CS353
DATABASE SYSTEMS

Final Report

Project Topic: **Social Network For Check-Ins**

Project Name:**'WhereTo'**

Project TA: Fuat Basık

Link to webpage: https://github.com/FaaizHaque/CS353-Group29

A Project By Group 29

Hamza Saeed Khan - 21500570 - Section 3

Faaiz Ul Haque - 21503527 - Section 3

Usama Saqib - 21500116 - Section 2

Askari Iqbal - 21504228 – Section 3

Contents

1. Introduction

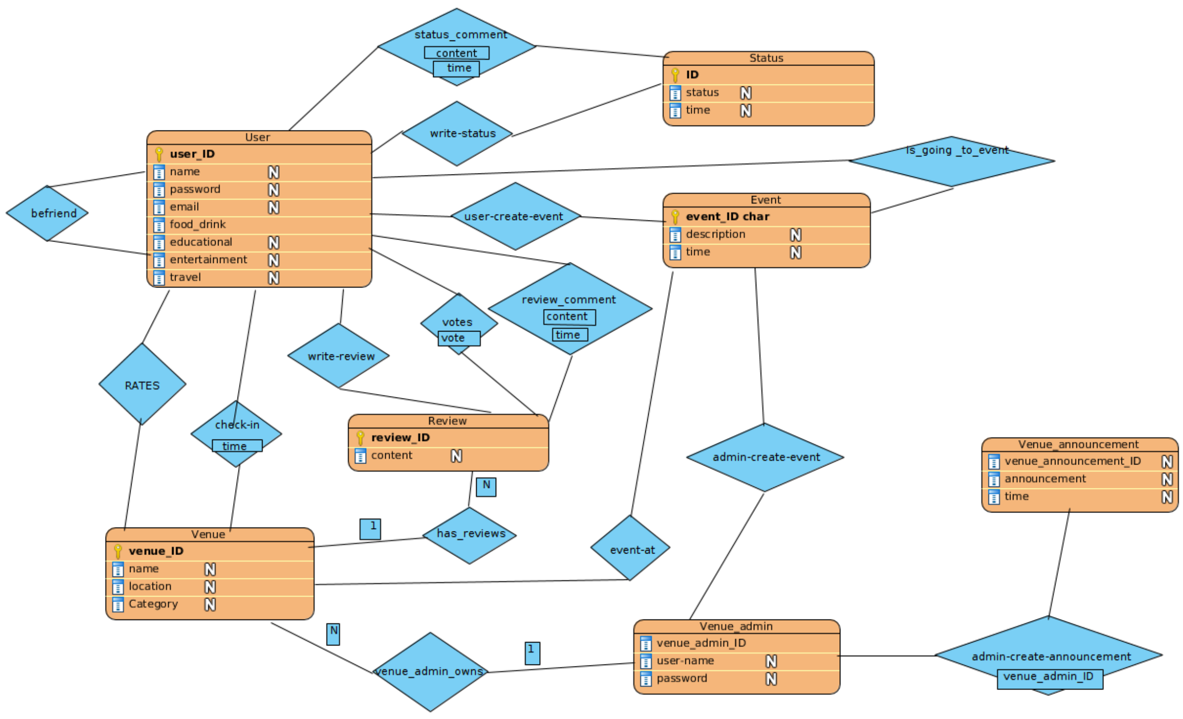
This is our final report for our project called 'WhereTo' which is a social network application for user check-ins. *WhereTo* allows it's users to check-in to a list of various venues in their areas, while interacting with their friends and sharing their experiences while using our application. *WhereTo* provides a convenient and hassle-free platform with user-friendly interfaces. A more extensive description of how our application functions can be found in section 6. User Manual, of this report. This report will also show our final E-R diagram after we made several changes in contrast to our proposal and design report. Furthermore, a list of our final tables will be stated, specifically those table schemas including primary and foreign keys. Our report also includes implementation details of our application, in specific what type of environment, frameworks, and languages we used and potential issues that appeared during implementation and how we solved them. Finally, the advanced database features we used will also be stated with their significance in our application.

Individual Contributions:
Most of the time when we worked on the project, we worked together, side by side, so you could say that the contribution was equally divided, because we would help each other do almost everything even if the jobs were being split in some way
In particular, tasks were assigned as follows:
Faaiz: MockUps/GUI using html and python
Askari: Advanced database components, relational schemas for the ER
Hamza & Usama: Worked together on completing E-R model and the SQL statements in mySQL, (as in creating tables, etc)
These are some tasks we assigned in specific, however, this does not mean one person did not at least help in someone else's job. The whole project was a team effort.

