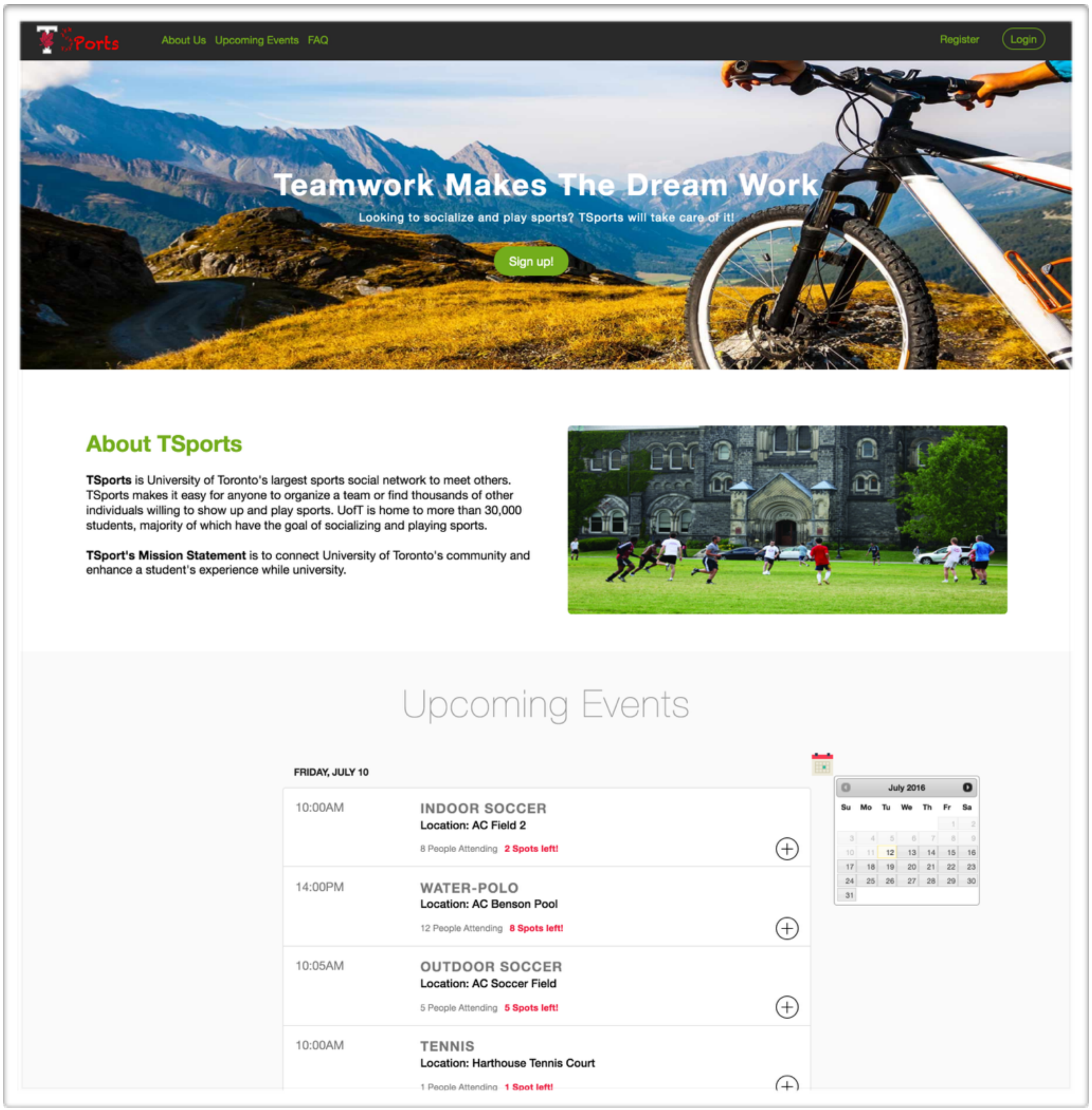


Teamwork Makes The Dream Work

Looking to socialize and play sports? TSports will take care of it!

About TSports

TSports is University of Toronto's largest sports social network to meet others. TSports makes it easy for anyone to organize a team or find thousands of other individuals willing to show up and play sports. UofT is home to more than 30,000 students, majority of which have the goal of socializing and playing sports.

TSport's Mission Statement is to connect University of Toronto's community and enhance a student's experience while university.

Upcoming Events

FRIDAY, JULY 10

10:00AM INDOOR SOCCER
Location: AC Field 2
8 People Attending 2 Spots left!

14:00PM WATER-POLO
Location: AC Benson Pool
12 People Attending 8 Spots left!

10:05AM OUTDOOR SOCCER
Location: AC Soccer Field
5 People Attending 5 Spots left!

10:00AM TENNIS
Location: Harthouse Tennis Court
1 People Attending 1 Spot left!

July 2016

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Final Report For a Sharing Economy Web App

Prepared by: Parham Oghabi, Yoomyung Jhe, Jing Wang, and Si Hua Cao Liu
July 27, 2016

1. Introduction

This document is a report to present our team's final sharing economy web application and to describe each part of our application and a high-level view of the architecture. It will also highlight our testing procedures, patching potential security vulnerabilities, and performance enhancements following our testing. It will also illustrate our web application's usability and viability.

This document also explains specific design features that reinforce our core goal, which is to connect University of Toronto's community and to enhance a student's experience while at university. Some of the implemented features which allow us to reach our goal are:

- One-to-one messaging between users
- Group messaging among users who are attending the same sporting event.
- After each event, users who attended the same sporting event can rate each other and leave comments anonymously. A user's rating in a specific sport and the comments left are set as public to be seen by all users.
- Optimized Search and Recommendation System to search for both users and events and results are ranked according to the user's query and personal data.

TSports is University of Toronto's largest sports social network to meet with others, making it easy for anyone to organize a team or find thousands of other individuals willing to show up and play sports. After signing up, students can create sporting events and set the number of people required for that sporting event. These features are explained thoroughly in the next section.

2. Web Application Architecture

The following shows a high-level view of the web application architecture and a description of each part of the web application and how they interact with one another.

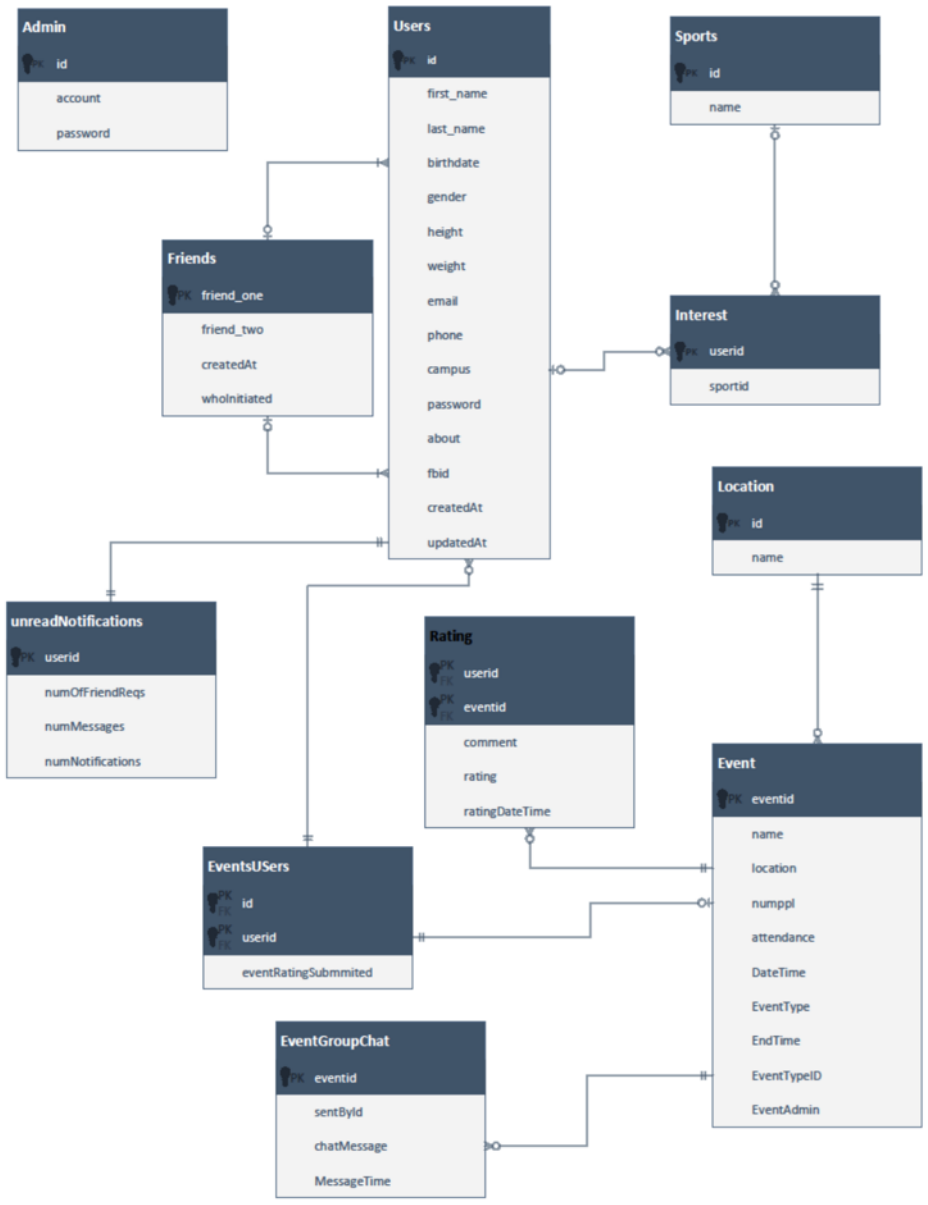
A. Application Design

From the main page, the user will be able to see the following links: About us, Upcoming Events, Log in, Registration, Signup, FAQ. Once registered, the user will be able to see other features of the website such as creating an event, searching an event or user, and editing personal information.

