

Project Report

Project Funding System

Meng Wang(mw3168) Yi Qin(yq468)

1. Introduction

Users are required to use their emails to register/log in for this fund platform. Their ids are the primary keys. Once registered and log in, they will own or modify his/her profile and then come into a project square. They could post projects or pledge for other interesting projects here (we allow a user pledging one project multiple times), and they could also like, comment these projects. They can also search the interesting project by keywords which maybe exist in description, tags. Users can also follow each other. When a user pledges a project, we would update the project raised money. If the raised money reaches the max money of project, we would insert this project into PublishedProject which stores all that successful pledging projects and drag it off from the project squares which are still waiting for investment. Besides, when a new day comes, we shall check the system time whether it meets the end time of this project. If it meets the time, meaning the project is expired or fail to get money. Then it would not show up in project squares any more and if the raised money reaches min money, it will be inserted into the PublishedProject. When a project is announced to be finished, the sponsors could rate it based on the videos, pictures and music it provided. Based on the users behaviors above, we would recommend some projects in the home page next time he/she logs in.

2. Database Design and Explanation

2.1 Overall Table Structure

User(id, email, username, password, hometown, interests, credit_card, legalname)
Project(pid, pname, desp, minmoney, maxmoney, endtime, tags, release_time)
PublishedProject(pid, create_time, fundmoney, status)
Follow(follower, followee, follow_time)
Post(uid, pid, post_time)
Like(uid, pid, like_time)
Comment(id, uid, pid, comment_time, comments, replied_id, replied_name)
Pledge(id, uid, pid, pledge_time, amount, transaction_status)
Rate(uid, pid, create_time, rating, rate_time)
Sample(sid, multimediaAddr, textarea, time)
Contain1(pid, sid)
Tag(tid, pname)
Contain2(pid, tid)

User([id](#),[email](#),[username](#),[password](#),[hometown](#),[interests](#),[credit_card](#),[legalname](#))
[id](#): an increment integer which serve as the primary key.

[email](#): the user must use their email to login or register.

[username](#): username can be changed by user since it is not the primary key.

[hometown](#): the basic information to fill, can be N/A

[interests](#): the basic information to fill, can be N/A

[credit_card](#): the credit card number to fill with for charging

[legalname](#): the official legal name of user

Primary key: id

Project([pid](#),[pname](#),[desp](#),[minmoney](#),[maxmoney](#),[endtime](#),[tags](#),[release_time](#))

[pid](#): project unique number as primary key.

[pname](#): project name is filled by user

[desp](#): every project can have a project description.

[minmoney](#): the minimum money for a project to start.

[maxmoney](#): the maximum money needed for a project. When this is achieved, the project will jump to successful pledge stage and investor will be charged.

[endtime](#): the deadline for a project in waiting for pledge stage. After this time, the project will jump to successful pledge stage or fail to pledge stage based on whether it get the min money.

[tags](#): the tag or category of project.

[release_time](#): the estimated time to finish the project which is set by user.

Primary key: pid

PublishedProject([pid](#),[create_time](#),[fundmoney](#),[status](#))

[pid](#): the project is which is defined in Project table.

[create_time](#): the time when the project raises money successfully and under rating.

[fundmoney](#): the final total pledged money for this project.

[status](#): when a project gathers money successfully, it will be inserted into published project table and there are two status: complete and pending. So this status represents whether the project announced has been finished by the author or not.

Follow([follower](#),[followee](#),[follow_time](#))

[follower](#): user who want to follow some other user.

[followee](#): user who is followed by others.

[follow_time](#): the time follower followed the followee.

Primary key: follower, followee

Foreign key Follow(follower) references User(email)

Foreign key Follow(followee) references User(email)

Post(email, pid, post_time)

email: the account which is defined in User table.

pid: the project is which is defined in Project table.

post_time: the time user post the project.

Primary key: follower, followee

Foreign key Post(email) references User(email)

Foreign key Post(pid) references Project(pid)

Like(uid, pid, like_time)

uid: the account which is defined in User table.

pid: the project is which is defined in Project table.

like_time: the time user typed like button of the project.

Primary key: uid, pid

Foreign key Like(uid) references User(id)

Foreign key Like(pid) references Project(pid)

Comment(uid, pid, comment_time , comments,replied_id,replied_name)

uid: the account which is defined in User table.

pid: the project is which is defined in Project table.

like_time: the time user posted a comment of the project.

comments: comment content.

replied_id: replier's id.

replied_name: replier's name.

Primary key: uid, pid, comment_time

Foreign key Comment(uid) references User(id)

Foreign key Comment(pid) references Project(pid)

Pledge(uid, pid, pledge_time, amount, transaction_status)

uid: the account which is defined in User table.

pid: the project is which is defined in Project table.

pledge: the time investor/user pledged the project.

amount: the amount of money investor pledged.

transaction_status: The status of pledging money. It can be failed/ successful / pending.

Primary key: uid, pid, pledge_time

Foreign key Pledge(uid) references User(id)

Foreign key Pledge(pid) references Project(pid)

Rate(uid, pid, rating, rate_time, create_time)

uid: the account which is defined in User table.

pid: the project is which is defined in Project table.

rating: the evaluation by investor

rate_time: the time when investor posted the evaluation.

create_time: the time when the project raises money successfully and under rating.

Primary key: uid, pid, pledge_time

Foreign key Rate(uid) references User(id)

Foreign key Rate(pid,create_time) references PublishedProject(pid,created_time)

Sample(sid,multimediaAddr,textarea,time)

sid: the unique sample number which is the primary key.

multimediaAddr: to store the multi media content of sample. e.g. picture, video, audio.

