**Components:**

1. **User Interface (Streamlit):**
   * **Category Selection:** Dropdown for choosing movie categories.
   * **Recommendations Display Area:** Shows recommended movies with title, poster, and description.
2. **Backend Server (Python):**
   * **API Handler:** Interacts with the Movie Database/API.
   * **Recommendation Engine:** Processes the data and generates recommendations.
3. **Movie Database/API:**
   * **Movie Data Fetcher:** Retrieves movie details based on the selected category.

**1. Detailed Architecture**

**Frontend (Streamlit):**

* **Category Selection Widget:**
  + **Component:** Dropdown menu.
  + **Function:** Allows users to select a category.
* **Recommendations Display Area:**
  + **Components:**
    - **Movie Title:** Text element for the movie title.
    - **Movie Poster:** Image element for the movie poster.
    - **Movie Description:** Text element for the movie description.

**Backend Server (Python):**

* **API Handler:**
  + **Function:** Fetches data from the Movie Database/API.
  + **Endpoints:**
    - **Movie Data Endpoint:** https://api.themoviedb.org/3/discover/movie
    - **Parameters:** category, api\_key
* **Recommendation Engine:**
  + **Function:** Applies a recommendation algorithm to generate movie suggestions.
  + **Algorithm:** Filters movies based on the selected category.
* **Data Processing Module:**
  + **Function:** Extracts and formats data for the frontend display.

