CS 353 Database Systems

Final Report

Group 1 - Section 3

AlphaBet

Social Betting Platform

Mert Aslan 21702818

Rahmiye Büşra Büyükgebiz 21702019

Ozan Aydın 21702957

Yüce Hasan Kılıç 21704070

1. Project Description

AlphaBet is a web based social betting application that allows users to see upcoming matches from different branches, place their bets, connect with their friends and follow editors. The system provides database information on upcoming matches with their individual bets, allows users to search for matches and editors to suggest or share their bet slips, and includes a market page where users can buy items with their Alpha coins. Users can also interact with other user’s bet slips by commenting on them or liking them, and can earn achievements for doing certain tasks.

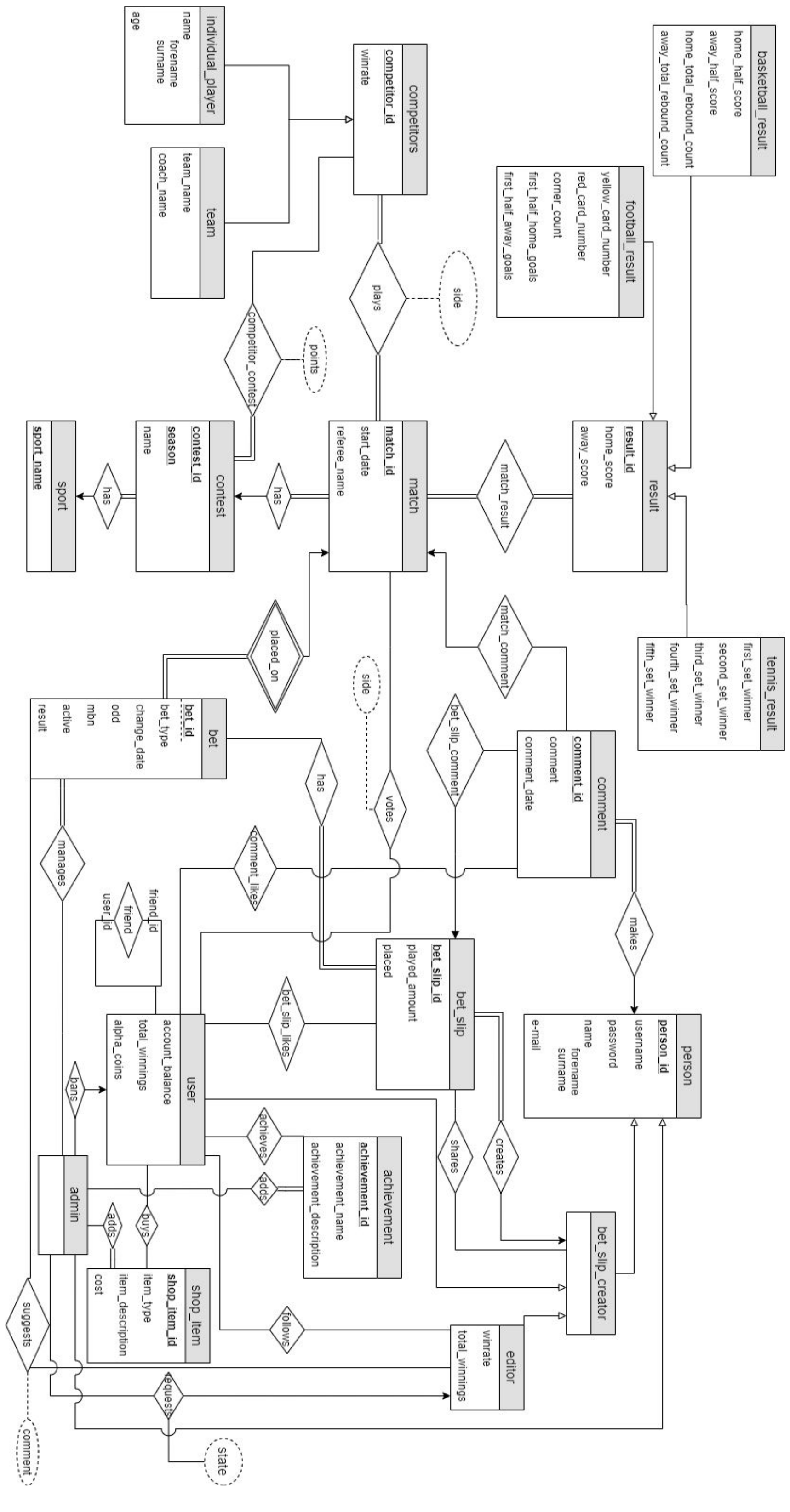
The application includes three types of users, admin, editor and standard user. The application’s UI and capabilities change according to each user, and each type of user has their own information stored in the database. Users can register as standard users or editors, however logging in as an editor requires an admin confirmation.

The first type of user, admin, is responsible for managing people, matches, market items and achievements. Logging in as admin will lead to the admin dashboard, which other users don’t have access to. From there, admin can remove or update bets, manage users and market items.

When a person registers as a standard user, they will be ready to use the application immediately after signing in. From the home page, they can use the match filter by typing a keyword and/or selecting a maximum MBN, a specific sport and a contest that is relevant to that sport, in order to search for matches. Users can then add bets from those matches to their betslips, and if MBN is satisfied and the user has enough money, they can play that bet slip. Played bet slips will appear under the pending bet slips session in the profile page, where the users can see their pending and concluded betslips, add or withdraw cash, see and add friends and view their achievements. Once a user plays a betslip, that betslip will be visible to that user’s friends in the feed page. Friends of that user can view, comment on and like their bet slip. In the editor's page, users can see registered and approved editors and can follow them in order to see their shared bet slips, single bet suggestions and their overall performance. In the market page, users can view and buy items if the price is less than their Alpha Coins.

Editors can also filter matches and place bets on their bet slips, but instead of playing them, they can share them with their followers. Editors can also suggest single match picks with a description

2. Final E/R


basketball_result
home_half_score
away_half_score
home_total_rebound_count
away_total_rebound_count
individual_player
name
forename
surname
age
competitors
competitor_id
winrate
team
team_name
coach_name
football_result
yellow_card_number
red_card_number
corner_count
first_half_home_goals
first_half_away_goals
plays
side
competitor_contest
points
sport
sport_name
has
contest
contest_id
season
name
has
match
match_id
start_date
referee_name
match_result
result
result_id
home_score
away_score
tennis_result
first_set_winner
second_set_winner
third_set_winner
fourth_set_winner
fifth_set_winner
placed_on
match_comment
bet
bet_id
bet_type
change_date
odd
mbn
active
result
side
has
votes
bet_slip_comment
comment
comment_id
comment
comment_date
manages
comment_likes
friend
friend_id
user_id
makes
bet_slip
bet_slip_id
played_amount
placed
bet_slip_likes
person
person_id
username
password
name
forename
surname
e-mail
user
account_balance
total_winnings
alpha_coins
bans
achieves
achievement
achievement_id
achievement_name
achievement_description
shares
creates
admin
adds
buys
adds
shop_item
shop_item_id
item_type
item_description
cost
bet_slip_creator
follows
suggests
editor
winrate
total_winnings
requests
comment
state

3. Final List of Tables

3.1.Person

Relational Model: person(person_id, username, password, forename, surname, e-mail)

PRIMARY KEY(person_id)

3.2.User

Relational Model: user(user_id, account_balance, total_winnings, alpha_coins)

PRIMARY KEY(user_id),

FOREIGN KEY(user_id) REFERENCES bet_slip_creator(creator_id)

3.3.User Friend

Relational Model: user_friend(user_id, friend_id)

PRIMARY KEY(user_id, friend_id),

FOREIGN KEY(user_id) REFERENCES user(user_id),

FOREIGN KEY(friend_id) REFERENCES user(user_id)

3.4.Editor

Relational Model: editor(editor_id, winrate, total_winnings)

PRIMARY KEY(editor_id),

FOREIGN KEY(editor_id) REFERENCES bet_slip_creator(creator_id)

3.5.User Follows

Relational Model: user_follows(editor_id, user_id)

PRIMARY KEY(editor_id, user_id),

FOREIGN KEY (editor_id) REFERENCES editor(editor_id),

FOREIGN KEY (user_id) REFERENCES user(user_id)

3.6.Admin

Relational Model: admin(admin_id)

PRIMARY KEY(admin_id)

FOREIGN KEY (admin_id) REFERENCES person(person_id)

3.7.Bet

Relational Model:

bet(bet_id, match_id, bet_type, change_date, odd, mbn, result, active)

PRIMARY KEY(bet_id, match_id),

FOREIGN KEY (match_id) REFERENCES match(match_id)

3.8.Bet Slip

Relational Model: bet_slip(bet_slip_id, creator_id, placed, played_amount)

PRIMARY KEY(bet_slip_id)

FOREIGN KEY(creator_id) REFERENCES bet_slip_creator(creator_id)

3.9.Shared Bet Slip

Relational Model: shared_bet_slip(bet_slip_id, sharer_id)

PRIMARY KEY(bet_slip_id, sharer_id)

FOREIGN KEY(bet_slip_id) REFERENCES bet_slip(bet_slip_id),

FOREIGN KEY(sharer_id) REFERENCES bet_slip_creator(creator_id)

3.10.Suggested Bet

Relational Model: suggested_bet(editor_id, bet_id, match_id, comment)

PRIMARY KEY(editor_id, bet_id),

FOREIGN KEY(editor_id) REFERENCES editor(editor_id),

FOREIGN KEY(bet_id, match_id) REFERENCES bet(bet_id, match_id)

3.11.Bet Slip Like

Relational Model: bet_slip_like(user_id, bet_slip_id)

PRIMARY KEY(user_id, bet_slip_id),

FOREIGN KEY(user_id) REFERENCES user(user_id)

FOREIGN KEY(bet_slip_id) REFERENCES bet_slip(bet_slip_id)

3.12.Included Bet

Relational Model: included_bet(bet_slip_id, match_id, bet_id)

