Database Project

Xinmei Jin (xj573)

Yi Li (yl4807)

1. Introduction about the whole design

Our goal is to design a database that is a kind of social networking site for students looking for jobs, as well as those companies looking for employees.

In the first part of this project, we aim to design the backend part, with the following schema, but our database may well change with our designing frontend process.

In the second part of this project, we aim to create the web-based user interface for the database designed in the first part of the project.

For frontend part, since we have two classes of registered users: Students and Company. So before log into this system, we firstly need to identify this user, the beginning page of this system is a login page, before log in or sign up, the user should choose 'Student' or 'Company', then according to user's selection we can proceed:

a) If the user is registered, check the corresponding Record table to judge if it's correct, in other words, only if both the account and password are correct and stored in the database will this user be able to see the home page of this system.

b) If the user is a new one, then there is a sign up button to lead this user to a specific sign up page, in the sign up page, firstly user has to choose to be a student or company, then user only needs create a new account, particularly, the new created account cannot exist in the Record table before, as well as a password. After the user click 'finish sign up' button, this account and its password should be inserted into the corresponding Record table and lead this user back to the log in page.

After the user log into this system, user will see the home page of this system. I plan to fill this home page with all kinds of advertisements and selections, through this home page, users are able to go to specific pages such as: My Account, Notifications, Messages, Friends, Search Line, My Applications and etc.

a) My Account contains my profile, setting and etc for students, in company view, My Account will contains company description, introduction, my followings, something like that.

b) Notifications contain those announcements from my following companies and some recommended announcements in which the user meets its requirements. For Company, the notification part will show all the announcements they have posted before. Click each announcement will lead to all the application towards this announcement, through these applications the company can see each applicant's profile and resume, so as to decide accept this application or not.

c) Messages and Friends button only for students because the companies don't need that. Friend requests will show at the Friends part, user can decide to accept this request or not.

d) Search button will allow users to input some keyword to search in the whole database, including student name, company name, job title, job location and etc.

e) My Applications for both company and students, user will see the current applications they are working at, and check the status.

Of course, the user can click back to the home page whenever they want and log out of this system.

2. **To turn this ER model into relational model, we get the following tables:**

Students(sid, sname, sGPA, slevel, suniversity, smajor, sresume, sdescription)
slevel indicates: Bachelor, Master or PhD

sdescription contains keyword of interests and qualifications
sresume and sdescription are used as TEXT type
primary key: sid

Company(cid, cname, caddress, cindustry, cemail, cdescription)
cindustry just briefly indicate what kind of type this company is
cdescription contains basic introduction of this company
primary key: cid

Friends(sid, fid, fname)
fid is also student ID, fname is student name.
Only mutual friend relationship will be recorded in this Friends table.
primary key: sid, fid

Friendrequest(fsenderid, freceiverid, request, requestdate, requeststatus)
fsenderid and freceiverid are both student ID, request is just a text that the sender wants to say, requeststatus can be accepted, declined or unanswered
primary key: fsenderid, freceiverid, requestdate
foreign keys: fsenderid, freceiverid

Announcements(aid, cid, cname, jobtitle, joblocation, jobsalary, jobmajor, joblevel, jobdescription, annoucedate)
jobtitle is keyword of this job, jobmajor corresponds to students' major, joblevel corresponds to student level
primary key: aid
foreigh keys: cid

Following(sid, cid, cname, followingdate)
primary key: sid, cid, followingdate
foreign keys: sid, cid

Application(applyid, sid, aid, applystatus, applydate)
applyid indicates one application process, applystatus can be accepted, declined or unanswered
primary key: applyid
foreign keys: sid, aid

Message(msenderid, mreceiverid, aid, text, messagedate)
msenderid can be student ID, mreceiverid is student ID, text can

be a brief sentence of this message
primary key: msenderid, mreceiverid, messagedate
foreign keys: aid

Notification(sid, aid, cname, announcedate)
primary key: sid, aid, announcedate
foreign keys: sid, aid

3. **I add some sample data in MySQL to test this schema, screenshots below show my testing data:**

Students

Result Grid | Filter Rows: Search | Edit: | Export/Import:

sid	gender	sname	sGPA	slevel	suniversity	smajor	sresume	sdescription
111111	female	Taylor	3.1	Master	New York University	Computer Engineering	NULL	NULL
112337	female	Anna	4	PhD	New York University	Financial Engineering	Financial Analysis, Database Systems	NULL
345569	male	Kalvin	3.7	PhD	New York University	Computer Science	Database System	NULL
443270	female	Carly	3.6	Master	Columbia University	Applied Mathematics	NULL	NULL
563124	female	Sunny	3.7	Master	Columbia University	Philosophy	NULL	NULL
660871	male	John	2.9	Master	Columbia University	Drama and Theatre Arts	shakespeare	NULL
752234	male	Henry	3.6	Bachelor	Duke University	Jazz Studies	NULL	NULL
754082	male	Partrik	3.3	PhD	Harvard University	Architecture	NULL	NULL
990864	female	Kate	3.1	Bachelor	Duke University	Chemistry	NULL	NULL

Company

Result Grid | Filter Rows: Search | Edit: | Export/Import:

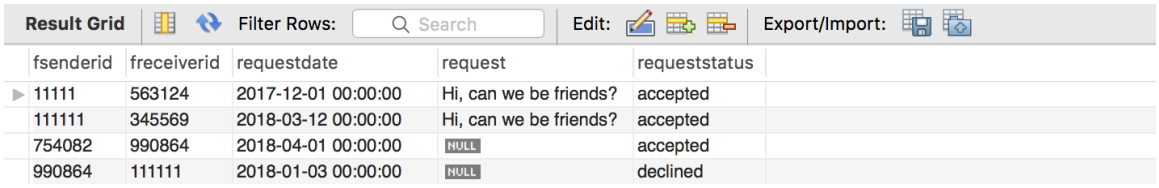
cid	cname	caddress	cindustry	cemail	cdescription
1111	ABB	New York, NY	Computer	1111@abb.cn	NULL
1234	PPY	Brooklyn, NY	Computer	1234@ppy.cn	NULL
2222	BCC	Jersey City, NJ	Internet	2222@bcc.cn	NULL
3333	CDD	Brooklyn, NY	Internet	3333@cdd.cn	NULL
3945	Facebook	New York, NY	Social Network	3945@fabk.cn	NULL
4444	DEE	Princeton, NJ	Network Security	4444@dee.cn	NULL
4568	Microsoft	New York, NY	Internet	4568@micro.cn	NULL
5555	EFF	Newark, NJ	Insurance	5555@eff.cn	NULL
6666	FGG	Queens, NY	Sports	6666@fgg.cn	NULL
7777	GHH	Flashing, NY	Architecture	7777@ghh.cn	NULL
8888	HII	New York, NY	Finance	8888@hii.cn	NULL
9999	IJJ	Jersey City, NJ	Electronic Devices	9999@ijj.cn	NULL

Friends

Result Grid | Filter Rows: Search | Edit: | Export/Import:

sid	fid	fname
111111	345569	Kalvin
111111	563124	Sunny
754082	990864	Kate

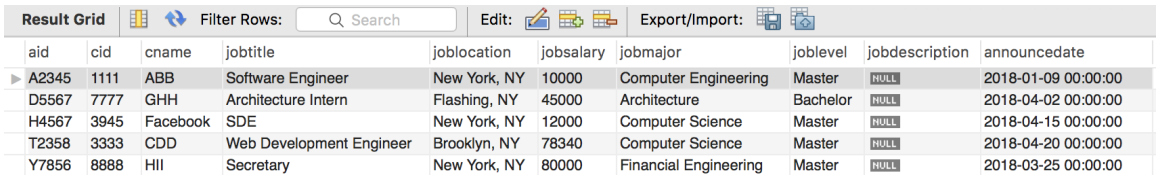
Friendrequest



Result Grid | Filter Rows: Search | Edit: | Export/Import:

fsenderid	freceiverid	requestdate	request	requeststatus
11111	563124	2017-12-01 00:00:00	Hi, can we be friends?	accepted
111111	345569	2018-03-12 00:00:00	Hi, can we be friends?	accepted
754082	990864	2018-04-01 00:00:00	NULL	accepted
990864	111111	2018-01-03 00:00:00	NULL	declined

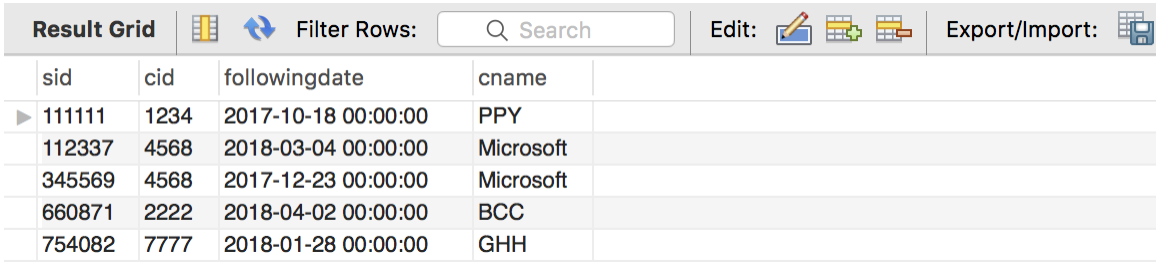
Announcements



Result Grid | Filter Rows: Search | Edit: | Export/Import:

aid	cid	cname	jobtitle	joblocation	jobsalary	jobmajor	joblevel	jobdescription	announcedate
A2345	1111	ABB	Software Engineer	New York, NY	10000	Computer Engineering	Master	NULL	2018-01-09 00:00:00
D5567	7777	GHH	Architecture Intern	Flashing, NY	45000	Architecture	Bachelor	NULL	2018-04-02 00:00:00
H4567	3945	Facebook	SDE	New York, NY	12000	Computer Science	Master	NULL	2018-04-15 00:00:00
T2358	3333	CDD	Web Development Engineer	Brooklyn, NY	78340	Computer Science	Master	NULL	2018-04-20 00:00:00
Y7856	8888	HII	Secretary	New York, NY	80000	Financial Engineering	Master	NULL	2018-03-25 00:00:00

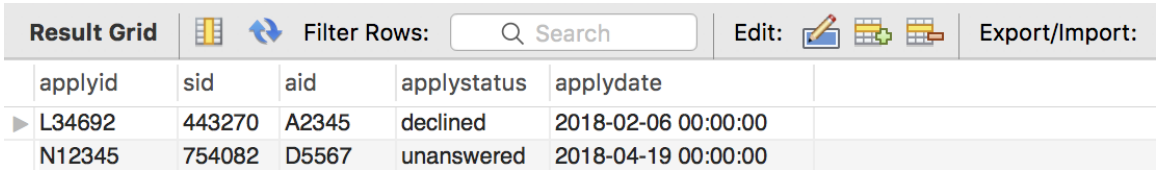
Following



Result Grid | Filter Rows: Search | Edit: | Export/Import:

sid	cid	followingdate	cname
111111	1234	2017-10-18 00:00:00	PPY
112337	4568	2018-03-04 00:00:00	Microsoft
345569	4568	2017-12-23 00:00:00	Microsoft
660871	2222	2018-04-02 00:00:00	BCC
754082	7777	2018-01-28 00:00:00	GHH

