Good morning, everyone.  
My name is Sharath Kumar, and today I’ll be presenting my project: **"Movie Recommendation System Using Python and Streamlit."**  
This project focuses on creating a web-based application that suggests movies to users based on their preferences.

Let’s begin with a quick introduction to what a movie recommendation system is.  
It’s essentially a system designed to **help users discover movies they might enjoy, based on their previous interactions or preferences.( based on genre, cast, ratings etc)**  
The **objective** of my project was to build a **user-friendly application** using Python and Streamlit, making it easy for anyone to get personalized movie recommendations in a visually appealing format.

Now, let’s dive into the methodology.  
For this project, I used the following **technologies**:

* **Python** for logic implementation,
* **Streamlit** for building the web interface,
* **TMDB API** to fetch movie details and posters,
* And libraries like **Pandas, NumPy, and Pickle** for data processing and model storage.

The **recommendation algorithm** is **content-based filtering** that uses **cosine similarity** to identify movies similar to the one a user selects.

Some **key features** of the app include:

* A dropdown menu where users can select a movie,
* Displaying the top 5 recommended movies along with their posters,
* And a “Favorites” option to save movies that users particularly like.

**Step 1:** The user selects a movie from the dropdown menu or types their choice.

**Step 2:** The system retrieves similar movies using pre-computed similarity scores.

**Step 3:** It then displays the recommendations with posters fetched from the TMDB API.

**Step 4:** Users can save their favorite movies, which are stored using Streamlit’s session state feature.

To conclude:  
This project successfully demonstrates a **functional, visually appealing movie recommendation system**. It combines content-based filtering with an intuitive UI built on Streamlit.

For **future enhancements**, I plan to:

* Incorporate **collaborative filtering** for better recommendations based on user behavior,
* Add options for **user reviews and ratings**,
* And eventually **deploy the app** for public use.