

# Project Report

**Principles of Database Systems (CS6083)**

**Submitted By:**

**Nazia Akhtar      na1886  
Sunidhi Shukla    ss10448**



**NYU**

**TANDON SCHOOL  
OF ENGINEERING**

## Table of Contents

<b>1. System Requirements:</b> .....	<b>5</b>
Use Cases: .....	5
<b>2. Database Design:</b> .....	<b>5</b>
The ER Diagram:.....	6
Design Details and Assumptions:.....	7
Relational Model: .....	8
Foreign Key Constraints:.....	9
The Database Tables:.....	10
Users:.....	10
Projects:.....	11
Followers: .....	13
CreditCard:.....	14
FundsPledged:.....	15
FundsRaised:.....	17
Rating:.....	17
Discussion:.....	18
SQL Queries:.....	19
<b>3. System Implementation:</b> .....	<b>29</b>
User Manual: .....	29
Sign up:.....	30
Login Page:.....	31
Login with empty username and/or password:.....	Error! Bookmark not defined.
Login with incorrect username and/or password:.....	32
Action Page: .....	33
Project Creation Page:.....	34
Home page:.....	35

<b>Logout Page:</b>	36
<b>Project Detail Page:</b>	36
<b>Comment and Likes:</b>	39
<b>Recommendations Page:</b>	41
<b>All Users Page:</b>	42
<b>Timeline Page:</b>	43
<b>My Account Page:</b>	Error! Bookmark not defined.
<b>3.2 Additional Features:</b>	44
<b>4. Conclusion:</b>	47
<b>5. References:</b>	47

## Abstract

This document describes the purpose, design and functioning of the crowdfunding website. The crowdfunding project comprised of phases of understanding the system requirements, designing the database, implementing the backend and frontend of the website and testing the system. The major objectives of the system are to allow users to post creative projects which need funding to accomplish. Additionally, social media website features are also provided. Users could also look up other peoples' projects and pledge funds for the projects which they found interesting. This document explains all these phases extensively along with screenshots from the actual system and code snippets.

## **1. System Requirements:**

Understanding the crowd funding website system requirements was the initial phase of the project. In this phase, we came up with all the actions that the user should be able to perform on our system and how the system should respond in different scenarios. The major use cases are given below:

### **Use Cases:**

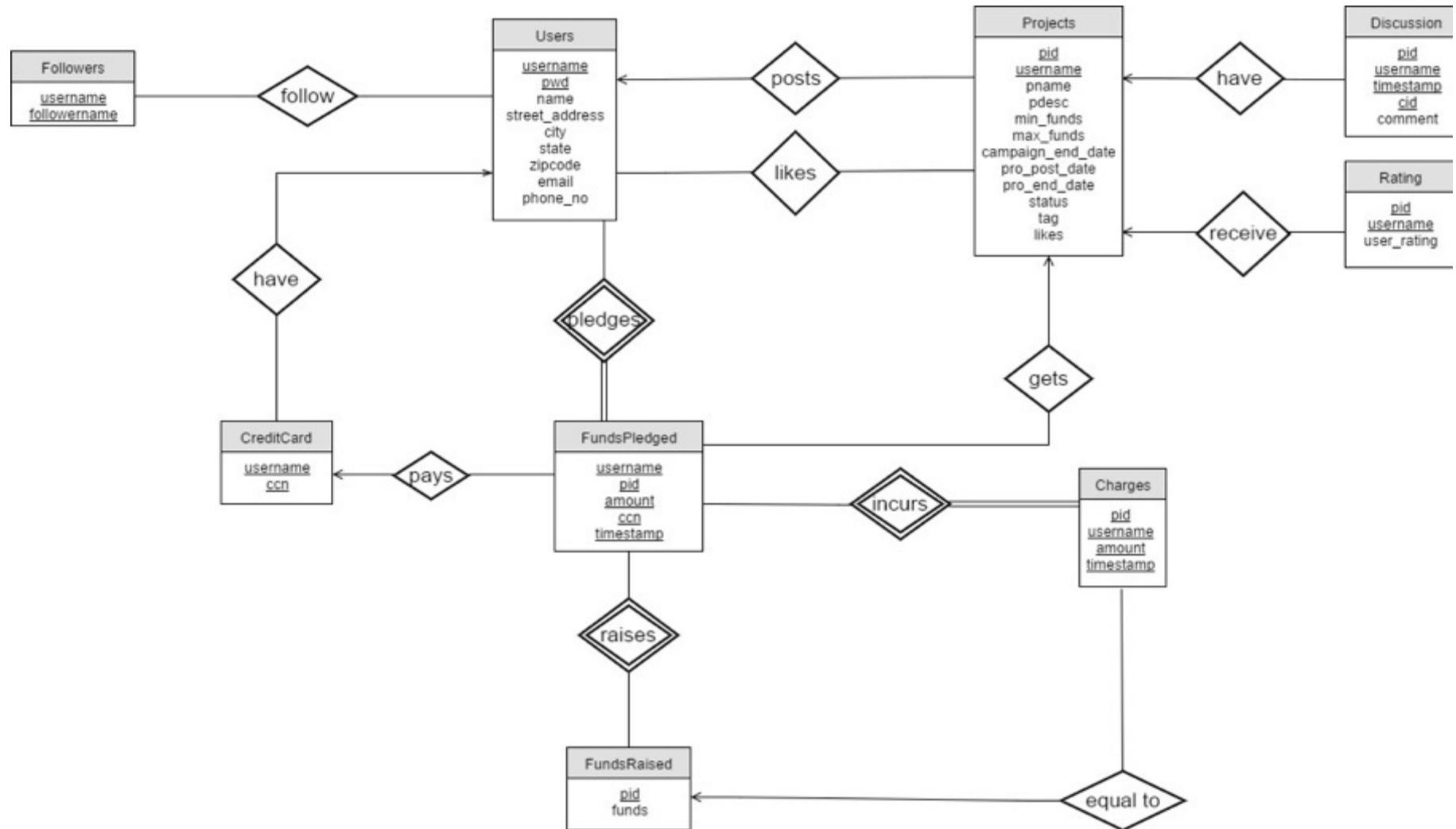
- New users can sign up
- Existing users must login to the system
- Users can post new projects
- Users can pledge funds for other peoples' projects
- Users can follow other users
- Users can see their timeline which will display the project of the users they are following
- User can like a project
- Based on user actions, recommendations are given to the user
- User can view his/her own profile
- User can click on a project and view all the details regarding that project
- User can post comments on a project and view other people comments

## **2. Database Design:**

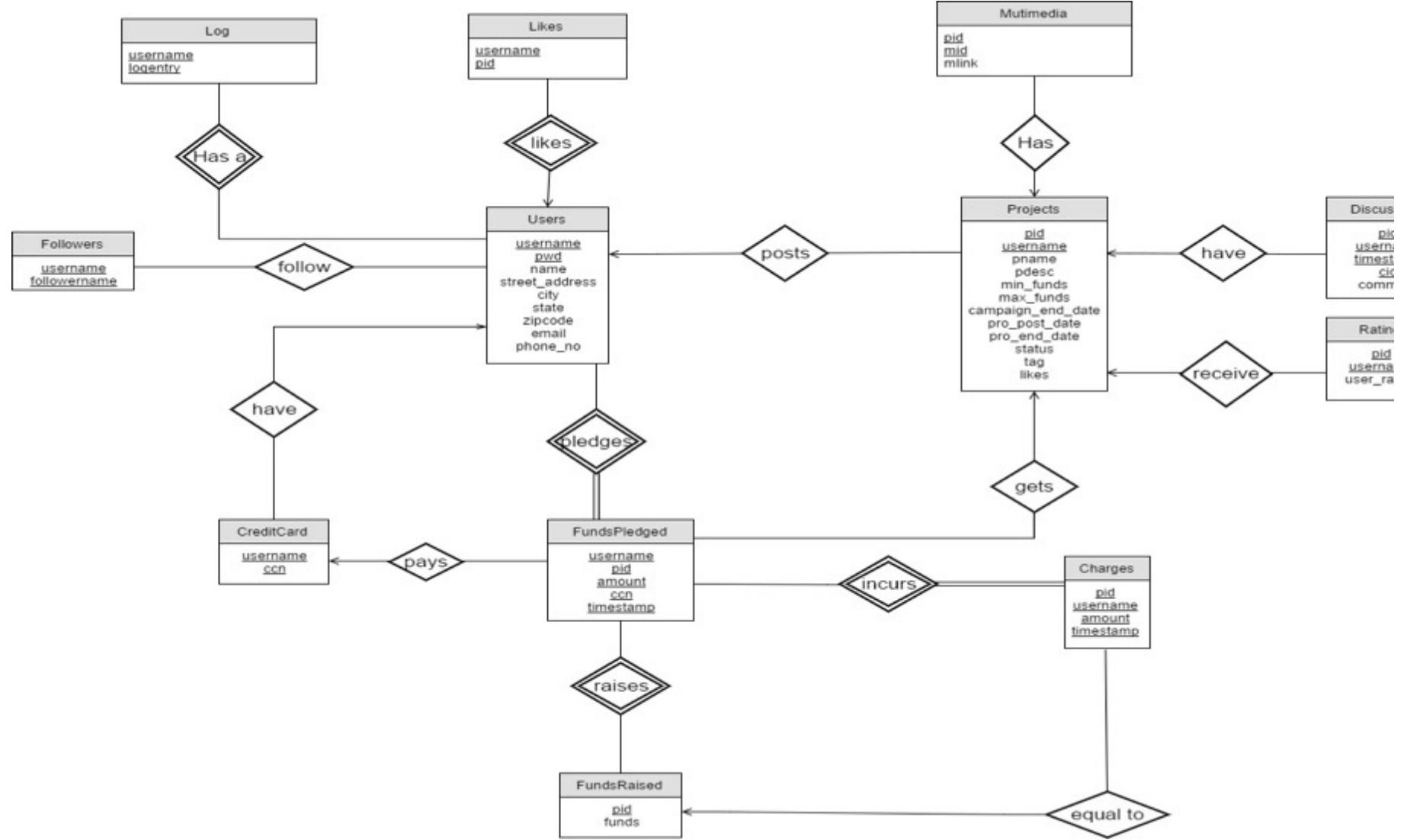
After jotting down all the use cases, the next step was to design the database system. The database design was modified as we incremented through the phases of our software development cycle and as our understanding of the system improved

