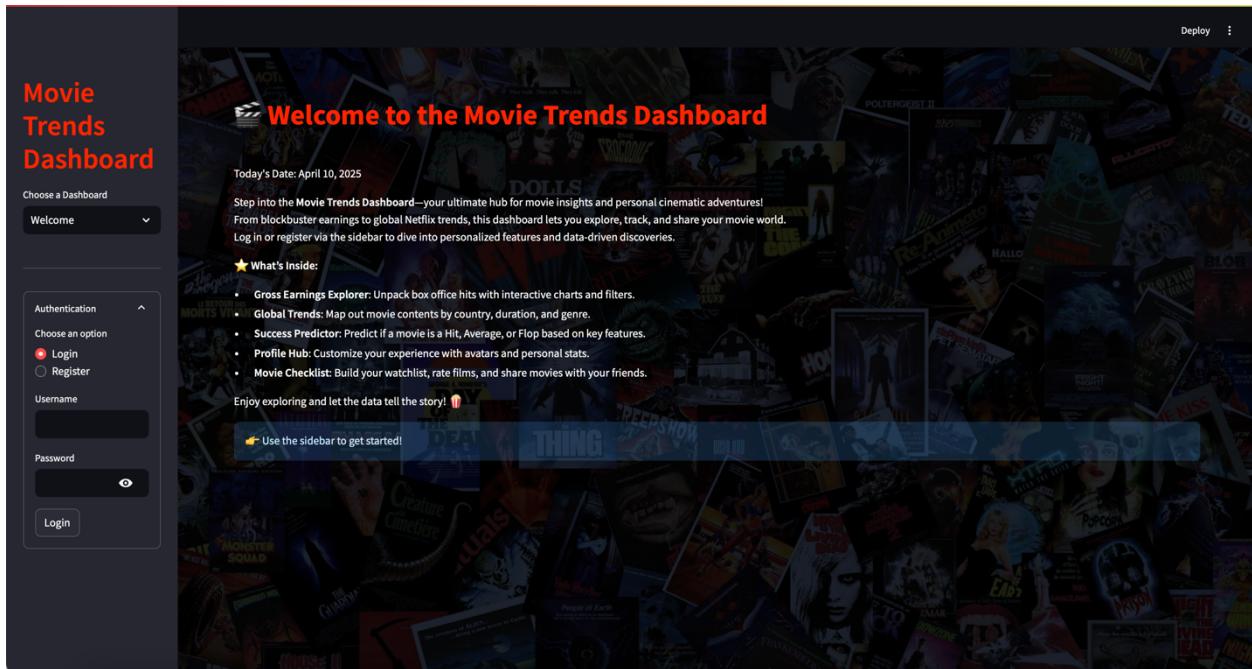
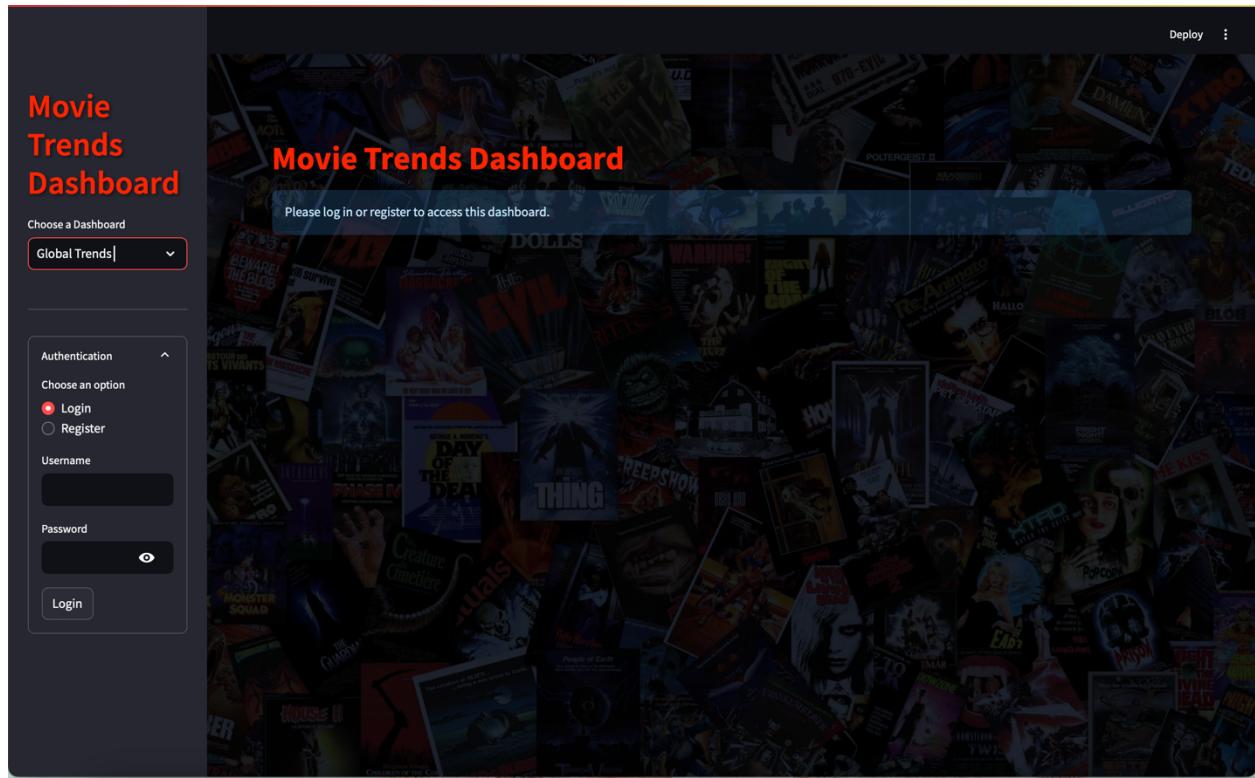


Image 1: Welcome Page Interface



*Image 2: Welcome Page Interface without Authentication*

```

200
201 # Sidebar
202 st.sidebar.title("Movie Trends Dashboard")
203 if st.session_state.logged_in:
204     st.sidebar.markdown(f"**logged in as:** {st.session_state.username}")
205     page = st.sidebar.selectbox("Choose a Dashboard", ["Welcome", "Gross Earnings", "Global Trends", "Success Predictor", "Profile"])
206     if st.sidebar.button("Logout"):
207         st.session_state.logged_in = False
208         st.session_state.username = None
209         st.session_state.movie_checklist = {}
210         st.session_state.notifications_shown = False
211         st.rerun()
212 else:
213     page = st.sidebar.selectbox("Choose a Dashboard", ["Welcome", "Gross Earnings", "Global Trends", "Success Predictor", "Profile"])
214     st.sidebar.markdown("---")
215     with st.sidebar.expander("Authentication", expanded=True):
216         auth_option = st.radio("Choose an option", ["Login", "Register"])
217         if auth_option == "Login":
218             username = st.text_input("Username")
219             password = st.text_input("Password", type="password")
220             if st.button("Login"):
221                 if check_login(username, password):
222                     st.session_state.logged_in = True
223                     st.session_state.username = username
224                     creds = load_credentials()
225                     st.session_state.movie_checklist = creds[username]["movie_checklist"]
226                     st.session_state.notifications_shown = False
227                     st.success(f"Welcome, {username}!")
228                     st.rerun()
229                 else:
230                     st.error("Invalid credentials")
231             elif auth_option == "Register":
232                 new_username = st.text_input("New Username")
233                 new_password = st.text_input("New Password", type="password")
234                 confirm_password = st.text_input("Confirm Password", type="password")
235                 full_name = st.text_input("Full Name")
236                 email = st.text_input("Email")
237                 if st.button("Register"):
238                     if new_password == confirm_password:
239                         if register_user(new_username, new_password, full_name, email):
240                             st.success("Registration successful! Please log in as {new_username}.")
241                         else:
242                             st.error("Username already exists!")
243                     else:
244                         st.error("Passwords do not match!")
245
246 # Main Logic
247 if not st.session_state.logged_in:
248     if page in ["Welcome", "Gross Earnings"]:
249         if page == "Welcome":
250             welcome()
251         elif page == "Gross Earnings":
252             show_gross_earnings()
253     else:
254         def set_background_image(image_url):
255             st.markdown(
256                 f"""
257                 <style>

```

*Image 3: Welcome.py Code*

## 2. Implemented Feature 2: User Authentication

The User Authentication feature secures access to personalized dashboard functionalities, ensuring user data privacy and enabling tailored experiences. It supports login and registration.

Developed in **auth.py** and integrated into **Welcome.py**, this feature forms for user input (username, password, and registration details like full name, DOB, email). Passwords are

hashed using a custom function to enhance security, and user data is stored in users.json. The implementation includes validation to prevent duplicate usernames and incorrect login attempts, displaying success or error messages accordingly. Post-login, users access to all dashboards, with session state managing authentication status.

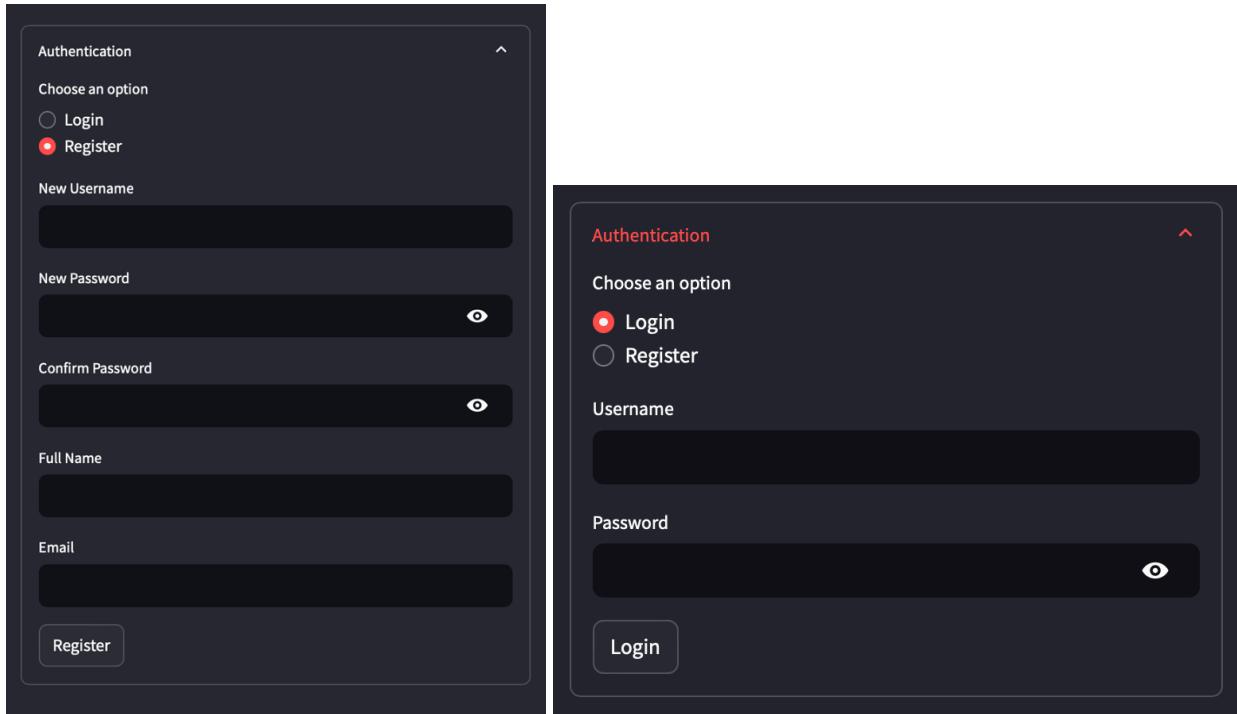


Image 4: User Authentication Interface

```

1 Welcome.py • auth.py ×
2 Implementation > auth.py > ...
3
4
5     if not os.path.exists(CREDENTIALS_FILE):
6         with open(CREDENTIALS_FILE, "w") as f:
7             json.dump({"admin": {
8                 "password": "123",
9                 "full_name": "Admin User",
10                "dob": None,
11                "email": "admin@example.com",
12                "avatar_path": None,
13                "movie_checklist": {},
14                "notifications": []
15            }}, f)
16
17    def load_credentials():
18        with open(CREDENTIALS_FILE, "r") as f:
19            creds = json.load(f)
20            for username in creds:
21                if "dob" not in creds[username]:
22                    creds[username]["dob"] = None
23                if "movie_checklist" not in creds[username]:
24                    creds[username]["movie_checklist"] = {}
25                if "notifications" not in creds[username]:
26                    creds[username]["notifications"] = []
27
28        return creds
29
30    def save_credentials(creds):
31        with open(CREDENTIALS_FILE, "w") as f:
32            json.dump(creds, f)
33
34    def check_login(username, password):
35        creds = load_credentials()
36        return username in creds and creds[username]["password"] == password
37
38    def register_user(username, password, full_name, email):
39        creds = load_credentials()
40        if username in creds:
41            return False
42        creds[username] = {
43            "password": password,
44            "full_name": full_name,
45            "dob": None,
46            "email": email,
47            "avatar_path": None,
48            "movie_checklist": {},
49            "notifications": []
50        }
51        save_credentials(creds)
52        return True
53
54    def update_user_profile(username, full_name=None, dob=None, email=None, password=None, avatar_path=None, movie_checklist=None, notifications=None):
55        creds = load_credentials()
56        if username not in creds:
57            return False
58        if full_name is not None:
59            creds[username]["full_name"] = full_name
60        if dob is not None:
61            creds[username]["dob"] = dob
62        if email is not None:
63            creds[username]["email"] = email

```

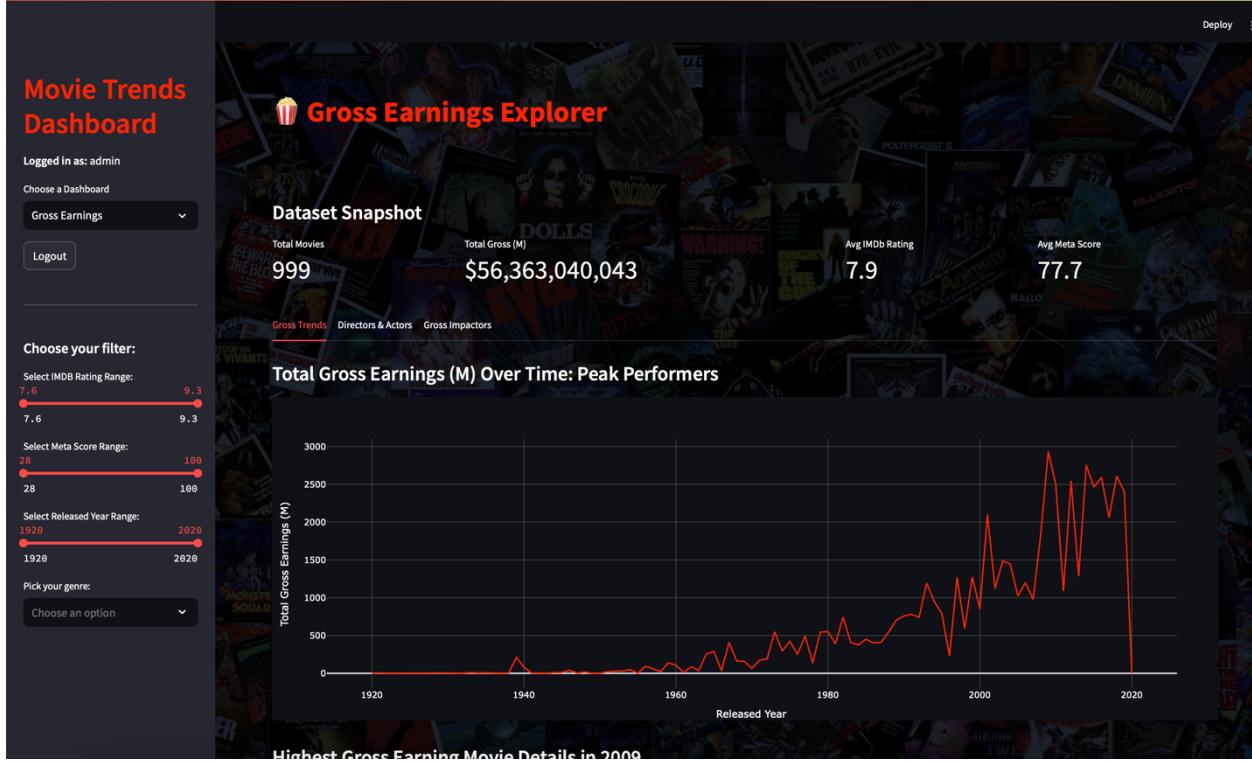
Image 5: auth.py Code

### 3. Implemented Feature 3: Gross Earnings Explorer

The Gross Earnings Explorer facilitates analysis of box office performance using the IMDb Top 1000 Movies dataset, offering insights into earnings trends, top performers, and influencing factors.

