PROJECT

ON

DATABASE MANAGEMENT SYSTEM OF GAMING PLATFORM

submitted in partial fulfill of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE SPECIALIZED IN ARTIFICIAL INTELLIGENCE

AND

MACHINE LEARNING

By

K.BHAVANA 21AG1A6625

V.YAMINI 21AG1A6660

M.SRUTHI 21AG1A6635

Under the guidance of

Mr.krishnanandh

Professor



DEPARTMENT OF COMPUTER SCIENCE SPECIALIZED IN ARTIFICIAL INTELLIGENCE

ACE Engineering College UGC AUTONOMOUS INSTITUTION

Accredited with NBA & NAAC „A‟ Grade (Approved by AICTE and Affiliated to Jawaharlal Nehru Technological University, Hyderabad) Ankushapur (V), Ghatkesar (M), Medchal– Malkajgiri Dist. 501 301 T.S. India. (2023-24)

**ACKNOWLEDGEMENT**

We would like to thank our guide, **Mr.krishnanandh Professor** for providing valuable guidance throughout this project.

We would like to express our gratitude towards **Mr.T.Srinivas Rao**, Associate Professor & Head of the Department for Training and Placement Department ACE Engineering College, Ghatkesar, Hyderabad for allowing us to do a project.

We would like to thank Dr. BL RAJU Sir, Principal, of ACE Engineering College, Ghatkesar, Hyderabad for giving an opportunity to carry out the project.

We would like to express our sincere thanks to the Management of ACE Engineering College for their kind cooperation and timely help during the course of a project.

We would like to express our sincere thanks to all Staff Members of the Department of ACE Engineering College for their kind cooperation and timely help during the course of our project.

We would also express our deep sense of gratitude to our beloved parents for their concern and anxiety with regards to the successful completion of this project.

K.BHAVANA 21AG1A6625

V.YAMINI 21AG1A6660

M.SRUTHI 21AG1A6635

**ABSTRACT**

*Gaming Platform Project with MySQL In this project, we aim to create a robust gaming platform using MySQL as the database management system.* *The gaming platform encompasses a comprehensive suite of features, including user authentication, game catalog management, player profiles, social interactions, and analytics. Our focus will be on designing and implementing the necessary database schema to support various aspects of the gaming ecosystem. Let’s break down the key components*:

*Entity Relationship Diagram (ERD):*

*We’ll start by creating an ERD that visually represents the relationships between different entities in our database. The ERD will include tables for games, genres, publishers, platforms, and sales regions.*

*The ERD will help us understand how data flows and how different entities are connected.*

*Database Tables and Columns:*

*Genre Table:*

*Contains a list of game genres (e.g., Cartoony, thriller,casual,stratergy,Racing).*

*Game Table:*

*Stores information about individual games.*

*Includes columns for the game title and a genre ID (linked to the genre table).*

*Publisher Table:*

*Represents video game publishers (companies that publish games).*

*By implementing this gaming platform project, we’ll gain practical experience in database design, SQL querying, and handling realistic data. Our goal is to create a functional and efficient system that supports the gaming industry.*

**Introduction**

In the contemporary digital landscape, gaming has emerged not only as a form of entertainment but as a thriving industry with a global reach. With millions of gamers engaging in various platforms and genres, the demand for immersive and interactive gaming experiences has never been higher. To meet this demand, the development of robust gaming platforms that offer seamless gameplay, social interaction, and personalized experiences has become paramount.

This introduction outlines the significance and scope of a gaming platform project integrated with MySQL, a leading relational database management system (DBMS). MySQL's robust features, scalability, and performance make it an ideal choice for managing the vast and dynamic data associated with gaming platforms.

The integration of MySQL into the gaming platform project not only ensures the seamless operation of critical functionalities but also enables the implementation of advanced features such as personalized recommendations, analytics-driven insights, and secure payment processing.

This introduction sets the stage for a comprehensive exploration of the gaming platform project, highlighting the pivotal role of MySQL in driving innovation, scalability, and data management excellence within the gaming industry. Through strategic integration and meticulous implementation, the project aims to deliver a cutting-edge gaming platform that caters to the evolving needs and preferences of gamers worldwide

**Functional requirements**

for a gaming platform project using a Database Management System (DBMS) like MySQL would include:

**User Authentication and Account Management:**

Users should be able to register, log in, and manage their accounts.

Password hashing and encryption for secure authentication.

Ability to recover forgotten passwords through email verification or security questions.

**Game Catalog Management:**

Ability to browse and search for available games.

Categorization of games by genre, popularity, release date, etc.

Detailed game descriptions, including screenshots, trailers, and user reviews.

Support for adding new games to the catalog and updating existing ones.

**Player Profiles and Progress Tracking:**

Each user should have a profile to track their gaming activity and achievements.

Ability to view game progress, including completed levels, achievements unlocked, and high

scores.

Personalization options for player profiles, such as avatars and bios.

**Social Features and Community Interaction:**

Support for social features like friend lists, messaging, and multiplayer gaming.

Ability to connect with friends, invite them to games, and join multiplayer sessions.

Community forums or chat rooms for discussions, tips, and sharing gaming experiences.

Notification system for friend requests, messages, and game invitations.

**Gameplay Statistics and Analytics:**

Tracking and analysis of gameplay statistics, such as playtime, win/loss ratios, and achievements.

Generation of personalized recommendations based on gaming preferences and behavior.

**Payment and Transactions:**

Integration with payment gateways for purchasing games, in-game items, or premium memberships.

Support for various payment methods, such as credit/debit cards, PayPal, and digital wallets.

**Content Management and Moderation:**

Management of user-generated content, including reviews, comments, and user-created levels.

**Admin Panel and Management Tools:**

Administrative dashboard for managing users, games, and content.

User management tools for account verification, banning, and moderation.

Analytics dashboard for monitoring platform performance, user engagement, and revenue metrics.

Content management tools for adding, updating, and removing games from the catalog.

These functional requirements provide a foundation for developing a robust gaming platform that offers a rich and engaging experience for players while facilitating efficient data management using a DBMS like MySQL.