## The ER Diagram:

The following ER Diagram was designed before implementation phase:



As we understood the system requirements better, we made some modifications to the existing ER Diagram which are reflected below:



The following changes were made to the design:

- Instead of a regular integer for ‘Pid’, we used UUID
- New tables were added such Log, Likes and Multimedia table was added
- All the date-time columns were changed to Time-Stamp

## **Design Details and Assumptions:**

In this database, users are uniquely identified by a username and the required data regarding the user such as password, full name, phone number, email and complete address are stored in the user table. Any user can post a project to gather funds and a user can also pledge funds for projects posted by other users. A project is uniquely identified by a project id and username to whom the project belongs. Additionally, project name, project description, project postdate, project end date, campaign end date, maximum and minimum funds needed to execute the project, status of the project and the number of likes the project received are stored in the project table. Each project has a tag as to specify what category of projects it belongs to. The status of the project could one of the following: “Completed”, “Failed”, “Ongoing”, “Accepting Funds”. A user can post multiple projects but a project must belong to only one user. Each user must provide one or more credit card number which is stored in the credit card table. Hence, a user can have one or more credit card numbers. A user can follow other users and see their likes, comments and posts which is stored in the followers table. Therefore, there is a many to many relationship between users and followers. When a project is posted, a user can help the project by pledging funds which is stored in the funds pledged table with the username as foreign key and the time at which funds were pledged. Note that Funds pledged table is a weak entity here since it has no primary key of its own. A user can pledge funds for multiple projects hence it is also a many to many relationship. The total number of funds which are raised by a project are stored in the funds raised table referencing the project id. When the end time for a campaign is reached or the maximum number of funds required for a project is raised, the project status is changed to “failed” or “completed” and if completed, the sponsors are charged for the amount they promised. These transactions are stored in Charges table. Users can write comments on various projects which is stored in the Discussion table. The system also gives recommendations of projects to the users based on their activity on the website. This is done through creating activity log in the log table.

## **Relational Model:**

We have the following tables in relational model:

*Users(username, password, name, street\_address, city, state, zipcode, email, phone\_no)*

*Projects(pid, username, pname, pdescription, min\_funds, max\_funds, camp\_end\_date, proj\_post\_date, proj\_end\_date, status, likes, tag)*

*Discussion(cid, pid, username, timestamp, comment)*

*Rating(pid, username, user\_rating)*

*Followers(username, followername)*

*Creditcard(username, ccn)*

*FundsPledged(username, pid, amount, ccn, timestamp)*

*FundsRaised(pid, funds)*

*Charges(pid, username, amount, timestamp)*

*Log(username, logentry)*

*Likes(pid, username)*

## Foreign Key Constraints:

*Project(username) references Users(username)*

*CreditCard(username) references Users(username)*

*Follower(username) references Users(username)*

*Discussion(username,pid) references Projects(username,pid)*

*Rating(username,pid) references Projects(username,pid)*

*FundsPledged(username,ccn) references CreditCard(username,ccn)*

*FundsPledged(pid) references Projects(pid)*

*Charges(username, pid) references FundsPledged(username, pid)*

*FundsRaised(pid) references FundsPledged(pid)*

## The Database Tables:

## Users:

## Projects:

The screenshot shows a database grid interface with various tools and filters at the top. The main area displays a table of project data with the following columns: pid, username, pname, pdescription, minfunds, maxfunds, camp\_end\_date, and pro\_. The data includes multiple entries for different users like Tom11, Lucy, Jessica, and Mike23, each with unique project names and descriptions.

	pid	username	pname	pdescription	minfunds	maxfunds	camp_end_date	pro_
▶	024a2c91-3481-11e7-8f87-a402b924d4d3	Tom11	Save water.	We need to save water.	100.00	200.00	2017-10-10 21...	2017-
	1011ee56-347f-11e7-8f87-a402b924d4d3	Tom11	Save water.	We need to save water.	100.00	200.00	2017-06-06 22...	2017-
	252e2a47-31dc-11e7-8f87-a402b924d4d3	tom11	save trees	We need to save trees	3000.00	5000.00	2017-06-01 13...	2017-
	346f5a1a-349a-11e7-8f87-a402b924d4d3	Tom11	Save water.	We need to save water.	100.00	20000.00	2017-06-06 20...	2017-
	3625f6af-31d7-11e7-8f87-a402b924d4d3	tom11	Save electricity.	We need to save electricity.	4000.00	7000.00	2017-05-07 16...	2017-
	37eac406-3480-11e7-8f87-a402b924d4d3	Tom11	Save water.	We need to save water.	100.00	200.00	2017-10-10 21...	2017-
	47879096-31d6-11e7-8f87-a402b924d4d3	tom11	Save water.	We need to save water.	100.00	200.00	2017-05-01 13...	2017-
	4cd0433c-3166-11e7-8f87-a402b924d4d3	Lucy	SoundBot II: Charging Your Speakers Can't Get ...	Revolutionary & Powerful High-Performance Pre...	200000.00	300000.00	2016-12-01 00...	2016-
	4de6ff96-3481-11e7-8f87-a402b924d4d3	Tom11	Save water.	We need to save water.	100.00	20000.00	2017-06-06 20...	2017-
	5215bc13-32bf-11e7-8f87-a402b924d4d3	Tom11	Save water.	We need to save water.	100.00	200.00	2017-05-04 17...	2017-
	52c6dae4-3166-11e7-8f87-a402b924d4d3	Jessica	RideGlide: The University Carpool Community	A forum and database of driver reviews/info wit...	200000.00	300000.00	2016-05-01 00...	2015-
	52c83bde-3166-11e7-8f87-a402b924d4d3	Jessica	The Worlds Family Tree	Imagine seeing a family tree of the entire world...	5000.00	8000.00	2015-05-01 00...	2014-
	52c9f0e4-3166-11e7-8f87-a402b924d4d3	Tom11	Jazz 2020	The most amazing jazz band from New Orleans ...	23000.00	30000.00	2016-05-01 00...	2015-
	52cb0544-3166-11e7-8f87-a402b924d4d3	Mike23	Painting Fest	Support the highly talented yet unrecognized ar...	12000.00	18000.00	2016-07-01 00...	2015-

Limit to 1000 rows | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	pdescription	minfunds	maxfunds	camp_end_date	pro_post_date	pro_end_date	status	tag	likes
▶	We need to save water.	100.00	200.00	2017-10-10 21...	2017-05-09 02:29:56	2017-10-20 22:10:00	accepting funds	environment	11
	We need to save water.	100.00	200.00	2017-06-06 22...	2017-05-09 02:16:00	2017-10-10 22:10:00	accepting funds	environment	39
	We need to save trees	3000.00	5000.00	2017-06-01 13...	2017-05-05 17:44:52	2017-08-01 13:00:00	accepting funds	environment	0
	We need to save water.	100.00	20000.00	2017-06-06 20...	2017-05-09 05:30:18	2017-10-10 21:00:00	accepting funds	environment	0
	We need to save electricity.	4000.00	7000.00	2017-05-07 16...	2017-05-05 17:09:33	2017-06-07 17:00:00	accepting funds	environment	0
	We need to save water.	100.00	200.00	2017-10-10 21...	2017-05-09 02:24:16	2017-10-20 21:00:00	accepting funds	environment	0
	We need to save water.	100.00	200.00	2017-05-01 13...	2017-05-05 17:02:52	2017-10-10 13:00:00	accepting funds	environment	0
our Speakers Can't Get ...	Revolutionary & Powerful High-Performance Pre...	200000.00	300000.00	2016-12-01 00...	2016-03-15 00:00:00	2017-04-15 00:00:00	ongoing	technology	22
	We need to save water.	100.00	20000.00	2017-06-06 20...	2017-05-09 02:32:03	2017-10-10 21:00:00	accepting funds	environment	0
	We need to save water.	100.00	200.00	2017-05-04 17...	2017-05-06 20:51:00	2017-05-31 18:00:00	accepting funds	environment	0
Carpool Community	A forum and database of driver reviews/info wit...	200000.00	300000.00	2016-05-01 00...	2015-10-15 00:00:00	2016-10-15 00:00:00	ongoing	career dev...	350
	Imagine seeing a family tree of the entire world...	5000.00	8000.00	2015-05-01 00...	2014-10-15 00:00:00	2015-10-15 00:00:00	failed	social	1248
	The most amazing jazz band from New Orleans ...	23000.00	30000.00	2016-05-01 00...	2015-10-15 00:00:00	2016-10-15 00:00:00	successful	jazz	324...
	Support the highly talented yet unrecognized ar...	12000.00	18000.00	2016-07-01 00...	2015-12-30 00:00:00	2016-12-15 00:00:00	successful	jazz	324...