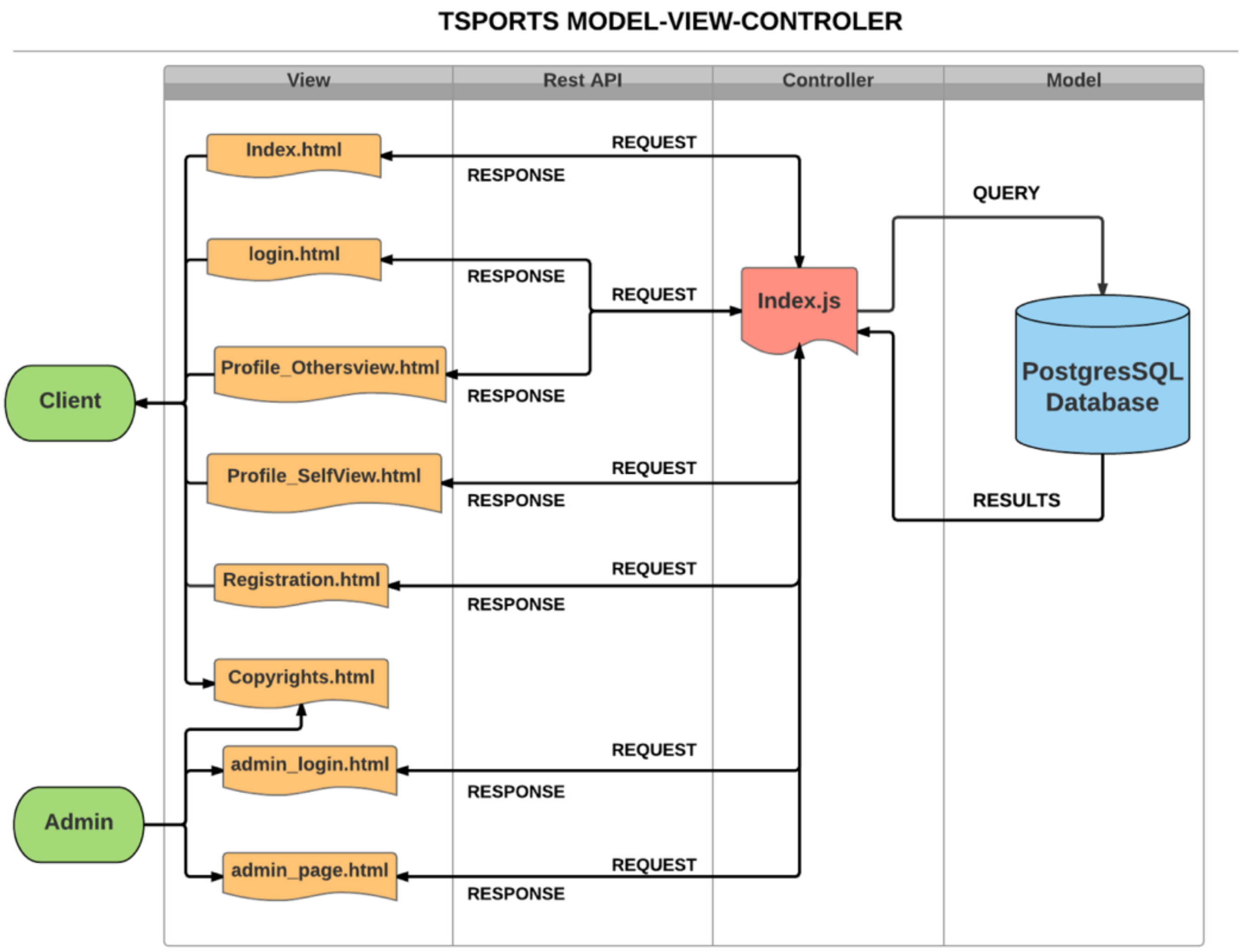
links	Description
About Us	Displays a brief description about the website.
Upcoming Events	Shows a list of Sports Events.
Log In	Shows a login page for the user to connect to TSPORTS System. It can be logged in with a Facebook account or with a regular TSPORTS account. If successful, it shows the profile page.
Registration	Displays a form to be filled in with personal information
Sign up	Redirects to the registration page
Admin	A dedicated link for admins to log in.
Copyright	Shows a page containing the links used to create the app.
Members Area	
About	Shows the user's information that was used upon registration.
Friends	Shows a list of list of friends.
Review	Shows reviews that the user has received.
Events	Shows a list of all Sports Events.
Search	It can search an user or an event.
Create Event	Displays a form to create a new event.
Sign out	Signs out the user and display the home page.
Admin Area	
Users	Displays all Members that are registered in the System.
Events	Displays all events created by all users
Status	Displays the total number of users, events and popular sports

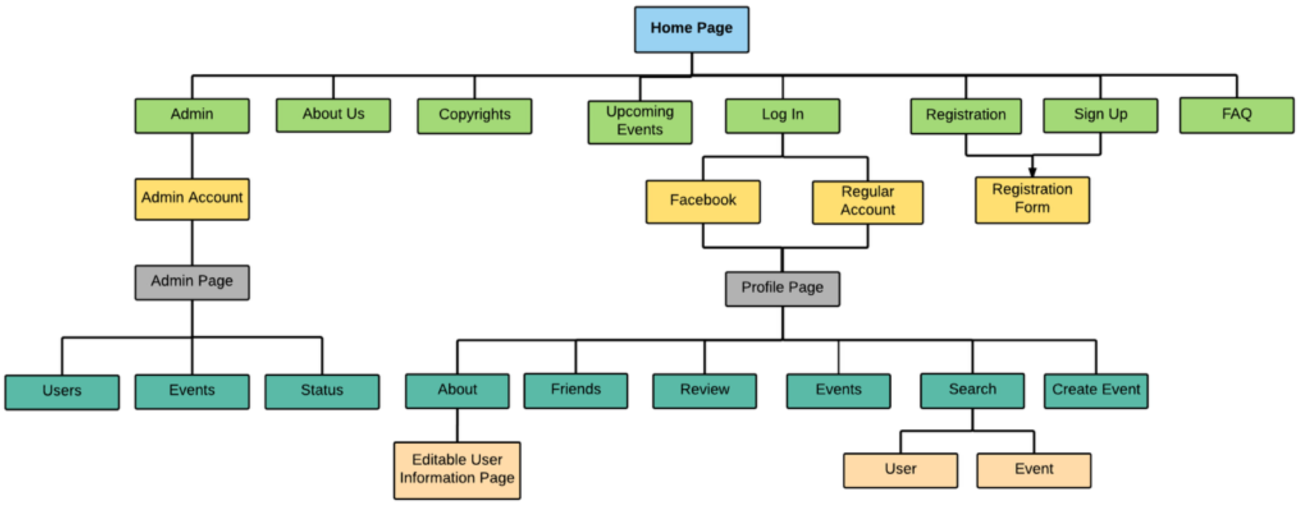
The database will store all information in 11 tables as depicted below.

TSPORTS Database ERD



B. Architecture

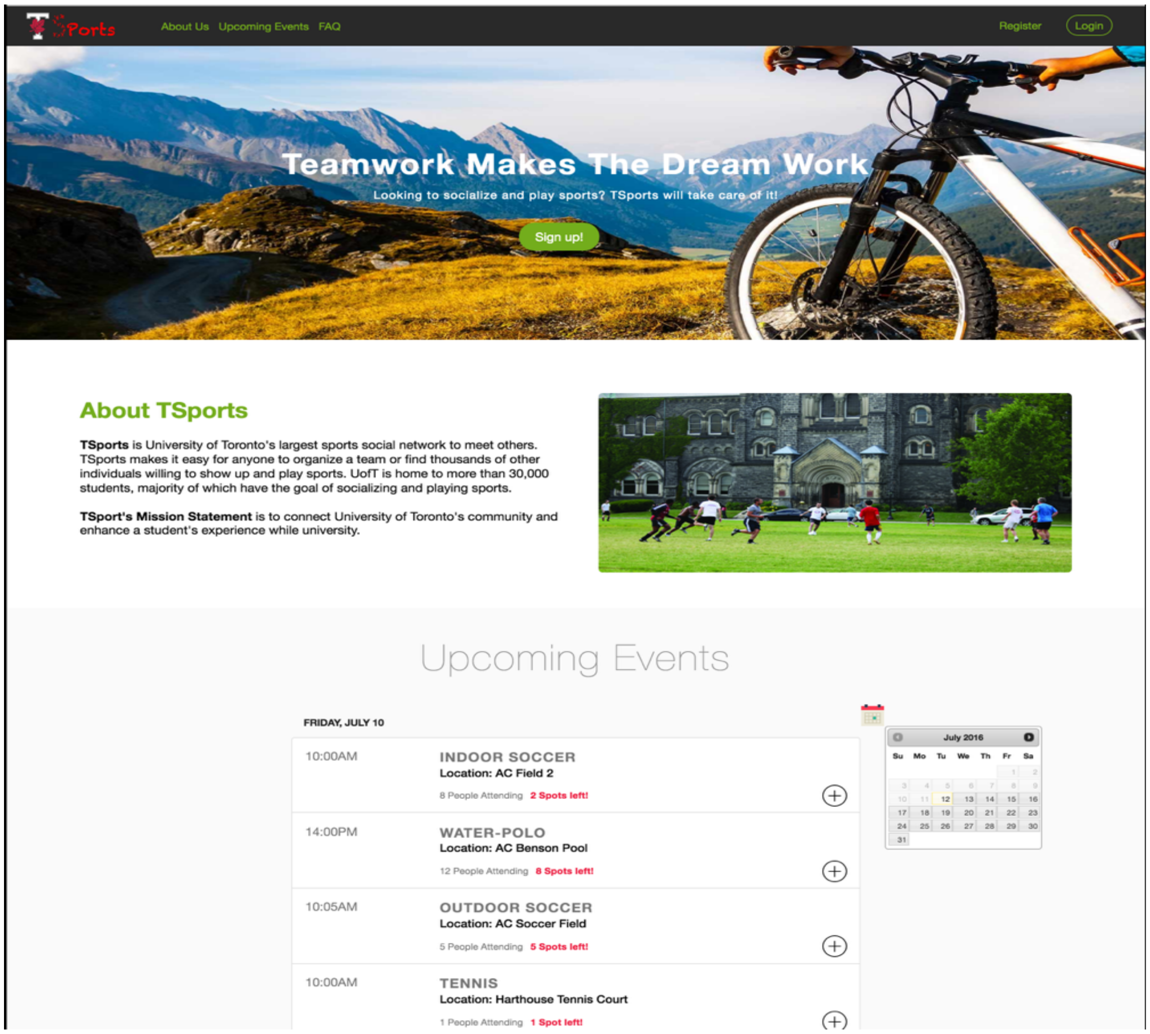




3. Design Features

The following shows a detailed design and a walkthrough of our web application and its various features.

A. Introduction Page (index.html)



Teamwork Makes The Dream Work

Looking to socialize and play sports? TSports will take care of it!

About TSports

TSports is University of Toronto's largest sports social network to meet others. TSports makes it easy for anyone to organize a team or find thousands of other individuals willing to show up and play sports. UofT is home to more than 30,000 students, majority of which have the goal of socializing and playing sports.

TSport's Mission Statement is to connect University of Toronto's community and enhance a student's experience while university.

Upcoming Events

FRIDAY, JULY 10

10:00AM INDOOR SOCCER
Location: AC Field 2
8 People Attending 2 Spots left!

