PES UNIVERSITY

Department of Computer Science & Engineering

Database Management System

UE23CS351A

Experiential Learning: Level 2 (Orange problem)

Esports Tournament Management System: A Database-Driven Web Application for Seamless Event Coordination

Team Name	Valoris
Team Members	PRAJWAL R (PES1UG23CS428) NISHANT (PES1UG23CS402)
Section	G
Department	CS
Campus	RR

Esports Tournament Management System: A Comprehensive Database-Driven Web Application

The project is a full-stack web application designed for managing eSports or gaming tournaments. It provides an intuitive admin interface for CRUD operations on teams, tournaments, games, and related entities, along with public-facing pages for users to view tournament data, analytics, and live streams. The system leverages Flask (Python) for server logic, MySQL for relational data storage, and Jinja2 templates for dynamic HTML rendering—allowing seamless handling of tournament registration, team participation, match recording, and statistics in a scalable, modular fashion.

User Requirement Specification

Purpose

The project aims to provide a robust Tournament Management System that enables organizers to create tournaments, register teams and players, schedule matches, record results, and automatically compute rankings, ensuring operational efficiency and competitive integrity across events. It targets a clean separation of concerns between administrative operations and competitive data flows, supporting reliable data capture and consistent updates to standings without manual recalculation. The solution is designed for deployment with a relational database (MySQL) to ensure data consistency, referential integrity, and easy export of schema and seed data for academic evaluation and reproducible setup.

Scope

In scope are features for tournament lifecycle management (creation, configuration, and status tracking), team and player management, registration workflows, match scheduling, result recording, and automated rankings/standings within each tournament. Administrative interfaces support privileged actions such as creating tournaments, managing registrations, publishing schedules, and validating results, while team-facing functions cover roster management and participation confirmations per tournament. Out of scope for the current iteration are payment gateways, broadcasting integrations, and anti-cheat analytics; these can be considered as future extensions once the core relational model and workflows are stable.

Detailed Description

The system manages esports events by centering all data around a Tournament, where each tournament has a unique identifier along with its name, game title, edition or season, start and end dates, the chosen competitive format such as single elimination or double elimination or round robin, and a status that indicates whether it is planned, live, or completed.
Each Tournament provides the context for participation, scheduling, match play, and rankings, ensuring that every action and record belongs clearly to a single event.
Teams register to take part in tournaments and each Team record includes a unique identifier with attributes such as name, region, and contact details to support communication and eligibility management.
Players form the roster of a team and each Player has a unique identifier with a handle, full name, role, and country, and every player belongs to exactly one team at a time for the purposes of a tournament.
Participation is captured using a Registration that links one team to one tournament and records the registration status such as pending or approved or rejected, the time of registration, and optional seeding information that can guide bracket creation or group allocation.
A single tournament can have many registrations and a single team can register across many tournaments, but each registration refers to exactly one tournament and exactly one team to avoid duplication and ambiguity.
Scheduling and result recording are handled through a Match that belongs to one tournament and stores the competitive stage such as groups or playoffs or finals, the round number, the scheduled time, the best of value that defines the series length, and the current status such as scheduled or in progress or final.
Every match involves two distinct teams, represented through explicit participation records that connect the match to each team and store the assigned side and score so that byes and walkovers are handled cleanly without overloading match data.
For formats that require a series of games or maps, an optional Game Set captures each individual game in order with its timing information and the winner of that game so the match winner is determined when one side reaches the required number of wins.

When a match is finalized its outcome contributes to the standings of the same tournament so that the leaderboard reflects the latest results without manual recalculation.
Overall rankings are maintained in a Standing that aggregates performance for each team within a tournament including wins, losses, points, and tie break measures such as map difference if needed for fair ordering.
Each tournament can maintain many standing records while any given team appears at most once in the standings of a particular tournament to preserve a clear and consistent leaderboard.
Administrative access is represented by Users with roles such as admin or team manager so that tournament creation, registration approval, scheduling, and result entry are controlled by authorized actors without changing the competitive entities themselves.

Programming languages

- Python 3.x for the web application and server-side logic.

Frameworks and libraries

- Flask as the web framework for routing, views, and request handling.
- Jinja2 for HTML templating within the Flask application.

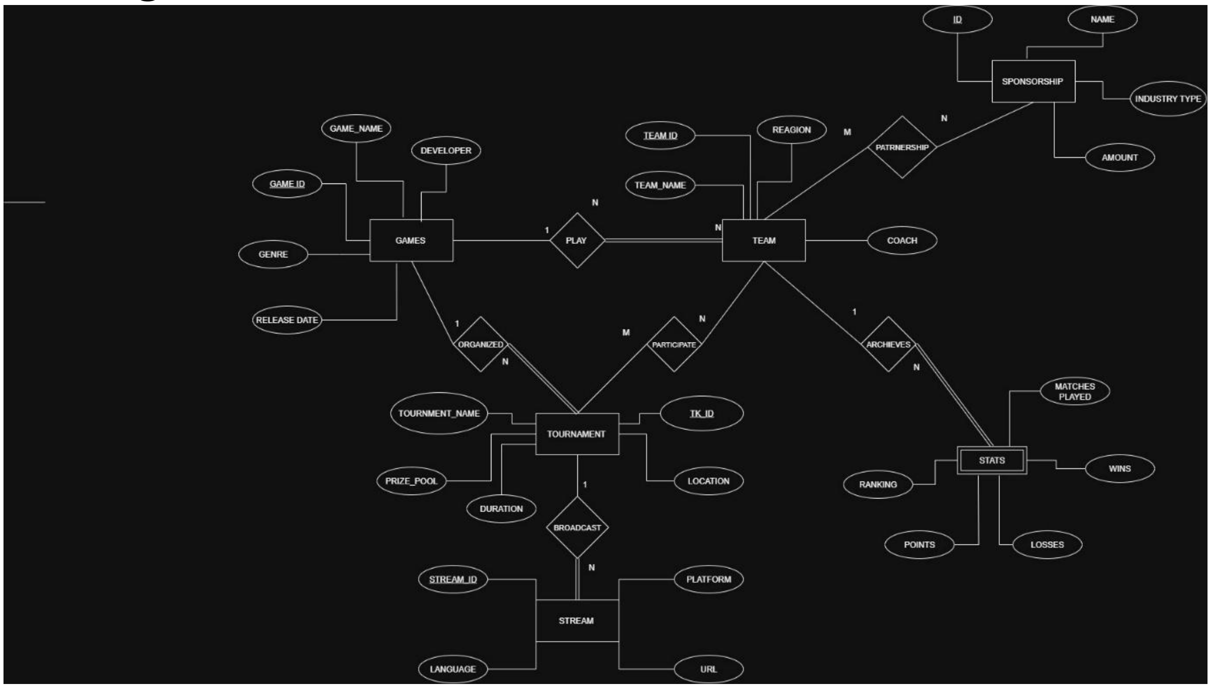
Database and DB tools

- MySQL as the relational database engine for development and testing.

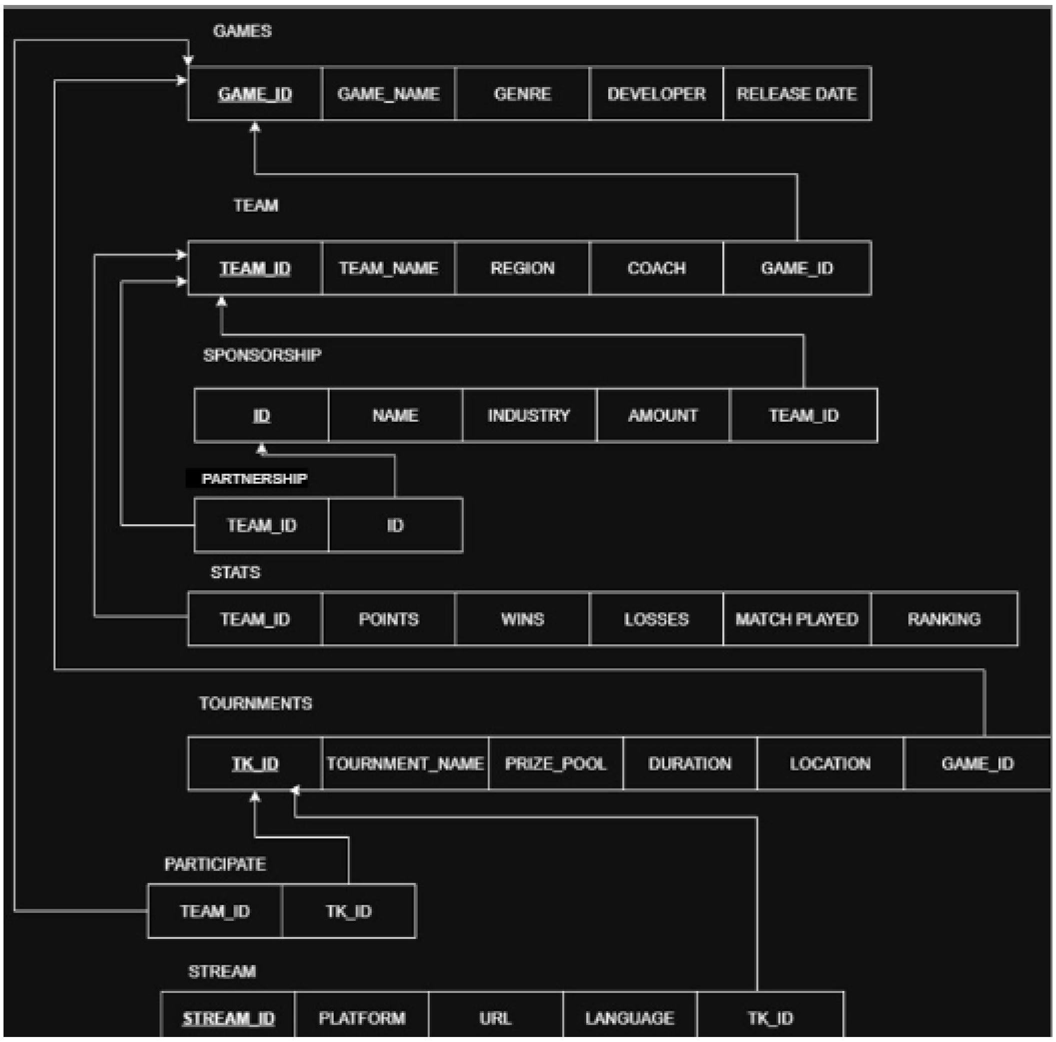
Front-end assets

- HTML, CSS, and JavaScript for the application UI rendered via Flask templates.

ER Diagram


ID
NAME
SPONSORSHIP
INDUSTRY TYPE
AMOUNT
N
M
PATRNERSHIP
GAME_NAME
DEVELOPER
GAME ID
GENRE
GAMES
RELEASE DATE
TEAM ID
REAGION
TEAM_NAME
TEAM
COACH
1
PLAY
N
N
1
ORGANIZED
N
M
PARTICIPATE
N
1
ARCHIEVES
N
TOURNMENT_NAME
TOURNAMENT
TK_ID
PRIZE_POOL
DURATION
LOCATION
1
BROADCAST
N
MATCHES PLAYED
STATS
WINS
RANKING
POINTS
LOSSES
STREAM_ID
PLATFORM
STREAM
LANGUAGE
URL