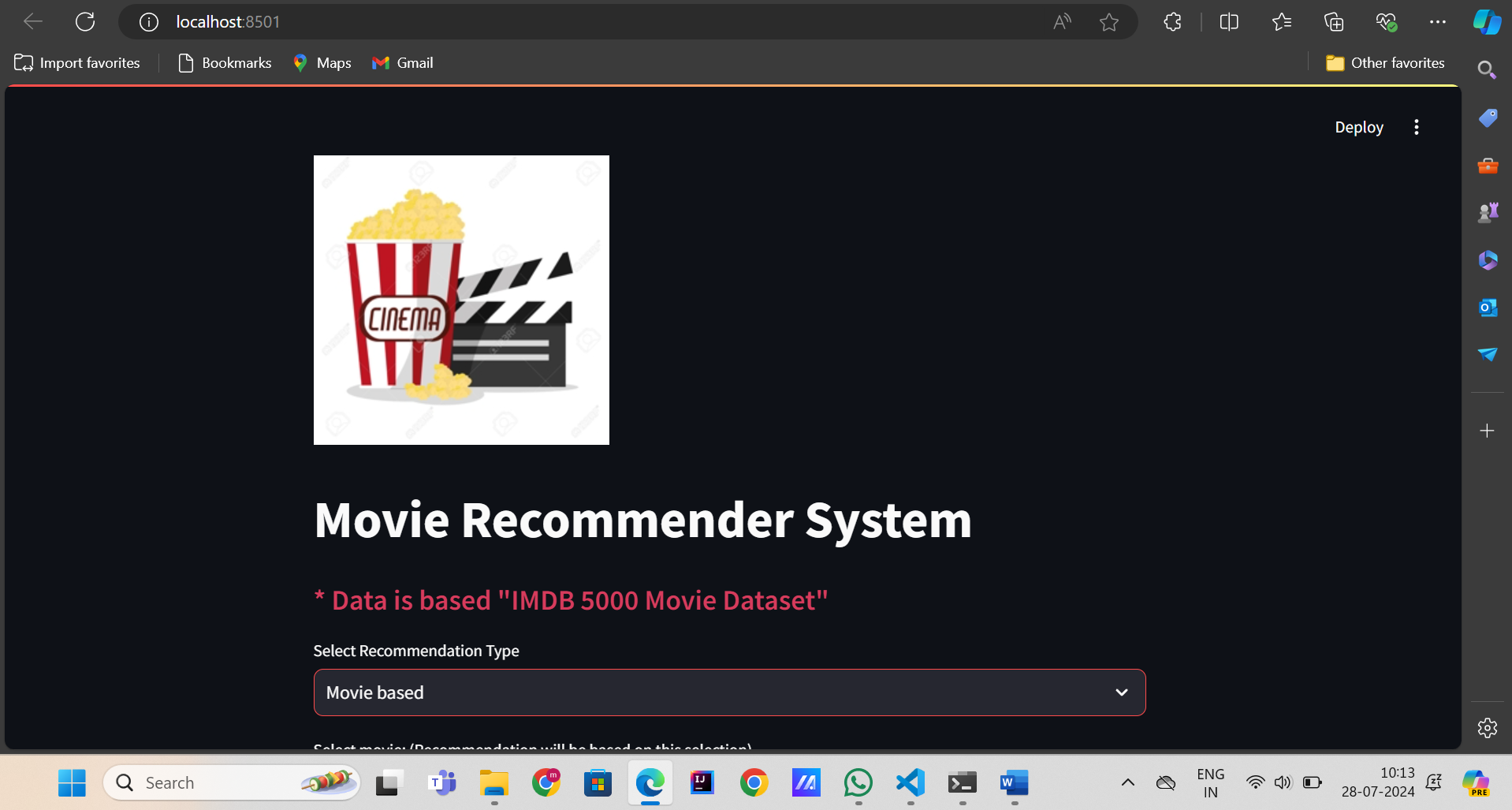
**Movie Database/API:**

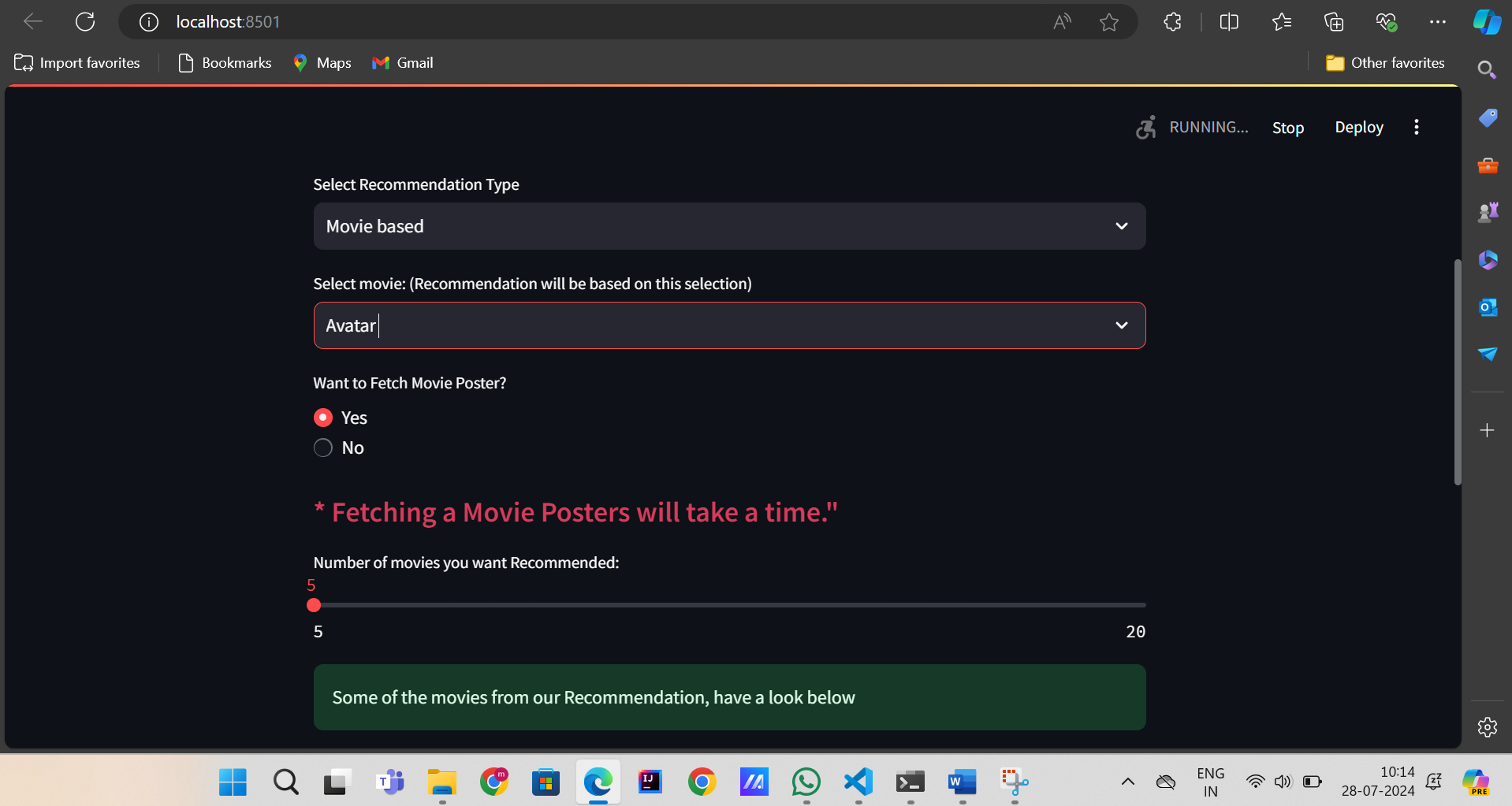
* **API:** TMDb API or a local dataset.
* **Components:**
  + **Movie Data Fetcher:** Retrieves and parses movie data based on the category.
  + **Data Parser:** Extracts relevant movie details from the API response.

