**CSCI 4390 Senior Project Initial Proposal**

**Title: Sports Chat+**

**Team Members:**

* **Ruben Aleman (ruben.aleman01@utrgv.edu)**
* **Joe Rivera (joe.rivera02@utrgv.edu)**
* **Abel Victorino (abel.victorino01@utrgv.edu)**
* **Emilio Aguirre (emilio.aguirre01@utrgv.edu)**

**Adviser: Dr. Zhixiang Chen**

**Project Description**

**Introduction**

**Sports Chat+** is a database-driven web platform designed to enhance the March Madness experience by integrating real-time chat, fantasy league features, and friendly wagering. The system will efficiently manage teams, players, games, user interactions, and transactions to create a seamless sports engagement experience.

**Table Naming Conventions**

To maintain consistency in database structure:

* **Primary Keys (PK)**: The primary key of each table is the unique identifier, named as <TableName>ID. Example: TeamID (PK), UserID (PK).
* **Foreign Keys (FK)**: Foreign keys reference related tables, following the pattern <ReferencedTable>ID. Example: TeamID (FK) in Players.
* **Standard Naming**: Table names are **plural** (e.g., Teams, Users), and column names use **camel case** or underscores for readability.

**Data Retrieval Efficiency**

To ensure optimal performance, the database will implement:

* **Indexing**: Frequently queried fields like UserID, TeamID, and GameID will be indexed to speed up search operations.
* **Normalization**: Data will be structured to **minimize redundancy**, ensuring efficient storage and retrieval.
* **Optimized Queries**: Joins and transactions will be designed to reduce query execution time, improving real-time data updates.

**User Access Control**

To manage security and access, user roles will be stored in the **UserPermissions** table:

* **Admin**: Full control over database management, user bans, and system settings.
* **Moderator**: Limited privileges, including chat moderation and managing user-generated content.
* **User**: Standard access, including participating in discussions, viewing games, and placing virtual bets.

User roles will restrict actions like **modifying game data, making payments, or changing permissions** to authorized personnel only.

**Security Considerations**

To protect user data and transactions, the system will implement:

* **Password Hashing**: User passwords will be stored using bcrypt for encryption and security.
* **Transaction Security**: Virtual coin transactions and game betting records will be verified and encrypted.
* **API Access Control**: Rate limiting and authentication measures will prevent unauthorized API requests and spam.
* **Role-Based Access Control (RBAC)**: Ensures only authorized users perform critical operations.

**Database Design Overview**

**Core Tables**

**1. Teams Table**

**Stores NCAA team details participating in March Madness.**

CREATE TABLE Teams (

TeamID INT PRIMARY KEY AUTO\_INCREMENT,

TeamName VARCHAR(100) NOT NULL UNIQUE,

CoachName VARCHAR(100),

Conference VARCHAR(100),

Wins INT DEFAULT 0,

Losses INT DEFAULT 0,

Seed INT CHECK (Seed BETWEEN 1 AND 16)

);

**2. Players Table**

**Tracks player details.**

CREATE TABLE Players (

PlayerID INT PRIMARY KEY AUTO\_INCREMENT,

TeamID INT,

PlayerName VARCHAR(100) NOT NULL,

Position ENUM('Guard', 'Forward', 'Center'),

HeightCM INT,

WeightKG INT,

JerseyNumber INT,

FOREIGN KEY (TeamID) REFERENCES Teams(TeamID) ON DELETE CASCADE

);

**3. Games Table**

**Tracks scheduled and completed games.**

CREATE TABLE Games (

GameID INT PRIMARY KEY AUTO\_INCREMENT,

Round ENUM('First Round', 'Second Round', 'Sweet 16', 'Elite 8', 'Final 4', 'Championship'),

DatePlayed DATE,

Location VARCHAR(255),

Team1ID INT,

Team2ID INT,

WinnerID INT,

ScoreTeam1 INT,

ScoreTeam2 INT,

FOREIGN KEY (Team1ID) REFERENCES Teams(TeamID) ON DELETE CASCADE,

FOREIGN KEY (Team2ID) REFERENCES Teams(TeamID) ON DELETE CASCADE,

FOREIGN KEY (WinnerID) REFERENCES Teams(TeamID) ON DELETE SET NULL

);

**4. GameStats Table**

**Stores player performance for each game.**

CREATE TABLE GameStats (

StatID INT PRIMARY KEY AUTO\_INCREMENT,

GameID INT,

PlayerID INT,

Points INT DEFAULT 0,

Rebounds INT DEFAULT 0,

Assists INT DEFAULT 0,

Steals INT DEFAULT 0,

Blocks INT DEFAULT 0,

MinutesPlayed INT DEFAULT 0,

FOREIGN KEY (GameID) REFERENCES Games(GameID) ON DELETE CASCADE,

FOREIGN KEY (PlayerID) REFERENCES Players(PlayerID) ON DELETE CASCADE

);