projects 2 ×

Apply

Revert

Output

**Followers:**

username	followername
Erica1	BobInBrooklyn
Erica1	Lucy
Erica1	Mike23
Jessica	BobInBrooklyn
Lucy	BobInBrooklyn
Lucy	Mike23
Mike23	BobInBrooklyn
Tom11	BobInBrooklyn
Tom11	Lucy
NULl	NULl

## CreditCard:

The screenshot shows a MySQL Workbench interface with a table named 'use\_Final Project'. The table has two columns: 'username' and 'ccn'. The data is as follows:

	username	ccn
▶	alice23	4576982367239865
	Bravo23	2345671298126743
	dyana5	457698236
	Erica1	4207123546786
	Erica1	4345625896798
	Jerry89	1289767645679879
	Jessica	327864521298
	Lucy	8723451234677
	Mike23	2345678912345
	sohos	1234874826796535
	Tom11	2314563489763
*	HULL	HULL

## Mutlimedia:

81-11e7-8f87-a402b924d4d3	024ae9cc-3481-11e7-8f87-a402b924d4d3
7f-11e7-8f87-a402b924d4d3	10129e24-347f-11e7-8f87-a402b924d4d3
dc-11e7-8f87-a402b924d4d3	252ed1bb-31dc-11e7-8f87-a402b924d4d3
9a-11e7-8f87-a402b924d4d3	3470194a-349a-11e7-8f87-a402b924d4d3
f7-11e7-8f87-a402b924d4d3	uuid()
80-11e7-8f87-a402b924d4d3	37eb731a-3480-11e7-8f87-a402b924d4d3
91-11e7-8f87-a402b924d4d3	4de7ba74-3481-11e7-8f87-a402b924d4d3
bf-11e7-8f87-a402b924d4d3	52166796-32bf-11e7-8f87-a402b924d4d3
f1-11e7-8f87-a402b924d4d3	7c5fd90d-34f1-11e7-8f87-a402b924d4d3
7d-11e7-8f87-a402b924d4d3	82c83d35-347d-11e7-8f87-a402b924d4d3
81-11e7-8f87-a402b924d4d3	8b6e8112-3481-11e7-8f87-a402b924d4d3
30-11e7-8f87-a402b924d4d3	9d83afaa-3480-11e7-8f87-a402b924d4d3
7e-11e7-8f87-a402b924d4d3	a4079898-347e-11e7-8f87-a402b924d4d3
be-11e7-8f87-a402b924d4d3	a5404845-32be-11e7-8f87-a402b924d4d3
7f-11e7-8f87-a402b924d4d3	a90f6ae1-347f-11e7-8f87-a402b924d4d3
ef-11e7-8f87-a402b924d4d3	be6544a6-34ef-11e7-8f87-a402b924d4d3
7-11e7-8f87-a402b924d4d3	cecfeb9d-31d7-11e7-8f87-a402b924d4d3
f0-11e7-8f87-a402b924d4d3	d6138b01-34f0-11e7-8f87-a402b924d4d3

**FundsPledged:**

username	pid	amount	ccn
Lucy	101	50000.00	8723451234677
Mike23	102	50000.00	2345678912345
Lucy	103	1000.00	8723451234677
Lucy	104	15000.00	8723451234677
Mike23	104	8000.00	2345678912345
Erica1	105	10000.00	4345625896798
tom11	105	5000.00	2314563489763
Erica1	106	100000.00	4345625896798
Tom11	106	50000.00	2314563489763
Lucy	109	2000.00	8723451234677
Mike23	109	5000.00	2345678912345
Lucy	110	100.00	8723451234677
NULL	NULL	NULL	NULL

FundsRaised:

pid	funds
104	23000.00
105	15000.00
106	150000.00
109	7000.00
NULL	NULL

Rating:

pid	username	user_rating
104	Lucy	4
105	Erica1	5
105	Tom11	4
106	Erica1	4
106	Tom11	5
109	Lucy	5
NULL	NULL	NULL

## Discussion:

The screenshot shows a MySQL Workbench interface with a table named 'discussion4'. The table has five columns: 'cid', 'username', 'pid', 'tcomment', and 'comment'. The data consists of 11 rows, with the first row being selected. The selected row's 'cid' value is highlighted in blue.

	cid	username	pid	tcomment	comment
▶	2882cef1-3678-11e7-8e95-a402b924d4d3	Tom11	7c5f0d3d-34f1-11e7-8f87-a402b924d4d3	2017-05-11 14:32:13	I like it!
	2b6d267c-3678-11e7-8e95-a402b924d4d3	Tom11	7c5f0d3d-34f1-11e7-8f87-a402b924d4d3	2017-05-11 14:32:18	I like it!
	306cbc9e-366a-11e7-8e95-a402b924d4d3	Tom11	7c5f0d3d-34f1-11e7-8f87-a402b924d4d3	2017-05-11 12:52:13	Very nice!
	8152a59e-3663-11e7-8e95-a402b924d4d3	Tom11	7c5f0d3d-34f1-11e7-8f87-a402b924d4d3	2017-05-11 12:04:23	Hii!
	8340f984-3668-11e7-8e95-a402b924d4d3	Tom11	7c5f0d3d-34f1-11e7-8f87-a402b924d4d3	2017-05-11 12:40:13	Very nice!
	837b594f-3663-11e7-8e95-a402b924d4d3	Tom11	7c5f0d3d-34f1-11e7-8f87-a402b924d4d3	2017-05-11 12:04:26	Hii!
	859d8953-3668-11e7-8e95-a402b924d4d3	Tom11	7c5f0d3d-34f1-11e7-8f87-a402b924d4d3	2017-05-11 12:40:17	Very nice!
	a3005df7-34df-11e7-8f87-a402b924d4d3	Mike23	1011ee56-347f-11e7-8f87-a402b924d4d3	2017-05-09 13:52:53	Hi!!!!
	f449f757-3662-11e7-8e95-a402b924d4d3	Tom11	a90eb83d-347f-11e7-8f87-a402b924d4d3	2017-05-11 12:00:26	HEY!!
*	ffa45c45-3662-11e7-8e95-a402b924d4d3	Tom11	a90eb83d-347f-11e7-8f87-a402b924d4d3	2017-05-11 12:00:45	HEY!!
	NULL	NULL	NULL	NULL	NULL

## SQL Queries:

- Create a record for a new user account, with a name, a login name, and a password.

```
INSERT INTO Users VALUES ('Oswald06','ossie45','Oswald Red', '78 Bleecker Street', 'NYC','NY',11201,'1678956337','oswald@test.com');
```

- List all projects that contain the keyword “jazz” and that are currently looking for funds, sorted in descending order by posting time.

```
select * from Projects where tag like '%jazz%' and status = 'accepting_funds'
ORDER BY pro_post_date DESC;
```

	pid	username	pname	pdescription	minfunds	maxfunds	camp_end_date	pro_post_date	pro_end_date	status	tag	likes
▶	108	BobInBrooklyn	New Orleans come to NYC	Help play the jazz band from new orleans play i...	200000.00	300000.00	2017-05-05 00:00:00	2017-03-15 00:00:00	2018-01-15 00:00:00	accepting_funds	jazz	120
*												

- List all users who have given money for projects containing the tag or category “jazz” in the past, sorted by the total amount they have successfully pledged (meaning, money that was actually charged).

```
select f.username
from FundsPledged f, Projects p natural join FundsRaised fr
where p.status = 'successful'
and p.tag like '%jazz%' and p.pro_end_date < now()
group by f.username
order by sum(f.amount) desc;
```

```
13
14
15 • select f.username
16 ☐ from FundsPledged f, Projects p natural join FundsRaised fr
17 where p.status = 'successful'
18 and p.tag like '%jazz%' and p.pro_end_date < now()
19 group by f.username
20 order by sum(f.amount) desc;
21
22
```

Result Grid | Filter Rows: [ ] Export: [ ] Wrap Cell Content: [ ]

	username
▶	Eric1
	Lucy
	Mike23
	tom11

- List all users who have completed at least 3 projects, and where each of their projects received an average rating of 4 stars or higher from its sponsors.

```
select distinct p.username
from Projects p
where p.status = 'successful' and p.pid in (select pid from Rating group by pid having AVG(user_rating)>=4) and
p.username in (select username from Projects group by username
having count(username)>=3);
```

```
34
35
36 • select distinct p.username
37   from Projects p
38 ✘ where p.status = 'successful' and p.pid in (select pid from Rating group by pid having AVG(user_rating)>=4) and
39   p.username in (select username from Projects group by username
40   having count(username)>=3);
41
42
43
44
45
46
47
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

username
Mike23

- List all comments by users that are followed by user "BobInBrooklyn".