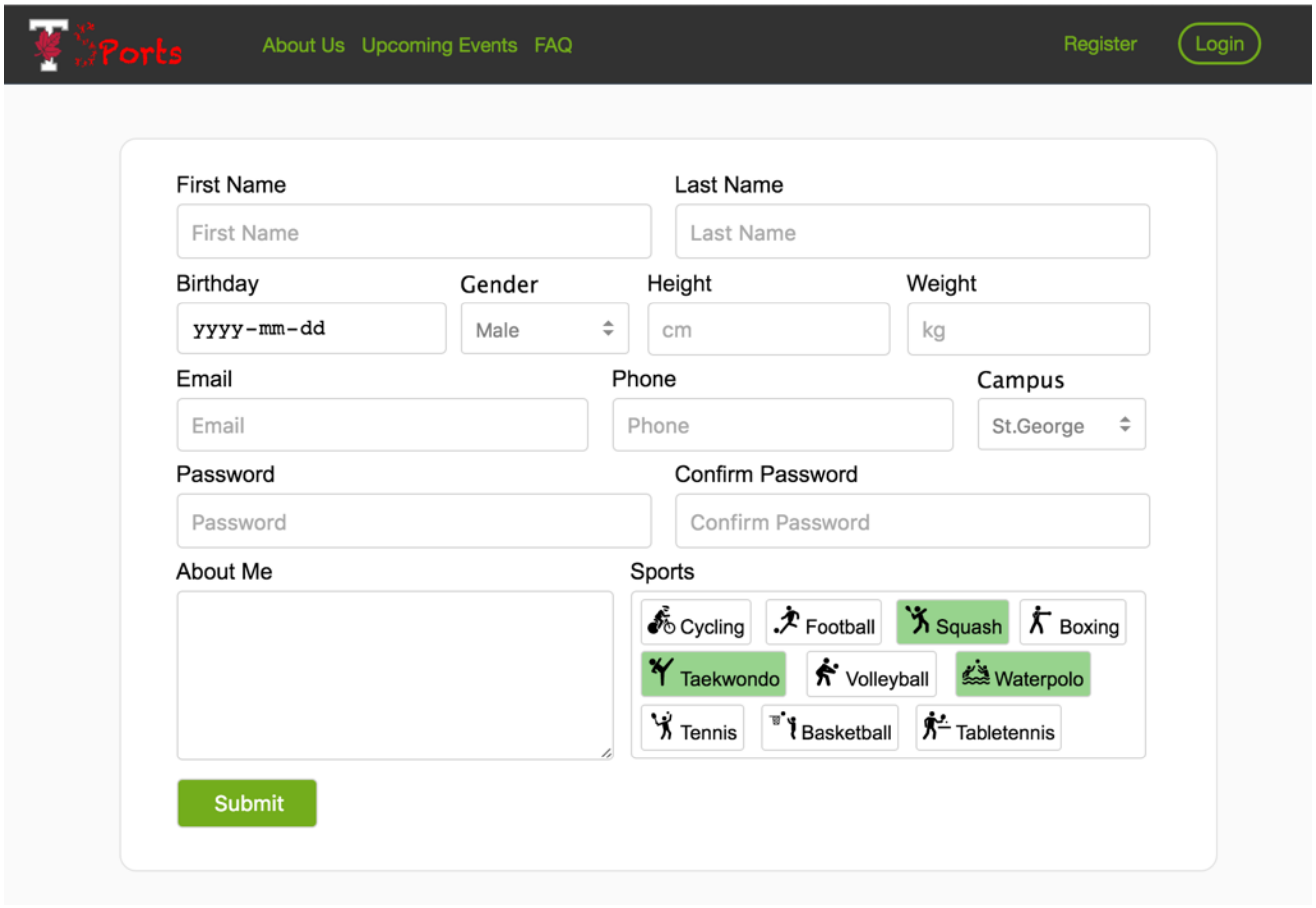
14:00PM WATER-POLO
Location: AC Benson Pool
12 People Attending 8 Spots left!

10:05AM OUTDOOR SOCCER
Location: AC Soccer Field
5 People Attending 5 Spots left!

10:00AM TENNIS
Location: Harthouse Tennis Court
1 People Attending 1 Spot left!

When a student initially enters www.tsports.ca, the above page is shown. There is an "About Us" and a "FAQ" in the navigation bar for a student to know more about TSports. If a user was already previously logged in, they would be directed automatically to their profile page. New users have two options. They can either click on register to create a new account or they can click on login and login via their Facebook account.

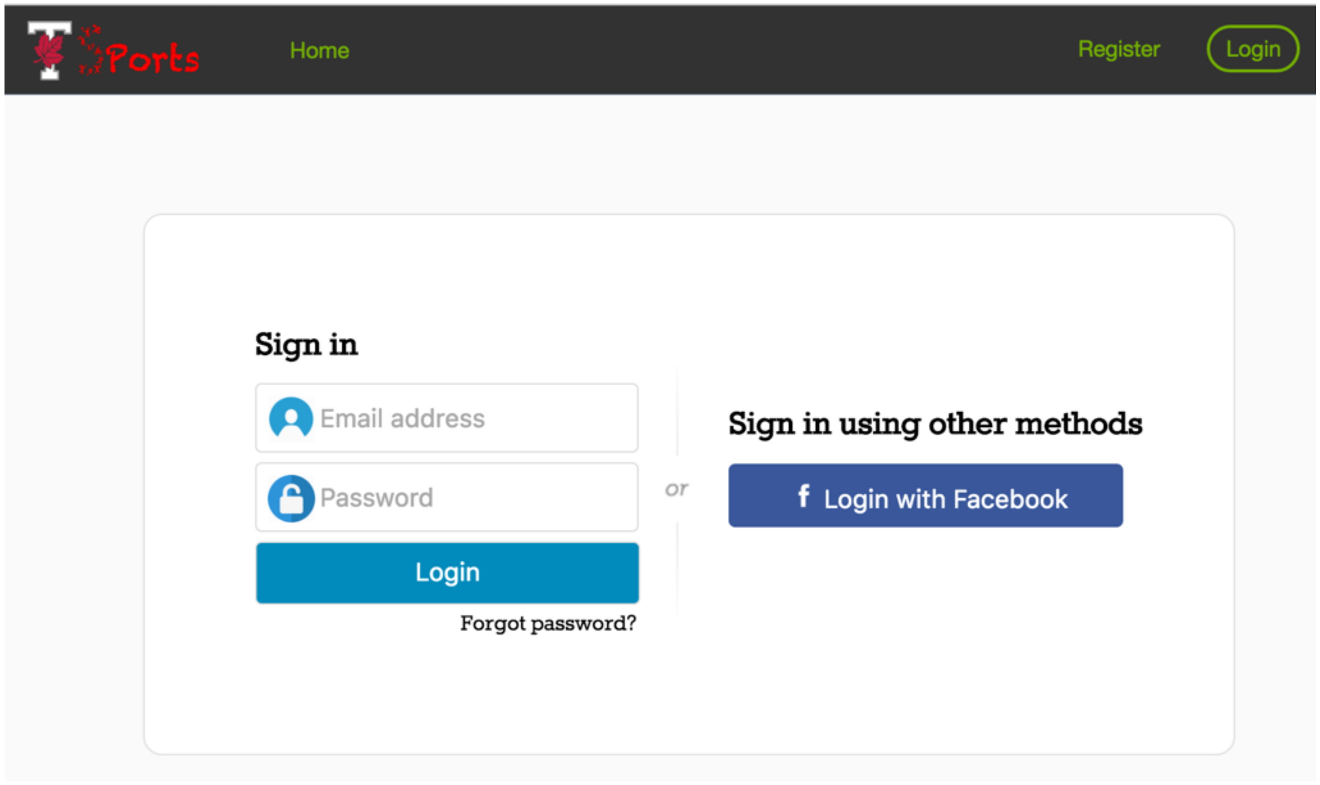
B. Registration Page (registration.html)



First Name: First Name
Last Name: Last Name
Birthday: yyyy-mm-dd
Gender: Male
Height: cm
Weight: kg
Email: Email
Phone: Phone
Campus: St.George
Password: Password
Confirm Password: Confirm Password
About Me
Sports: Cycling, Football, Squash, Boxing, Taekwondo, Volleyball, Waterpolo, Tennis, Basketball, Tabletennis

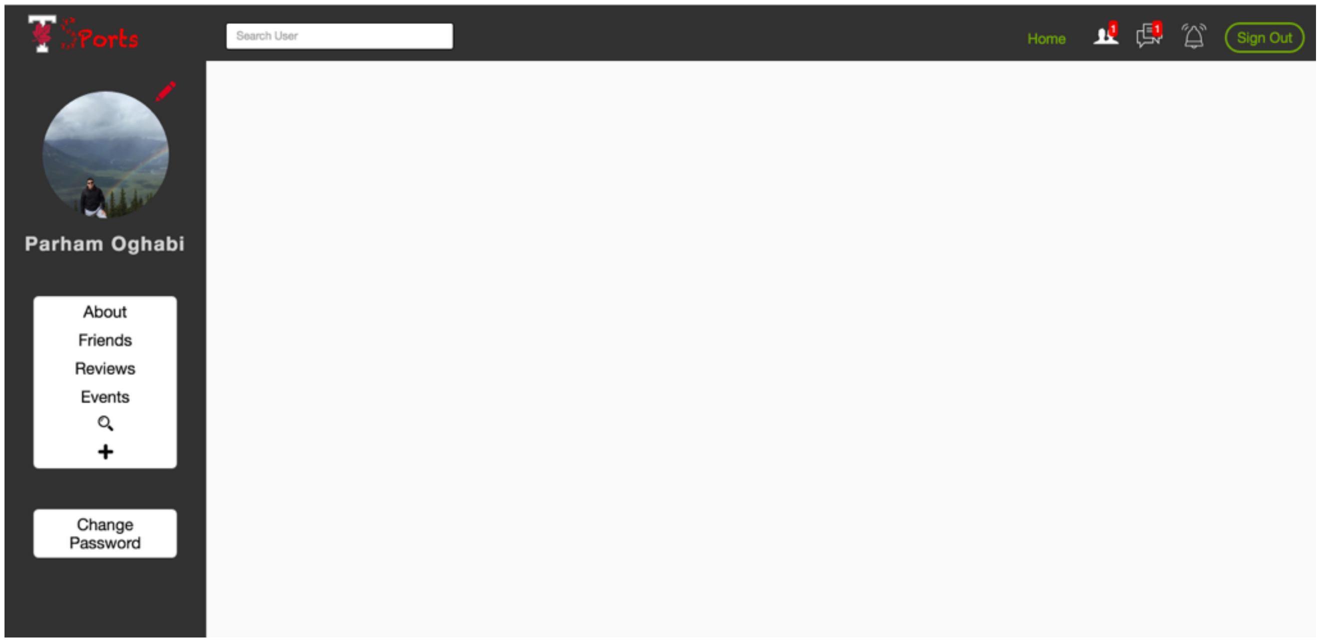
In order to be able to use more of the site's features, students should register by creating an account. Upon clicking "Register" or "Sign up!" in the introduction page, students will be taken to the registration page where they have to enter their information. Students should enter a brief description about themselves and select the sports that they are interested in order for us to use this information to suggest events that they would be interested in.