Relational Schema



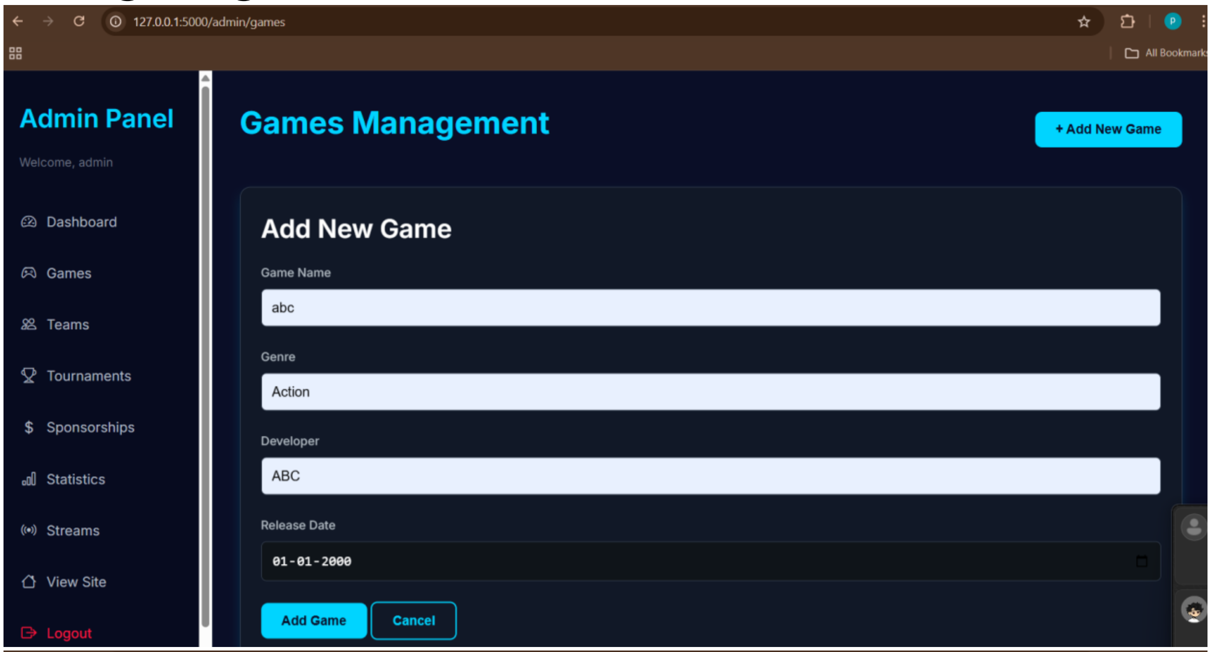
DDL Commands

Refer Create_database.sql file for rest of the sql queries

```
-- Create the database

CREATE DATABASE IF NOT EXISTS `gaming_tournament_db`;


USE `gaming_tournament_db`;


-- Table: admin_users

CREATE TABLE `admin_users` (

  `USER_ID` int NOT NULL AUTO_INCREMENT,
  `USERNAME` varchar(50) NOT NULL,
  `PASSWORD` varchar(100) NOT NULL,
  `CREATED_AT` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`USER_ID`),
  UNIQUE KEY `USERNAME` (`USERNAME`)
);

-- Table: games
CREATE TABLE `games` (
  `GAME_ID` int NOT NULL,
  `GAME_NAME` varchar(100) NOT NULL,
  `GENRE` varchar(50) NOT NULL,
  `DEVELOPER` varchar(100) NOT NULL,
  `RELEASE_DATE` date NOT NULL,
  PRIMARY KEY (`GAME_ID`),
  CONSTRAINT `chk_game_name` CHECK ((`GAME_NAME` <> '')),
  CONSTRAINT `chk_genre` CHECK ((`GENRE` <> ''))
);

-- Table: participate
CREATE TABLE `participate` (
  `TEAM_ID` int NOT NULL,
  `TK_ID` int NOT NULL,
  PRIMARY KEY (`TEAM_ID`,`TK_ID`),
  KEY `fk_partnership_tournament` (`TK_ID`),
  CONSTRAINT `fk_partnership_team` FOREIGN KEY (`TEAM_ID`) REFERENCES `team` (`TEAM_ID`)
ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_partnership_tournament` FOREIGN KEY (`TK_ID`) REFERENCES `tournaments`
(`TK_ID`) ON DELETE CASCADE ON UPDATE CASCADE
);

-- Table: partnership
CREATE TABLE `partnership` (
  `TEAM_ID` int NOT NULL,
  `ID` int NOT NULL,
  PRIMARY KEY (`TEAM_ID`,`ID`),

  KEY `fk_participates_sponsorship` (`ID`),

  CONSTRAINT `fk_participates_sponsorship` FOREIGN KEY (`ID`) REFERENCES `sponsorship` (`ID`) ON
DELETE CASCADE ON UPDATE CASCADE,

  CONSTRAINT `fk_participates_team` FOREIGN KEY (`TEAM_ID`) REFERENCES `team` (`TEAM_ID`)
ON DELETE CASCADE ON UPDATE CASCADE

);


-- Table: sponsorship

CREATE TABLE `sponsorship` (

  `ID` int NOT NULL AUTO_INCREMENT,

  `NAME` varchar(100) NOT NULL,

  `INDUSTRY` varchar(50) NOT NULL,

  `AMOUNT` decimal(15,2) NOT NULL,

  `TEAM_ID` int NOT NULL,

  PRIMARY KEY (`ID`),

  KEY `idx_sponsorship_team` (`TEAM_ID`),

  CONSTRAINT `fk_sponsorship_team` FOREIGN KEY (`TEAM_ID`) REFERENCES `team` (`TEAM_ID`)
ON DELETE CASCADE ON UPDATE CASCADE,

  CONSTRAINT `chk_amount` CHECK ((`AMOUNT` > 0)),

  CONSTRAINT `chk_sponsor_name` CHECK ((`NAME` <> ''))

);


-- Table: stats

CREATE TABLE `stats` (

  `TEAM_ID` int NOT NULL,

  `POINTS` int NOT NULL DEFAULT '0',

  `WINS` int NOT NULL DEFAULT '0',

  `LOSSES` int NOT NULL DEFAULT '0',

  `MATCH_PLAYED` int NOT NULL DEFAULT '0',

  `RANKING` int NOT NULL,

  PRIMARY KEY (`TEAM_ID`),

  CONSTRAINT `fk_stats_team` FOREIGN KEY (`TEAM_ID`) REFERENCES `team` (`TEAM_ID`) ON
DELETE CASCADE ON UPDATE CASCADE,

  CONSTRAINT `chk_losses` CHECK ((`LOSSES` >= 0)),

  CONSTRAINT `chk_match_consistency` CHECK ((`MATCH_PLAYED` >= (`WINS` + `LOSSES`))),
  CONSTRAINT `chk_matches` CHECK ((`MATCH_PLAYED` >= 0)),
  CONSTRAINT `chk_points` CHECK ((`POINTS` >= 0)),
  CONSTRAINT `chk_ranking` CHECK ((`RANKING` > 0)),
  CONSTRAINT `chk_wins` CHECK ((`WINS` >= 0))
);

-- Table: stream
CREATE TABLE `stream` (
  `STREAM_ID` int NOT NULL AUTO_INCREMENT,
  `PLATFORM` varchar(50) NOT NULL,
  `URL` varchar(255) NOT NULL,
  `LANGUAGE` varchar(30) NOT NULL,
  `TK_ID` int NOT NULL,
  PRIMARY KEY (`STREAM_ID`),
  KEY `idx_stream_tournament` (`TK_ID`),
  CONSTRAINT `fk_stream_tournament` FOREIGN KEY (`TK_ID`) REFERENCES `tournaments` (`TK_ID`)
ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `chk_language` CHECK ((`LANGUAGE` <> '')),
  CONSTRAINT `chk_platform` CHECK ((`PLATFORM` <> '')),
  CONSTRAINT `chk_url` CHECK ((`URL` <> ''))
);

-- Table: team
CREATE TABLE `team` (
  `TEAM_ID` int NOT NULL AUTO_INCREMENT,
  `TEAM_NAME` varchar(100) NOT NULL,
  `REGION` varchar(50) NOT NULL,
  `COACH` varchar(100) NOT NULL,
  `GAME_ID` int NOT NULL,
  PRIMARY KEY (`TEAM_ID`),
  KEY `idx_team_game` (`GAME_ID`),
  KEY `idx_team_name` (`TEAM_NAME`),
  CONSTRAINT `fk_team_game` FOREIGN KEY (`GAME_ID`) REFERENCES `games` (`GAME_ID`) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `chk_region` CHECK ((`REGION` <> '')),
  CONSTRAINT `chk_team_name` CHECK ((`TEAM_NAME` <> ''))
);

-- Table: tournaments
CREATE TABLE `tournaments` (
  `TK_ID` int NOT NULL AUTO_INCREMENT,
  `TOURNAMENT_NAME` varchar(150) NOT NULL,
  `PRIZE_POOL` decimal(15,2) NOT NULL,
  `DURATION` varchar(50) NOT NULL,
  `LOCATION` varchar(100) NOT NULL,
  `GAME_ID` int NOT NULL,
  `STATUS` enum('UPCOMING','ONGOING','COMPLETED') DEFAULT 'UPCOMING',
  `WINNER_TEAM_ID` int DEFAULT NULL,
  PRIMARY KEY (`TK_ID`),
  KEY `idx_tournament_game` (`GAME_ID`),
  KEY `idx_tournament_name` (`TOURNAMENT_NAME`),
  KEY `fk_tournament_winner` (`WINNER_TEAM_ID`),
  CONSTRAINT `fk_tournament_game` FOREIGN KEY (`GAME_ID`) REFERENCES `games`
(`GAME_ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_tournament_winner` FOREIGN KEY (`WINNER_TEAM_ID`) REFERENCES `team`
(`TEAM_ID`),
  CONSTRAINT `chk_prize_pool` CHECK ((`PRIZE_POOL` > 0)),
  CONSTRAINT `chk_tournament_name` CHECK ((`TOURNAMENT_NAME` <> ''))
);
```

8. Walk through the Web page
(showing all CRUD operations and functions/procedures)

Adding new game



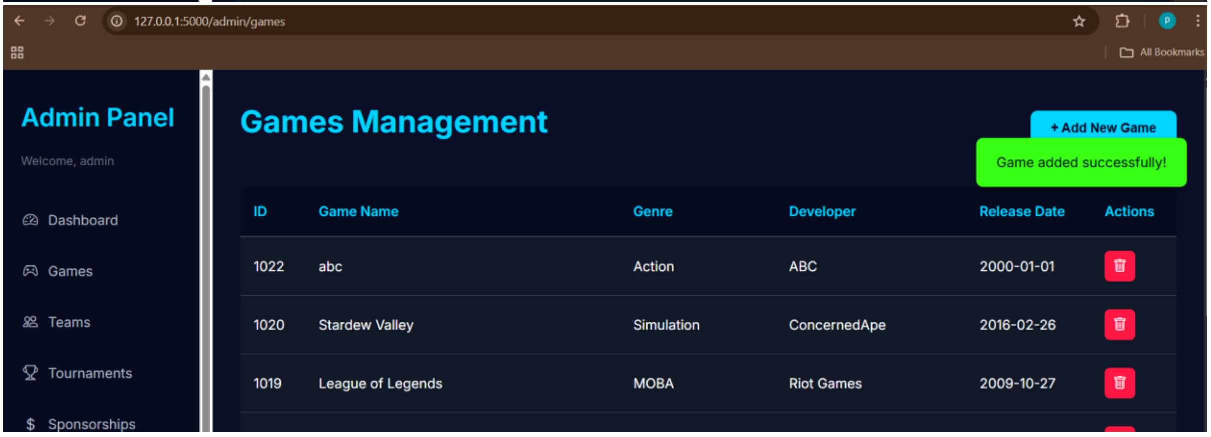
Add New Game

Game Name: abc

Genre: Action

Developer: ABC

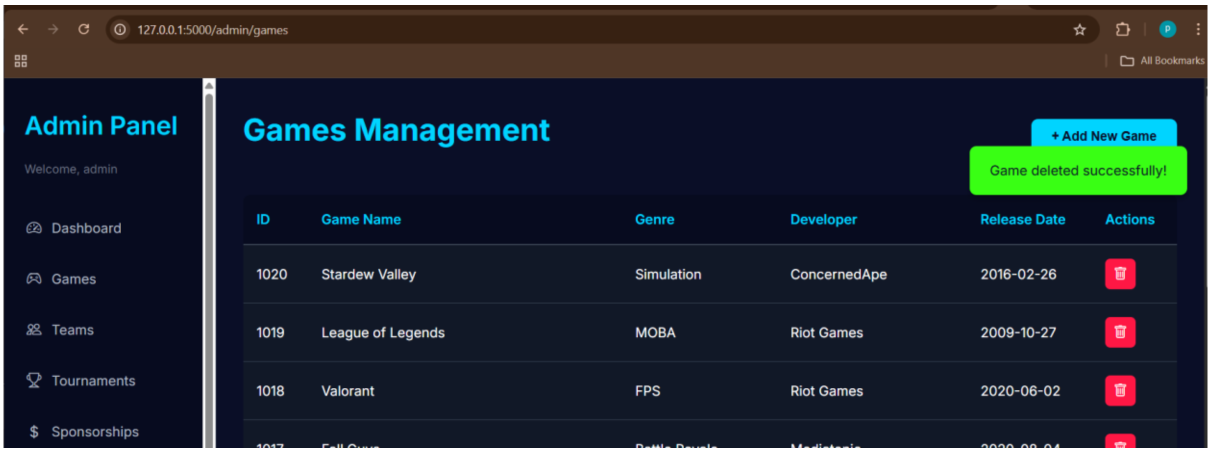
Release Date: 01-01-2000



Game added successfully!

ID	Game Name	Genre	Developer	Release Date	Actions
1022	abc	Action	ABC	2000-01-01	🗑
1020	Stardew Valley	Simulation	ConcernedApe	2016-02-26	🗑
1019	League of Legends	MOBA	Riot Games	2009-10-27	🗑

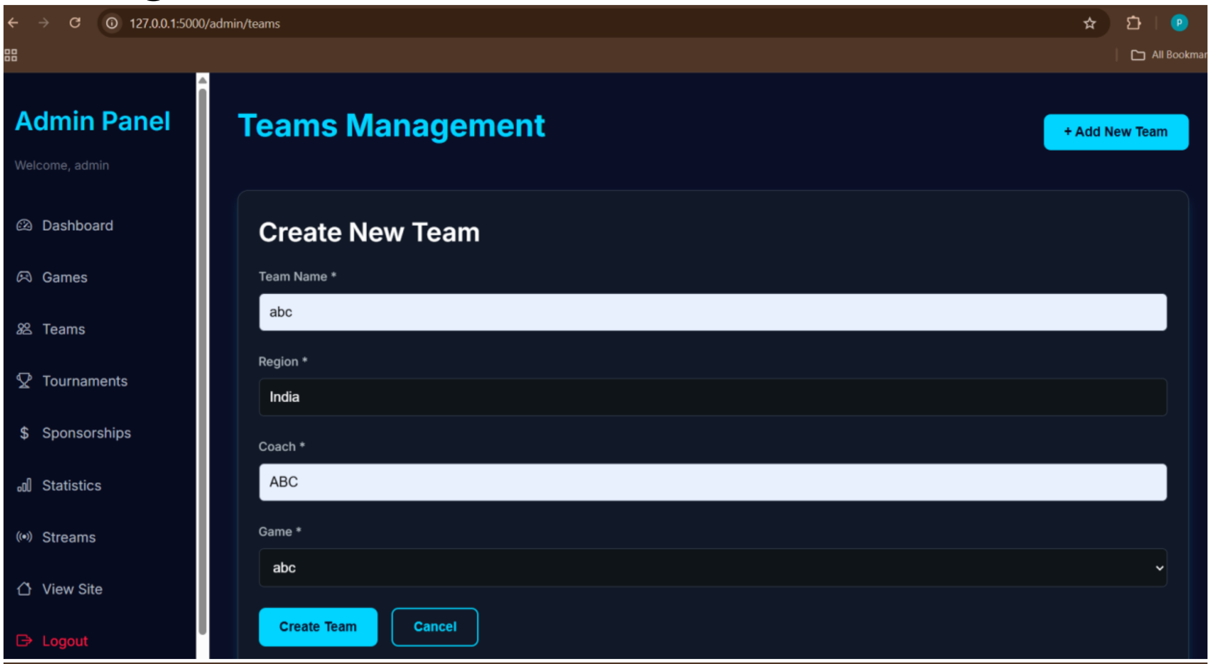
Game abc deleted



Game deleted successfully!

ID	Game Name	Genre	Developer	Release Date	Actions
1020	Stardew Valley	Simulation	ConcernedApe	2016-02-26	🗑
1019	League of Legends	MOBA	Riot Games	2009-10-27	🗑
1018	Valorant	FPS	Riot Games	2020-06-02	🗑

Creating new team


127.0.0.1:5000/admin/teams

All Bookmar

Admin Panel

Welcome, admin

Dashboard
Games
Teams
Tournaments
Sponsorships
Statistics
Streams
View Site
Logout

Teams Management

+ Add New Team

Create New Team

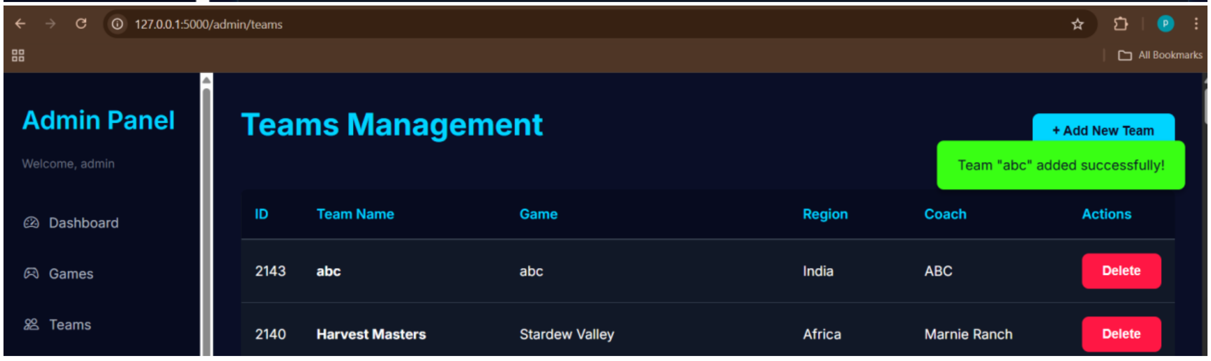
Team Name *
abc

Region *
India

Coach *
ABC

Game *
abc

Create Team Cancel


127.0.0.1:5000/admin/teams

All Bookmarks

Admin Panel

Welcome, admin

Dashboard
Games
Teams

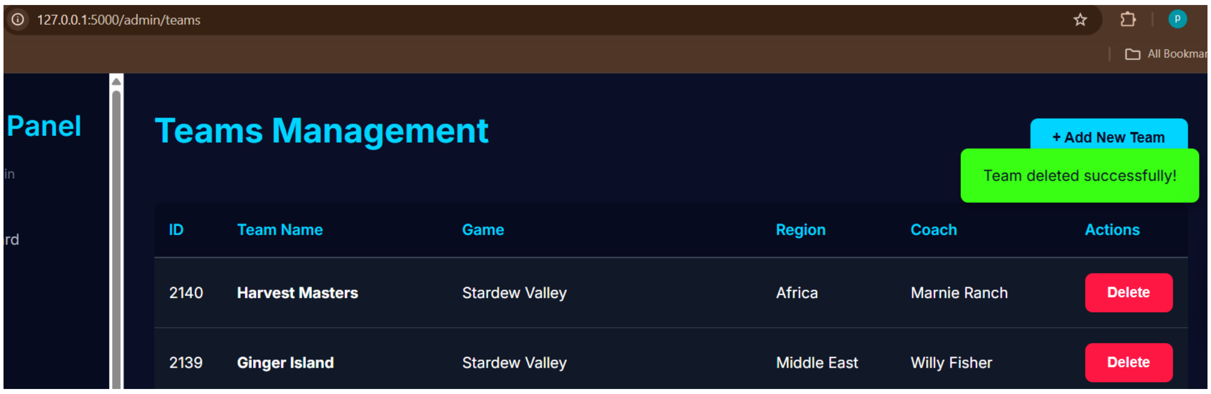
Teams Management

+ Add New Team

Team "abc" added successfully!

ID	Team Name	Game	Region	Coach	Actions
2143	**abc**	abc	India	ABC	Delete
2140	**Harvest Masters**	Stardew Valley	Africa	Marnie Ranch	Delete

Deleted the abc team


127.0.0.1:5000/admin/teams

All Bookmar

Panel

in

rd

Teams Management

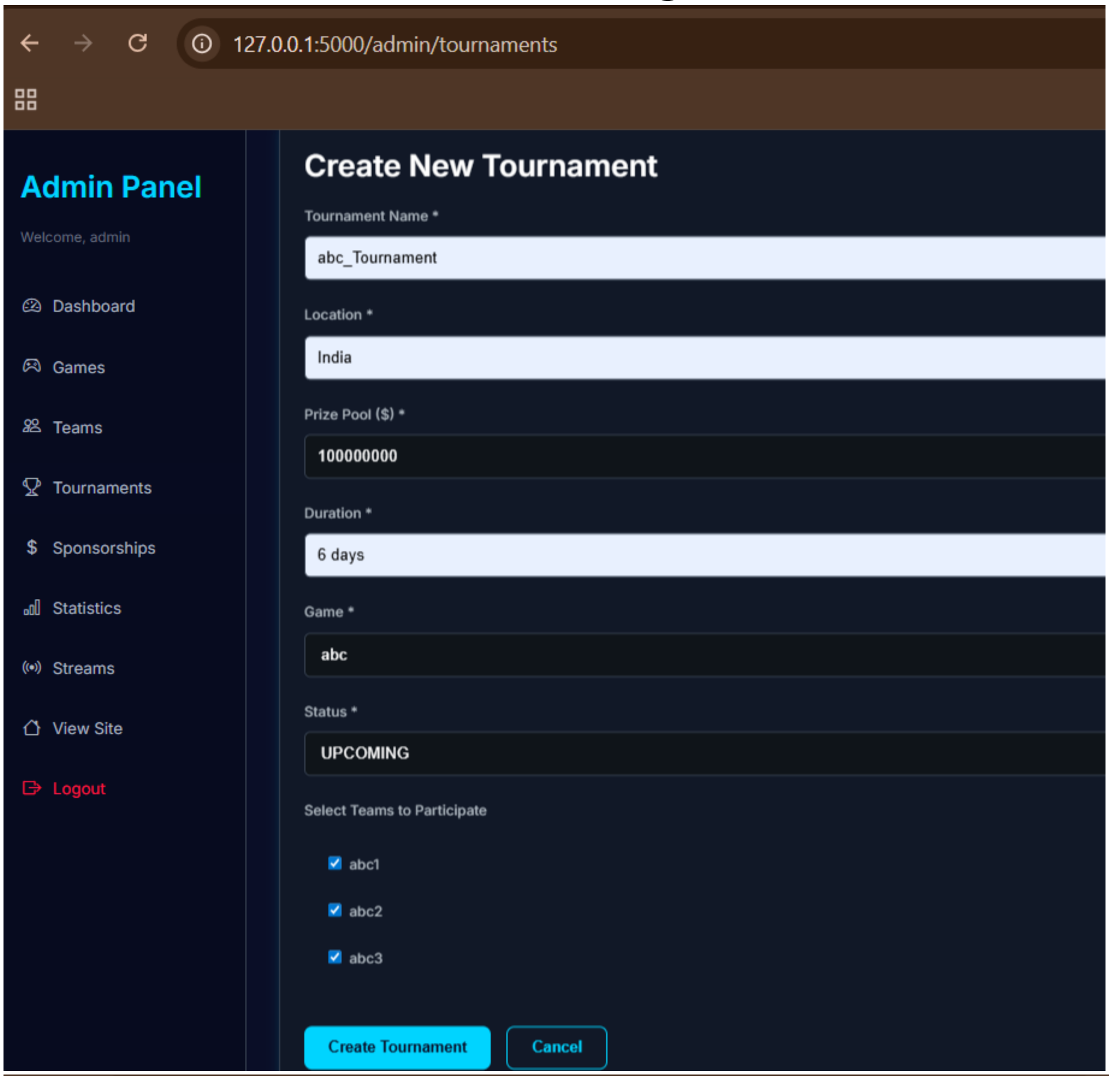
+ Add New Team

Team deleted successfully!