We all worked on compiling it all together (as in the integration of SQL, python, HTML, CSS)

2. Revised E-R Model

We received some feedback regarding our initial E-R model stating the overall design seemed to simple, and required more entities. We added the following entities to our original diagram: Status, Event, Venue_Announcement, Venue_Admin and Review. Furthermore, we created many more relations to further define these entities and how they interact with each other. We also had incorrect notation of many-to-one, one-to-one, and many-many entities and also improved on those. We added weak-entities and total participation where applicable.



3. Final List Of Table Schemas

1 **User**(user_id, name , password, email, food_drink, educational, entertainment, travel)
Domain of name must have max of 30 characters and not null
Domain of 'password' must have a minimum of 8 characters
'user_id' is a candidate key and the selected primary key

2 **Venue**(venue_id, name, location, category)
Domain of 'category' has only four options, as listed in user entity's attributes (food_drink, educational, entertainment, travel)
'venue_id' is a candidate key and the selected primary key

3 **Review**(review_id, content)
'review_id' is a candidate key and the selected primary key

4 **Status**(status_id, status, time)

Domain of 'time' is standard to the 24 hour clock (i.e (0-23):(00-60))
'status_id' is a candidate key and the selected primary key

5 **Event**(event_id, description, time)
Domain of 'time' is standard to the 24 hour clock (i.e (0-23):(00-60))
'event_id' is a candidate key and the selected primary key

6 **Venue_Admin**(venue_admin_id, username, password)
    Domain of 'password' must have a minimum of 8 characters
    'venue_admin_id' is a candidate key and the selected primary key

7 **Venue_Announcement**(venue_announcement_id, announcement, time)
    Domain of 'time' is standard to the 24 hour clock (i.e (0-23):(00-60))
    'venue_announcement_id' is a candidate key and the selected primary key

8 **befriend**(friend_ID_1, friend_ID_2, friendship_status)
    Domain of 'friendship_status' is either friends or not friends (0 or 1 or bool)
    Primary key is friend_ID_1 and friend_ID_2 combined
    FK: 'friend_ID_1' referencing User (user_ID)
    FK: 'friend_ID_2' referencing User (user_ID)

9 **check-in**(user_ID, venue_ID , time)
    Domain of 'time' is standard to the 24 hour clock (i.e (0-23):(00-60))
    Primary key is user_ID and venue_ID combined
    FK: 'user_ID' referencing User
    FK: 'venue_ID' referencing venue

10 **write-status**(user_ID, status_ID)
    Primary key is a combination of user_ID and status_ID
    FK: 'user_ID' referencing User
    FK: 'status_ID' referencing Status

11 **status-comment**(status_ID, user_ID, content, time)
    Domain of 'time' is standard to the 24 hour clock (i.e (0-23):(00-60))
    FK: 'user_ID' referencing User
    FK: 'status_ID' referencing Status

12 **review-comment**(review_ID, user_ID, content, time)
    Domain of 'time' is standard to the 24 hour clock (i.e (0-23):(00-60))
    Primary key is review_ID and user_ID together
    FK: 'user_ID' referencing User
    FK: 'review_ID' referencing Review

13 **votes**(review_ID, user_ID, vote)
    Domain of 'vote' is 1 or -1 (upvote or downvote)
    Primary key must be a combination of all attributes since review-comment has the primary key of review_ID + user_ID
    FK: 'user_ID' referencing User
    FK: 'review_ID' referencing Review

14 **has-reviews**(venue_ID, review_ID)
Primary key is venue_ID + review_ID
FK: ‘venue_ID’ referencing Venue
FK: ‘review_ID’ referencing Review

15 **rates**(user_ID, venue_ID, rating)
Domain of ‘rating’ is from 0 to 5 (stars)
Primary key: user_ID + venue_ID + rating
FK: ‘user_ID’ referencing User
FK: ‘venue_ID’ referencing Venue

16 **write-review**(writers_ID, review_ID)
Primary key: writers_ID + review_ID
FK: ‘writers_ID’ referencing User (user_ID)
FK: ‘review_ID’ referencing Review

17 **user-creates-event**(event_ID, user_ID)
Primary key: event_ID + user_ID
FK: ‘event_ID’ referencing Event
FK: ‘user_ID’ referencing User

18 **admin-creates-event**(venue_ID, venue_announcement_ID)
Primary key: venue_ID + venue_announcement_ID
FK: ‘venue_ID’ referencing Venue
FK: ‘venue_announcement_ID’ referencing Venue_Announcement