**User Interaction Tables**

**5. Users Table**

**Stores user details.**

CREATE TABLE Users (

UserID INT PRIMARY KEY AUTO\_INCREMENT,

Username VARCHAR(50) UNIQUE NOT NULL,

Email VARCHAR(100) UNIQUE NOT NULL,

PasswordHash VARCHAR(255) NOT NULL,

FavoriteTeamID INT NULL,

FOREIGN KEY (FavoriteTeamID) REFERENCES Teams(TeamID) ON DELETE SET NULL

);

**6. ChatRooms Table**

**Defines different chat types.**

CREATE TABLE ChatRooms (

RoomID INT PRIMARY KEY AUTO\_INCREMENT,

RoomName VARCHAR(255) NOT NULL UNIQUE,

RoomType ENUM('Team', 'Game', 'General') NOT NULL DEFAULT 'General',

TeamID INT NULL,

GameID INT NULL,

FOREIGN KEY (TeamID) REFERENCES Teams(TeamID) ON DELETE SET NULL,

FOREIGN KEY (GameID) REFERENCES Games(GameID) ON DELETE SET NULL

);

**7. ChatMessages Table**

**Stores chat messages.**

CREATE TABLE ChatMessages (

MessageID INT PRIMARY KEY AUTO\_INCREMENT,

RoomID INT NOT NULL,

UserID INT NOT NULL,

Message TEXT NOT NULL,

Timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (RoomID) REFERENCES ChatRooms(RoomID) ON DELETE CASCADE,

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE

);

**8. UserPermissions Table**

**Defines access control.**

CREATE TABLE UserPermissions (

UserID INT PRIMARY KEY,

Role ENUM('Admin', 'Moderator', 'User') NOT NULL DEFAULT 'User',

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE

);

**Betting System Tables**

**9. UserCoins Table**

**Tracks user virtual coins.**

CREATE TABLE UserCoins (

UserID INT PRIMARY KEY,

CoinBalance INT DEFAULT 0,

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE

);

**10. Bets Table**

Stores betting activity.

CREATE TABLE Bets (

BetID INT PRIMARY KEY AUTO\_INCREMENT,

GameID INT NOT NULL,

User1ID INT NOT NULL,

User2ID INT NOT NULL,

WagerAmount INT NOT NULL,

BetStatus ENUM('Pending', 'In Progress', 'Completed', 'Cancelled') DEFAULT 'Pending',

WinnerID INT NULL,

FOREIGN KEY (GameID) REFERENCES Games(GameID) ON DELETE CASCADE,

FOREIGN KEY (User1ID) REFERENCES Users(UserID) ON DELETE CASCADE,

FOREIGN KEY (User2ID) REFERENCES Users(UserID) ON DELETE CASCADE,

FOREIGN KEY (WinnerID) REFERENCES Users(UserID) ON DELETE SET NULL

);

**Table Descriptions**

* **Teams Table:** Stores NCAA team information, including names, coaches, and conference details.
* **Players Table:** Tracks individual player details, including team affiliation and statistics.
* **Games Table:** Maintains records of games, including matchups, scores, and winners.
* **GameStats Table:** Logs player performance metrics per game.
* **Users Table:** Stores user credentials and preferences.
* **ChatRooms Table:** Defines various chat rooms for team and game discussions.
* **ChatMessages Table:** Stores user messages exchanged in chat rooms.
* **UserPermissions Table:** Manages user roles and access levels.
* **UserCoins Table:** Tracks users’ virtual coin balances.
* **Bets Table:** Records betting activities between users.
* **Payments Table:** Manages real-money transactions for purchasing virtual coins.
* **PaymentTransactions Table:** Logs all coin-related financial transactions.

**ER Diagram**

A diagram of a diagram

AI-generated content may be incorrect.

**Table Relationship Diagram**

**A diagram of a diagram

AI-generated content may be incorrect.**

**User Access Diagram**

**A diagram of a user access control diagram

AI-generated content may be incorrect.**

**DATA Flow Diagram**

**A diagram of data flow

AI-generated content may be incorrect.**

|  |  |
| --- | --- |
| Table Name | Purpose |
| Teams | Stores NCAA teams |
| Players | Tracks team rosters |
| Games | Manages game schedule and results |
| Users | Stores user information |
| GameStats | Tracks player performance |
| Chat | Supports real-time messaging |
| UserPermissions | Controls user access levels |

**Database Relationships**

1. Teams can have multiple Players.
2. Games involve two Teams and track a Winner.
3. GameStats record performance for each Player in a specific Game.
4. Users can chat about Games.
5. Users can select Favorite Teams.
6. Users can place Bets on Games.
7. Users can purchase virtual currency via Payments.

PaymentTransactions log each User's financial interactions