C. Login Page (login.html)



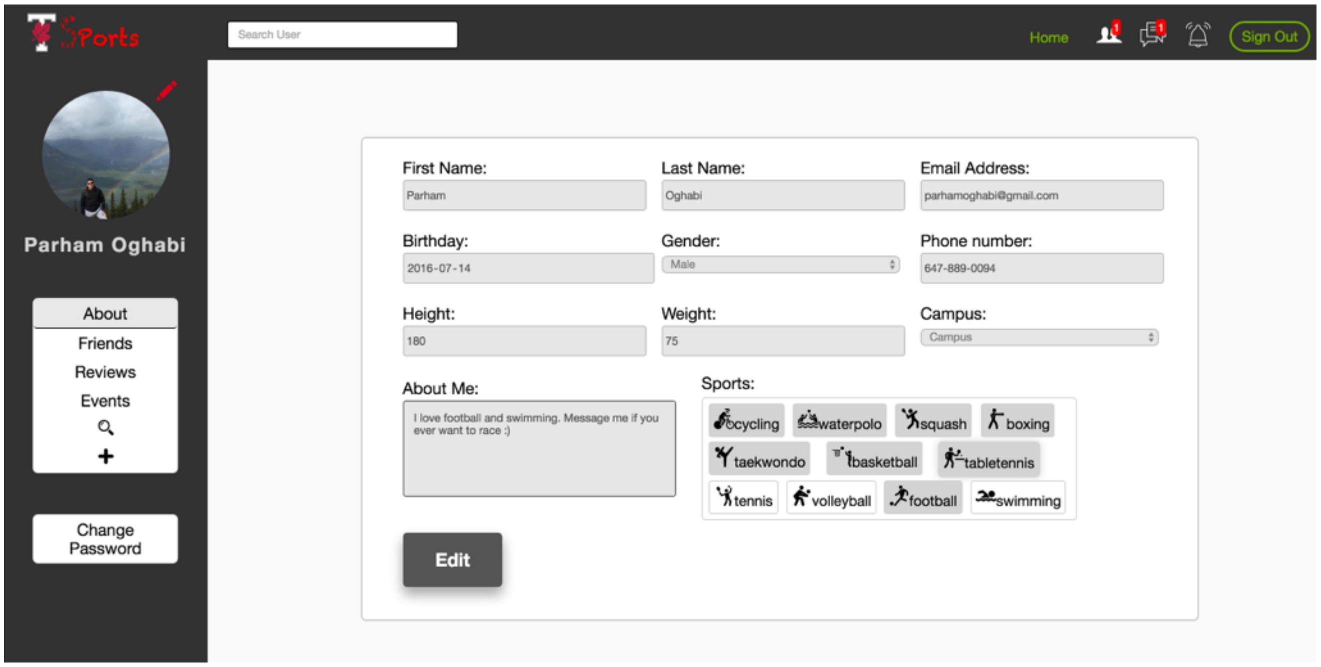
In order to enter the web application, users can either use their account which they created in the registration page or they can use third-party authentication mechanisms such as Facebook.

D. Home Page (Profile_SelfView.html)



When the user logs in, they are directed to their home page. The home page contains all the information related to that specific user. User's can upload their profile picture using the pencil icon next to their display picture.

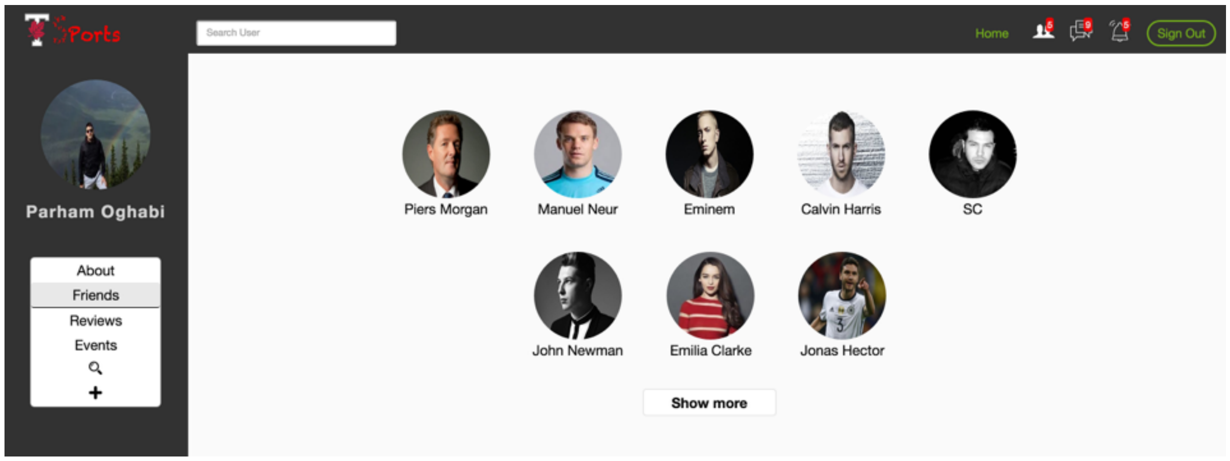
1. "About"



First Name: Parham
Last Name: Oghabi
Email Address: parhamoghabi@gmail.com
Birthday: 2016-07-14
Gender: Male
Phone number: 647-889-0094
Height: 180
Weight: 75
Campus: Campus
About Me: I love football and swimming. Message me if you ever want to race :)
Sports: cycling waterpolo squash boxing taekwondo basketball tabletennis tennis volleyball football swimming

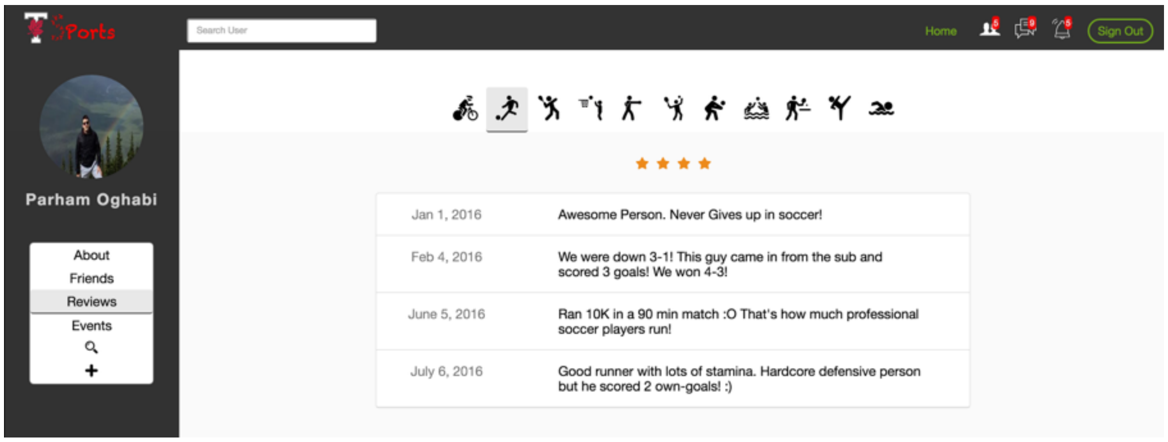
By clicking on the "About" button, the user can see all their personal information. They can edit these informations by clicking on the "Edit" button.

2. “Friends”



By clicking on the “Friends” button, the user can see a list of all their friends. Upon clicking on any of their pictures, they will be redirected to their friend’s profile.

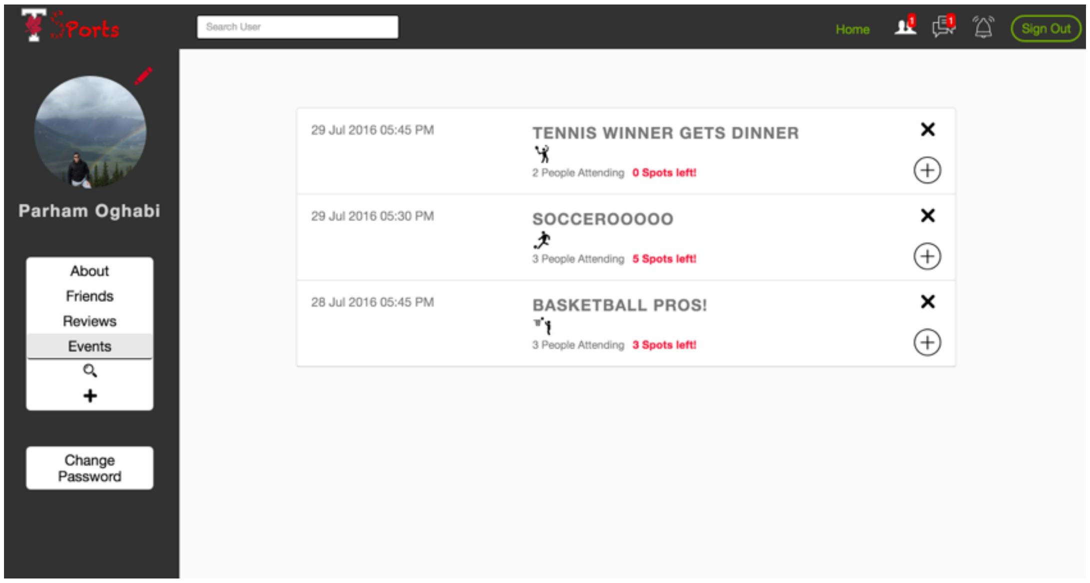
3. “Reviews”



Jan 1, 2016	Awesome Person. Never Gives up in soccer!
Feb 4, 2016	We were down 3-1! This guy came in from the sub and scored 3 goals! We won 4-3!
June 5, 2016	Ran 10K in a 90 min match :O That's how much professional soccer players run!
July 6, 2016	Good runner with lots of stamina. Hardcore defensive person but he scored 2 own-goals! :)

By clicking on the “Reviews” button, a list of all the sporting events the user has participated and was rated in will be shown on the top. The user has a different rating and set of comments in each of the sporting events they attended. They can click on different sporting events to see their rating and the comments left by others. These anonymous ratings and comments can be left only by other users who attended the same sporting event as the user. In this case, the user has received an overall rating of 4/5 stars in football. These publicized ratings and comments in each sporting event allows other users to deem the user’s proficiency level in a sport and make informed decisions. Ratings are based on a 5-star system and comments define the user’s overall performance.