19 **event-at**(event_ID, venue_ID)
Primary Key: event_ID + venue_ID
FK: ‘event_ID’ referencing Event
FK: ‘venue_ID’ referencing Venue

20 **venue-admin-owns**(venue_admin_ID, venue_ID)
Primary Key: venue_admin_ID + venue_ID
FK: ‘venue_ID’ referencing Venue
FK: ‘venue_admin_ID’ referencing Venue_Admin

21 **admin-create-announcement**(venue_admin_ID, venue_announcemnet_ID)
Primary Key: venue_admin_ID + venue_announcement_ID
FK: ‘venue_admin_ID’ referencing Venue_Admin
FK: ‘venue_announcement_ID’ referencing Venue_Announcement

22. **is_going_to_event**(user_ID, event_ID)
Primary Key: user_ID + even_ID

FK: ‘user_ID’ referencing User
FK: ‘event_ID’ referencing Event

4. Implementation Details

For this project we have made use of python and used a web application framework called "Flask". We also made use of the module called as "PyMySQL" that aids in connection to our database system. The Database we used is the same one we had used for our programming project on the dijkstra machine. Using the right ports, passwords and hosting addresses a connection was made to that server to allow for different machines to be able to run the code and utilize the server. This was a quick fix to check the applications behavior with multiple users without hosting our code online.

So once the database is created, and send our queries and then further we were able to create our corresponding tables and use python programming to check logic such as already existing usernames, etc. We created sign-up interfaces using html, and are running it on a local host via python programming and the help of our framework, flask. Then we implemented it sign-up methods, the MySQL requests and the stored procedures.

We make use of Javascript to make the static HTML code more dynamic. We used it to check the user inputs and confirm if they were correct and according to our liking. We also used JavaScript to add and remove some features for admins and normal users that had to be directly handled and not done through the database.

Using Python and Javascipt as the two scripting languages were we able to make this application. Python handled running the web server and communicating with the DataBase while Javascipt allowed us to make the application more dynamic and "smart".

We were entirely new to the Flask framework and it took a while to research about it and get familiarized with and therefore we encountered a lot of problems when coding our project. It was difficult to integrate python and html to create the user interfaces. Some problems we faced were with uploading images, passing the URLs to the function, correct navigation between pages. Other problems we faced were actual integration and connection to the database through flask. We did not face any major problems where the solutions should be stated, however, we were able to solve all these problems with extensive researching.

5. Advanced Database Components

We have made use of triggers for our advanced database component for multiple functionalities to aid in our system. We have added triggers for when a similar email is added twice so that we do not copy data and are informed. We also add triggers for statuses uploaded of friends so we can notify users to new statuses. They are also informed of likes and comments by using triggers in order to alert is of such events.

a) Reports

To display the most liked Venues to a non user to browse. This allows for them to observe the most famous venues.

```
Select venue.name

from ( select venue.name , count (votes) as total_votes
        from rates, venue
        where rates.venue_ID = venue.venue_Id
        group by venue_id
      )

Order by total_votes DESC
```

Gets the average review of a venue by calculating all reviews of the users for a venue.

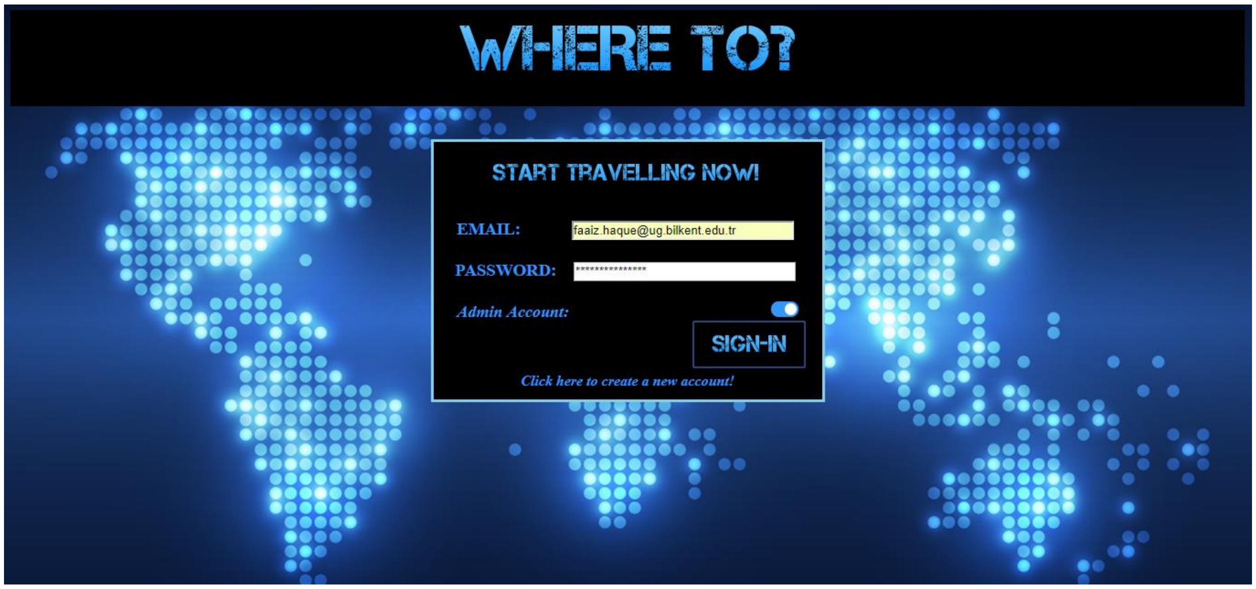
```
Select avg(rating)

from rates

where venue_ID = current_Venue
```