The type is BLOB, Binary Large Object type for multi media storage in MySQL. What is actually stored in database is a link for here.

text area: text content of sample.

time: the post time of sample.

Primary key: sid

Contain1(pid,sid)

pid: the project is which is defined in Project table.

sid: the sample which is defined in Sample table.

Primary key: pid, sid

Foreign key Contain(pid) references Project(pid)

Foreign key Contain(sid) references Sample(sid)

Tag(tid,tname)

tid: tag id.

tname: tag name.

Primary key: id

Contain2(pid,tid)

pid: the project is which is defined in Project table.

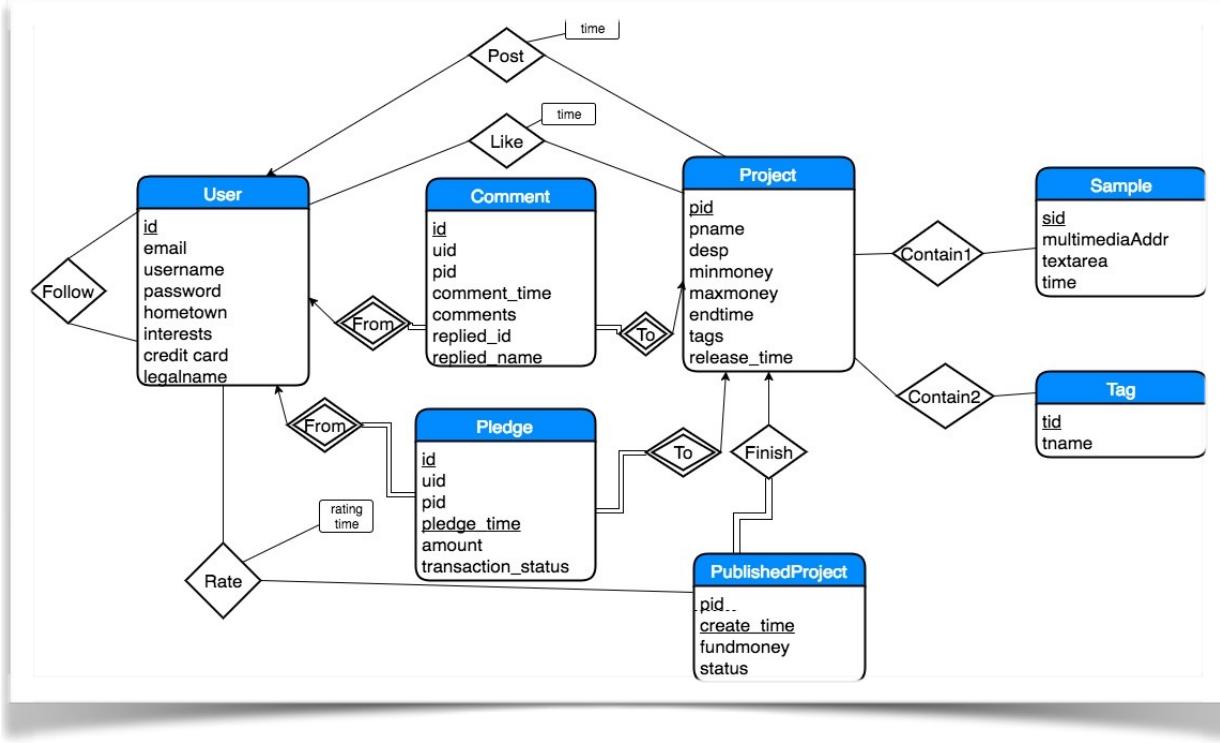
tid: the tag which is defined in Sample table.

Primary key: pid, tid

Foreign key Contain(pid) references Project(pid)

Foreign key Contain(tid) references Tag(tid)

2.2 ER Model



2.3. Triggers in Database

- When a new record is added to Table Pledge, **Pledge_after_insert trigger** will check whether the sum of money exceed the max money of corresponding project and then we need to add a new record in Table PublishedProject:

```

CREATE DEFINER='root'@'localhost' TRIGGER `kickstar`.`Pledge_AFTER_INSERT`
AFTER INSERT ON `Pledge` FOR EACH ROW
BEGIN
| IF EXISTS (SELECT pid FROM Project natural join
| (SELECT pid, sum(amount) as sum_am FROM Pledge
| WHERE pid = New.pid GROUP BY pid) as B
| WHERE sum_am >= maxmoney)
| THEN INSERT INTO `PublishedProject` VALUES (New.pid, now(),
| (SELECT sum(amount) as sum_am FROM Pledge
| WHERE pid = New.pid), 'pending');
| END IF;
END

```

Testing the trigger:

Before inserting new record in Pledge, PublishedProject table:

pid	create_time	fundmoney	status
4	2017-03-30 10:56:01	200.00	pending

Insert a new Pledge:

```
INSERT INTO `kickstar`.`Pledge`  
VALUES('Monica', 2, NOW(), 300);
```

After insertion,

pid	create_time	fundmoney	status
2	2017-04-16 18:00:52	350.00	pending
4	2017-03-30 10:56:01	200.00	pending

We can see, the project 2 go to publishedProject automatically.(Maxmoney = 300)

2. When a new record is added to Table PublishedProject,

PublishedProject_After_Insert trigger will changing status to posted in Table Post:

```
1 DELIMITER $$  
2 CREATE TRIGGER charge_pledge| AFTER INSERT ON PublishedProject  
3 FOR EACH ROW  
4 UPDATE Pledge  
5 SET transaction_status = "posted"  
6 WHERE pid = NEW.pid;  
7 $$  
8 DELIMITER ;
```