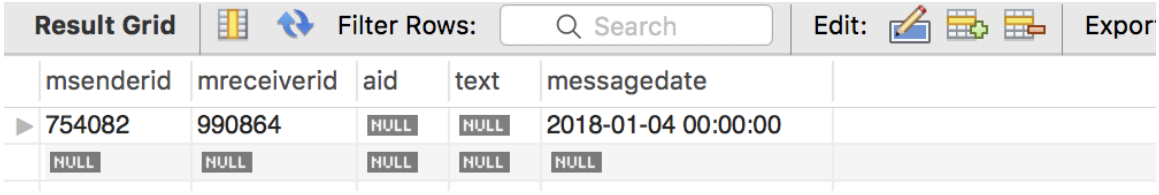
Application



Result Grid | Filter Rows: Search | Edit: | Export/Import:

applyid	sid	aid	applystatus	applydate
L34692	443270	A2345	declined	2018-02-06 00:00:00
N12345	754082	D5567	unanswered	2018-04-19 00:00:00

Message



Result Grid | Filter Rows: Search | Edit: | Expor

msenderid	mreceiverid	aid	text	messagedate
754082	990864	NULL	NULL	2018-01-04 00:00:00
NULL	NULL	NULL	NULL	NULL

Notification

Result Grid | Filter Rows: Search | Edit: | Export/Import:

sid	aid	announcedate	cname
111111	D5567	2018-04-02 00:00:00	NULL
111111	H4567	2018-04-15 00:00:00	NULL
111111	Y7856	2018-03-25 00:00:00	NULL
112337	H4567	2018-04-15 00:00:00	Facebook
345569	H4567	2018-04-15 00:00:00	Facebook

4. Write down those queries to test this schema, screenshots as following:

1) Create a table to record the login account name and password

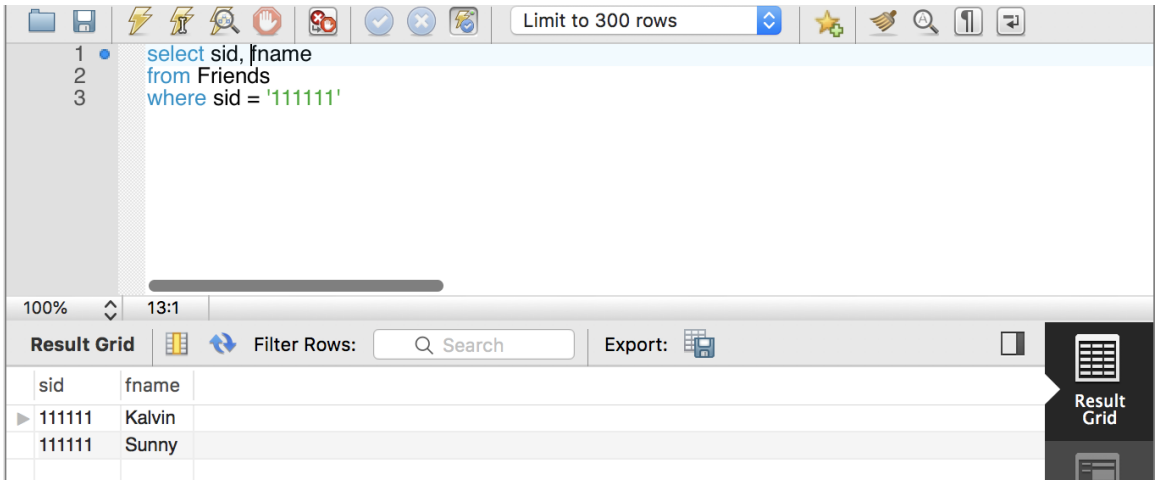
```
1  create table `Record` (
2    `accountid`  int(6) not null,
3    `accountname` varchar(10) not null,
4    `loginname` varchar(10) not null,
5    `password` varchar(10) not null,
6    primary key (`accountid`)
7  );
8
```