**ENTITY RELATIONSHIP DIAGRAM FOR GAMING PLATFORM**

**Attributes**

Player:

* player\_id (Primary Key)
* username
* email
* password

Game:

* game\_id (Primary Key)
* title
* genre
* developer
* publisher

Score:

* score\_id (Primary Key)
* player\_id (Foreign Key)
* game\_id (Foreign Key)
* points

Achievement:

* achievement\_id (Primary Key)
* name
* description

Inventory:

* inventory\_id (Primary Key)
* player\_id (Foreign Key)
* item\_name
* description

Tournament:

* tournament\_id (Primary Key)
* title
* organizer
* date\_time

Payment:

* payment\_id (Primary Key)
* player\_id (Foreign Key)
* transaction\_id

Subscription:

* subscription\_id (Primary Key)
* player\_id (Foreign Key)
* plan\_name
* duration
* price

Chat:

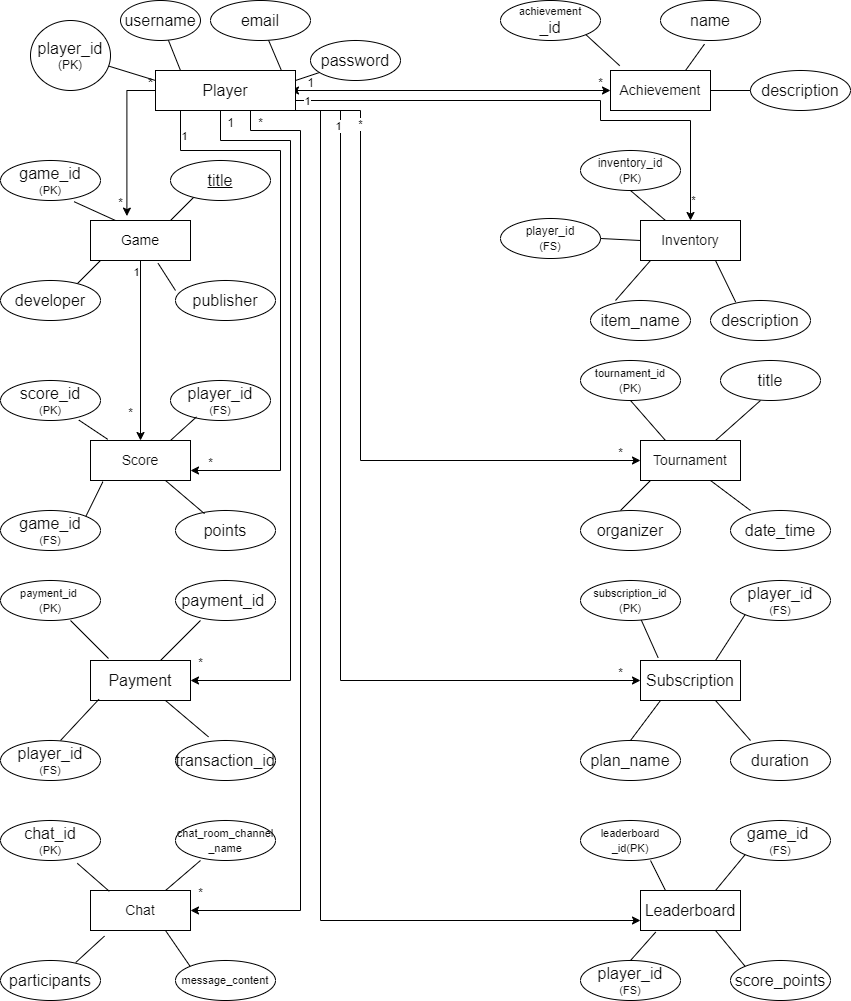
* chat\_id (Primary Key)
* chat\_room\_channel\_name
* participants
* message\_content

Leaderboard:

* leaderboard\_id (Primary Key)
* game\_id (Foreign Key)
* player\_id (Foreign Key)
* score\_points

**Description of the entities and their relationships:**

* One player can play multiple games(one-to-many).
* One player can have multiple scores(one-to-many).
* One player can achieve multiple achievements(one-to-many).
* One player can have multiple items in their inventory(one-to-many).
* One player can participate in multiple tournaments(many-to-many).
* One player can make multiple payments(one-to-many).
* One player can have multiple subscriptions.(one-to-many).
* One player can participate in multiple chats(many-to-many).
* One game can have multiple scores(one-to-many).



**Queries to create the tables in the Database:**

* CREATE TABLE Games (

Game\_ID INT PRIMARY KEY,

Title VARCHAR(50) NOT NULL,

Genre VARCHAR(20) NOT NULL,

Developer VARCHAR(20) NOT NULL,

Publisher VARCHAR(20) NOT NULL

);

* CREATE TABLE Score (

Score\_id INT PRIMARY KEY,

PlayerID int not null,

foreign key(PlayerID) references

Players(PlayerID),

Game\_ID int,

foreign key(Game\_ID) references

Games(Game\_ID),

points int not null);

* CREATE TABLE Achievement(

AchievementID INT PRIMARY KEY,

Name VARCHAR(50) NOT NULL,

Description VARCHAR(50) NOT NULL

);

* CREATE TABLE Inventory(

InventoryID INT PRIMARY KEY,

PlayerID INT, foreign key(UserID) references

Players(PlayerID),

Itemname VARCHAR(50) NOT NULL,

Description VARCHAR(100) NOT NULL

);

* CREATE TABLE Tournaments(

TournamentID INT PRIMARY KEY,

Title VARCHAR(50) NOT NULL,

Start\_date datetime,end\_date datetime NOT NULL,

Organizer VARCHAR(50) NOT NULL

);

* CREATE TABLE payments (

paymentID INT PRIMARY KEY,

PlayerID INT, foreign key(PlayerID) references

Players(PlayerID),

transactionID INT(250) NOT NULL

);

* CREATE TABLE Subscription (

Subscription\_id int primary key,

PlayerID int,

foreign key(PlayerID) references Players(PlayerID),

plan\_name varchar(30),

duration varchar(30),

price int not null);

* CREATE TABLE Chat (

chatID INT PRIMARY KEY,

channel\_name VARCHAR(50),

participants VARCHAR(100),

message\_content TEXT);

