ChatGPT

Sprint 11 – One-Page Project Approach Document

Project Title:

Movie Streaming Platform Database

🔍 Problem Statement:

Design and implement a robust, normalized database system to manage users, movies, subscriptions, watch history, and ratings for a streaming service. The system should ensure data integrity, support analytics, and be optimized for performance.

📦 Key Entities & Relationships:

- **User Account**: Manages user credentials and subscription info.
- **Movie**: Stores metadata about each movie.
- **Subscription**: Tracks user payments and plan periods.
- **Watch History**: Logs viewing activity by users.
- **Rating**: Allows users to rate and review movies.

Relationships:

- One user → many subscriptions
- One user ↔ many movies via watch_history & rating
- One movie ↔ many users via watch_history & rating

Normalization:

- All tables are in **3NF** to eliminate redundancy and ensure atomicity.
- Proper use of **foreign keys** to maintain referential integrity.

Constraints Implemented:

- **Primary Keys** (all tables)
- **Foreign Keys** with ON DELETE CASCADE where appropriate
- **NOT NULL**, **UNIQUE**, **CHECK**, **ENUM** used to ensure data quality

Performance & Security Enhancements:

- Indexes on `watch_date`, `genre`, and `rating` columns
- Stored procedures and triggers for automation

- Transaction block for critical updates
- JSON column to manage flexible user preferences

📈 Goal:

Deliver a scalable, query-optimized, and automation-ready SQL solution reflecting real-world streaming platform functionality.

Prepared for SE105: Managing and Querying Database