100% 25:6

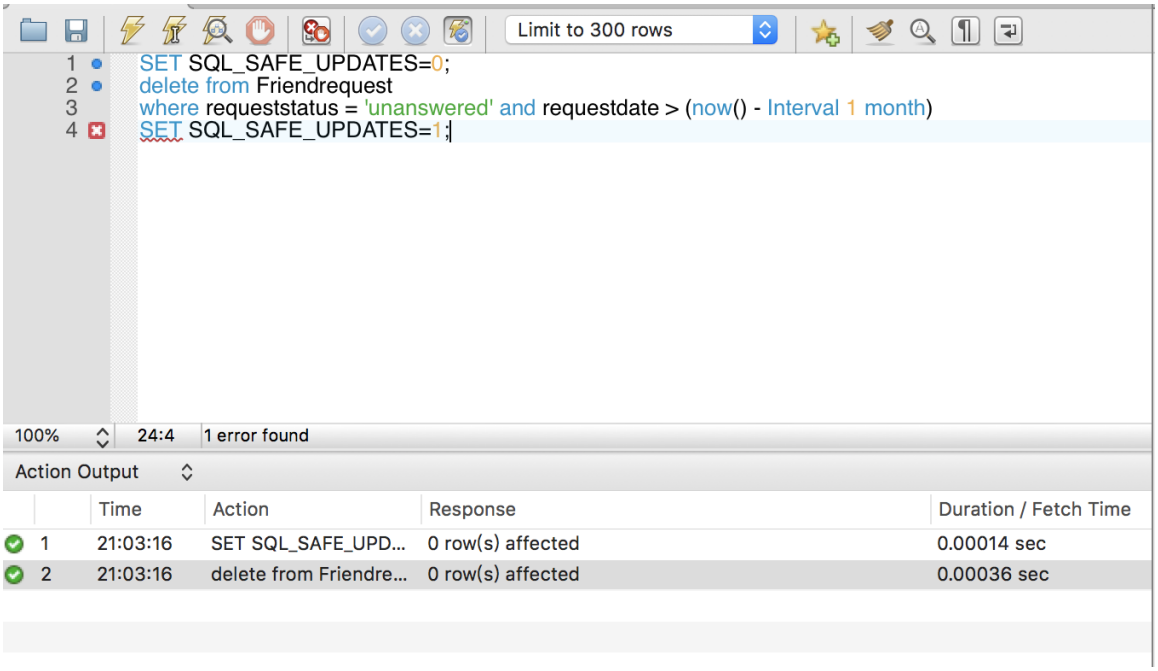
Action Output

	Time	Action	Response
1	17:41:52	create table `Record...	0 row(s) affected
2	17:42:05	SELECT * FROM Job...	0 row(s) returned

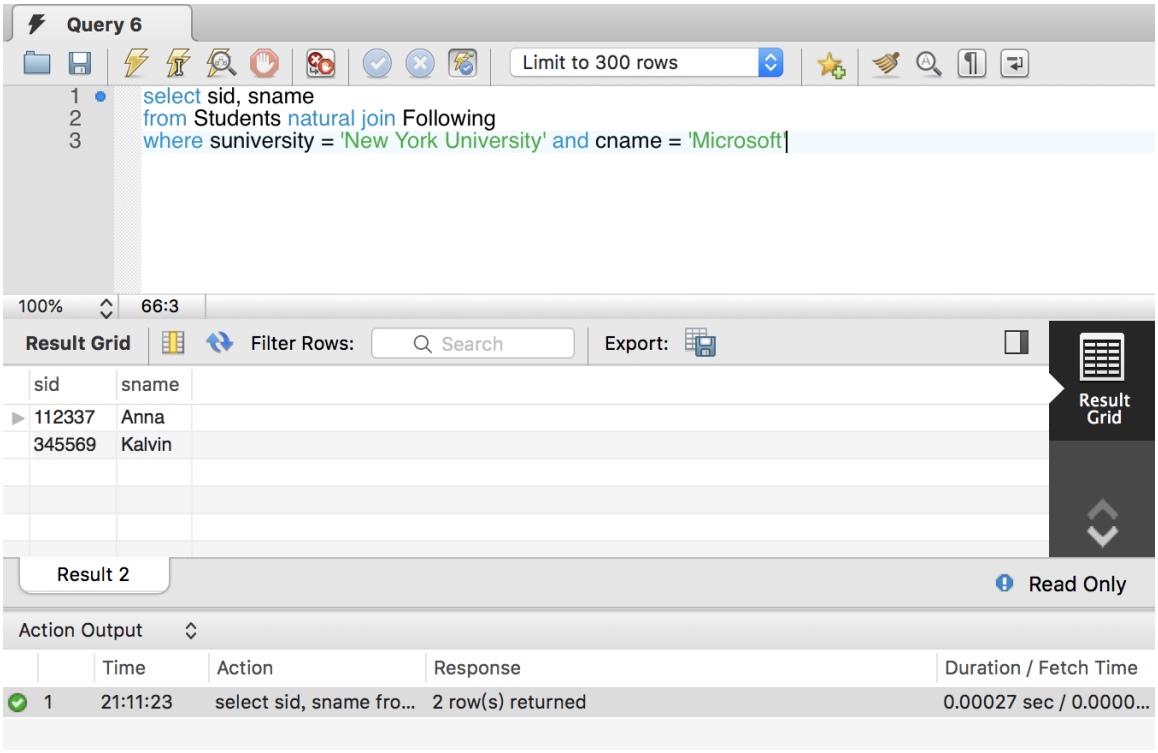
2) Here we need to select a particular user, so for example we want to select the student ID is 111111, his friends.



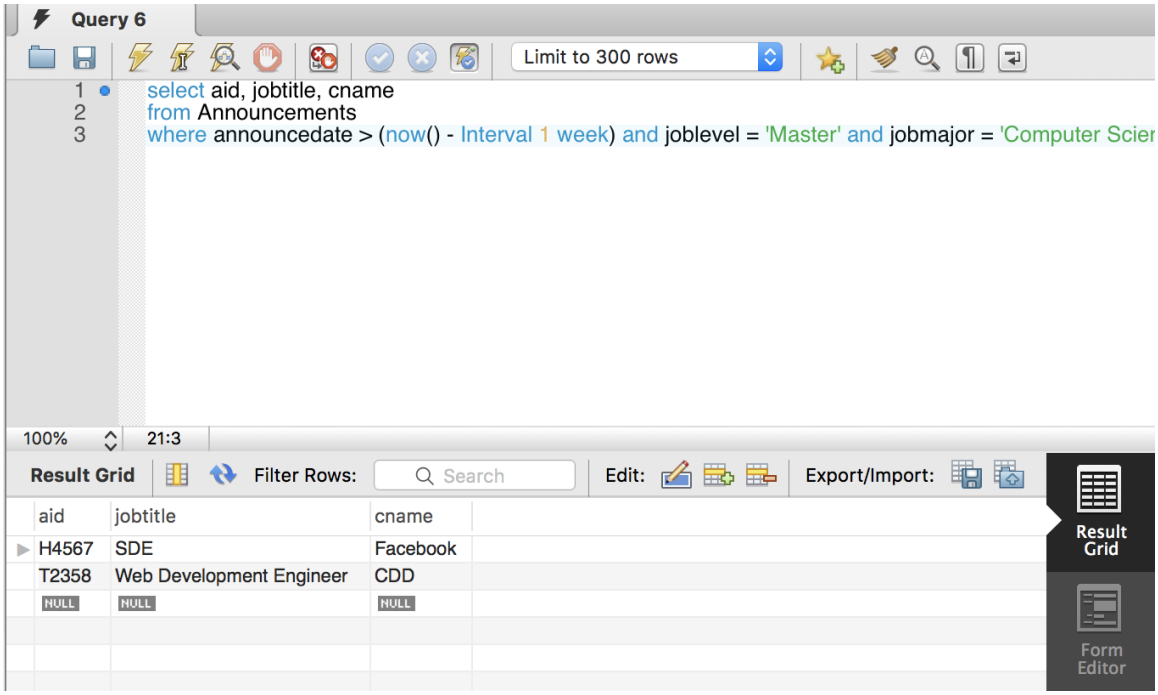
3) To delete a row in Friendrequest table, I need to set SQL safe update to 0, after that, set it back to 1 to guarantee the database security.