```
select distinct d.username,d.comment
from Discussion d, Followers f
where f.followername = 'BobInBrooklyn';
```

```
31
32 • select distinct d.username,d.comment
33   from Discussion d, Followers f
34   where f.followername = 'BobInBrooklyn';
35
< Result Grid | Filter Rows: [ ] Export: [ ] Wrap Cell Content: [ ]
```

username	comment
Lucy	This is an interesting project
Lucy	How will you about implementing this?
Jessica	follow us guys
Mike23	These guys are awesome

- Insert a new project for a particular user, with a name, description, and other needed info.

INSERT INTO Projects VALUES (111,'Pooh06','Sizzling in the Kitchen Flynn Style','Help us raise the funds to film our pilot episode!', 75000,100000, '2017/04/20','2017/04/23','2017/08/23','accepting\_funds','TV show',50);

pid	username	pname	pdescription	minfunds	maxfunds	camp_end_date	pro_post_date	pro_end_da
101	Lucy	SoundBot II: Charging Your Speakers Can't Get ...	Revolutionary & Powerful High-Performance Pre...	200000.00	300000.00	2016-12-01 00:00:00	2016-03-15 00:00:00	2017-04-15 00:00:00
102	Jessica	RideGlide: The University Carpool Community	A forum and database of driver reviews/info wit...	200000.00	300000.00	2016-05-01 00:00:00	2015-10-15 00:00:00	2016-10-15 00:00:00
103	Jessica	The Worlds Family Tree	Imagine seeing a family tree of the entire world...	5000.00	8000.00	2015-05-01 00:00:00	2014-10-15 00:00:00	2015-10-15 00:00:00
104	Tom11	Jazz 2020	The most amazing jazz band from New Orleans ...	23000.00	30000.00	2016-05-01 00:00:00	2015-10-15 00:00:00	2016-10-15 00:00:00
105	Mike23	Painting Fest	Support the highly talented yet unrecognized ar...	12000.00	18000.00	2016-07-01 00:00:00	2015-12-30 00:00:00	2016-12-15 00:00:00
106	Mike23	Dance Crew Next Door	Dance Crew Next door is the ultimate hiphop da...	115000.00	250000.00	2016-07-01 00:00:00	2015-12-30 00:00:00	2016-12-15 00:00:00
107	Erica1	Painting Pros	An exhibition in the heart of New York City	200000.00	300000.00	2018-01-01 00:00:00	2017-03-15 00:00:00	2018-04-15 00:00:00
108	BobInBrooklyn	New Orleans come to NYC	Help play the jazz band from new orleans play i...	200000.00	300000.00	2017-05-05 00:00:00	2017-03-15 00:00:00	2018-01-15 00:00:00
109	Mike23	Balley Classes	Balley classes for autistic children	5000.00	80000.00	2016-07-01 00:00:00	2015-12-30 00:00:00	2016-12-15 00:00:00
110	Lucy	GIRLS STATE a new musical comedy TV project	In this new TV show "All Politics is Vocal" as high...	50000.00	75000.00	2017-01-01 00:00:00	2017-06-16 00:00:00	2017-04-15 00:00:00
111	Pooh06	Sizzling in the Kitchen Flynn Style	Help us raise the funds to film our pilot episode!	75000.00	100000.00	2017-04-20 00:00:00	2017-04-23 00:00:00	2017-08-23 00:00:00
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

- Insert a pledge to sponsor a project, for a particular user, project, and amount.

**INSERT INTO FundsPledged VALUES ('Lucy',111,1000,8723451234677);**

The screenshot shows a MySQL Workbench interface. At the top, there is a code editor window with the following content:

```

33
34 • INSERT INTO FundsPledged VALUES ('Lucy',111,1000,8723451234677);
35

```

Below the code editor is a toolbar with various icons for managing rows and columns. Underneath the toolbar is a result grid titled "Result Grid". The grid has four columns: "username", "pid", "amount", and "cn". The data in the grid is as follows:

	username	pid	amount	cn
▶	Lucy	101	50000.00	8723451234677
	Mike23	102	50000.00	2345678912345
	Lucy	103	1000.00	8723451234677
	Lucy	104	15000.00	8723451234677
	Mike23	104	8000.00	2345678912345
	Erica1	105	10000.00	4345625896798
	tom11	105	5000.00	2314563489763
	Erica1	106	100000.00	4345625896798
	Tom11	106	50000.00	2314563489763
	Lucy	109	2000.00	8723451234677
	Mike23	109	5000.00	2345678912345
	Lucy	110	100.00	8723451234677
	Lucy	111	1000.00	8723451234677
*		HULL	HULL	HULL

- Write queries for the end of a funding campaign. E.g., you could use triggers to detect when a campaign is fully funded or time is up; if successfully funded, generate charges to sponsors' credit cards.

## TRIGGER:

**DELIMITER //**

**create**

**Trigger end\_of\_campaign**

**After Insert On FundsPledged**

**For Each Row**

**BEGIN**

**DECLARE**

**v\_count , total INT;**

```
SELECT sum(amount)
INTO v_count
FROM FundsPledged
where pid = NEW.pid;
select maxfunds into total from Projects where pid = NEW.pid;
```

**IF v\_count > total**

**THEN**

**INSERT INTO Charges (select pid,amount,username from FundsPledged where pid = NEW.pid);**

**UPDATE Projects SET status ="Completed" where pid=NEW.pid;**

**END IF;**

**END //**

**DELIMITER ;**

**SCHEDULER 1:**

**DELIMITER |**

**CREATE EVENT time\_check**

**ON SCHEDULE EVERY 1 DAY**

**DO**

**BEGIN**

**UPDATE Projects SET status = "Failed" where camp\_end\_date >= NOW() and status ="ongoing" and status!="successful";**

**END |**

**DELIMITER ;**

**SCHEDULER 2:**

**DELIMITER |**

