CS2102 Project

Group 12: Topic B Crowdfunding

Group Members:

Chia Wen Kai A0140215Y

Daniel Koh Chong Xiang A0140195L

Zhang Jiyi A0130620B

Sun Bangjie A107436J

Introduction

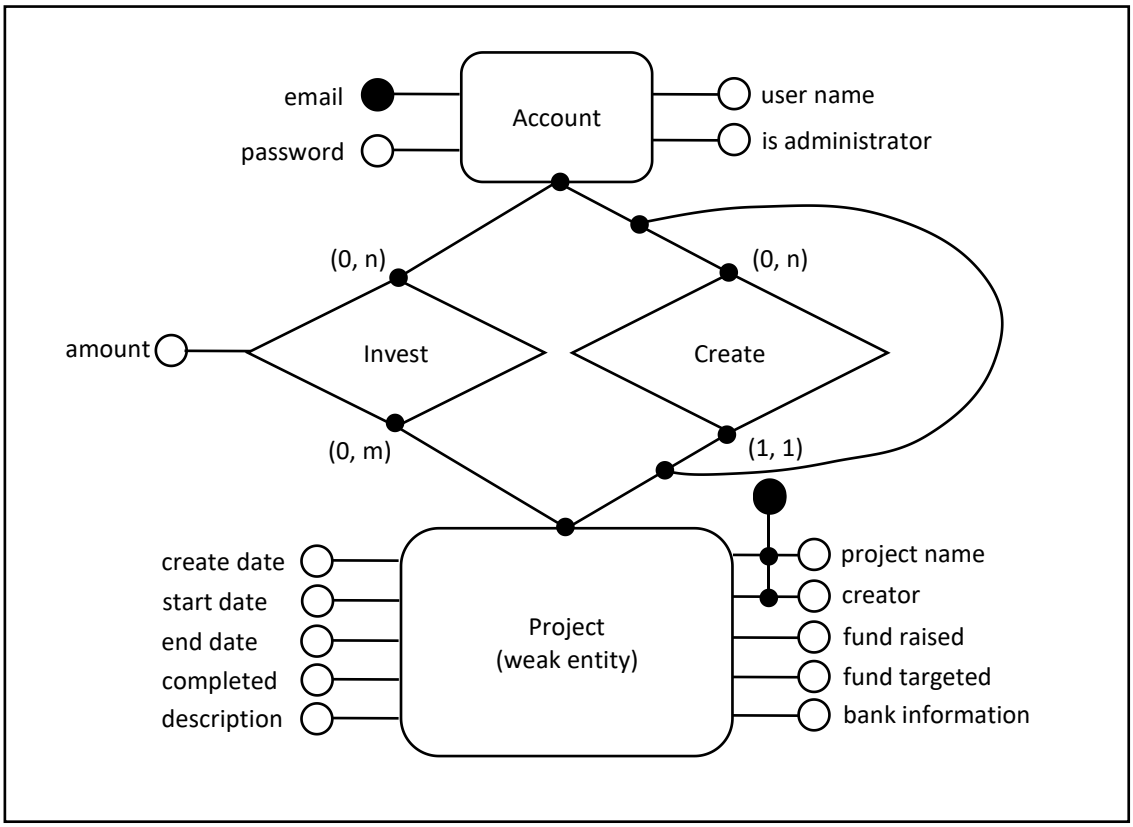
In our project, we used Bitnami Stack and pgAdmin to manage our web server and database management system. To implement the web page, we used HTML and PHP for page content and CSS for page design. The database management system used is PostgreSQL.

The aim of our project is to build a catalogue of projects looking for crowdfunding where entrepreneurs can advertise their projects and investors can invest in projects they are interested in.

Users of our web site can play both roles of entrepreneurs and investors. Each user has an account with email address and password. Users can browse all information of the projects and fund them with any amount of money. They can also create projects which they want to advertise.

Our database system keeps track and updates all information of projects. Only administrators can modify information and delete projects and accounts.

ER diagram



Explanation:

Account entity is used to store information users of two types: normal users and administrators. There is "is administrator" attribute to indicate the type of the account. When users sign up, they cannot use duplicate email addresses. Hence, we use email as

the primary key of account entity. User name can be same as long as email addresses are different. Password protects the account.