PRIMARY KEY(bet_slip_id, match_id),

FOREIGN KEY(bet_slip_id) REFERENCES bet_slip(bet_slip_id),

FOREIGN KEY(bet_id, match_id) REFERENCES bet(bet_id, match_id), FOREIGN KEY(match_id) REFERENCES match(match_id)

3.13.Bet SlipCreator

Relational Model: bet_slip_creator(creator_id)

PRIMARY KEY(creator_id),

FOREIGN KEY(creator_id) REFERENCES person(person_id)

3.14.Sport

Relational Model: sport(sport_name)

PRIMARY KEY(sport_name)

3.15.Contest

Relational Model: contest(contest_id, season, sport_name, name)

PRIMARY KEY(contest_id, season),

FOREIGN KEY(sport_name) REFERENCES sport(sport_name)

3.16.Match

Relational Model: match(match_id, start_date, contest_id, season, sport_name referee_name)

PRIMARY KEY(match_id),

FOREIGN KEY(contest_id, season) REFERENCES contest(contest_id, season),

FOREIGN KEY(sport_name) REFERENCES sport(sport_name)

3.17.Competitor

Relational Model: competitor(competitor_id, winrate)

PRIMARY KEY(competitor_id)

‘

3.18.Competitor Match

Relational Model: competitor_match(competitor_id, match_id, side)

PRIMARY KEY (competitor_id, match_id),

FOREIGN KEY (competitor_id) REFERENCES competitor(competitor_id),

FOREIGN KEY (match_id) REFERENCES match(match_id)

3.19.Competitor Contest

Relational Model: competitor_contest(competitor_id, contest_id, season, points)

PRIMARY KEY(competitor_id, contest_id, season),

FOREIGN KEY (competitor_id) REFERENCES competitor(competitor_id),

FOREIGN KEY(contest_id, season) REFERENCES contest(contest_id, season)

3.20.Individual Player

Relational Model: individual_player(player_id, forename, surname, age)

PRIMARY KEY(player_id),

FOREIGN KEY(player_id) REFERENCES competitor(competitor_id)

3.21.Team

Relational Model: team(team_id, team_name, coach_name)

PRIMARY KEY(team_id),

FOREIGN KEY(team_id) REFERENCES competitor(competitor_id)

3.22.Result

Relational Model: result(result_id, match_id, home_score, away_score)

PRIMARY KEY(result_id, match_id),

FOREIGN KEY(match_id) REFERENCES match(match_id)

3.23.Basketball Result

Relational Model: basketball_result(result_id, home_half_score, away_half_score, home_total_rebound_count, away_total_rebound_count)

PRIMARY KEY (result_id),

FOREIGN KEY (result_id) REFERENCES result(result_id)

3.24.Football Result

Relational Model: football_result(result_id, yellow_card_number, red_card_number, corner_count, first_half_home_goals, first_half_away_goals)

PRIMARY KEY(result_id),

FOREIGN KEY(result_id) REFERENCES result(result_id)

3.25.Tennis Result

Relational Model: tennis_result(result_id, first_set_winner, second_set_winner)

PRIMARY KEY(result_id),

FOREIGN KEY(result_id) REFERENCES result(result_id)

3.26.Comment

Relational Model: comment(comment_id, person_id, comment, comment_date)

PRIMARY KEY(comment_id),

FOREIGN KEY(person_id) REFERENCES person(person_id)

3.27.Bet Slip Comment

Relational Model: bet_slip_comment(comment_id, bet_slip_id)

PRIMARY KEY(comment_id, bet_slip_id),

FOREIGN KEY(comment_id) REFERENCES comment(comment_id),

FOREIGN KEY(bet_slip_id) REFERENCES bet_slip(bet_slip_id)

3.28.Match Comment

Relational Model: match_comment(match_id, comment_id)

PRIMARY KEY(match_id, comment_id),

FOREIGN KEY(match_id) REFERENCES match(match_id)

FOREIGN KEY(comment_id) REFERENCES comment(comment_id)

3.29.Comment Likes

Relational Model: comment_likes(comment_id, user_id)

PRIMARY KEY(comment_id, user_id),

FOREIGN KEY(user_id) REFERENCES user(user_id,

FOREIGN KEY(comment_id) REFERENCES comment(comment_id)

3.30.Votes

Relational Model: votes(user_id, match_id, side)

PRIMARY KEY(user_id, match_id),

FOREIGN KEY(user_id) REFERENCES user(user_id),

FOREIGN KEY(match_id) REFERENCES match(match_id)

3.31.Manages

Relational Model: manages(admin_id, bet_id, match_id)

PRIMARY KEY(admin_id, bet_id, match_id),

FOREIGN KEY(admin_id) REFERENCES admin(admin_id),

FOREIGN KEY(bet_id, match_id) REFERENCES bet(bet_id, match_id)

3.32. Approves

Relational Model: approves(editor_id, state)

PRIMARY KEY(editor_id),

FOREIGN KEY(editor_id) REFERENCES editor(editor_id)

3.33.Shop Item

Relational Model: shop_item(shop_item_id, item_type, item_description, cost)

PRIMARY KEY(shop_item_id, item_type)

3.34.Bought Item

Relational Model: bought_item(shop_item_id, user_id, item_type)

PRIMARY KEY(shop_item_id, user_id, item_type),

FOREIGN KEY(shop_item_id, item_type) REFERENCES shop_item(shop_item_id, item_type),

FOREIGN KEY(user_id) REFERENCES user(user_id)

3.35. Added Item

Relational Model: added_item(shop_item_id, item_type, admin_id)

PRIMARY KEY (shop_item_id, admin_id, item_type),

FOREIGN KEY(shop_item_id, item_type) REFERENCES shop_item(shop_item_id, item_type),,

FOREIGN KEY(admin_id) REFERENCES admin(admin_id)

3.36. Achievement

Relational Model: achievement(achievement_id, achievement_name, achievement_description)

PRIMARY KEY (achievement_id)

3.37. Gained Achievement

Relational Model: gained_achievement(achievement_id, user_id)

PRIMARY KEY(achievement_id, user_id),

FOREIGN KEY(achievement_id) REFERENCES achievement(achievement_id)

FOREIGN KEY(user_id) REFERENCES user(user_id)

3.38. Added Achievement

Relational Model: added_achievement(admin_id, achievement_id)

PRIMARY KEY(admin_id, achievement_id),
FOREIGN KEY(admin_id) REFERENCES admin(admin_id),
FOREIGN KEY(achievement_id) REFERENCES achievement(achievement_id)

3.39. Bans

Relational Model: bans(user_id, admin_id)
PRIMARY KEY(user_id, admin_id),
FOREIGN KEY(user_id) REFERENCES user(user_id),
FOREIGN KEY(admin_id) REFERENCES admin(admin_id)

4. Implementation Details

MySQL was used for the database and DataGrip was used for the database management environment. In order to automate database creation and populate the database, we first created SQL files in which we create our schemas. We also used the Flyway framework to remodel the application's database schema easily when there is something that needs to be modified. On each modification, we updated our database from one version to next using migrations via Flyway.

For the user interface, we used Javascript and ReactJS framework. In order to communicate with the backend through HTTP requests, we used axios library which is a Javascript library and it can perform automatic JSON data transformation. Transitions between pages, error messages and login/registration credentials are managed on the frontend using the data retrieved from the backend.

The Flask framework in Python is used to implement system operations on the backend. Connection with the frontend is established via the endpoints provided by Flask. mySQLdb library of Flask was used to execute SQL commands in a database session. Cursors were used to fetch data from the database into the application and also to execute insert, update and delete operations on the database. Postman was used for the testing.

On the frontend, we have faced some synchronization problems. The data retrieved from the backend was not synchronous with the updates on the backend. For example, when a user adds a bet to their bet slip, the frontend failed to show the update on bet slip when we first tried. In order to solve this problem, contexts and hooks were utilized, which are React utils that help update a component whenever a change is detected.

Rahmiye Büşra Büyükgebiz was responsible for the common functionalities: Signup and login functionalities for different user types (user, editor, admin), and additional features such as market and achievements. She also implemented Market, Profile, Register and

Login pages which consist of common and additional functionalities with the help of Yüce and Ozan. Mert Aslan was responsible for the functionality of “user makes a bet”. He wrote the queries of listing possible bets including selection ratio, filtering them according to MBN, sport, contest and keyword, placing bet slip, user commenting and liking on their friends’ bet slip. On the frontend, he worked on the Home page with Yüce and Ozan. Ozan Aydın was responsible for the functionality of “editor suggests bet”. He wrote queries of editors which are suggesting a bet for a match, sharing bet slips based on some constraints and showing performance of an editor. He worked on almost all pages on the frontend, connecting the frontend with the backend via HTTP requests. Yüce Hasan Kılıç was responsible for the functionality of admin management. He wrote queries of admin changes odd of a bet, cancels a bet, modifies achievements and market items and bans/unbans users. He also designed the website.

5. Advanced Database Components

5.1. Reports

5.1.1. Editor Number of Bet Slips Won Query

```
WITH editor_slips AS (SELECT bet_slip_id, creator_id as editor_id FROM
bet_slip WHERE creator_id = @creator_id), editor_bet_id AS (SELECT * FROM
included_bet NATURAL JOIN editor_slips)
SELECT COUNT(bet_slip_id) AS won_bet_slip_count FROM editor_slips WHERE NOT
EXISTS (SELECT bet_id, match_id FROM editor_bet_id NATURAL JOIN bet WHERE
bet_slip_id = bet_slip_id AND (result = 'LOST' OR result = 'PENDING')) GROUP
BY editor_id
```

This query finds the total amount of winning bet slips the editor has played since the creation of the account.

5.1.2. Editor Single Bet Pick Win Count

```
WITH editor_bets AS (SELECT bet_id, match_id, editor_id FROM suggested_bet
WHERE editor_id = @editor_id)
SELECT COUNT(bet_id) AS won_count FROM editor_bets NATURAL JOIN bet GROUP BY
editor_id, result HAVING result = 'WON'
```

This query finds the total number of winning single bet picks of an editor since the creation of the account.