```
CREATE EVENT time_check
ON SCHEDULE EVERY 1 day
DO
BEGIN
UPDATE Projects SET status = "ongoing" where fundsraised.funds>projects.minfunds and now()>projects.camp_end_date;
END |
```

**DELIMITER;**

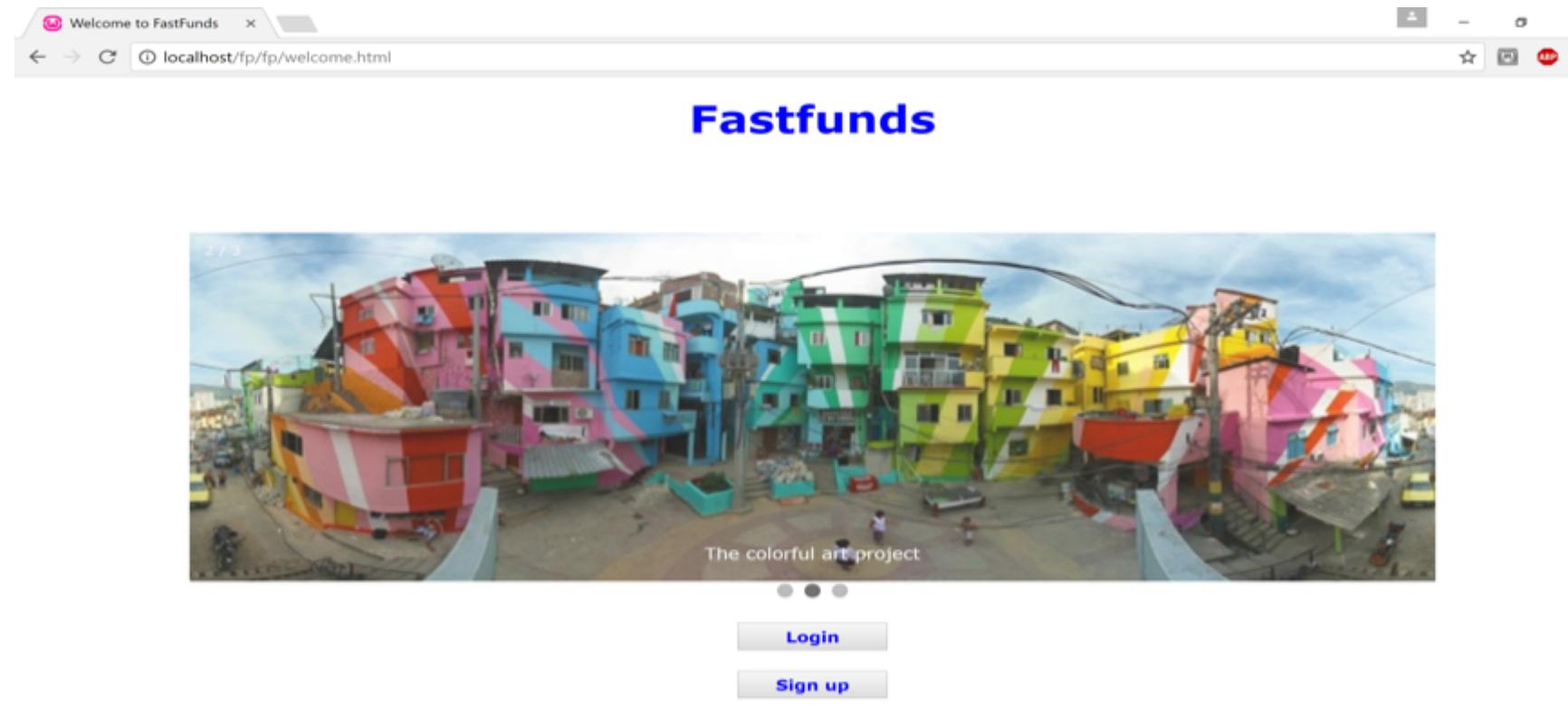
### 3. System Implementation:

Now that the database design and the required SQL queries, triggers and schedulers were written, the next step was to implement the actual crowdfunding website. We used PHP and MySQL for the backend and HTML and CSS for the frontend. The following user manual explains in depth on how to use our system.

#### User Manual:

The way this web app works is, first we need to make a user account. For that we must sign up.

#### The welcome page:



## Sign up:

This stores the user information and creates a profile for him/her. In sign up page, potential user must fill all the necessary fields which includes email, username, password and full name as required fields and optionally, user can enter street address, city, zip code, state, phone number and credit card number. Once the user signs up for the new account, it directs to the homepage.

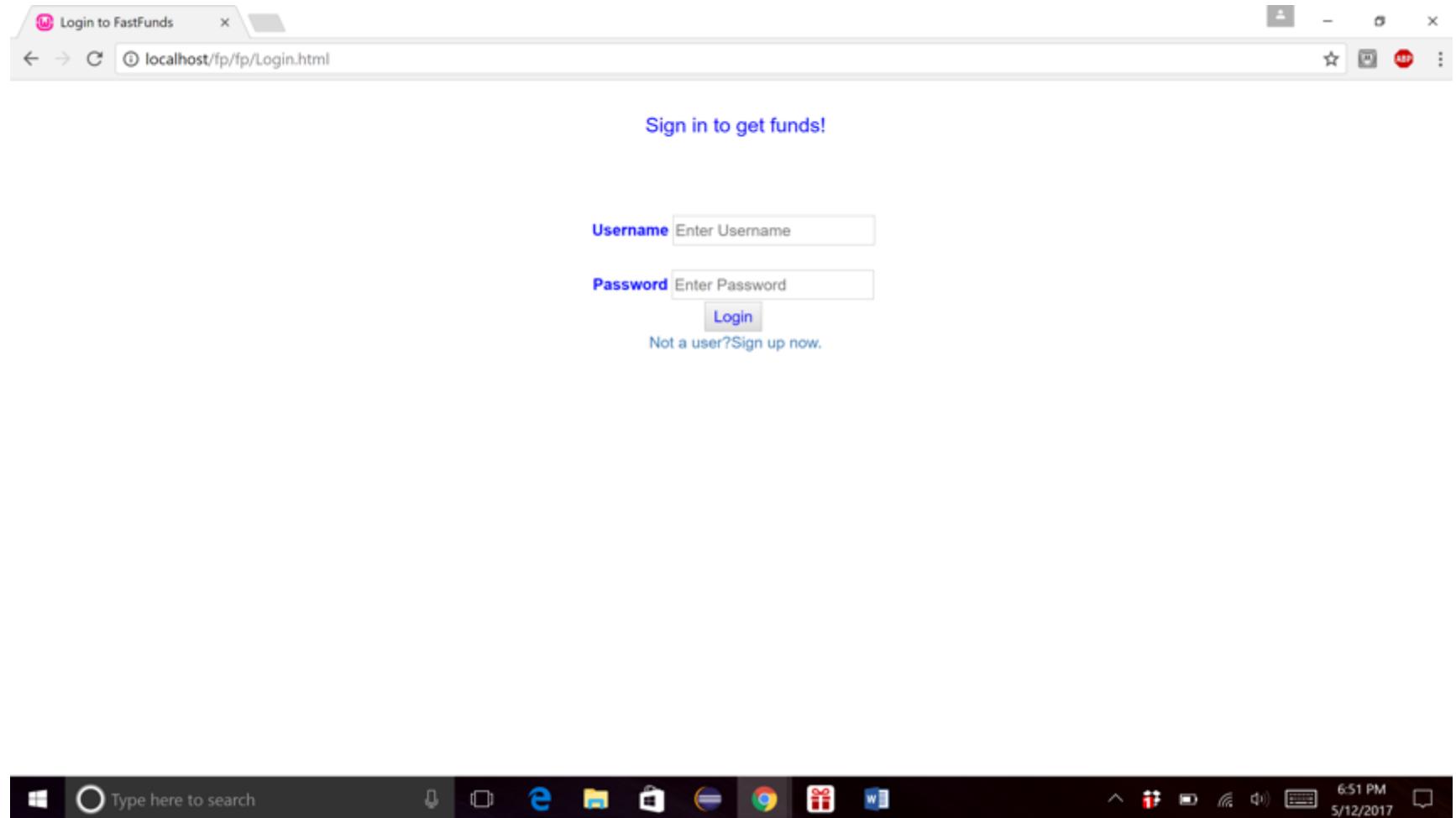
The screenshot shows a web browser window with the title bar "Sign up!" and the URL "localhost/fp/fp/newuser.php". The page contains a form with the following fields:

Field	Description
Email ID:	Enter email ID
Username:	Enter username
Password:	Enter password
Full name:	Enter your full name here
Street address:	Enter your street address
City:	Enter your City
State:	Enter your State
Zipcode:	Enter zipcode
Phone number:	Enter phone number
Credit Card number:	Enter your Credit Card number

At the bottom of the form is a "Submit" button.

## Login Page:

Here, the user must enter the user name and the password. These credentials are then tallied with the one stored in the database for the security and validation purpose. If they are a match, the user is logged in and they are directed to the action page.



## Login with incorrect username and/or password:

If the username or password is incorrect, error messages are displayed.



---

Incorrect username or password.

## Action Page:

Once a user is successfully logged in, he/she is navigated to the action page where a user can specify whether he/she want to create a project or funds a campaign. This is also the user's timeline.

What do you want to do

localhost/fp/fp/action.php

Fund a campaign.

Post a new campaign.

Alice Shaw  
New York  
NY  
Tom11alice23  
Erica Emory  
NYC  
NY  
Tom11Erica1  
Painting Pros  
An exhibition in the heart of New York City  
ongoing  
art  
345678

Lucy Rodriguez  
NYC  
NY  
Tom11Lucy  
SoundBot II: Charging Your Speakers Can't Get Any Easier  
Revolutionary & Powerful High-Performance Premium Wireless Speakers for Everyone w/ Qi Wireless Charging, Bluetooth, Siri, Dynamic Bass  
ongoing  
technology  
22

Oswald Red  
NYC  
NY

### Project Creation Page:

If the user clicks on create a project button, he/she is taken to project creation page. On the project creation page, the user must enter project name, project description, minimum funds and maximum funds for the project, multimedia needed for the project, campaign end date, project start date, project end date and project tag. All newly created projects will have project status as “accepting Funds” by default. Click on the submit button to create this new project.

The screenshot shows a web browser window with the following details:

- Title Bar:** Post a campaign!
- Address Bar:** localhost/fp/fp/post.php
- Content Area:**
  - Campaign name:** Enter campaign name.
  - Campaign description:** Enter campaign description.
  - Upload Photos:** Choose File No file chosen
  - Minimum funds needed:** Enter minimum funds to qualify.
  - Maximum funds needed:** Enter maximum funds needed.
  - The last date for campaigning is:** mm/dd/yyyy --:-- --
  - This project will be completed by:** mm/dd/yyyy --:-- --
  - Status:** Status ▾ (dropdown menu)
  - Tag:** Enter one appropriate tag.
  - Submit:** A button labeled "Submit".
- Taskbar:** Shows the Windows Start button, a search bar with "Type here to search", and various pinned application icons (File Explorer, Edge, File Hub, Google Chrome, Microsoft Word, Microsoft Excel). The taskbar also displays the date and time (5/12/2017, 6:53 PM).

**Home page:**

If the user clicks on the fund a campaign page, user can see all the projects that are recently being posted by other users. This is like the feature on Instagram where all the activities of different users are posted.