Project entity is a weak entity that cannot be uniquely identified by its own attributes alone since it must be created by one and only one account of either normal user or administrator. Hence, creator of the project must be referenced to email (primary key) in the account entity. Creator, together with the project name, can uniquely identify a project. Other information about a project includes dates that project is created, starts and ends, description of the project, target fund, fund raised so far and whether the project has completed its fund raising. Bank information is also obtained in order for investors to transfer funds.

There are two relationship sets: create and invest. Create is the relationship between Account and Project such that an account can create no project, one project or many projects, and a project can only be created by one and only one account. Such relationship is indicated by the cardinality of (0, n) and (1, 1) in the ER diagram respectively. On the other hand, Invest is another relationship between Account and Project. One account can invest in zero, one or many projects, and one project can be invested by zero, one or many accounts as well. The cardinality, thus, can be (0, n) and (0, m) respectively. Moreover, Invest relationship set has one attribute of “amount” that indicates the amount of money that any account invests in any project.

Based on the above ER diagram, we have created the relational schema for our database. Constraints will be explained in the next section.

Relational Schema

```
--create account table--
CREATE TABLE account(
account_name VARCHAR(64) NOT NULL,
account_email VARCHAR(64) PRIMARY KEY,
account_password VARCHAR(64) DEFAULT 'Group12',
is_admin char(1) NOT NULL CHECK (is_admin = 'T' OR is_admin = 'F'));

--create project table--
CREATE TABLE project(
project_name VARCHAR(256) NOT NULL,
creator VARCHAR(64) REFERENCES account(account_email),
raised Numeric DEFAULT 0,
target Numeric NOT NULL,
created date NOT NULL,
project_start date NOT NULL,
project_end date NOT NULL CHECK (project_end >= project_start) ,
completed boolean DEFAULT false,
description VARCHAR(256) DEFAULT 'Please donate to our project. Thank you!',
bankinfo VARCHAR(256) NOT NULL,
picture_url VARCHAR(256),
PRIMARY KEY (project_name, creator));
```

--create invest table--
CREATE TABLE invest(
account_email VARCHAR(64) REFERENCES account(account_email),
project_name VARCHAR(256),
creator VARCHAR(64),
amount Numeric CHECK (amount > 0),
FOREIGN KEY(project_name, creator) REFERENCES project(project_name, creator) ON UPDATE CASCADE);

Explanation:
As shown in the ER diagram, account_email is the primary key of the account table. account_name and account_password should not be NULL as these are the necessary information to verify a user in order to login in. is_admin is either 'T' or 'F', constrained by the CHECK clause. This indicates the type of the account and different web pages with different features will be shown accordingly.

In project table, project_end should not be smaller than project_start and it is constrained by the CHECK clause as well. We added a new field to store the URL of picture of the project to allow better display. We did not create a table for Create relationship set because we include the necessary information, more specifically creator of the project, in the table of project. This guarantees that one project can only have one creator. This can be seen in the FOREIGN KEY constraints on the creator field. The composite primary key of project table allows projects created by different users to have same names just in case they are doing similar projects.

Lastly, invest table is created to keep track the history of funding. Since invest table is the relationship set in the ER diagram, account_email must reference to that in account table, and project_name and creator must reference to those in project table. The only additional field is the amount of fund raised which should be larger than 0, constrained by the CHECK clause.

Next section includes the sample data that we inserted into our database system following the format:
--create account--
INSERT INTO account VALUES(account_name, account_email, password, is_admin);
--create project--
INSERT INTO project VALUES(project_name, creator, raised, target, created, project_start, project_end, completed, description, bankinfo, picture_url);
--create account--
INSERT INTO invest VALUES(account_email, project_name, creator, amount);