5.1.3. Pending Bet Slips of a User

```
WITH user_bet_slips AS (SELECT bet_slip_id FROM bet_slip WHERE creator_id =
@creator_id AND placed = TRUE), pending_slip AS (SELECT DISTINCT bet_slip_id
FROM user_bet_slips NATURAL JOIN included_bet NATURAL JOIN bet WHERE result
= 'PENDING'), all_bet_data AS (SELECT * FROM pending_slip NATURAL JOIN
included_bet NATURAL JOIN bet), match_data AS (SELECT * FROM all_bet_data
NATURAL JOIN competitor_match), all_competitors AS (SELECT competitor_name,
competitor_id FROM (SELECT player_id AS competitor_id, CONCAT(forename, ' ',
surname) AS competitor_name FROM individual_player) AS temp UNION (SELECT
team_name AS competitor_name, team_id AS competitor_id FROM team))
SELECT * FROM match_data NATURAL JOIN all_competitors
```

This query finds all of the bet slips of the user that currently include at least one bet that is pending, which makes the betslip itself a pending betslip.

5.2. Views

```
CREATE VIEW total_achievement_count
AS
  SELECT COUNT(achievement_id) AS total_count FROM achievement;
```

This view shows the total amount of achievements that are currently existing in the system. This view is used in the Profile UI page to show the total number of possible achievements a user can obtain.

5.3. Triggers and Constraints

5.3.1. Triggers

```
CREATE TRIGGER achievement_check_3
AFTER UPDATE
ON bet_slip FOR EACH ROW
BEGIN
    DECLARE slip_count INT;
        SET slip_count = ( SELECT COUNT(*) FROM bet_slip WHERE placed = 1 AND
        creator_id = NEW.creator_id AND creator_id NOT IN (SELECT editor_id AS
        creator_id FROM editor));
    IF slip_count = 1
       THEN
          INSERT INTO gained_achievement VALUES (3, NEW.creator_id);
    END IF;
 END
```

This is an example of the triggers used in the system, which automatically grants a user an achievement once they place their first bet slip. Other triggers include;

- Automatically determining the result of a bet whenever a result is inserted to the result table. This is done by comparing the results with the bet types in order to determine the state of the bet.
- Auto-incrementing “bet_id” primary key whenever a bet is added to the bet table. This is done via triggers, as the bet table has a composite primary key composed of match_id and bet_id together. Trigger allows us to reset the bet_id index to 1 per match_id.

5.3.2. Constraints

```
CHECK(sport_name IN('TENNIS', 'FOOTBALL',
'BASKETBALL'))
```

This is an example of a constraint in the system, which checks if the sport name to be added is one of “FOOTBALL”, “BASKETBALL” or “TENNIS”. There are other constraints in the system such as;

- Bet results should be one of “WON”, “LOST” or “PENDING”
- Side value of a tuple in a match should be either “HOME” OR “AWAY”
- Editor request states should either be “APPROVED” or “ENDED”

Other than check constraints, our system has many foreign key constraints that disallow insertion if the insertion violates the foreign key constraint.

5.3.3. Stored Procedures

```
CREATE PROCEDURE SelectAllEditorBetslips
AS
SELECT bet_slip_id FROM bet_slip WHERE creator_id IN (SELECT editor_id AS
creator_id FROM editor)
GO;
```

This is an example of many stored procedures we have used in our system. This procedure returns all bet slip id’s of bet slips that have been created by editors. These id’s are later used to gather all information of an editor’s bet slip in order to display them in the user interface.

6. User’s Manual

6.1.For Users

6.1.1. Register and Login Page

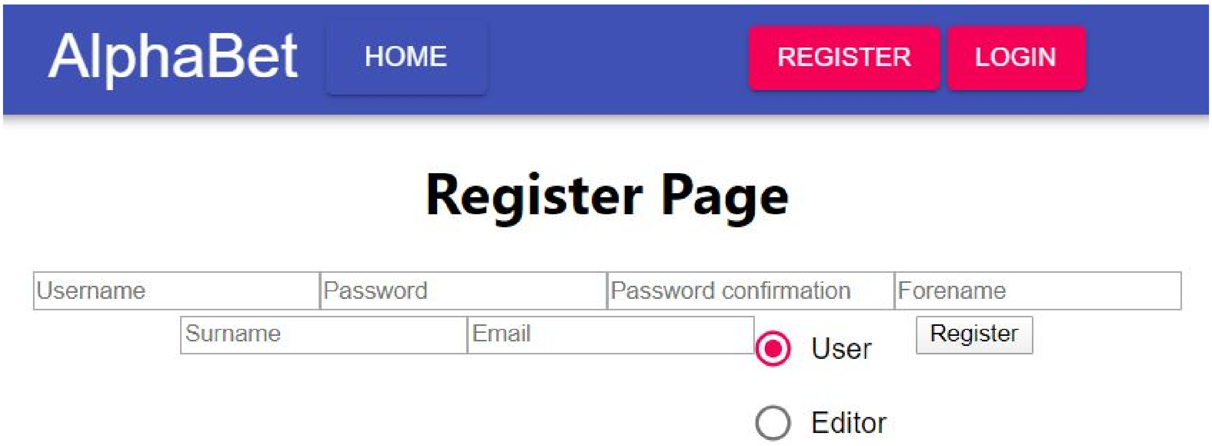


Figure 1 : Register Page

Figure 1 shows the register screen which shows up when a user clicks on the “Register” button on the navigation bar. By filling the information (“Username, Password, Password Confirmation, Forename, Surname, Email”) and selecting user type, either “User” or “Editor”, a user can register to the system and an editor can send a registration request to the admin.

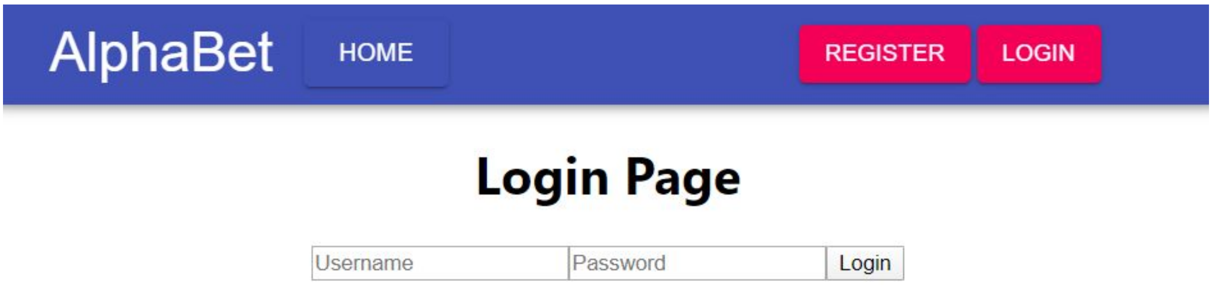
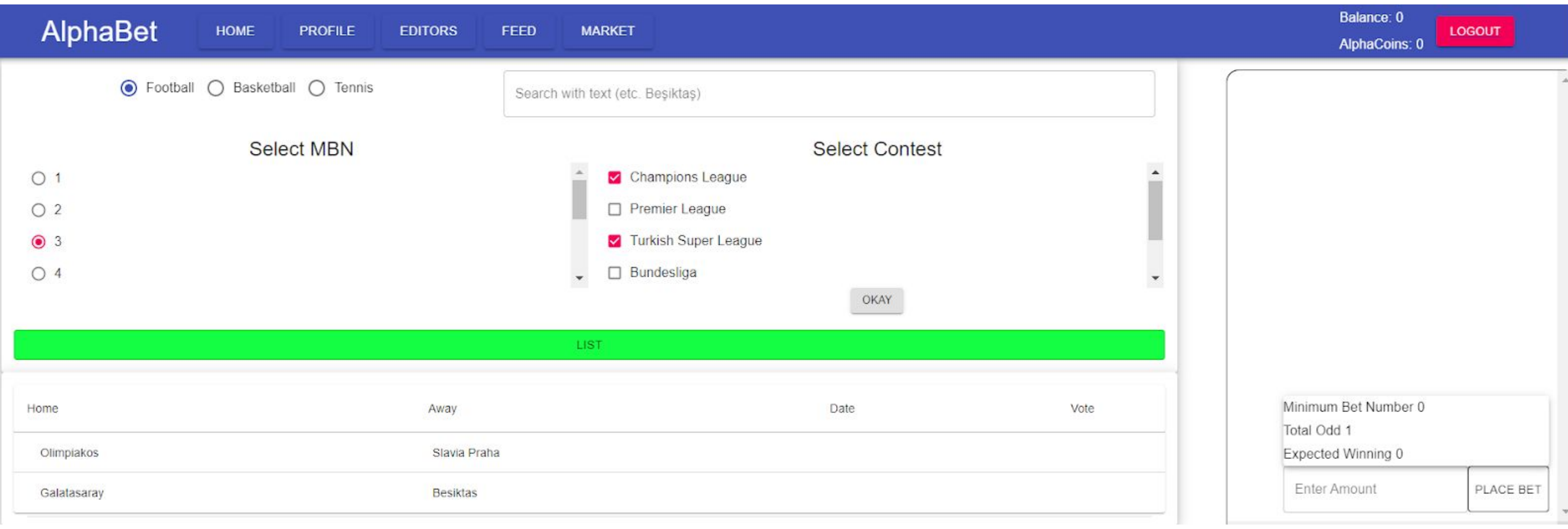


Figure 2 : Login Page

Figure 2 shows the login screen which shows up when a user clicks on the “Login” button on the top right of the navigation bar. All user types can login to the system by entering username and password. If a pending editor, an unregistered person or a banned user tries to login to the system, an error message pops up.

