User Interface Design and Integration Report

Huaye Li

1. Introduction

The sixth week of the Personalized Movie Recommendation System project focused on the design, development, and integration of a user interface that allows users to search for and receive personalized movie recommendations in real-time. The primary objective of this stage was to translate the backend AI model into a usable, accessible, and visually engaging web application using modern frontend technologies. This report outlines the user interface (UI) design process, technologies and tools used, integration with the recommendation model, and the associated implementation challenges.

2. UI Design Process

2.1 User-Centered Design Philosophy

The UI was developed with a user-centered design (UCD) framework, focusing on simplicity, usability, and responsiveness. The core idea was to ensure that users with varying technical backgrounds could interact with the recommendation system without any prior training or assistance.

Key design objectives included:

- Minimalistic layout for ease of navigation
- Visual consistency with a custom theme reflecting the Clemson University identity
- Clear user prompts for input (user ID and search query)
- Real-time results display with visual elements such as movie posters and brief descriptions

2.2 Wireframe Development

Initial low-fidelity wireframes were created using **Uizard**, a rapid prototyping tool that enables fast layout design through AI-assisted elements.

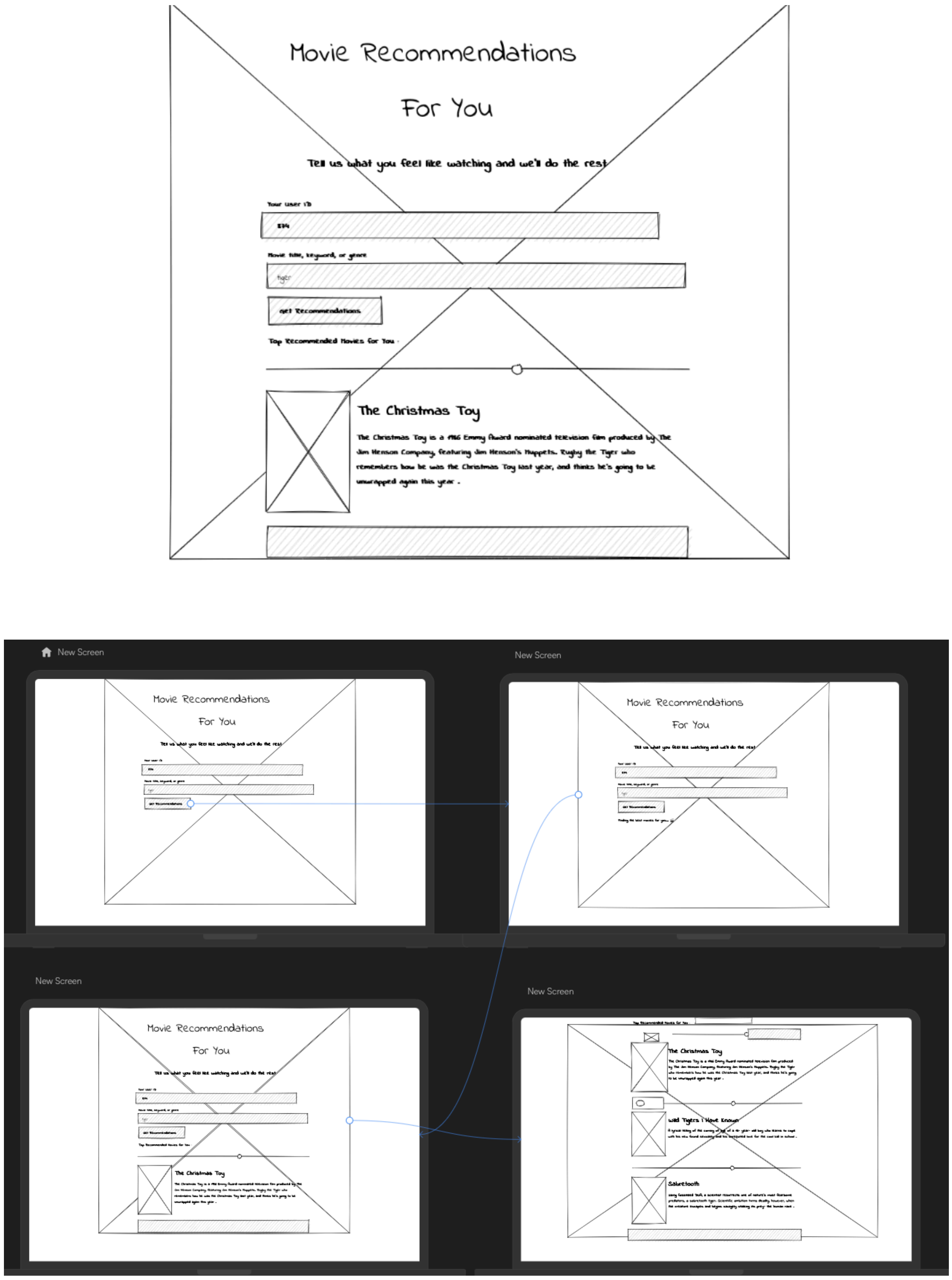

Movie Recommendations
For You
Tell us what you feel like watching and we'll do the rest
Your user ID
Movie title, keyword, or genre
tiger
Get Recommendations
Top Recommended Movies for You
The Christmas Toy
The Christmas Toy is a 1986 Emmy Award nominated television film produced by The Jim Henson Company, featuring Jim Henson's Muppets. Rugby the Tiger who remembers how he was the Christmas Toy last year, and thinks he's going to be unwrapped again this year.
New Screen
Wild Tigers I Have Known
Sabretooth

Figure 1: Wireframes

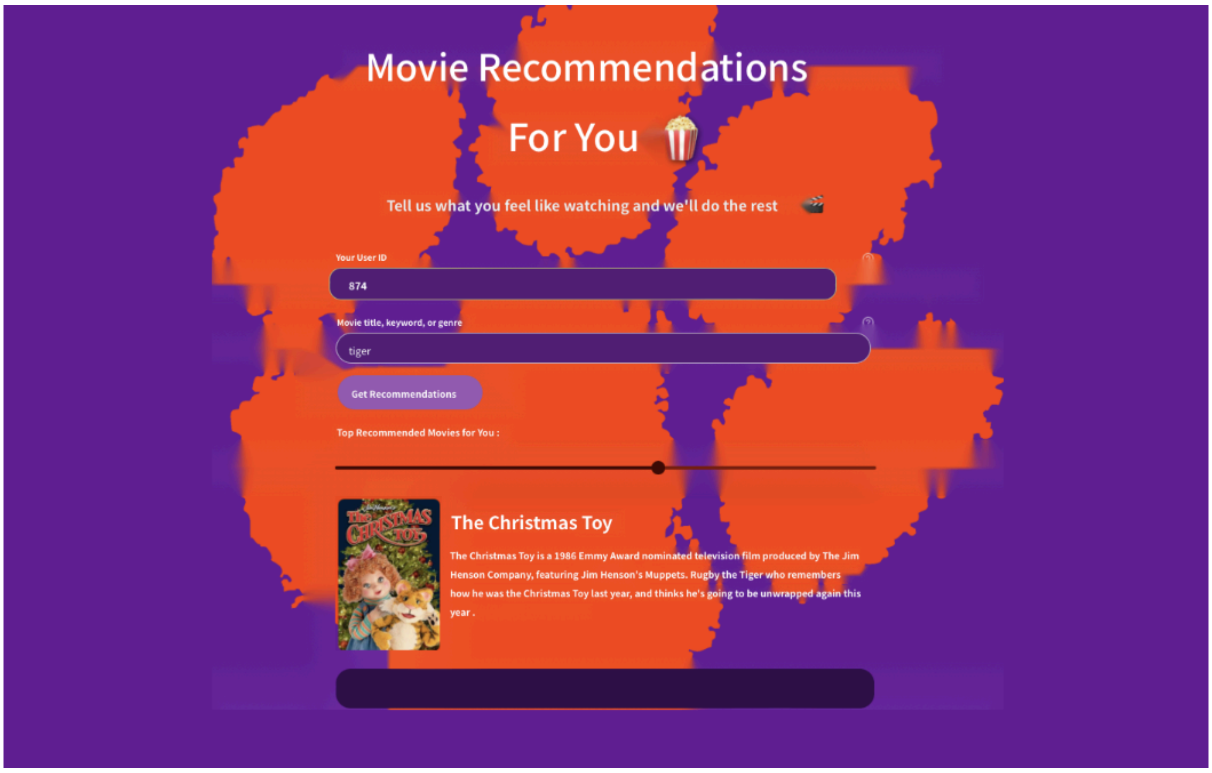
These wireframes provided a blueprint for the logical arrangement of components such as:

- Header and title section

- Input fields for user ID and search term
- Call-to-action button (“Get Recommendations”)
- Scrollable display of recommendation cards with images and text

2.3 High-Fidelity Prototypes

The wireframes were iteratively refined into high-fidelity prototypes that captured the final look and feel of the application. Font styles, color palettes, and spacing were carefully adjusted for optimal visual appeal and accessibility.



Movie Recommendations

For You

Tell us what you feel like watching and we'll do the rest

Your User ID

874

Movie title, keyword, or genre

tiger

Top Recommended Movies for You :

The Christmas Toy

The Christmas Toy is a 1986 Emmy Award nominated television film produced by The Jim Henson Company, featuring Jim Henson's Muppets. Rugby the Tiger who remembers how he was the Christmas Toy last year, and thinks he's going to be unwrapped again this year .

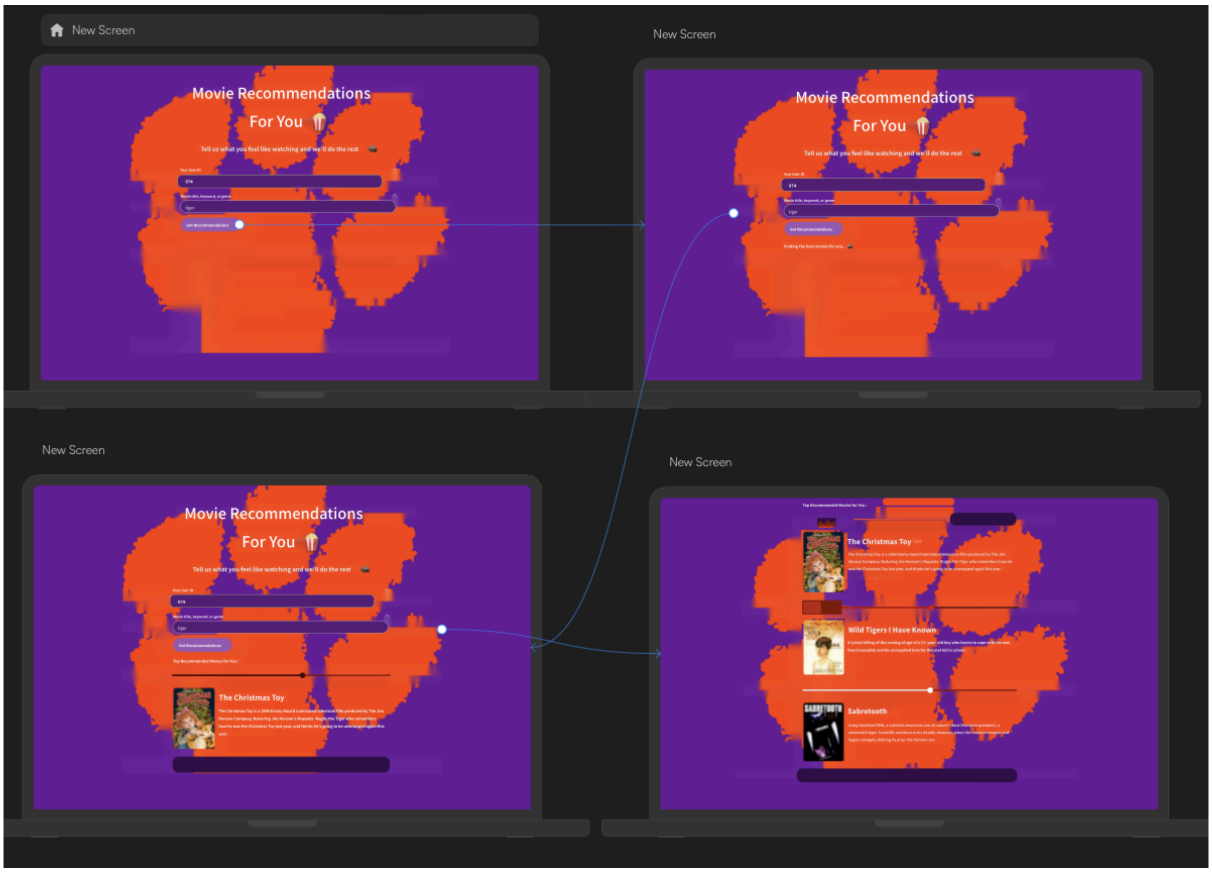


Figure 2: FigHigh-Fidelity Prototypes

These prototypes were used to guide the actual front-end implementation in Streamlit.

2.4 Prototype Access

The complete interactive prototype designed using **Uizard** can be accessed via the following link:

Uizard Prototype – AI Movie Recommendation UI

This prototype includes both wireframes and high-fidelity visual representations of the application layout. It served as a foundational reference throughout the front-end development process and demonstrates the intended user experience across all key interface components.

3. Technologies Used

The front-end was implemented using the following tools and technologies:

- **Streamlit**: A lightweight and Python-native web application framework, ideal for rapid deployment of data science interfaces. It allowed seamless integration with the existing AI model and simplified UI implementation.
- **HTML/CSS (via Streamlit's markdown)**: Used for styling and formatting elements such as titles, text input, and visual alignment.

- **Custom CSS injection**: A custom CSS block was embedded in the app.py file to override default styles and create a visually cohesive interface with a branded color scheme and responsive layout.
- **Uizard**: Used to generate both wireframes and interactive prototypes.
- **OMDb API**: Used to fetch movie posters via IMDb IDs and display them in the interface using Streamlit.image().

4. Integration with AI Model