Sample Data

--create account-
INSERT INTO account VALUES('Adam Lee','adam1993@gmail.com','hunter2', 'F');
INSERT INTO account VALUES('Zhang Ji Yi', 'jiyi@gmail.com','jiyi123', 'T');
INSERT INTO account VALUES('Zack Tan', 'zackyt@gmail.com','abc123', 'F');

```
INSERT INTO account VALUES('Daniel Koh', 'daniel@gmail.com','daniel123', 'T');
INSERT INTO account VALUES('Xiao Ming', 'goodmorning@gmail.com','ggwp', 'F');
INSERT INTO account VALUES('Sun Bang Jie', 'bangjie@gmail.com','bangjie123', 'T');
INSERT INTO account VALUES('Chia Wen Kai', 'wenkai@gmail.com','wenkai123', 'T');
INSERT INTO account VALUES('Bruce Li', 'bruceli@hotmail.com','123321', 'F');
INSERT INTO account VALUES('Da Ming', 'edgydude1888@gmail,com','daming123', 'F');
INSERT INTO account VALUES('Jacky Mao', 'oldguy88@gmail.com','ee123', 'F');
INSERT INTO account VALUES('Shi Ming Yi', 'rencicharityfund@gmail.com','notascammer', 'F');
INSERT INTO account VALUES('Lee Kong Hee', 'leekonghee@gmail.com','abba123', 'F');
INSERT INTO account VALUES('Nick Tan', 'nicktan111@hotmail.com','flywin', 'F');
INSERT INTO account VALUES('Susan Tan', 'susantan99@hotmail.com','tan123', 'F');
INSERT INTO account VALUES('Larry Su', 'wewlad@gmail.com','iamlarry', 'F');
INSERT INTO account VALUES('Bao Zhong', 'guarantee@hotmail.com','winforsure', 'F');
INSERT INTO account VALUES('Sam Tan', 'heyimsam@gmail.com','hospital', 'F');
INSERT INTO account VALUES('Han Fei', 'ma3111@hotmail.com','gewozhan', 'F');
INSERT INTO account VALUES('Wang Fei', 'wangwang@hotmail.com','zhan', 'F');
```

--create projects--

```
INSERT INTO project VALUES('School
fees','adam1993@gmail.com',0,1000,'2017-03-10','2017-04-10','2017-10-10','false','Raising
money for my school fees','111-33333-444','http://www.mis.ac.th/fees.jpg');

INSERT INTO project VALUES('Making Potato Salad',
'bruceli@hotmail.com',0,15,'2017-03-10','2017-04-10','2017-04-10','false','Need money for
lunch','012345678','http://food.fnr.sndimg.com/content/dam/images/food/fullset/2006/9/22/0/ig07
07_potato_salad1.jpg.rend.hgtvcom.616.462.suffix/1428086524917.jpeg');

INSERT INTO project VALUES('Food for the needy',
'edgydude1888@gmail,com',0,215,'2017-07-10','2017-08-10','2017-12-27','false','Money raised
will go into food supplies for the
needy','123459784','http://challengeforsustainability.org/wp-content/uploads/2014/07/food-donati
on.jpg');

INSERT INTO project VALUES('Retirement fund',
'oldguy88@gmail.com',0,500,'2017-03-11','2017-04-11','2018-12-22','false','CPF not
enough.','5559999955','http://www.retirementplanningconnecticut.com/wp-content/uploads/2016/
12/Retirement.jpg');

INSERT INTO project VALUES('Charity Performance',
'rencicharityfund@gmail.com',0,10000,'2017-03-12','2017-04-12','2017-10-10','false','We are a
charity organisation looking for sponsors for our annual
performance','987654321','http://www.tunbridgewellskidsonthego.co.uk/wp-content/uploads/2014
/07/Flyer-for-Charity-Concert-2014-e1405025428575.jpg');

INSERT INTO project VALUES('Funds for hospital fees',
'nicktan111@hotmail.com',0,2030,'2017-07-11','2017-08-11','2017-12-20','false','Father in hospital,
no money for
fees','123789456','https://www.ucsf.edu/sites/default/files/styles/300w/public/fields/field_insert_file/
news/calculator-stethescope-hospital-bill-costs_0.jpg');
```

```
INSERT INTO project VALUES('New web series',
'susantan99@hotmail.com',0,2050,'2017-06-11','2017-06-11','2018-01-31','false','Starting a new
web series. Need funds.',
'885296374','http://www.indiantelevision.com/sites/drupal7.indiantelevision.co.in/files/images/intern
et-images/2016/04/25/Web-Series-Scripted.png');

INSERT INTO project VALUES('Trip to Europe',
'wewlad@gmail.com',0,1500,'2017-11-15','2018-11-20','2019-11-25','false','Looking to fund my
trip to
Europe.','963852741','https://shatorch.com/wp-content/uploads/2016/05/eurpoa-travel.jpg');

INSERT INTO project VALUES('BBQ party',
'heyimsam@gmail.com',0,199,'2017-10-25','2018-04-11','2018-11-11','false','Hosting a party, all
funders
welcomed!','963741852','https://shatorch.com/wp-content/uploads/2016/05/eurpoa-travel.jpg');
```