* CREATE TABLE Leaderboard (

Leaderboard\_ID INT PRIMARY KEY,

PlayerID int not null,

foreign key(PlayerID) references

Players(PlayerID),

Game\_ID int,

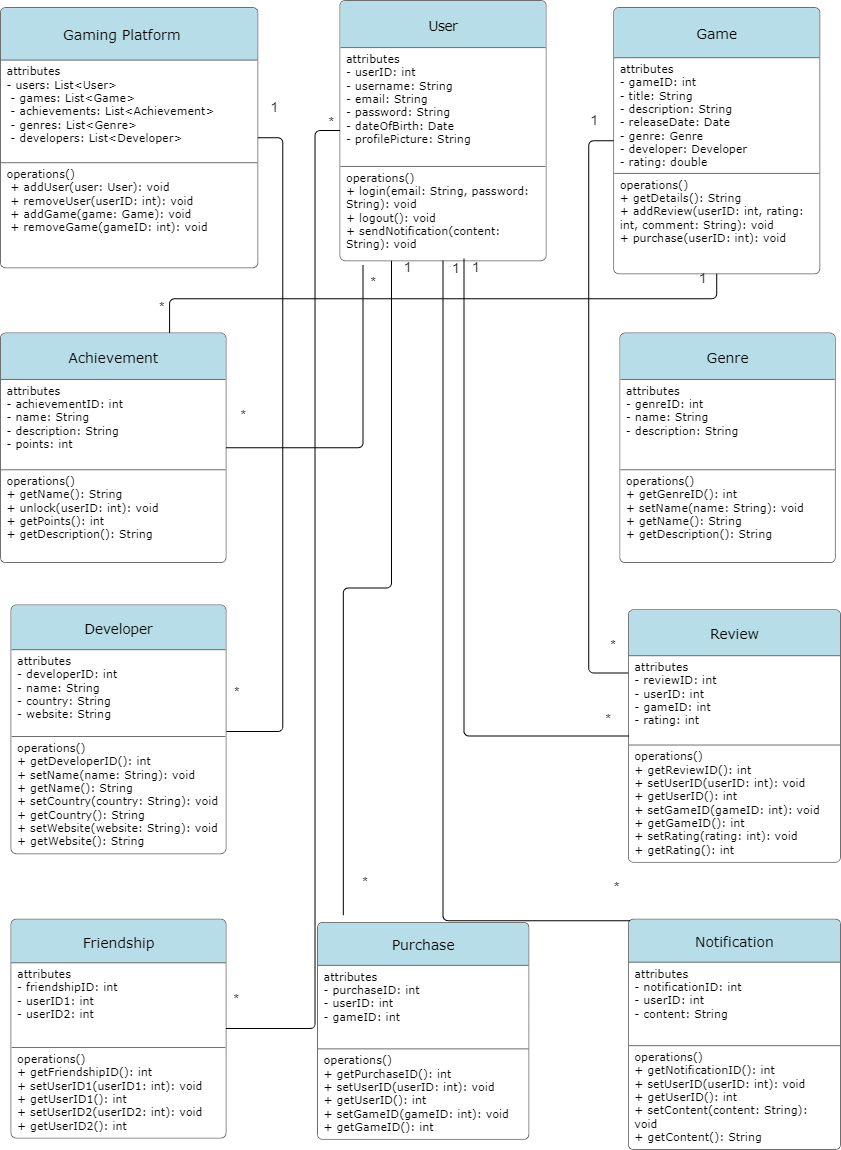
foreign key(Game\_ID) references

Games(Game\_ID),

Score\_points INT

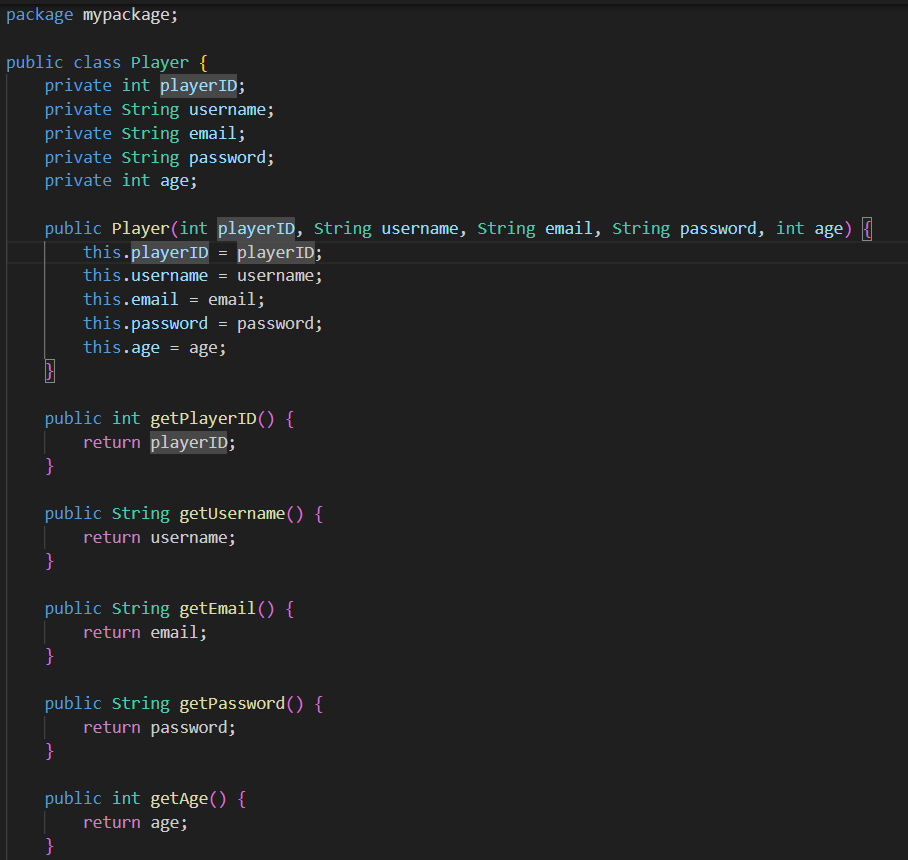
);