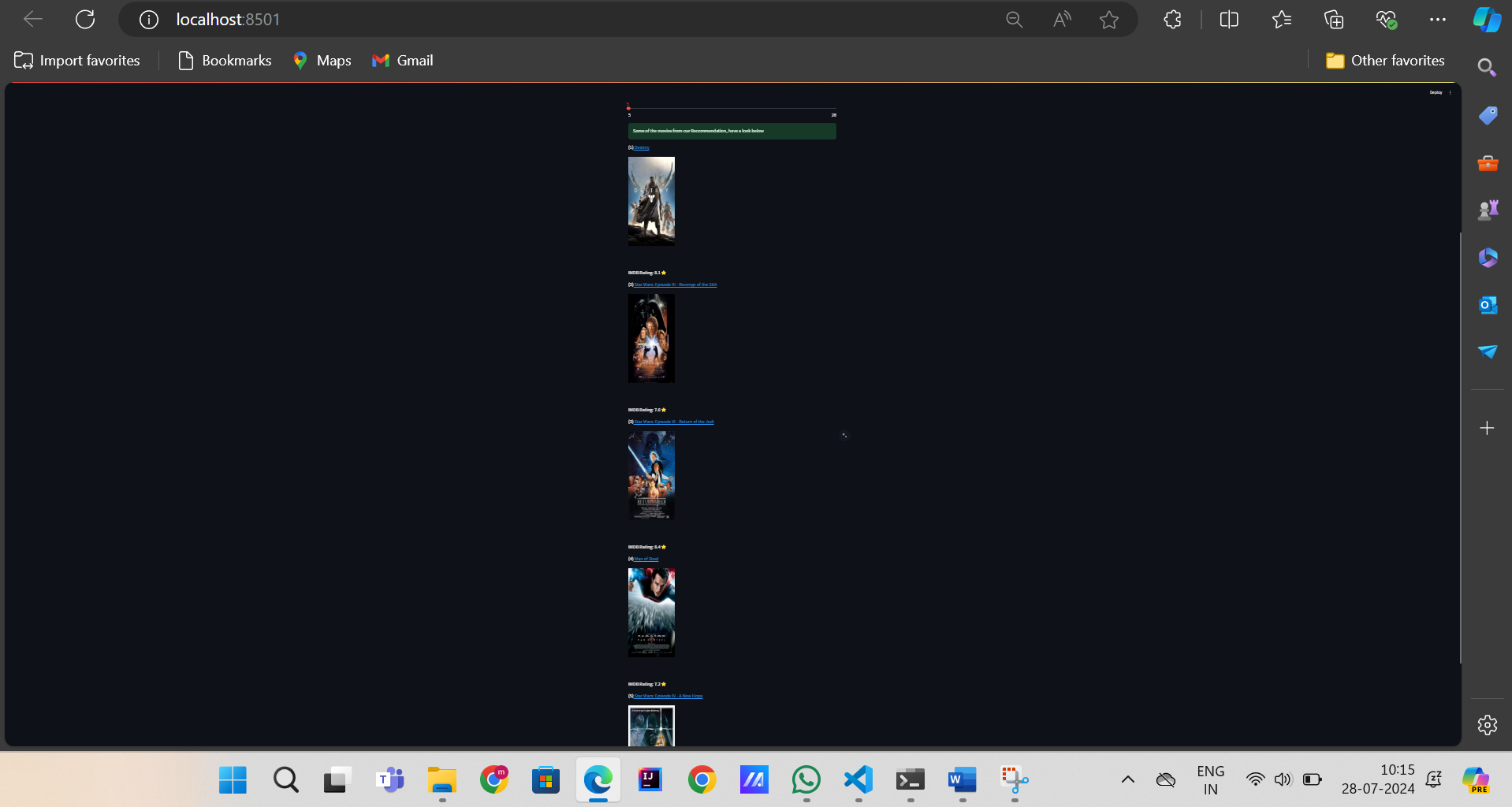
**Data Flow:**

1. **User Interaction:**
   * The user selects a category from the Streamlit interface.
2. **Request Handling:**
   * The Streamlit frontend sends a request to the backend server with the selected category.
3. **Data Fetching:**
   * The backend server sends a request to the Movie Database/API to fetch movie data for the selected category.
4. **Recommendation Generation:**
   * The backend processes the fetched data and applies the recommendation algorithm.
5. **Response Handling:**
   * The backend sends the list of recommended movies back to the Streamlit frontend.
6. **Display Recommendations:**
   * The frontend displays the recommended movies in the Recommendations Display Area.

**Visual Diagram:**

****

****

****