Objects	* Untitled @fundclo... X	* Untitled @fundclo... X	PublishedProject
	email pid pledge_time amount transaction_stat		
	bob@gmail.com 1 2017-04-02 14:31: 5.00 posted		
►	cathy@gmail.com 4 2017-03-10 14:30: 20.00 pending		
●	kate@gmail.com 4 2017-02-20 14:39: 20.00 pending		
●	stacey@gmail.com 4 2017-04-01 14:31: 35.00 pending		
	cathy@gmail.com 5 2017-03-11 14:30: 10.00 posted		

```
1 INSERT INTO `PublishedProject` VALUES(4,"2017-4-15","XXX",100000)|
```

All pledging relevant to project 4 is changed to posted successfully.

email	pid	pledge_time	amount	transaction_st
bob@gmail.com	1	2017-04-02 14:31:	5.00	posted
► cathy@gmail.com	4	2017-03-10 14:30:	20.00	posted
● kate@gmail.com	4	2017-02-20 14:39:	20.00	posted
● stacey@gmail.com	4	2017-04-01 14:31:	35.00	posted
cathy@gmail.com	5	2017-03-11 14:30:	10.00	posted

3. Web Platform implementation of System

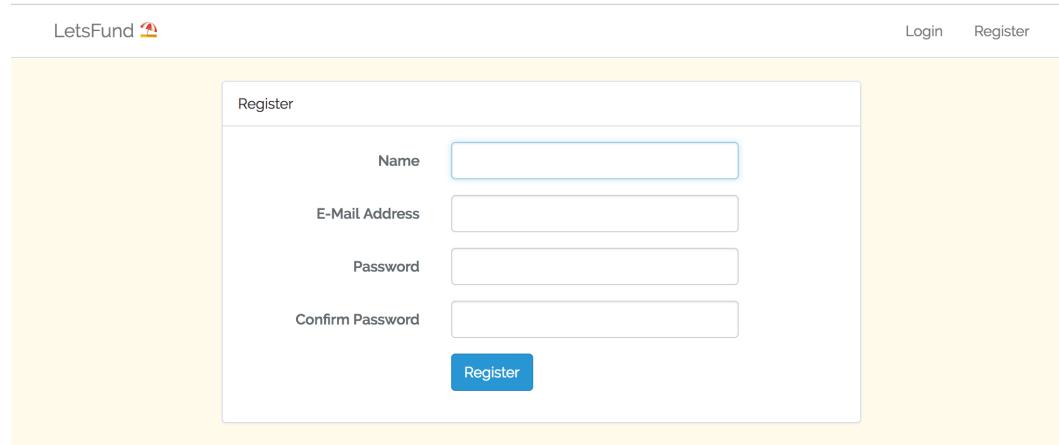
3.1 Configure Backend Framework and Connection to network and database

We choose the **Laravel Framework** as the backend foundation for the reason that the laravel framework is based on PHP and provide many basic internal properties:

- Route Caching: Route caching feature speeds up the application route registration.
- Authentication: Authentication is the most important part of any web application and developers spent enormous time writing the authentication code. Laravel 5 contains the ready to use inbuilt authentication system.
- Directory structure: All the classes are fully name spaced now.
- Session, cookies and the encryption of password along network and in database.
- Convenience: many operations are included and just modify some parameters in config file, such as configuring the socket and connect to network, file system arrangement.

3.2 Basic Function Implementation and Design

1. Register



The screenshot shows a 'Register' form within a web application. The form is titled 'Register' and contains four input fields: 'Name', 'E-Mail Address', 'Password', and 'Confirm Password'. Below the fields is a blue 'Register' button. The background of the page is a light yellow color. At the top, there is a navigation bar with the text 'LetsFund' and icons for search and refresh, followed by 'Login' and 'Register' links.

Operation in database:

Create a record for a new user account, with a email address, a name, and a password.

```
DB::insert('insert into users (name, email, password) values (?, ?, ?)', [$data['name'], $data['email'], bcrypt($data['password'])]);
```

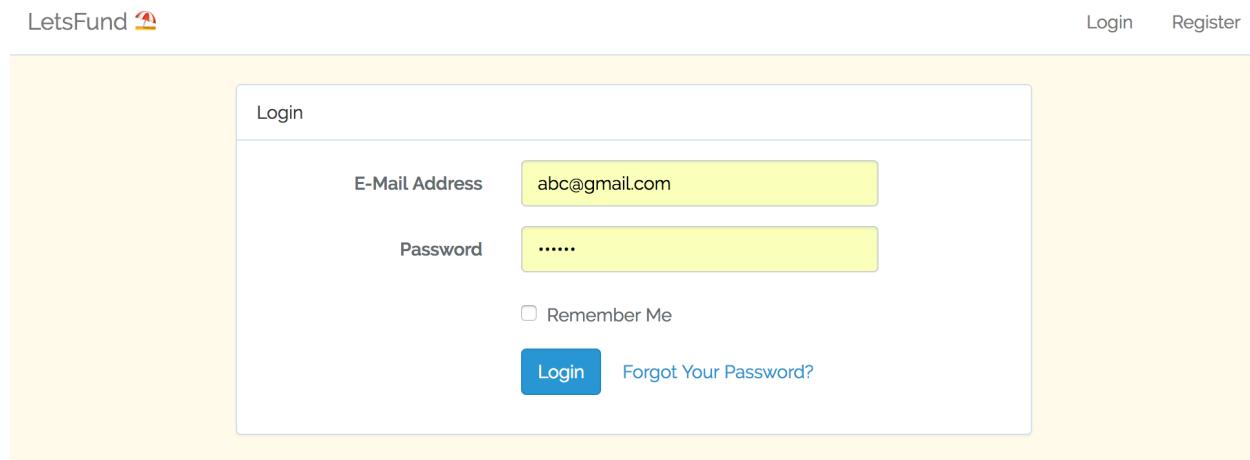
Use the prepared statement for insertion and add the tuple to the database