Coded in **GrossEarnings.py**, the feature processes `imdb_top_1000.csv` with Pandas, excluding records with missing gross values for accuracy. Begin with the Dataset Snapshot,

it displays four essential metrics in a four-column layout, calculated based on the user's applied filters (e.g., IMDb Rating, Meta Score, Released Year, and Genres).



*Image 6: Gross Earnings Explorer Interface*

This dashboard employs Plotly for three tabs: Gross Trends displays a line chart of total gross earnings over time, users clicking a peak reveals details of the highest-grossing movie for that year (e.g., title, gross, poster), Directors & Actors presents bar charts of the top 10 directors and actors by gross earnings, with hover-over statistics, plus mean and median earnings below each chart, and Gross Impactor offers scatter plots to explore correlations between gross earnings and variables such as runtime or IMDb rating.

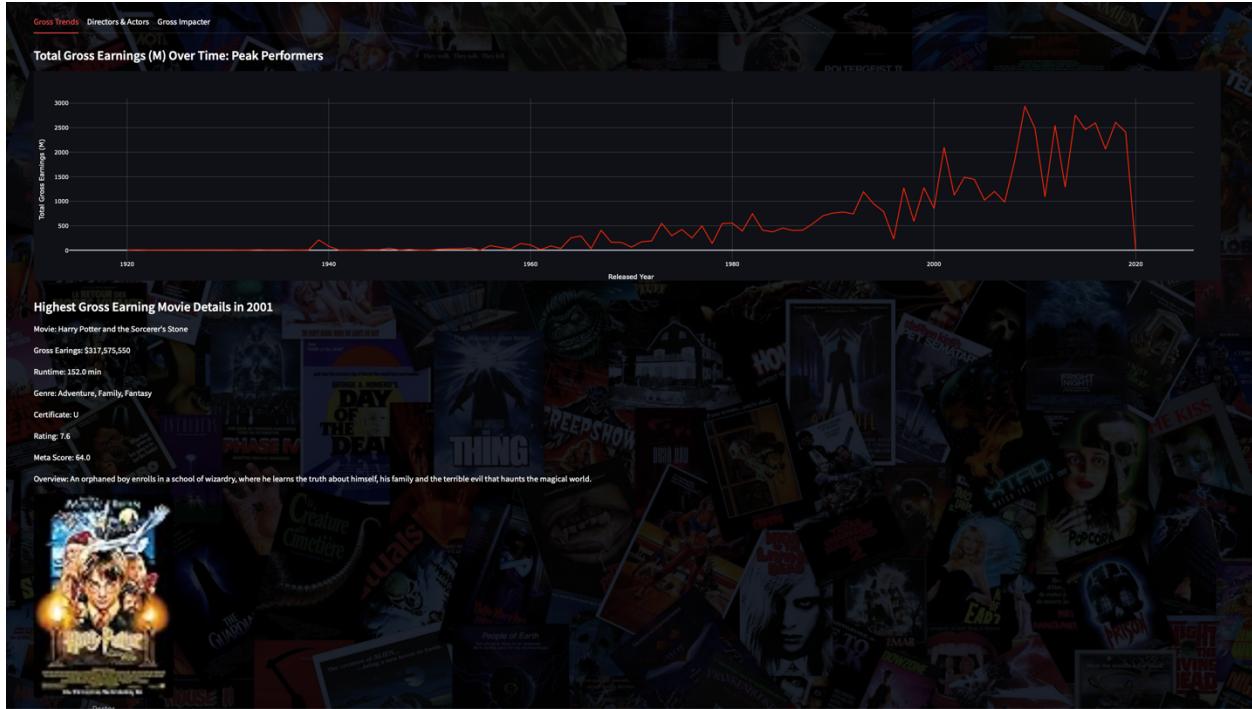


Image 7: Gross Trends tab

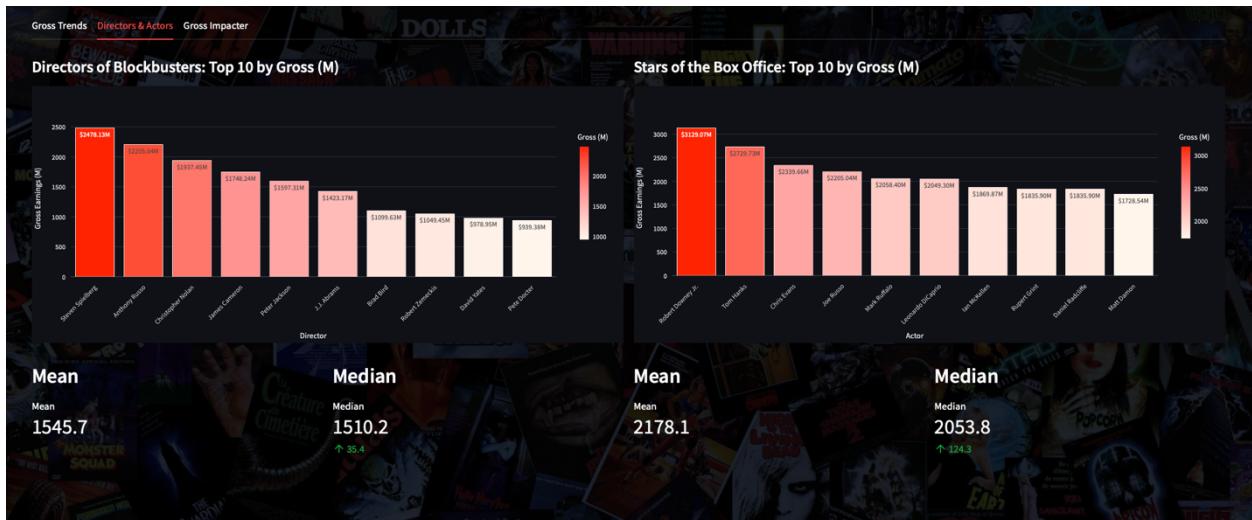
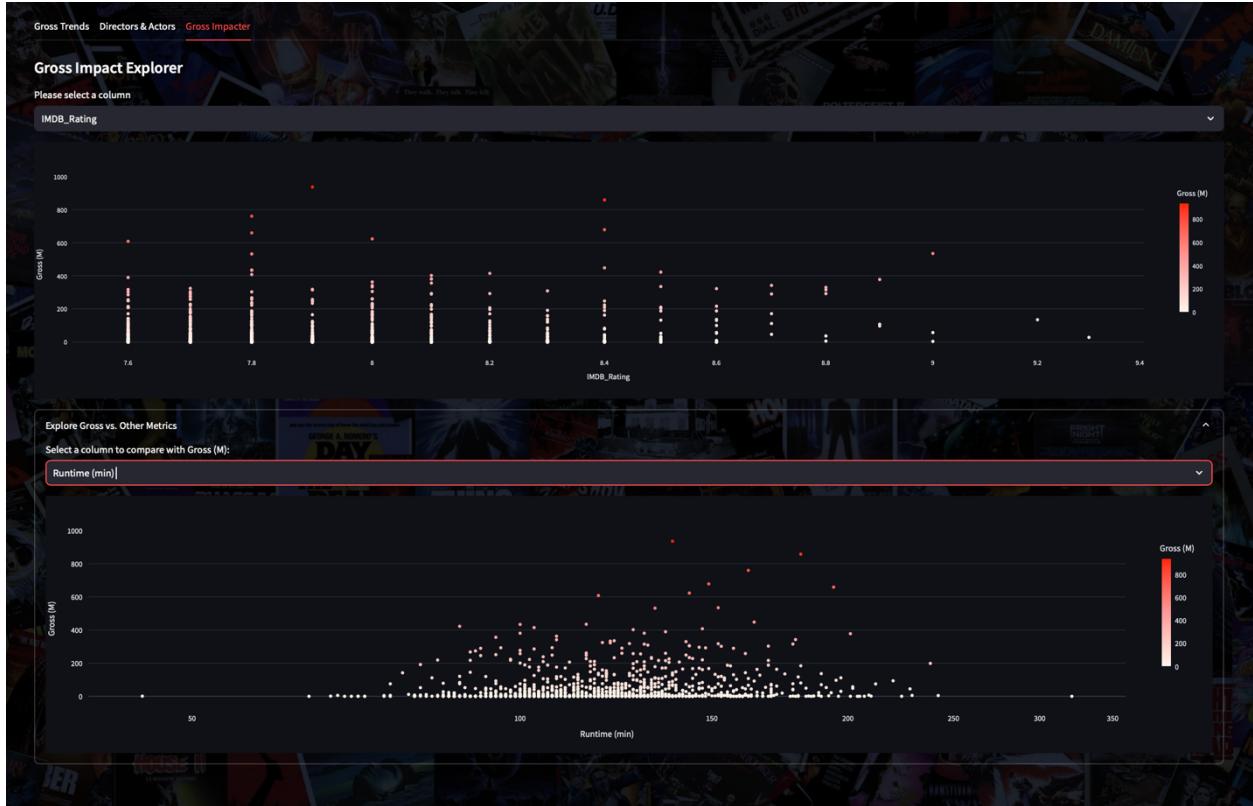


Image 8: Directors & Actors tab

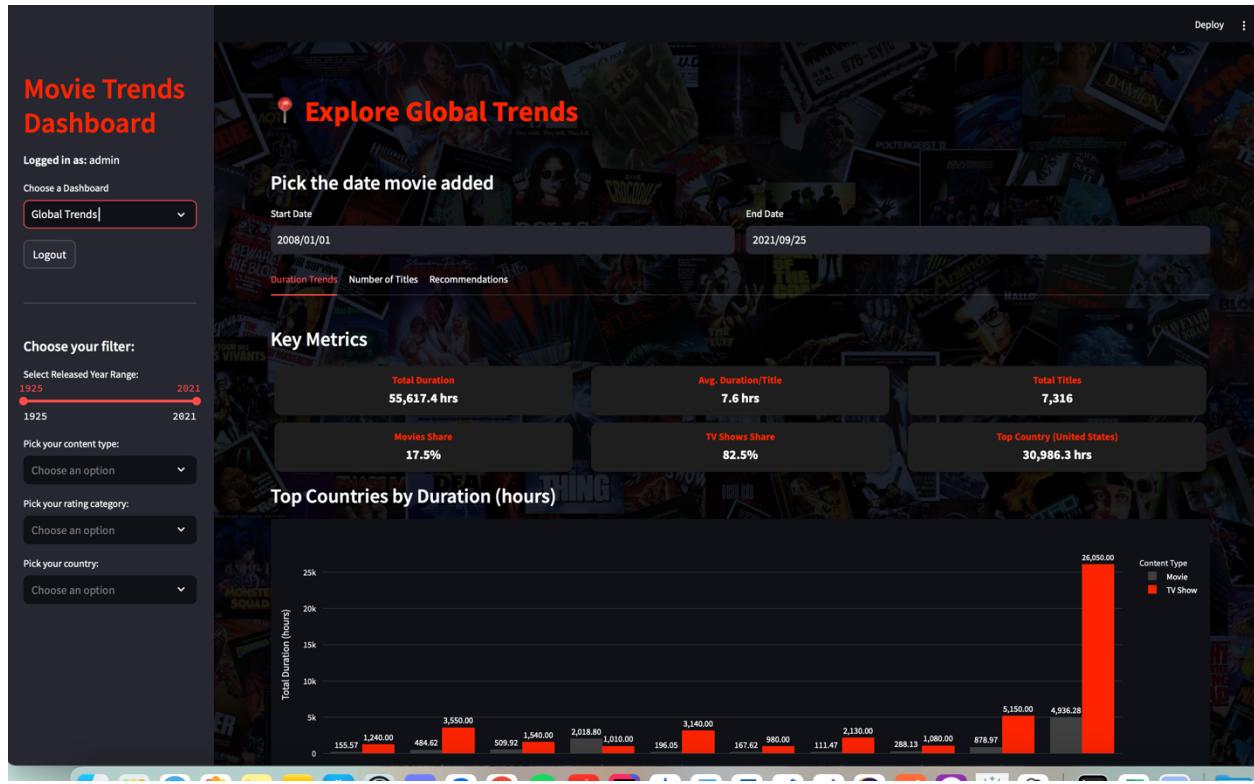


*Image 9: Gross Impactor tab*

#### 4. Implemented Feature 4: Explore Global Trends

The Explore Global Trends module visualizes Netflix content trends using `netflix_titles.csv`, targeting users interested in global streaming patterns, with a focus on duration, geographic distribution, and genre insights.

Implemented in **GlobalTrend.py**, the module preprocesses data with Pandas, standardizing country fields and imputing null durations. It offers filters for year, type, rating, country, and date added, enabling precise exploration.



*Image 10: Global Trends Interface*

This dashboard employs Plotly for three tabs: Duration Trends features a bar chart of total duration (hours) for the top 10 countries, accompanied by key metrics (e.g., total duration, average per title), Number of Titles displays a choropleth map of title counts by country, users clicking a country generates a genre breakdown donut chart, and Recommendations provides five random title recommendations based on a selected genre, exportable as a CSV file.