6.1.2. Home Page



Balance: 0
AlphaCoins: 0

● Football ○ Basketball ○ Tennis

Select MBN

○ 1
○ 2
● 3
○ 4

Select Contest

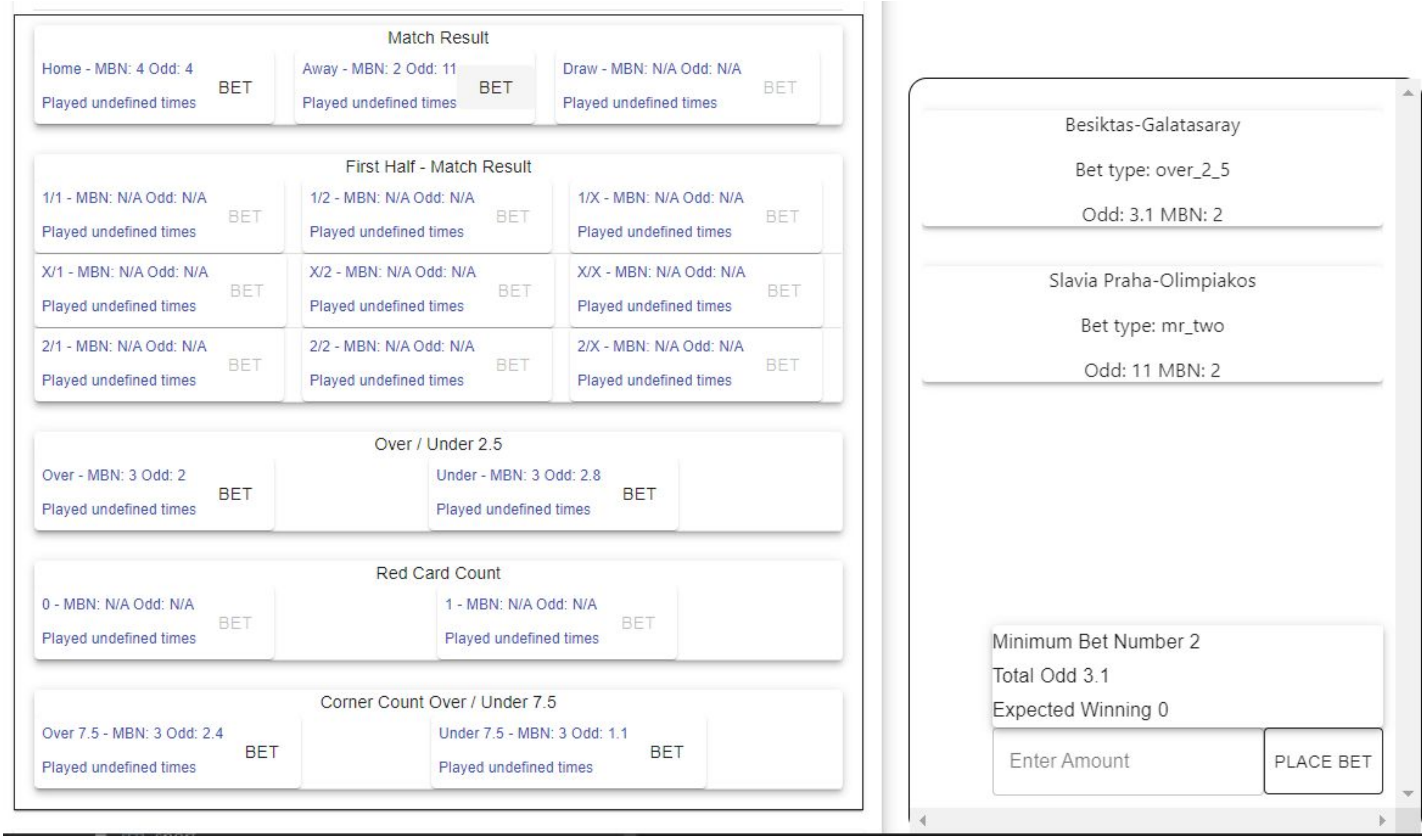
☑ Champions League
□ Premier League
☑ Turkish Super League
□ Bundesliga

Home	Away	Date	Vote
Olimpiakos	Slavia Praha		
Galatasaray	Besiktas		

Minimum Bet Number 0
Total Odd 1
Expected Winning 0

Figure 3 : Home Page

Figure 3 shows the homepage of the user where the user can filter matches and bets and view them. At the top right corner, the user can view their account balance and their AlphaCoins. The user can log out of the system by clicking on the “LOGOUT” button. Navigation bar shows other UI pages the user can visit, which are Profile, Editors, Feed and Market. The user can filter bets by entering a keyword on the text field at the top (“Search with text”), or filter according to sport type, MBN and contest name. While filtering, the user is able to choose multiple contests and when the user chooses another sport type, the contests for other sport types become disabled. After clicking on a match, bets of the match show up below the match name.


Match Result

Home - MBN: 4 Odd: 4
Played undefined times

Away - MBN: 2 Odd: 11
Played undefined times

Draw - MBN: N/A Odd: N/A
Played undefined times

First Half - Match Result

1/1 - MBN: N/A Odd: N/A
Played undefined times

1/2 - MBN: N/A Odd: N/A
Played undefined times

1/X - MBN: N/A Odd: N/A
Played undefined times

X/1 - MBN: N/A Odd: N/A
Played undefined times

X/2 - MBN: N/A Odd: N/A
Played undefined times

X/X - MBN: N/A Odd: N/A
Played undefined times

2/1 - MBN: N/A Odd: N/A
Played undefined times

2/2 - MBN: N/A Odd: N/A
Played undefined times

2/X - MBN: N/A Odd: N/A
Played undefined times

Over / Under 2.5

Over - MBN: 3 Odd: 2
Played undefined times

Under - MBN: 3 Odd: 2.8
Played undefined times

Red Card Count

0 - MBN: N/A Odd: N/A
Played undefined times

1 - MBN: N/A Odd: N/A
Played undefined times

Corner Count Over / Under 7.5

Over 7.5 - MBN: 3 Odd: 2.4
Played undefined times

Under 7.5 - MBN: 3 Odd: 1.1
Played undefined times

Besiktas-Galatasaray

Bet type: over_2_5

Odd: 3.1 MBN: 2

Slavia Praha-Olimpiakos

Bet type: mr_two

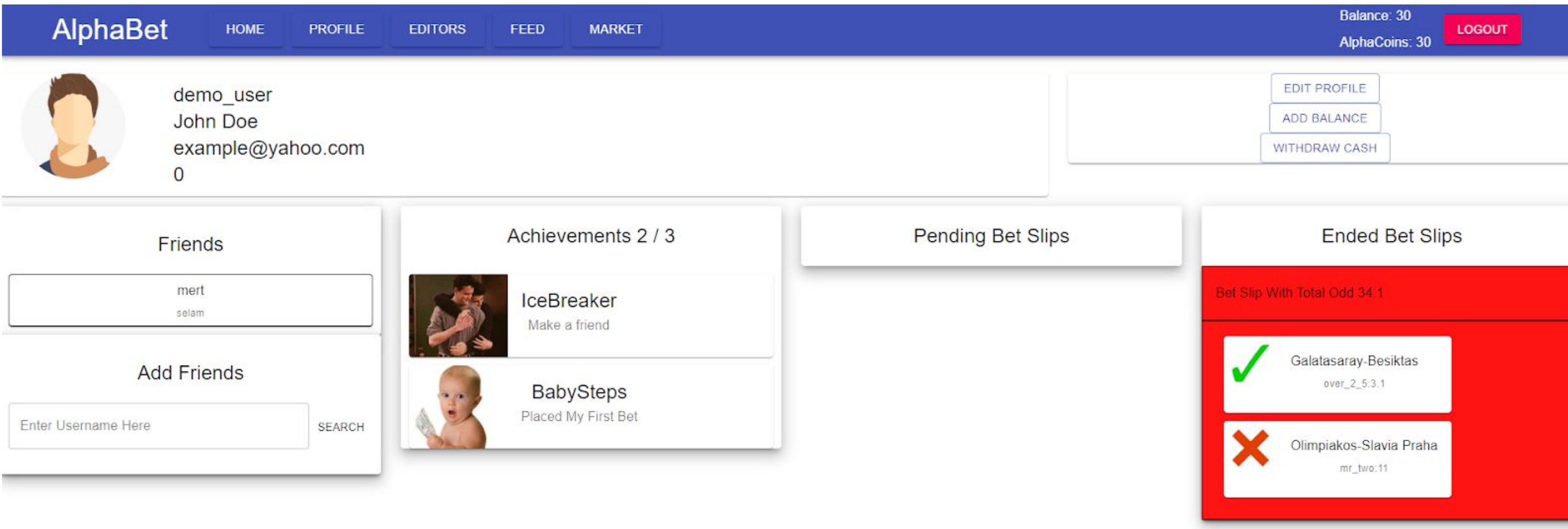
Odd: 11 MBN: 2

Minimum Bet Number 2
Total Odd 3.1
Expected Winning 0

Figure 4 : Filter Results and Bet Slip

Figure 4 shows the result of filtering the bets in the system with the criteria shown in Figure 3. A user can click on “BET” button in order to add the bet to the betslip. Users can enter an amount to play in bet slip and click on the “PLACE BET” button. If MBN constraint is satisfied, and the user has enough account balance, the betslip is placed.