4. “Events” + “Rating”

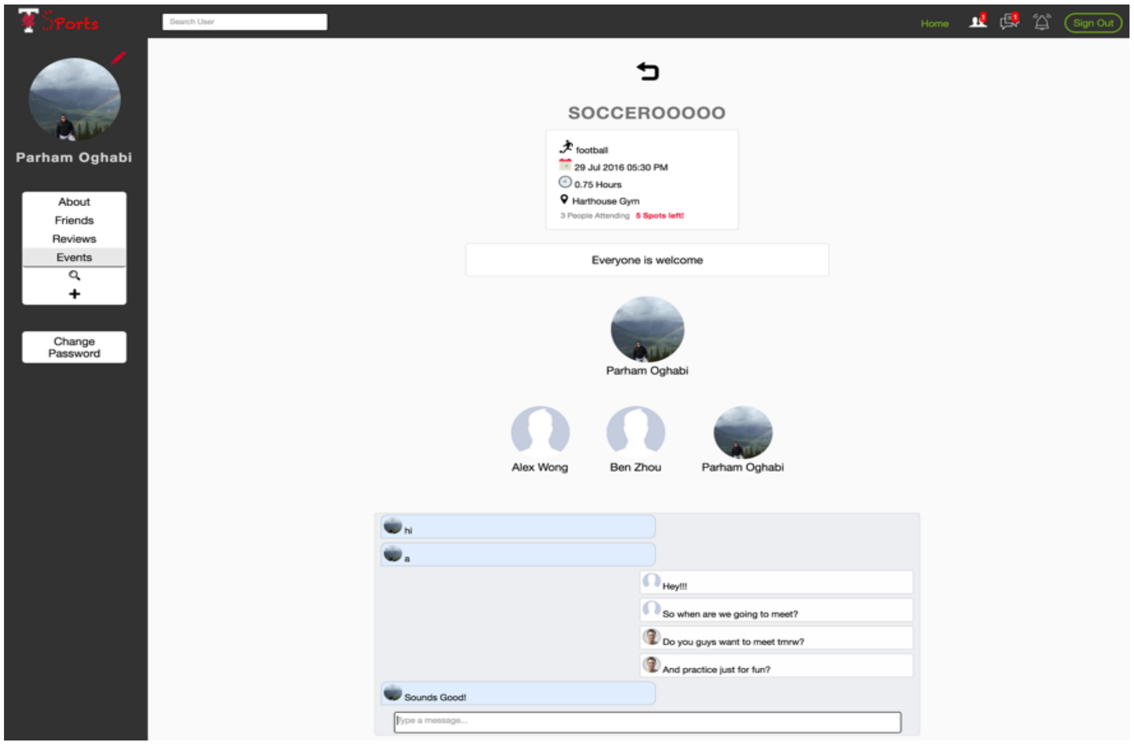


29 Jul 2016 05:45 PM — TENNIS WINNER GETS DINNER — 2 People Attending 0 Spots left!

29 Jul 2016 05:30 PM — SOCCEROOOOO — 3 People Attending 5 Spots left!

28 Jul 2016 05:45 PM — BASKETBALL PROS! — 3 People Attending 3 Spots left!

By clicking on the “Events” button, a list of all the upcoming sporting events the user has joined and will attend is shown. The user can view the event details by clicking the “+” button and can leave the event by clicking the “x” button. If the user is the admin of the event (the person who created the event), then upon pressing “x”, the event will be cancelled.



SOCCEROOOOO

football
29 Jul 2016 05:30 PM
0.75 Hours
Harthouse Gym
3 People Attending 5 Spots left!

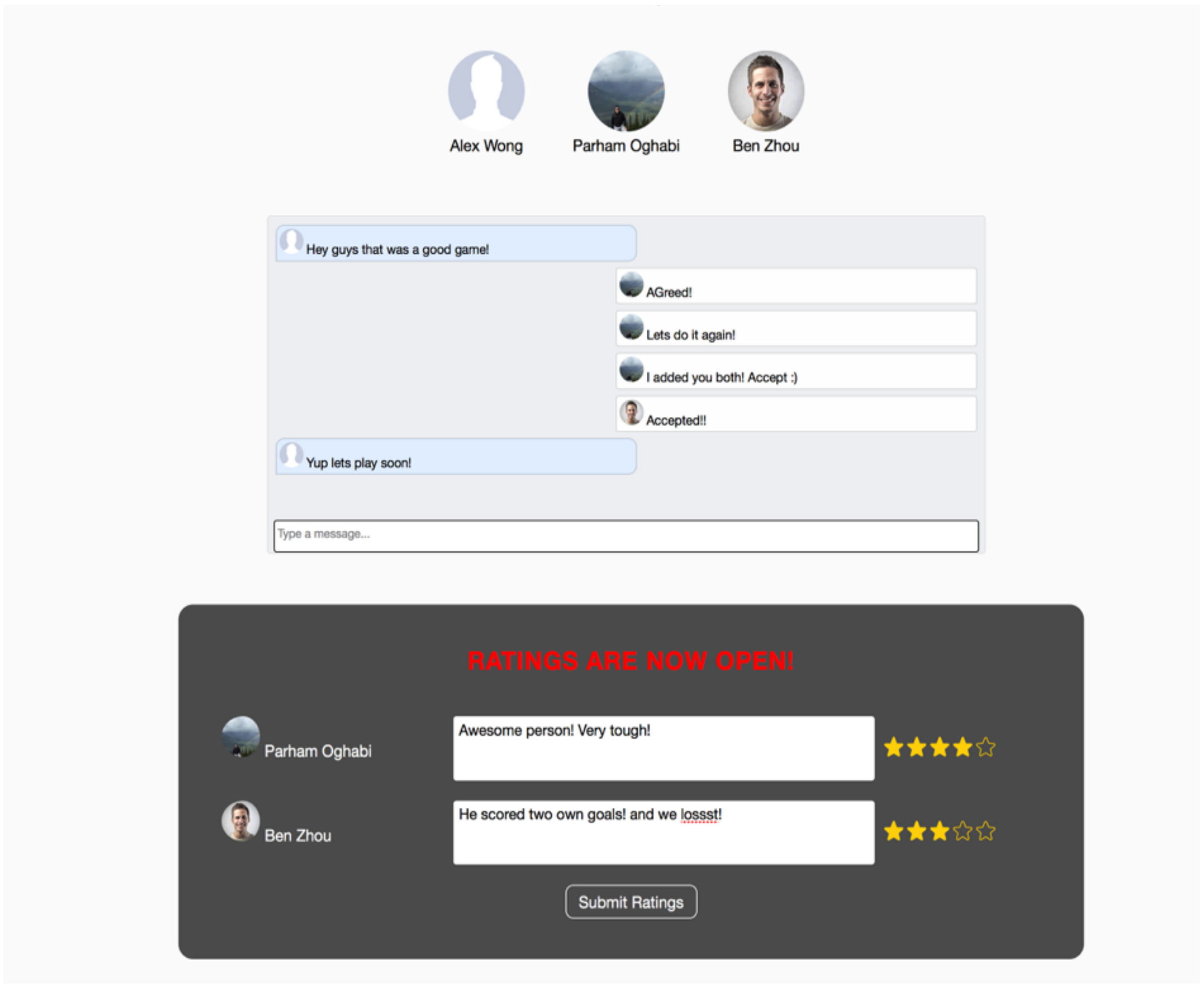
Everyone is welcome

Parham Oghabi

Alex Wong
Ben Zhou
Parham Oghabi

hi
a
Hey!!!
So when are we going to meet?
Do you guys want to meet tmrw?
And practice just for fun?
Sounds Good!
Type a message...

Upon clicking on a specific event, details for that event are shown. Details include the event name, event sport type, date and time, event location, description of the event, number of people attending, and the number of spots left for that event. In addition, the event admin is shown (the user who created the event), the users who are attending the event are shown, and a group chat between users who are attending that event.



Hey guys that was a good game!

AGreed!

Lets do it again!

I added you both! Accept :)

Accepted!!

Yup lets play soon!

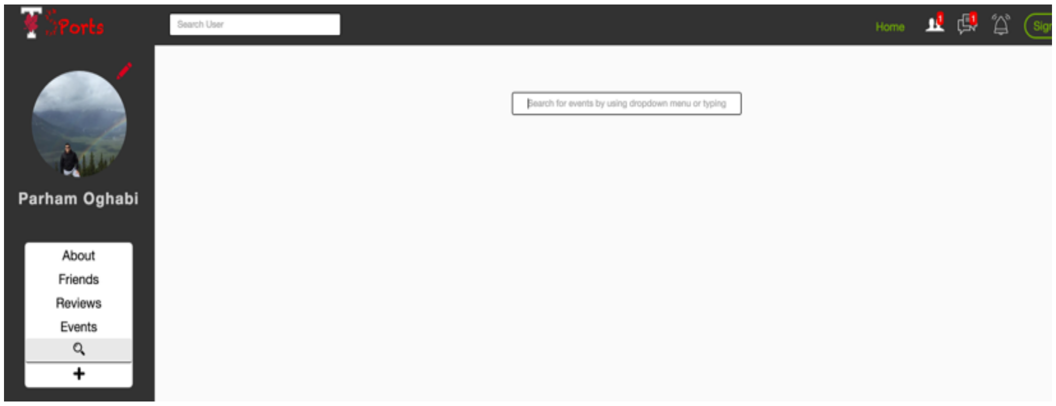
RATINGS ARE NOW OPEN!

Parham Oghabi — Awesome person! Very tough! ★★★★☆

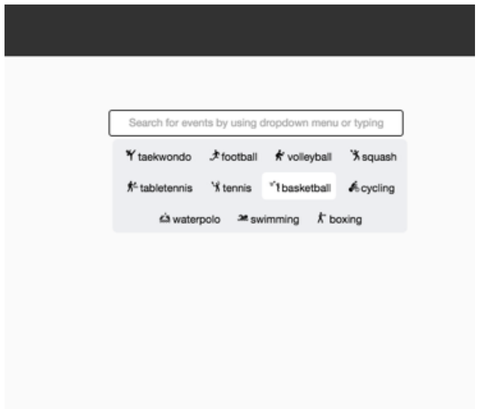
Ben Zhou — He scored two own goals! and we lossst! ★★★☆☆

After the event has ended. A rating and comment box will become available below the group chat for that event. Users can rate the other users who were in the same event and submit their ratings.

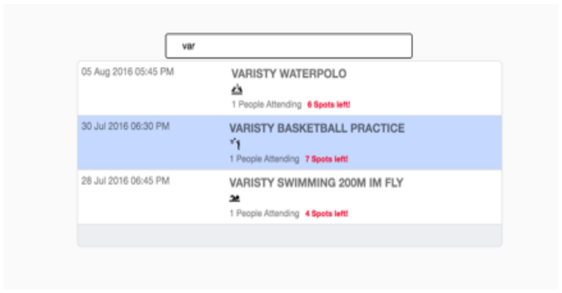
5. “Search Event”



By clicking on the “search” button, an input form will be displayed and the user can use the search bar. Users can either search by typing or by using a drop-down menu of the different sports. A smart recommendation system is implemented to rank the results shown based on the user’s query.



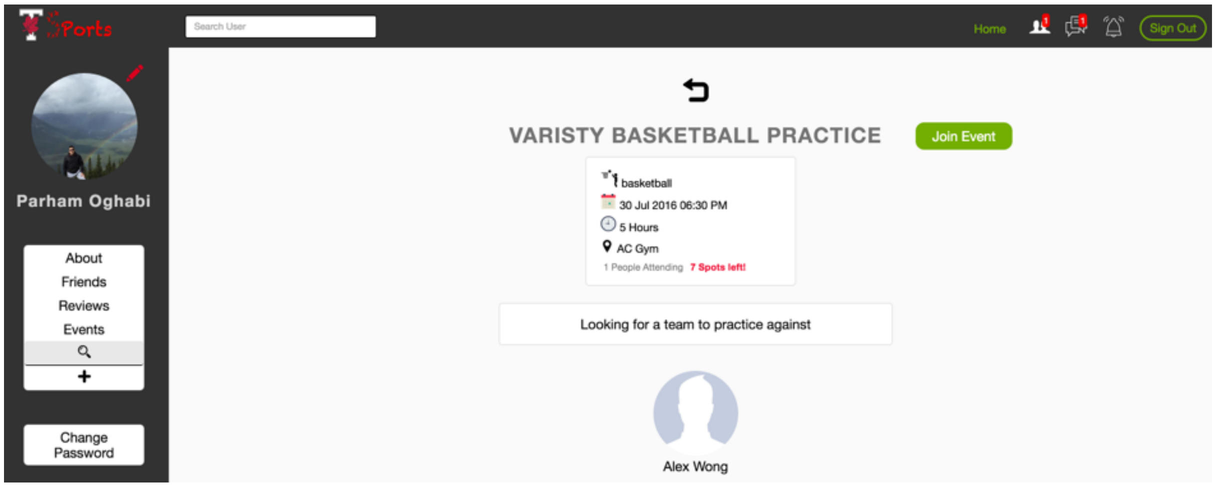
Users can click on the search bar and a drop-down menu will appear containing all the different sports. Upon selecting a sport, the drop-down menu will disappear and a list of related events will be shown.



var

05 Aug 2016 05:45 PM	VARISTY WATERPOLO 1 People Attending 6 Spots left!
30 Jul 2016 06:30 PM	VARISTY BASKETBALL PRACTICE 1 People Attending 7 Spots left!
28 Jul 2016 06:45 PM	VARISTY SWIMMING 200M IM FLY 1 People Attending 4 Spots left!