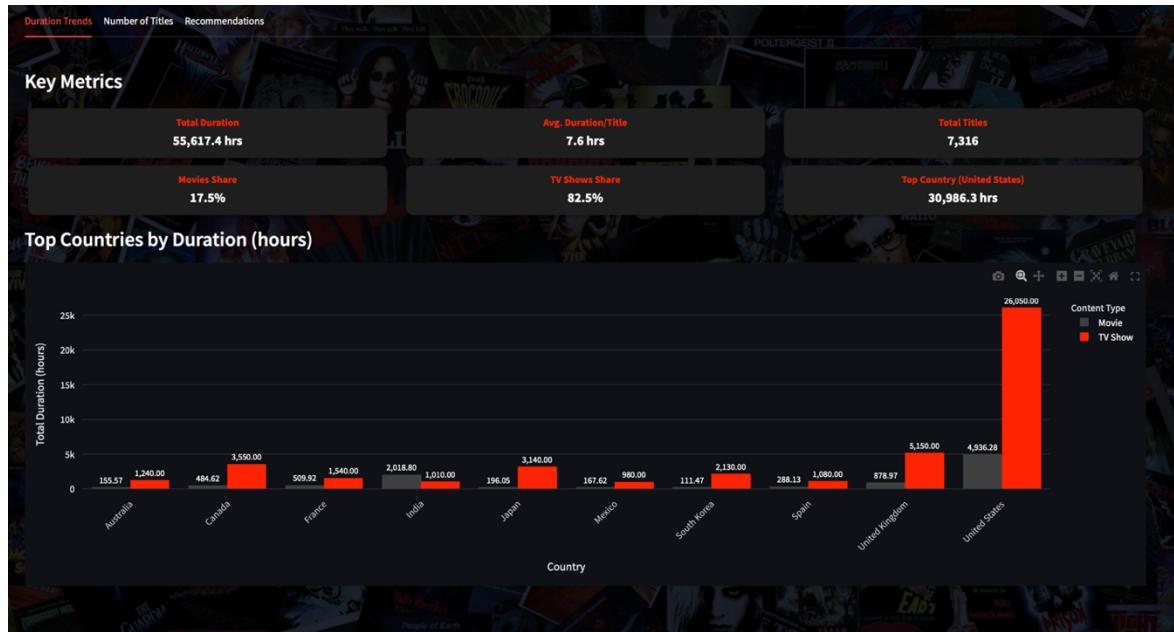


Image 11: Duration Trends tab

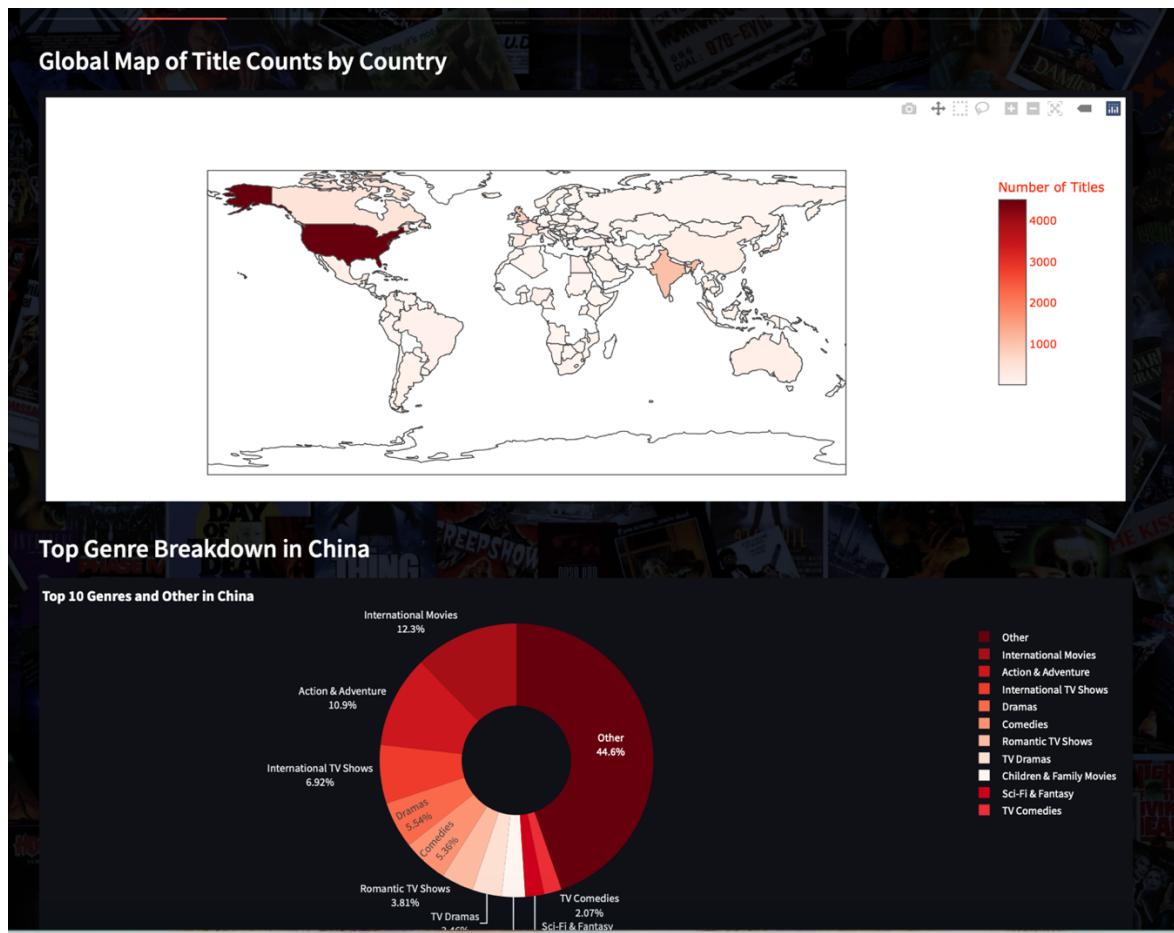


Image 12: Number of Titles tab

Title	Type	Genre	Rating	Release Year
Acts of Violence	Movie	Action & Adventure	R	2018
Charlie's Angels	Movie	Action & Adventure, Comedies	PG-13	2000
Krishna Cottage	Movie	Action & Adventure, Horror Movies, International Movies	TV-14	2004
Sniper: Legacy	Movie	Action & Adventure	R	2014
The Art of the Steal	Movie	Action & Adventure, Comedies, Independent Movies	R	2013

*Image 13: Recommendations tab*

## 5. Implemented Feature 5: Success Predictor

The Success Predictor forecasts a movie's potential success (Blockbuster, Bust, or Average), enhancing the dashboard's appeal for creators and fans by providing predictive insights based on accessible inputs.

Developed in **SuccessPredictor.py**, the module creates a form for runtime, Meta score, year, and genre inputs. A pre-trained model (**success\_predictor\_model.pkl**), built with scikit-learn, classifies movies based on these features. A result is displayed in a colored box: green for "Blockbuster" red for "Bust" or yellow for "Average" alongside the input parameters. A Blockbuster represents movies with gross earnings exceeding \$100 million and IMDb ratings of 8 or higher; a Bust represents movies with gross earnings below \$20 million and IMDb ratings below 7; an Average encompasses all other movies not meeting the Blockbuster or Bust criteria.

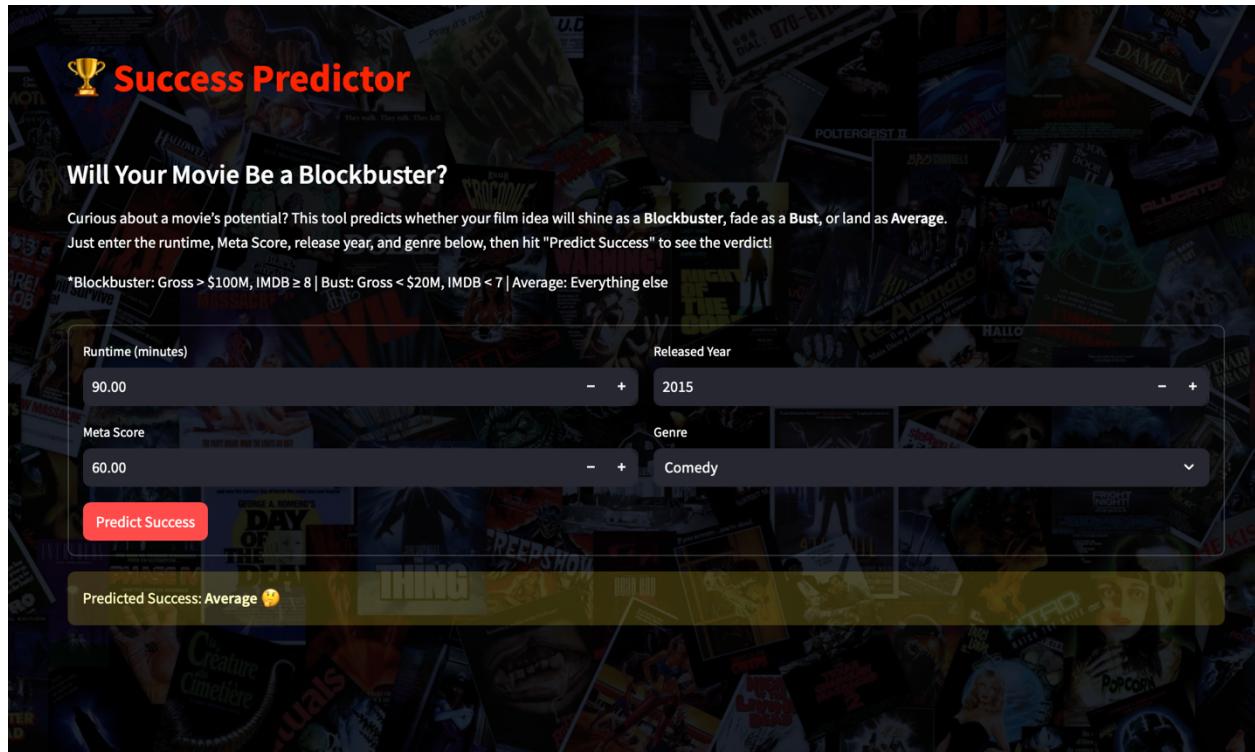


Image 14: Success Predictor Interface

The screenshot shows a code editor window with the file 'predictor.py' open. The code is a Python script for predicting movie success based on various features. It includes data cleaning, feature encoding, model training, and saving. The code is color-coded for readability.

```
Implementation > predictor.py > ...
7   print("Starting script ... ")
8
9   # Load and preprocess data
10  try:
11      df = pd.read_csv("imdb_top_1000.csv")
12      print("CSV loaded successfully. Shape:", df.shape)
13  except FileNotFoundError:
14      print("Error: imdb_top_1000.csv not found in Implementation folder!")
15      exit(1)
16
17  # Clean data
18  df["Gross"] = df["Gross"].str.replace(",","").fillna(0).astype(float) / 1e6
19  df["Runtime"] = df["Runtime"].str.extract(r"(\d+]").astype(float)
20  df["Released_Year"] = pd.to_numeric(df["Released_Year"], errors="coerce")
21  df = df.dropna(subset=[ "Released_Year"])
22  df["Released_Year"] = df["Released_Year"].astype(int)
23  df["Meta_score"] = df["Meta_score"].fillna(df["Meta_score"].mode()[0])
24  df["Genre"] = df["Genre"].str.split(",").str[0].str.strip()
25  print("Data cleaned. Shape after cleaning:", df.shape)
26
27  # Define success categories
28  def classify_success(row):
29      if row["Gross"] > 100 and row["IMDB_Rating"] > 8.0:
30          return "Hit"
31      elif row["Gross"] < 20 and row["IMDB_Rating"] < 7.0:
32          return "Flop"
33      else:
34          return "Average"
35
36  df["Success_Category"] = df.apply(classify_success, axis=1)
37  print("Success categories assigned.")
38
39  # Features and target
40  features = ["Runtime", "Meta_score", "Released_Year", "Genre"]
41  X = df[features]
42  y = df["Success_Category"]
43
44  # Encode categorical data (Genre)
45  encoder = OneHotEncoder(sparse_output=False, handle_unknown="ignore")
46  genre_encoded = encoder.fit_transform(X[["Genre"]])
47  genre_df = pd.DataFrame(genre_encoded, columns=encoder.get_feature_names_out(["Genre"]))
48  X_encoded = pd.concat([X[["Runtime", "Meta_score", "Released_Year"]].reset_index(drop=True), genre_df], axis=1)
49  print("Features encoded. Shape of X_encoded:", X_encoded.shape)
50
51  # Split and train
52  X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.2, random_state=42)
53  model = RandomForestClassifier(n_estimators=100, random_state=42)
54  model.fit(X_train, y_train)
55  print("Model trained.")
56
57  # Save model and encoder
58  with open("success_predictor_model.pkl", "wb") as f:
59      pickle.dump(model, f)
60  with open("genre_encoder.pkl", "wb") as f:
61      pickle.dump(encoder, f)
62  print("Model and encoder saved as success_predictor_model.pkl and genre_encoder.pkl")
63
64  # Evaluate model
```

Image 15: predictor.py Code

## 6. Implemented Feature 6: Profile & Movie List

The Profile & Movie List module personalizes user interactions through watchlist management, profile customization, and social features, catering to movie enthusiasts seeking a tailored experience.

Implemented in **MovieChecklist.py**, the module uses JSON (users.json) for data management. The Profile tab displays user details (username, full name, DOB, email, avatar) and supports editing name, DOB, email, password, and avatar uploads.