**UML DIAGRAM**



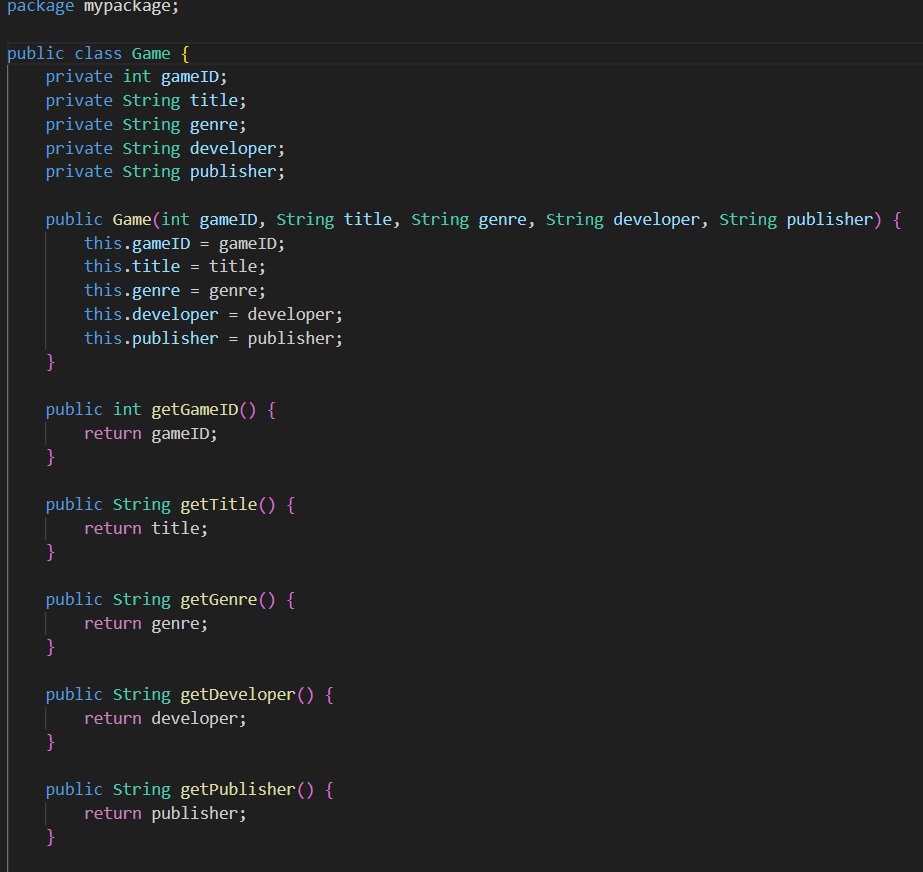
All Classes Java file codes :