ID	Team Name	Game	Region	Coach	Actions
2140	**Harvest Masters**	Stardew Valley	Africa	Marnie Ranch	Delete
2139	**Ginger Island**	Stardew Valley	Middle East	Willy Fisher	Delete

Created 3 new teams for abc game

Will create new tournament for the game abc



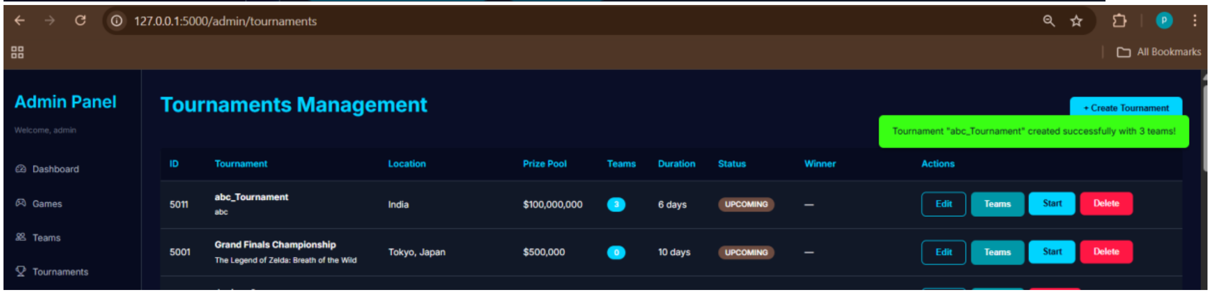
127.0.0.1:5000/admin/tournaments

Create New Tournament

Tournament Name *	abc_Tournament
Location *	India
Prize Pool ($) *	100000000
Duration *	6 days
Game *	abc
Status *	UPCOMING

Select Teams to Participate

- [x] abc1
- [x] abc2
- [x] abc3

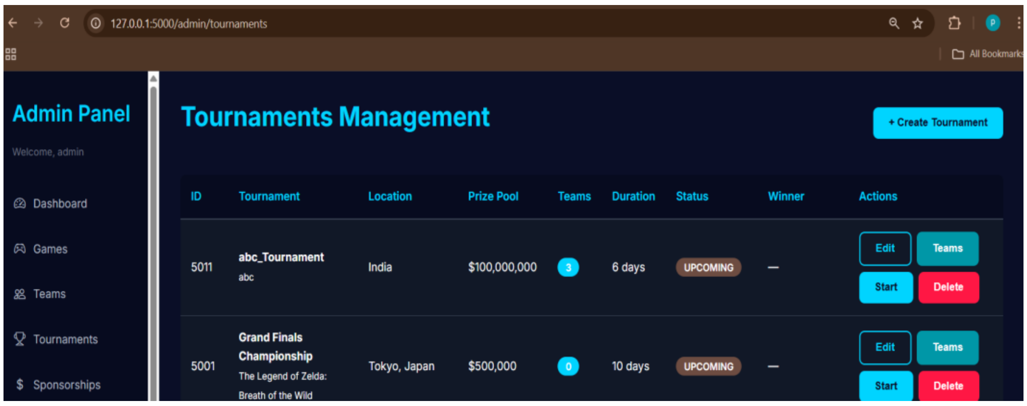


127.0.0.1:5000/admin/tournaments

Tournaments Management

Tournament "abc_Tournament" created successfully with 3 teams!

ID	Tournament	Location	Prize Pool	Teams	Duration	Status	Winner	Actions
5011	abc_Tournament abc	India	$100,000,000	3	6 days	UPCOMING	—	Edit Teams Start Delete
5001	Grand Finals Championship The Legend of Zelda: Breath of the Wild	Tokyo, Japan	$500,000	0	10 days	UPCOMING	—	Edit Teams Start Delete


127.0.0.1:5000/admin/tournaments

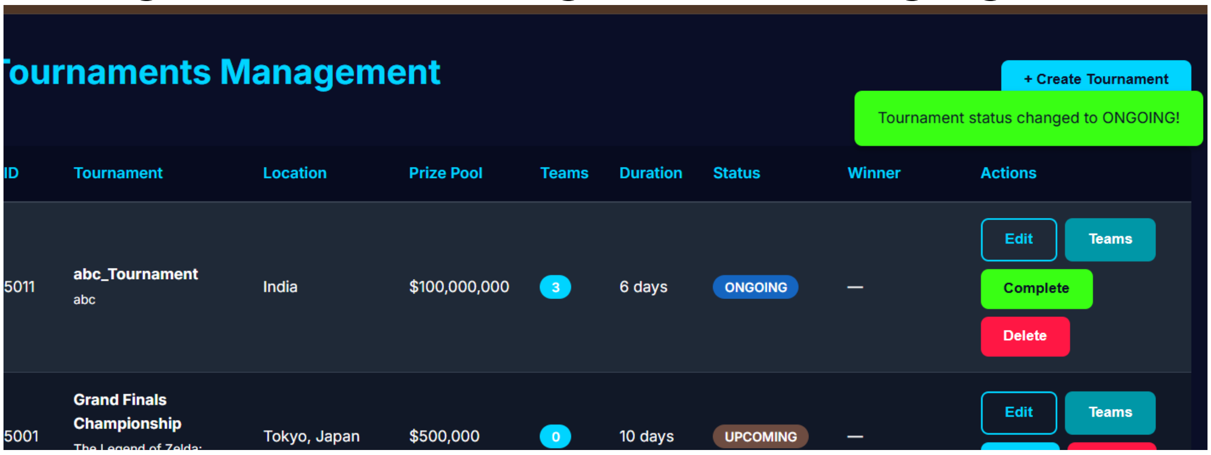
Admin Panel

Welcome, admin

Tournaments Management

ID	Tournament	Location	Prize Pool	Teams	Duration	Status	Winner	Actions
5011	abc_Tournament abc	India	$100,000,000	3	6 days	UPCOMING	—	Edit Teams Start Delete
5001	Grand Finals Championship The Legend of Zelda: Breath of the Wild	Tokyo, Japan	$500,000	0	10 days	UPCOMING	—	Edit Teams Start Delete

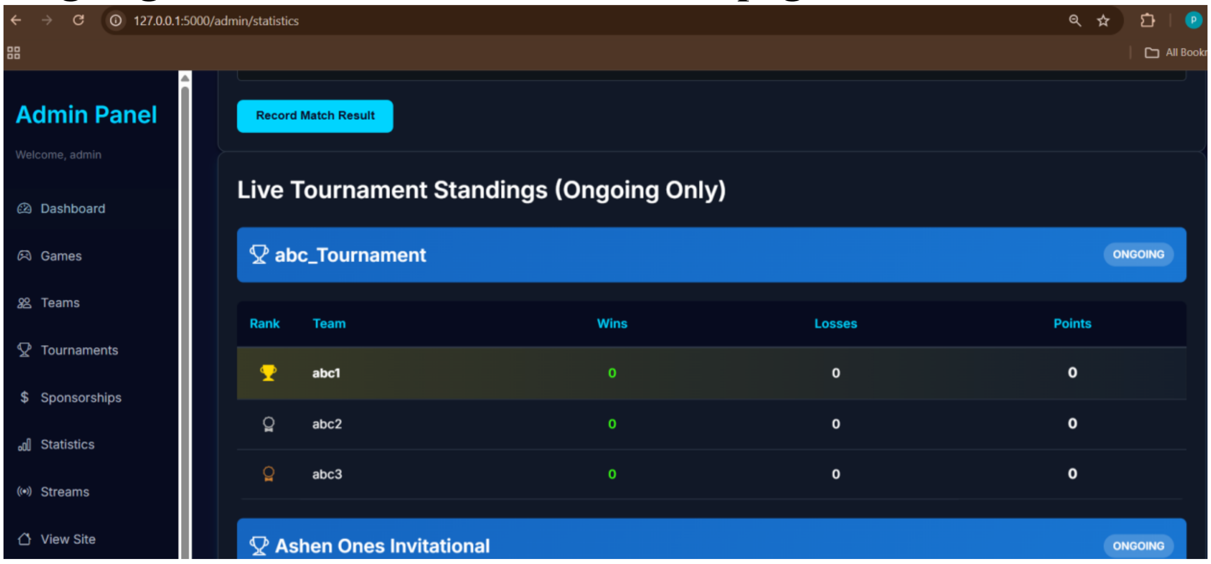
Pressing the Start button changes the status to ongoing


Tournaments Management

Tournament status changed to ONGOING!

ID	Tournament	Location	Prize Pool	Teams	Duration	Status	Winner	Actions
5011	abc_Tournament abc	India	$100,000,000	3	6 days	ONGOING	—	Edit Teams Complete Delete
5001	Grand Finals Championship The Legend of Zelda:	Tokyo, Japan	$500,000	0	10 days	UPCOMING	—	Edit Teams

Ongoing tournaments visible in statistic page


127.0.0.1:5000/admin/statistics

Admin Panel

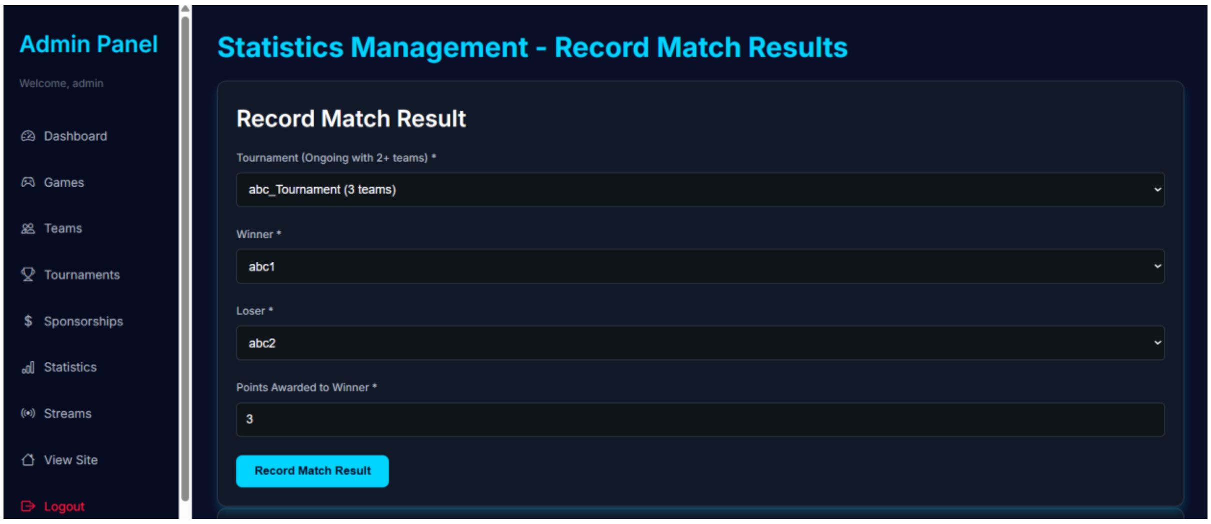
Welcome, admin

Live Tournament Standings (Ongoing Only)

abc_Tournament ONGOING

Rank	Team	Wins	Losses	Points
🏆	abc1	0	0	0
	abc2	0	0	0
	abc3	0	0	0

Ashen Ones Invitational ONGOING


Admin Panel

Welcome, admin

Dashboard
Games
Teams
Tournaments
Sponsorships
Statistics
Streams
View Site
Logout

Statistics Management - Record Match Results

Record Match Result

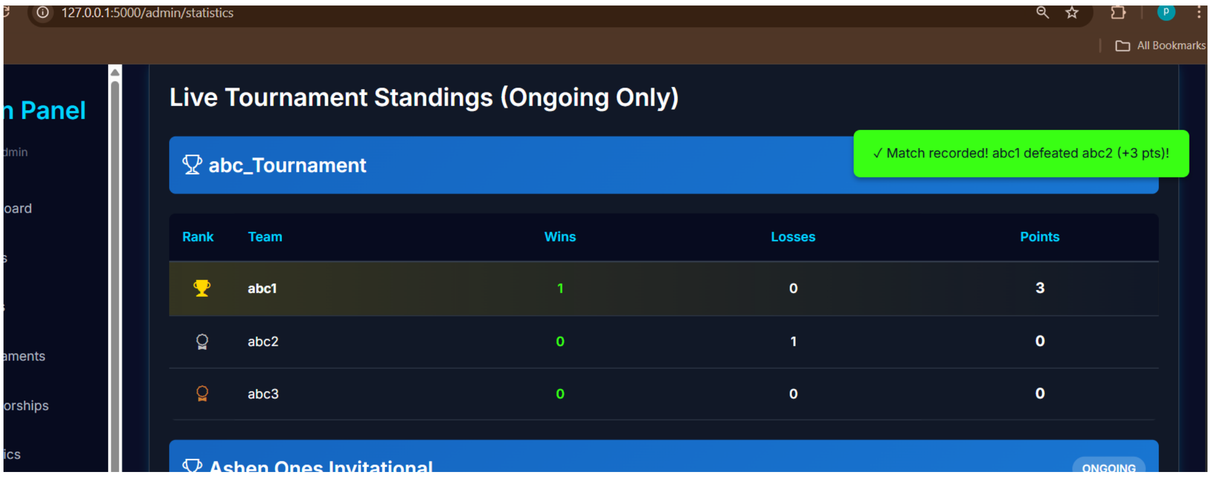
Tournament (Ongoing with 2+ teams) *: abc_Tournament (3 teams)

Winner *: abc1

Loser *: abc2

Points Awarded to Winner *: 3

Record Match Result


127.0.0.1:5000/admin/statistics

Live Tournament Standings (Ongoing Only)

abc_Tournament

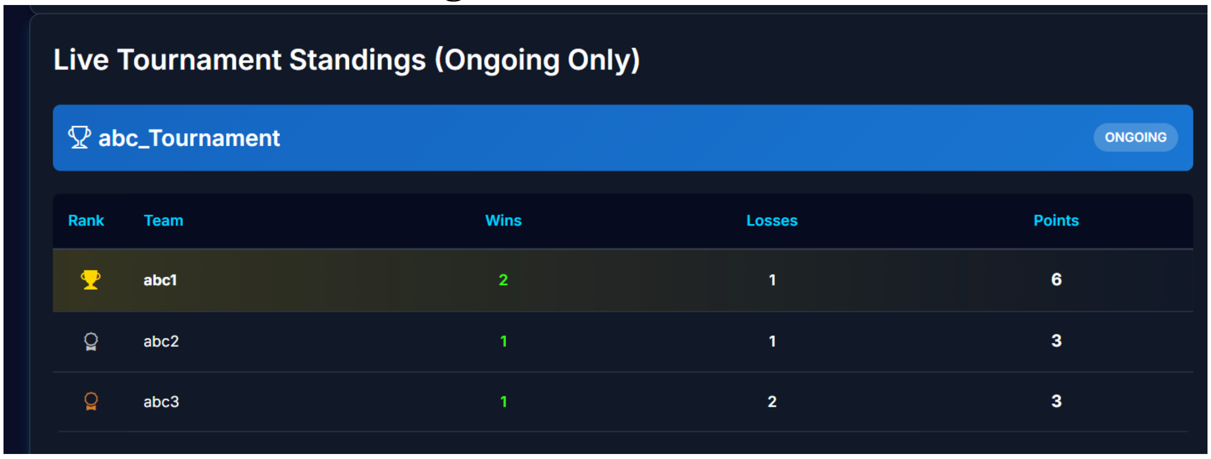
✓ Match recorded! abc1 defeated abc2 (+3 pts)!