The screenshot shows a web browser window with the title bar "Campaigns" and the URL "localhost/fp/fp/Projects.php". The main content area features a large header "FastFunds" and the sub-header "Contribute to the awesome campaigns!". Below this is a navigation bar with links: Logo, Home, Users, Your Projects, Rate Projects, a dropdown menu "Tags", a "submit" button, Your Recommendations, a search bar, and a "Logout" link. The main content area displays several campaign cards. Each card has a blue header with the text "Save water." or "save trees". Below the header is the NYU Tandon School of Engineering logo and name. The body of each card contains the text "We need to save water." or "We need to save trees". There are also small purple progress bars at the bottom of each card. The cards are arranged in two rows: the top row has three cards, and the bottom row has three cards.

Save water.

NYU TANDON SCHOOL OF ENGINEERING

We need to save water.

Save water.

NYU TANDON SCHOOL OF ENGINEERING

We need to save water.

save trees

NYU TANDON SCHOOL OF ENGINEERING

We need to save trees

Save water.

Save electricity.

Save water.

## **Logout Page:**

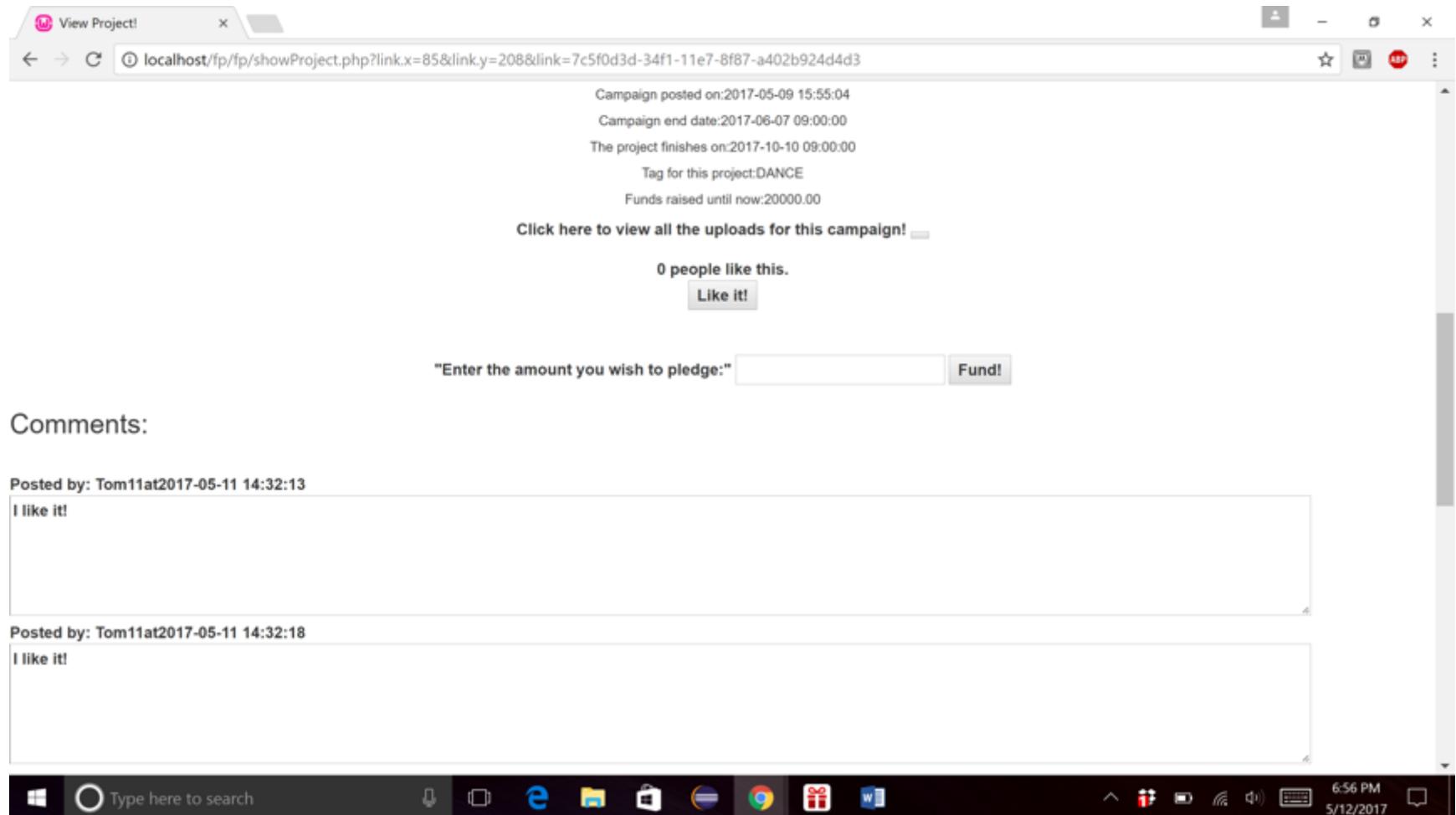
Every page has a logout button on the top right corner. User can click this button to log out and end his/her session when they are done using the website. Clicking this button ends the current session and the next time user opens the website, he/she must log in again.



## **Project Detail Page:**

On the homepage, all the projects are displayed in a grid format. A user can click on any of the project which will navigate him/her to the project detail page. On the project detail page, the project name, project description, campaign end date, project start date, project end date, minimum funds needed, maximum funds needed, project tag, amount of funds raised and project status are displayed. A user can specify the amount of money he/she wants to pledge for this project and this amount will be added to the FundsPledged table for the project and the total number of funds raised will be updated by adding this new amount.

The screenshot shows a web browser window displaying a project detail page for 'dance' on the 'FastFunds!' platform. The URL in the address bar is `localhost/fp/fp/showProject.php?link.x=85&link.y=208&link=7c5f0d3d-34f1-11e7-8f87-a402b924d4d3`. The page features a large yellow smiley face icon with the word 'dance' written below it. Below the icon, the text 'Let us groove.' and 'accepting funds' is visible. At the bottom, it states 'Minimum Funds needed:100.00' and 'Maximum Funds needed:20000.00'. The top navigation bar includes links for 'Logo', 'Home', 'Users', 'Your Projects', 'Rate Projects', 'Your Recommendations', a search bar, and a 'Logout' button. The 'Home' link is currently active, indicated by a dark background.

A screenshot of a web browser window showing a project page. The title bar says "View Project!". The address bar shows the URL "localhost/fp/fp/showProject.php?link.x=85&link.y=208&link=7c5f0d3d-34f1-11e7-8f87-a402b924d4d3". The page content includes campaign details: posted on 2017-05-09 15:55:04, end date 2017-06-07 09:00:00, and finishes on 2017-10-10 09:00:00. It also lists a tag "DANCE" and funds raised "20000.00". A link "Click here to view all the uploads for this campaign!" is present. Below this, it says "0 people like this." and has a "Like it!" button. A text input field for pledges is labeled "Enter the amount you wish to pledge:" followed by a "Fund!" button. The bottom of the screen shows a Windows taskbar with various icons and a search bar.

Campaign posted on:2017-05-09 15:55:04  
Campaign end date:2017-06-07 09:00:00  
The project finishes on:2017-10-10 09:00:00  
Tag for this project:DANCE  
Funds raised until now:20000.00

[Click here to view all the uploads for this campaign!](#)

0 people like this.  
[Like it!](#)

"Enter the amount you wish to pledge:"  Fund!

Comments:

Posted by: Tom11at2017-05-11 14:32:13  
I like it!

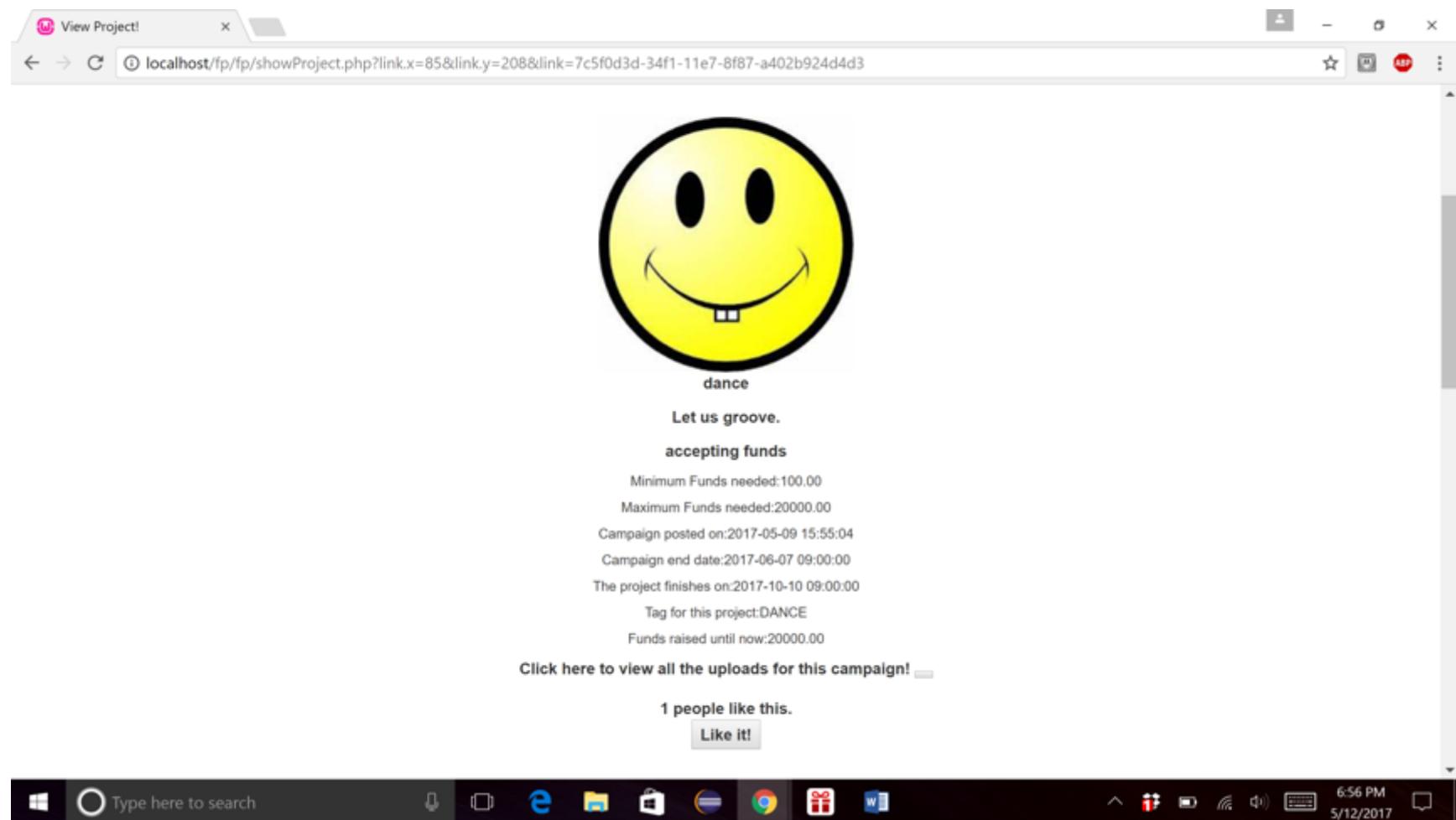
Posted by: Tom11at2017-05-11 14:32:18  
I like it!

Type here to search         

6:56 PM 5/12/2017

## Comment and Likes:

There is a comments section on the bottom on the project detail page where a user can post comments regarding the project. He/she can view the previous comment made by other users. Additionally, a user can like the project by clicking the like button and the number of likes are also displayed for the project.