Users can also type in the search bar such as an event name or a sport type and a list of related events will be shown.



VARISTY BASKETBALL PRACTICE

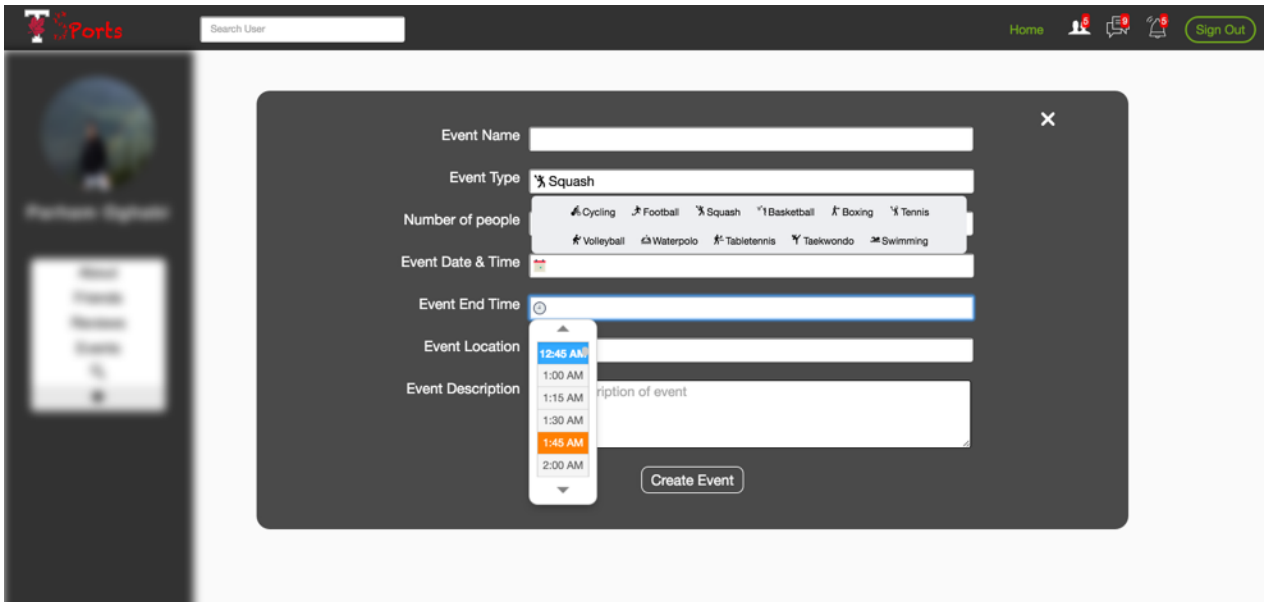
basketball
30 Jul 2016 06:30 PM
5 Hours
AC Gym
1 People Attending 7 Spots left!

Looking for a team to practice against

Alex Wong

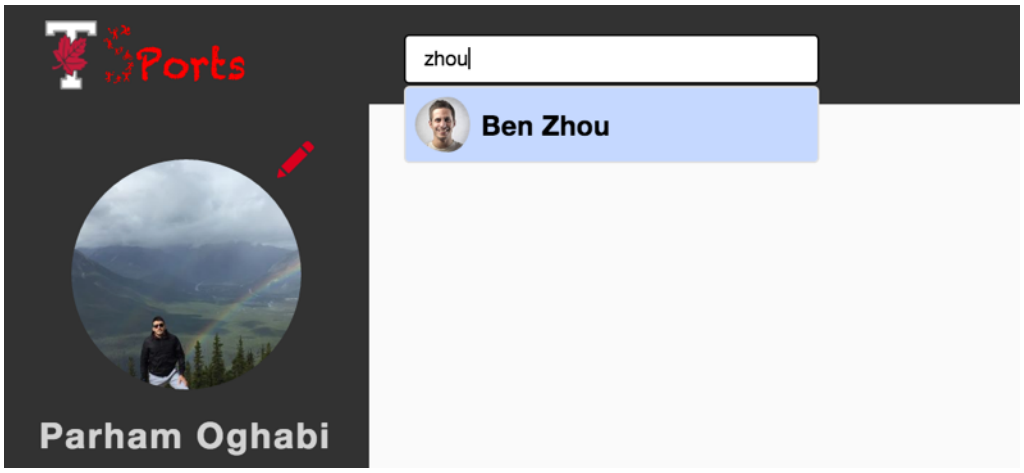
Upon clicking on a specific event, the details of that event, as mentioned in the previously, will be shown and the user will be given the option to join the event. If the user has not joined that event, they will only be able to see the details of the event mentioned above except the group chat.

6. “Create Event”



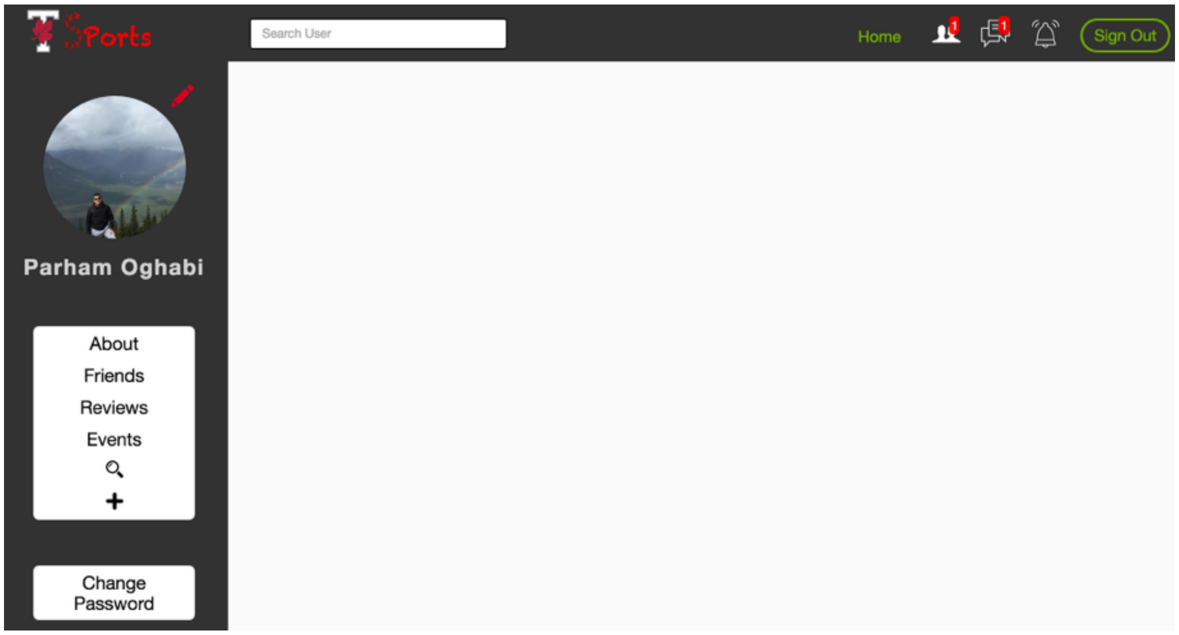
By clicking on the “+” button, the background is blurred and a form is created. The user can then create an event by filling out the form. Upon clicking “Create Event”, the sporting event is created and the user becomes the admin of that event.

7. “Search User”

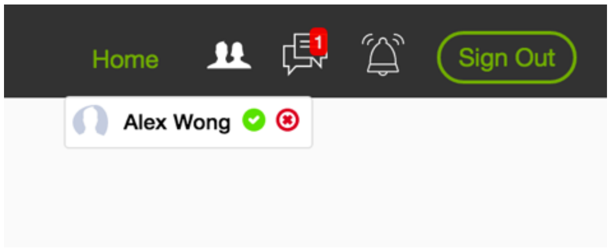


The “Search User” input in the top navigation bar allows the user to search for other users by name and to click and view their profile.

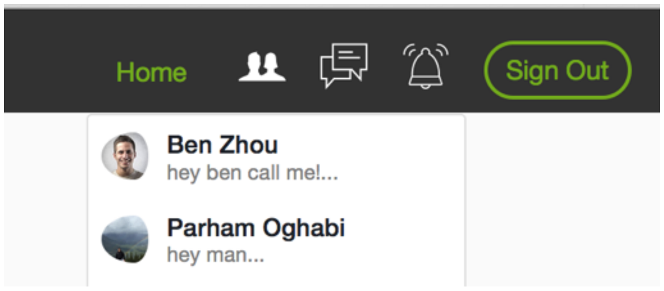
8. Navigation Bar Icons + Messaging



The home page comes with a sticky navigation bar. The right side of the nav bar displays information such as the number of friend requests (if any), number of messages (if any), and any notifications.



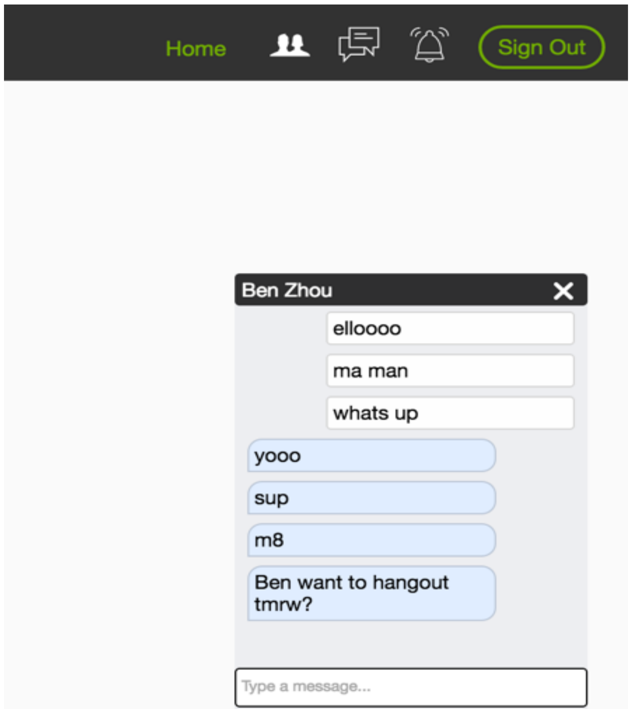
A drop-down menu will be shown upon clicking on any of these 3 icons. The above figure shows that we have a friend request when the friend icon is clicked.



