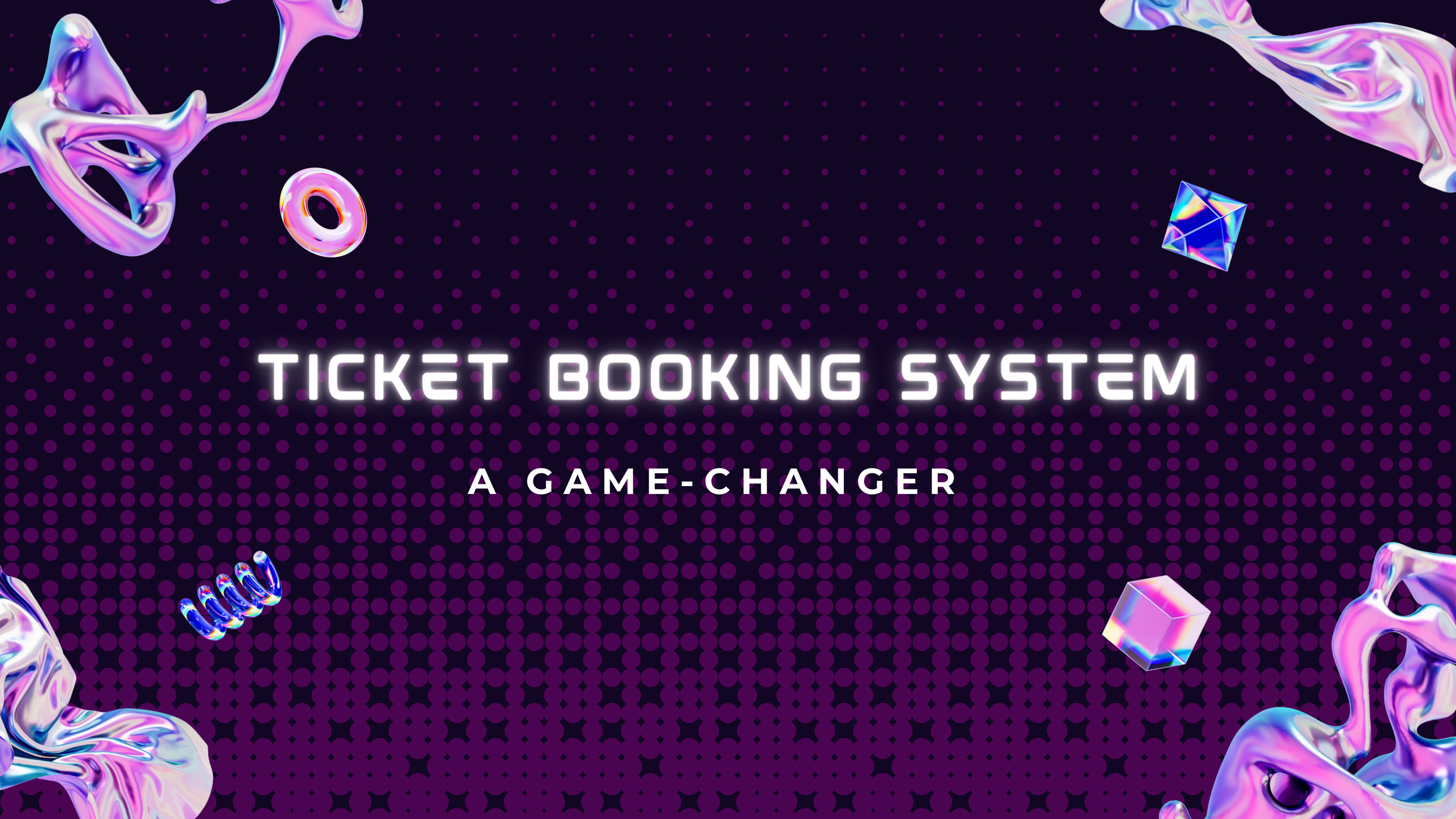

TICKET BOOKING SYSTEM
A GAME-CHANGER

GROUP 4 (ROLL NUMBERS 31-40)

TEAM LEADER: ABHISEK BISWAS 22CS8036

TEAM MEMBERS:

ARJYA DUTTA	22CS8031
SWARUP PAL	22CS8032
SHUBHAJIT SHAW	22CS8034
SHIVAM KUMAR	22CS8035
SUNRIT SINGHA	22CS8037
KUNDURTHI ASHUTOSHRAM	22CS8038
ABHISHEK DOVARI	22CS8039
ARINDAM KUMAR	22CS8040

PROBLEM STATEMENT

To develop a TicketMaster as a leading ticketing platform where event-executors can sell tickets for concerts, theatres, and sporting events and audience/users can purchase tickets. Probable features of the system may include a visual map of seat matrix, text-based user interface, user registration and authentication, ticket booking, etc

We have chosen to build Ticketmaster for a movie theatre.