The screenshot shows a web browser window displaying a project detail page. The title bar reads "View Project!". The URL in the address bar is "localhost/fp/fp/showProject.php?link.x=85&link.y=208&link=7c5f0d3d-34f1-11e7-8f87-a402b924d4d3". The main content area features a large yellow smiley face icon with the word "dance" written below it. Below the icon, there is a comment "Let us groove." and a section titled "accepting funds" which includes "Minimum Funds needed:100.00", "Maximum Funds needed:20000.00", "Campaign posted on:2017-05-09 15:55:04", "Campaign end date:2017-06-07 09:00:00", and "The project finishes on:2017-10-10 09:00:00". It also lists "Tag for this project:DANCE" and "Funds raised until now:20000.00". A link "Click here to view all the uploads for this campaign!" is present. At the bottom, it says "1 people like this." and has a "Like it!" button. The taskbar at the bottom shows various icons and the system tray indicates the date and time as 6:56 PM, 5/12/2017.

View Project! X

localhost/fp/fp/showProject.php?link.x=85&link.y=208&link=7c5f0d3d-34f1-11e7-8f87-a402b924d4d3

Posted by: Tom11at2017-05-11 12:40:13  
Very nice!

Posted by: Tom11at2017-05-11 12:04:26  
Hi!!

Posted by: Tom11at2017-05-11 12:40:17  
Very nice!

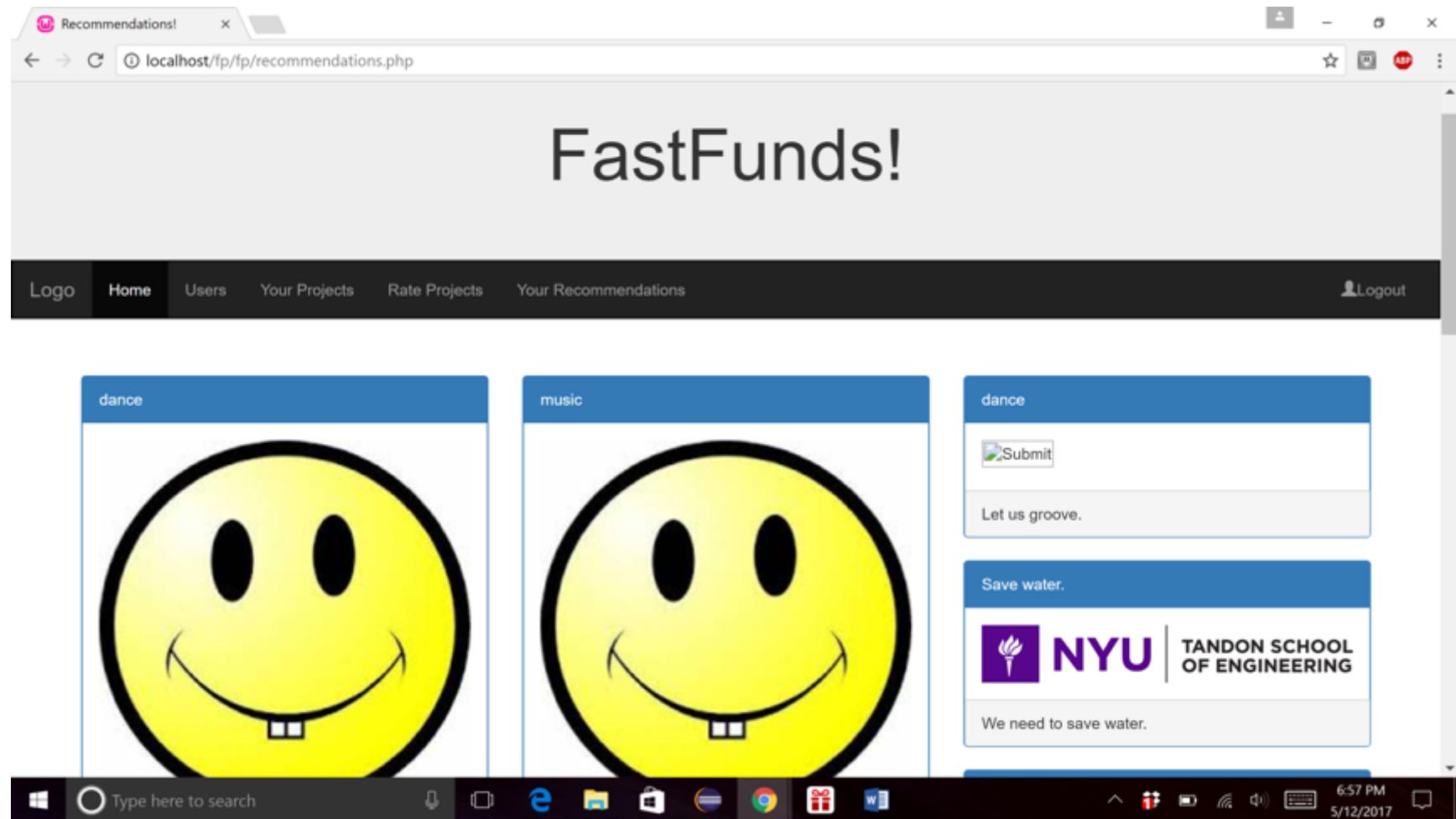
**YOUR COMMENT:** Enter your comment here.

FastFunds Copyright

Type here to search             6:56 PM 5/12/2017

## Recommendations Page:

User can click on the recommendations tab to view all the projects which have been recommended by the system. Recommendations are given based on user clicks which are fetched from the activity logs which are stored in the database.



## All Users Page:

User can click on the Users tab to view all the users who are using this system. For each user, username, the city he/she belongs to and a follow button are provided. To follow a user, click on the follow button after which the user will receive their posts and updated on his/her timeline.

The screenshot shows a web browser window titled "FastFunds Users" with the URL "localhost/fp/fp/allUsers.php". The page has a dark header bar with links for "Logo", "Home", "Users", "Your Projects", "Rate Projects", and "Logout". Below the header, a red banner reads "Here are our valued Users!!".

The main content displays two user profiles:

- Alice Shaw**:
  - Profile picture
  - Username: Alice Shaw
  - Followers: 2
  - Following: 4
  - Email Address: alice@gmail.com
  - From: New York, NY
  - FOLLOW** button
- Bob Dillan**:
  - Profile picture
  - Username: Bob Dillan
  - Followers: 2
  - Following: 4
  - Email Address: bob@test.com
  - From: [unclear]
  - FOLLOW** button

At the bottom of the screen, the Windows taskbar is visible with icons for File Explorer, Edge, Task View, Start, and others. The system tray shows the date and time as "5/12/2017 6:57 PM".

## Timeline Page:

This page displayed all the projects of the users which the current user has followed

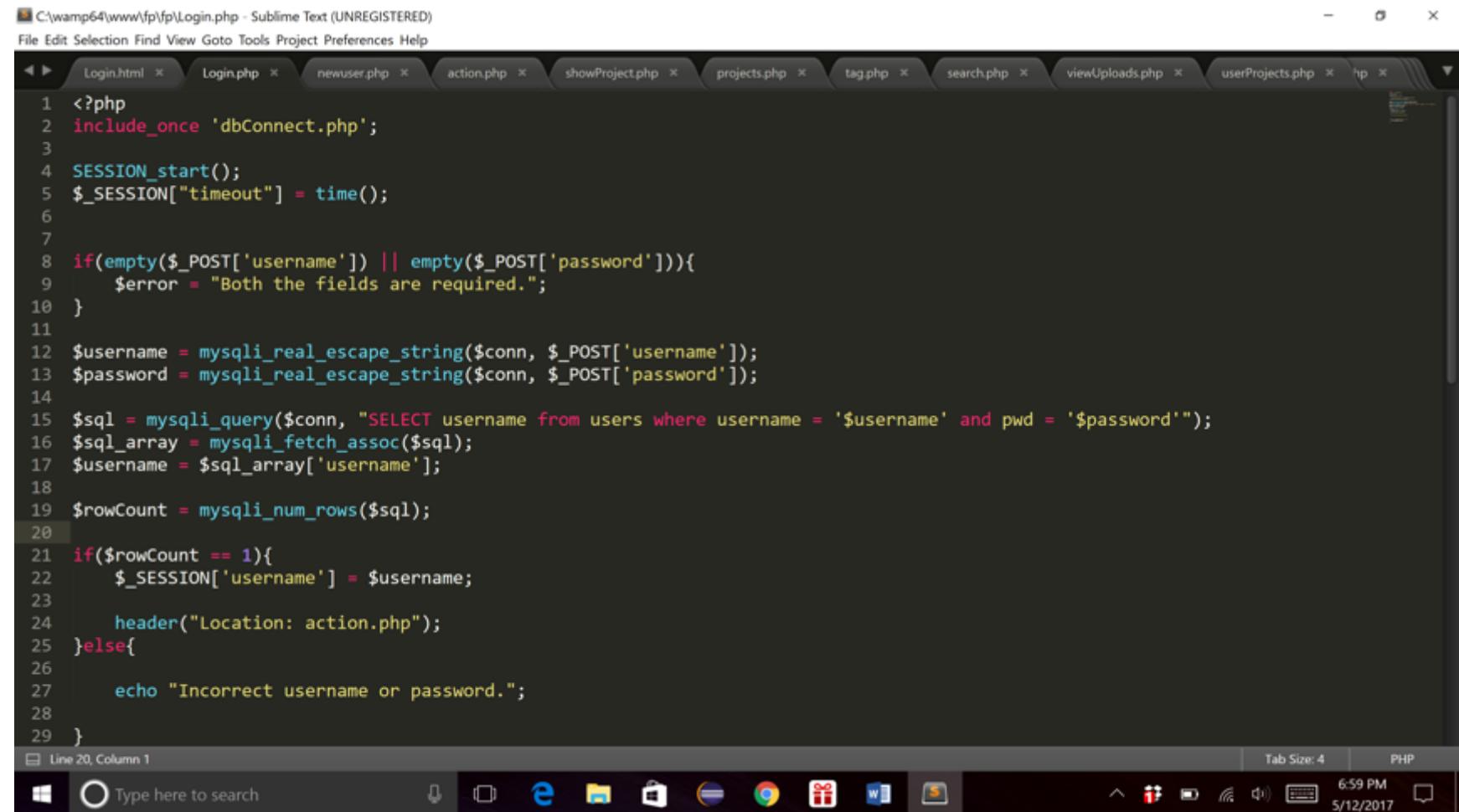
The screenshot shows a web browser window displaying a timeline page. The address bar indicates the URL is `localhost/fp/fp/action.php`. The page content lists several users and their posts:

- Alice Shaw  
New York  
NY  
Tom11alice23  
Erica Emory  
NYC  
NY  
Tom11Erica1  
Painting Pros  
An exhibition in the heart of New York City  
ongoing  
art  
345678
- Lucy Rodriguez  
NYC  
NY  
Tom11Lucy  
SoundBot II: Charging Your Speakers Can't Get Any Easier  
Revolutionary & Powerful High-Performance Premium Wireless Speakers for Everyone w/ Qi Wireless Charging, Bluetooth, Siri, Dynamic Bass  
ongoing  
technology  
22
- Oswald Red  
NYC  
NY  
Tom11Oswald06

At the bottom of the browser window, the Windows taskbar is visible, showing the Start button, a search bar with the placeholder "Type here to search", and icons for File Explorer, Edge, Task View, File Explorer, Google Chrome, and Word.

### 3.2 Additional Features:

- In the event of cross site scripting attacks on the website, we have used mysqli\_real\_escape\_string() and htmlspecialchars() to avoid SQL injections.



The screenshot shows a Sublime Text window with multiple tabs open at the top, including 'Login.html', 'Login.php', 'newuser.php', 'action.php', 'showProject.php', 'projects.php', 'tag.php', 'search.php', 'viewUploads.php', 'userProjects.php', and 'fp'. The 'Login.php' tab is active, displaying the following PHP code:

```
<?php
include_once 'dbConnect.php';

SESSION_start();
$_SESSION["timeout"] = time();

if(empty($_POST['username']) || empty($_POST['password'])){
    $error = "Both the fields are required.";
}

$username = mysqli_real_escape_string($conn, $_POST['username']);
$password = mysqli_real_escape_string($conn, $_POST['password']);

$sql = mysqli_query($conn, "SELECT username from users where username = '$username' and pwd = '$password'");
$sql_array = mysqli_fetch_assoc($sql);
$username = $sql_array['username'];

$rowCount = mysqli_num_rows($sql);

if($rowCount == 1){
    $_SESSION['username'] = $username;
    header("Location: action.php");
} else{
    echo "Incorrect username or password.";
}

```

The code handles user input validation, database queries using prepared statements (implied by the escaping), and session management. It also includes basic error handling and redirection logic.

The screenshot shows a Sublime Text window with multiple tabs open, including 'action.php' which is the active tab. The code in 'action.php' is a PHP script that retrieves user information from a MySQL database and displays it. It uses mysqli\_query and mysqli\_fetch\_array functions to fetch data and echo it back to the browser. The code also includes logic to query the 'Projects' table based on the user's username.

```
31 $result1 = mysqli_query($conn, $sql1);
32
33 while($row = mysqli_fetch_array($result1)) {
34
35     echo "<br>";
36     echo "<b>";
37     echo htmlspecialchars($row["name"]);
38     echo "<br>";
39     echo htmlspecialchars($row["city"]);
39     echo "<br>";
40     echo htmlspecialchars($row["state"]);
40     echo "<br>";
41     $u = $row["username"];
42
43     $sql3 = "select * from Projects where username='$u'";
44
45     echo "$username";
46     echo "$u";
47     $result3 = mysqli_query($conn, $sql3);
48
49     while($row3 = mysqli_fetch_array($result3)) {
50
51         echo "<br>"; "<br>"; "<br>"; "<br>"; "<br>";
52
53         echo "<b>";
54         echo htmlspecialchars($row3["pname"]);
54         echo "<br>";
55         echo htmlspecialchars($row3["pdescription"]);
55         echo "<br>";
56         echo htmlspecialchars($row3["status"]);
56         echo "<br>";
57         echo htmlspecialchars($row3["tag"]);
57         echo "<br>";
58         echo htmlspecialchars($row3["likes"]);
58         echo "<br>";
59
59
59
```

- We have implemented the user activity logs in the database based on the user clicks

MySQL Workbench

Local instance MySQL57

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema

SCHEMAS

No object selected

SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* creatingTables insertingData Queries SQL File 10\*

```
1 use Final_Project;
2
3 select * from users;
4
5 select * from creditcard;
6
7 select * from discussion;
8
9 select * from log;
10
11
12 SELECT distinct(tag) from Projects;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

username	logentry
Tom11	Mike23

log 1 x Apply Revert

Action Output

#	Time	Action	Message	Duration / Fetch
1	19:00:51	use Final_Project	0 row(s) affected	0.000 sec
2	19:00:54	select *from log LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

7:01 PM 5/12/2017

## **4. Conclusion:**

This project has enabled us to understand the concepts of the database in a better way and implementing it on a real time working application. It has also helped us understand the basic difference between theoretical knowledge and practical implementation. It allowed us to solve some of the most trivial implementation issues, that are certainly going to help in other feature endeavors as well. The most challenging part of designing the certain database and normalizing it helped us understand the cost specific metrics of implementations. It was a challenging but a very interesting experience to work on this project.

This project is up on GitHub at: <https://github.com/sunishu/FastFunds>

## **5. References:**

- <http://stackoverflow.com>
- <https://www.w3schools.com>
- <http://getbootstrap.com>