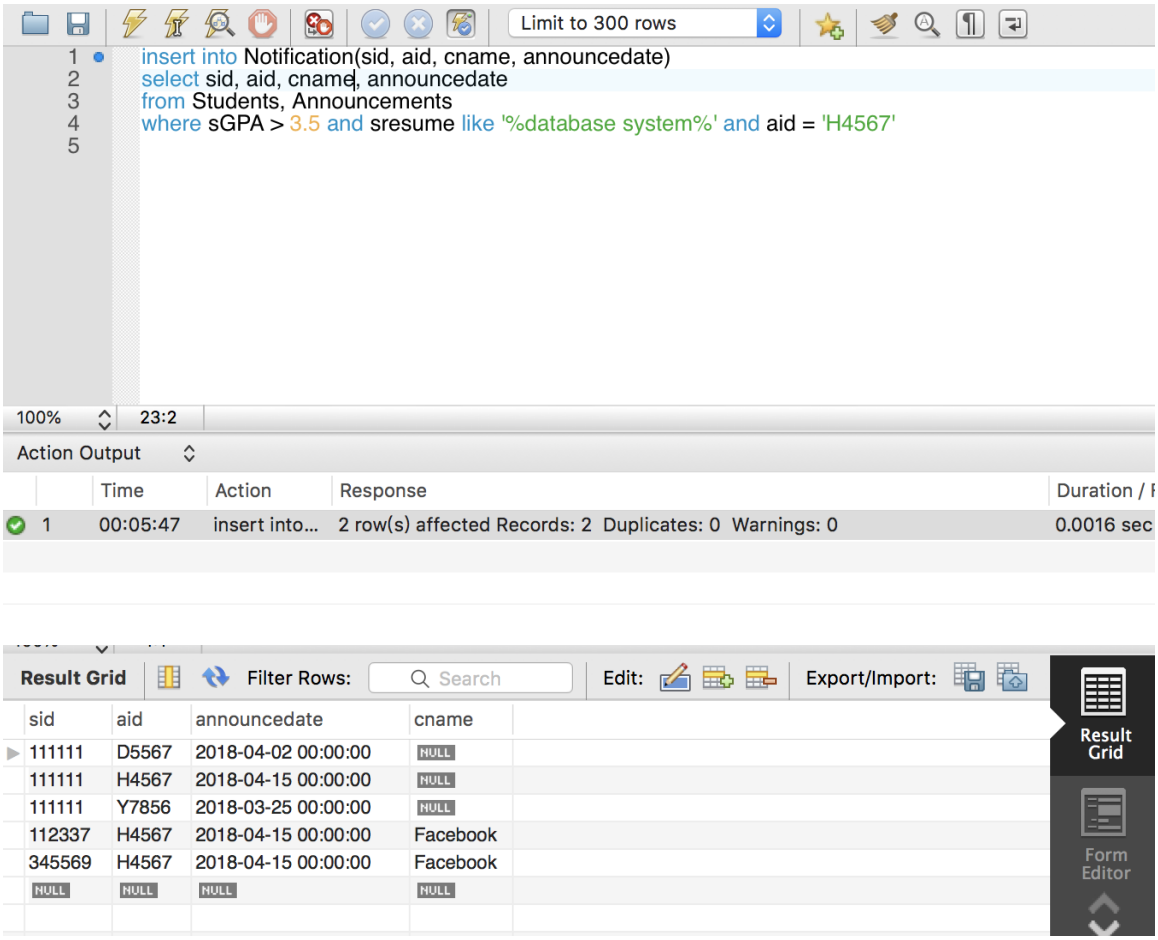
4) List all students from NYU meeting the requirement