Sample table:

We can see the password is encrypted by MD5 to ensure the security of password. And token for verifying user identity.

id	name	email	password	remember_token	created_at	updated_at
1	cathy	cathy@gmail.com	\$2y\$10\$RFjU.e3X/jNQqymcDr4jaejK1tVyCvGK...	739wGsm90JAOSKnMuFKhw43S20YsnE4a62S...	2017-04-19 20:45:56	2017-04-19
2	Jiaxiang Lin	jiaxiang.lin@nyu.edu	\$2y\$10\$a2zUhrRAxKhgCIYT2.u/3OKG.rpJcgO...	s6t1KrXYGh7aBCl4Q1g1JMZEuEsQYHuQWN...	2017-04-19 20:50:57	2017-04-19
3	mengmeng	mengmeng@gmail.com	\$2y\$10\$AVa/oUD2iovbjzdYyfSPJ.s/A2iFLjKtTsF...	wgd77g0hhpljxfcaYS7uhp5izWl2u2Jel092jO...	2017-04-28 23:00:51	2017-04-28
4	Yi Qin	yq468@nyu.edu	\$2y\$10\$SWY8tOj8casHeY4yepmyKOTyQFNQY...	ttyW4DlpHoMuAN836XTEUWLixxb3suJ9aUW0...	2017-04-29 15:00:22	2017-04-29
5	Katelyn	qinyi468@gmail.com	\$2y\$10\$TXJrsGghFbz9OOnmPlZvevlex.4z8c....	ElstCNtsFrPpGISNBHFVhjCcl0JQ1Jf7gVpCaEr...	2017-04-29 15:02:39	2017-04-29
6	katelyn1	qinyi4681@gmail.com	\$2y\$10\$FB08clRrEKYM2lWjdhlQoxxaWm1Hc...	XCP4vOBi4hfChoxjnCLW5wYe1ZUwZYsuRm...	2017-04-29 15:05:15	2017-04-29
7	yy	qi@sina.com	\$2y\$10\$Wip5ZUyY5fbOe3ICmZG8OijCjUF8...	mkY8Yi12oH5moWsOi1hWH1FYoxOm4d9Q6N...	2017-05-02 03:10:25	2017-05-02
8	abc	abc@gmail.com	\$2y\$10\$SnRwcFMmDzICalzSV1Vxyeuk67eBBii...	BGzbwvN5Lnp4M05hoXbndo NkaNSqUsgL...	2017-05-03 19:17:10	2017-05-03
9	uuu	uuu@gmail.com	\$2y\$10\$sl3ca.31.KuC8xSk6dOhxeoAjw3gakR...	HyMMIiwtld2LqrPX1XoP82quGlYxUPJiAhiO...	2017-05-05 14:59:25	2017-05-05
10	yjtyyj	rgr@efs.com	\$2y\$10\$Y0JyW2mkOYCdf4Go4kXZMOBt3Ogc...	8e5RbZtpma0fGCWrutens41bwL7X90UqWgj3V...	2017-05-05 15:07:51	2017-05-05

2. log in



The screenshot shows a login page for 'LetsFund'. At the top right are 'Login' and 'Register' buttons. The main area has a light yellow background and contains a white login form. The form has a header 'Login'. It includes fields for 'E-Mail Address' (containing 'abc@gmail.com') and 'Password' (containing '.....'). Below these is a 'Remember Me' checkbox followed by 'Login' and 'Forgot Your Password?' buttons.

For the log in function, we provide remember me and forget the password function. If you forget the password, we will send you a email for resetting password. If you type remember me, you do not need to type password for the second time signing in.

3. Create the profile

You can upload your avatar and type some basic information, such as hometown, date of birth, interest, credit card number and legal name. If you do not upload the avatar, a default image will be used for your avatar.

Edit Profile

Upload Avatars
 No file chosen

Hometown:

Date of Birth:

Interest:

Credit Card:

Legal Name:

Operation in database:

Create a record in table user_profiles and store the information user types.

```
DB::insert('insert into user_profiles (id, imagename, hometown, birthday, interest, creditcard, legalname) values (?, ?, ?, ?, ?, ?, ?)', [$id, $imageName, $request['hometown'], $request['birthday'], $request['interest'], $request['creditcard'], $request['legalname']] );
```

Use SQL query embedded with prepared statement.

Resulting table:

id	hometown	birthday	interest	creditcard	legalname	imagename
1	Shenzhen	1993-10-08 00:00:00	gym	123456	cathy wang	default_profile.jpg
2	Harbin	1992-11-06 00:00:00	movie	234567	Jiaxiang Lin	default_profile.jpg
3	new york	2017-04-28 00:00:00	reading	456789	meng wang	default_profile.jpg
6	beijing	2017-04-30 00:00:00	music	1123444	yiqin	user6.JPG
8	beijing	2017-05-03 00:00:00	fff	12345	ddd	user8.png
9	dwdad	2017-05-05 00:00:00	sdasdada	1231321	rgeer	default_profile.jpg
10	defe	2017-05-05 00:00:00	eeefe	123425	bfdbf	user10.jpg
11	ooo	2017-05-05 00:00:00	deuce	12312312323	dcd	user11.jpg
12	yujj	2017-05-05 00:00:00	bjkbkjbjk	657567567	bjkb nkj	user12.jpg
14	beijing	2017-05-09 00:00:00	ddd	12333	dddd	user14.jpg

4. Create a project

The screenshot shows the 'Post a new project' form on the LetsFund website. The form includes fields for Project Name, Description, file uploads, Fund Ends At, and Tags, along with a large central text area for the project description.