Ben Zhou
hey ben call me!...

Parham Oghabi
hey man...

Upon clicking the message icon, a drop-down menu will be shown containing a list of all the people we have messaged.



Ben Zhou

elloooo

ma man

whats up

yooo

sup

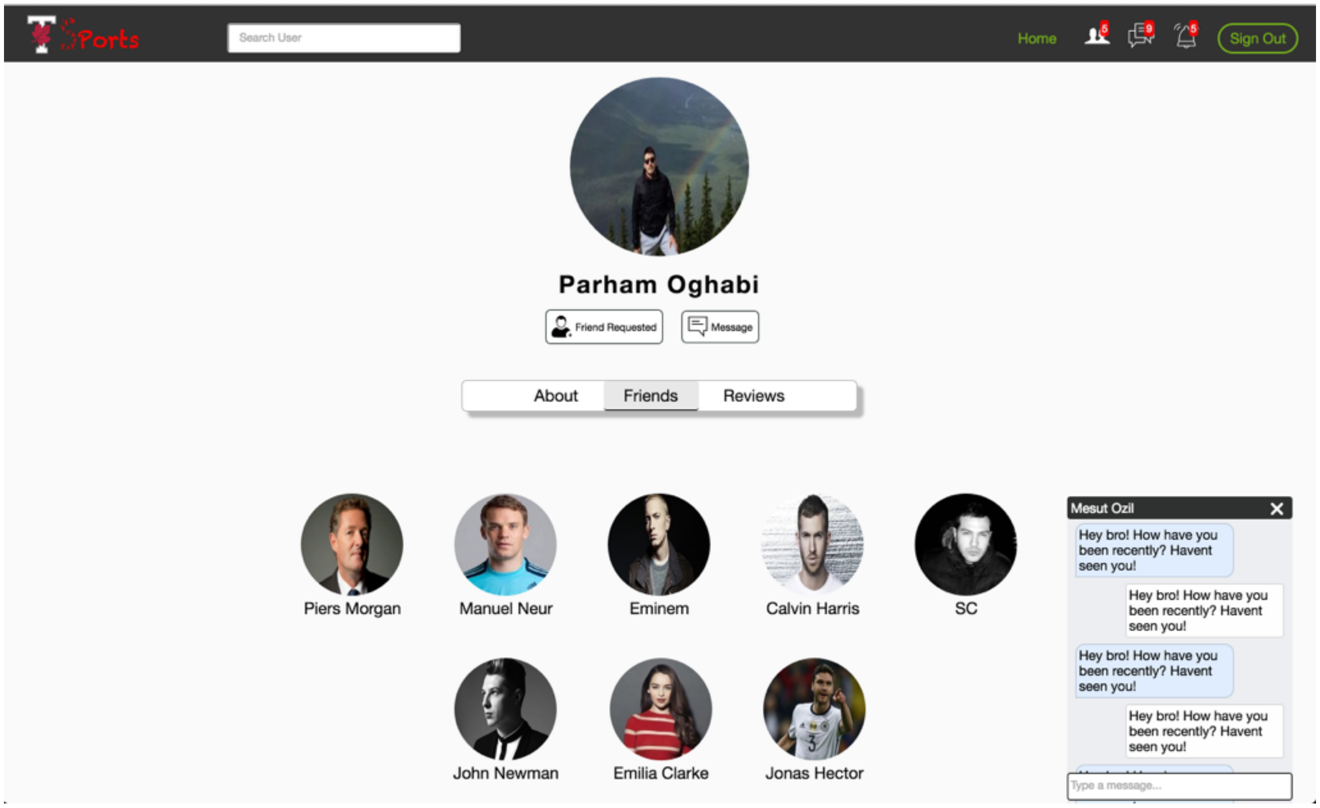
m8

Ben want to hangout tmrw?

Type a message...

Upon choosing a message from the dropdown menu under the messages icon, a ChatBox will open on the bottom right of the viewport and you will be able to chat with that user. In any page and at any moment, the user can use the navigation bar to chat with other users.

E. Other User’s Page (Profile_OthersView.html)



Hey bro! How have you been recently? Havent seen you!

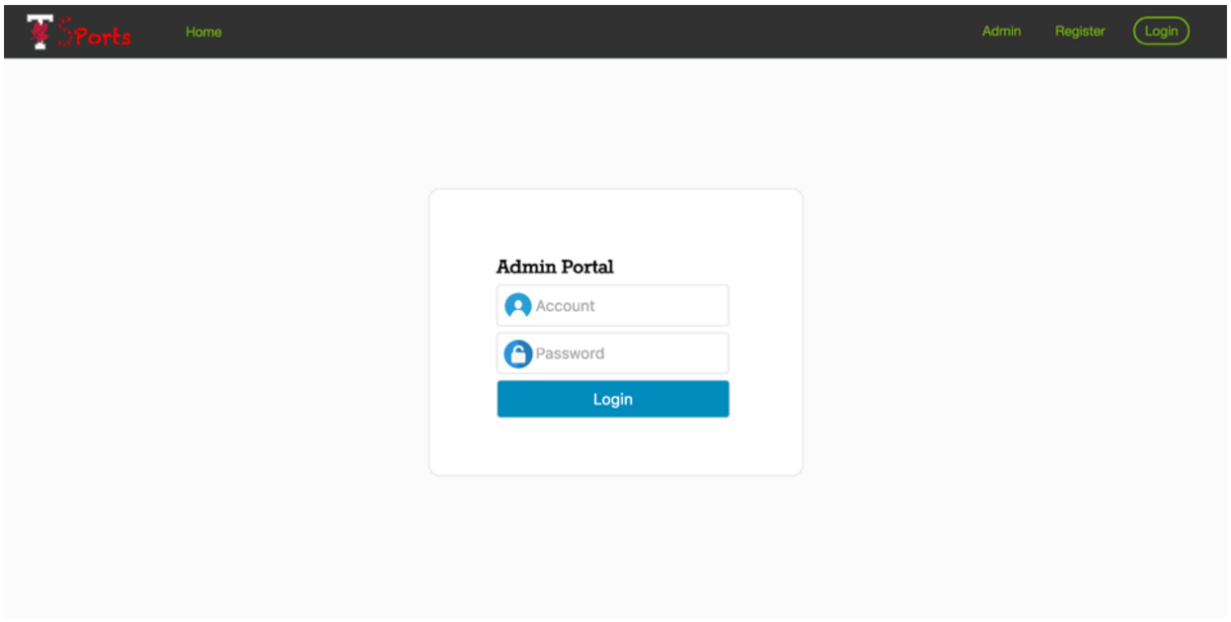
Hey bro! How have you been recently? Havent seen you!

Hey bro! How have you been recently? Havent seen you!

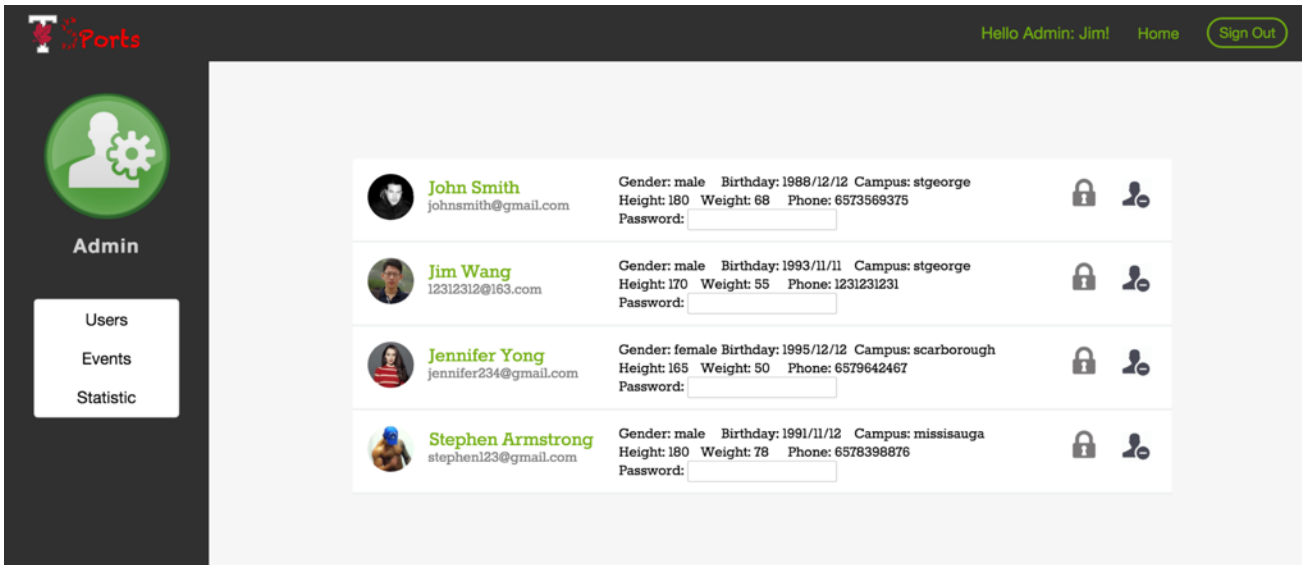
Hey bro! How have you been recently? Havent seen you!

When a logged in user clicks on a picture of another user, whether on an image or using the search user bar, they will be redirected to that user’s profile page. The above picture shows a user’s profile as seen by other users. A logged-in user can view another user’s “About” information, their friends, the reviews (rates and comments) they have received in different sporting events. They can also add them as a friend or message them using the one-to-one chat box. The logged-in user has their navigation bar always on the top and can return to their homepage by clicking the “Home” button or the “TSports” logo.

4. Admin



Our web application will support an admin view of the system. Our team will be acting as the admin and the admin will be accessible via www.tsports.ca/admin. Upon entering that URL, you will be prompted to enter a username and password which is known only by our team



John Smith
johnsmith@gmail.com
Gender: male Birthday: 1988/12/12 Campus: stgeorge
Height: 180 Weight: 68 Phone: 6573569375
Password:

Jim Wang
12312312@163.com
Gender: male Birthday: 1993/11/11 Campus: stgeorge
Height: 170 Weight: 55 Phone: 1231231231
Password:

Jennifer Yong
jennifer234@gmail.com
Gender: female Birthday: 1995/12/12 Campus: scarborough
Height: 165 Weight: 50 Phone: 6579642467
Password:

Stephen Armstrong
stephen123@gmail.com
Gender: male Birthday: 1991/11/12 Campus: missisauga
Height: 180 Weight: 78 Phone: 6578398876
Password:

Upon logging in, the admin is redirected to the dashboard. Our team will be able to view a table of all the users and events and be able to delete users and reset their passwords or delete events. Admins will also be able to see aggregate information such as the total number of events, the total number of users, the most desired sporting event, and other useful analytics.

5. Unit Testing

Our testing was done using Mocha.

Test file: test/test_index.js

Supported by modules such as mocha, chai and supertest, the unit tests test the back-end (server-side) functions. Here, tests test the main basic features of website. The other features, which are not included in the list of unit tests, are described in the report in details.