b) Triggers

- Upon creation of an event by a user, an invite is sent to all of the user’s friends.
- Before voting on a review or rating on a view , it is checked if user has performed such an action before or not.
- Before writing a review, we check if the user has visited that place before or not.

6. User Manual

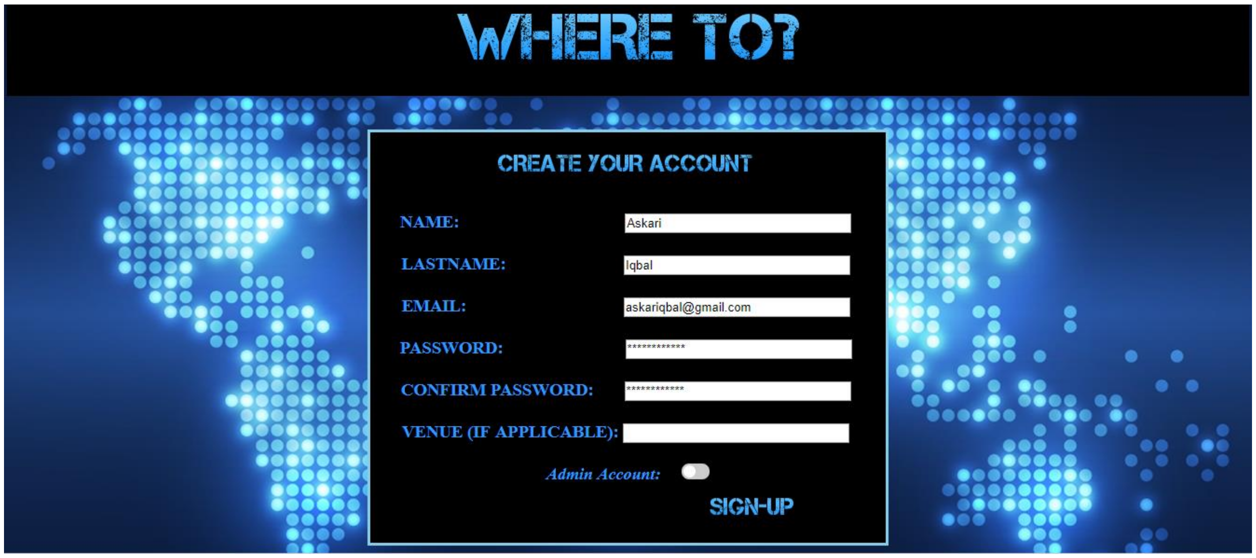
6.1 Sign-In



EMAIL: faaiz.haque@ug.bilkent.edu.tr
PASSWORD: ***************
Admin Account: on

This is the log-in page of our application where existing users can sign-in to their accounts using their previously created e-mail and password that have been saved in our database.
If a user has not created an account in the past they may sign-up (sign-up step given below).