Rank	Team	Wins	Losses	Points
🏆	abc1	1	0	3
	abc2	0	1	0
	abc3	0	0	0

Ashen Ones Invitational ONGOING

Ranking updated automatically

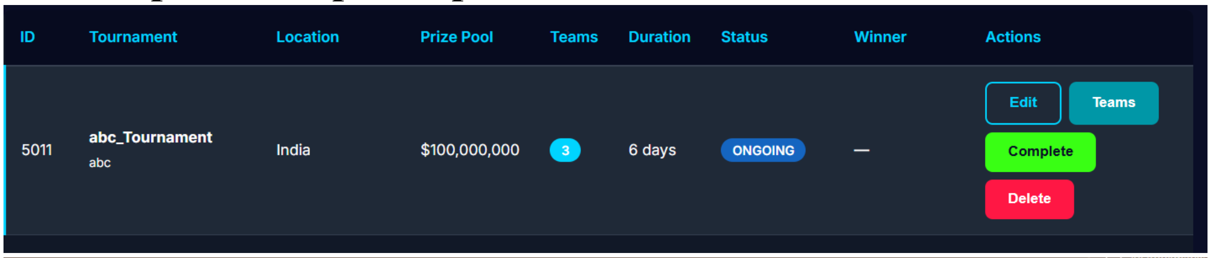
Given the below standings


Live Tournament Standings (Ongoing Only)

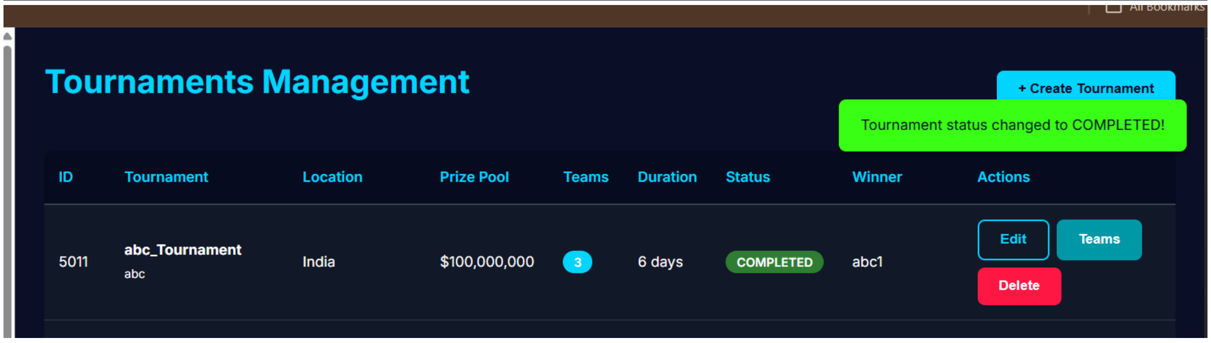
abc_Tournament ONGOING

Rank	Team	Wins	Losses	Points
🏆	abc1	2	1	6
	abc2	1	1	3
	abc3	1	2	3

When I press complete option in tournaments



ID	Tournament	Location	Prize Pool	Teams	Duration	Status	Winner	Actions
5011	abc_Tournament abc	India	$100,000,000	3	6 days	ONGOING	—	Edit Teams Complete Delete



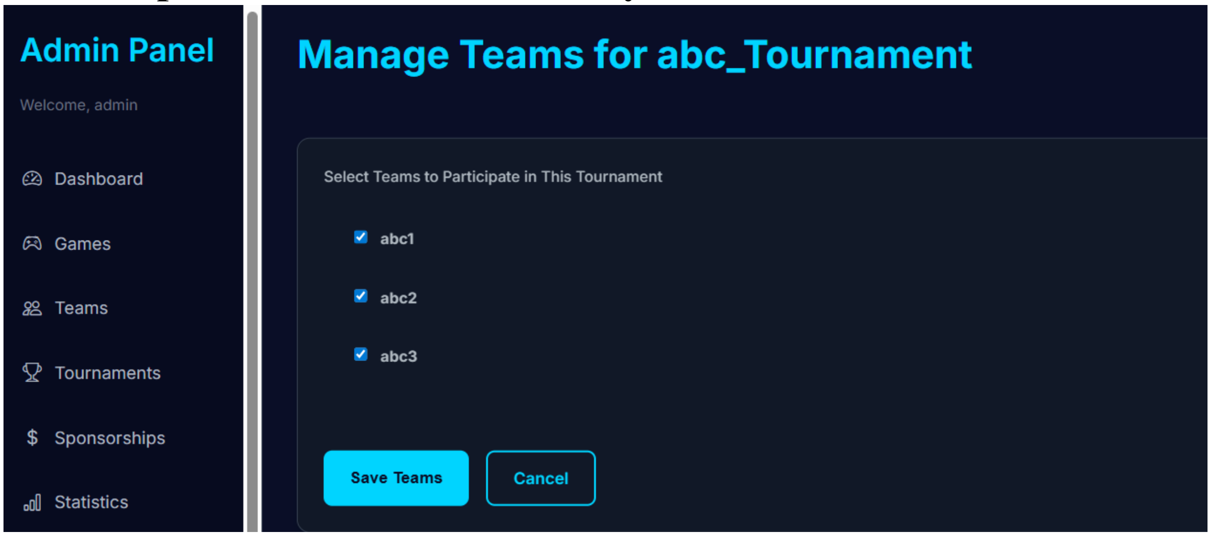
Tournaments Management

Tournament status changed to COMPLETED!

ID	Tournament	Location	Prize Pool	Teams	Duration	Status	Winner	Actions
5011	abc_Tournament abc	India	$100,000,000	3	6 days	COMPLETED	abc1	Edit Teams Delete

Winner automatically chosen based on ranking

Teams option in tournaments lets you add or remove teams

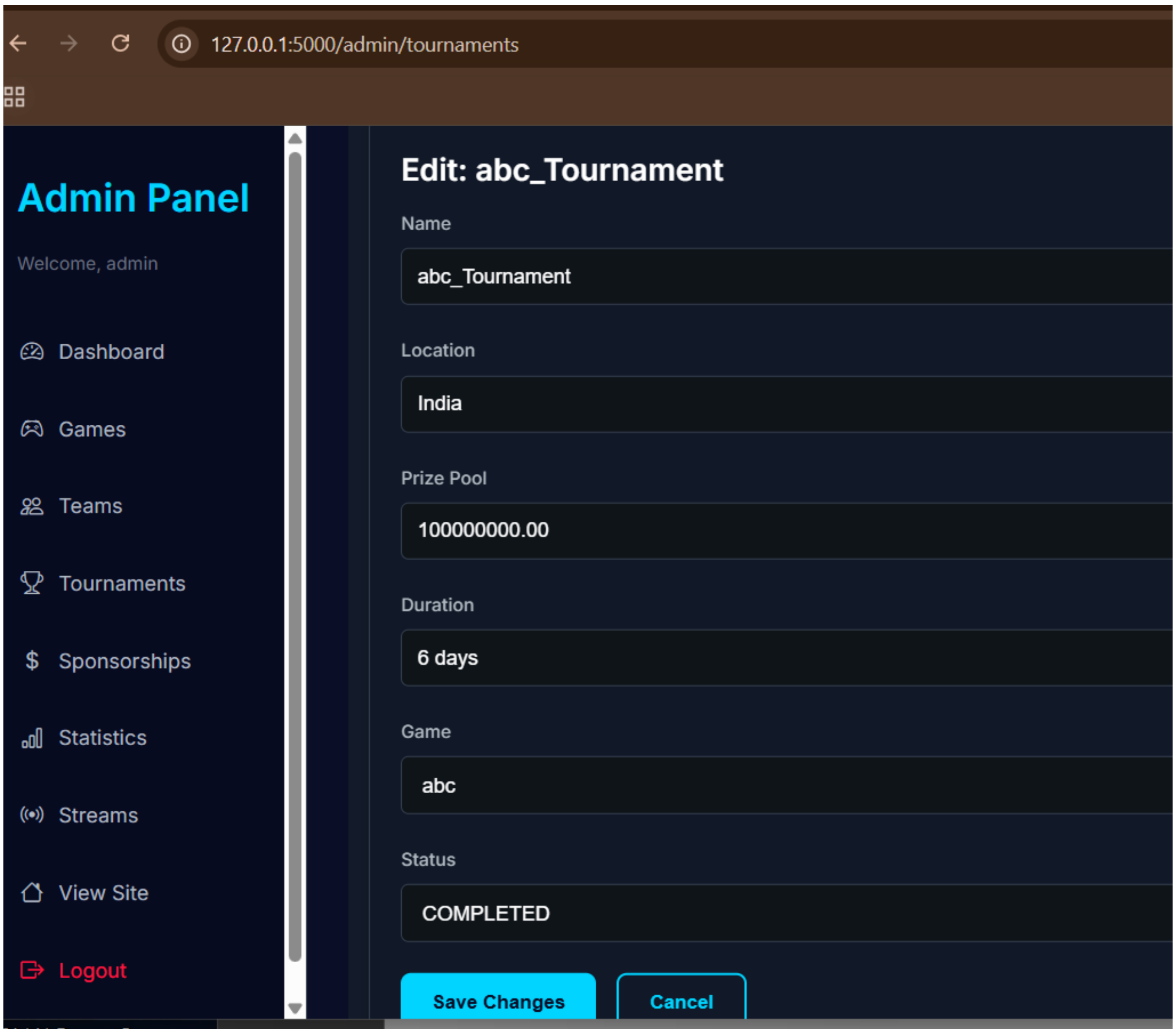


Manage Teams for abc_Tournament

Select Teams to Participate in This Tournament

- [x] abc1
- [x] abc2
- [x] abc3

Edit option lets us edit the details of tournament

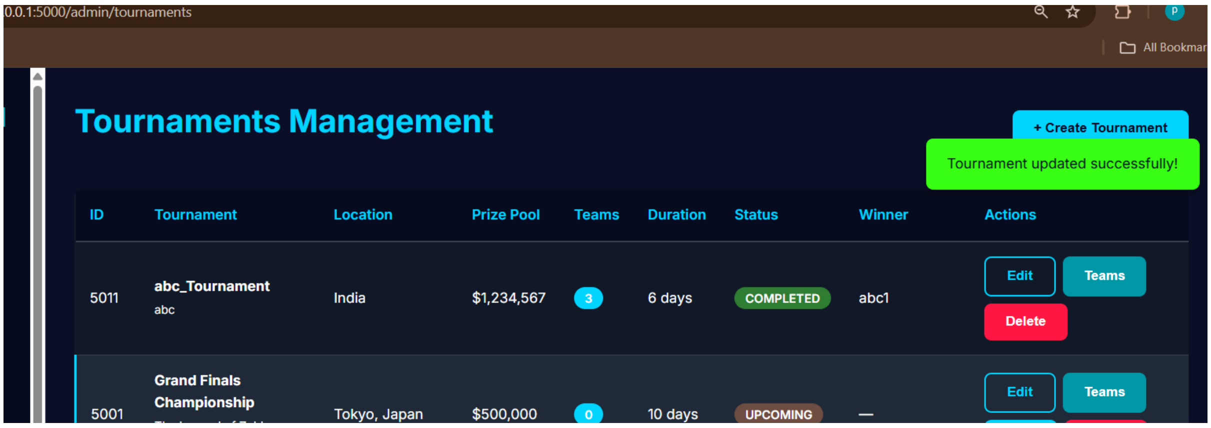


127.0.0.1:5000/admin/tournaments

Edit: abc_Tournament

Name	abc_Tournament
Location	India
Prize Pool	100000000.00
Duration	6 days
Game	abc
Status	COMPLETED

Updated the prize pool using edit option



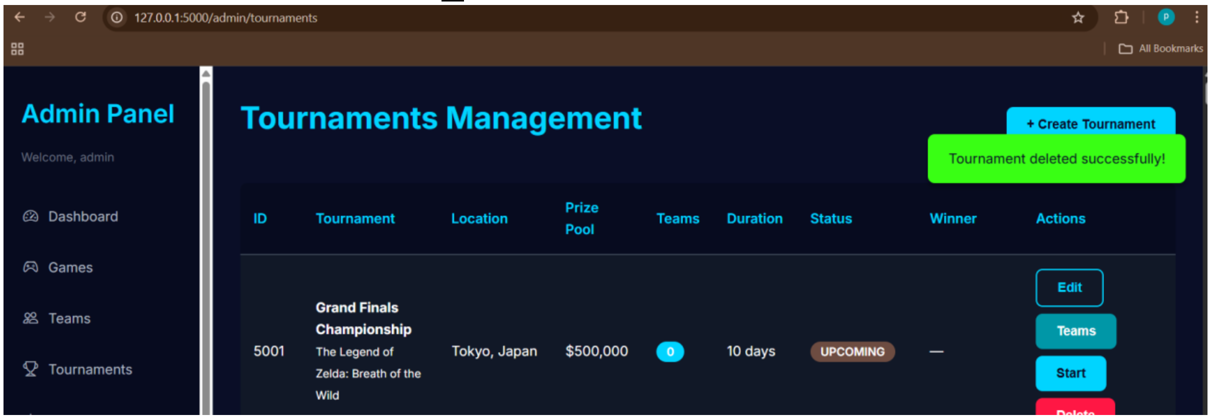
0.0.1:5000/admin/tournaments

Tournaments Management

Tournament updated successfully!

ID	Tournament	Location	Prize Pool	Teams	Duration	Status	Winner	Actions
5011	abc_Tournament abc	India	$1,234,567	3	6 days	COMPLETED	abc1	Edit Teams Delete
5001	Grand Finals Championship	Tokyo, Japan	$500,000	0	10 days	UPCOMING	—	Edit Teams

Deleted tournament abc_tournament



127.0.0.1:5000/admin/tournaments

Admin Panel

Welcome, admin

Dashboard
Games
Teams
Tournaments

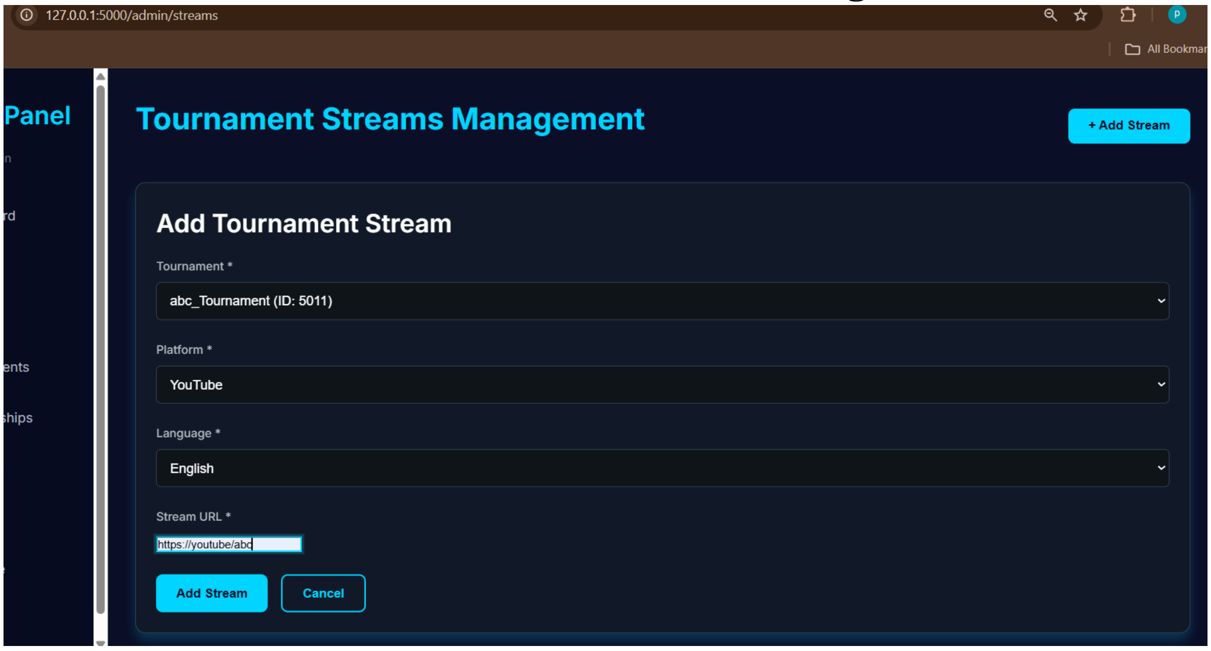
Tournaments Management

+ Create Tournament

Tournament deleted successfully!