6.1.3. Profile Page



Balance: 30
AlphaCoins: 30

demo_user
John Doe
example@yahoo.com
0

Friends

mert
selam

Add Friends

Enter Username Here

Achievements 2 / 3

IceBreaker
Make a friend

BabySteps
Placed My First Bet

Pending Bet Slips

Ended Bet Slips

Bet Slip With Total Odd 34.1

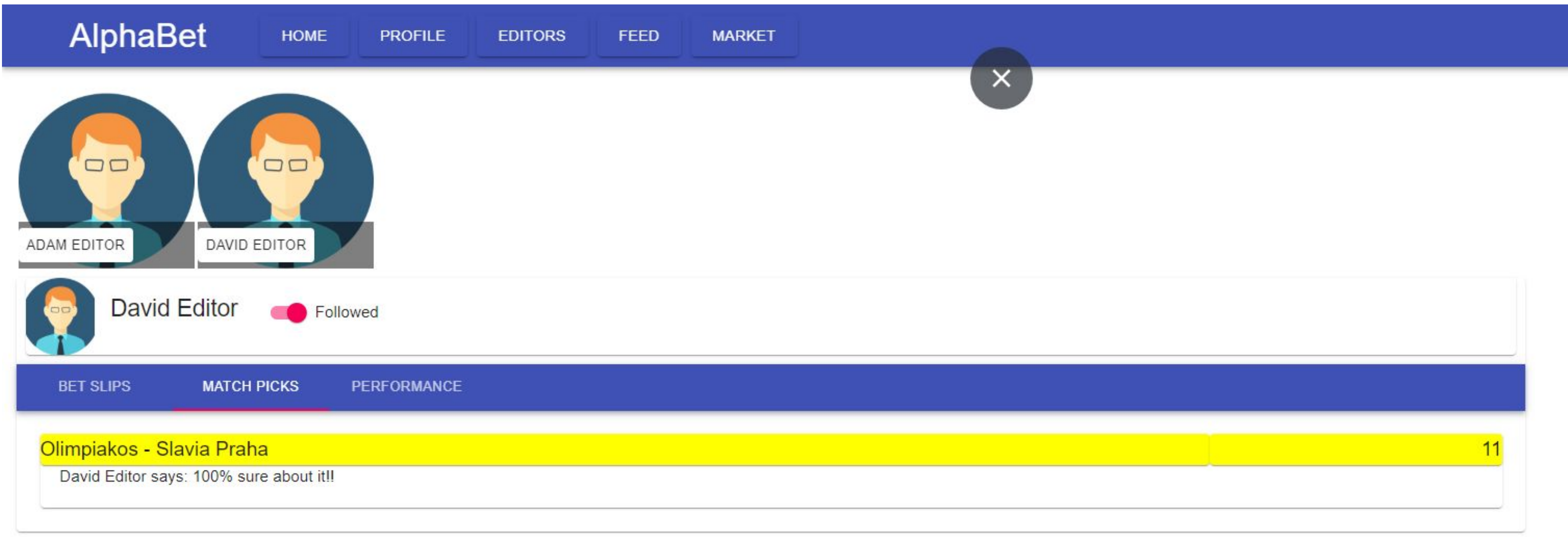
✓ Galatasaray-Besiktas
over_2_5:3.1

✗ Olimpiakos-Slavia Praha
mr_two:11

Figure 5 : Profile Page

Figure 5 shows the profile page of a user. The user can view their account details, edit their profile information, add balance to their accounts and withdraw cash from the system. They can search for existing users in the system in order to add them as friends. When they add a user as a friend, their friend is shown in the “Friend” section. A user can also view their achievements that they have earned through completing several tasks such as adding a user as a friend or placing their first bet. Last but not least, the user can view their pending and ended bet slips. By clicking on any of these bet slips, the user can view the included bets and their status.

6.1.4. Editors

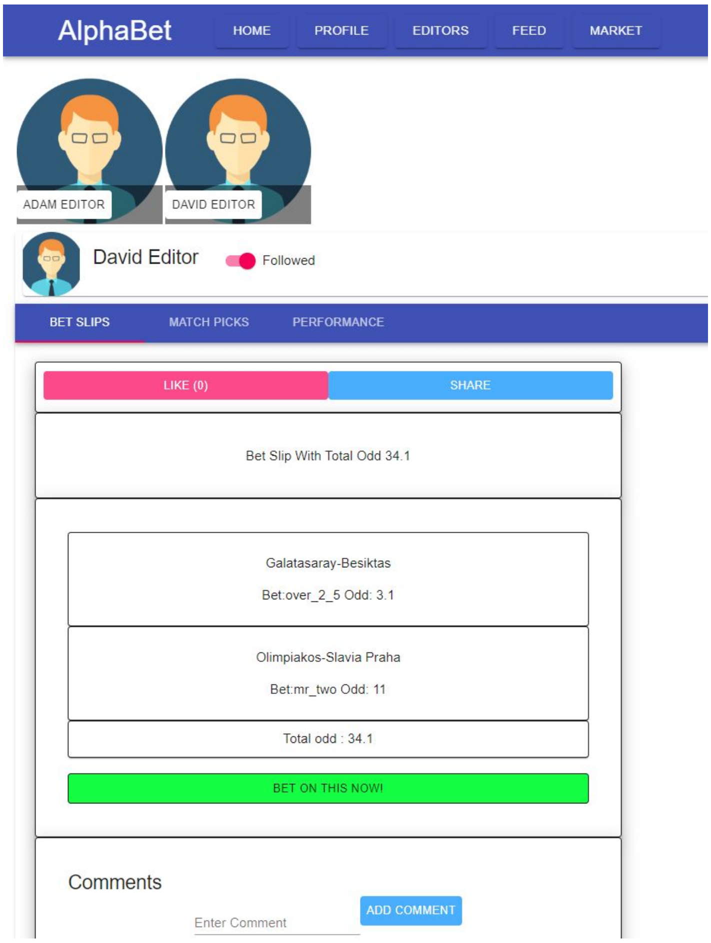


Olimpiakos - Slavia Praha 11
David Editor says: 100% sure about it!!

Figure 6: Editor's Match Picks

Figure 6 shows the Editor bar and Match Pick tab. From the tab, users can see the approved editors and click the button to follow them. Once followed, that editor's content is available to the user, including Match Picks, where users can see bet suggestions from the editors.

Figure 7 shows the Editor's Bet Slips, which shows the bet slips that the editor shared. Users can like any certain bet slip and share that bet slip in their feed. Users can also click on the "Bet On This Now" button in order to copy the bets of that bet slip into their own bet slip with a single click. Users can also comment on the shared bet slip by clicking the add comment button.



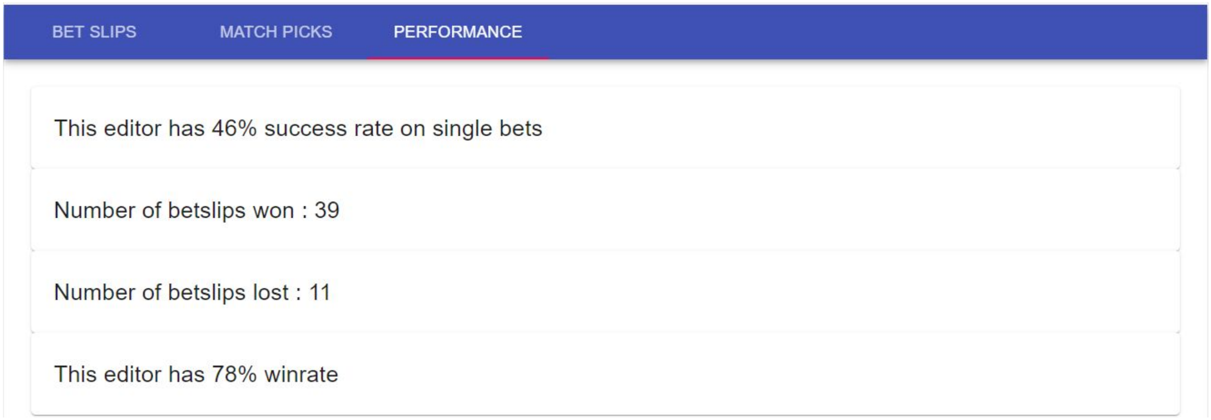
Bet Slip With Total Odd 34.1

Galatasaray-Besiktas
Bet:over_2_5 Odd: 3.1

Olimpiakos-Slavia Praha
Bet:mr_two Odd: 11

Total odd : 34.1

Figure 7: Editor's Bet Slips



This editor has 46% success rate on single bets

Number of betslips won : 39

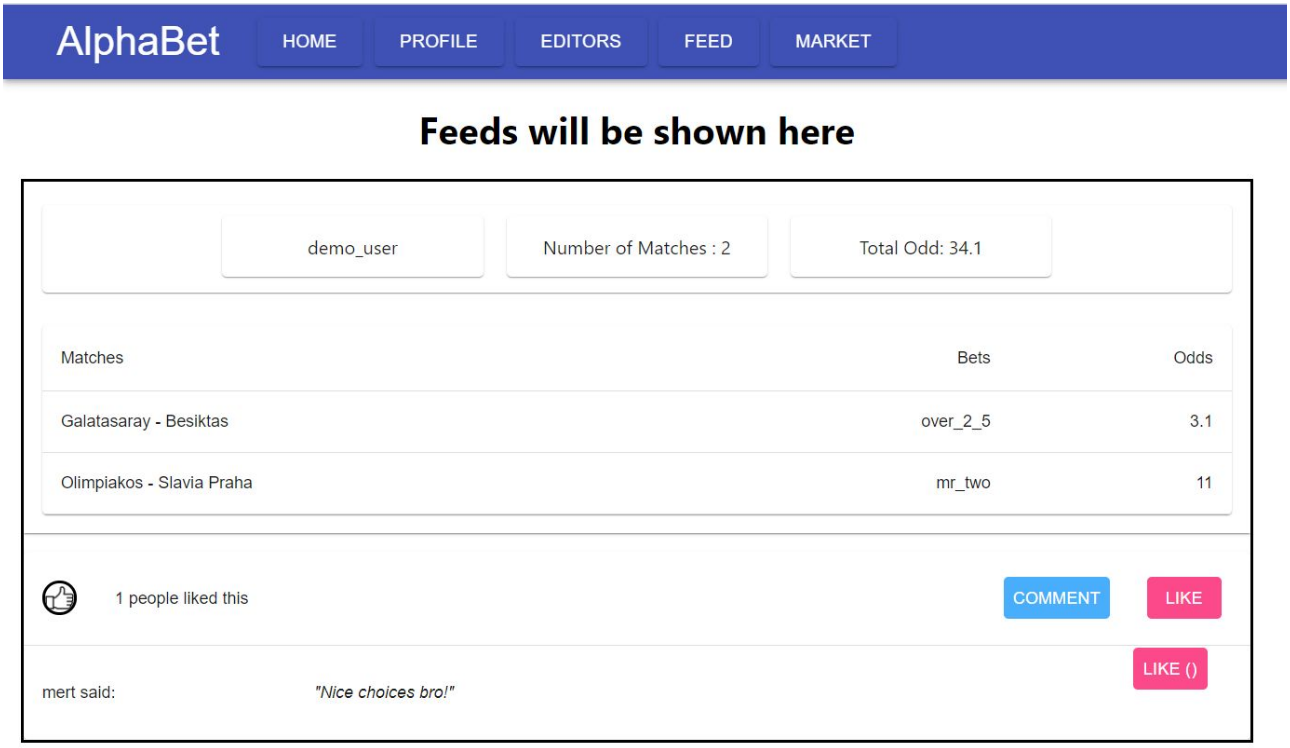
Number of betslips lost : 11

This editor has 78% winrate

Figure 8: Editor’s Performance

Figure 8 shows the Performance Page of the editor, which is visible only by following the editor. Here, the editor's overall success rate on the suggested single bets are shown, along with number of bet slips won and lost, and the win rate calculated by using the above values.

