# REPORT ON MAD – 1 PORJECT FOR JAN TERM

## Author

Rohit Vishwas Shinde
Roll Number: 21f3002241
Email ID: 21f3002241@ds.study.iitm.ac.in

I am a second-year student at Walchand College of Engineering Sangli with a passion for technology. I am always eager to learn about the latest advancements in technology and how they can be used to solve real-world problems. In addition to my love for technology, I am also interested in astrophysics. I find the mysteries of the universe fascinating and enjoy exploring new ideas and concepts related to the field. In my free time, I like to read books, watch documentaries, and participate in online forums related to technology and astrophysics. I am excited to continue my studies and pursue a career that combines my interests in both technology and astrophysics.

## Description

Based on my understanding, the project involves creating a movie booking app that allows users to register, select and search movies, theatres, choose seats, book tickets, write reviews, give feedback. The app needs to be user-friendly and responsive, and should provide a seamless experience for users. This report will provide details on the technologies used, the database schema design, the API design, and the architecture and features of the app.

## Technologies used

For this project, I used the Flask web framework, along with various Flask extensions such as Flask-SQLAlchemy, Flask-WTF, Flask-Migrate, Flask-Login, and Flask. We also used HTML, CSS, and jinja2 templates for the frontend.

Bootstrap is a popular open-source CSS framework that provides a collection of pre-built UI components and styles for building responsive web pages and applications.

REST (Representational State Transfer) is a popular architectural style for building web APIs. RESTful APIs provide a standard way of accessing and manipulating resources over HTTP. By using these technologies, a movie booking app can be developed with a robust and scalable architecture. Flask can be used to handle HTTP requests and responses, SQLAlchemy can be used to interact with the database.

## DB Schema Design

We designed the database schema to have thirteen tables: Users, Movies, Cast, Place, Feedback, Movie_cast, Theatre, Review, Screen, Show, Tier, Seats, and Tickets. The Users table has columns for user ID, username, firstname, lastname, password, and age. The Movies table has columns for movie ID, movie name, info, poster link, and rating. The Cast table has columns for Cast ID, firstname, lastname, type, and photo. Place table has place Id, city, state and pin columns. Feedback has username and feedback along with Id. Movie_cast table defines many to many relationship between movies and cast. Theatre table has columns for theatre ID, name, photo and place_id to establish relationship between Theatre and Place tables. Review has content, user_id and movie_id along with review ID. Screen table has ID, number and theatre_id columns. Show has show_id, movie_id, time and screen_id. Tier table has ID, number column which ranges from 1 to 3, defining different kinds of seats available, screen_id and price of seat. Price will be based on tier and not on movie. Seats table ID, number of rows, number of Columns, status (booked and available) and tier ID. Finally, Tickets table has ID, booked at, user_id, seat_id and show_id. We added constraints to ensure data integrity, such as unique usernames and email addresses, and foreign key constraints.

## API Design

We created APIs for user registration, user login, movie listing, seat selection, and theatre search. We used Flask-RESTful to define the endpoints and handle requests and responses. The YAML file for the API is submitted separately.

## Architecture and Features

The project is organized using the Model-View-Controller (MVC) architecture. The controllers and models are in the app.py file, and the templates are in the 'templates' directory. We implemented default features such as user registration, user login, movie listing, seat selection, and ticket booking. Admin can add, remove, update and delete movies, cast, theatres, screens, tiers, seats and cast. He can also associate cast with movies they casted. We also added additional features such as movie search, cast search, filtering movies of particular cast, top-10 trending, and update profile for users. Users can send feedback for platform to admin and write reviews for movies which will be displayed on movies tab. Overall, the project provides a user-friendly and seamless experience for booking movies online.

## Video

Here's a link to a video demonstrating the movie booking app project:
https://drive.google.com/file/d/1TkkKkmq6tWgOxEICub30aGTD2otz_YHi/view?usp=share_link