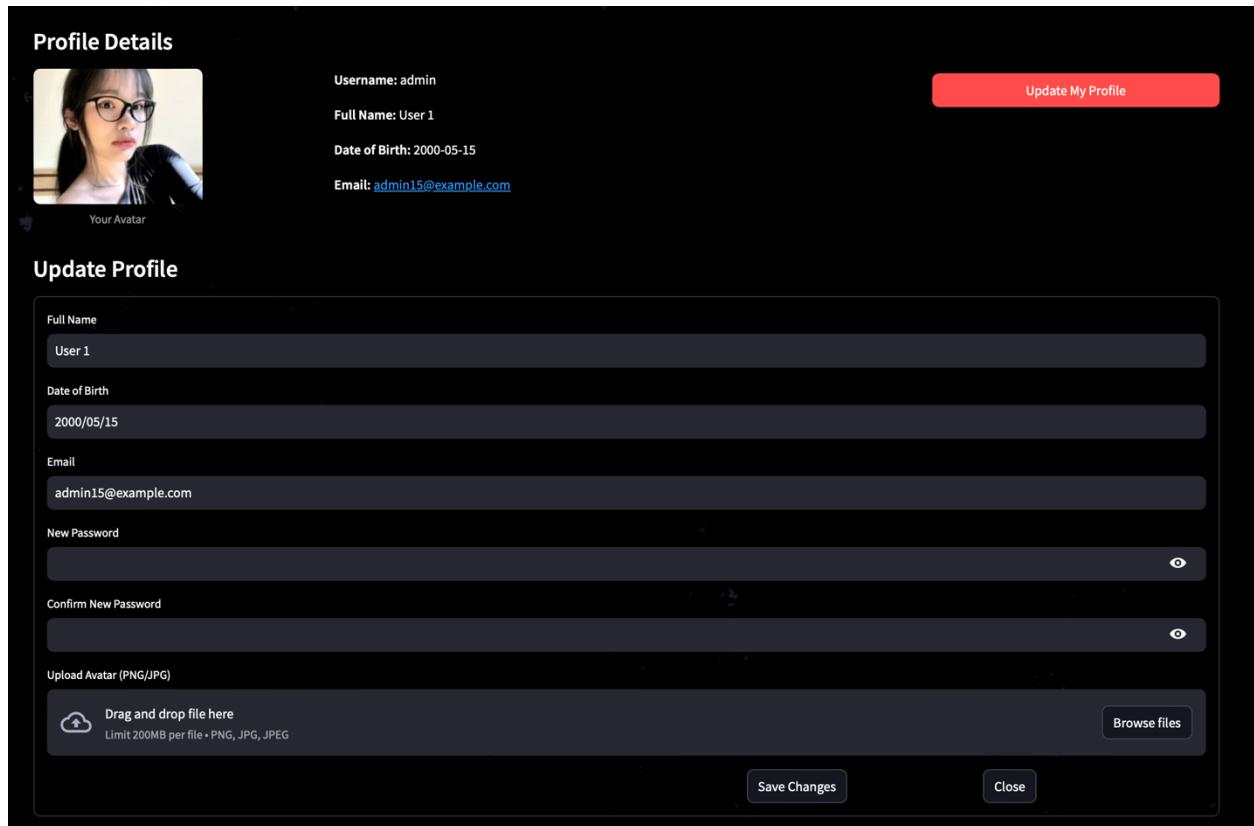


Image 16: Profile Tab Interface

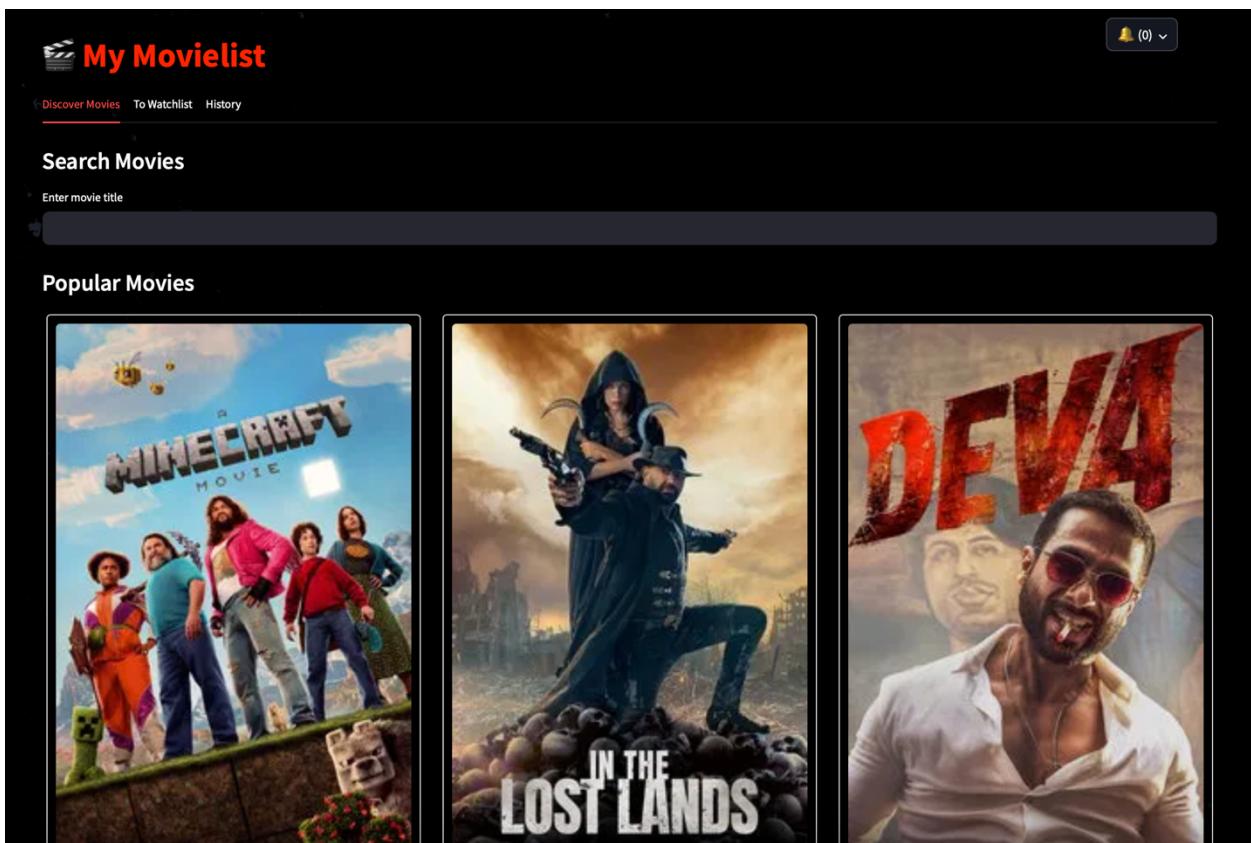
```

① users.json •
Implementation > ② users.json > {} lan > [ ] notifications
1  {"admin": {
2    "password": "admin123",
3    "full_name": "User 1",
4    "email": "admin15@example.com",
5    "avatar_path": "avatars/admin_50BE3AEC-4585-4C86-AF39-3BD6605FCAD5.jpg",
6    "dob": "2000-05-15",
7    "movie_checkList": [
8      {"1165067": {"title": "Cosmic Chaos", "watched": false, "poster": "/mClzWv7gBqgXfjZXp49Enyoex1v.jpg", "rating": 5.714,
9       "1373723": {"title": "The Codes of War", "watched": true, "poster": "/oXeiQaFRK90pxxhP5iKPXQqAIN1.jpg", "rating": 7.9,
10      "1261050": {"title": "The Quiet Ones", "watched": false, "poster": "/7NLY1jNwtZXiyVzwVoBeAhaBE8i.jpg", "rating": 4.5,
11        "762509": {"title": "Mufasa: The Lion King", "watched": true, "poster": "/lurEK87kukWNNaHd0zYnsi3yzJrs.jpg", "rating": 8.5,
12          "notifications": [
13            {"message": "lan shared 'Snow White' with you!", "read": true},
14            {"message": "lan shared 'Moana 2' with you!", "read": true},
15            {"message": "lan shared 'Captain America: Brave New World' with you: Watch thissss", "read": true},
16            {"message": "lan shared 'Popeye the Slayer Man' with you: this is so gooood", "read": true}]}},
17
18    "testuser": {
19      "password": "pass123",
20      "full_name": "Test User",
21      "email": "test@example.com",
22      "dob": null, "movie_checklist": {}},
23      "notifications": [
24        {"message": "admin shared 'Cosmic Chaos' with you: Check out 'Cosmic Chaos'!", "read": false}]}},
25
26    "lan": []
27      "password": "56",
28      "full_name": "Lan",
29      "email": "lan1505@gmail.com",
30      "dob": null,
31      "movie_checklist": [
32        {"1229730": {"title": "Carjackers", "watched": false, "poster": "/mtYwRRC7fKt16o0qZJxmBptcdzT.jpg", "rating": 6.656,
33         "822119": {"title": "Captain America: Brave New World", "watched": false, "poster": "/pzIdUEMWhWzfvLI3TwxUG2wGoi.jpg",
34           "1388366": {"title": "Popeye the Slayer Man", "watched": false, "poster": "/nVwu3mN7hr1yF467pGct3yQFM41.jpg", "rating": 8.5,
35             "notifications": []}}}

```

Image 17: users.json Code

The Movie List includes three tabs. Discover Movies tab (TMDb API search with pagination) searches for movies or browse popular titles, adding them to the watchlist. To Watchlist tab enables users to manage unwatched movies efficiently. Here, users can view titles with details like IMDb ratings (e.g., 8.5/10) and TMDb poster images, mark movies as watched with a checkmark (✓) to move them to the History tab with the date, remove entries using a cross (✗), or share recommendations via a share button (👤) by selecting a friend's username and adding a custom message, triggering a notification. History tab rates watched movies (1–9 with emojis) or removes them from the list.



*Image 18: My Movielist Interface*

The screenshot shows the 'Discover Movies' tab of the My Movielist application. At the top, there are navigation links: 'My profile' (in grey), 'My Movielist' (in red, indicating the current tab), 'Discover Movies' (in red), 'To Watchlist' (in grey), and 'History' (in grey). A notification bell icon with '(0)' is in the top right corner. Below the header, the title 'My Movielist' is displayed with a clapperboard icon. The main section is titled 'Search Movies' and features a search bar with the placeholder 'Enter movie title' and the input 'avatar'. Below the search bar are 'Previous' and 'Next' buttons. A 'Did you mean:' dropdown menu lists three suggestions: 'Avataro Sentai Donbrothers vs. Zenkaiger', 'Avatar: Creating the World of Pandora', and 'The King's Avatar: For the Glory'. The main content area displays the movie 'Avatar (2009)' with a thumbnail image, release date (2009-12-15), rating (7.588/10), and a brief overview: 'In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and protecting an alien civilization.' An 'Add to Movielist' button is shown below the overview. Below this, two more movie entries are partially visible: 'Avatar (2006)' and 'Avatar (1916)'. The entire interface has a dark theme.

Image 19: Discover Movies tab

The screenshot shows the 'To Watchlist' tab of the My Movielist application. The layout is similar to Image 19, with the same header and navigation links. The main content area displays two movies on the watchlist: 'Cosmic Chaos (5.714/10)' and 'The Quiet Ones (4.5/10)'. Each movie entry includes a thumbnail image, its title, its rating, and three circular icons with symbols: a checkmark, a cross, and a person icon. The interface maintains the dark theme established in the previous image.

Image 20: To Watchlist tab

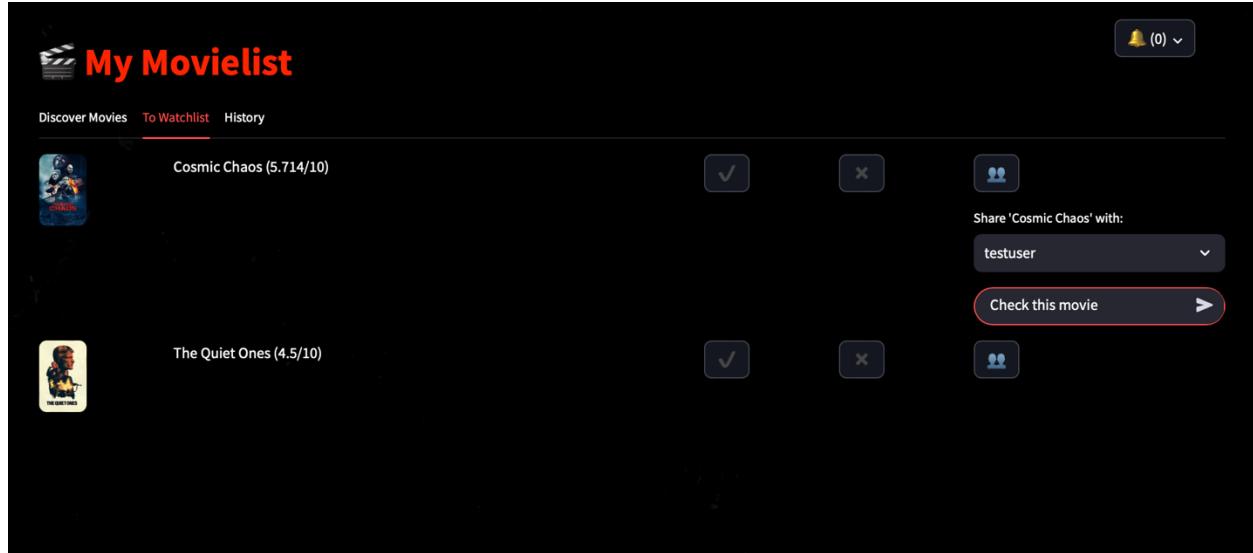


Image 21: Buttons Interface

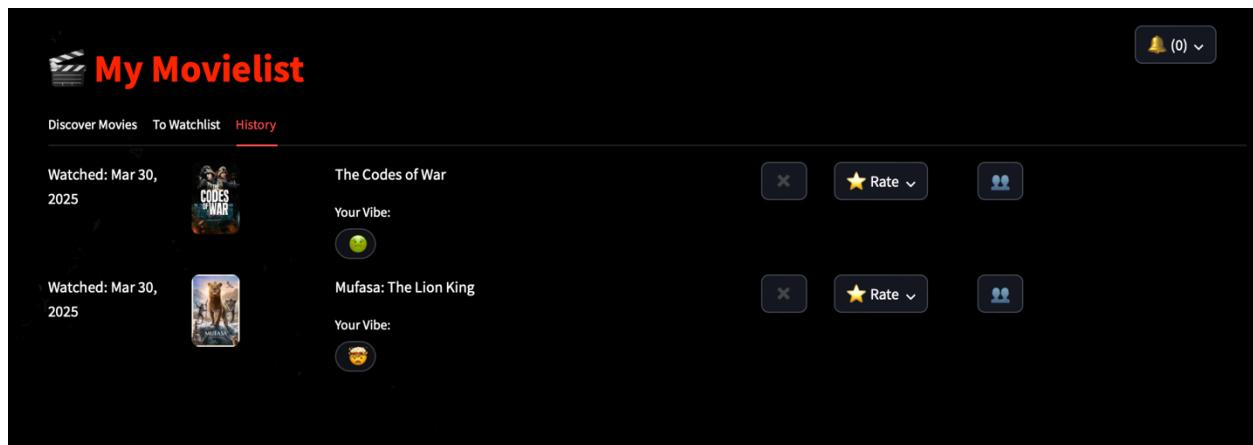


Image 22: History tab

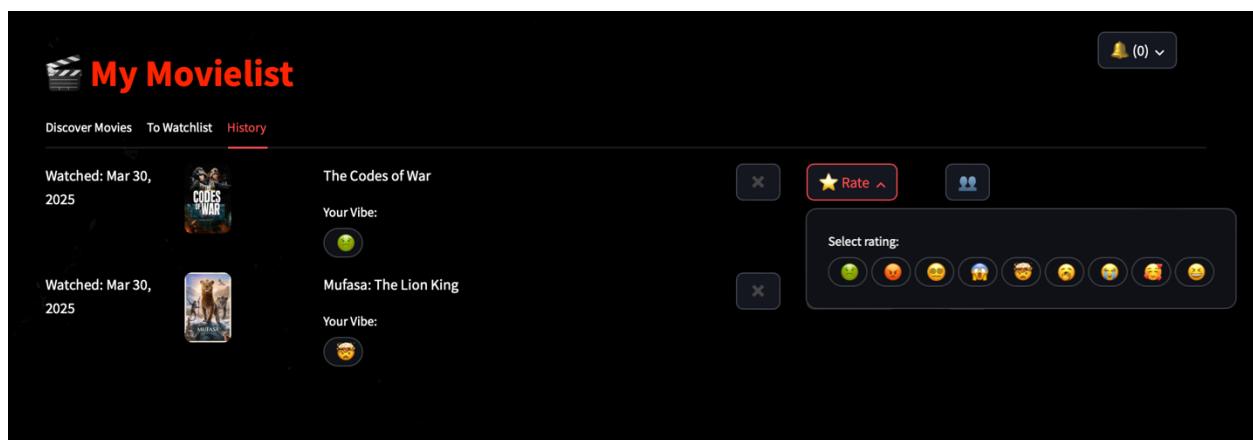


Image 23: Rating Button Emojis Interface

## 7. Implemented Feature 7: Notification Management

The Notification Management feature enhances the social interactivity by enabling users to receive and manage movie-sharing notifications, fostering collaboration among registered users. Integrated within the Profile & Movie List module, this feature allows users to view notifications triggered when a friend shares a movie recommendation.

Users can click "Mark All as Read" to update all unread notifications to read status or click "Clear All Notifications" to remove all messages from the list, resetting the notification count to zero.

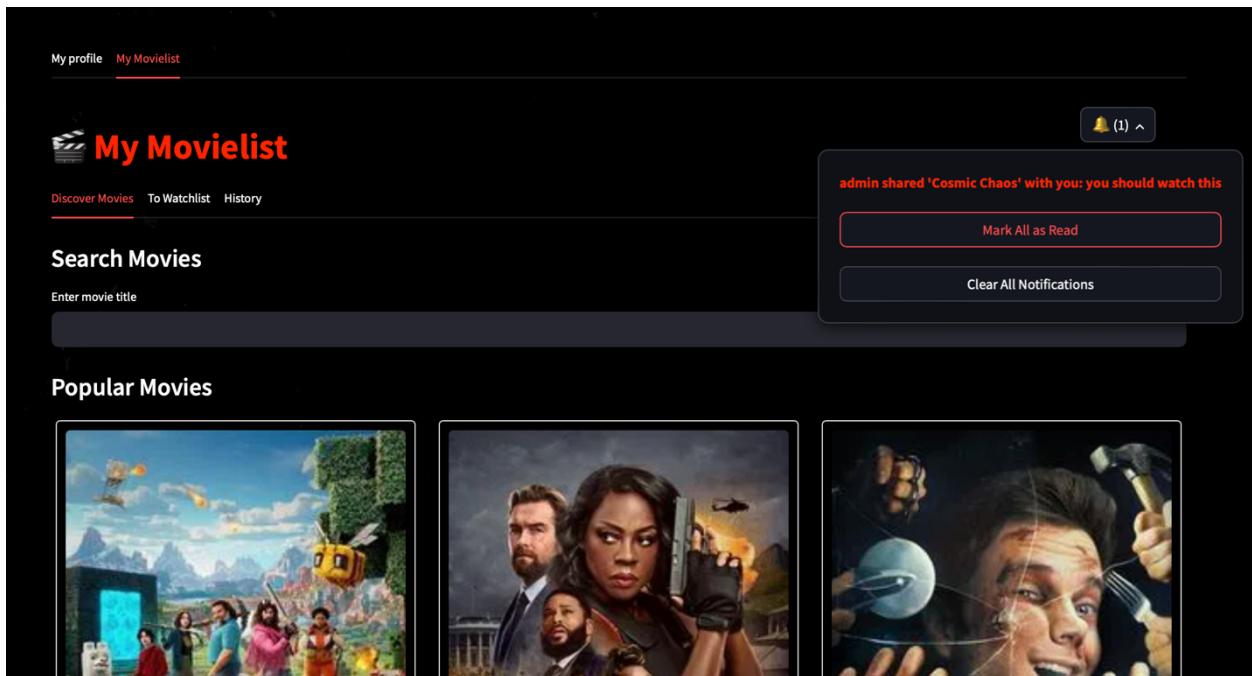


Image 24: Notification Management Interface

## F. EVALUATION TECHNIQUE

To assess the effectiveness and quality of the "Movie Trends Dashboard," this project employs the ISO/IEC 9126-1 software quality model, a standardized framework that evaluates software products across five key characteristics: Functionality, Reliability, Usability, Efficiency, and Portability (Ho-Won Jung, 2004). This model was selected for its comprehensive approach, ensuring the dashboard meets both user needs and technical

standards as outlined in the project proposal (January 26, 2025). Evaluation was conducted between April 5 and April 11, 2025, through a structured survey questionnaire distributed to 10 random students at Douglas College, targeting a diverse sample of movie enthusiasts, aspiring analysts, and computer science peers. The survey consisted of 15 questions rated on a 5-point Likert scale (1 = Strongly Disagree, 5 = Strongly Agree), assessing aspects such as feature performance, interface clarity, system stability, responsiveness, and cross-device compatibility. Participants also provided qualitative feedback on their overall impressions and suggestions for improvement (see Appendix B).

The survey results indicated strong performance, with an overall average score of 4.05, reflecting high user satisfaction. The dashboard excelled in Functionality (average 4.2 for feature performance and watchlist management), supporting seamless tracking and sharing of movies. Usability was a notable strength (average 4.4 for interface clarity), as the design facilitated easy navigation across user groups. Portability also performed well (average 4.2 for browser and device compatibility). However, Reliability (average 3.7 for data persistence) and Efficiency (average 3.8 for loading speed) revealed areas for improvement, with users noting occasional issues with notification persistence and responsiveness. Additionally, the prediction feature scored lower, with feedback highlighting a lack of clarity and concerns about data freshness.