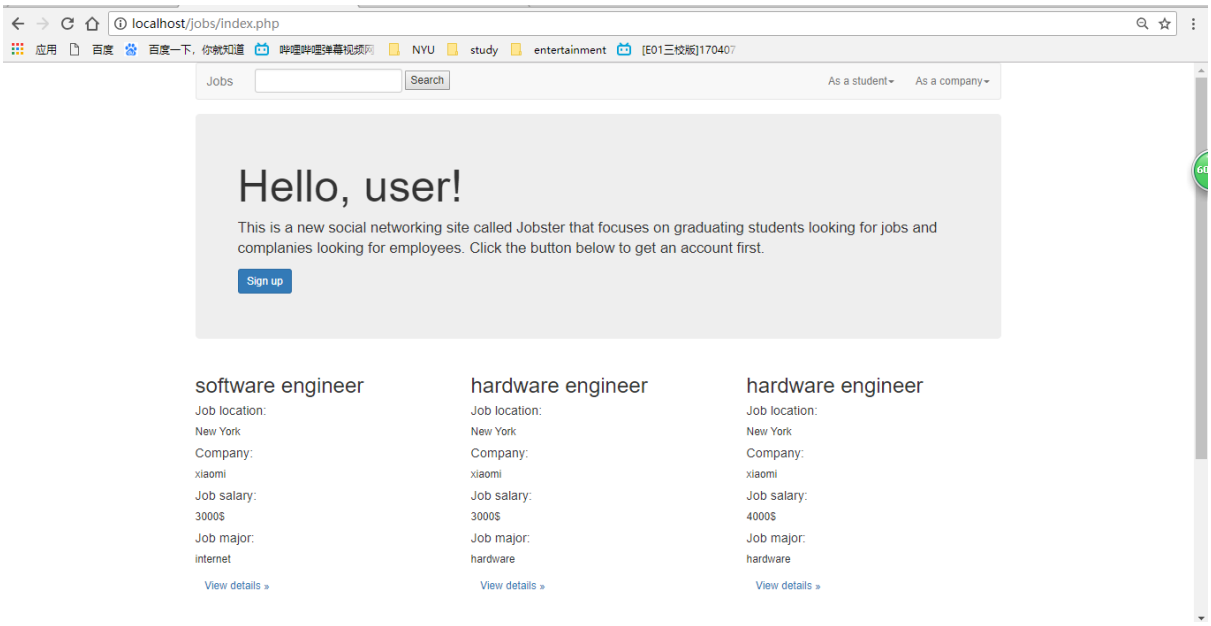
5) List all job announcements posted in last week which are looking for someone is MS and study at CS



6) For each student who meets the requirement, we add the sid, particular aid and its corresponding company name cname to the Notification table.

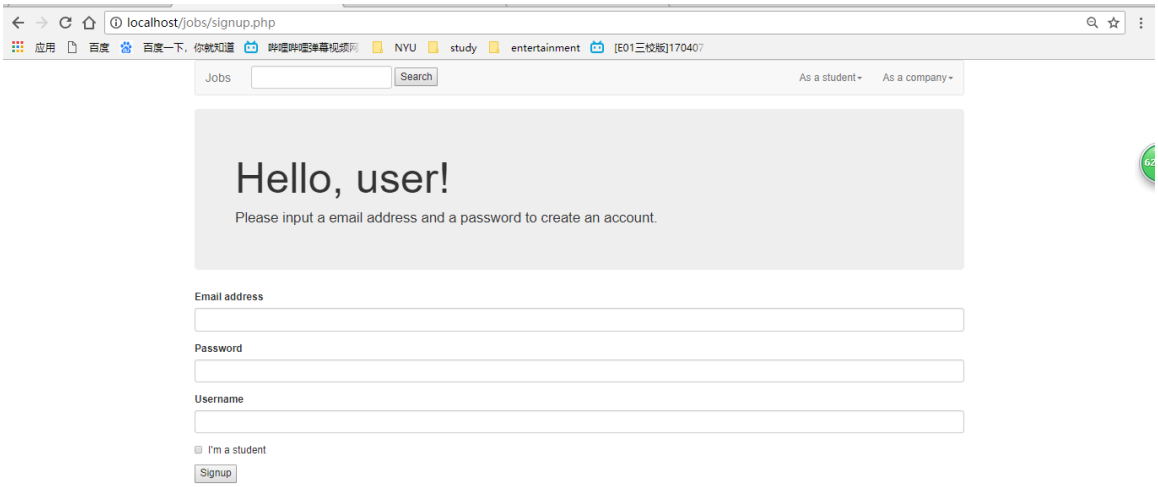


5. **Design of the web-based user interface**
In this project, we created the web-based user interface for the database designed in the first part of the project.



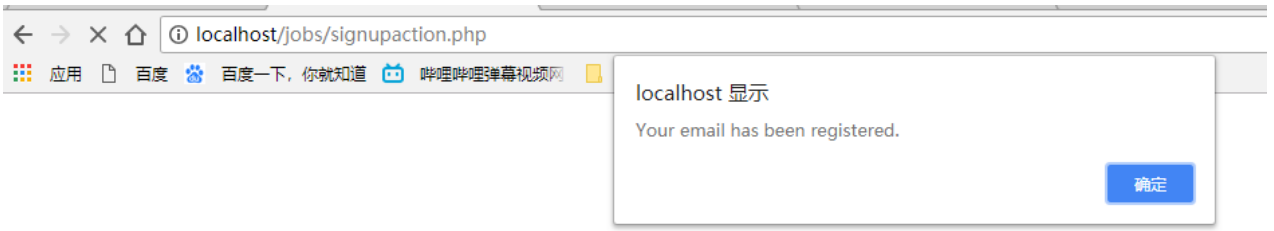
The home page in shown as above, including the "sign up", "search", "As a student", "As a company" part and listing all the jobs posted at the bottom which can be detailed by clicking the view details button.

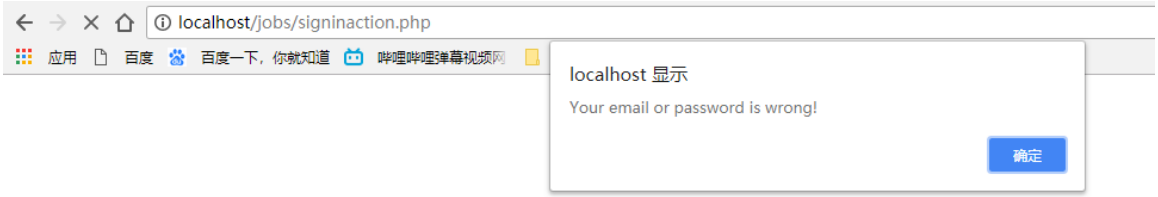
When clicking the sign up button, page as below will show up:

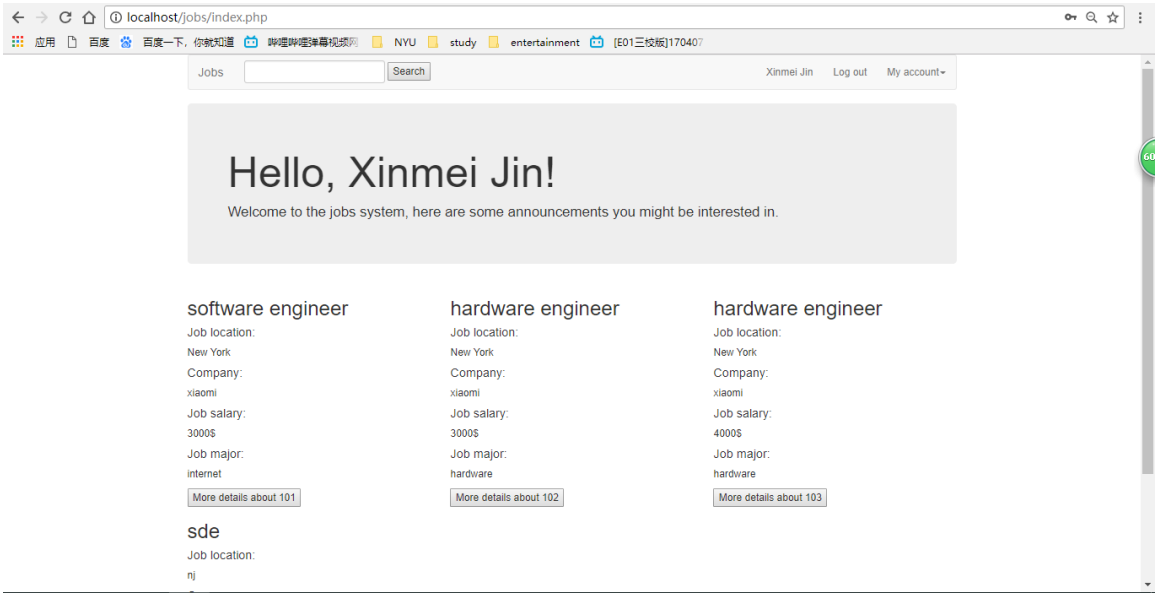


We designed the same sign up page for both companies and students and we put a "I am a student" selection for them to identify if they are companies or students.

We also designed a check part for the email that if this email has the right format and if the email has been registered before.



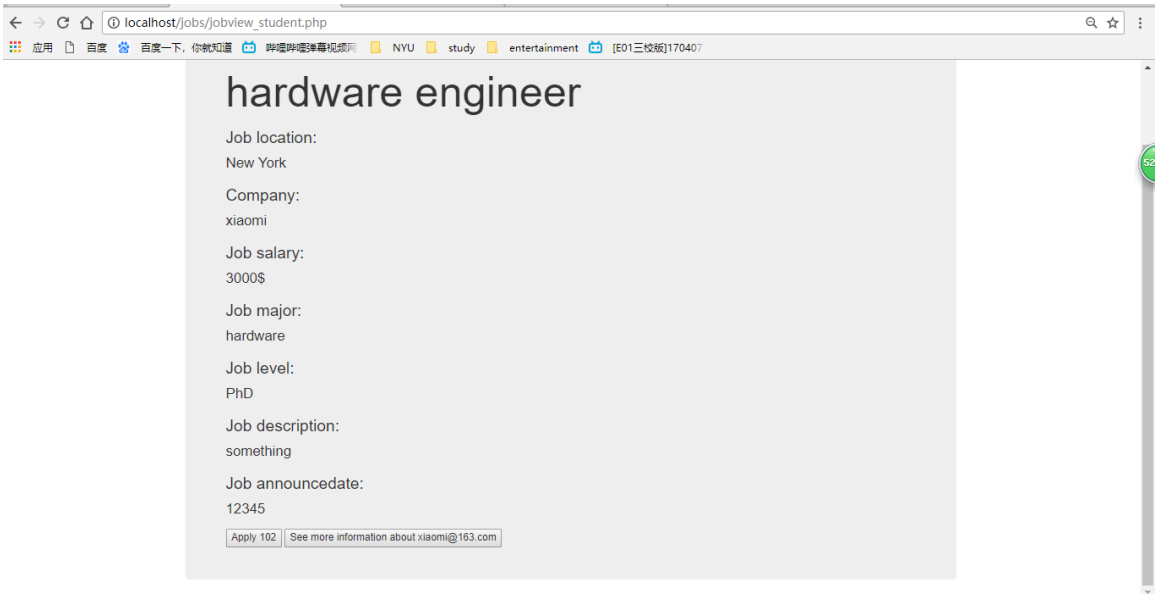




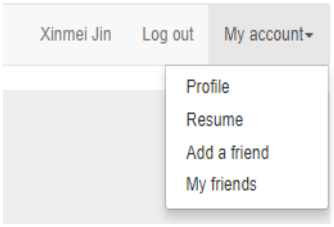
This is a sample homepage for a student. We set this student followed the company “xiaomi” and set her major as “internet”, so all the job related to internet and “xiaomi” are recommended for