Project Name:

Upload a Project Cover
Choose File No file chosen

Description:

Upload files in Description

- Upload image file
Choose File No file chosen
- Upload audio file
Choose File No file chosen
- Upload video file
Choose File No file chosen

Fund Ends At:

Target Minimum Money:

Target Maximum Money:

Project Release Time: (x) ▾

Currently Raised Money:

Tags:

Add Project

Start a new project and fill out the information which is needed for a new project.

Operation in database:

```
DB::insert('insert into projects (pid, user_id, pname, desp, minmoney, maxmoney, endtime, release_time, raisedmoney, projectcover) values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)', [$project->pid, $request['user_id'], $request['pname'], $request['desp'], $request['minmoney'], $request['maxmoney'], $request['endtime'], $request['release_time'], $request['raisedmoney'], $request['projectcover']]);
```

We create a new record for the new project in project table.

Resulting table:

pid	user_id	pname	desp	minmoney	maxmoney	endtime	release_time	raisedmoney	projectcover
1	1	project music	this is a music project	10000.00	20000.00	2017-04-18 16:51:09	2017-04-27 16:51:13	10101.00	user6number7.jpg
2	2	jiaxiang's project	This is a project created by jiaxiang	111.00	222.00	2017-05-19 00:00:00	2017-06-19 00:00:00	0.00	user6number8.jpg
3	1	cathy's project	cathy's project	100.00	200.00	2017-06-19 00:00:00	2017-07-19 00:00:00	229.00	default_cover.jpg
4	1	test tag	test tag	10000.00	50000.00	2017-06-21 00:00:00	2017-04-19 00:00:00	6004.00	default_cover.jpg
5	1	request project	this is a test	20000.00	30000.00	2017-04-22 00:00:00	2017-04-26 00:00:00	0.00	default_cover.jpg
8	1	test tags array	test a tags array	100.00	200.00	2017-04-21 00:00:00	2017-04-21 00:00:00	0.00	default_cover.jpg
9	1	4 project	4project	11100.00	222000.00	2017-04-22 00:00:00	2017-05-20 00:00:00	5000.00	default_cover.jpg
10	1	project 5	5th project	3300.00	4400.00	2017-05-22 00:00:00	2017-06-22 00:00:00	4540.00	default_cover.jpg
18	1	6th project	this is the 6th project!	8000.00	9000.00	2017-04-27 00:00:00	2017-05-25 00:00:00	9030.00	default_cover.jpg
19	2	project jazz	this is a jazz project	30000.00	40000.00	2017-05-27 00:00:00	2017-06-27 00:00:00	32869.00	default_cover.jpg

5. read, like, and comment on project

Let's take photo Edit

End Time: 11 months from now Created By:Puloceba



Tags: photos

1

Raised Money: 0

Minimum Money: 100

Maximum Money: 1000

0%

Pledge

Description Updates Comments Pledges History

This is a photography album and people all like photography.

Image File: 

Audio File: 

Video File: 



- Read and like:** In this project detail page, you can read the description , which contain some text, image, audio, and video files and the amount of money it already have. The progress bar shows the money rate now/max money. Also you can click the heart button to like this project. After you like it, it will appear in your like feeds page. Also, the digit on the right hand side of heart button is the number of likes this project get.

- Edit:**

Also, if you are the author of this project, you can click edit to edit the information of this project.

- Check the author's profile:**

click the link of author

- Pledge:**

Click the pledge button to fund this project.

- See the projects' updating or posting and pledge history**

- Comment it, reply other's reply and see other's comment:**

You can add a new comment to this project or you can reply others' comments

Operation in database:

```
DB::insert('insert into comments (body, project_pid, user_id, replied_id, replied_name)
values (?, ?, ?, ?, ?)', [$body, $this->pid, Auth::user()->id, $replied_id, $replied_name]);
```

We create a new record for the new comment in Comments table.

Resulting table:

id	project_pid	user_id	body	created_at	updated_at	replied_id	replied_name
1	18	1	my first comment	2017-04-25 15:42:21	2017-04-25 15:42:26	NULL	NULL
2	18	1	here is my comment	2017-04-25 20:21:50	2017-04-25 20:21:50	NULL	NULL
4	10	1	I'm commenting the 5th project	2017-04-25 20:38:45	2017-04-25 20:38:45	NULL	NULL
5	10	1	another comments	2017-04-25 20:43:26	2017-04-25 20:43:26	NULL	NULL
6	10	1	another try!	2017-04-25 20:49:09	2017-04-25 20:49:09	NULL	NULL
7	18	1	cathy leaves a comment here.	2017-04-25 20:49:44	2017-04-25 20:49:44	NULL	NULL
8	19	1	cathy leaves a comment here	2017-04-28 20:56:58	2017-04-28 20:56:58	NULL	NULL
9	19	2	Jiaxiang leaves a comment here	2017-04-28 22:06:38	2017-04-28 22:06:38	1	Jiaxiang Lin
10	19	2	hi!	2017-04-28 22:07:06	2017-04-28 22:07:06	2	Jiaxiang Lin
11	4	1	hi	2017-04-28 18:10:18	2017-04-28 18:10:20	NULL	NULL

6. pledge money for projects

Pledge a project

Pledge Money: 0

pledge

You can type the amount of money that you want to pledge.