The integration process involved combining the trained TF-IDF recommendation model with the front-end components to enable dynamic querying. Key integration steps included:

1. **User Input Handling**:
    - Streamlit's number_input and text_input widgets were used to collect user ID and search terms.
    - Input validation was added to prevent errors from non-numeric IDs or empty queries.
2. **Model Invocation**:
    - On button click (st.button("Get Recommendations")), the recommend_movies() function was called with user inputs.
    - The backend executed cosine similarity calculations and returned the top 10 recommended movies.
3. **Poster Fetching and Display**:
    - The fetch_poster_with_cache() function retrieved corresponding poster URLs via the OMDb API.
    - Each recommendation was displayed in a card format using st.columns() and custom HTML styling.
4. **Theme Styling**:
    - A custom theme was applied using embedded CSS, including background imagery, button colors, and input field styles to align with Clemson's visual identity.

5. Usability and Functionality

The Streamlit-based application successfully delivers:

- Real-time movie recommendations personalized by user history and search intent
- A visually clean and navigable interface with no external dependencies
- Poster-enhanced result cards that make exploration more engaging

The application was tested across multiple devices (desktop, tablet) and browsers (Chrome, Safari, Firefox) to ensure cross-platform usability.

6. Challenges Faced

- **Poster Image Unavailability**: Some IMDb entries lacked posters in OMDb, requiring fallback mechanisms to avoid breaking the layout. A default "No Image Available" placeholder was used.

- **Styling Limitations in Streamlit**: Native styling options in Streamlit are limited. To address this, embedded CSS was used creatively to simulate component styling such as input field glow effects and card shadows.
- **Query Mismatches**: Users often entered queries that did not match any content. A fuzzy-matching feature is under consideration for the next version to improve recall.

7. Conclusion

The integration of the AI model with a responsive, branded, and intuitive user interface was successfully completed. The Streamlit-based front-end allows users to interact with the system naturally while receiving personalized results in real-time. The design process—from wireframes to prototypes to implementation—followed modern usability principles and demonstrated strong alignment between user goals and system functionality.