--create investments--

```
INSERT INTO invest VALUES('heyimsam@gmail.com', 'Retirement fund', 'oldguy88@gmail.com',
30);
UPDATE project SET raised = (SELECT SUM(amount) FROM invest WHERE creator
='oldguy88@gmail.com' AND project_name= 'Retirement fund' ) WHERE creator
='oldguy88@gmail.com' AND project_name= 'Retirement fund' ;

INSERT INTO invest VALUES('susantan99@hotmail.com', 'Charity Performance',
'rencicharityfund@gmail.com', 50);
UPDATE project SET raised = ( SELECT SUM(amount) FROM invest WHERE creator
='rencicharityfund@gmail.com' AND project_name= 'Charity Performance') WHERE creator
='rencicharityfund@gmail.com' AND project_name= 'Charity Performance';

INSERT INTO invest VALUES('goodmorning@gmail.com', 'Making Potato Salad',
'bruceli@hotmail.com', 5);
UPDATE project SET raised = ( SELECT SUM(amount) FROM invest WHERE creator
='bruceli@hotmail.com' AND project_name= 'Making Potato Salad' ) WHERE creator
='bruceli@hotmail.com' AND project_name= 'Making Potato Salad';

INSERT INTO invest VALUES('leekonghee@gmail.com', 'Retirement fund', 'oldguy88@gmail.com',
45);
UPDATE project SET raised =( SELECT SUM(amount) FROM invest WHERE creator
='oldguy88@gmail.com' AND project_name= 'Retirement fund') WHERE creator
='oldguy88@gmail.com' AND project_name= 'Retirement fund';

INSERT INTO invest VALUES('wangwang@hotmail.com', 'BBQ party', 'heyimsam@gmail.com', 60);
UPDATE project SET raised = ( SELECT SUM(amount) FROM invest WHERE creator
='heyimsam@gmail.com' AND project_name= 'BBQ party' ) WHERE creator
='heyimsam@gmail.com' AND project_name= 'BBQ party';
```

Explanation:

For any investment to a project, once we insert an entry into invest table, we also need to update the “raised” field in the project table of that particular project.

Other commands
that helps in our implementation of web page

--show all projects--

SELECT * FROM project;

Explanation:
The command to show all projects is used in the main page to display a list of projects. However, in each page we display 10 projects and users can click on "previous page" and "next page" as the number of projects is likely to be large.

--delete project-

projectUserEmail, projectName: read from front end
DELETE FROM invest WHERE creator=projectUserEmail AND project_name=projectName;
DELETE FROM project WHERE creator = projectUserEmail AND project_name = projectName;

--delete user-

myEmail: read from front end
DELETE FROM invest WHERE creator=myEmail;
DELETE FROM project WHERE creator=myEmail;
DELETE FROM account WHERE account_email = myEmail;

Explanation:
Administrators can delete projects and users using the commands shown above. We need to delete the entries in the invest table first in order to delete a project, or otherwise foreign key constraints of invest table will be violated. Similarly, entries in invest and project tables have to be deleted in order to delete an account, or otherwise foreign key constraints of the two tables will be violated.