6.1.5. Feed



Feeds will be shown here

demo_user | Number of Matches : 2 | Total Odd: 34.1

Matches	Bets	Odds
Galatasaray - Besiktas	over_2_5	3.1
Olimpiakos - Slavia Praha	mr_two	11

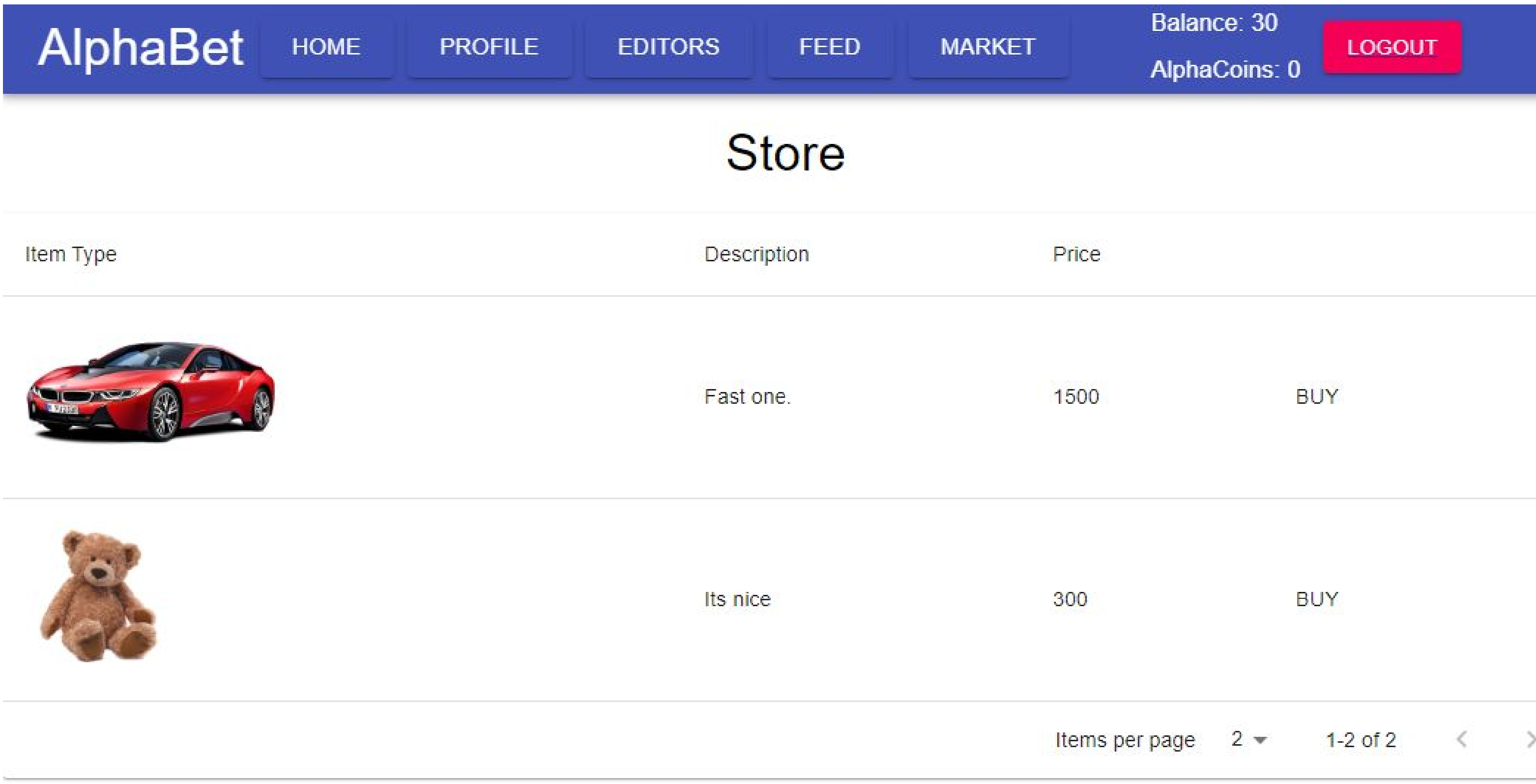
1 people liked this

mert said: *"Nice choices bro!"*

Figure 9: Feed Page

Figure 8 shows the feed page where users can see the bet slips played by their friends. Users can click the comment button in order to comment on any betslip they see, and they can also like that betslip by clicking the like button. Any user can also like the comments that other people left on bet slips.

6.1.6. Market



Balance: 30

AlphaCoins: 0

Store

Item Type	Description	Price	
	Fast one.	1500	BUY
	Its nice	300	BUY

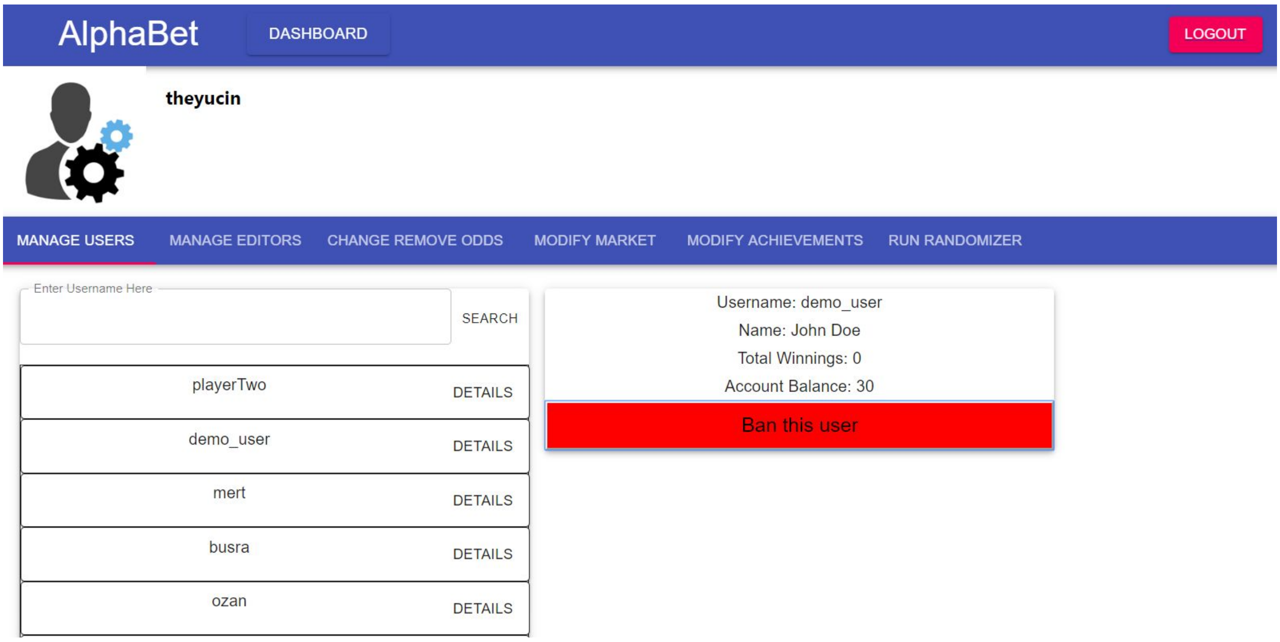
Items per page 2 1-2 of 2

Figure 10 : Market Page

Figure 9 shows the Market page where users can spend their AlphaCoins on items in store. When the user clicks on the “BUY” button, AlphaCoins is checked if it has enough credit to buy the item. When a user buys the item, its price is deducted from AlphaCoins of the user.