Based on these findings, several enhancements were implemented. To improve reliability, notification persistence was strengthened by optimizing data storage processes. The prediction feature was refined by adding explanatory tooltips to enhance clarity. Additionally, user feedback prompted the introduction of stricter authentication measures to bolster security and a refined sharing mechanism to better control visibility. These changes improved overall system performance and user experience, aligning the dashboard more closely with audience expectations.

## **G. REFLECTION**

The development of the "Movie Trends Dashboard" offered a rich learning experience, marked by significant challenges, valuable insights, and rewarding milestones. Reflecting on the process, several key aspects stand out, shaping my approach to data analytics and software development.

One of the earliest hurdles was the unexpected need to change the project's focus from a mobile application (an Activity Reminder app) to a Python-based dashboard for movie analytics, prompted by client feedback just before the proposal deadline. This last-minute transition, occurring on January 24, 2025, demanded a rapid overhaul of the project scope, requiring me to rewrite the proposal, research new technologies, and redefine objectives under tight time constraints. By leveraging prior coursework and adapting to dashboard concepts, I successfully revised and secured approval for the proposal by January 26, 2025. This experience underscored the importance of flexibility and efficient time management, lessons I applied throughout the project.

Data preparation presented another challenge, as the large IMDb and Netflix datasets contained inconsistencies, such as missing values and non-standardized formats. Cleaning these datasets, completed by February 16, 2025, required meticulous handling to ensure analytical accuracy. Using Python libraries like Pandas and NumPy, I streamlined the process, automating tasks like null value imputation and format unification. This challenge highlighted the critical role of robust data preprocessing, prompting me to prioritize reusable scripts for efficiency in future projects.

Initiating the coding phase posed a steep learning curve, as my prior experience with Streamlit was limited. Beginning on February 6, 2025, I faced difficulties in implementing the dashboard layout and interactive features, such as filters and visualizations. To overcome this, I consulted online tutorials and adapted sample code, enabling the creation of a functional prototype with custom styling by February 18, 2025. This process taught me the value of self-directed learning and resourcefulness, skills that proved essential as the project grew in complexity.

The project's greatest reward lay in the opportunity to learn Python and Streamlit, new tools that expanded my technical repertoire. Despite the dashboard's imperfections, stemming from my nascent knowledge, mastering these technologies was immensely satisfying. Choosing to embrace Python's versatility and Streamlit's rapid development capabilities allowed me to confront challenges head-on, enhancing my skills and confidence. By April 13, 2025, delivering a functional dashboard—flaws and all—affirmed the value of perseverance and the joy of acquiring new expertise.

## H. WORK DATE/HOURS LOGS

The Table 1 below documents the activities, dates, times, statuses, and notes for the development of the project from January 6, 2025, to April 13, 2025. All tasks were completed by the author, covering research, coding, testing, documentation, and presentation phases.

ACTIVITY	DATE	TOTAL TIME	STATUS	NOTES
Conducting Interest and Expertise Survey to identify skills and technologies	January 12, 2025	1:00	Complete	Survey identified interest in dashboards and Python; ruled out mobile development.
Researching topics for the project, reviewing papers and prior works	January 14, 2025	2:00	Complete	Explored mobile apps and dashboards; initially chose Activity Reminder Mobile App.
Writing the draft proposal for the client	January 20, 2025	6:00	Complete	Drafted proposal for mobile app; later revised for dashboard focus.
Discussing initial topic (Activity Reminder Mobile App) with client	January 23, 2025	0:30	Complete	Client feedback suggested dashboard over mobile app for broader impact.

Changing topic to Python-based dashboard for movie trends	January 24, 2025	0:30	Complete	Topic shifted to dashboard development; approved by client.
Rewriting the proposal for the new topic	January 24, 2025	7:00	Complete	Revised proposal to focus on movie analytics; submitted and approved January 26, 2025.
Cleaning Netflix dataset using Jupyter Notebook	January 26, 2025	2:00	Complete	Removed missing values and standardized formats for netflix_titles.csv.
Writing the initial progress report	January 29, 2025	5:00	Complete	Documented topic change and data preparation progress.
Writing interview questions for user feedback	February 1, 2025	1:00	Complete	Prepared questions for Participants A and B on dashboard preferences.
Creating and adding client to the GitHub repository	February 1, 2025	0:05	Complete	Added client as collaborator; updated from instantaneous to reflect minimal effort.
Drafting initial ideas for dashboard design	February 2, 2025	1:00	Complete	Sketched layout ideas for sidebar and modules (e.g., visualizations, checklist).
Coding initial dashboard prototype in Python	February 6, 2025	10:00	Complete	Built basic Streamlit app with data loading and simple charts.
Interviewing Participant A for user feedback	February 8, 2025	0:30	Complete	Gathered feedback on desired features (e.g., filters, interactivity).
Interviewing Participant B for user feedback	February 10, 2025	0:30	Complete	Confirmed interest in predictive and personal features; met deadline.
Analyzing interview results	February 14, 2025	2:00	Complete	Identified demand for Success Predictor and notifications from feedback.

Documenting interview questions and responses	February 14, 2025	2:00	Complete	Uploaded Interviews.pdf to GitHub, summarizing user insights.
Cleaning IMDb dataset using Jupyter Notebook	February 16, 2025	1:00	Complete	Standardized imdb_top_1000.csv for gross earnings analysis.
Adding IMDb dataset to the repository	February 16, 2025	0:05	Complete	Uploaded imdb_top_1000.csv; updated from instantaneous for accuracy.
Coding the Welcome page for the dashboard	February 18, 2025	1:00	Complete	Completed Welcome.py with layout, styling, and sidebar navigation.
Coding the "Explore Global Trends" module	February 20, 2025	5:00	Complete	Developed GlobalTrend.py with Netflix visualizations (maps, bar charts).
Drafting the midterm progress report	February 24, 2025	1:00	Complete	Summarized prototype, interviews, and initial modules in MidtermReport.pdf.
Coding the "Gross Earnings Explorer" module	March 1, 2025	7:00	Complete	Built GrossEarnings.py with line, bar, and scatter plots, plus filters.
Enhancing dashboards with custom styling	March 5, 2025	8:00	Complete	Improved UI aesthetics (e.g., colors, layouts) across all modules.
Drafting progress report #2	March 9, 2025	4:00	Complete	Detailed visualization progress and user feedback integration.
Creating user authentication system	March 14, 2025	2:30	Complete	Added login/registration to Welcome.py and auth.py, using users.json.

Adding profile editing functionality	March 18, 2025	5:45	Complete	Enabled profile updates (name, DOB, email, password) in MovieChecklist.py.
Integrating avatar upload feature	March 19, 2025	0:30	Complete	Added image upload for avatars to profile page.
Creating movie checklist with search and pagination	March 22, 2025	5:00	Complete	Implemented TMDb API search with suggestions and pagination in MovieChecklist.py.
Drafting progress report #3	March 23, 2025	0:45	Complete	Summarized authentication and checklist development.
Implementing watchlist and history tabs	March 26, 2025	4:15	Complete	Added watchlist actions (mark watched, remove) and history display.
Integrating emoji-based user rating system	March 27, 2025	1:15	Complete	Enabled 1–9 emoji ratings for watched movies in History tab.
Creating notification system	March 29, 2025	1:45	Complete	Built notification popover with read/unread styling and management options.
Drafting progress report #4	March 30, 2025	0:30	Complete	Documented checklist enhancements and notifications.
Implementing Success Predictor module	April 1, 2025	5:00	Complete	Integrated success_predictor_model.pkl into SuccessPredictor.py for movie classification.
Fixing remaining bugs in the codebase	April 2, 2025	1:00	Complete	Resolved UI glitches and API timeout issues for stability.
Updating README file with project details	April 2, 2025	0:15	Complete	Revised README.md with instructions and feature overview.
Conducting user evaluation via	April 3, 2025	4:00	Complete	Surveyed 12 users; interviewed 3 for feedback on usability and features.

surveys and interviews				
Drafting detailed user instructions and guide	April 5, 2025	5:00	Complete	Wrote UserGuide.docx with setup and navigation instructions.
Creating presentation slides for the project	April 5, 2025	4:00	Complete	Developed Presentation.pptx covering novelty, tech stack, and demo.
Drafting progress report #5	April 6, 2025	1:00	Complete	Summarized testing and documentation progress.
Finalizing technical documentation	April 7, 2025	3:00	Complete	Completed TechnicalDocs.pdf with code and dependency details.
Finalizing project report	April 8, 2025	4:00	Complete	Revised FinalReport.docx with all sections, including timeline.
Rehearsing presentation and demo	April 9, 2025	2:00	Complete	Practiced live demo and slide delivery for clarity.
Submitting final deliverables	April 10, 2025	0:30	Complete	Uploaded all files to GitHub and submitted to supervisor.
Delivering final presentation	April 13, 2025	0:30	Complete	Presented project to class, showcasing dashboard demo.

*Table 1. Work Log for the Movie Trends Dashboard Project*

## I. CONCLUDING REMARKS

The "Movie Trends Dashboard" represents a significant step forward in making complex movie data accessible and engaging for both casual enthusiasts and industry analysts. By integrating diverse datasets from IMDb, Netflix, and TMDb, and leveraging Python-based tools like Streamlit and Plotly, this project successfully delivers a multifaceted platform that combines analytical depth with user-centric personalization. The dashboard's core modules—Gross Earnings Explorer, Global Trends, Success Predictor, and Profile &

Movie List—address key research questions by uncovering historical and global film trends, predicting movie success, and fostering collaborative user experiences.

The project's development journey, spanning January to April 2025, demonstrated the power of iterative design and user feedback in shaping a robust tool. From overcoming initial challenges like dataset inconsistencies and a steep learning curve with Streamlit to implementing advanced features such as predictive modeling and notification systems, the process underscored the importance of adaptability and technical resilience. Evaluation using the ISO/IEC 9126-1 model confirmed high usability and functionality, with an overall user satisfaction score of 4.05, though areas like reliability and efficiency highlighted opportunities for future refinement.

Ultimately, this research contributes to the intersection of data analytics and entertainment, offering a scalable, interactive solution that empowers users to explore cinematic trends and personalize their movie-watching experiences. Future enhancements could include real-time data integration, enhanced predictive accuracy, and mobile optimization to further broaden its impact. The "Movie Trends Dashboard" not only fulfills its objectives but also lays a foundation for continued innovation in data-driven storytelling.

## J. REFERENCES

Few, S. (2022). *Stephen Few on Data Visualization: 8 Core Principles*. Tableau Software.

Hayes, G. (2019). *Which Programming Language Should Data Scientists Learn First?*. Towards Data Science.

Amar, N., & Baliga, S. (2022). *Design and Development of Analytical Dashboard*. Journal of Current Research in Engineering and Science.

Lee, K, Park, J, Kim, I & Choi, Y (2016). *Predicting movie success with machine learning techniques: ways to improve accuracy*. Springer. Information Systems frontiers, vol(in press). DOI:10.1007/s10796-016-9689z

Ho-Won Jung, Seung-Gweon Kim, & Chang-Shin Chung. (2004). *Measuring Software Product Quality: A Survey of ISO/IEC 9126*. IEEE Software, 21(05), 88–92. doi:10.1109/ms.2004.1331309

AI tools:

- ChatGPT: Utilized for drafting initial outlines of the project proposal, progress reports, and user guide section.
- Grok (xAI): Employed for debugging Python code, particularly for resolving issues with TMDb API integration in MovieChecklist.py and optimizing JSON handling in auth.py.
- Grammarly AI: Applied for proofreading and refining the clarity of the final report, user guide, and presentation slides to ensure professional language and coherence.

## **APPENDIX A: INTERVIEW QUESTIONS FOR MOVIES TRENDS AND INSIGHTS FOR INTERACTIVE DASHBOARDS**

### **1. Introduction**

To develop a user-centric Movies Trends and Insight Dashboard, a series of user interviews were conducted. The objective was to gain insights into user expectations, preferences, and challenges when analyzing movie-related data. The feedback from these interviews helped shape the dashboard's design and functionality, ensuring it provides meaningful and actionable insights.

### **2. Interview Process**

To gather diverse perspectives on movie analytics dashboards, three participants were interviewed:

Participant A – A casual moviegoer who enjoys exploring movie ratings and audience reviews.

Participant B – A data analyst who works with datasets and uses analytical tools.

### **3. Interview Methodology**

Format: One-on-one interviews conducted in person and via video calls.

Duration: Each interview lasted approximately 30–45 minutes.

Questionnaire Structure: The interview was divided into three sections: User Interaction & Experience, Data Presentation & Interpretation, and Customization & Integration.

### **4. Interview Questions**

#### **a) User Interaction & Experience**

- How do you typically explore and analyze movie-related data?
- What key features or functionalities would you like to see in a movie analytics dashboard?
- Have you used any movie analytics dashboards before? If so, what did you like/dislike about them?

#### **b) Data Presentation & Interpretation**

- What type of visualizations (e.g., bar charts, pie charts, heatmaps) do you find useful for exploring data?

- Do you prefer dashboards that provide high-level summaries or detailed drill-down capabilities?
- How important is real-time data in your movie analysis?

### c) Customization & Integration

- What filters or search criteria would be most useful for you?
- Do you prefer a dashboard with default templates, or would you like the ability to customize reports?

## RESPONSES

Participant A – A casual moviegoer who enjoys exploring movie ratings and audience reviews.

### a) Personal Information

Full Name: Jayden

Occupation: Douglas College Student

Age: 21

Location: Vancouver, Canada

Interests: Watching movies, exploring new genres, staying updated on trending films, and reading audience reviews.

Movie Preferences and Habits:

- Favorite Genres: Action, comedy, and sci-fi.
- Movie-Watching Frequency: Watches movies 2–3 times a week, often on weekends or during breaks from studies.
- Decision-Making Process: Relies heavily on ratings, reviews, and recommendations from friends or social media.

### b) User Interaction & Experience

- **How do you typically explore and analyze movie-related data?**

“Okay, so I usually just go to IMDb to check ratings and reviews, or Netflix to see what’s trending or watch trailers on YouTube. I also check Instagram and Facebook too,

sometimes my friends post about cool movies or there are memes about them. Basically, I just vibe with whatever's popular or looks good.”

- **What key features or functionalities would you like to see in a movie analytics dashboard?**

“I’d love something super easy to use that shows me, like, the top-rated movies and what’s trending. Maybe a feature that recommends movies based on what I’ve watched before? It’d be cool to compare movies by country or year, for example, ‘What were the best movies in 2020?’ or ‘What’s popular in Korea right now?’”

- **Have you used any movie analytics dashboards before? If so, what did you like/dislike about them?**

“Nah, I haven’t used any fancy dashboards or anything. I mostly just use IMDb or Netflix. I like how IMDb shows ratings and reviews, but I wish it was more interactive, like letting me filter movies by year or genre. Netflix is great for recommendations, but sometimes it feels like it’s recommending the same stuff over and over.”

### c) Data Presentation & Interpretation

- **What type of visualizations (e.g., bar charts, pie charts, heatmaps) do you find useful for exploring data?**

“Bar charts and pie charts are easy to understand. Like, a bar chart showing the top 10 movies would be cool. Heatmaps seem kind of complicated, I’d probably need someone to explain what I’m looking at.”

- **Do you prefer dashboards that provide high-level summaries or detailed drill- down capabilities?**

“I’d rather have a quick summary first. If I want more details, I can click on something to see them. But honestly, if there’s too much information, I’ll probably just zone out.”

- **How important is real-time data in your movie analysis?**

“I’m interested in real-time data because I can be updated to trending movies and stuffs.”

#### **d) Customization & Integration**

- **What filters or search criteria would be most useful for you?**

“Filters by genre, release year, and streaming platform would be most useful. Maybe one for ratings too, like, only show me movies with 7/10 or higher. I don’t want to waste my time on bad movies.”

- **Do you prefer a dashboard with default templates, or would you like the ability to customize reports?**

“Default templates are fine for me since I’m not very tech-savvy. However, having a few customization options, like choosing which metrics to display, would be nice.”

Participant B – A data analyst who works with datasets and uses analytical tools.