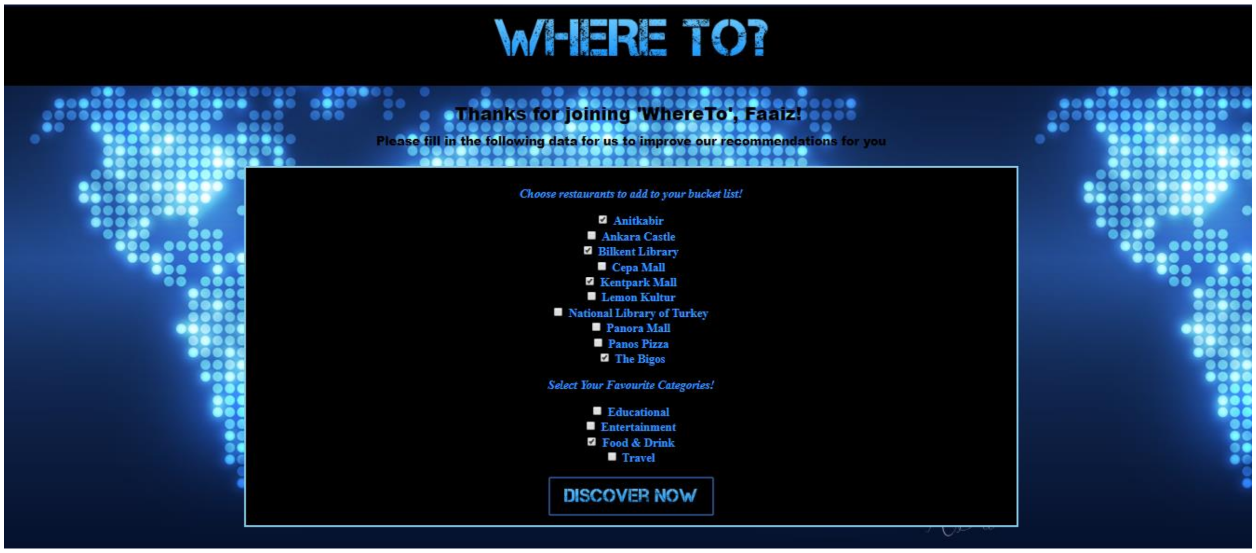
6.2 Sign-Up



NAME: Askari
LASTNAME: Iqbal
EMAIL: askariqbal@gmail.com
PASSWORD: ************
CONFIRM PASSWORD: ************
VENUE (IF APPLICABLE):
Admin Account: off

A user can register a new account by providing a few details as illustrated above. The venue data field should be left empty, as its only applicable for admin accounts. Once a user successfully provides his/her details, they will be prompted to the next part of signing up, which is explained in detail below. At this point a user's account is already created, and they can continue the rest of the sign-up process later. From here you may log-out and sign-in later using the e-mail and password you provided when signing up.

6.3 Personalize Profile



Thanks for joining 'WhereTo', Faaiz!

Please fill in the following data for us to improve our recommendations for you

Choose restaurants to add to your bucket list!

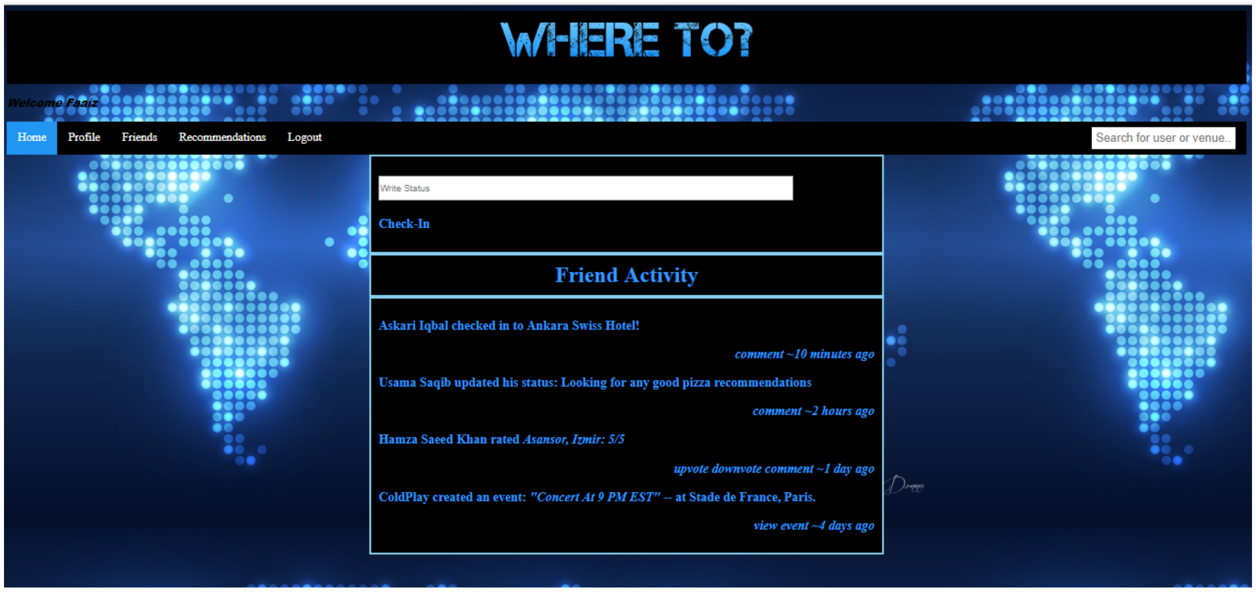
- [✓] Anitkabir
- [ ] Ankara Castle
- [✓] Bilkent Library
- [ ] Cepa Mall
- [✓] Kentpark Mall
- [ ] Lemon Kultur
- [ ] National Library of Turkey
- [ ] Panora Mall
- [ ] Panos Pizza
- [✓] The Bigos

