**Execution document :**

1. **Movie Recommendation**

Softwares used : STS IDE, Postgresql 9.6 or above, Maven latest, Java 8 or above

Step1 : create database epsilon change owner to your database owner

Step2 : Create schema and the tables using the below given queries

Step3 : insert data using below queries

Step4 : Load the application from Git (https://github.com/nishasumesh/test\_repo -> MovieManager)

Step5 : Change the DB credentials in application.yml file

Step6 : Change log file path in logback.xml

Step7 : Build and Execute the project (mvn clean install , mvn spring-boot:run)

-- Database: epsilon

-- DROP DATABASE epsilon;

CREATE DATABASE epsilon

WITH

OWNER = cometadmin

ENCODING = 'UTF8'

LC\_COLLATE = 'C'

LC\_CTYPE = 'C'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1;

-- SCHEMA: movie

-- DROP SCHEMA movie ;

CREATE SCHEMA movie

AUTHORIZATION nishaks;

-- Table: movie.film

-- DROP TABLE movie.film;

CREATE TABLE movie.film

(

filmid text COLLATE pg\_catalog."default" NOT NULL,

filmname text COLLATE pg\_catalog."default",

director text COLLATE pg\_catalog."default",

actors text[] COLLATE pg\_catalog."default",

category text[] COLLATE pg\_catalog."default",

link text COLLATE pg\_catalog."default",

language text COLLATE pg\_catalog."default",

CONSTRAINT film\_pkey PRIMARY KEY (filmid)

)

TABLESPACE pg\_default;

ALTER TABLE movie.film

OWNER to nishaks;

-- Table: movie."user"

-- DROP TABLE movie."user";

CREATE TABLE movie."user"

(

user\_id text COLLATE pg\_catalog."default" NOT NULL,

name text COLLATE pg\_catalog."default",

username text COLLATE pg\_catalog."default",

password text COLLATE pg\_catalog."default",

CONSTRAINT user\_pkey PRIMARY KEY (user\_id)

)

TABLESPACE pg\_default;

ALTER TABLE movie."user"

OWNER to nishaks;

-- Table: movie.userfilm

-- DROP TABLE movie.userfilm;

CREATE TABLE movie.userfilm

(

userfilmid text COLLATE pg\_catalog."default" NOT NULL,

status text COLLATE pg\_catalog."default",

userid text COLLATE pg\_catalog."default",

filmid text COLLATE pg\_catalog."default",

CONSTRAINT userfilm\_pkey PRIMARY KEY (userfilmid),

CONSTRAINT filmid FOREIGN KEY (filmid)

REFERENCES movie.film (filmid) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT userid FOREIGN KEY (userid)

REFERENCES movie."user" (user\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE movie.userfilm

OWNER to nishaks;

insert into movie.film (filmid, filmname, director, actors, category, link, language) values

('flm04', 'The Dark Knight', 'Christopher Nolan' ,'{Christian Bale,Heath Ledger,Aaron Eckhart,Michael Caine}',

'{superhero,crime,adventure}', 'https://www.youtube.com/embed/NMz0X-0Ldjg', 'English');

insert into movie.film (filmid, filmname, director, actors, category, link, language) values

('flm05', 'SpiderMan', 'Sam Raimi', '{Tobey Maguire,Willem Dafoe,Kirsten Dunst}', '{superhero,crime,adventure,action,fantasy}',

'https://www.youtube.com/embed/BoM7G8qdD1w', 'English');

insert into movie.film (filmid, filmname, director, actors, category, link, language) values

('flm03', 'Iron Man', 'Jon Favreau', '{Robert Downey Jr.,Terrence Howard,Gwyneth Paltrow,Leslie Bibb}', '{superhero,comedy}',

'https://www.youtube.com/embed/mAA3XMBFoEw', 'English');

insert into movie.film (filmid, filmname, director, actors, category, link, language) values

('flm02', 'Godzilla', 'Roland Emmerich', '{Matthew Broderick,Jean Reno}', '{fiction,monster,horror,superhero}',

'https://www.youtube.com/embed/O\_of2SQrcd0', 'English');

insert into movie.film (filmid, filmname, director, actors, category, link, language) values

