

Sports Mania

Team Number - 68

Team Members:

- Saatvik Bhatnagar - 2019097
- Saurav Rao - 2019271
- Ajay Kumar - 2019293
- Rahul Dahiya - 2019318
- Taral Jain - 2019392

Description: A system that maintains the database related to sports facilities in college. The Database maintains a Self-Growth portal where students can upload their daily logs and instructors can verify it. Database can maintain different events related to sports and facilities in college for students to participate.

Features:

- Real Time Inventory of Equipment
- Availability of facilities
- Fine Payment
- Crowdfunding for new facilities
- Admin who will approve SG Proposal
- Self Growth portal and daily SG Log which can be verified by instructor
- Equipment issue forum
- Guide for playing all games e.g. game rules & how to use equipments
- If item is sort in inventory, then report to incharge
- Booking of court and professors will be given preference
- Report about equipment damage and there will be a forum where you can see all complaints by students and professors
- Instruction to use electricity in various courts
- Information about Gaming events going to be held in College
- Student Registration

- Students can form groups and can call if they need members for a game
- People can chat to each other

Bonus:

- With the help of AI we determine the usual time a user plays and send them notifications only in that particular time when some other user requests for a player.
- Tell students the current occupancy of different courts
- API of whole data

Stakeholders:

- Students
- Guard
- Incharge of sports
- Professors
- Coach
- Academic admin
- Sports coordinator

Queries for each Stakeholder

User (Any):

1. Create Account
2. Update Info
3. Login
4. Verification (only for profs,guards)
5. Create request for player
6. Read all requests
7. Update own requests
8. Delete own requests
9. If user doesn't specify the delete time, delete the request after 3hrs of start time
10. Delete account

Student:

1. Check fine amount
2. Request for equipment
3. Check available equipments
4. Request for additional player
5. See global requests for players
6. Respond to request
7. Apply for sg
8. Apply for intra-murals etc .
9. Check upcoming/ongoing competitions
10. Fine Payment
11. Raise a complaint/ give feedback
12. Request for cleaning court
13. Request for new equipments
14. Check instructor timings
15. //Query for rules of different games from database
16. //Check the instruction of using electricity in courts
17. Fill logs of day for sg
18. //Request instructor for timings
19. //Check announcements

Guard:

1. Issue equipment
2. Select venue (of duty)
3. See available equipments
4. Unassign equipment
5. Report student
6. Request for cleaning
7. Report damage
8. //Report deterrence against unlawful activities

Sports Admin:

1. Add equipments
2. Check due fine
3. See feedback/complaints
4. //Collect fine
5. See equipments
6. //Block students from portal temporary or permanent
7. Delete all students from portal who passed out
8. //Request poll /ask question
9. Assign sports coordinators position
10. See list of all students on the portal
11. Assign instructor position

Instructor:

1. Change availability timings
2. See requests for timings
3. Raise complaint/give feedback
4. Respond to request for timings
5. Edit current availability status
6. Respond to daily log of students
7. Verify the log is correct or not
8. Assign student credit that sg is completed by student
9. Verify the resume of student // Instructor asks each student to send him the resume on mail
10. Taking attendance of student

Sports coordinator:

1. Add hours to student portal if he go to play outside clg
2. Add announcements
3. Add events
4. //See list of student enrolled
5. Make forms for the event
6. Winner of events

Academic admin:

1. Assign sports admin
2. Approve students sg request
3. Verify if they had completed it or not
4. Delete users records
5. Check due fines of childrens
6. Get list of student completed SG
7. Open the sg portal so that students can apply

Populating the Tables with Data:

create database t68;

use t68;

CREATE TABLE Student

```
(
    UID VARCHAR(50) NOT NULL UNIQUE,
    RollNumber MEDIUMINT NOT NULL UNIQUE PRIMARY KEY CHECK
(RollNumber<=999999 and RollNumber>=100000),
    isSportsAdmin BOOLEAN DEFAULT false,
    Program TinyText CHECK(Program in ('BTECH', 'MTECH', 'PHD','OTHERS')),
    Year YEAR,
    AmountDue INT CHECK(AmountDue>=0),
    foreign key(UID) references from AutheticationSystem(UID)
);
```

CREATE TABLE User

```
(
    UID VARCHAR(50) NOT NULL UNIQUE,
    Type TINYTEXT NOTNULL CHECK(Type in ('Professor','Student','Sports
Admin','Academic Admin','Guard')),
    Email VARCHAR(150) CHECK(Email LIKE `_%@IIITD.AC.IN`),
    FirstName TINYTEXT NOT NULL,
    LastName TINYTEXT NOTNULL,
    foreign key(UID) references from AutheticationSystem(UID)
```