6.2.For Admin

6.2.1. Manage Users



playerTwo

demo_user

mert

busra

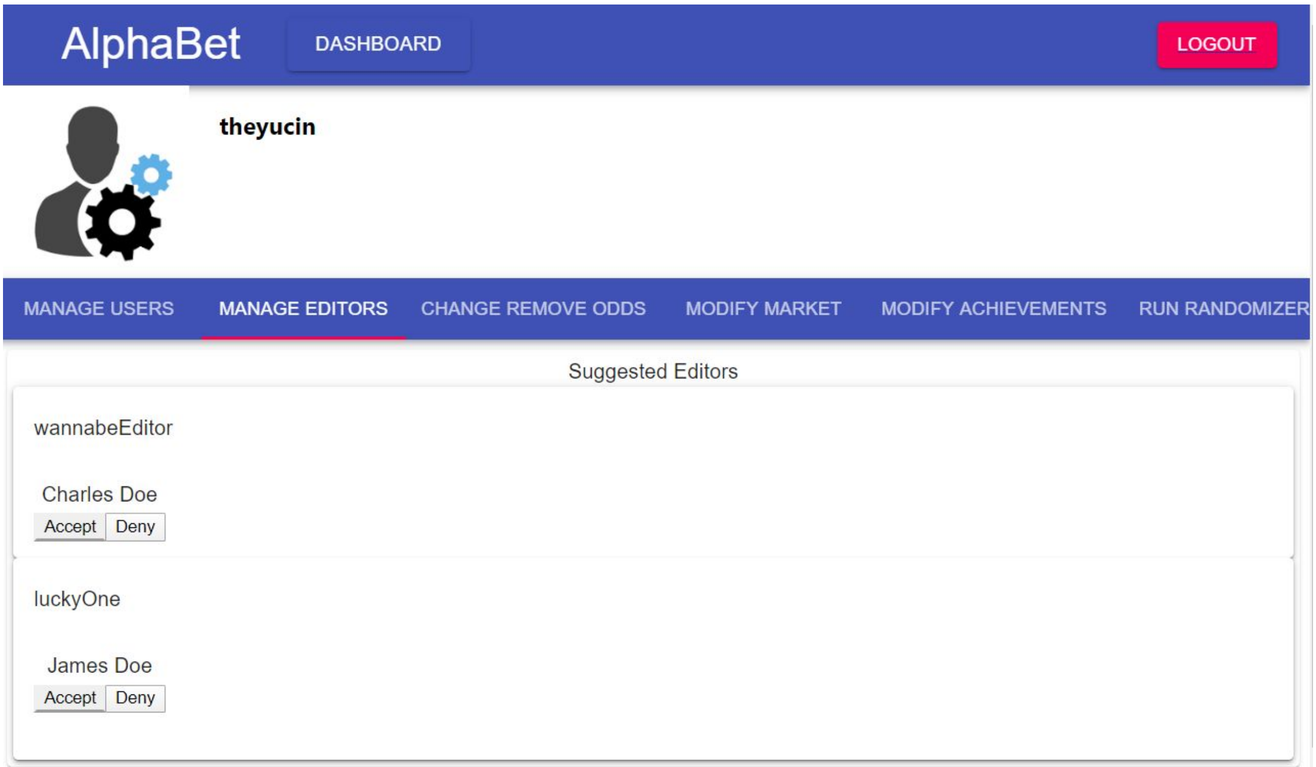
ozan

Username: demo_user
Name: John Doe
Total Winnings: 0
Account Balance: 30

Figure 11 : Manage Users Page

Figure 10 shows Manage Users tab in Admin Dashboard. In this tab, admin can search for users with their usernames, display additional details about them such as name, account balance and total winnings. Also, admin can ban a user by adding them to blacklist or unban users in blacklist. Banned users receive an error message in the login page when they try to login.

6.2.2. Manage Editor Request



theyucin

Suggested Editors

wannabeEditor

Charles Doe

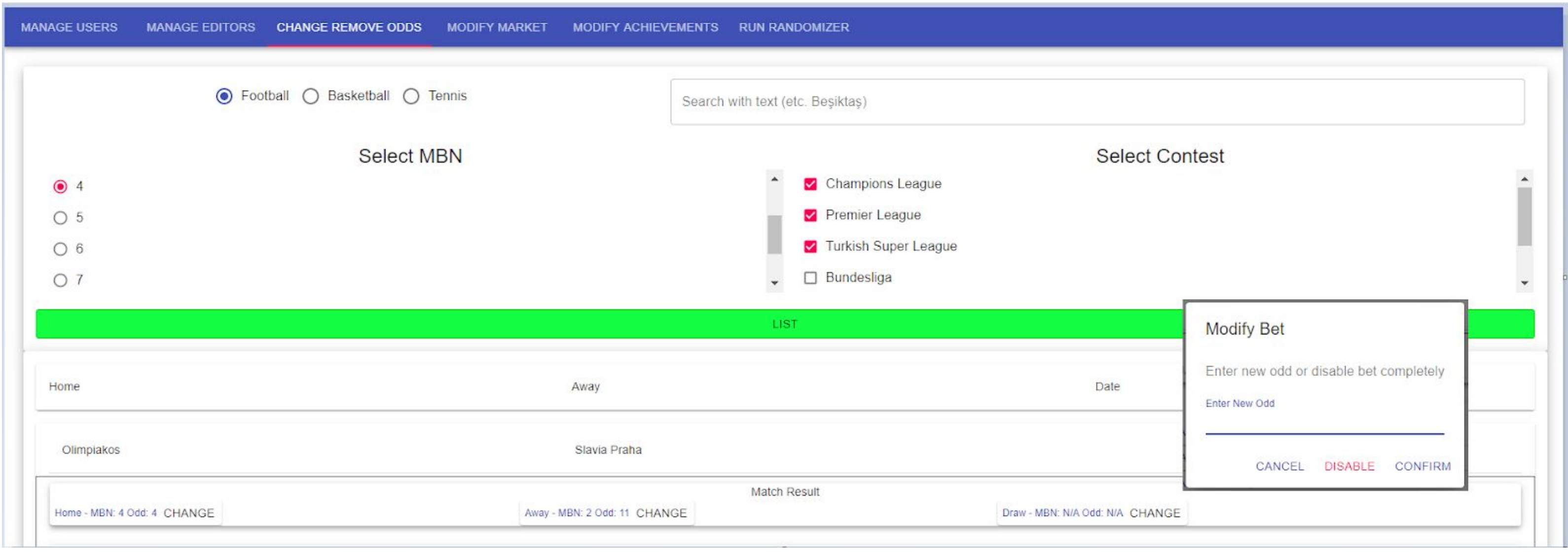
luckyOne

James Doe

Figure 12 : Manage Editor Requests Page

Figure 11 shows Manage Editor Requests tab in Admin Dashboard. In this tab, an admin can see editor requests that are created when a person tries to register as an editor. Admin can either accept or decline those requests by changing request status from “Pending” to “Approved” or “Declined” in the database. Approved editors will be able to login to the system as editors and use editor features. Declined editors’ requests will be deleted and no longer shown in the dashboard.

6.2.3. Change/Remove Odds of Bets



● Football ○ Basketball ○ Tennis

Search with text (etc. Beşiktaş)

Select MBN

● 4
○ 5
○ 6
○ 7

Select Contest

☑ Champions League
☑ Premier League
☑ Turkish Super League
□ Bundesliga

Home	Away	Date
Olimpiakos	Slavia Praha	

Match Result

Home - MBN: 4 Odd: 4 CHANGE

Away - MBN: 2 Odd: 11 CHANGE

Draw - MBN: N/A Odd: N/A CHANGE

Modify Bet

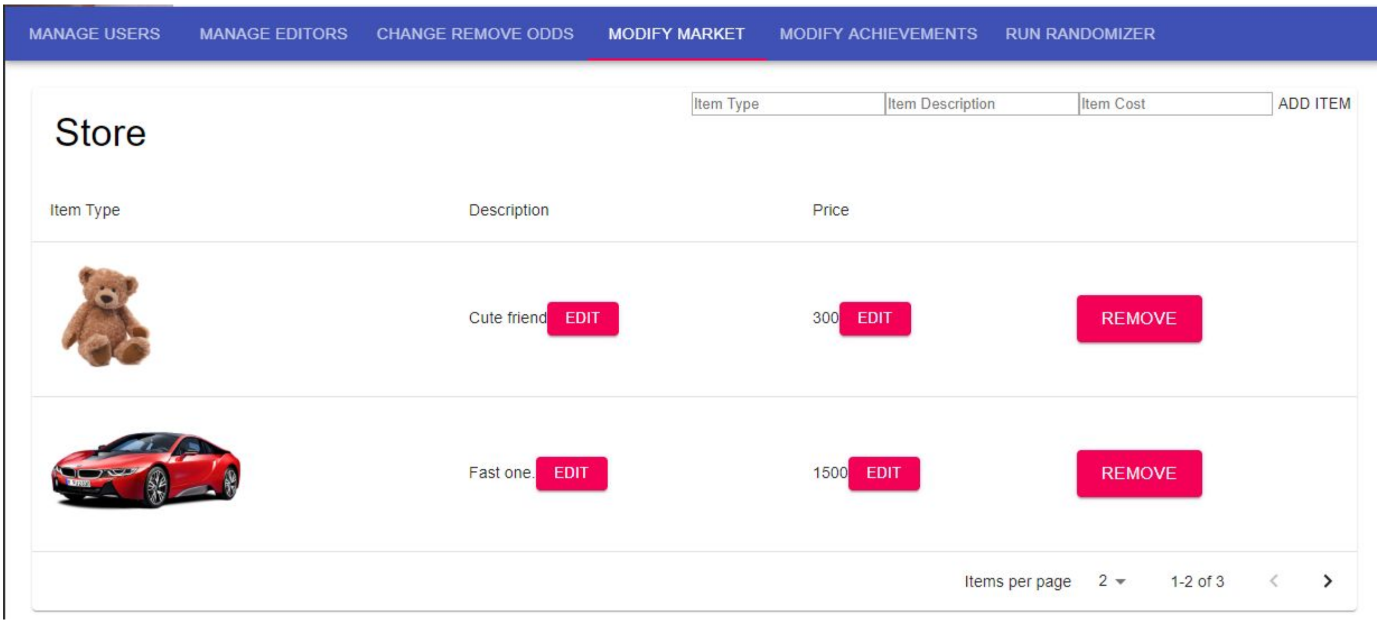
Enter new odd or disable bet completely

Enter New Odd

Figure 13 : Change/Remove Odds Page

Figure 12 shows the change/remove odds page for admins where they can filter bets by entering a keyword on the text field at the top (“Search with text”), or filter according to sport type, MBN and contest name. While filtering, they are able to choose multiple contests and when they choose another sport type, the contests for other sport types become disabled. After clicking on a match, bets of the match show up below the match name and admins can change the odd of a bet or disable a bet by clicking on the “Change” button.

6.2.4. Modify Market



Store

Item Type	Description	Price	
	Cute friend EDIT	300 EDIT	REMOVE
	Fast one. EDIT	1500 EDIT	REMOVE

Items per page 2 1-2 of 3

Figure 14 : Modify Market Page

Figure 13 shows the modify market page for admins. Admins can add a new item by entering the new item’s type, description and cost, and clicking on the “Add Item” button. They can also edit the cost and the description of the item by clicking the respective “Edit” buttons, and can remove an item from the market completely by clicking the “Remove” button.