('flm01' ,'Jurassic World', 'Colin Trevorrow', '{Chris Pratt,Jake Johnson,Nick Robinson,Irrfan Khan}', '{adventure,action,science fiction,thriller,horror,superhero}',

'https://www.youtube.com/embed/GHUGMxk6Xpo', 'English');

insert into movie."user" (user\_id, name, username, password) values ('a01', 'Nisha K S', 'nisha', 'nisha');

insert into movie."user" (user\_id, name, username, password) values ('a02', 'Administrator', 'admin', 'admin');

insert into movie.userfilm (userfilmid, status, userid, filmid) values ('uf01', 'watching', 'a01', 'flm02');

insert into movie.userfilm (userfilmid, status, userid, filmid) values ('uf02', 'watched', 'a01', 'flm01');

insert into movie.userfilm (userfilmid, status, userid, filmid) values ('uf03', 'watched', 'a02', 'flm01');

insert into movie.userfilm (userfilmid, status, userid, filmid) values ('uf04', 'watched', 'a02', 'flm01');

Screen shots:

Login url : <http://localhost:8081/login>

Creds : nisha/nisha or admin/admin

1. Login screen

Graphical user interface

Description automatically generated

A picture containing text, different, same, screenshot

Description automatically generated

1. Unsigned user

A picture containing text, electronics, person, display

Description automatically generated

**Project 2: Conference Manager**

1. Create DB

-- Table: movie.conference

-- DROP TABLE movie.conference;

CREATE TABLE movie.conference

(

confid text COLLATE pg\_catalog."default" NOT NULL,

section text COLLATE pg\_catalog."default",

eventname text COLLATE pg\_catalog."default",

eventstart text COLLATE pg\_catalog."default",

eventend text COLLATE pg\_catalog."default",

eventduration text COLLATE pg\_catalog."default",

dayofevent text COLLATE pg\_catalog."default",

eventprformer text COLLATE pg\_catalog."default",

CONSTRAINT conference\_pkey PRIMARY KEY (confid)

)

TABLESPACE pg\_default;

ALTER TABLE movie.conference

OWNER to nishaks;

1. Insert data

INSERT INTO movie.conference(

confid, section, eventname, eventstart, eventend, eventduration, dayofevent, eventprformer)

VALUES ('c01', 'morning', 'lunch break', '2021-01-18 12:00', '2021-01-18 14:30', '138', '1', 'Committee');

INSERT INTO movie.conference(

confid, section, eventname, eventstart, eventend, eventduration, dayofevent, eventprformer)

VALUES ('c02', 'morning', 'lunch break', '2021-01-19 12:00', '2021-01-19 14:30', '138', '2', 'Committee');

INSERT INTO movie.conference(

confid, section, eventname, eventstart, eventend, eventduration, dayofevent, eventprformer)

VALUES ('c03', 'morning', 'lunch break', '2021-01-20 12:00', '2021-01-20 14:30', '138', '3', 'Committee');

INSERT INTO movie.conference(

confid, section, eventname, eventstart, eventend, eventduration, dayofevent, eventprformer)

VALUES ('c04', 'evening', 'networking', '2021-01-18 16:00', '2021-01-18 20:00', '240', '1', 'Committee');

INSERT INTO movie.conference(

confid, section, eventname, eventstart, eventend, eventduration, dayofevent, eventprformer)

VALUES ('c05', 'evening', 'networking', '2021-01-19 16:00', '2021-01-19 20:00', '240', '2', 'Committee');

INSERT INTO movie.conference(

confid, section, eventname, eventstart, eventend, eventduration, dayofevent, eventprformer)

VALUES ('c06', 'evening', 'networking', '2021-01-20 16:00', '2021-01-20 20:00', '240', '3', 'Committee');

1. Data upload can be done directly from postgresql client.
2. Data upload can also be done from front end (Pending)
3. Load the application from Git (https://github.com/nishasumesh/test\_repo/tree/main/ConferenceManager)
4. Change the DB credentials , csv download data file path in application.yml file
5. Change log file path in logback.xml
6. Build and Execute the project (mvn clean install , mvn spring-boot:run)

Screen shots:

1. Login
2. Error message
3. List all existing
4. Add new
5. Added screen
6. CSV file download

Graphical user interface, application

Description automatically generated

Table

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generated

Table

Description automatically generated