)

CREATE TABLE Accident

(
GuardID VARCHAR(50) ,
AccidentID VARCHAR(50) NOT NULL UNIQUE ,
Status Boolean DEFAULT true,
FineAmount INT DEFAULT 0 (CHECK FineAmount>=0),
Extras TEXT,
DateTime DATETIME NOT NULL
foreign key(GuardID) references from Guard(GuardID)

)

CREATE TABLE Sports_Coordinator

(
UID VARCHAR(50) NOT NULL UNIQUE,
Foreign key(UID) references from AutheticationSystem(UID)

)

CREATE TABLE Coach

(
UID VARCHAR(50) NOT NULL UNIQUE ,
Foreign key(UID) references from AutheticationSystem(UID)

)

CREATE TABLE Guard

(
UID VARCHAR(50) NOT NULL,
GuardID VARCHAR(10) PRIMARY KEY,
Shift String Not Null
Foreign key(UID) references from AutheticationSystem(UID)

)

CREATE TABLE Professor

(
UID VARCHAR(50) NOT NULL,
Foreign key(UID) references from AutheticationSystem(UID)

)

```
Create table Player(  
    UID VarChar(10) Not Null  
);
```

```
Create table Sport(  
    sportsID Integer Not Null,  
    name Varchar(50) Not Null,  
    primary key(SportsID))  
);
```

```
Create table Handbook(  
    sportsID Integer Not Null,  
    rules text,  
    foriegn key(sportsID) references from Sports(sportsID)  
);
```

```
Create table Academic_Admin(  
    UID VarChar(10) Not Null,  
    foreign key(UID) references from User(UID)  
);
```

```
Create table Request(  
    EquipmentID VarChar(10),  
    forign key(EquipmentID) references from Equipment(EquipmentID)  
);
```

```
Create table Sports_Admin(  
    UID VarChar(10) Not Null,  
    foreign key references from User(UID)  
);
```

```
Create table Comment(  
    UID VarChar(10) Not Null,  
    DateTime datetime Not Null,  
    Content String Not Null,  
    foreig key(UID) references from User(UID)  
);
```

```
Create table Equipment(  
    EquipmentID VarChar(10) Not Null,  
    SportsID Integer Not Null,  
    Equipment String Not Null,  
    Availability boolean Not Null,  
    primary key(EquipmentID),  
    foreign key(SportsID) references from Sports(SportsID)  
);
```

```
Create table Request_Player(  
    UID VarChar(10) Not Null,  
    RequestID Integer NotNull,  
    DateTime datetime Not Null,  
    SportsID Integer Not Null,  
    primary key(RequestID),  
    foreign key(SportsID) references from Sports(SportsID)  
);
```

```
Create table Authentication_System(  
    UID VarChar(10) Not Null,  
    password VarChar(10) Not Null,  
    primary key(UID, password)  
);
```


Create table Course(

courseID String Not Null,

StartDate Date Not Null,

EndDate Date Not Null,

Credits TinyInt Not Null,

primary key(courseID)

);

Create table Student_Log_Data(

UID VarChar(10) Not Null,

credit TinyInt,

primary key(UID)

);

Create table Proposal(

UID VarChar(10) Not Null,

receiving_date date Not Null,

acceptance_date date Not Null,

status boolean Not Null,

comments text,

primary key(UID)

);

Create table Attendance_Data(

UID VarChar(10) Not Null,

start_time time Not Null,

end_time time Not Null,

verification boolean Not Null,

Summary text Not Null,

primary key(UID)

);

Create table Complaint(

complaintID VarChar(10) Not Null,

status String Not Null,

DateTime datetime Not Null,

Extras VarChar,

primary key(complaintID)

);

Create table Event(

EventID VarChar(10) Not Null,

Timings timestamp Not Null,

Description text,

primary key(EventID)

);

Create table Facility(

FacilityID Integer Not Null,

SportsID Integer Not Null,

name VarChar Not Null,

occupancy VarChar Not Null,

primary key(FacilityID),

foreign key(SportsID) references from Sports(SportsID)

);

Create table Announcement(

AnnouncementID Integer Not Null,

SportsID Integer Not Null,

SportsName VarChar Not Null,

comments text,

DateTime datetime Not Null,

FormLink text NotNull,
primary key(AnnouncementID),
foreign key(SportsID, SportsName) references from Sports(SportsID, SportsName)
);

Work Distribution: Till now, everything has been done in the presence of all the group members and hence, there is no individual work distribution yet.