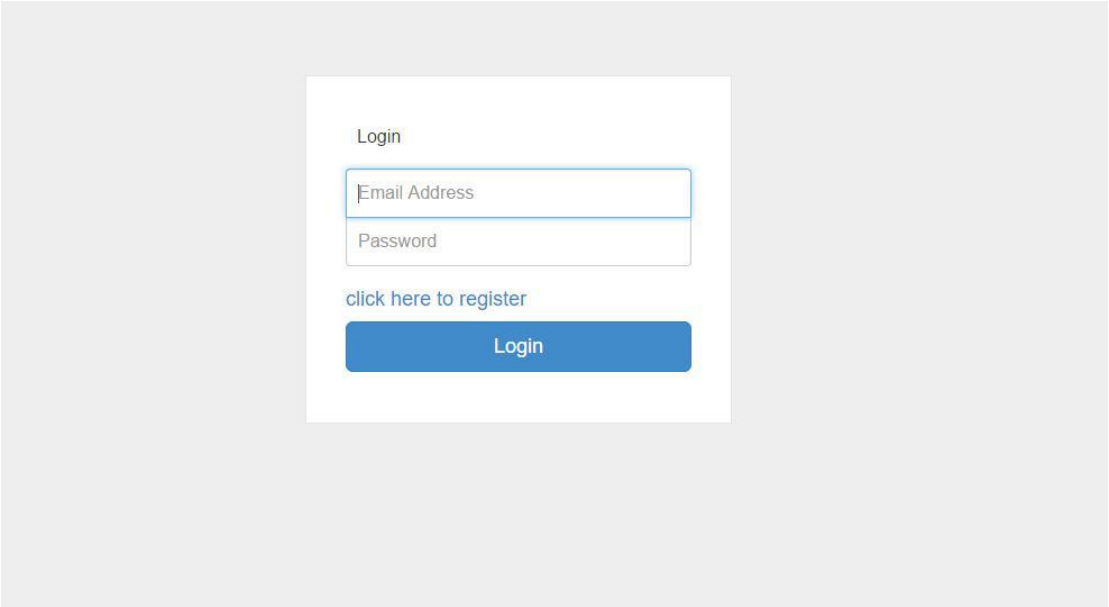
--update projects--

updatedField, updatedTable, value: read from front end
UPDATE updatedTable SET updatedField = value;

Explanation:
Lastly, both normal users and administrators can update projects using the above command.

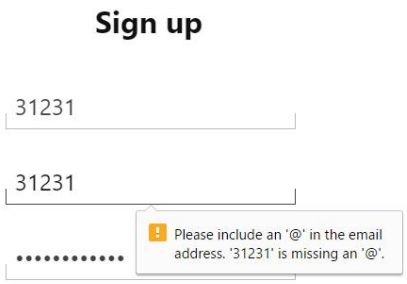
Screenshots
of web page interface and features

1. Login page



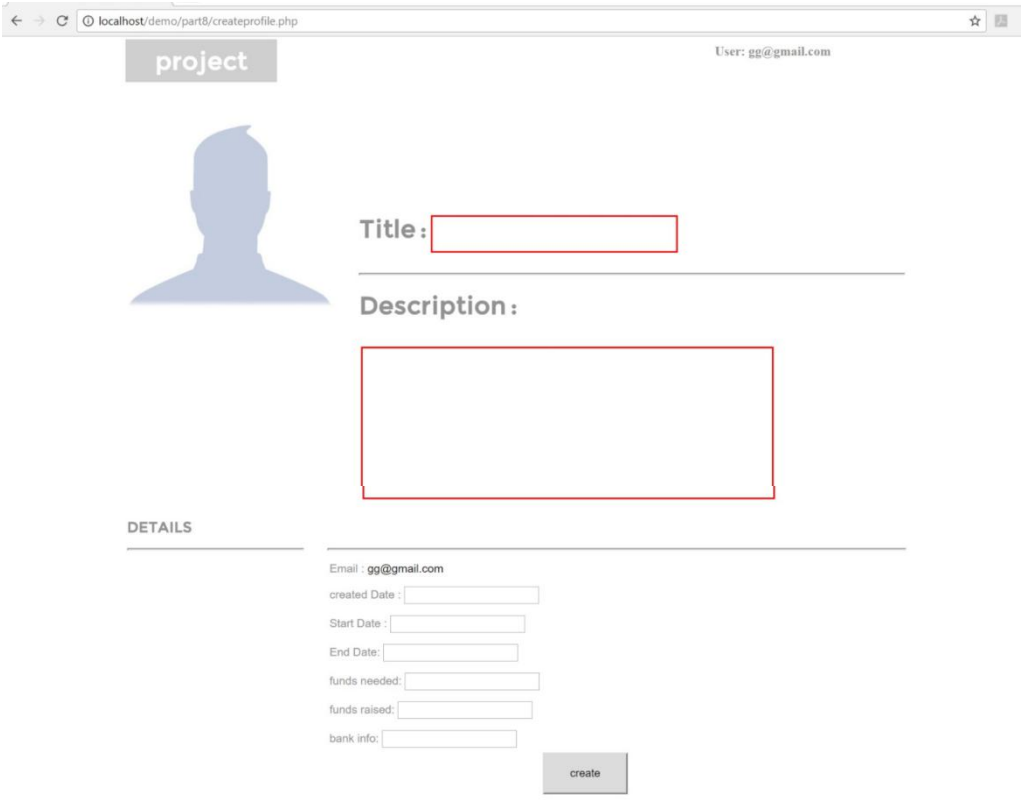
This login page requires users to enter email address and password in order to access the main page of our website. New user is able to create new account by clicking “click here to register” button which will be directed to a sign up page.