Full Name: Lead Dinh

#### **a) Personal Information:**

- **What is your position in the company?**

“I am an Entry-Level Data Analyst at Layfield company. My role involves assisting senior analysts with data collection, cleaning, and visualization.”

- **What types of datasets do you typically work with in your role?**

“I mostly work with sales data, customer feedback, and website analytics. My tasks include cleaning the data, creating reports, and helping the team identify trends or areas for improvement.”

#### **b) User Interaction & Experience**

- **How do you typically explore and analyze data?**

“I primarily use tools like Tableau, Power BI, and Python for data analysis. My workflow involves cleaning and preprocessing data, creating visualizations, and building dashboards.”

- **What key features or functionalities would you like to see in an analytics dashboard?**

"I'd like to see a dashboard that is intuitive yet powerful, with features like advanced filtering, predictive analytics, and the ability to handle large datasets efficiently."

- **Have you used any analytics dashboards before? If so, what did you like/dislike about them?**

"Yes, I've used Power BI and Tableau for basic tasks. I like how they make it easy to create visualizations, but I sometimes find the interface overwhelming, especially when trying to use advanced features."

#### c) Data Presentation & Interpretation

- **What type of visualizations (e.g., bar charts, pie charts, heatmaps) do you find useful for exploring data?**

"I find bar charts, line graphs, and scatter plots most useful for exploring trends and correlations. Heatmaps are great for identifying patterns in large datasets, but they need to be well-designed to avoid confusion."

- **Do you prefer dashboards that provide high-level summaries or detailed drill- down capabilities?**

"As an entry-level analyst, I prefer dashboards that provide high-level summaries with the option to drill down if needed. Too much detail upfront can be overwhelming, but having the ability to explore deeper is important for learning."

- **How important is real-time data in your movie analysis?**

"Real-time data isn't critical for most of my tasks right now, but I can see its value for certain projects, like monitoring website traffic or sales performance. For now, I mostly work with historical data."

#### d) Customization & Integration

- **What filters or search criteria would be most useful for you?**

"Basic filters like date range, category, and numerical thresholds are most useful for me. I'd also appreciate filters that allow me to compare subsets of data, like region or product type."

- **Do you prefer a dashboard with default templates, or would you like the ability to customize reports?**

"Default templates are great for me because they save time and provide a good starting point. However, having some customization options, like choosing which metrics to display or adjusting chart colors, would be helpful as I grow more confident in my skills."

## APPENDIX B: USER FEEDBACK SURVEY

Timestamp	Age Range	Primary Interest	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10	Q 11	Q 12	Q 13	Q 14	Q 15	Comment
05/04/2025	21–30	Enthusiast, CS Student	5	3	4	4	4	4	4	4	2	4	4	4	4	4	4	When I save changes in profile, it clears my password .
05/04/2025	16–20	Enthusiast	5	4	5	3	4	5	5	4	4	4	4	4	5	4	5	Nothing : ) The app is good tho.
06/04/2025	21–30	Enthusiast	4	2	4	4	3	4	5	5	5	5	5	5	5	3	3	Predictor confusing , charts need real-time data.
08/04/2025	21–30	Enthusiast, CS Student	4	5	5	4	4	4	4	5	4	4	5	4	5	4	4	Sharing should be limited to friends, doesn't make sense otherwise .
09/04/2025	21–30	Enthusiast, Analyst	3	3	4	4	4	2	4	4	4	5	5	4	5	4	4	Good web, still has flaws but still :>
09/04/2025	30–40	Other	3	2	2	2	4	2	3	4	4	2	4	2	2	4	3	Data outdated, crashed during export, predictions off.
10/04/2025	21–30	CS Student	4	4	4	5	4	4	5	4	5	5	4	4	5	4	3	Nice interface, but login security weak (random)

																	email/password).
10/04/2025	21–30	Enthusiast	5	5	5	5	5	5	4	4	4	5	5	4	4	5	Good.
11/04/2025	21–30	Enthusiast, CS Student	5	4	5	4	5	5	5	4	4	4	5	4	3	4	[No comment provided]
11/04/2025	21–30	Analyst, CS Student	4	2	4	4	4	2	4	3	4	4	3	4	4	4	Notifications UI needs work, love the rating emoji, predictor unclear.

### Functionality:

- Q1: The dashboard's features (e.g., watchlists, trend analysis, predictions) work as expected.
- Q2: The Success Predictor provides accurate and useful movie outcome forecasts.
- Q3: The Movie List allows seamless addition and removal of movies to/from watchlists.
- Q4: The data export feature in Explore Global Trends produces usable outputs (e.g., CSV files).

### Reliability:

- Q5: The dashboard operates without crashes or errors during use.
- Q6: Watchlists and notifications save correctly across sessions.

### Usability:

- Q7: The interface (e.g., sidebar, buttons) is intuitive and clear.
- Q8: Instructions and labels are straightforward.
- Q9: Visualizations are easy to understand.

### Efficiency:

- Q10: Logging in and accessing features feels seamless.
- Q11: The dashboard performs consistently across browsers (e.g., Chrome, Firefox).
- Q12: The dashboard loads features promptly.

Q13: Filtering data or sharing movies feels responsive.

**Portability:**

Q14: The dashboard runs smoothly on different devices (e.g., laptop, tablet).

Q15: The dashboard is accessible via web browsers without installation hassles.

## APPENDIX C: INSTALLATION GUIDE

### Installation Instructions

To run the Movie Trends and Insights Dashboard locally, follow these steps:

#### Prerequisites

- Python 3.8 or higher installed on your system.
- Git installed (optional, for cloning the repository).
- An internet connection (for TMDb API calls).

#### Step 1: Clone the Repository

1. Open a terminal or command prompt.
2. Run the following command to clone the repository:

```
git clone https://github.com/LanDinh15/W25_4495_S2_LanD.git
```

3. Navigate to the project directory:

```
cd W25_4495_S2_LanD/Implementation
```

#### Step 2: Install Required Packages

```
pip install streamlit pandas numpy seaborn matplotlib plotly requests streamlit-plotly-events
```

#### Step 3: Configure the TMDb API

The Movie Checklist feature uses the TMDb API for movie searches. The API key is hardcoded in `MovieChecklist.py` (`e206cf8b0ba47f28233d0a28ff83c414`) for simplicity. For security in production:

1. Consider replacing the hardcoded key with an environment variable:
  - Create a `.env` file in the `Implementation` directory.
  - Add: `TMDB_API_KEY=your_personal_key`.
  - Modify `MovieChecklist.py` to load the key using `os.getenv("TMDB_API_KEY")`.
2. Obtain a personal key at [themoviedb.org](https://themoviedb.org) if needed, though the provided key is functional for testing.

## **Step 4: Run the Dashboard**

Start the Streamlit application by running the following command:

```
streamlit run Welcome.py
```

## **Step 5: Using the Dashboard**

Authentication: Register a new account or use test credentials (Username: admin, Password: admin123) to access Profile and Movie Checklist features.

Navigation: Use the sidebar to explore Gross Earnings Explorer, Explore Global Trends, Success Predictor, or Profile tabs.

Troubleshooting:

- If the app fails to load, verify Python and library installations.
- Ensure an internet connection for TMDb API functionality.
- Check the terminal for error messages.

## APPENDIX D: USER GUIDE

### USER INSTRUCTIONS: Movie Trends Dashboard

Welcome to the **Movie Trends Dashboard**, a Python-based web application built with Streamlit that lets users explore movie data, predict success, manage their watchlist, and personalize cinematic experience! This guide will highlight how to use the dashboard, step-by-step, with screenshots to illustrate each feature. No matter what background you are, this tool has something for you.

---

#### A. SYSTEM REQUIREMENTS

To operate the Movie Trends Dashboard, the following prerequisites must be met:

- **Operating System:** Compatible with Windows, macOS, or Linux.
- **Python Version:** Python 3.8 or higher.
- **Dependencies:** Required Python libraries include streamlit, pandas, plotly, pickle, requests, and numpy. Install these via the command:  
*pip install streamlit pandas plotly requests numpy*
- **Project Files:** Source code (Welcome.py, GrossEarnings.py, GlobalTrend.py, MovieChecklist.py, SucessPredictor.py, auth.py, predictor.py) and datasets (imdb\_top\_1000.csv, netflix\_titles.csv) must be present in the working directory.

#### B. LAUNCHING THE APPLICATION

To initiate the Movie Trends Dashboard:

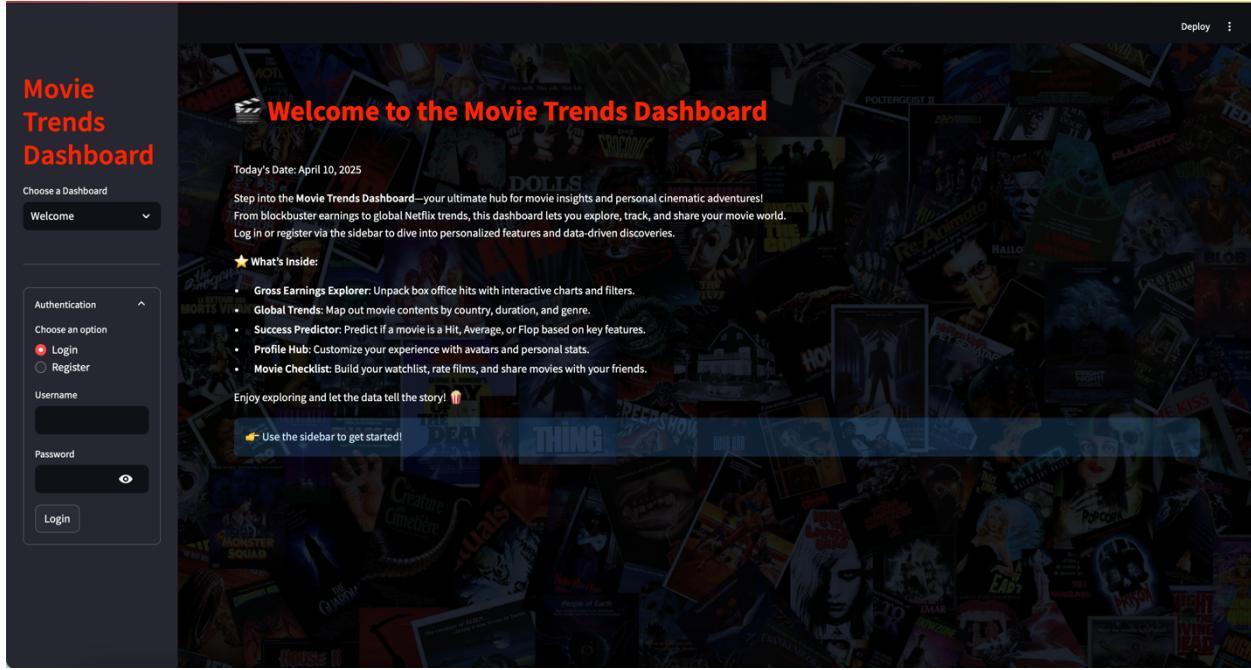
1. Open a terminal or command prompt.
2. Navigate to the directory containing Welcome.py (e.g., Implementation/).
3. Execute the following command: *streamlit run Welcome.py*
4. Access the application by opening a web browser and navigating to <http://localhost:8501>.
5. Upon launching the application, the user is directed to the Welcome page by default.

#### C. OPERATIONAL PROCEDURES

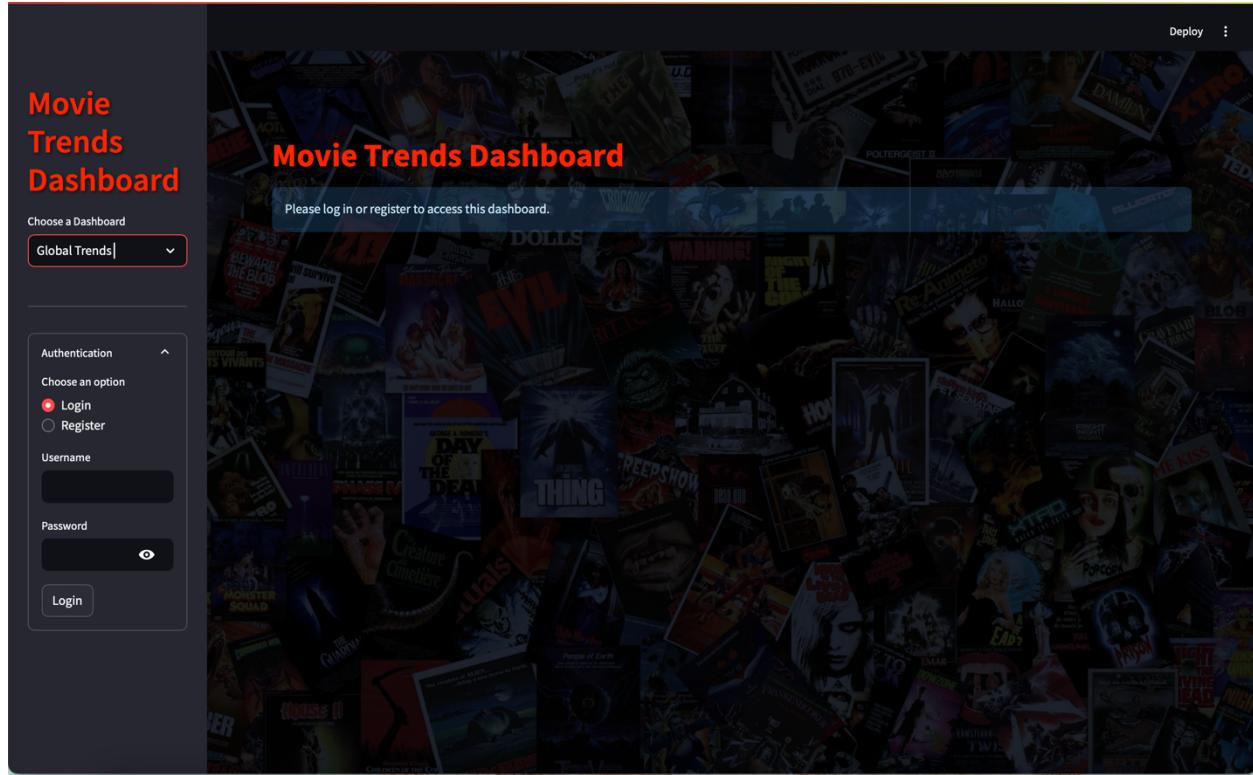
##### 1. WELCOME PAGE

The Welcome page provides an overview of the dashboard's capabilities.

The interface displays a title, the current date (e.g., "April 10, 2025"), and a detailed summary of available features, including Gross Earnings Explorer, Global Trends, Success Predictor, Profile Hub, and Movie Checklist. Users are prompted to authenticate via the sidebar.



If users do not log in or register, access is limited to the Welcome and Gross Earnings dashboards only. To unlock the full range of functionalities, users must complete the authentication process.

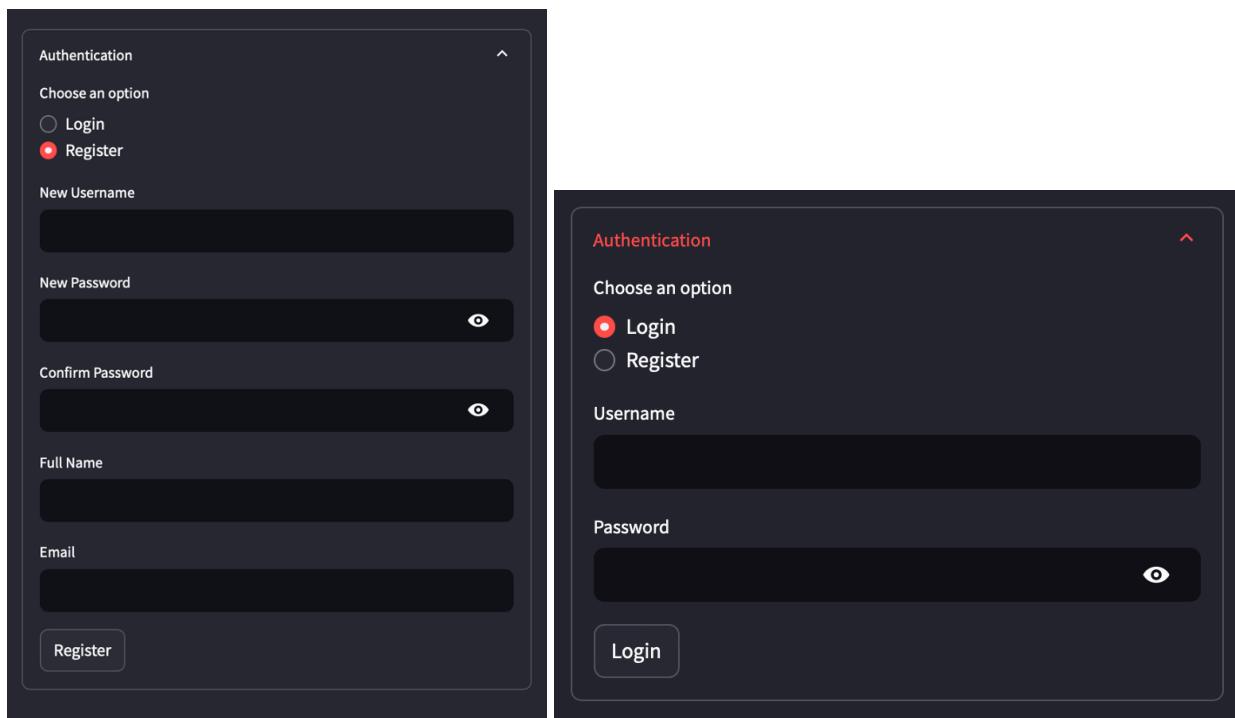


## 2. USER AUTHENTICATION

Authentication enables access to personalized and restricted features.