LetsFund 24 Start a project Project Square abc

Thanks for your sponsorship!

create aaaaa

End Time: 4 weeks from now Created By:Puloceba

Tags: photos

0

Raised Money: 90

Minimum Money: 100

Maximum Money: 1000

9%

Pledge

After you type pledge, the raised money will be changed and the also the progress bar. The session message will hint you that you pledge successfully.

```
DB::update('update projects set raisedmoney = ? where pid = ?',
           [($project->raisedmoney)+$amount, $project->pid]);
```

Resulting table:

pid	user_id	pname	desp	minmoney	maxmoney	endtime	release_time	raisedmoney	projectcover
45	14	create a new	aaaaaaa	100.00	1000.00	2018-05-09 00:00:00	2018-05-09 00:00:00	0.00	user14number7.jpg
46	14	create a new	This a charity activity!	100.00	1000.00	2018-05-09 00:00:00	2018-05-09 00:00:00	0.00	user14number8.jpg
47	14	create aaaaa	ffffvvve	100.00	1000.00	2017-06-09 00:00:00	2017-06-09 00:00:00	0.00	default_cover.jpg
48	14	create aaaaa	ffffvvve	100.00	1000.00	2017-06-09 00:00:00	2017-06-09 00:00:00	0.00	default_cover.jpg
49	14	create aaaaa	ffffvvve	100.00	1000.00	2017-06-09 00:00:00	2017-06-09 00:00:00	0.00	default_cover.jpg
50	14	create aaaaa	ffffvvve	100.00	1000.00	2017-06-09 00:00:00	2017-06-09 00:00:00	0.00	default_cover.jpg
51	14	create aaaaa	ffffvvve	100.00	1000.00	2017-06-09 00:00:00	2017-06-09 00:00:00	90.00	default_cover.jpg

We can see the raised money is changed

7. Rate those projects

Rate:Let's take photo

Image File: 

Audio File: 

Video File: 

Rating: 0
1
2
3
4
5

You can give rate which has 6 level. And type some text of rate.

Operation in database:

```
DB::insert('insert into rates (user_id, project_pid, rating, rate_content, created_at, updated_at) values (?, ?, ?, ?, ?, ?)', [$request['user_id'], $request['project_pid'], $request['rating'], $request['rate_content'], Carbon::now(), Carbon::now()]);
```

Resulting table:

user_id	project_pid	rating	rate_content	created_at	updated_at
6	33	3	great	2017-05-01 22:21:04	2017-05-01 22:21:04
6	33	3	great	2017-05-01 22:22:26	2017-05-01 22:22:26
8	34	0	ggg	2017-05-03 19:57:17	2017-05-03 19:57:17
8	34	5	kkk	2017-05-03 19:57:37	2017-05-03 19:57:37
8	38	4	good	2017-05-09 18:34:07	2017-05-09 18:34:07
12	38	3	good	2017-05-05 17:50:36	2017-05-05 17:50:36
14	52	4	good	2017-05-09 18:45:58	2017-05-09 18:45:58

8. search using keywords that are matched against project titles, description and tags

The screenshot shows a search results page for 'Project Square'. At the top, there are navigation links: 'Start a project', 'Project Square', and a user profile 'Puloceba'. Below the header, the title 'Search Result' is displayed. Three search results are shown in cards:

- test tags array**: An image of a child holding a sunflower. Below the image are two lines of text: 'test111' and 'ddddd'.
- test the image**: An image of a woman and a child sharing a milkshake. Below the image are two lines of text: 'test a tags array' and 'test the image'.
- test demo**: A small image of a person's face. To its right is a code snippet:

```
public static <T> void printItemPrice(Bin<T> bin){  
    Iterator<T> iterator = bin.iterator();  
    while(iterator.hasNext())  
        System.out.println("The price of item is" + iterator.next().getPrice());  
}
```

Operation in database:

```
DB::select('select * from projects natural join project_tag join tags on tags.id = project_tag.tag_id  
          where pname like ? or desp like ? or tags.name like ? limit 10', ['%' . $search_content . '%', '%' . $search_content . '%', '%' . $search_content . '%']);
```

9. Follow other user and see their profile

You can see the profile of the project's author. The profile contains some basic information of author and you can also follow or unfollow them. You can also see the author's other projects and the average rating score of their finished projects.

Operation in database:

Follow someone:

We create a new record for the new following relationship in Followers table.

```
DB::insert('insert into followers (user_id, following_id, created_at, updated_at) values (?, ?, ?, ?, ?)',  
          [$follower_id, $following_id, Carbon::now(), Carbon::now()]);
```

Unfollow someone:

```
DB::delete('delete from followers where user_id = ? and following_id = ?', [$follower_id, $following_id]);
```

We delete record for the corresponding following relationship in Followers table.

katelyn1's Profile

	Hometown: beijing
	Date of Birth: 04-30-2017
	Interest music
	Legal Name yiqin
	Project Number 13
	Followers 1

katelyn1's Projects

- [test the search](#) raised money: 0
- [test111](#) raised money: 10100
Average user rating
 2.3 / 5

[Follow](#)

10. Recommendation system

We use the **weighted average content recommendation algorithm** to calculate a recommendation score and give recommendation result: the searching result will add 2 to the recommendation, the followed author's project will add 2 to the recommendation.