ID	Tournament	Location	Prize Pool	Teams	Duration	Status	Winner	Actions
5001	Grand Finals Championship The Legend of Zelda: Breath of the Wild	Tokyo, Japan	$500,000	0	10 days	UPCOMING	—	Edit Teams Start

Streams section allows to add stream link to a given tournament



127.0.0.1:5000/admin/streams

Tournament Streams Management

+ Add Stream

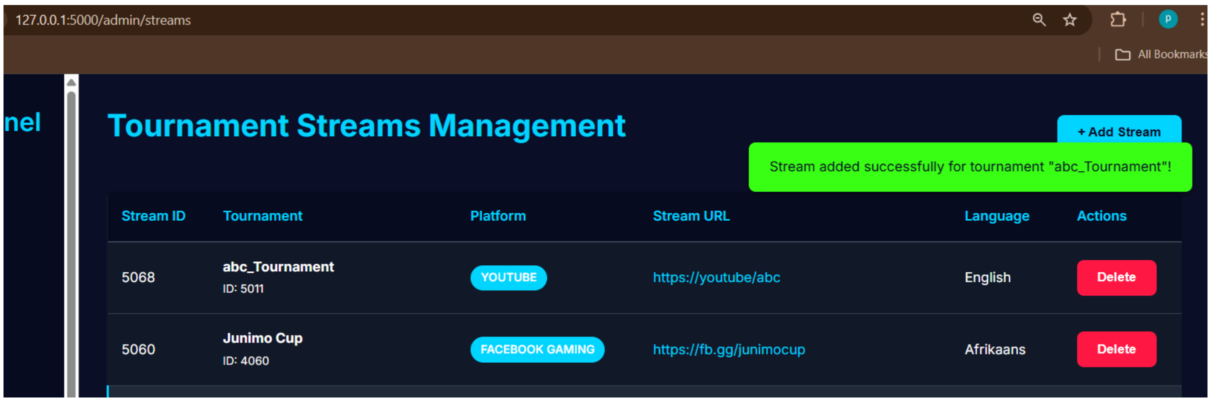
Add Tournament Stream

Tournament *
abc_Tournament (ID: 5011)

Platform *
YouTube

Language *
English

Stream URL *
https://youtube/abc



127.0.0.1:5000/admin/streams

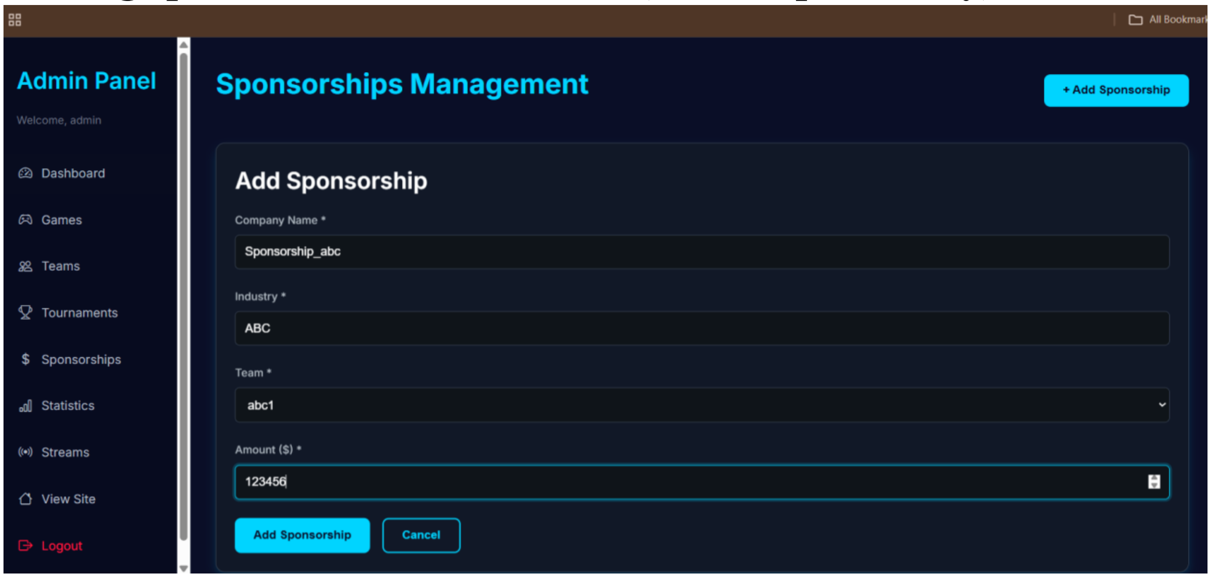
Tournament Streams Management

+ Add Stream

Stream added successfully for tournament "abc_Tournament"!

Stream ID	Tournament	Platform	Stream URL	Language	Actions
5068	abc_Tournament ID: 5011	YOUTUBE	https://youtube/abc	English	Delete
5060	Junimo Cup ID: 4060	FACEBOOK GAMING	https://fb.gg/junimocup	Afrikaans	Delete

Sponsorship section allows adding sponsors to a give team
Adding sponsor ABC to team abc1(created preciously)



Sponsorships Management

+ Add Sponsorship

Add Sponsorship

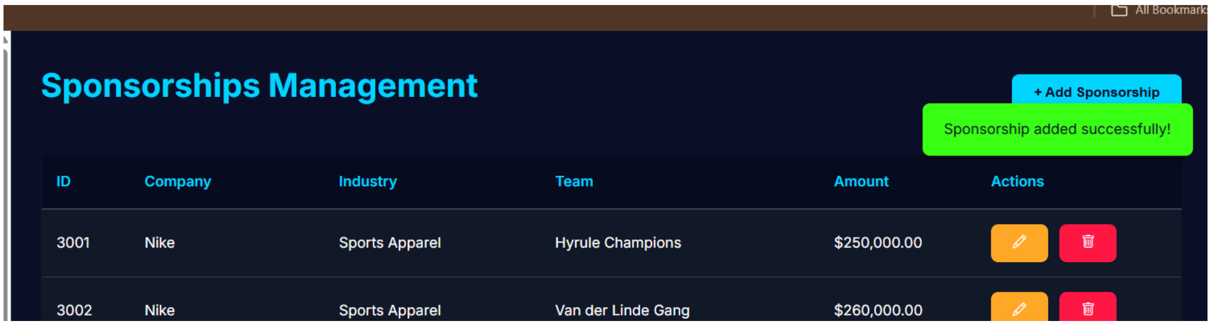
Company Name *
Sponsorship_abc

Industry *
ABC

Team *
abc1

Amount ($) *
123456

Add Sponsorship Cancel



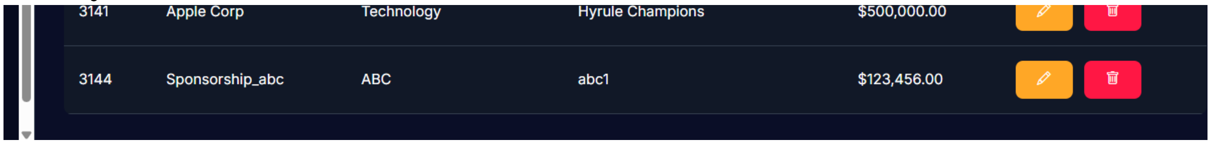
Sponsorships Management

+ Add Sponsorship

Sponsorship added successfully!

ID	Company	Industry	Team	Amount	Actions
3001	Nike	Sports Apparel	Hyrule Champions	$250,000.00	
3002	Nike	Sports Apparel	Van der Linde Gang	$260,000.00	

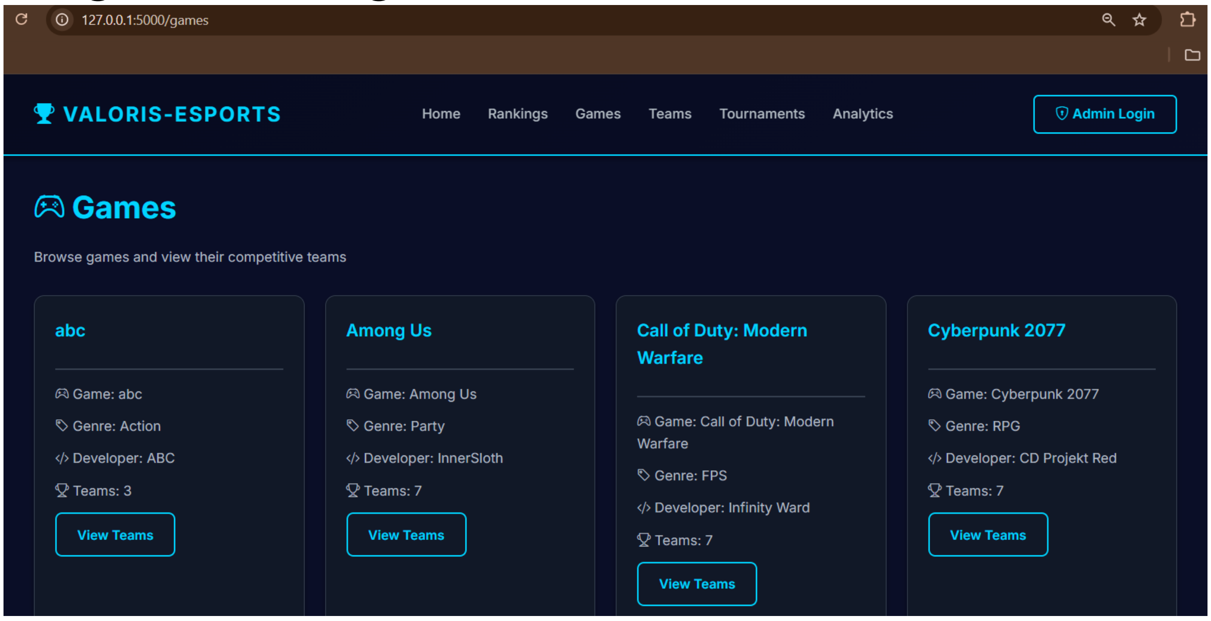
Entry in UI table



3141	Apple Corp	Technology	Hyrule Champions	$500,000.00	
3144	Sponsorship_abc	ABC	abc1	$123,456.00	

All the updates done are visible from public without admin login

Abc game visible in games section


127.0.0.1:5000/games

VALORIS-ESPORTS

Home Rankings Games Teams Tournaments Analytics

Admin Login

Games

Browse games and view their competitive teams

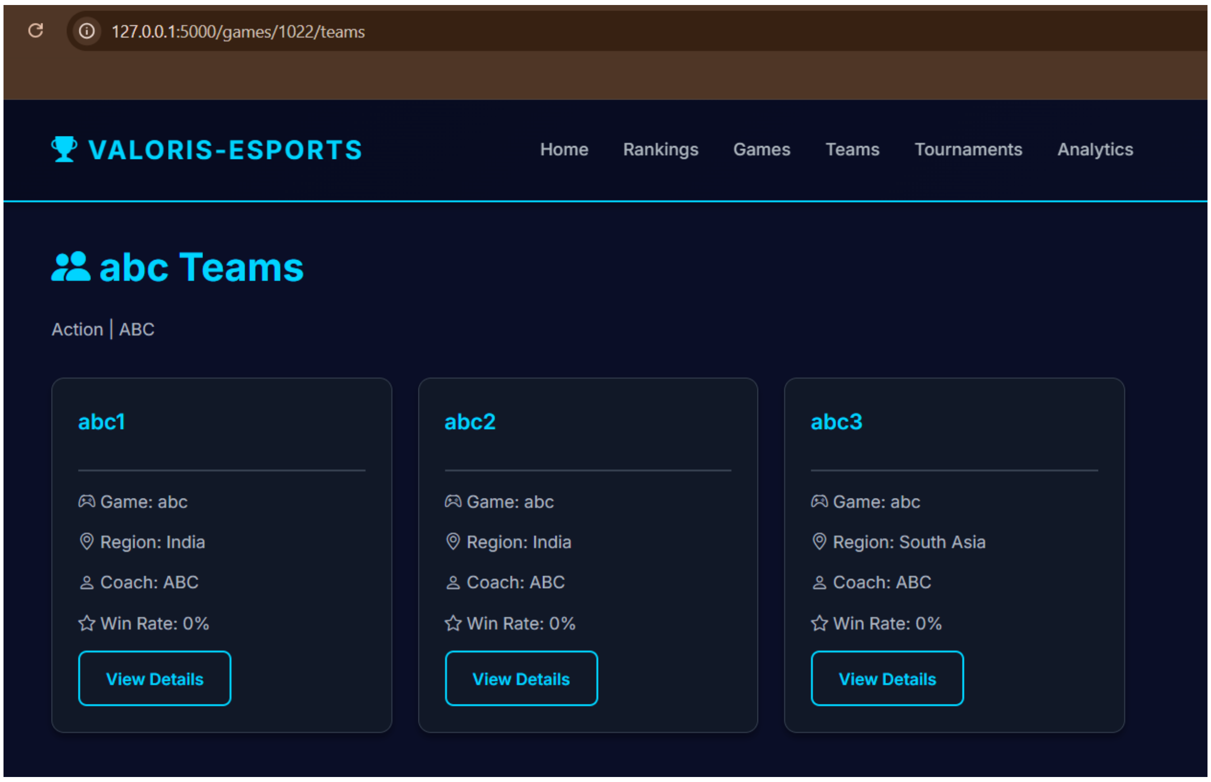
abc
Game: abc
Genre: Action
Developer: ABC
Teams: 3
View Teams

Among Us
Game: Among Us
Genre: Party
Developer: InnerSloth
Teams: 7
View Teams

Call of Duty: Modern Warfare
Game: Call of Duty: Modern Warfare
Genre: FPS
Developer: Infinity Ward
Teams: 7
View Teams

Cyberpunk 2077
Game: Cyberpunk 2077
Genre: RPG
Developer: CD Projekt Red
Teams: 7
View Teams

View teams button shows all teams in the respective game


127.0.0.1:5000/games/1022/teams

VALORIS-ESPORTS

Home Rankings Games Teams Tournaments Analytics

abc Teams

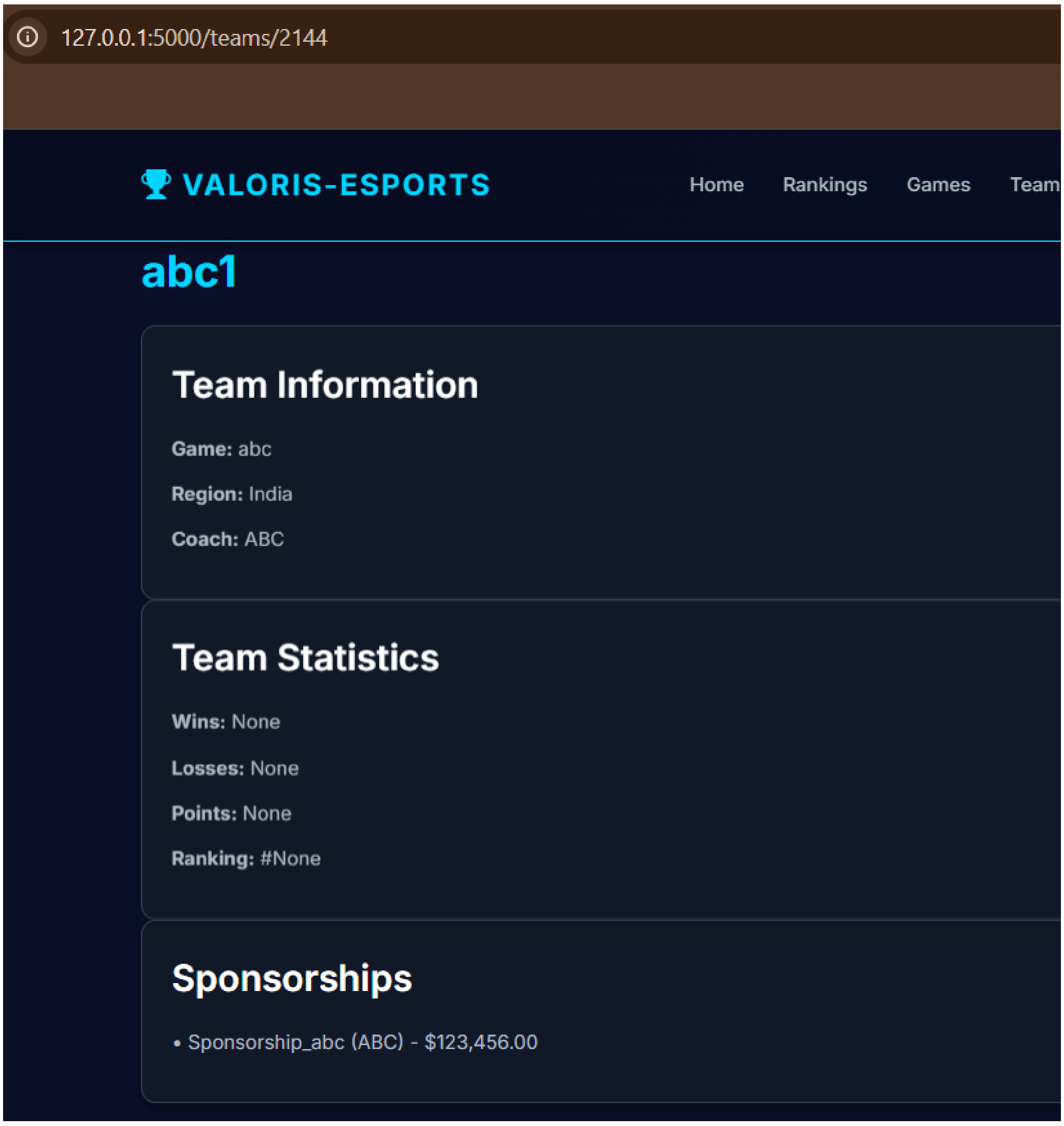
Action | ABC

abc1
Game: abc
Region: India
Coach: ABC
Win Rate: 0%
View Details

abc2
Game: abc
Region: India
Coach: ABC
Win Rate: 0%
View Details

abc3
Game: abc
Region: South Asia
Coach: ABC
Win Rate: 0%
View Details

Viewing details of team abc1


127.0.0.1:5000/teams/2144

abc1

Team Information

Game: abc

Region: India

Coach: ABC

Team Statistics

Wins: None

Losses: None

Points: None

Ranking: #None

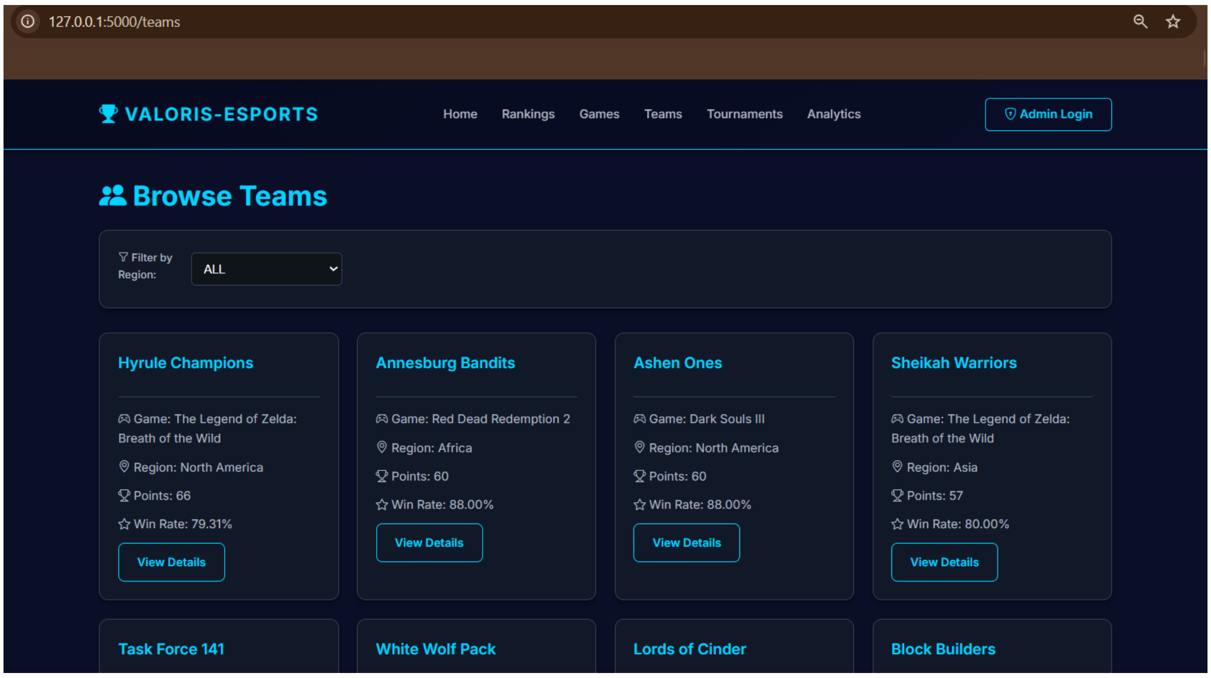
Sponsorships

- Sponsorship_abc (ABC) - $123,456.00

Sponsorships updated here

Public pages

Teams section



Browse Teams

Filter by Region: ALL

Hyrule Champions
Game: The Legend of Zelda: Breath of the Wild
Region: North America
Points: 66
Win Rate: 79.31%
View Details