Select Your Favourite Categories!

- [ ] Educational
- [ ] Entertainment
- [✓] Food & Drink
- [ ] Travel

After registering an account, another sign-up step will be prompted to the user where they may add a list of venues they wish to visit to their bucket list. Additionally, they can mention their favorite categories in order to be recommended venues by the system that are relevant to their preferences.

6.4 Home Page



Friend Activity

Askari Iqbal checked in to Ankara Swiss Hotel!

comment ~10 minutes ago

Usama Saqib updated his status: Looking for any good pizza recommendations

comment ~2 hours ago

Hamza Saeed Khan rated *Asansor, Izmir: 5/5*

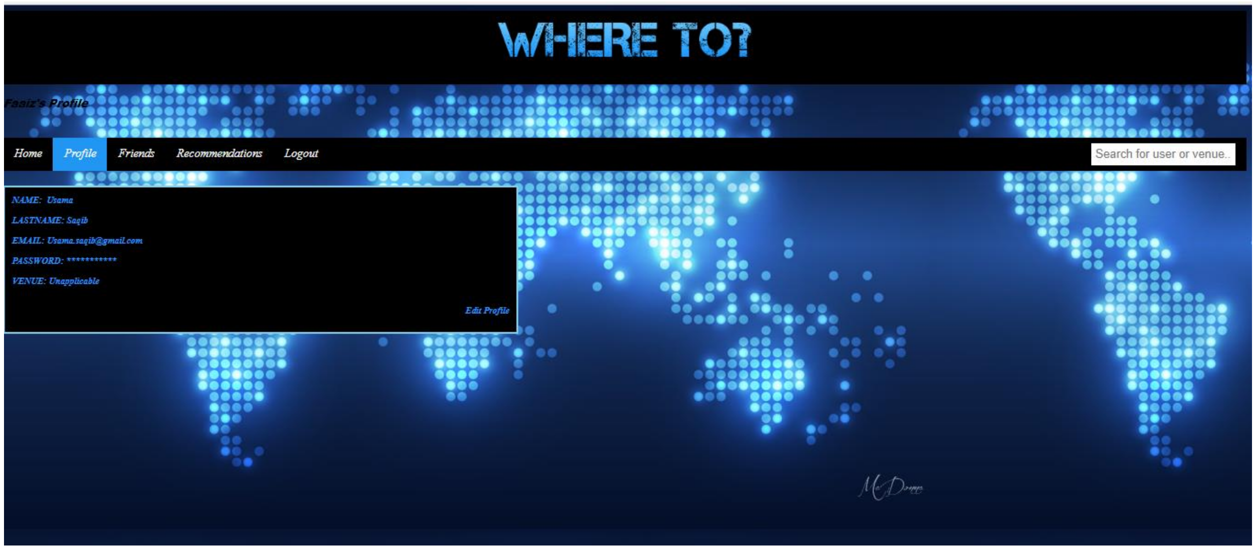
upvote downvote comment ~1 day ago

ColdPlay created an event: *"Concert At 9 PM EST"* -- at Stade de France, Paris.

view event ~4 days ago

The users are prompted to the home-page after successfully creating an account or logging in to an already existing account. As above we can see the user Faaiz has logged in to his account. As opposed to our original home page mock-up given in the design report, we decided to keep it more simple in our actual implementation. As you can see, all we can see in the home page is our news feed, or in this case friend activity. A user can interact with his friend's activity by commenting, or upvoting and downvoting ratings. This is all a user can do on a homepage, however the user can navigate to other menus from the homepage and any other page while logged in through the menu bar given at the top. A user can see his personalized profile, his list of friends, his recommendation list, or log-out of his/her account. Additionally users can search for a user or venue in the search box given in the top right corner.