Procedure:

- Registration (for new users):
  - Select "Register" from the radio options.
  - Input a unique username, password, full name, and email address.
  - Click the "Register" button to create an account. A success message will confirm registration.
  
- Login (for existing users):
  - Select "Login" from the radio options.
  - Enter the registered username and password.
  - Click the "Login" button. Upon successful authentication, the sidebar updates to display the user's username and expanded dashboard options.



For testing purposes, users may utilize the following pre-existing account credentials:

Username: admin

Password: admin123

### 3. GROSS EARNINGS EXPLORER

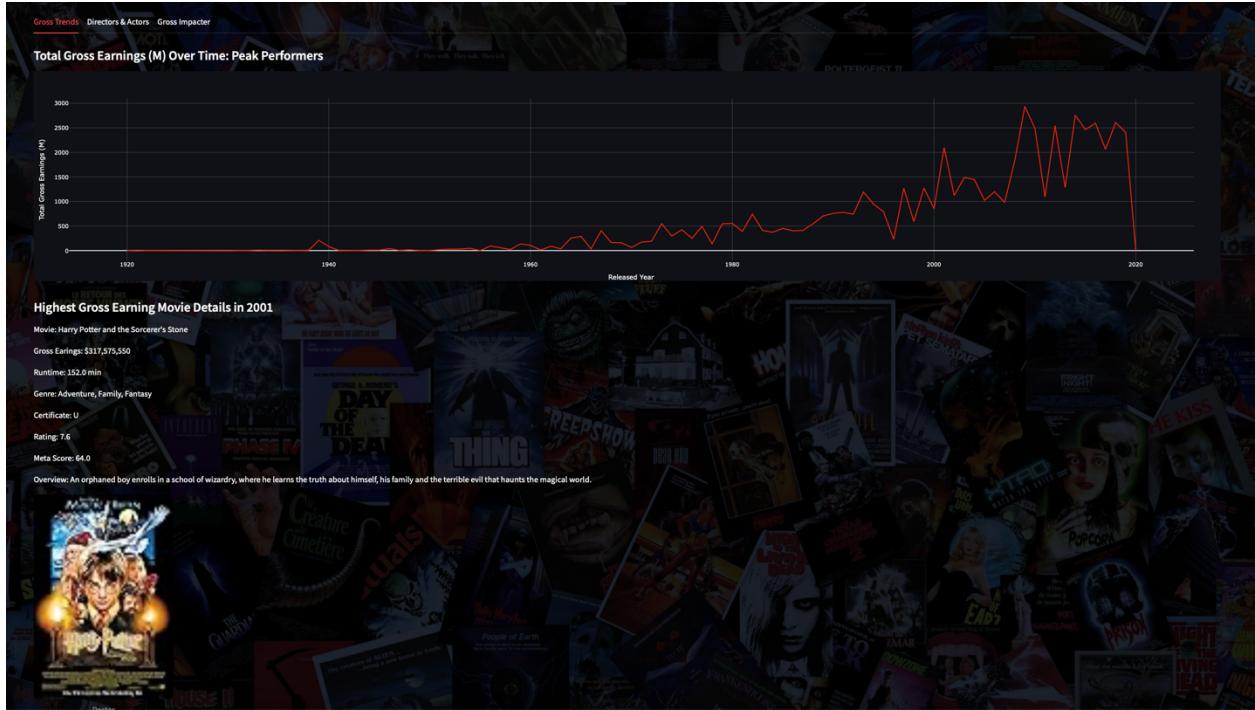
This module analyzes box office performance data from IMDb's top 1000 movies. Users can access this by selecting "Gross Earnings" from the sidebar dropdown menu.

Begin with the Dataset Snapshot, it displays four essential metrics in a four-column layout, calculated based on the user's applied filters (e.g., IMDb Rating, Meta Score, Released Year, and Genres).

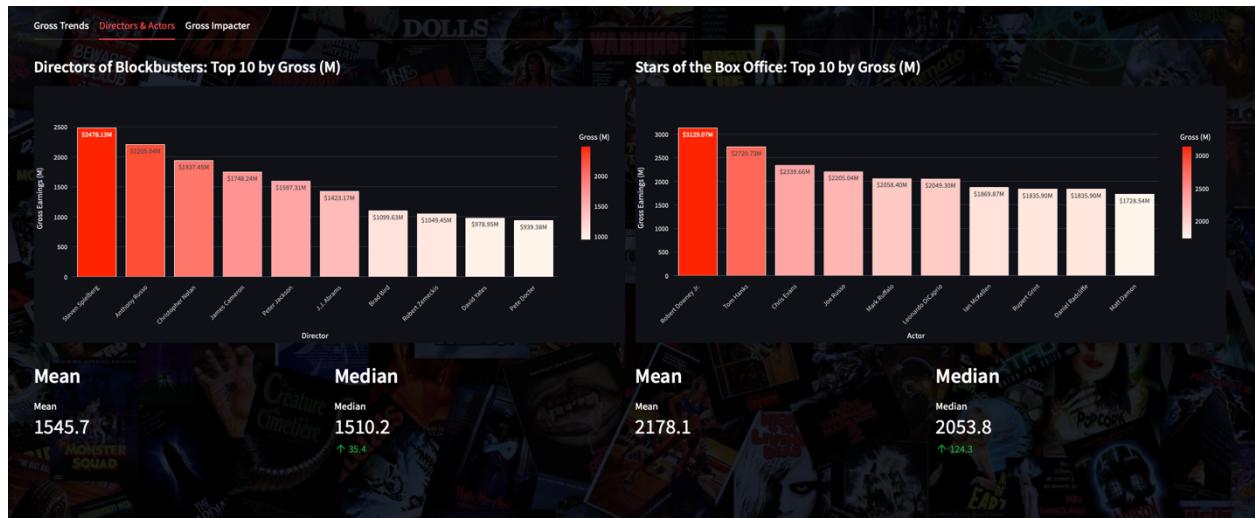
Functionalities:

- Sidebar Filters:
  - Adjust sliders for IMDb Rating (e.g., 7.0–9.0), Meta Score (e.g., 50–100), and Released Year (e.g., 2000–2020).
  - Select multiple genres (e.g., Action, Drama) via a multiselect widget.
- Tabs:

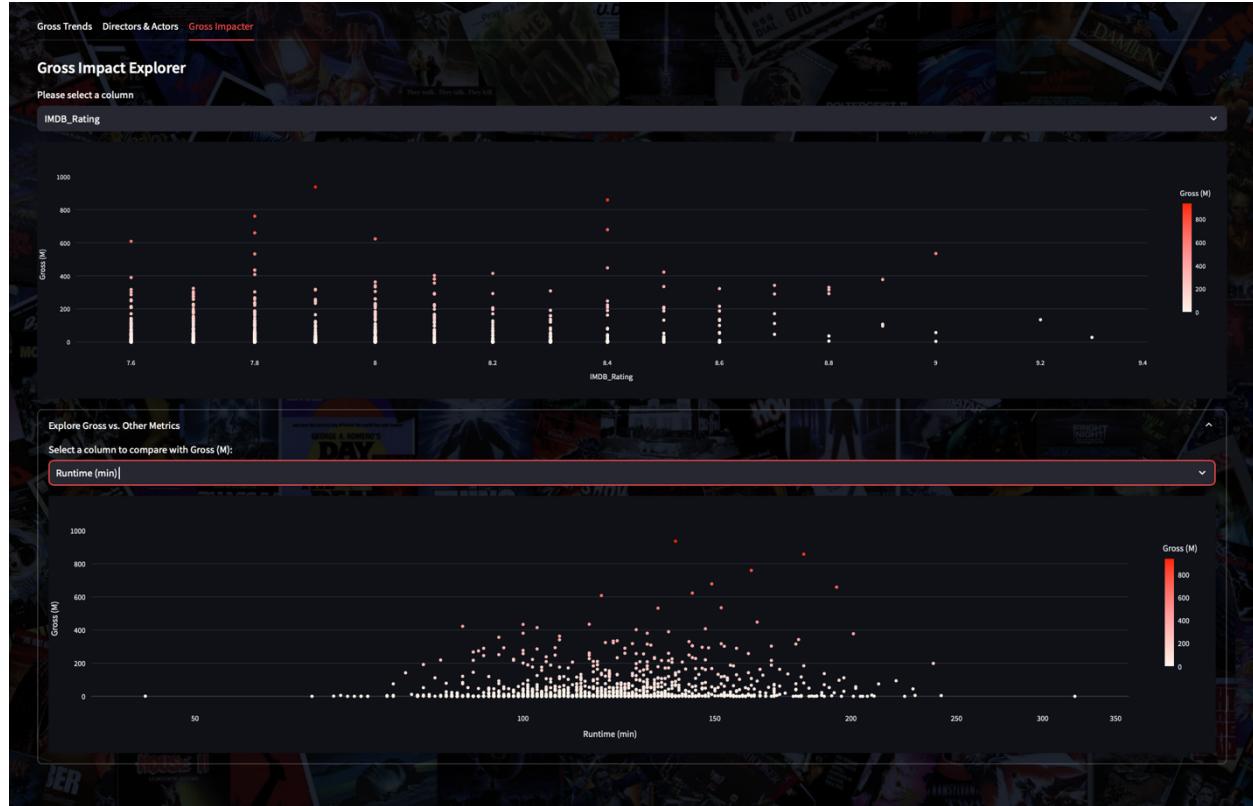
- **Gross Trends:** Displays a line chart of total gross earnings over time. Clicking a peak reveals details of the highest-grossing movie for that year (e.g., title, gross, poster).



- **Directors & Actors:** Presents bar charts of the top 10 directors and actors by gross earnings, with hover-over statistics, plus mean and median earnings below each chart.



- **Gross Impactor:** Offers scatter plots to explore correlations between gross earnings and variables such as runtime or IMDb rating.

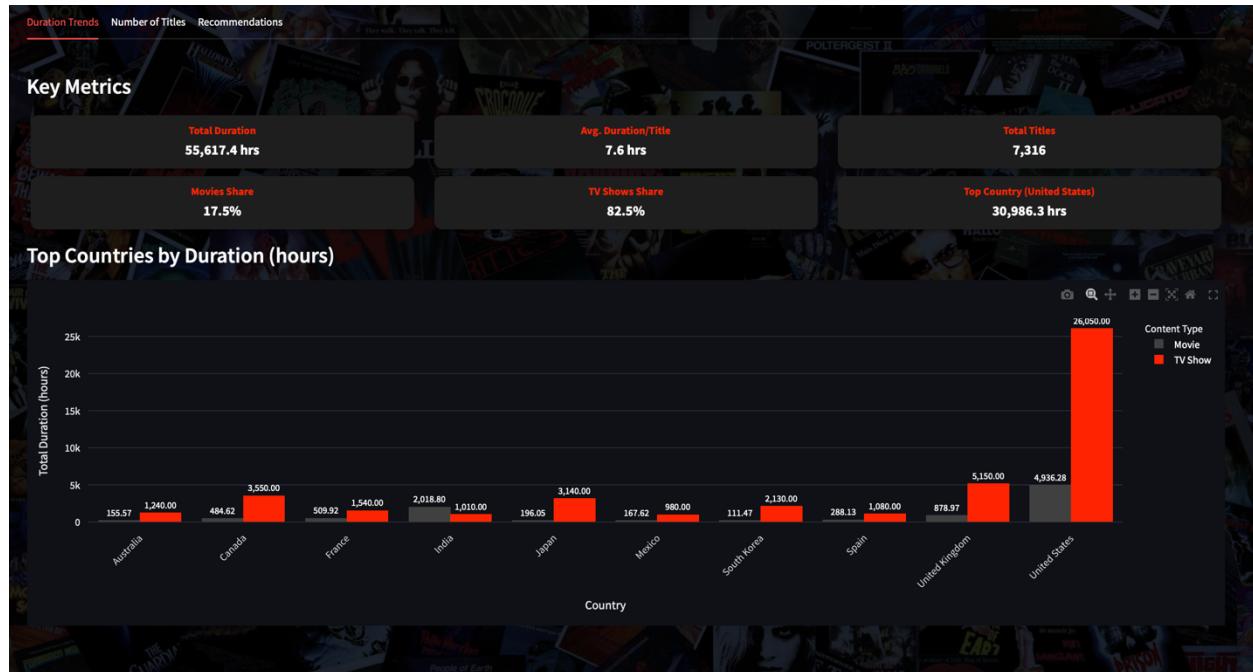


#### 4. GLOBAL TRENDS

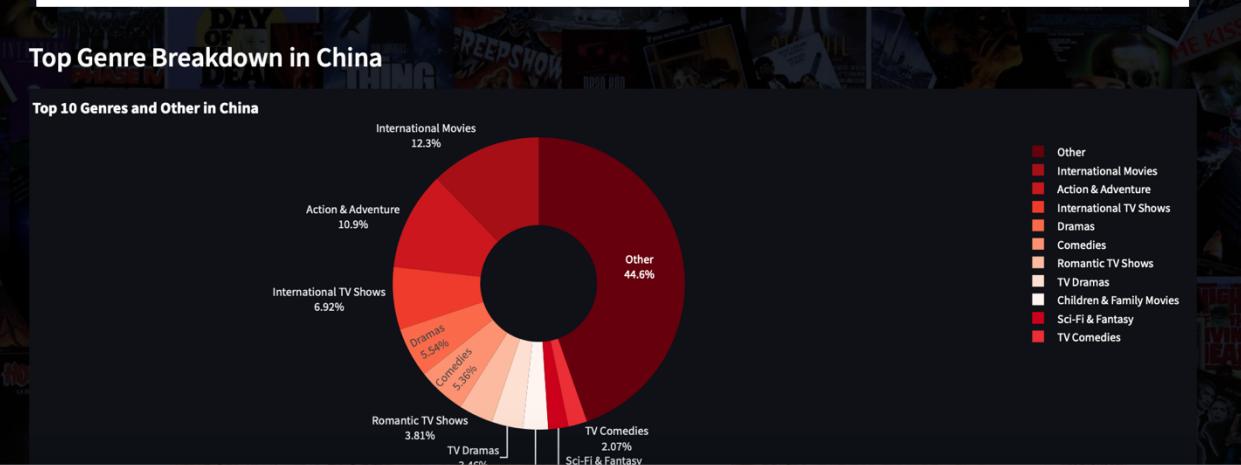
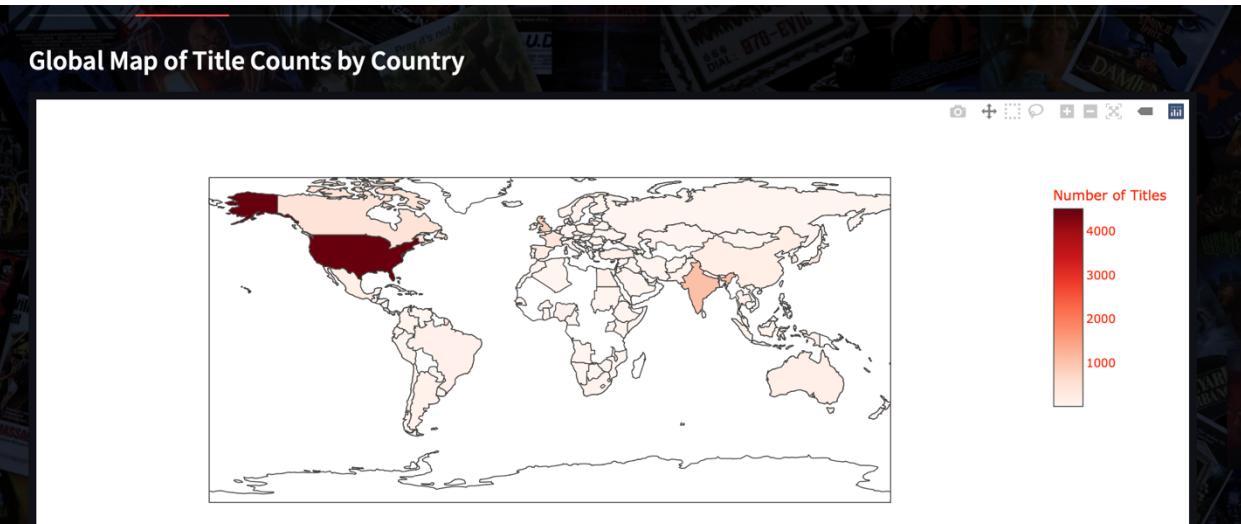
This module examines Netflix content trends across countries and genres. Users can access this by selecting “Global Trends” from the sidebar dropdown menu (requires authentication).