2. Sign up page



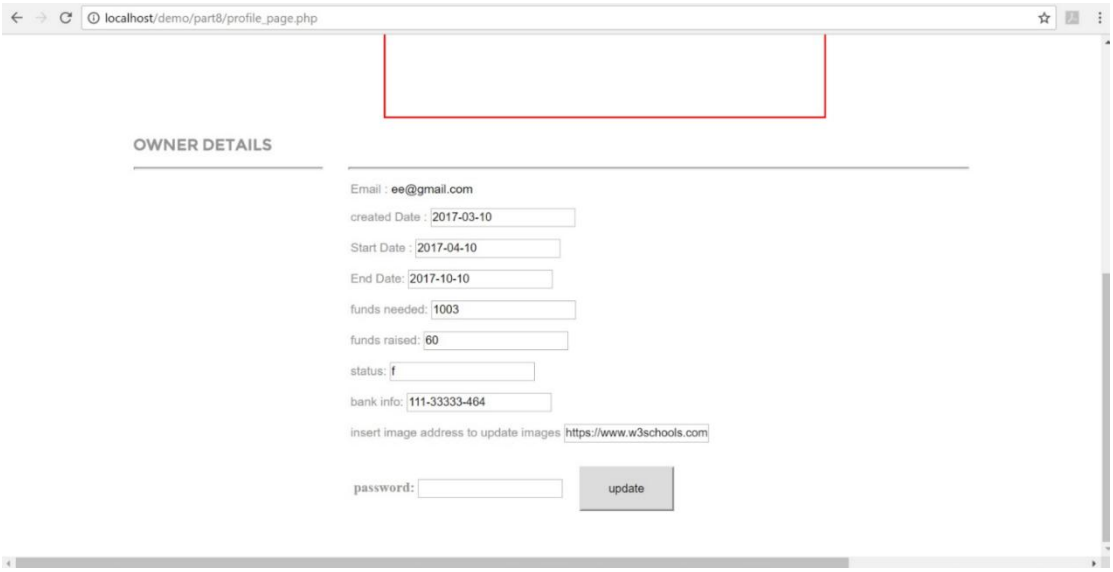
This sign up page requires users to fill up full name, email address and password with certain requirements. Email address must follow the format of XX@XX.com. Password must be at least 8 characters with at least one number, one alphabet and one alphabet in upper case. Invalid entries will not pass this page and warnings will be shown.

3. Page for creating a project



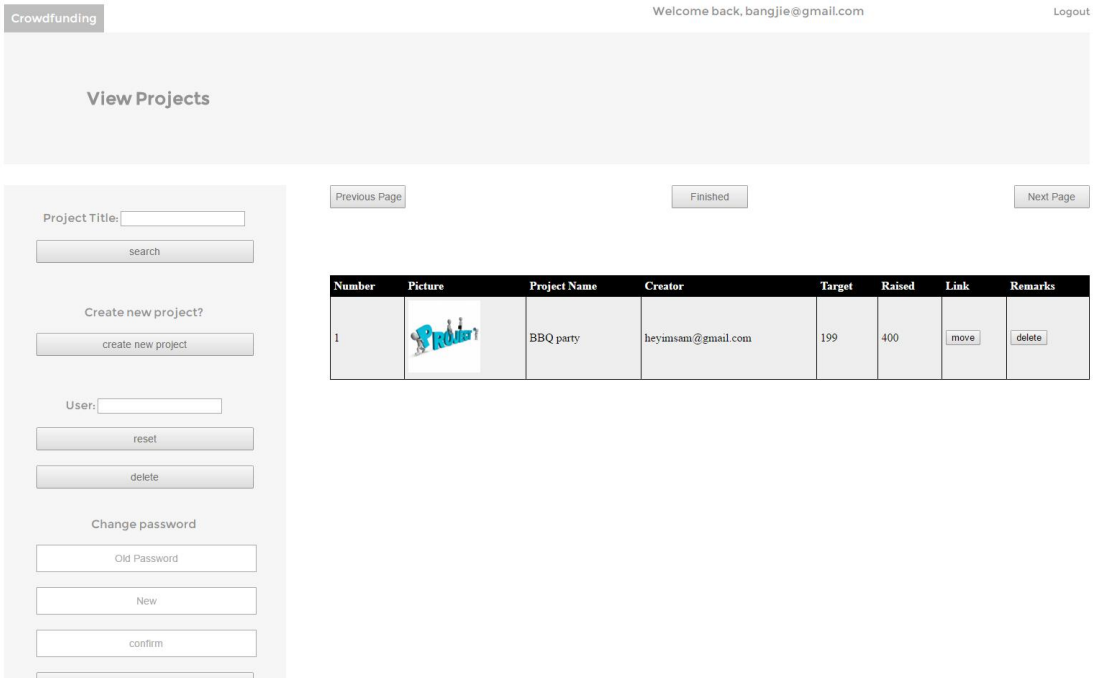
This is the page for creating a project by any user. Users can fill up the form and click "create" to submit information to our database.