How to run test file:

1. "npm install" to download modules (which are included in package.json).
2. Without running server, run test file by "npm test". It will automatically run the server and close the server after testing.

(NOTE: Please do not run the server when running test. It will throw error although unit tests run successfully)

Sample users for testing:

1. {ID: 183, name: tester two, email: tester2@tsports.com, password: 123123, …}
2. {ID: 184, name: tester three, email: tester3@tsports.com, password: 123123, …}

Sample friend relationship for testing:

- ID:183 and ID:184 are friends.

Sample events for testing:

1. {eventid:18, name: Event_Test, eventtype:tennis, eventadminid: 183,…}
   - Members: ID:183 and ID:184.
2. {eventid:41, name: Event_Test2, eventtype:tennis, eventadminid: 184,…}
   - Members: ID:184.
3. {eventid: <varies>, name: TEST_CREATE, eventtype: football, eventadminid:183, …}
   - This third sample event is created by "Create Event" test. One is created each time we run test.

List of unit tests and brief descriptions:

A. **Registration feature**

   i. Request /UserRegistration: message -> ‘ok’ :
   Using the sample user info defined in test_index.js, try a registration by POST request with URL of “/UserRegistration” to the server. If success, the server will send the message, “ok”.

   ii. Registration with email of existing user: message -> ‘emailexists’:
   Try to register with the same info used above. This test must get the message, “emailexists” from the server since the user was registered already. This test ensures that only one user can be created with one email in our website.

   iii. Delete existing sample user from the database:
   Delete the sample user from the database to prevent an error when running test next time.

B. **Login feature**

   i. Login with the normal existing user: message -> ‘ok’:
   Login with the sample user (name: tester two; email: tester2@tsports.com; password: 123123). The server sends “ok” on success.

   ii. Login with non-existing user: message -> ‘invalidlogin’:
   Try to login with the wrong email or password. The server sends “invalidlogin” on failing of login. This test must get “invalidlogin” message.

   iii. Logout: /SignOut
   Sign out the currently login-ed user.

C. **Get/change the user information**

   i. Get the user's basic info when login: /GetLoginUserInfo:
   Request for the sample user’s information (profile image, name, …etc) to display them on profile page. Check if the correct information is returned by the server.

   ii. Get the user's friends list: /GetUserFriends:
   Request for the sample user’s friends list to the server.

   iii. Get all the users who are attending event: /GetEventUsers:
   Request for a list of users who are attending the sample event.

   iv. Get friend's profile: /GetDisplayedProfileInfo:
   Request for a very basic information of the friend to the server.

   v. Get friend's profile about info: /GetUserAboutInfo:
   Request for the detailed information of the sample user’s friend (ID=184) (i.e. “about”).

   vi. Change the user's password: /ChangePassword:
   Change the current password to the new password (set it back after test).

D. **Event related feature**

   i. Create Event: /CreateNewEvent:

Create another sample event using the sample user's account. It will redirect to " /Profile_SelfView.html?EventSuccess=yes". Test successes on match.

E. **Search related features**

i. Search User: /SearchUsers:

Search "tester" on the user search bar and check what we get. First one must be "tester three" who is a friend of the sample user (i.e. the second sample user).

ii. Search Events by click: /SearchEventsByClick:

Search "tennis" on the event search bar. "Event_Test2" must be a first one (the second sample event created by "tester three").

F. **Live messaging features**

i. Get log of group messages within eventID=18: /GetEventMessages:

Two messages are sent for the sample. Test will success on match.

ii. Get log of 1-to-1 messages with user=184: / GetMessageHistoryWithUser:

Two messages are sent for the sample. Test will success on match.

G. **Security: SQL injection**

i. SQL injection attempt: injection should fail:

We tried with "*{"email":"no@no.com and password = '123333'); DELETE FROM Users WHERE id=64; -- ", "password":"123123"}*" and query doesn't affect Users table. The server only returns "invalidlogin" message.

6. Security

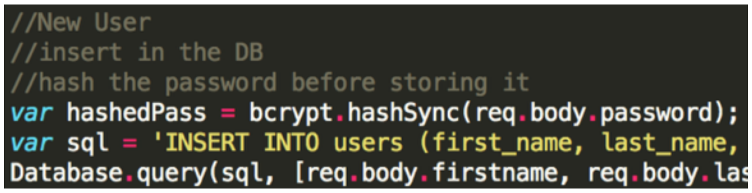
A. Hashing the Password

Reason: The database of the website may be compromised and its data may be obtained by someone else. So our team decided to store the hashed password instead of the plain text for each user into the database.

Method: Bcrypt-nodejs is a JS bcrypt library for NodeJS providing various APIs for hashing the password, and it's based on the one-way function which means the attacker cannot get the plain password even if the database is compromised.

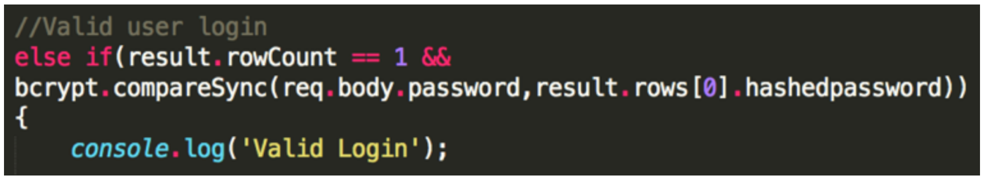
- Bcrypt.hashSync("plain_password") – we use this function to hash the password before stored to the database when the user registers to our website.
- Bcrypt.compareSync("plain_password", "hash_password") – we use this function to compare the password input from the user when the user logins our website with the hashed password corresponding to this user in the database. If both passwords match, then this function will return true and then we allow the user to login and vice versa.

Snippet: Hash the password before storing into database when the user registers:



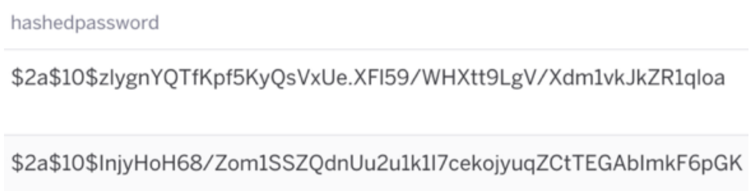
```
//New User
//insert in the DB
//hash the password before storing it
var hashedPass = bcrypt.hashSync(req.body.password);
var sql = 'INSERT INTO users (first_name, last_name,
Database.query(sql, [req.body.firstname, req.body.las
```

Compare the password input from the user with the hashed password when the user logins:



```
//Valid user login
else if(result.rowCount == 1 &&
bcrypt.compareSync(req.body.password,result.rows[0].hashedpassword))
{
    console.log('Valid Login');
```

Hashed password in the database:

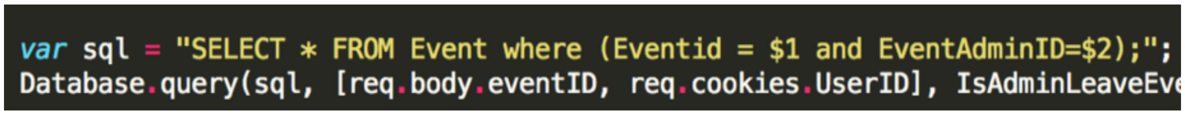


hashedpassword
$2a$10$zlygnYQTfKpf5KyQsVxUe.XFI59/WHXtt9LgV/Xdm1vkJkZR1qloa
$2a$10$InjyHoH68/Zom1SSZQdnUu2u1k1I7cekojyuqZCtTEGAblmkF6pGK

B. Prevent SQL Injection using prepared statements

Reason: Attacker can concatenate malicious SQL string into the database query string to steal data from database or even destroy the whole database. So our team decided to use prepared statements to prevent SQL injection. Prepared Statements are resilient against SQL injection, because query and data are sent to the SQL server separately.

Method: The way we implement the prepared statements is to send our query program to the database server first, like



```
var sql = "SELECT * FROM Event where (Eventid = $1 and EventAdminID=$2);";
Database.query(sql, [req.body.eventID, req.cookies.UserID], IsAdminLeaveEv
```

and use some placeholders like $1, $2 to wait for the executed data from client-side, then next time we parse and send the data coming from client-side to the database program we just built up so that the attackers cannot alter our program and do any harm.

7. Performance Enhancements Upon Testing

A. PNG vs SVG

Initially lots of PNGs and JPGs were being used. After our initial iteration of the web application, we realized that our images folder summed up to ~4MB. We replaced our PNGs with SVGs and our images folder decreased to 1.7MB, which is a significant improvement. On average, each of our PNG icons were 40KB whereas each SVG was only 4KB.

B. Content Delivery Networks

Initially all external libraries such as the jQuery, jQuery UI, and jQuery cookie plugin were put into the assignment folder. This decreased the loading speed of the website. Upon using a CDN, it resulted in an increase in the speed with which the content is delivered to the users. This is because a server that is closest to the user is determined by the CDN and optimizes the speed with which content is delivered. The files were deployed live and the loading speed of the website was tested from multiple locations across the world using https://geopeeker.com. The speed was significantly faster when using a CDN.

C. Polling vs WebSockets

Initially, the friend and message notifications and the messaging was done through polling. Consistent AJAX requests were sent at fixed time intervals to see if any messages or friend requests were received. After interacting with website, we noticed that the delay and latency in the website was slightly noticeable. We changed our server side code to use web sockets instead to enable real-time messaging and real-time notifications. Sockets have 2 way connections and require fewer network operations to send a packet because the webSocket connection is already open and the client is listening on the connection. However, using Ajax, connection setup and tear down is consistently required since HTTP is non-persistent. This was our biggest enhancement to the website, both as a feature and for performance.

8. Enhancements

A. Usage of combination of REST API/AJAX and WebSockets

Using a combination of REST API and WebSockets allowed us to optimize our website’s speed and prevent the website from making unnecessary HTTP requests. On a large scale, if we were to continue using polling for messaging and notifications, not only would the messages and notifications not be displayed to users in real-time, but there would be a large load and strain put on the server.

B. Real-Time Messaging

The usage of WebSockets to enable real-time messaging between users rather than the usage of AJAX and long polling. The one-to-one messaging feature can also be used on any page and at any time.

C. Event Group Chat Real-Time Messaging

When a user opens an event, they will automatically join a room on the server dedicated to that event only and the client will listen to any messages received from other users in that event in real-time.

D. Real-Time Notifications with sound (Friend Request and Messaging)

Upon receiving a friend request or a message from another user, the user will be notified by a notification sound and the number of friend requests of messages will be updated in the browser in real-time.

E. Easy to use Interface

The web application is very easy to use and has many features which are very common across popular social networks across the world.

F. Error-Catching

Necessary error catching features were implemented such as:

- Trying to login with an invalid username and password
- User trying to change their password but entering two different passwords upon confirmation of the new password.

9. Github Repository + Live Link + Admin Account

GitHub Repository Link:

https://github.com/hiei23/csc309a4

Username: csc309a4TA
Password: ta123456

Web App Live Link:

www.tsports.ca

Username: csc309a4ta@gmail.com
Password: ta123456

Web App Admin Account:

www.tsports.ca/admin OR localhost:3000/admin

Username: jimwong
Password: 12345678