Functionalities:

- Sidebar Filters:
- Specify a date range, content type (Movie/TV Show), rating category (e.g., Kids, Adult), and countries.
- Tabs:
- **Duration Trends:** Features a bar chart of total duration (hours) for the top 10 countries, accompanied by key metrics (e.g., total duration, average per title).



- **Number of Titles:** Displays a choropleth map of title counts by country. Clicking a country generates a genre breakdown donut chart.



- **Recommendations:** Provides five random title recommendations based on a selected genre, exportable as a CSV file.

**Personalized Recommendations**

Select a genre you prefer:

Action & Adventure

Recommended titles for Action & Adventure in all countries:

Title	Type	Genre	Rating	Release Year
Acts of Violence	Movie	Action & Adventure	R	2018
Charlie's Angels	Movie	Action & Adventure, Comedies	PG-13	2000
Krishna Cottage	Movie	Action & Adventure, Horror Movies, International Movies	TV-14	2004
Sniper: Legacy	Movie	Action & Adventure	R	2014
The Art of the Steal	Movie	Action & Adventure, Comedies, Independent Movies	R	2013

Download Recommendations as CSV

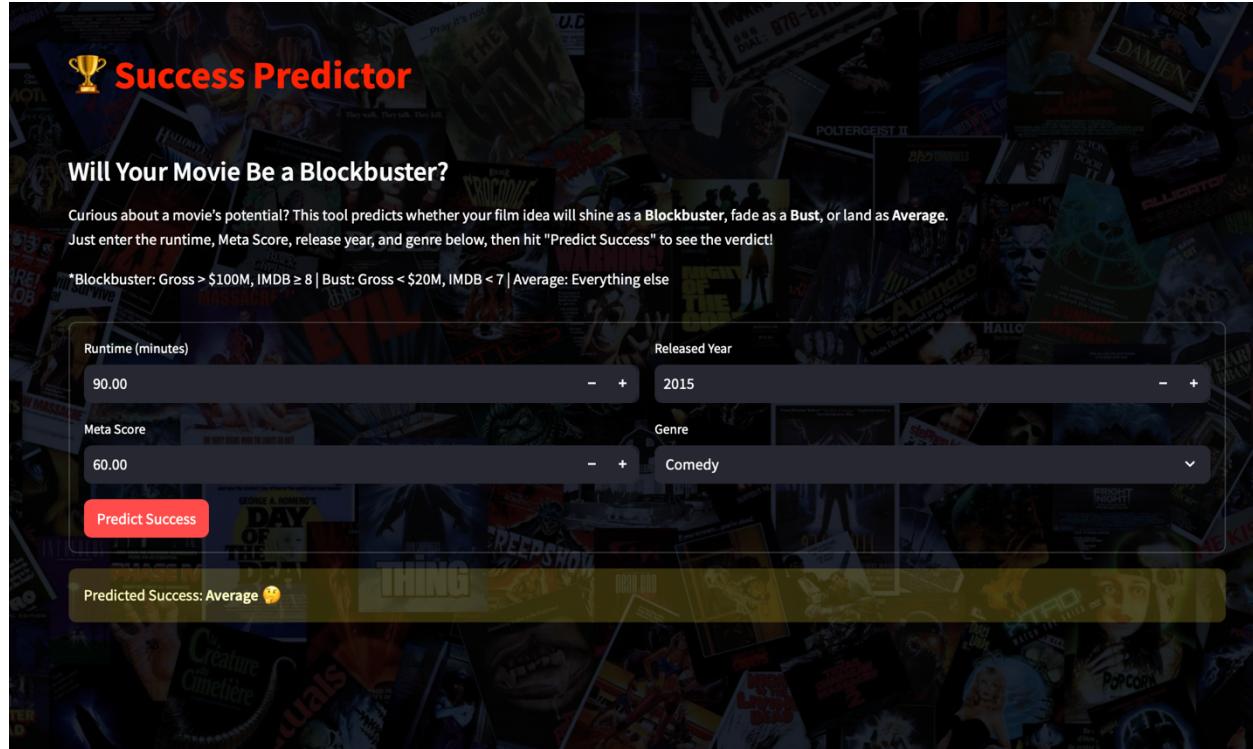
## 5. SUCCESS PREDICTOR

Predicts a movie's success category (Blockbuster, Bust, or Average) based on input features, utilizing a pre-trained machine learning model. Users can access this by selecting "Success Predictor" from the sidebar dropdown menu (requires authentication).

Procedure:

- Review the introductory text outlining the prediction process.
- Input the following parameters:
  - Runtime (e.g., 120 minutes).
  - Meta Score (e.g., 75).
  - Released Year (e.g., 2020).
  - Genre (e.g., Action).
- Click "Predict Success" to generate a prediction.

A result is displayed in a colored box: green for "Blockbuster" red for "Bust" or yellow for "Average" alongside the input parameters. A Blockbuster represents movies with gross earnings exceeding \$100 million and IMDb ratings of 8 or higher; a Bust represents movies with gross earnings below \$20 million and IMDb ratings below 7; an Average encompasses all other movies not meeting the Blockbuster or Bust criteria.



## 6. PROFILE MANAGEMENT

Allows users to manage their profile and maintain a movielist (search for movies, build a watchlist, rate films, and share recommendations.). Users can access this by selecting “Profile” from the sidebar dropdown menu (requires authentication).

Tabs:

- My Profile Tab:
  - Displays user details (username, full name, DOB, email, avatar).
  - Edit by clicking "Update My Profile" :
    - Updating *Full Name*, *Date of Birth*, or *Email*.
    - Uploading a new *Avatar* (PNG/JPG).
    - Optionally changing *Password*.
  - Clicking *Save Changes* to confirm or *Close* to cancel.

### Profile Details



Your Avatar

Username: admin  
Full Name: User 1  
Date of Birth: 2000-05-15  
Email: [admin15@example.com](mailto:admin15@example.com)

**Update My Profile**

### Update Profile

Full Name

Date of Birth

Email

New Password

Confirm New Password

Upload Avatar (PNG/JPG)  
 Drag and drop file here  
Limit 200MB per file • PNG, JPG, JPEG

**Save Changes** **Close**

- My Movielist Tab:
- **Discover Movies:** Search for movies or browse popular titles, adding them to the watchlist.

 My Movielist

(0) ▾

Discover Movies To Watchlist History

### Search Movies

Enter movie title

Popular Movies



Discover Movies To Watchlist History

### Search Movies

Enter movie title

avatar

Previous Next

Did you mean:

Avatar: Creating the World of Pandora      The King's Avatar: For the Glory      Avataro Sentai Donbrothers vs. Zenkaiger

**Avatar (2009)**



Release Date: 2009-12-15  
Rating: 7.588/10

Overview: In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and protecting an alien civilization.

Add to Checklist

**Avatar (2006)**



Release Date: 2006-04-11  
Rating: 5.941/10

Overview: Tension mounts between a quadriplegic man and his wife as she prepares a bath for him.

Add to Checklist

- **To Watchlist:** Manage movies added to the watchlist with the following capabilities:
  - View movies users added but haven't watched, displayed with details such as title, rating (e.g., 8.5/10), and a poster image if available.
  - Click ✓ to mark a movie as watched, which updates its status and moves it to the "History" tab with the current date.
  - Click ✘ to remove a movie from the watchlist, deleting it from your list entirely.
  - Click 🙋 to share a movie with another user. Here, Users can select a friend's username from a dropdown list of registered users and type a custom message in a text input field and send it, notifying the selected friend via their notifications panel

**My Movielist**

Discover Movies **To Watchlist** History

Cosmic Chaos (5.714/10)

The Quiet Ones (4.5/10)

✓ ✘ 🙋

✓ ✘ 🙋

**My Movielist**

Discover Movies **To Watchlist** History

Cosmic Chaos (5.714/10)

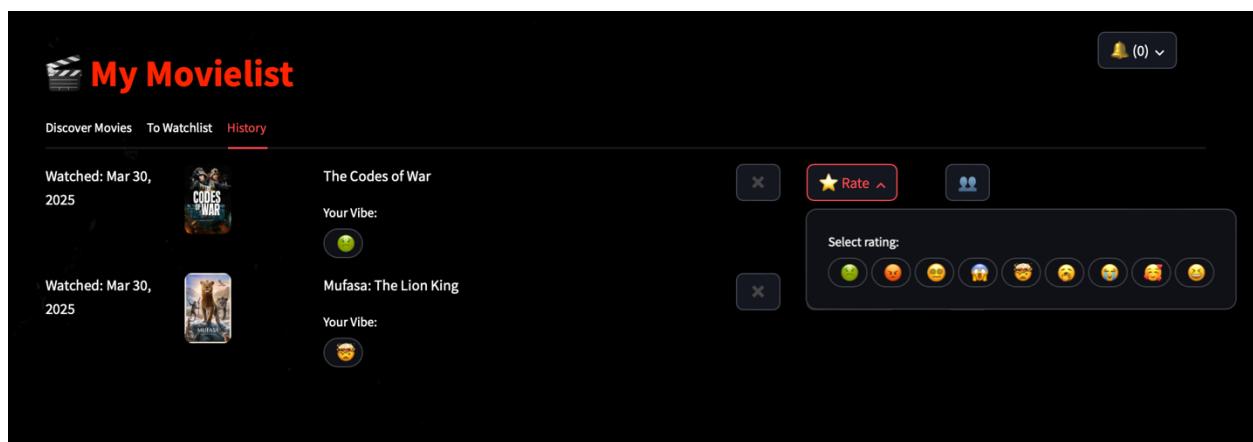
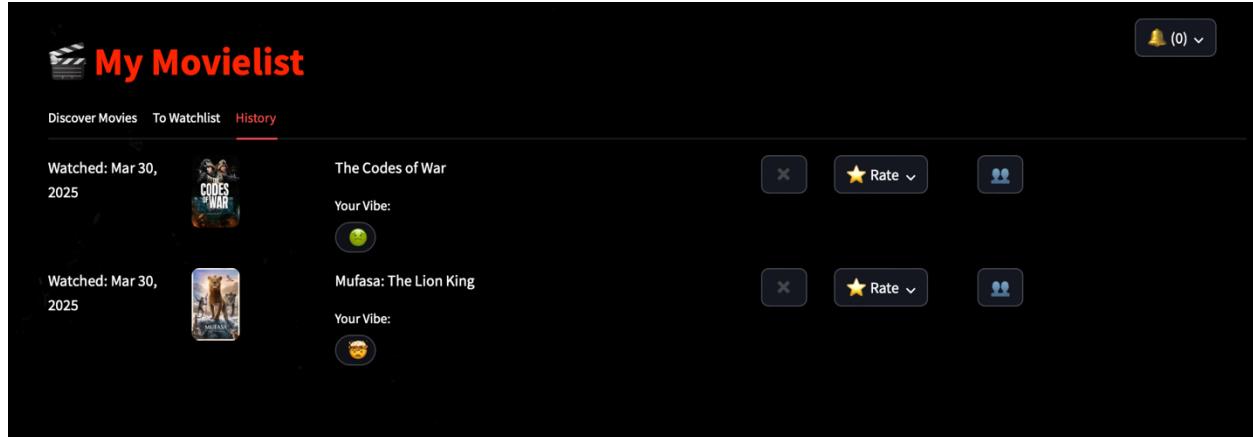
The Quiet Ones (4.5/10)

✓ ✘ 🙋

Share 'Cosmic Chaos' with:  
testuser

Check this movie ➤

- **History:** Rate watched movies (1–9 with emojis) or remove them from the list.



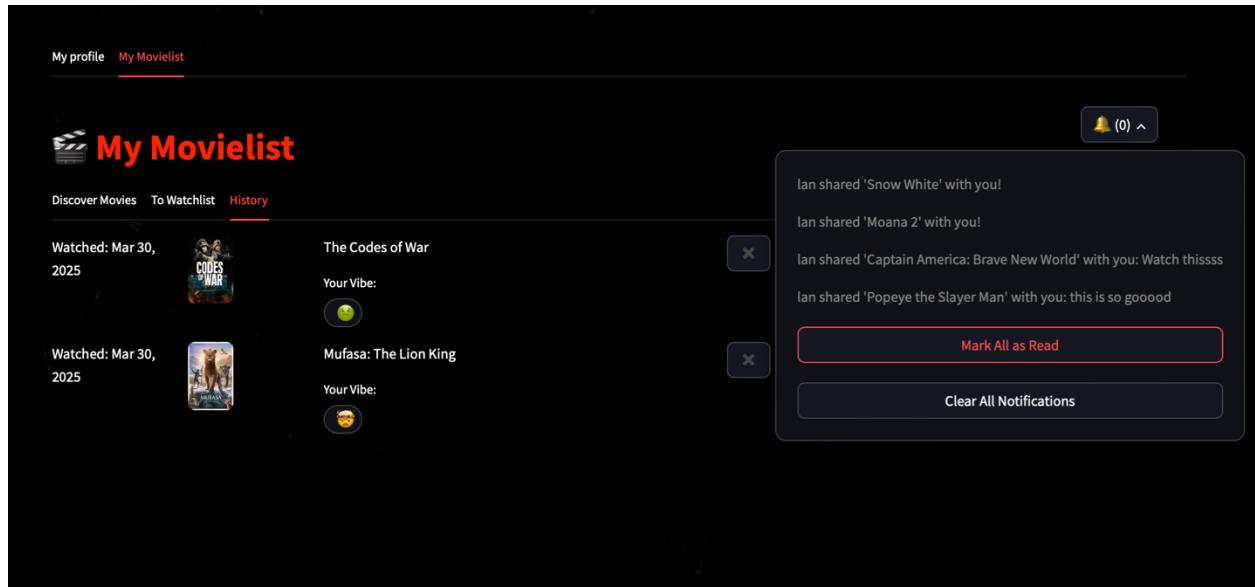
## 7. NOTIFICATION MANAGEMENT

The Notification feature enables users to receive, view, and manage messages related to movie sharing. Notifications are accessed via a bell icon (🔔) located at the top-right of the interface, displaying the count of unread messages.

Functionalities:

- **View Notifications:** Clicking the bell icon opens a popover displaying a list of messages (e.g., "User1 shared 'Inception' with you: Check this out!"), with unread messages in bold red text and read messages in gray.
- **Mark as Read:** Click "Mark All as Read" within the popover to update all unread notifications to read status, changing their appearance to gray.

- Clear Notifications: Click "Clear All Notifications" to remove all messages from the list, resetting the notification count to zero.



## D. CONCLUSION

The Movie Trends Dashboard offers a robust platform for movie data analysis and personal movie management. By following this guide, users can effectively navigate its features, from exploring gross earnings to predicting movie success and maintaining a watchlist. For further assistance, consult the project documentation or contact the development team.