Update information of a project

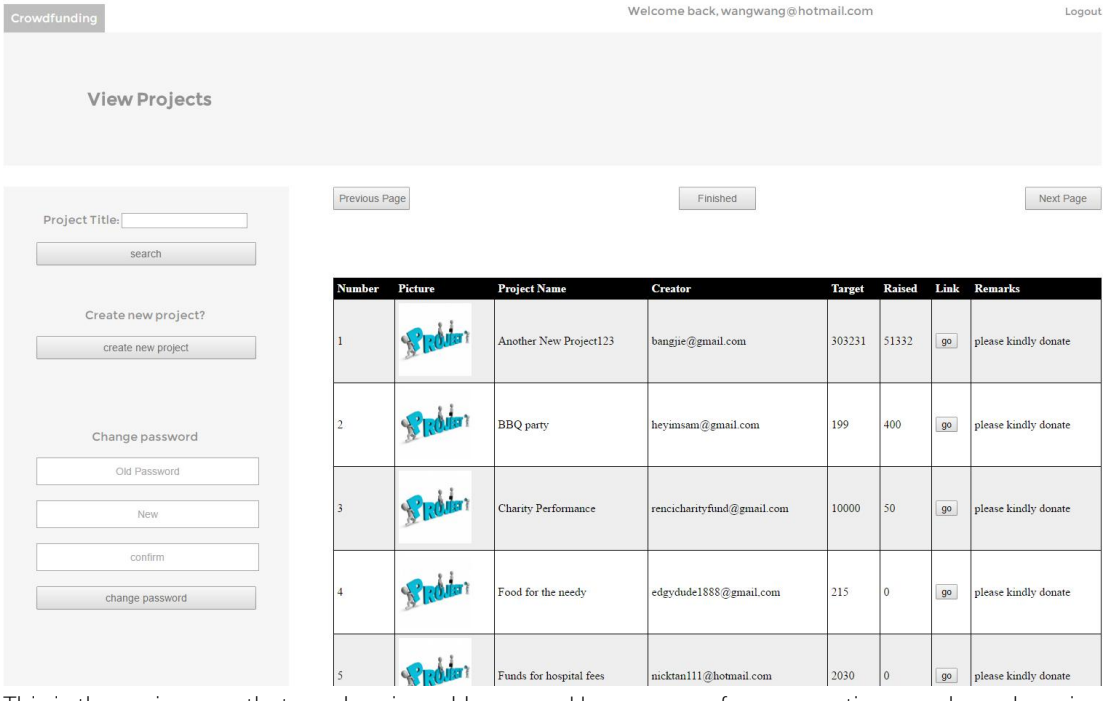


All users can update projects. However, users with non-administrator accounts can only update their own projects while administrators can update projects from all users. Password is required for non-administrators in order to validate and re-confirm the update.