Annesburg Bandits
Game: Red Dead Redemption 2
Region: Africa
Points: 60
Win Rate: 88.00%
View Details

Ashen Ones
Game: Dark Souls III
Region: North America
Points: 60
Win Rate: 88.00%
View Details

Sheikah Warriors
Game: The Legend of Zelda: Breath of the Wild
Region: Asia
Points: 57
Win Rate: 80.00%
View Details

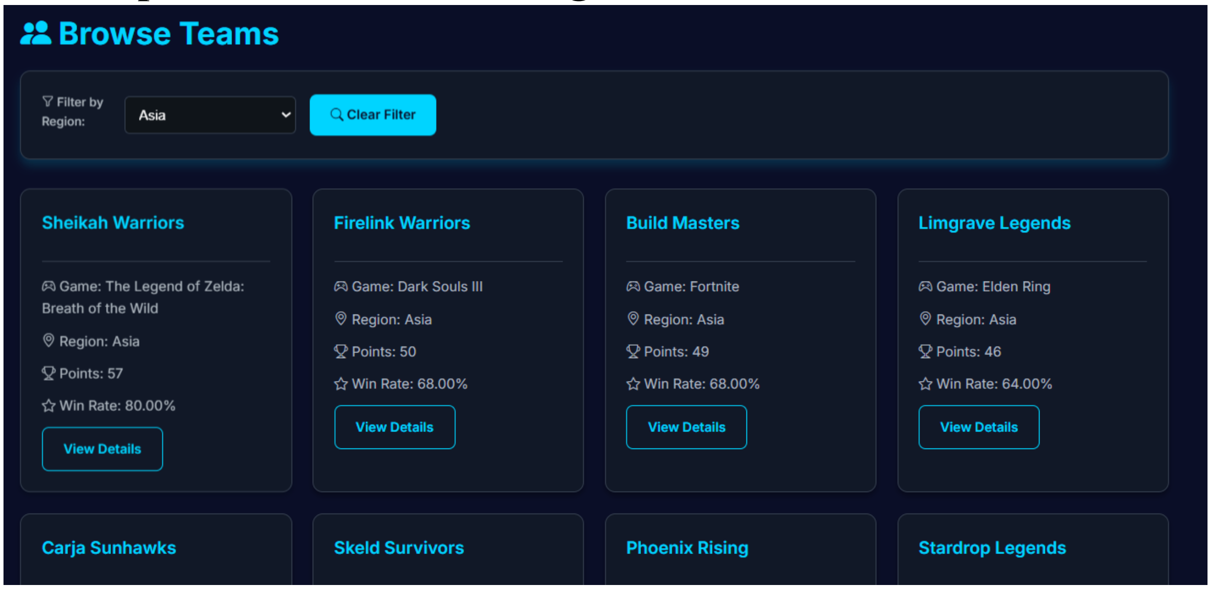
Task Force 141

White Wolf Pack

Lords of Cinder

Block Builders

Filter option filters based on region



Browse Teams

Filter by Region: Asia
Clear Filter

Sheikah Warriors
Game: The Legend of Zelda: Breath of the Wild
Region: Asia
Points: 57
Win Rate: 80.00%
View Details

Firelink Warriors
Game: Dark Souls III
Region: Asia
Points: 50
Win Rate: 68.00%
View Details

Build Masters
Game: Fortnite
Region: Asia
Points: 49
Win Rate: 68.00%
View Details

Limgrave Legends
Game: Elden Ring
Region: Asia
Points: 46
Win Rate: 64.00%
View Details

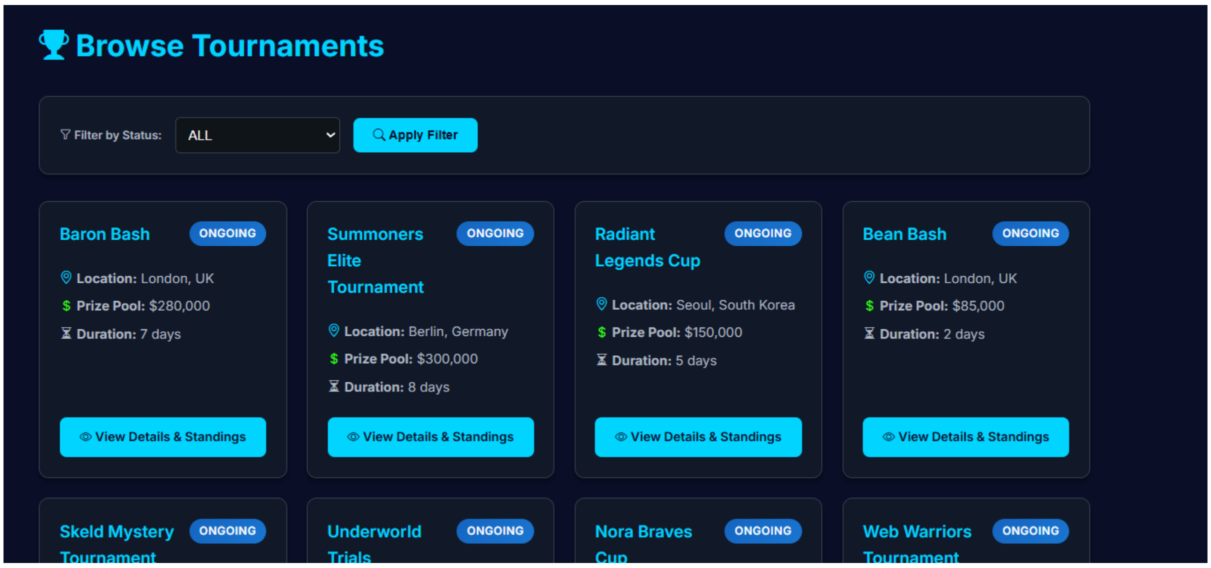
Carja Sunhawks

Skeld Survivors

Phoenix Rising

Stardrop Legends

Tournaments section



Browse Tournaments

Filter by Status: ALL
Apply Filter

Baron Bash ONGOING
Location: London, UK
Prize Pool: $280,000
Duration: 7 days
View Details & Standings

Summoners Elite Tournament ONGOING
Location: Berlin, Germany
Prize Pool: $300,000
Duration: 8 days
View Details & Standings

Radiant Legends Cup ONGOING
Location: Seoul, South Korea
Prize Pool: $150,000
Duration: 5 days
View Details & Standings

Bean Bash ONGOING
Location: London, UK
Prize Pool: $85,000
Duration: 2 days
View Details & Standings

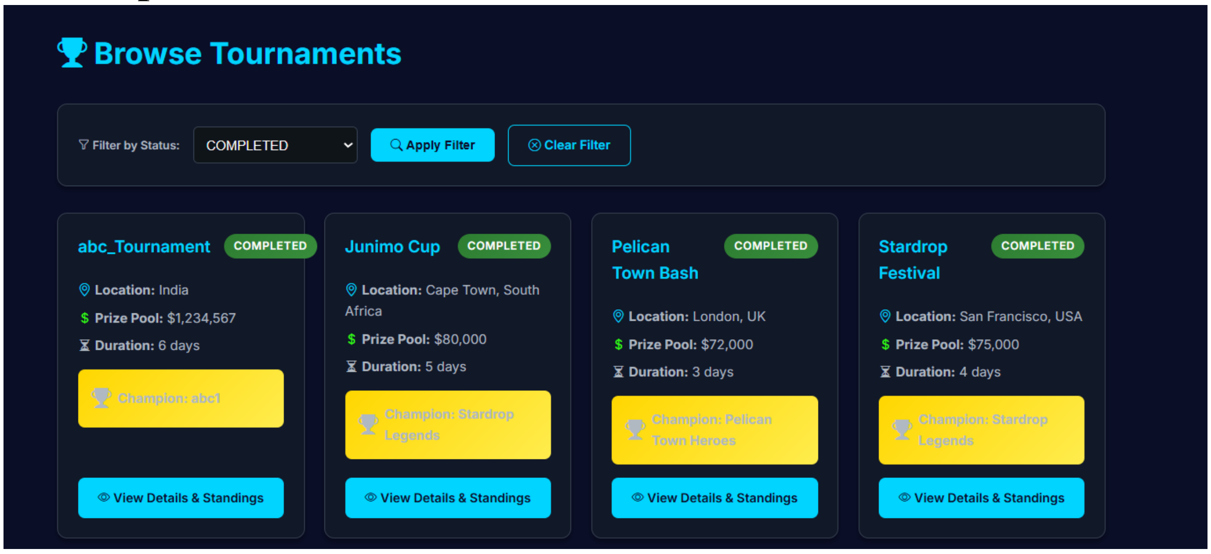
Skeld Mystery Tournament ONGOING

Underworld Trials ONGOING

Nora Braves Cup ONGOING

Web Warriors Tournament ONGOING

Filter option based on status of tournaments



Browse Tournaments

Filter by Status: COMPLETED
Apply Filter
Clear Filter

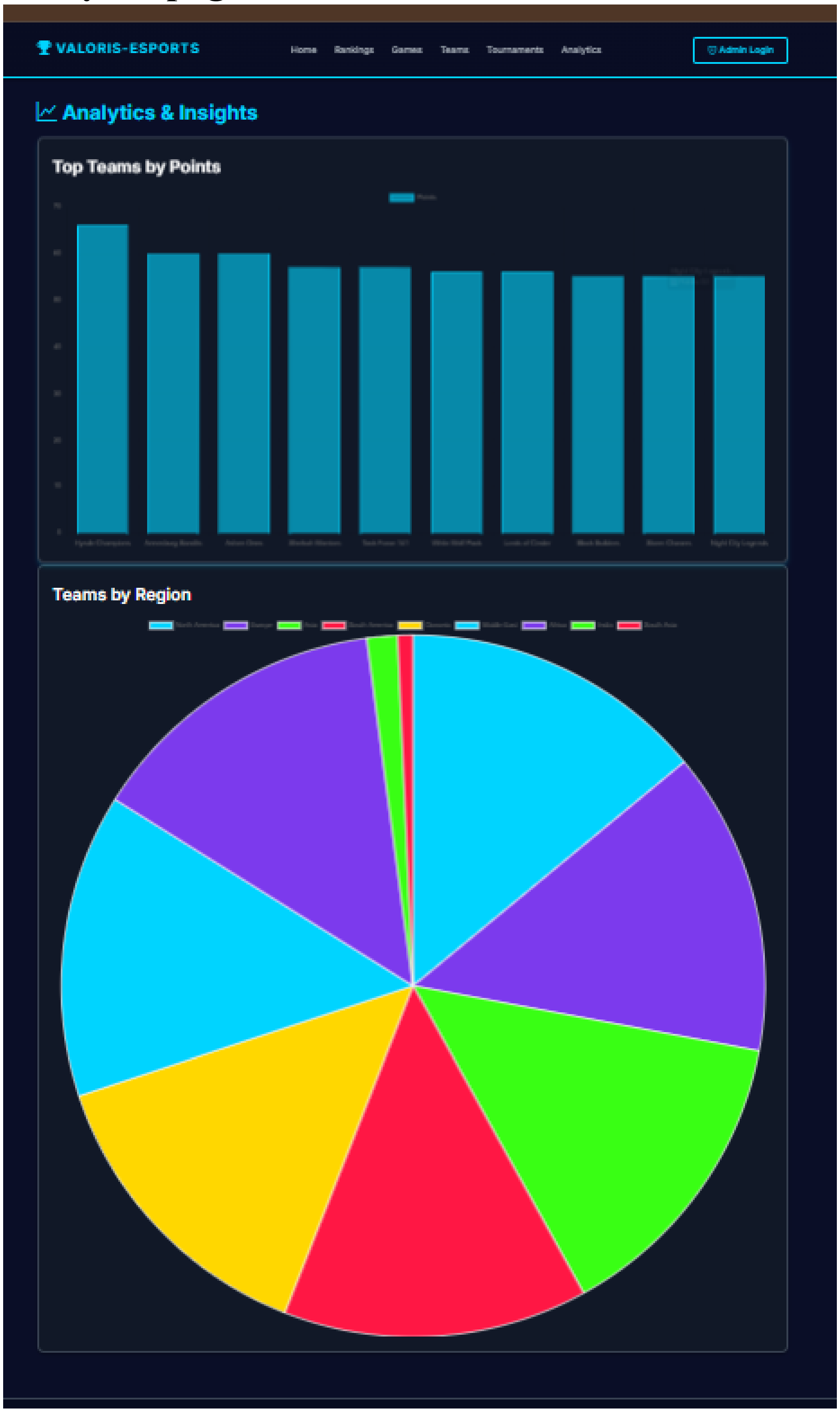
abc_Tournament COMPLETED
Location: India
Prize Pool: $1,234,567
Duration: 6 days
Champion: abc1
View Details & Standings

Junimo Cup COMPLETED
Location: Cape Town, South Africa
Prize Pool: $80,000
Duration: 5 days
Champion: Stardrop Legends
View Details & Standings

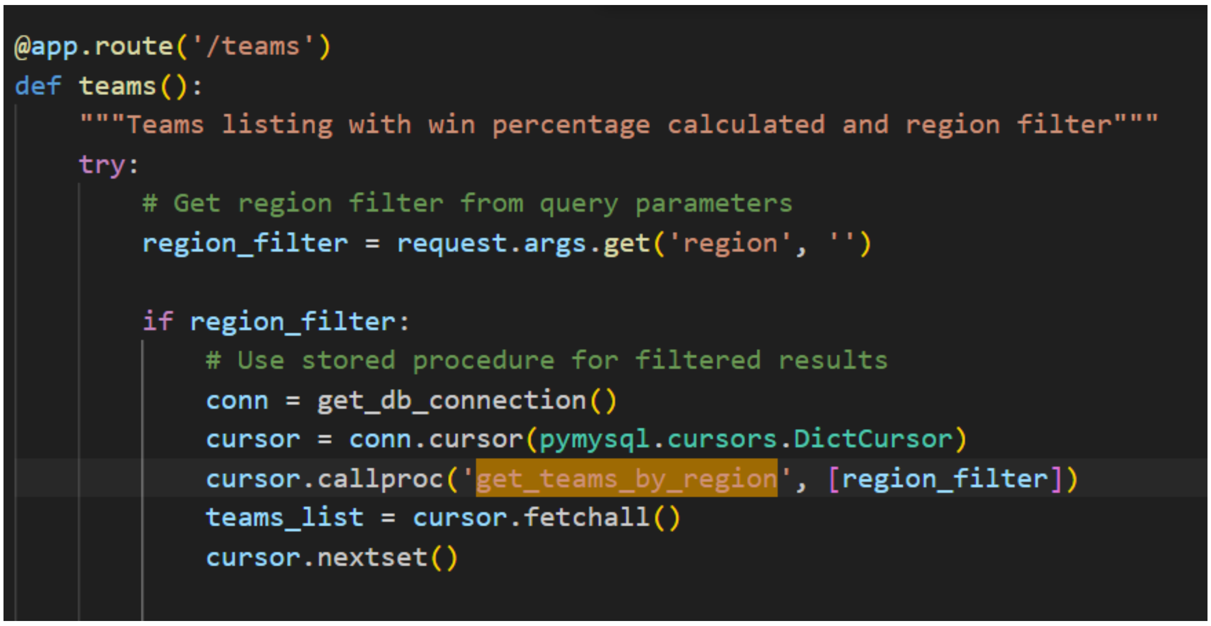
Pelican Town Bash COMPLETED
Location: London, UK
Prize Pool: $72,000
Duration: 3 days
Champion: Pelican Town Heroes
View Details & Standings

Stardrop Festival COMPLETED
Location: San Francisco, USA
Prize Pool: $75,000
Duration: 4 days
Champion: Stardrop Legends
View Details & Standings