OS FEATURES

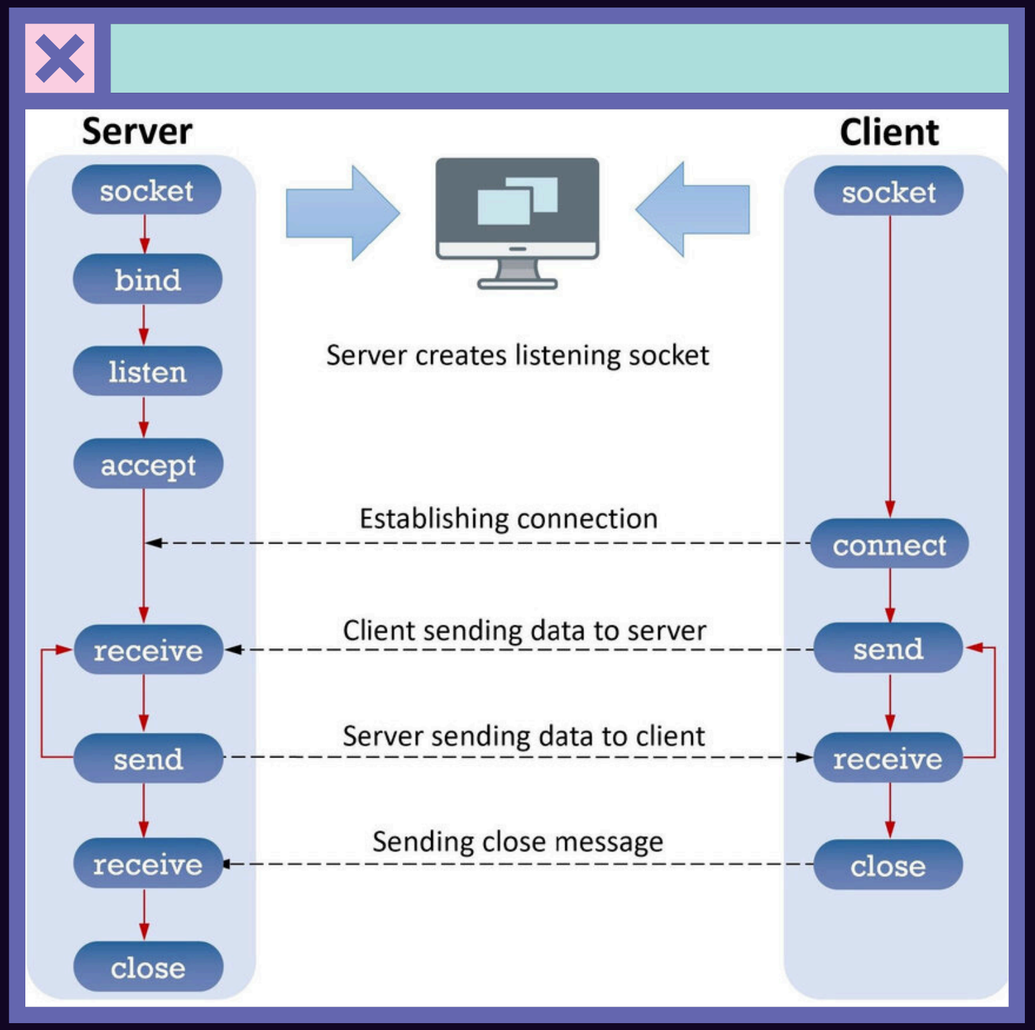
The operating system concepts utilized in this project have been listed below.

- Multi-threading:
  - The project implements threads using pthread library to handle various client requests such as booking movie tickets, and checking previous bookings.
  - Usage of threads promotes concurrency and finer control, thereby streamlining the booking process and enabling us to serve multiple users simultaneously.