**-player.java:**

* 

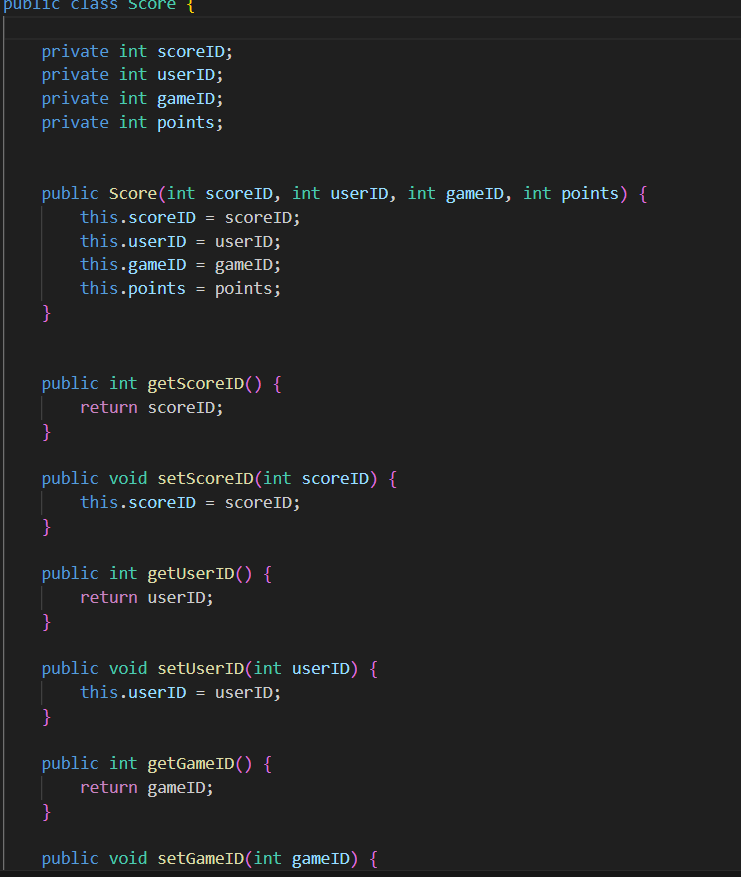
****

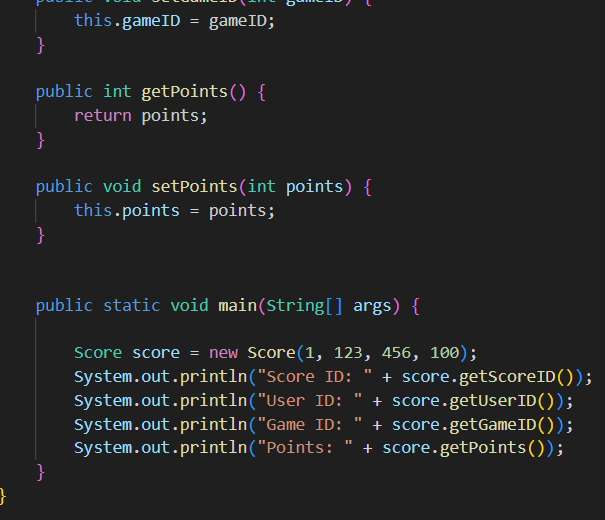
**-Games.java:**

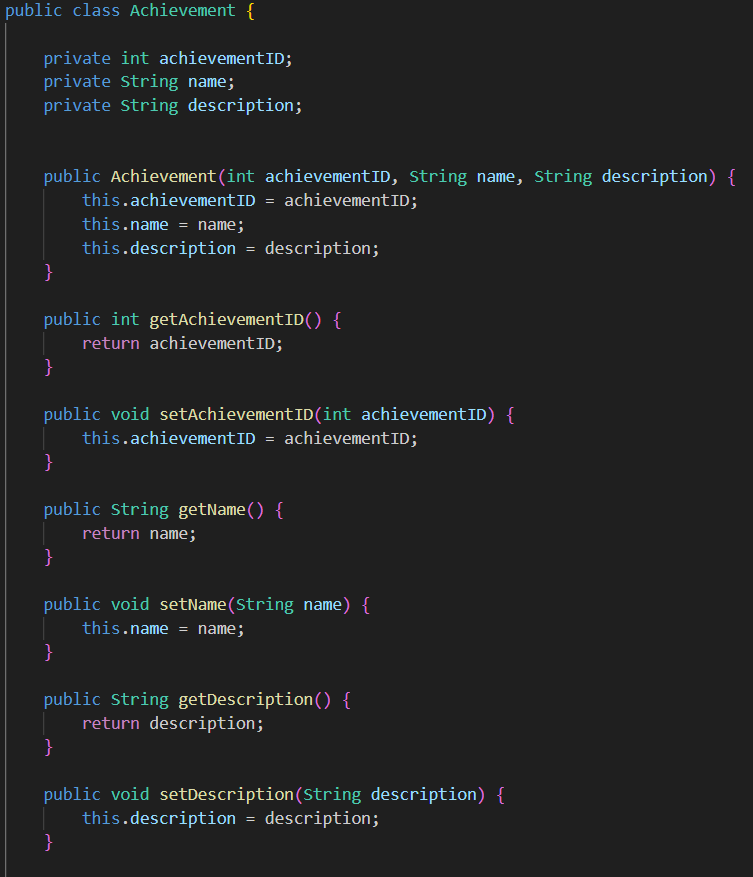


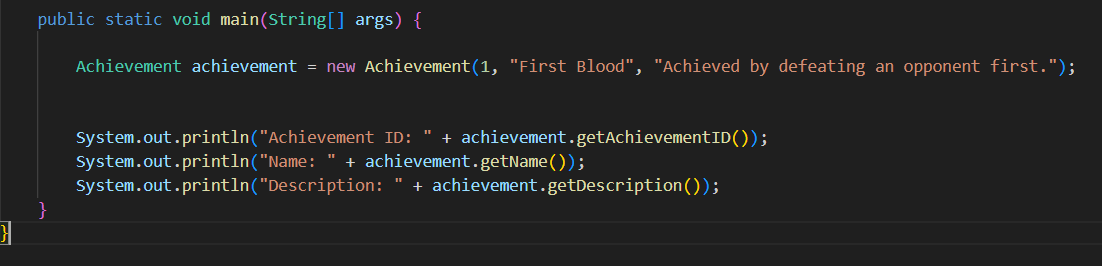


**-Score.java:**

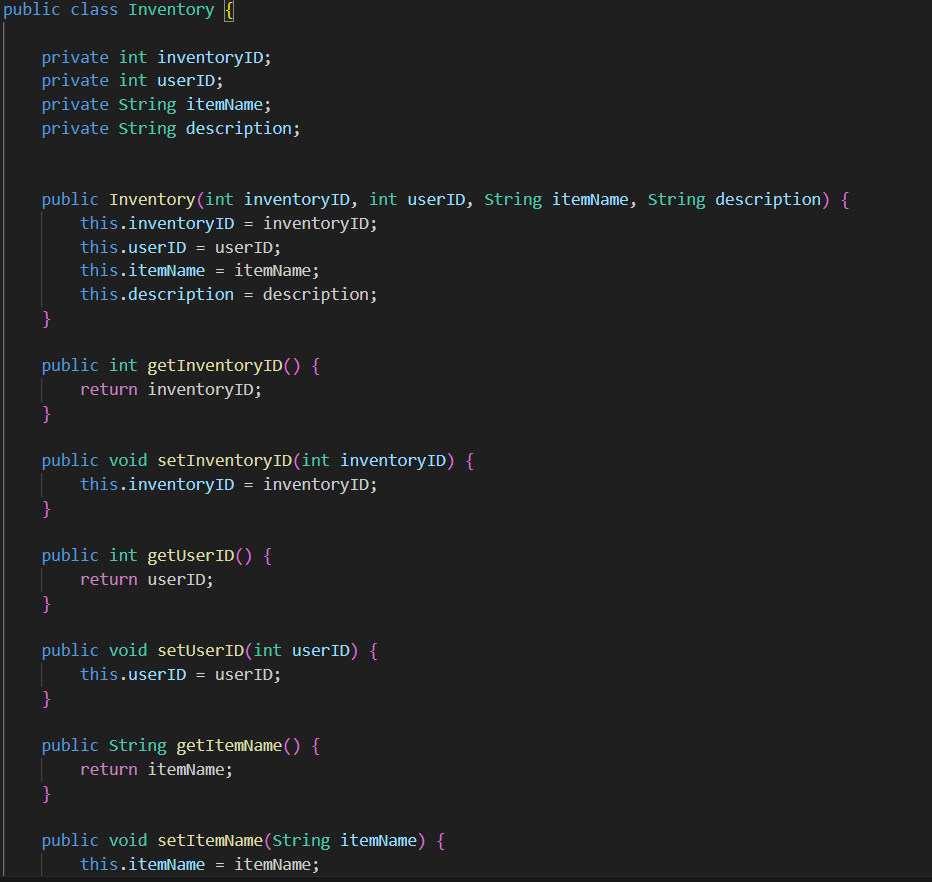




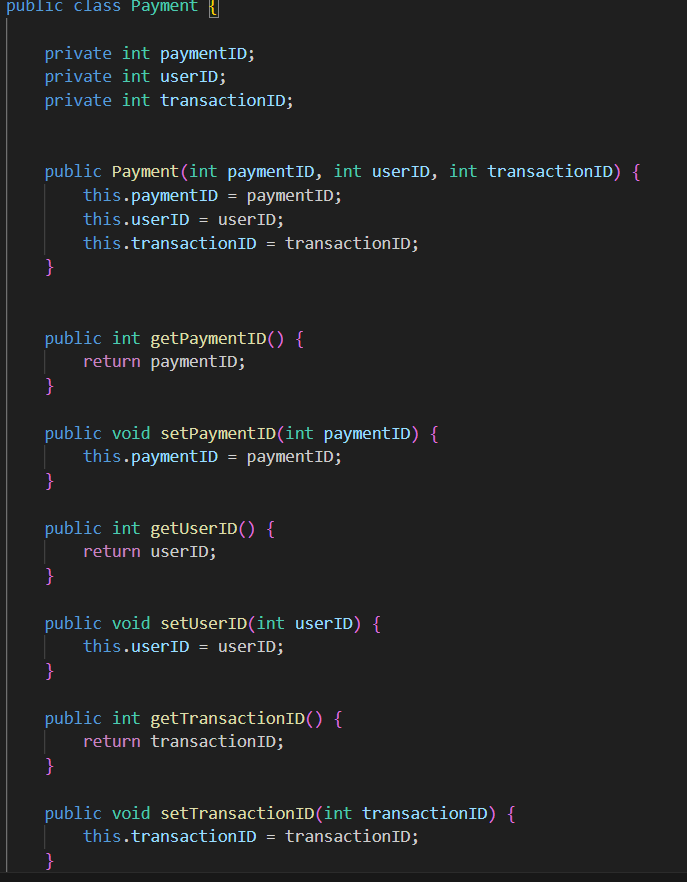