4. Main page to display projects (by administrator)

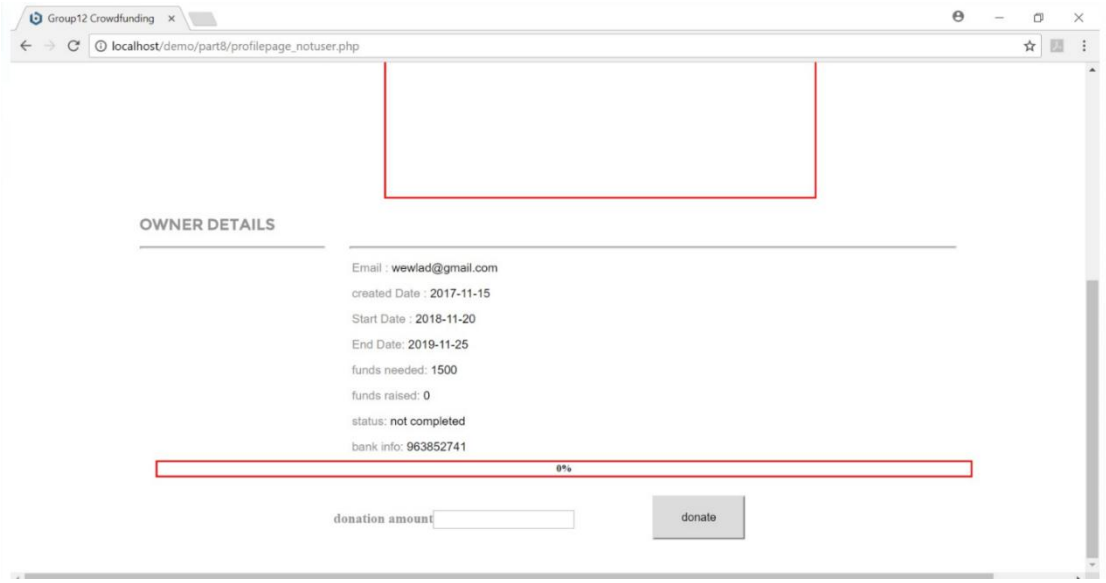


Main page to display projects (by non-administrator)

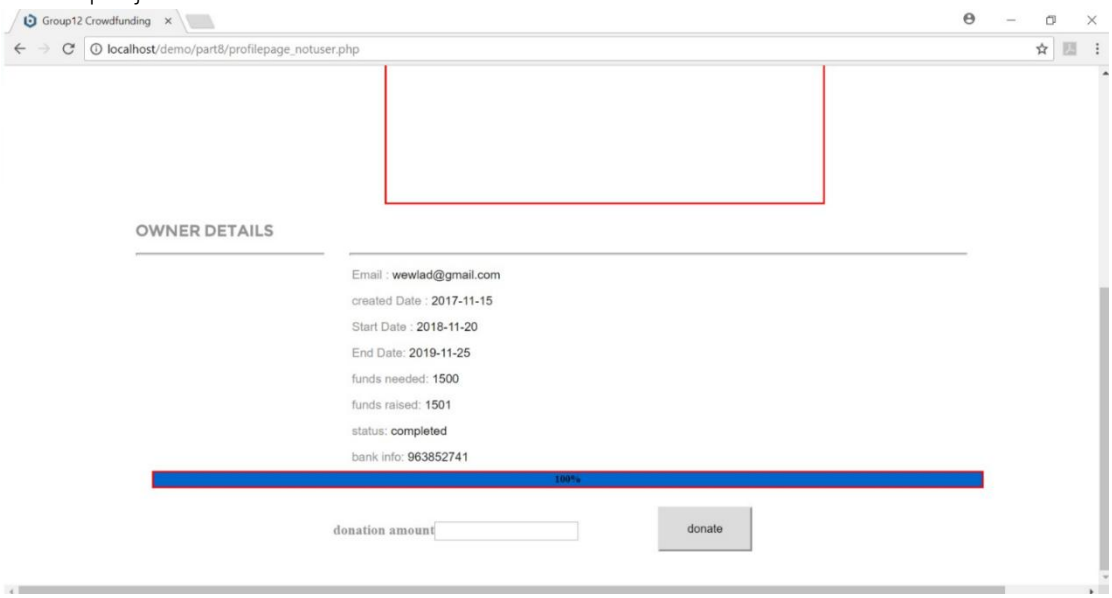


This is the main page that can be viewed by users. Users can perform operations, such as changing password and creating new projects, at the sidebar. However, the “delete” button to delete user and projects can only appear if the account is administrator. Administrators are also able to reset and change password of any user account of non-administrator. In the main page area on the right, users can view other users’ projects using “go” button, and they can go to their own projects and update them using “move” button. Administrators can access any user project using the “move” button, they cannot donate. If they wish to donate, they can do so by creating a user account. Lastly, “Finished” button is for displaying all the projects that have already completed funding.
After clicking the “go” button, the profile page of the project will be shown (see next page).

5. Profile page of a project



When project is funded:



This is the profile page of a project with a progress bar showing the current funding progress. Once the progress bar hits 100%, the project is funded. The status of the project will be changed to "completed" from "not completed". Users can still view projects that have already completed the targets. Administrators will delete funded projects on the monthly basis. Users can enter in the amount field at "donation amount" and click "donate" to invest in the project.