6.5 Profile



NAME: Usama

LASTNAME: Saqib

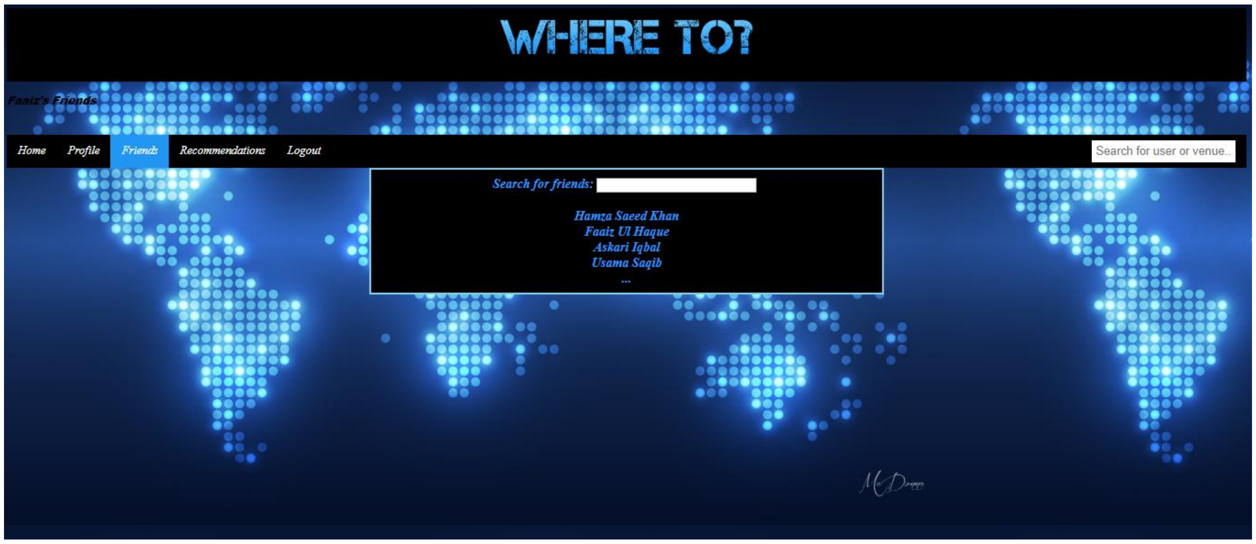
EMAIL: Usama.saqib@gmail.com

PASSWORD: ***********

VENUE: Unapplicable

This page is shown after a user selects the profile option from the menu bar at the top of the page. The profile page is simple and simply shows a user's basic details. A user can edit his/her profile by clicking the button in the bottom right corner labelled 'edit profile'. Data such as name, lastname, password, email may be changed.

6.6 Friends



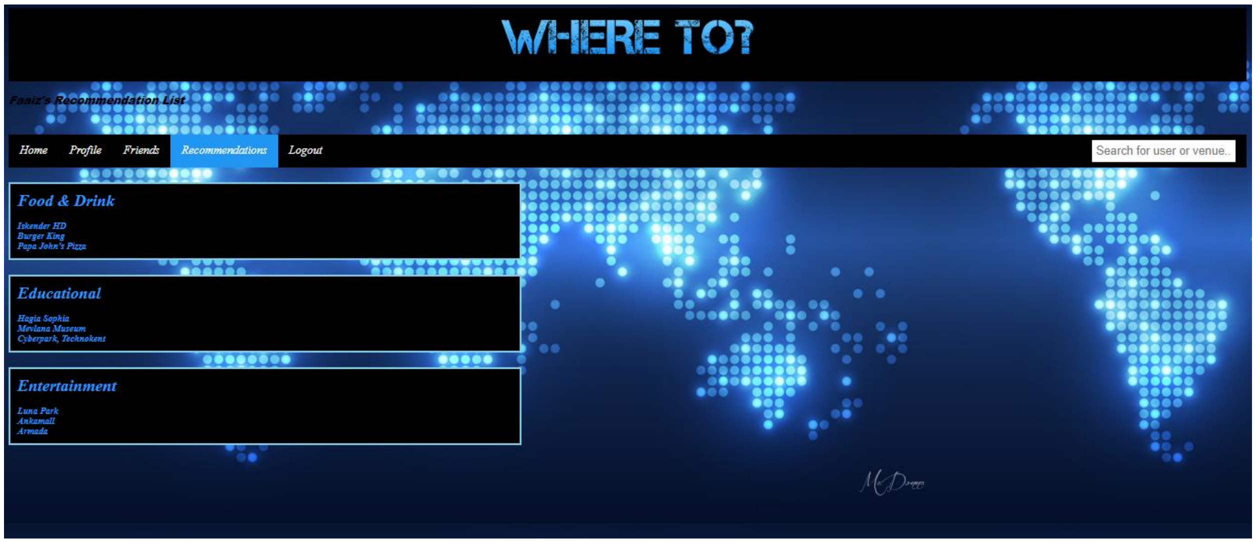
Faaiz's Friends

Search for friends:

Hamza Saeed Khan
Faaiz Ul Haque
Askari Iqbal
Usama Saqib
...

This page is shown after a user selects the menu labelled friends in the menu bar at the top. This page will simply show a list of the user's friends, from where a user can visit his friend's profile by clicking on their corresponding name. A user can also search for a specific friend simply for convenience purposes.

6.7 Recommendations



Faaiz's Recommendation List

Food & Drink

Iskender HD
Burger King
Papa John's Pizza

Educational

Hagia Sophia
Mevlana Museum
Cyberpark, Technokent

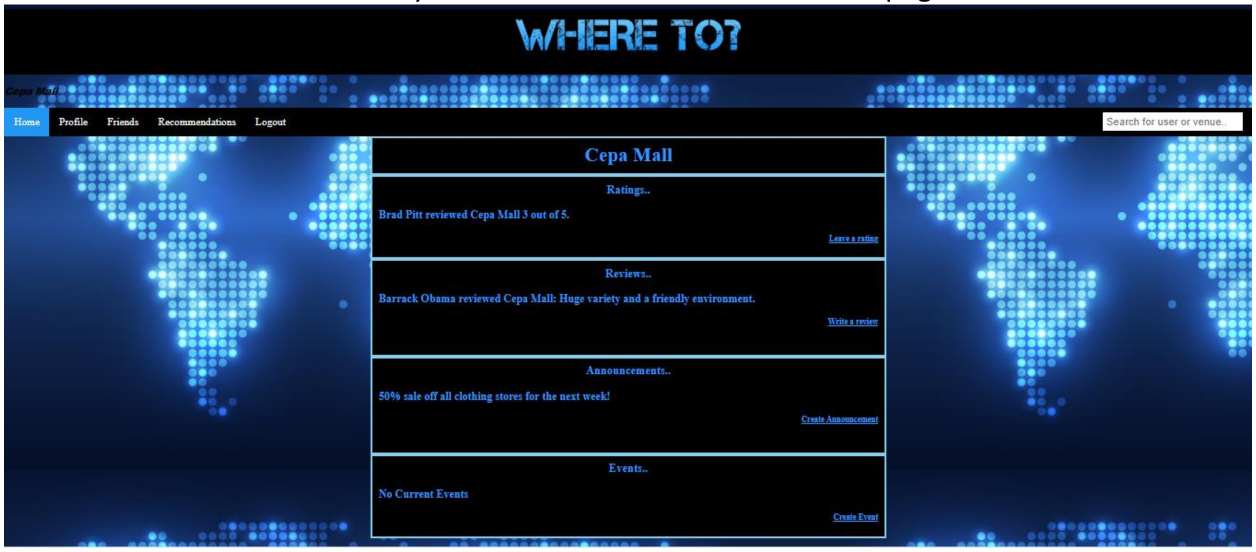
Entertainment

Luna Park
Ankamall
Armada

This page is shown after a user selects the recommendations menu in the menu bar. The recommendations are shown based off the places that are currently in your bucket list when personalizing your profile, places that fall under your favourite category, and places that are similar to those you have visited in the past. Each recommended place is shown under its respective category.

6.8 Venue Page

This page is prompted after a user uses the search for venue option and selects a venue. Users can also select a venue page from their friend’s reviews or check-ins. Users can leave reviews on check-ins which receive comments and upvotes and downvotes. Users can also rate venues out of 5 stars. Users can also create events at the venue by specifying a time and a description of the event. Users can check-in to the venue as well. The admins can perform the same tasks as the users and in addition they can create announcements on the page.



Cepa Mall

Ratings..

Brad Pitt reviewed Cepa Mall 3 out of 5.

Leave a rating

Reviews..

Barrack Obama reviewed Cepa Mall: Huge variety and a friendly environment.

Write a review

Announcements..

50% sale off all clothing stores for the next week!

Create Announcement

Events..

No Current Events

Create Event