# **SUMMER TRAINING/INTERNSHIP**

# **PROJECT REPORT**

# **(Term June-July 2025)**

# **AI-Powered Movie Recommender System Using Hybrid Techniques (Collaborative & Content-Based Filtering)**

# **Submitted by**

# **Sayan Mondal (12314751)**

# **Kadambala Likhith (12314034)**

# **Priyam Saxena (12313674)**

# **Course Code ……………………**

# **Under the Guidance of**

# **(Name of mentor with designation)**

# **School of Computer Science and Engineering**Lovely Professional University - LPU | Jalandhar

# 

**Lovely Professional University, Punjab**

## BONAFIDE CERTIFICATE

Certified that this project report **“AI-POWERED MOVIE RECOMMENDER SYSTEM USING HYBRID TECHNIQUES (COLLABORATIVE & CONTENT-BASED FILTERING)”** is the Bonafide work of **PRIYAM SAXENA, SAYAN MONDAL, and KADAMBALA LIKHITH** who carried out the project work under my supervision.

**SIGNATURE**  
<<Name of the Supervisor>>

**PRIYAM SAXENA**  
**SAYAN MONDAL**  
**KADAMBALA LIKHITH**

**SIGNATURE**

<<Signature of the Head of the Department>>  
SIGNATURE  
<<Name>>  
**HEAD OF THE DEPARTMENT**

<<Signature of the Supervisor>>

# **ACKNOWLEDGEMENT**

I express my sincere gratitude to my mentor \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ for providing guidance and continuous encouragement throughout this project. This project would not have been possible without their support and hard work.

# **TABLE OF CONTENTS**

1. **Introduction**
2. **Training Overview**
3. **Project Details**
4. **Implementation**
5. **Results and Discussion**
6. **Conclusion**

# 

# **CHAPTER 1: INTRODUCTION**

### Overview of Training Domain

Recommender Systems using Machine Learning

### Objective of the Project

To build an AI-based Movie Recommender System using Hybrid Recommendation approaches that integrate both Collaborative Filtering and Content-Based Filtering techniques for personalized movie recommendations.

# **CHAPTER 2: TRAINING OVERVIEW**

### Tools & Technologies Used

* Python, FastAPI, React.js, Tailwind CSS
* Surprise Library, Scikit-learn, Pandas, Matplotlib, Seaborn

### Areas Covered During Training

* Data Collection
* Exploratory Data Analysis (EDA)
* Feature Engineering
* Model Building (SVD++, Content-based Filtering)
* API Development
* Frontend Integration
* Deployment

### Daily/Weekly Work Summary

(You can fill this section based on your weekly progress logs)

# 

# **CHAPTER 3: PROJECT DETAILS**

### Title of the Project

AI-Powered Movie Recommender System Using Hybrid Techniques

### Problem Definition

To solve the problem of movie recommendation using user behavior (ratings, swipes) and movie content (genres, tags).

### Scope and Objectives

* Deliver personalized recommendations
* Enhance user engagement via interactive swipe UI
* Deploy as a full-stack application

### System Requirements

* Python 3.x
* Node.js / React.js
* VSCode / Terminal

### Architecture Diagram

A diagram of a software development process

AI-generated content may be incorrect.

### **Explanation:**

|  |  |
| --- | --- |
| Component | Role |
| React Frontend | Swipe UI (Left = Dislike, Right = Like) |
|  | Fetch Recommendations via API |
|  | Display Posters, Genres, History |
| FastAPI Backend | API Endpoints: /swipe/, /recommend/ |
|  | Handles User Swipe Data (swipes.json) |
|  | Calls recommender\_model.py for predictions |
| Recommendation Model | Hybrid Model (TFIDF + SVD++) |
|  | Returns Top N Recommendations |
| Dataset | CSV Files (ratings, movies, tags) |
|  | TMDb Poster API for Poster URLs |

### Data Flow / UML Diagrams

A screenshot of a computer

AI-generated content may be incorrect.

### **Explanation:**

**Entities:**

* User (React Frontend)
* Backend API (FastAPI)
* Recommendation Model
* Dataset (CSV files: Movies, Ratings, Tags)
* TMDb API (for posters)

**Flow:**

1. **User** sends swipe actions or refresh requests.
2. **React Frontend** sends API requests to **FastAPI Backend**.
3. **Backend** stores/retrieves **swipe history (swipes.json)**.
4. **Backend** fetches recommendations via:
   * **Collaborative Filtering (SVD++)**
   * **Content-Based Filtering (TF-IDF, Cosine)**
5. **Backend** fetches Posters via **TMDb API** if needed.
6. Backend returns recommendations to **Frontend**.
7. User sees updates.

# **CHAPTER 4: IMPLEMENTATION**

### Tools Used

* Backend: FastAPI, Surprise, Scikit-learn
* Frontend: React, Tailwind CSS, Framer Motion

### Methodology

Hybrid Recommendation combining TF-IDF + Cosine Similarity with Collaborative Filtering (SVD++)

### Modules / Screenshots

* Swipe Interface (React)
* Recommendation API (FastAPI)
* User History Tracking (JSON-based)

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

### Code Snippets

A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer program

AI-generated content may be incorrect.

# 

# **CHAPTER 5: RESULTS AND DISCUSSION**

### Output / Report

* Visualizations: User preferences, genre distributions, model performance comparisons.
* Logs and Plots saved under /report\_logs, /report\_images.

### Challenges Faced

* Integration of content and collaborative models
* Ensuring API-UI synchronization

### Learnings

* Hands-on with end-to-end machine learning pipeline
* Real-world deployment using FastAPI and React

# 

# **CHAPTER 6: CONCLUSION**

### Summary

Successfully built and deployed a hybrid movie recommender system with swipe-based UI integrated with personalized AI recommendations.

# 

# **ROLES DISTRIBUTION FOR PRESENTATION**

| **Person** | **Topics to Present** |
| --- | --- |
| **Sayan Mondal** | Model Foundations (Collaborative & Content Filtering), Backend Integration |
| **Kadambala Likhith** | Data Collection, EDA, Feature Engineering, Visualizations, Reporting, Comparisons |
| **Priyam Saxena** | Frontend (React), Swipe UI, APIs, Backend-Frontend Integration, Deployment, User Flow |