Analytics page

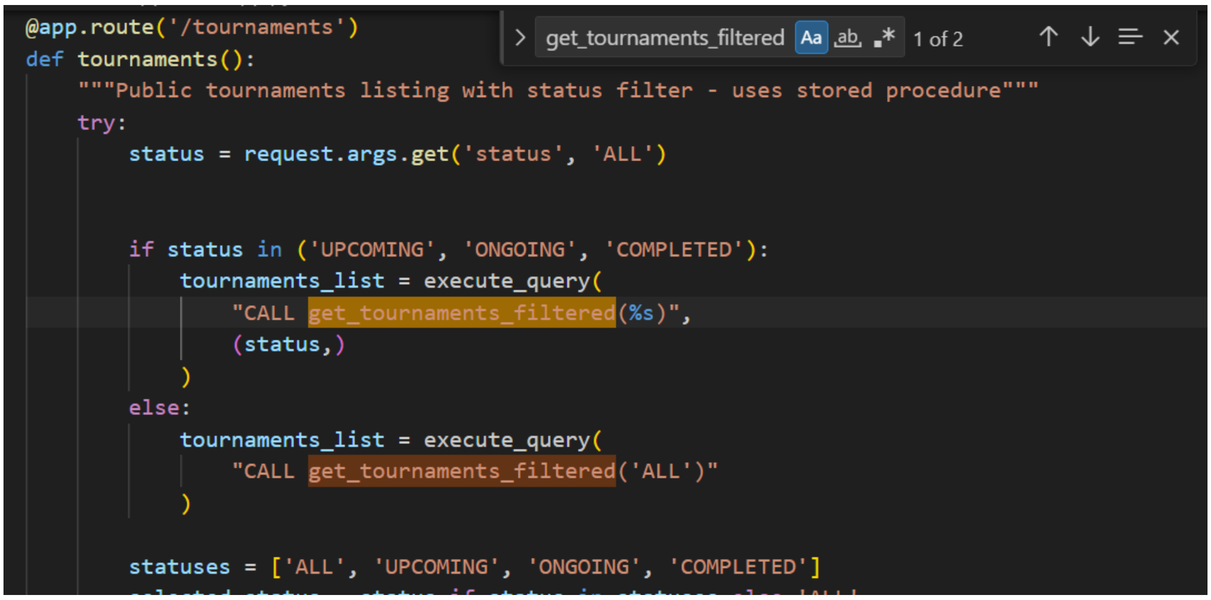


Code snippets for invoking the Procedures/Functions/Trigger



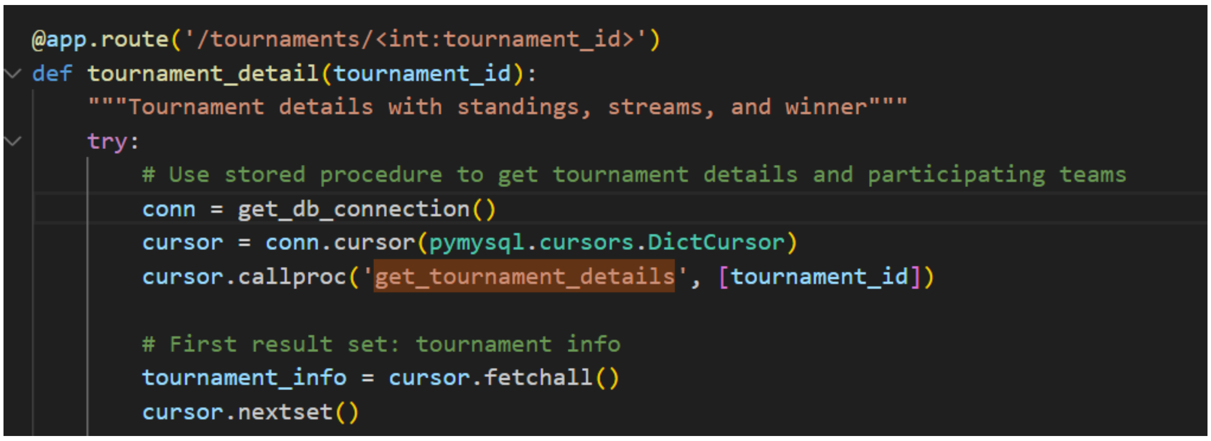
```python
@app.route('/teams')
def teams():
    """Teams listing with win percentage calculated and region filter"""
    try:
        # Get region filter from query parameters
        region_filter = request.args.get('region', '')

        if region_filter:
            # Use stored procedure for filtered results
            conn = get_db_connection()
            cursor = conn.cursor(pymysql.cursors.DictCursor)
            cursor.callproc('get_teams_by_region', [region_filter])
            teams_list = cursor.fetchall()
            cursor.nextset()
```

for filter option in the teams section (procedure)



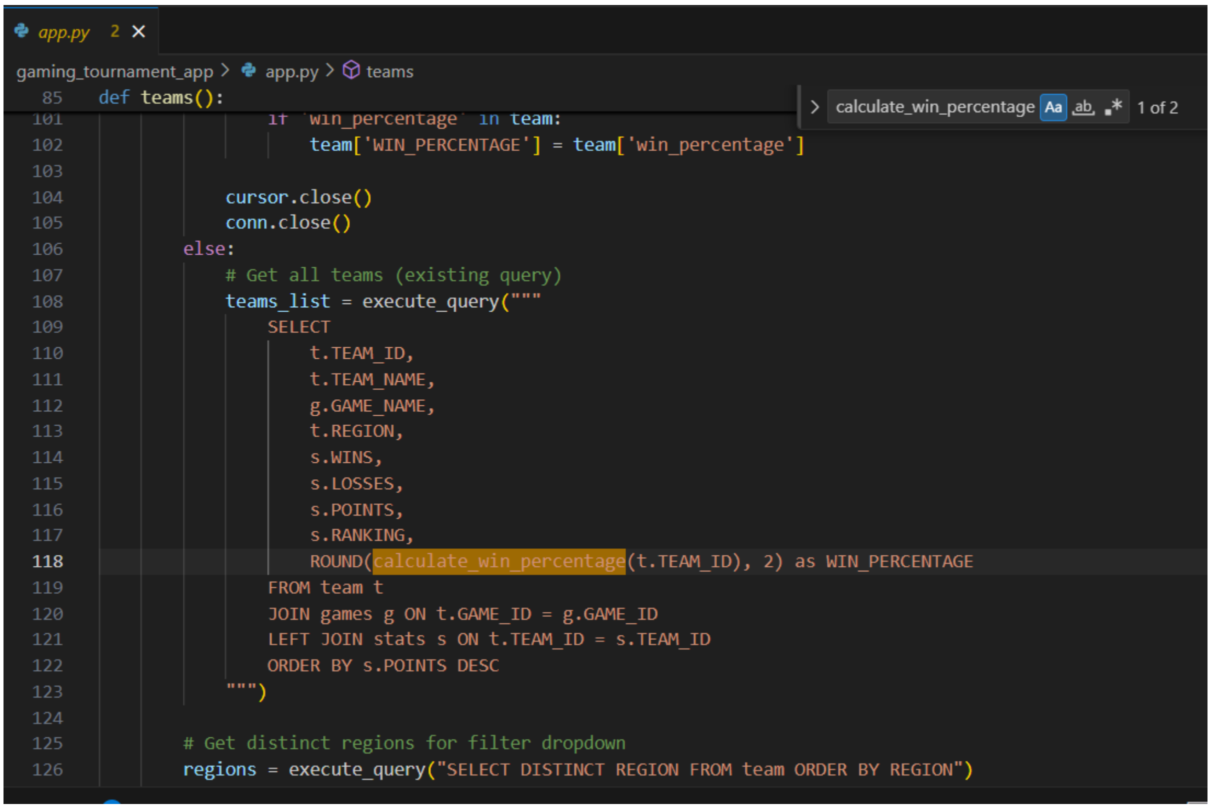
```python
@app.route('/tournaments')
def tournaments():
    """Public tournaments listing with status filter - uses stored procedure"""
    try:
        status = request.args.get('status', 'ALL')


        if status in ('UPCOMING', 'ONGOING', 'COMPLETED'):
            tournaments_list = execute_query(
                "CALL get_tournaments_filtered(%s)",
                (status,)
            )
        else:
            tournaments_list = execute_query(
                "CALL get_tournaments_filtered('ALL')"
            )

        statuses = ['ALL', 'UPCOMING', 'ONGOING', 'COMPLETED']
```

Handles status-based filtering of tournaments via procedure.



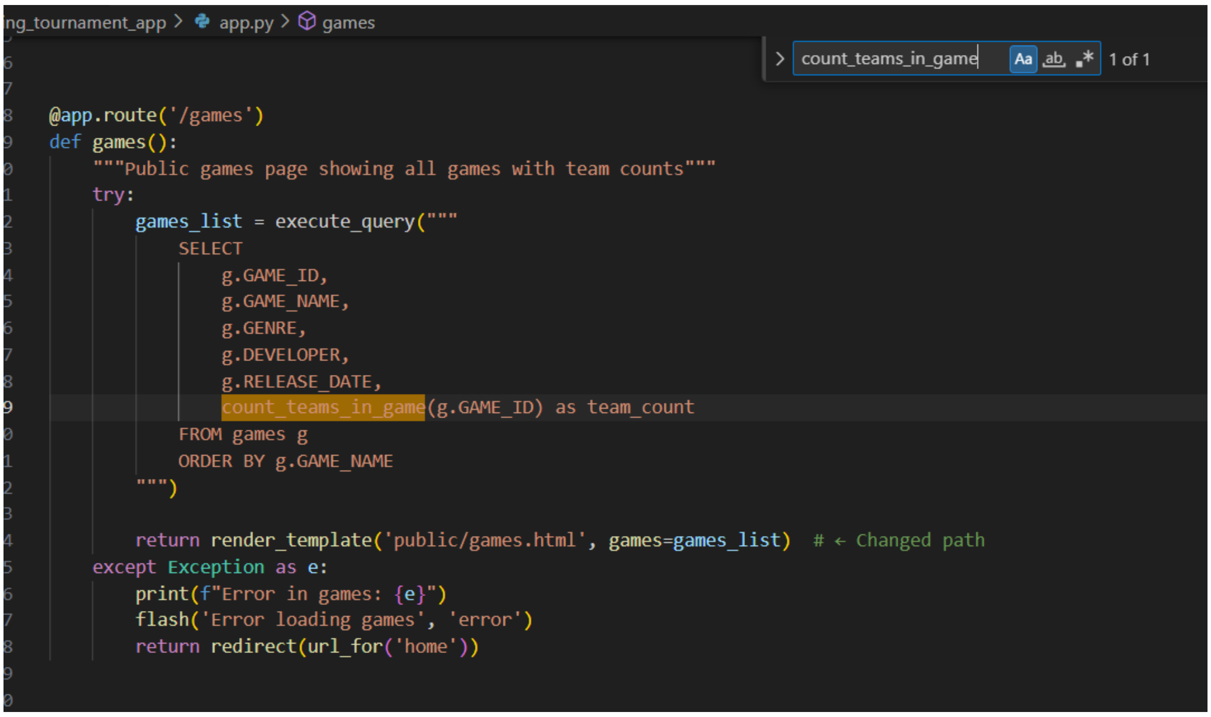
```python
@app.route('/tournaments/<int:tournament_id>')
def tournament_detail(tournament_id):
    """Tournament details with standings, streams, and winner"""
    try:
        # Use stored procedure to get tournament details and participating teams
        conn = get_db_connection()
        cursor = conn.cursor(pymysql.cursors.DictCursor)
        cursor.callproc('get_tournament_details', [tournament_id])

        # First result set: tournament info
        tournament_info = cursor.fetchall()
        cursor.nextset()
```

Functions



```python
def teams():
                if 'win_percentage' in team:
                    team['WIN_PERCENTAGE'] = team['win_percentage']

            cursor.close()
            conn.close()
        else:
            # Get all teams (existing query)
            teams_list = execute_query("""
                SELECT
                    t.TEAM_ID,
                    t.TEAM_NAME,
                    g.GAME_NAME,
                    t.REGION,
                    s.WINS,
                    s.LOSSES,
                    s.POINTS,
                    s.RANKING,
                    ROUND(calculate_win_percentage(t.TEAM_ID), 2) as WIN_PERCENTAGE
                FROM team t
                JOIN games g ON t.GAME_ID = g.GAME_ID
                LEFT JOIN stats s ON t.TEAM_ID = s.TEAM_ID
                ORDER BY s.POINTS DESC
            """)

        # Get distinct regions for filter dropdown
        regions = execute_query("SELECT DISTINCT REGION FROM team ORDER BY REGION")
```

Calculates win percentage for a team based on wins and matches played.

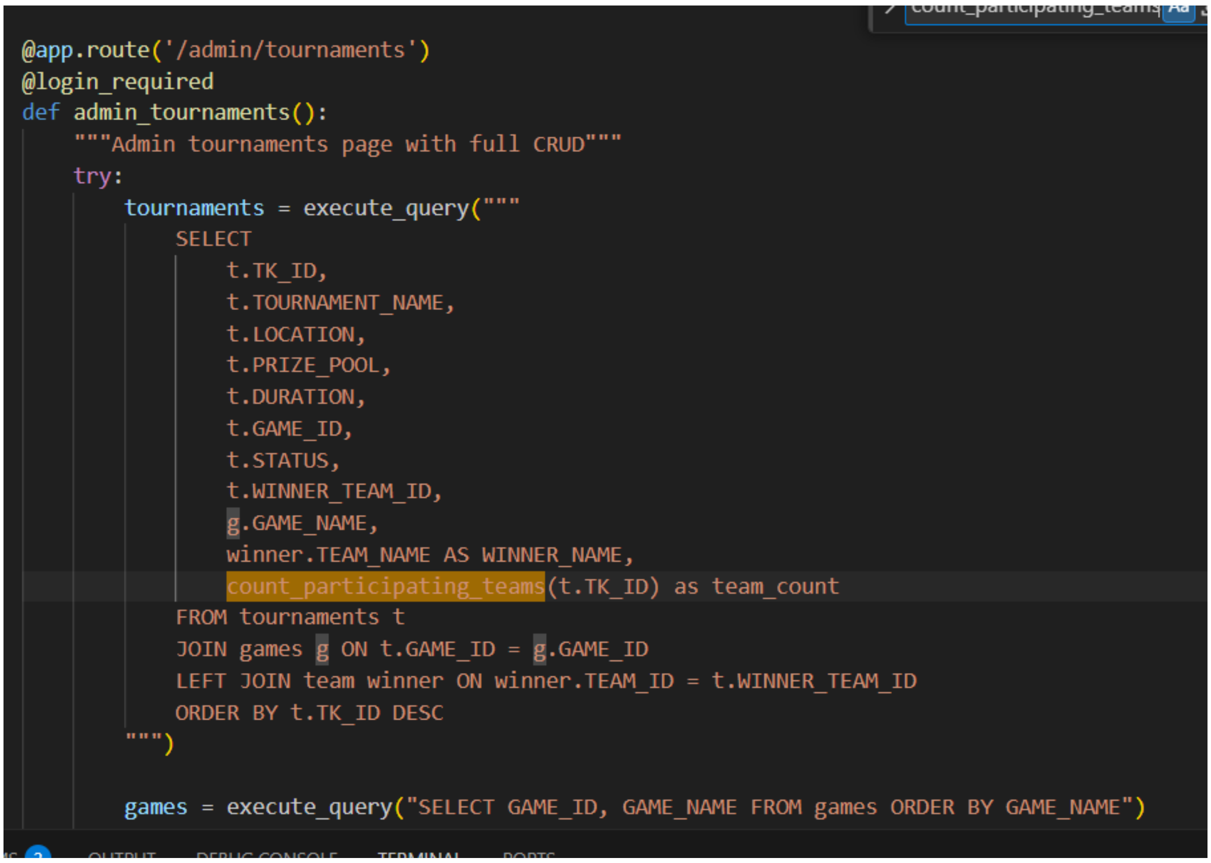


ng_tournament_app > app.py > games

count_teams_in_game Aa ab .* 1 of 1

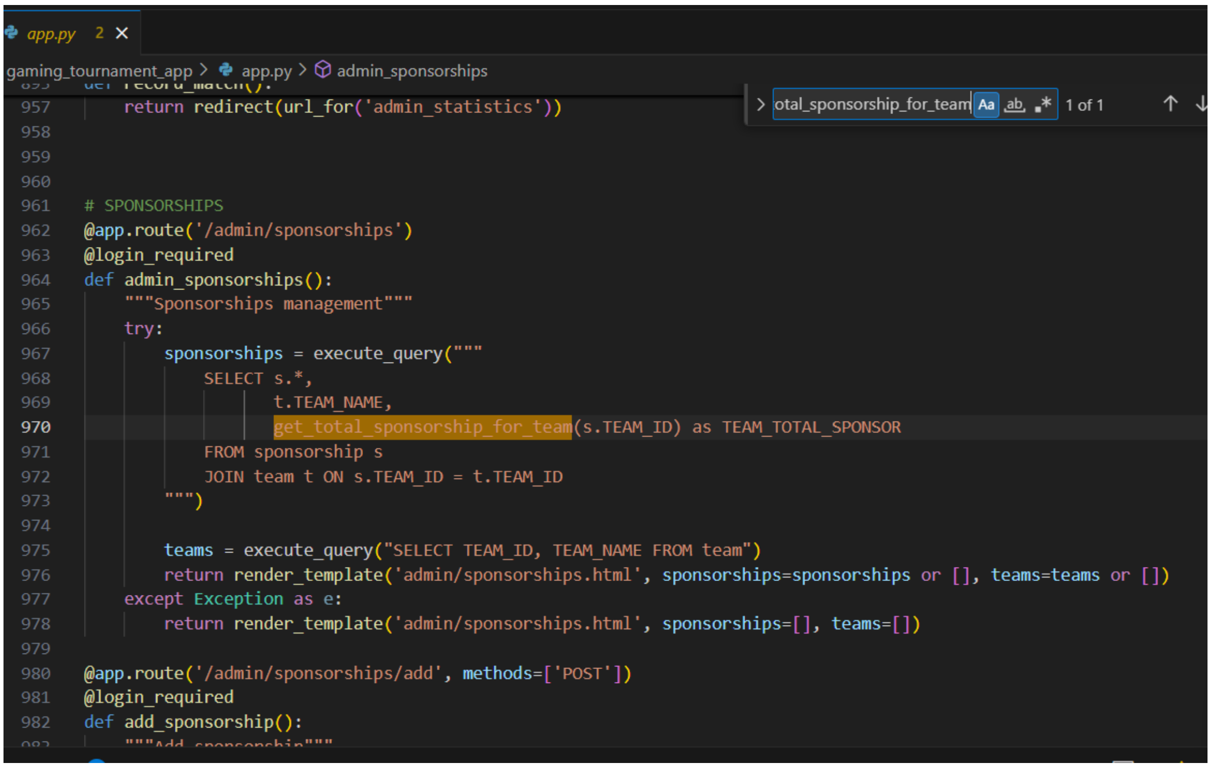
```python
@app.route('/games')
def games():
    """Public games page showing all games with team counts"""
    try:
        games_list = execute_query("""
            SELECT
                g.GAME_ID,
                g.GAME_NAME,
                g.GENRE,
                g.DEVELOPER,
                g.RELEASE_DATE,
                count_teams_in_game(g.GAME_ID) as team_count
            FROM games g
            ORDER BY g.GAME_NAME
        """)

        return render_template('public/games.html', games=games_list)  # ← Changed path
    except Exception as e:
        print(f"Error in games: {e}")
        flash('Error loading games', 'error')
        return redirect(url_for('home'))
```