Recommended For You



[test tag](#)
[test tag](#)



[test111](#)
ddddd



Let's take photo
This is a photography album and people all like photography.

When you open a new project, this project will add 1. We will always recommend those with high scores.

3.3 Extra Features of System and Design

1. Reply comments

The screenshot shows a comment section with the following entries:

- Puloceba on 2017-05-09 16:11:26: I think it's nice [Reply](#)
- abc on 2017-05-03 19:17:10: @ Puloceba: Yes, I agree [Reply](#)

Below the comments, there is a text input field labeled "Your comment here." and a blue "Add Comment" button.

2. Auto completion function in search part

The screenshot shows a search interface with the following elements:

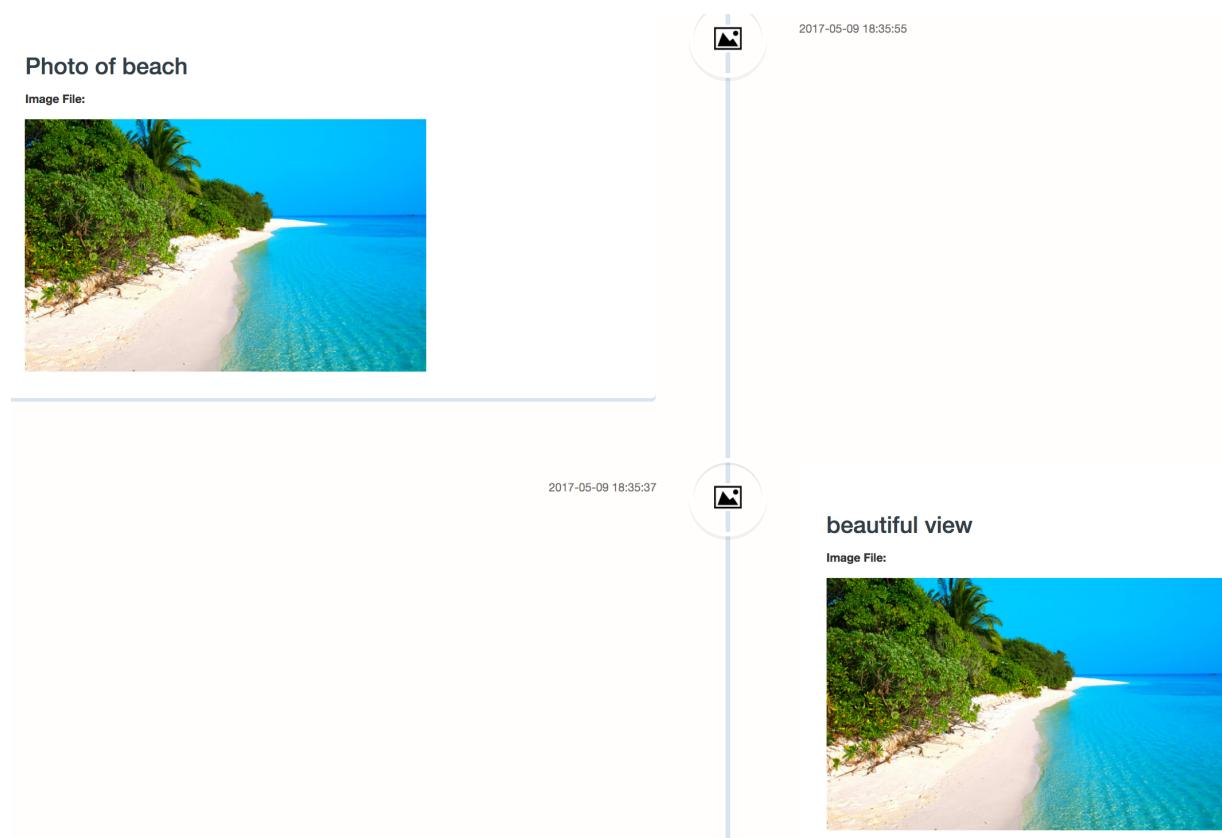
- LetsFund Start a project Project Square
- A search bar containing "dril" with suggestions: "drink" and "drink a coffee".
- A large image of pink flowers.
- Navigation arrows on the left and right sides of the flower image.

3. Averaging score of rating

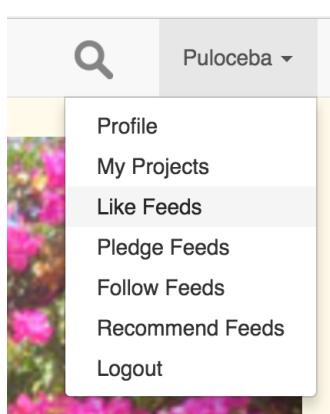
The screenshot shows a rating history section with the following data:

- Average user rating: 3.5 / 5
- abc give rate: on 2017-05-03 19:17:10
good
- <script>alert('piyebuding')</script> give rate: on 2017-05-05 17:36:50
good

4. Timeline decoration of posting



5. The convenience and show real-time information



All feeds are relevant to you

What's popular now



[Let's take photo](#)

This is a photography album and people all like photography.