- Semaphores:
- Our code makes use of semaphores to control access to critical resources, which in this case, are the seats inside the movie theatre.
  - It ensures that multiple users, who may be online at the same time, are unable to book the same seat. Thus, the problem of multiple or duplicate bookings is avoided using semaphores.

- Socket programming:
- While not strictly part of OS, socket programming is central to our project. We've made use of sockets to set up communication between server and clients, to simulate working of a real-world movie theatre.
  - One machine (or terminal) acts as the server, logging all transactions and changes made to database. Other machines (or terminals) act as users to book movie tickets, or as admins to add or delete movies from the theatre's catalogue.

DIAGRAM OF SOCKET PROGRAMMING



PROJECT STRUCTURE

The high-level structure of our project is described by:

- Source Code Files:

  - server.c : For keeping track of all transactions made by users or admins. It takes care of server side code
  - user.c: For booking movie tickets and displaying a user's previous bookings. Contains various helper functions.
  - admin.c: For updating list of movies and users. Contains various helper functions.
  - runner.c: For enabling user registration, user/admin login, and to launch the project.

- Database Files:

  - movies.txt: For storing names, ratings, audience score, genres, timings of movies
  - seat_matrix.txt: For storing booked and empty seats for each show
  - users.txt: For storing ID, username, password of users and admin
  - bookings.txt: For storing logs of every booking made by users in the system

FUNCTIONALITY OF USER CODE

The user.c file controls all the user (client) activities. It has the following major functions:

- book_movie_seats():

  - Lets user apply filters of their choice - movie genre, rating, audience score- and fetches the list of matching results.
  - Prompts them to choose one of the movie shows, select number of seats as well as the seat numbers
  - Displays price per seat, which changes based on number of vacant seats currently available
  - Allows user a fixed time period to enter their ID to confirm the booking. Inability to do so leads to a timeout failure.
  - Updates bookings.txt and seat_matrix.txt files on successful completion of seat bookings

- display_bookings():

  - Prompts user to enter their ID to view their previous bookings
  - Fetches a list of bookings from the bookings.txt file corresponding to user's ID
  - Displays the bookings on user's terminal

DYNAMIC PRICING FORMULA

We used a simple but effective formula to increment price per seat based on the percentage of seats already booked.

Base price per seat: Rs. 100

- For less than 25% seats booked, dynamic price is equal to base price
- For more than 25& but less than 50% seats booked, dynamic price is 110% of base price
- For more than 50% but less than 75% seats booked, dynamic price is 120% of base price
- For more than 75% seats booked, dynamic price is 130% of base price

FUNCTIONALITY OF ADMIN CODE

The admin.c file controls all the admin activities. Admin is treated as a special client in the system, with certain privileges. The major functions of admin.c include:

- update_movies():

  - Prompts admin to choose whether they wish to add or delete a movie show from the theatre's catalogue
  - For adding a new entry, admin is directed to enter movie name, timing, rating, etc
  - After all details have been entered, the function adds a new entry to movies.txt, and creates a blank seat matrix in the seat_matrix.txt file
  - For deleting an entry, admin is prompted to enter relevant details such as movie name and timing
  - Corresponding to admin input, the entry for the movie show is removed from movies.txt and seat_matrix.txt files.

- update_users():

  - Prompts admin to choose whether they wish to create an account for a new user, or delete an existing user account from the system
  - To add a new user, the admin allots an ID, username and password to the user.
  - Subsequently, the function adds the new entry to users.txt file. The new user account may now be utilized to book movie tickets.
  - To delete an existing user account, the admin enters the user ID. If such an ID exists in the system, that user account is removed from users.txt file.

SUMMARY OF PROJECT

The project achieves the basic functionalities outlined in the problem statement, which include user registration, booking seats, generating movie tickets, authentication and a simple, clean text-based UI.

Innovations/Additional features introduced in our project are:

- Dynamic pricing, to better reflect real-world cost vs demand situation
- Admin control, enabling the admin to modify movie catalogue and user database
- Movie filters, to customize the booking process as per each unique user's needs

MEMBER ROLES

Coding: Abhisek, Swarup, Sunrit, Shivam

Testing: Subhajeet, Dovari, Ashutosh

Presentation and supervision: Abhisek

Research and PPT making: Sunrit, Arindam, Arjya, Dovari

BRAINSTORMING: EVERYONE!