**-Achievement.java:** ****

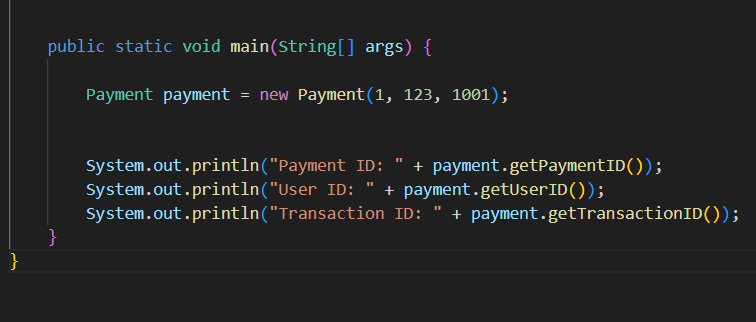
****

**-Inventory.java:**

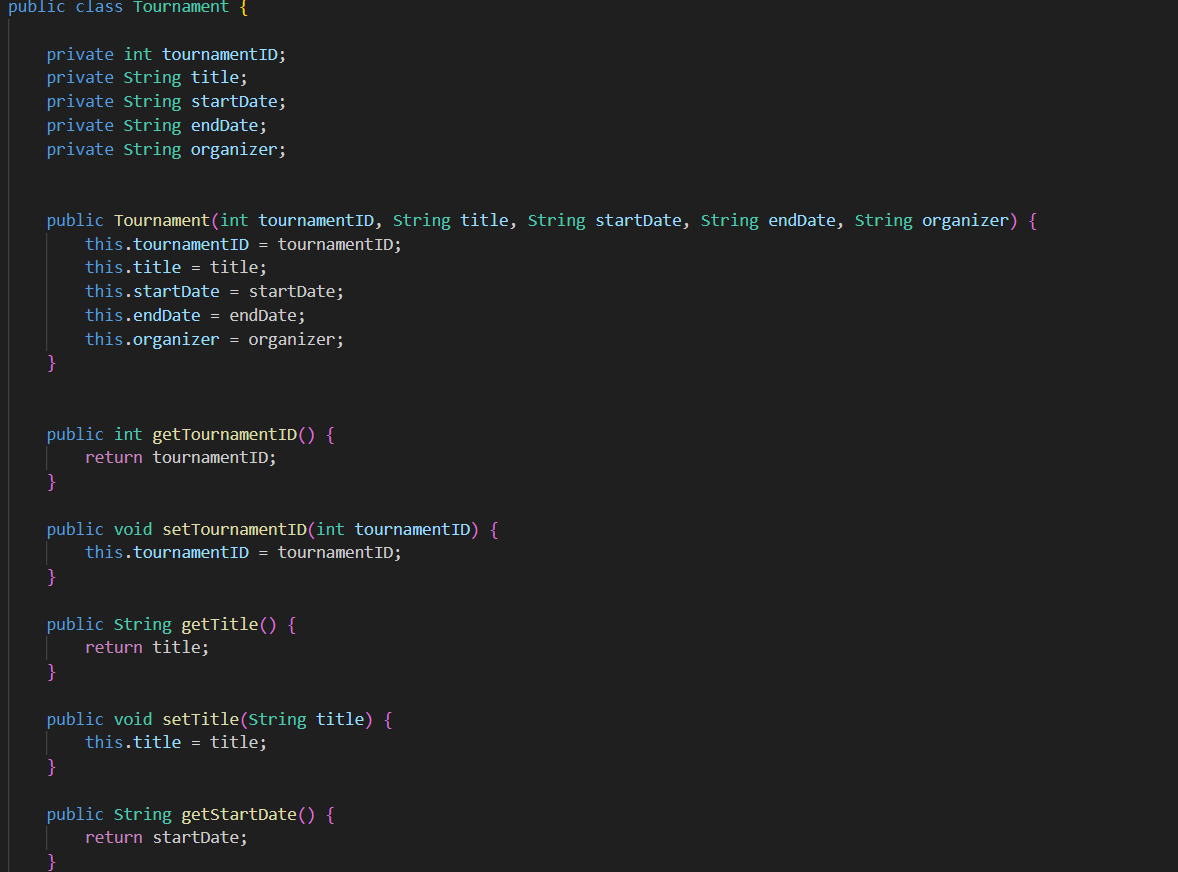
****

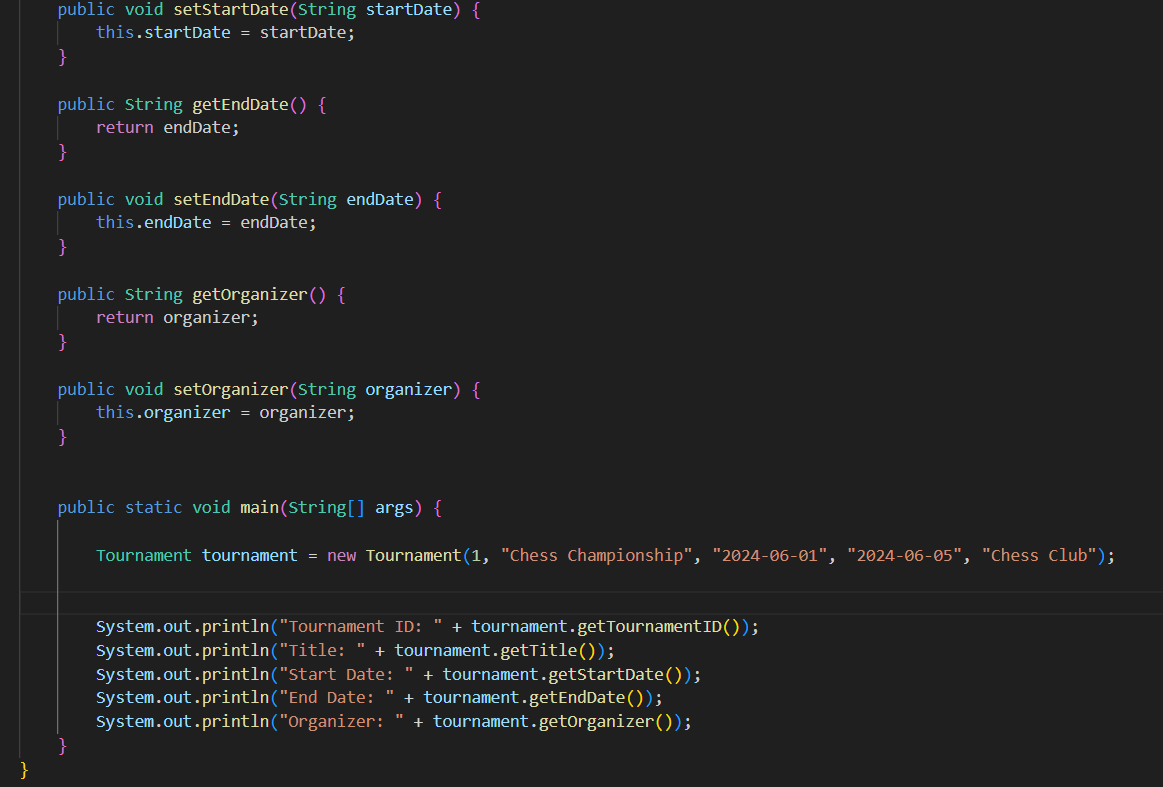
****

**-Payments.java:** ****

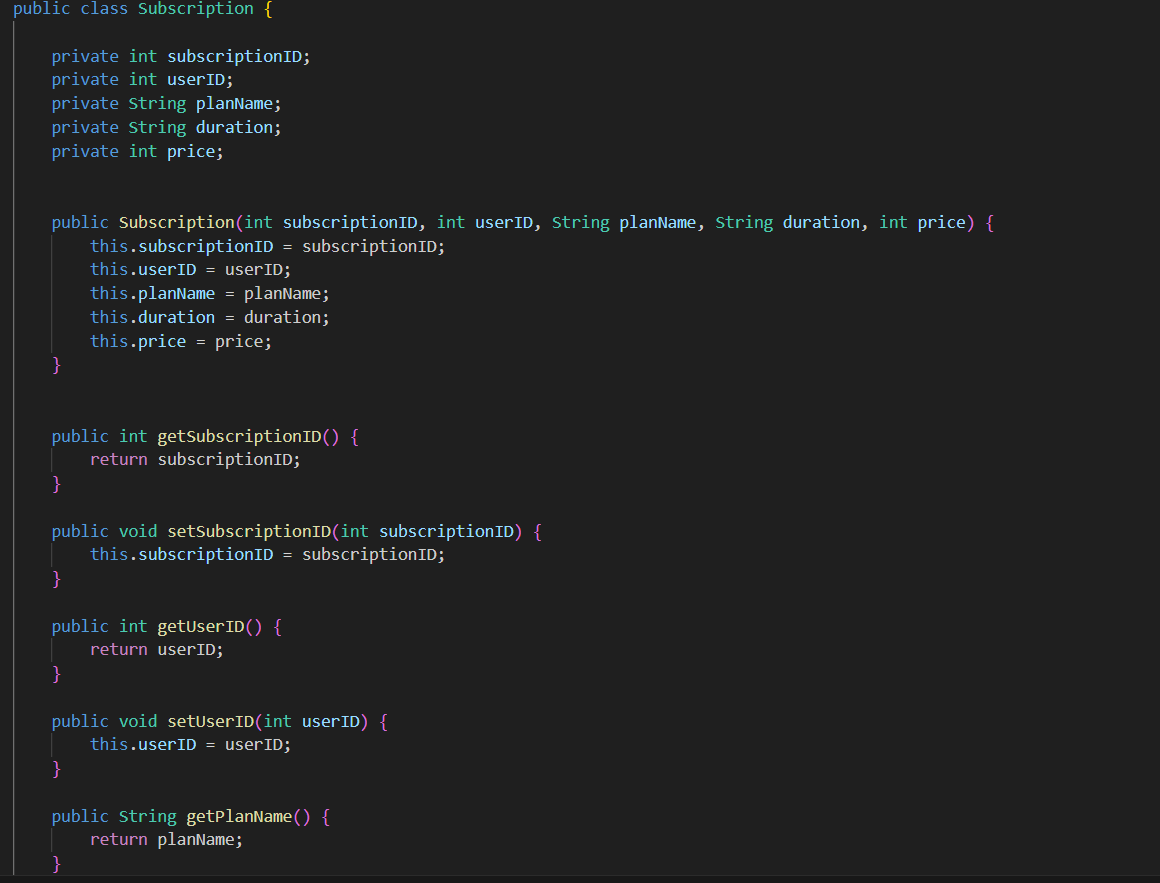
****

**- Tournament.java: -**

****

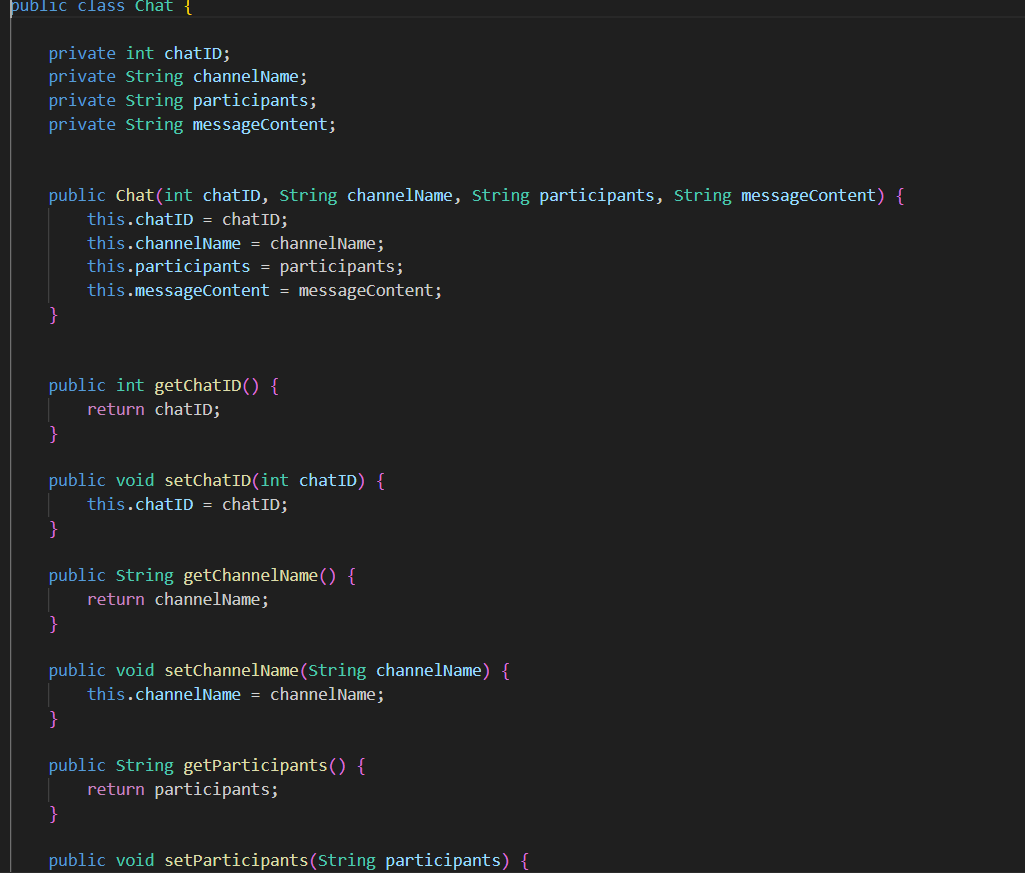