Counts how many teams belong to a specific game.



```python
@app.route('/admin/tournaments')
@login_required
def admin_tournaments():
    """Admin tournaments page with full CRUD"""
    try:
        tournaments = execute_query("""
            SELECT
                t.TK_ID,
                t.TOURNAMENT_NAME,
                t.LOCATION,
                t.PRIZE_POOL,
                t.DURATION,
                t.GAME_ID,
                t.STATUS,
                t.WINNER_TEAM_ID,
                g.GAME_NAME,
                winner.TEAM_NAME AS WINNER_NAME,
                count_participating_teams(t.TK_ID) as team_count
            FROM tournaments t
            JOIN games g ON t.GAME_ID = g.GAME_ID
            LEFT JOIN team winner ON winner.TEAM_ID = t.WINNER_TEAM_ID
            ORDER BY t.TK_ID DESC
        """)

        games = execute_query("SELECT GAME_ID, GAME_NAME FROM games ORDER BY GAME_NAME")
```

Counts the number of teams participating in a specific tournament



app.py 2 ×

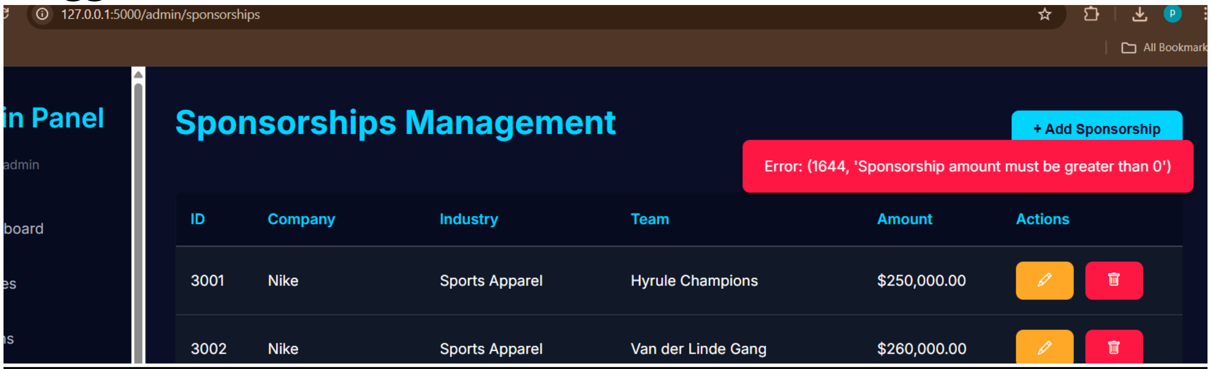
gaming_tournament_app > app.py > admin_sponsorships

otal_sponsorship_for_team Aa ab .* 1 of 1

```
    return redirect(url_for('admin_statistics'))



# SPONSORSHIPS
@app.route('/admin/sponsorships')
@login_required
def admin_sponsorships():
    """Sponsorships management"""
    try:
        sponsorships = execute_query("""
            SELECT s.*,
                   t.TEAM_NAME,
                   get_total_sponsorship_for_team(s.TEAM_ID) as TEAM_TOTAL_SPONSOR
            FROM sponsorship s
            JOIN team t ON s.TEAM_ID = t.TEAM_ID
        """)

        teams = execute_query("SELECT TEAM_ID, TEAM_NAME FROM team")
        return render_template('admin/sponsorships.html', sponsorships=sponsorships or [], teams=teams or [])
    except Exception as e:
        return render_template('admin/sponsorships.html', sponsorships=[], teams=[])

@app.route('/admin/sponsorships/add', methods=['POST'])
@login_required
def add_sponsorship():
```

Calculates total sponsorship amount for a team by summing sponsorship amounts.

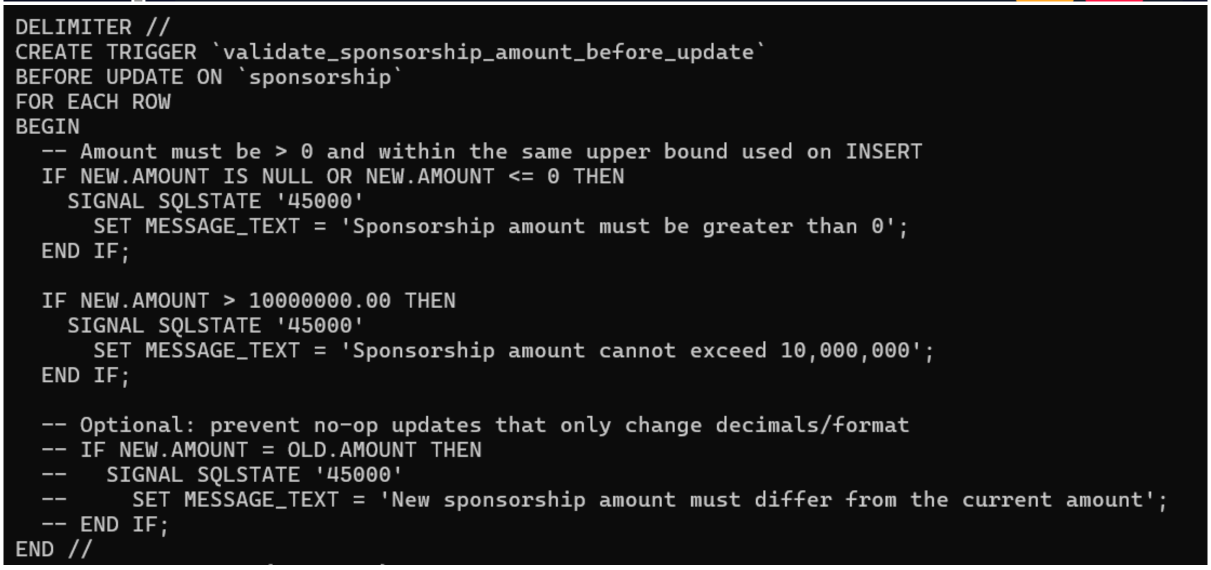
Triggers



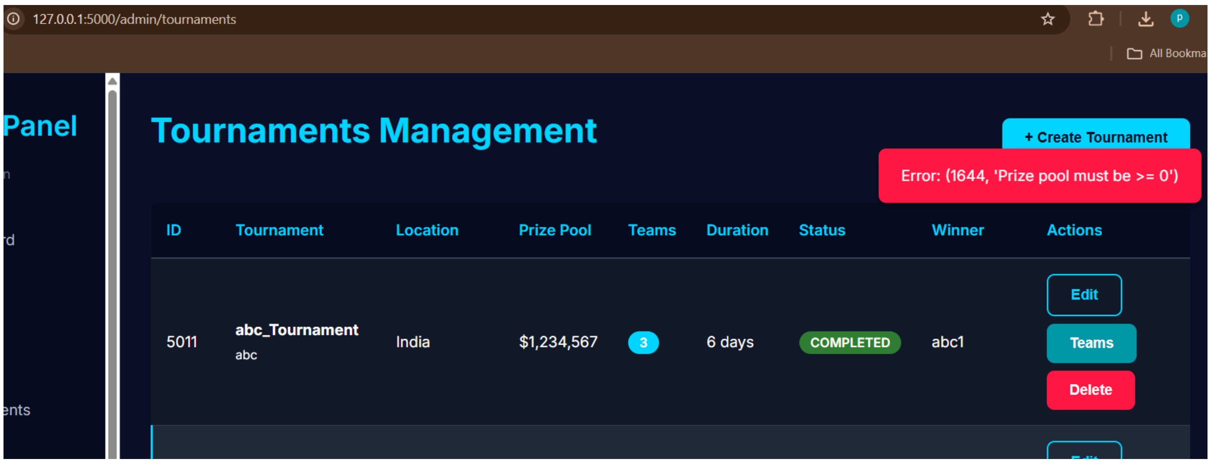
Sponsorships Management

Error: (1644, 'Sponsorship amount must be greater than 0')

ID	Company	Industry	Team	Amount	Actions
3001	Nike	Sports Apparel	Hyrule Champions	$250,000.00	
3002	Nike	Sports Apparel	Van der Linde Gang	$260,000.00	



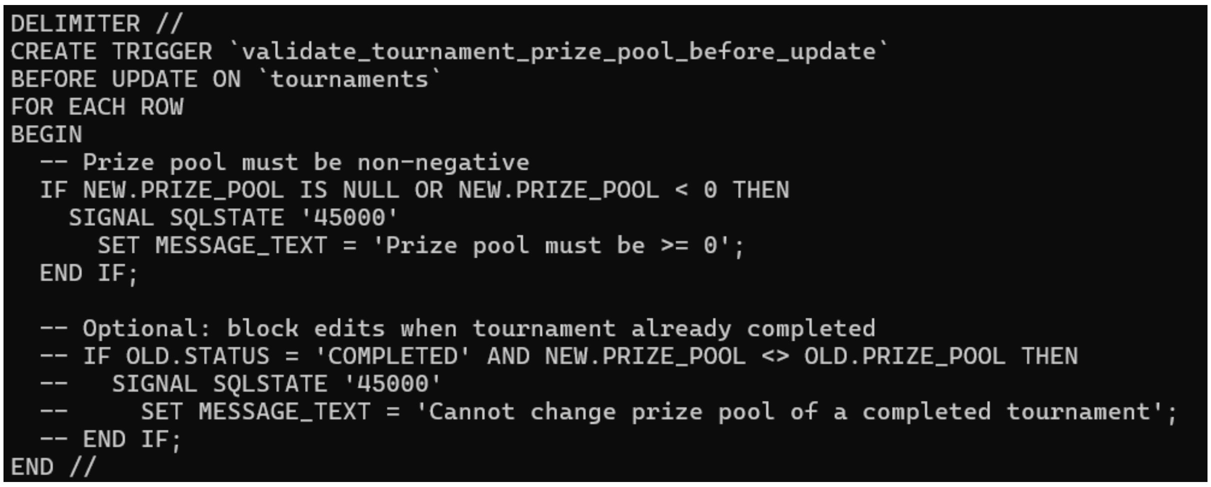
```
DELIMITER //
CREATE TRIGGER `validate_sponsorship_amount_before_update`
BEFORE UPDATE ON `sponsorship`
FOR EACH ROW
BEGIN
  -- Amount must be > 0 and within the same upper bound used on INSERT
  IF NEW.AMOUNT IS NULL OR NEW.AMOUNT <= 0 THEN
    SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'Sponsorship amount must be greater than 0';
  END IF;

  IF NEW.AMOUNT > 10000000.00 THEN
    SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'Sponsorship amount cannot exceed 10,000,000';
  END IF;

  -- Optional: prevent no-op updates that only change decimals/format
  -- IF NEW.AMOUNT = OLD.AMOUNT THEN
  --   SIGNAL SQLSTATE '45000'
  --     SET MESSAGE_TEXT = 'New sponsorship amount must differ from the current amount';
  -- END IF;
END //
```



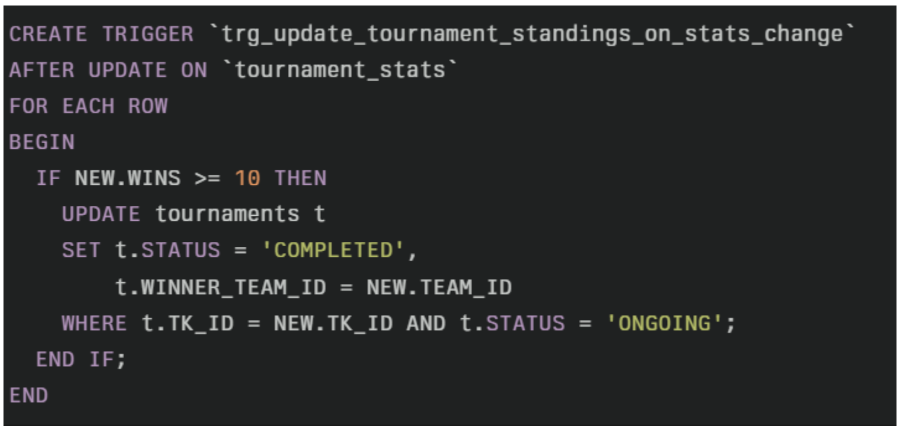
Tournaments Management

Error: (1644, 'Prize pool must be >= 0')

ID	Tournament	Location	Prize Pool	Teams	Duration	Status	Winner	Actions
5011	**abc_Tournament** abc	India	$1,234,567	3	6 days	COMPLETED	abc1	Edit Teams Delete

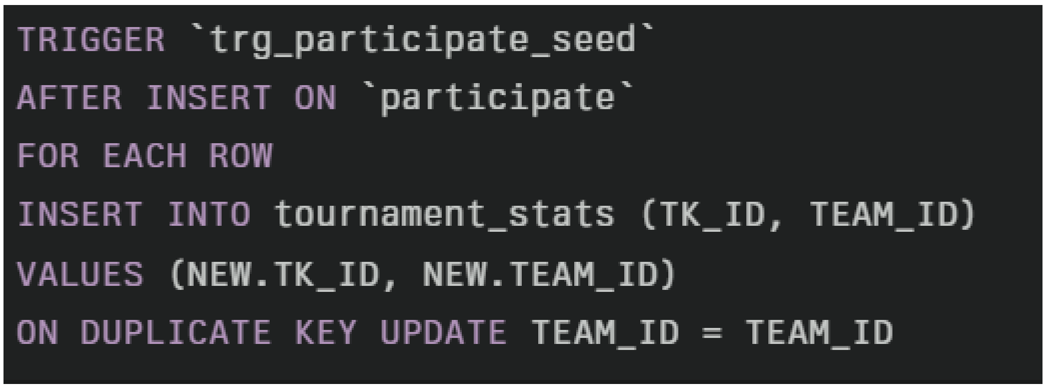


```sql
DELIMITER //
CREATE TRIGGER `validate_tournament_prize_pool_before_update`
BEFORE UPDATE ON `tournaments`
FOR EACH ROW
BEGIN
  -- Prize pool must be non-negative
  IF NEW.PRIZE_POOL IS NULL OR NEW.PRIZE_POOL < 0 THEN
    SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'Prize pool must be >= 0';
  END IF;

  -- Optional: block edits when tournament already completed
  -- IF OLD.STATUS = 'COMPLETED' AND NEW.PRIZE_POOL <> OLD.PRIZE_POOL THEN
  --   SIGNAL SQLSTATE '45000'
  --     SET MESSAGE_TEXT = 'Cannot change prize pool of a completed tournament';
  -- END IF;
END //
```



```sql
CREATE TRIGGER `trg_update_tournament_standings_on_stats_change`
AFTER UPDATE ON `tournament_stats`
FOR EACH ROW
BEGIN
  IF NEW.WINS >= 10 THEN
    UPDATE tournaments t
    SET t.STATUS = 'COMPLETED',
        t.WINNER_TEAM_ID = NEW.TEAM_ID
    WHERE t.TK_ID = NEW.TK_ID AND t.STATUS = 'ONGOING';
  END IF;
END
```

Automatically marks tournament as COMPLETED and sets winner when a team reaches 10 wins



```sql
TRIGGER `trg_participate_seed`
AFTER INSERT ON `participate`
FOR EACH ROW
INSERT INTO tournament_stats (TK_ID, TEAM_ID)
VALUES (NEW.TK_ID, NEW.TEAM_ID)
ON DUPLICATE KEY UPDATE TEAM_ID = TEAM_ID
```

Automatically creates tournament stats entry when a team joins a tournament