6.2.5. Modify Achievements

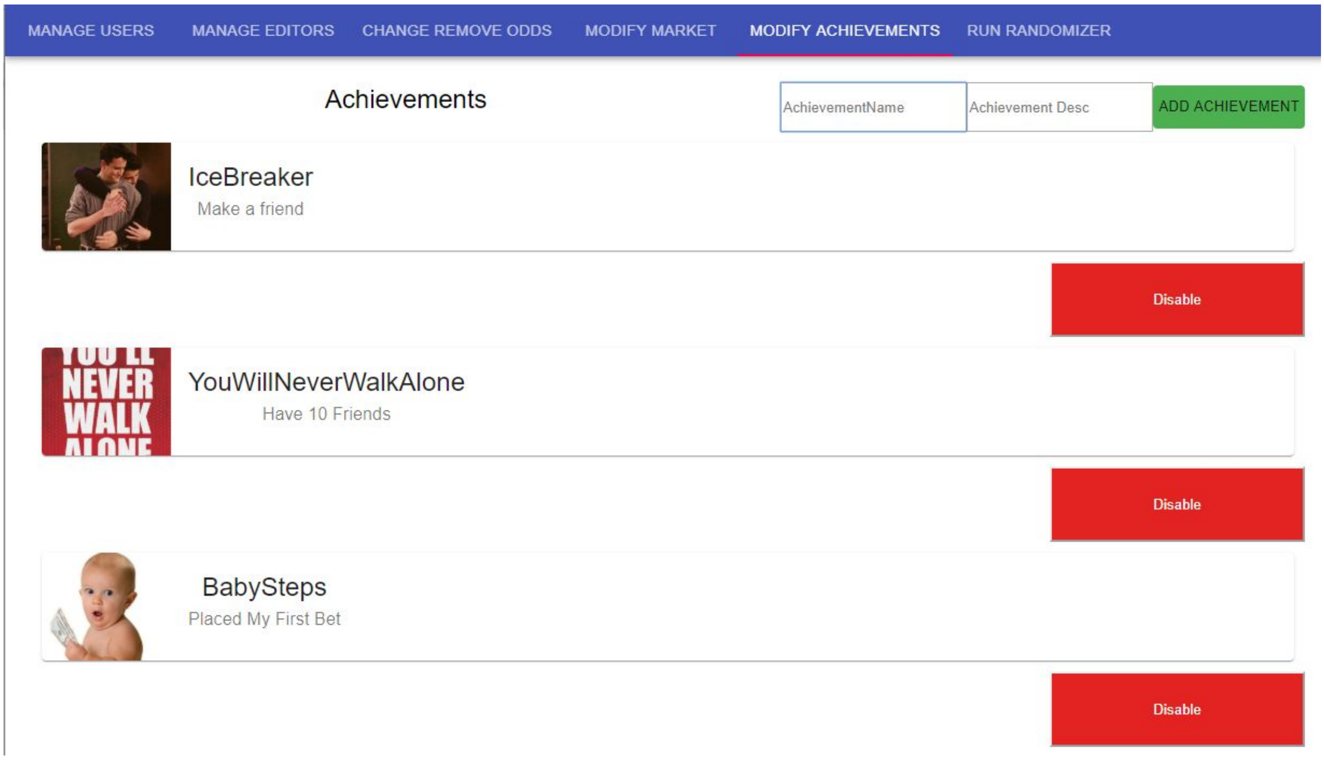
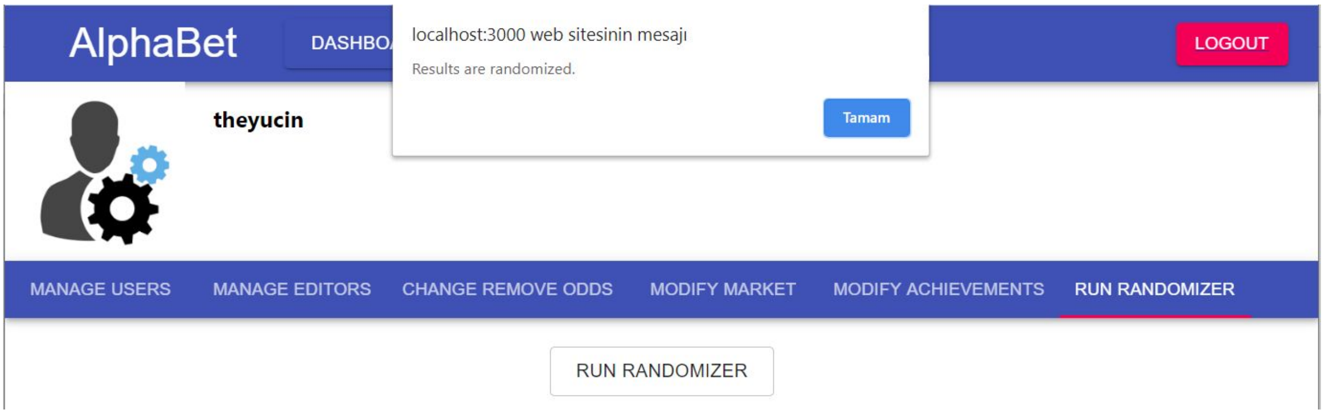


Figure 15 : Modify Achievements Page

Figure 14 shows the modify achievements page for admins where they can add new achievements by entering achievement name and description and also disable achievements by clicking on the “Disable” button.

6.2.6. Randomizer



localhost:3000 web sitesinin mesajı

Results are randomized.

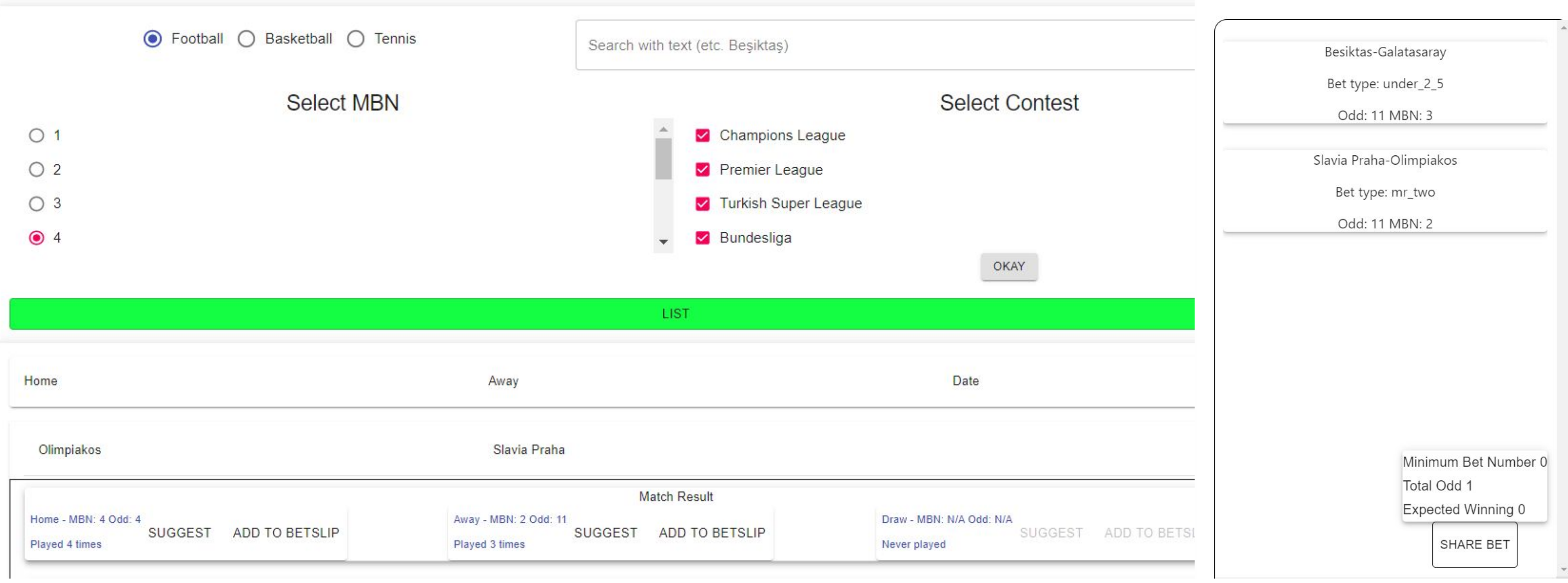
Tamam

Figure 16 : Result Randomizer Page

Figure 15 shows the activation button for the effect called “randomizer”. This effect distributes random results to each unfinished match in the system, which in turn causes the bets in the system to result in either a loss or a win. Users are then rewarded their winnings if their bet slips won.

6.3.For Editors

6.3.1. Editor Home Page



● Football ○ Basketball ○ Tennis

Search with text (etc. Beşiktaş)

Select MBN

○ 1
○ 2
○ 3
● 4

Select Contest

☑ Champions League
☑ Premier League
☑ Turkish Super League
☑ Bundesliga

Home	Away	Date
Olimpiakos	Slavia Praha	

Match Result

Home - MBN: 4 Odd: 4
Played 4 times

Away - MBN: 2 Odd: 11
Played 3 times

Draw - MBN: N/A Odd: N/A
Never played

Besiktas-Galatasaray
Bet type: under_2_5
Odd: 11 MBN: 3

Slavia Praha-Olimpiakos
Bet type: mr_two
Odd: 11 MBN: 2

Minimum Bet Number 0
Total Odd 1
Expected Winning 0

Figure 17 : Editor Home Page

Figure 16 shows the Editor Home page, which is visible only to approved editors. Here, the filtering system works exactly the same as previous homepages where an editor can search for any match in the database. While viewing the bets of that match, instead of a “Bet” button, two buttons are visible, titled “Suggest” and “Add to Betslip”. Suggest button will open a pop-up screen where the editor can write a description and suggest that bet to the users, to be visible in the Single Match Pick section of the editors page. Add to betslip will add that bet to editor’s bet slip, and within the bet slip, the button is labeled “Share Bet Slip”, as editors cannot play bet slips but can share them with their followers inside the Shared Bets section in the editor’s page that are available to users.

7. Website

Our project information website link: https://busrabgz.github.io/