****

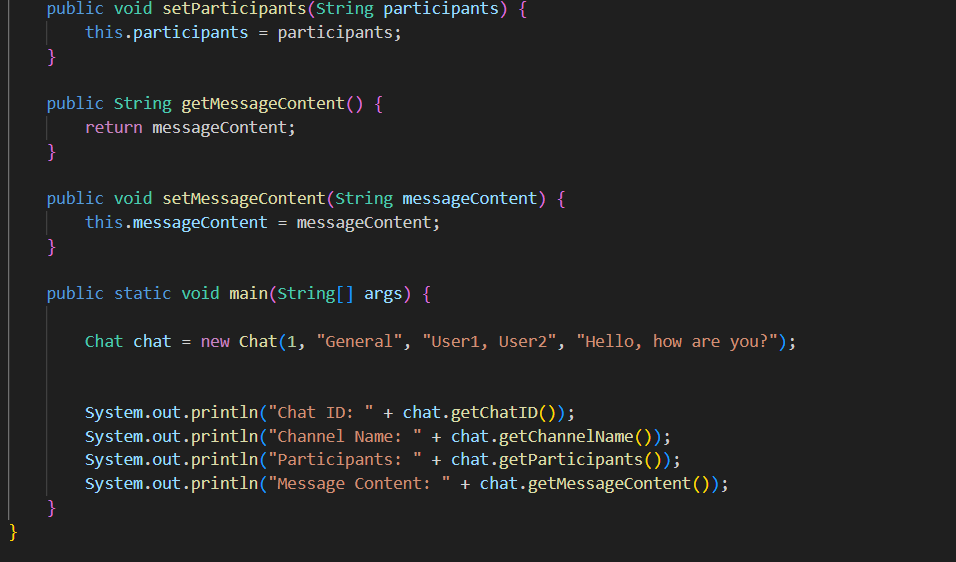
**-Subscription.java:**

****

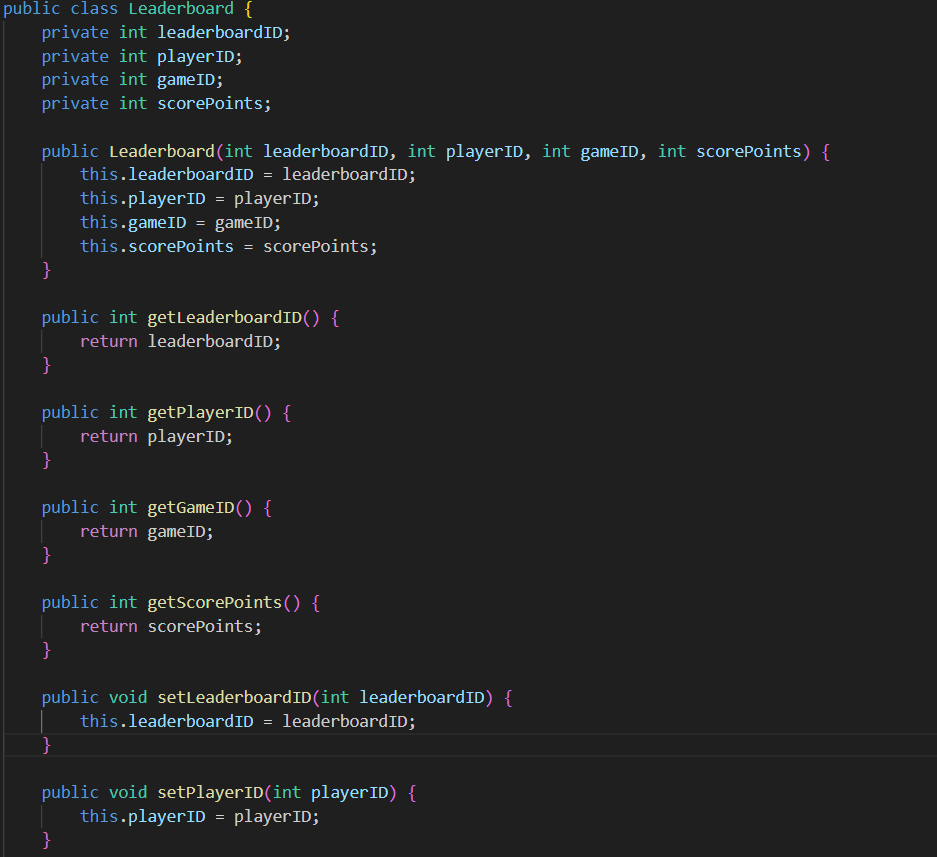
****

**-chat.java:**

****

****

**-leaderboard.java:**

****

****

Challenges list :

* Designing a normalised and efficient database schema to handle complex relationships like many-to-many (e.g.,games,tournament,chat).
* Creating the tables in a specific order and inserting the values accordingly .
* Implementing and managing the various one-to-many and many-to-many relationships in Java objects and ensuring data integrity.
* Ensuring that the data entered into the system is valid and consistent.