[db demo](#)

test1111

Know which project is pledged most recently

4. The procedure of a user using our system

1. Register(Every url will direct to this page)

Register

Name	<input type="text"/>
E-Mail Address	<input type="text"/>
Password	<input type="password"/>
Confirm Password	<input type="password"/>

[Register](#)

2. log in

Login

E-Mail Address	<input type="text" value="puloceba@hotmail.com"/>
Password	<input type="password" value="....."/>

Remember Me

[Login](#) [Forgot Your Password?](#)

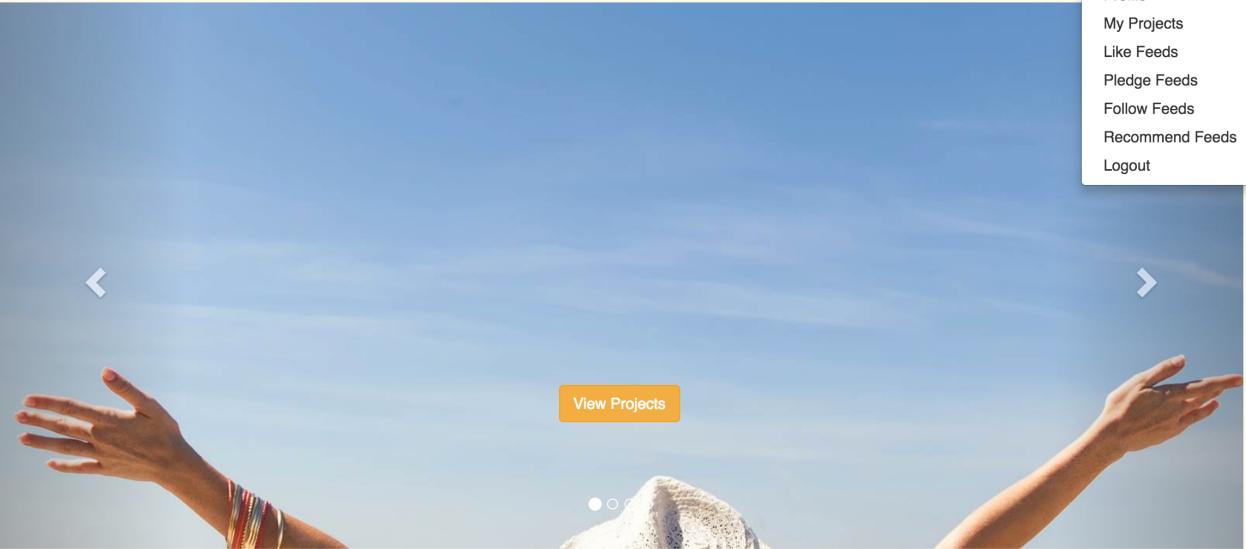
3. Home page customized for you (/home)

You can see projects which are created by who you follow. The projects recommended to you and popular projects now. Then, go to the profile to fill it up.

LetsFund Start a project Project Square

[Puloceba ▾](#)

- [Profile](#)
- [My Projects](#)
- [Like Feeds](#)
- [Pledge Feeds](#)
- [Follow Feeds](#)
- [Recommend Feeds](#)
- [Logout](#)

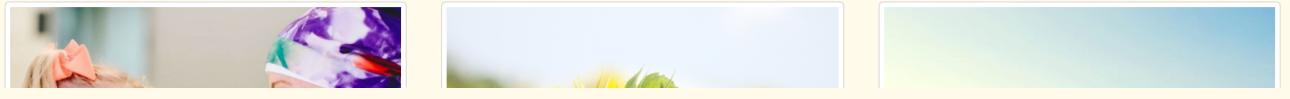


[View Projects](#)













4. Fill you profile (/profile), then go to the top bar

LetsFund Start a project Project Square Puloceba

Edit Profile

Upload Avatars
 No file chosen

Hometown:

Date of Birth:

Interest:

Credit Card:

Legal Name:

5. Start a new project(/projects/create)

LetsFund Start a project Project Square Puloceba

Post a new project

Project Name:

Upload a Project Cover
 No file chosen

Description:

Upload files in Description
Upload image file
 No file chosen

6. edit, like, pledge, comment a project (/projects/{pid}) 7. Add postings by click add posting

End Time: 11 months from now Created By:Puloceba

Let's take photo Edit

Tags: photos
 1

Raised Money: 0
Minimum Money: 100
Maximum Money: 1000

Description Updates Comments Pledges History

This is a photography album and people all like photography.

LetsFund Start a project Project Square Puloceba

Post new files for project: Let's take photo

Description:

Upload image
 No file chosen

Upload audio
 No file chosen

Upload video
 No file chosen

8. Can announce finished after get enough money

End Time: 11 months from now Created By:Puloceba

Let's take photo Edit Announce Finished

Tags: photos
 1

Raised Money: 100
Minimum Money: 100
Maximum Money: 1000

9. In profile, rate the finished projects

katelyn

Hometown:

Date of Birth:

Interest

Credit Card

Legal Name

Pledge History

Pledged **project music** amount: 100

Pledged **cathy's project** amount: 10 Rate

Pledged **cathy's project** amount: 4 Rate

10. You will go to the project detail page and rating history will be shown.

Average user rating
2.3 / 5

katelyn1 give rate: ★★★★☆ on 2017-04-29 15:05:15
great

katelyn1 give rate: ★★★★★ on 2017-04-29 15:05:15

katelyn1 give rate: ★★★★☆ on 2017-04-29 15:05:15

11. Search the projects by keywords

LetsFund [Start a project](#) Project Square

dril

drink

drink a coffee

5. Conclusion and Acknowledgements

In this project, we use Laravel 5.4 framework to write our back-end code, mysql as our database, Apache as server and jquery, javascript and bootstrap to do the front-end work. Our project could guard against SQL injection and cross-site scripting attacks. And we also support keeping state for a user session.

The authors would like to thank Professor Torsten Suel and TAs for their time and energy devoted in course CS6083 and their generous help and guidance in helping us finishing this project.