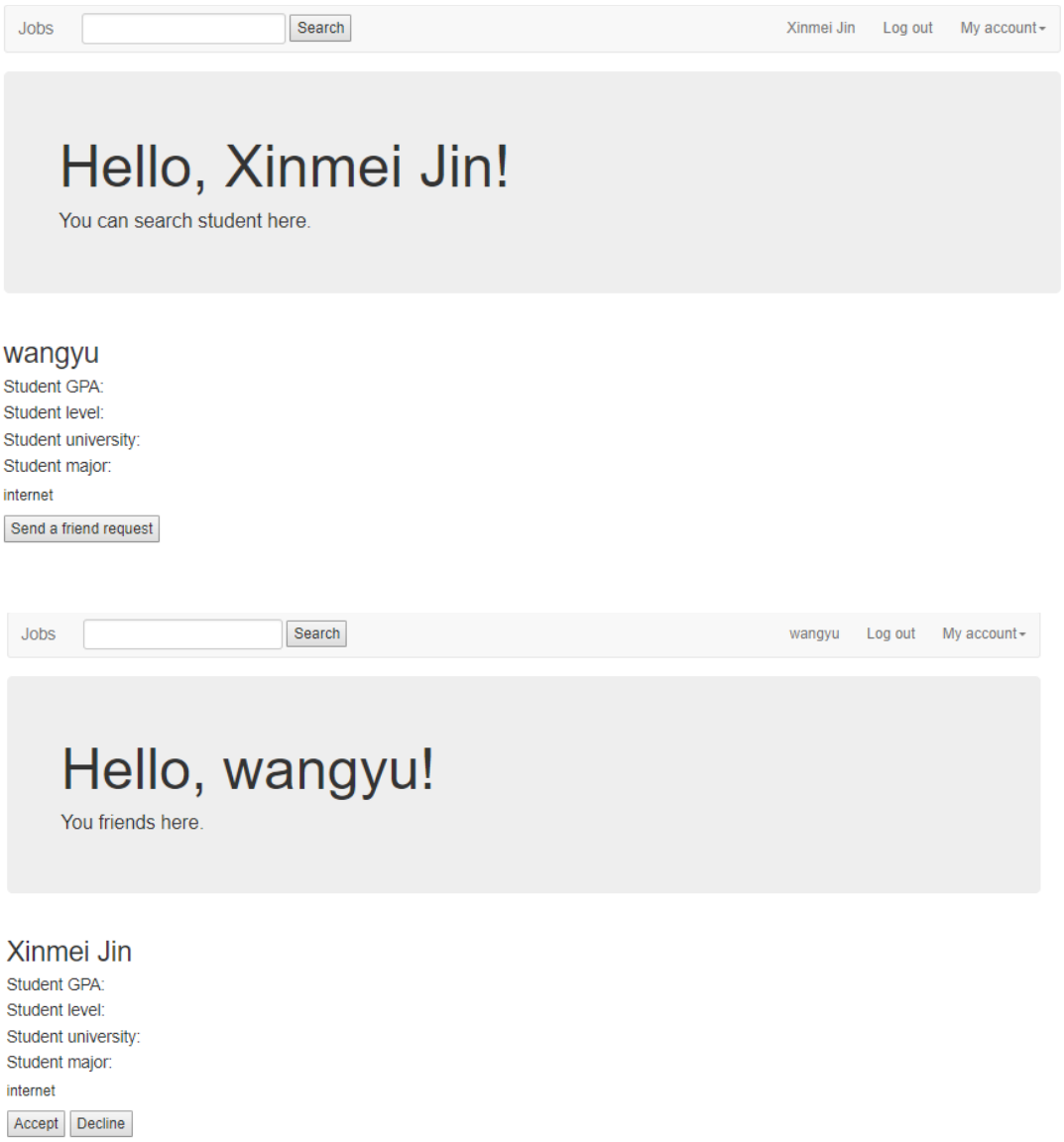
her, listing at the bottom of her homepage.

As clicking the view details button, students can see the details of the job, apply for the job as well as see more information about the company where the student can choose to follow the company.

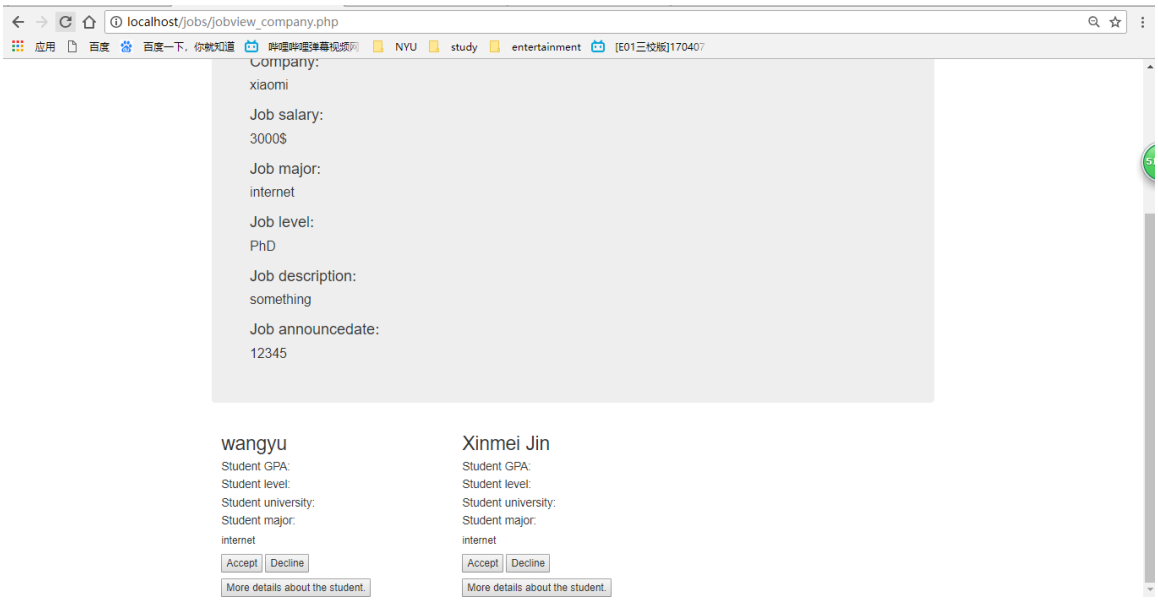


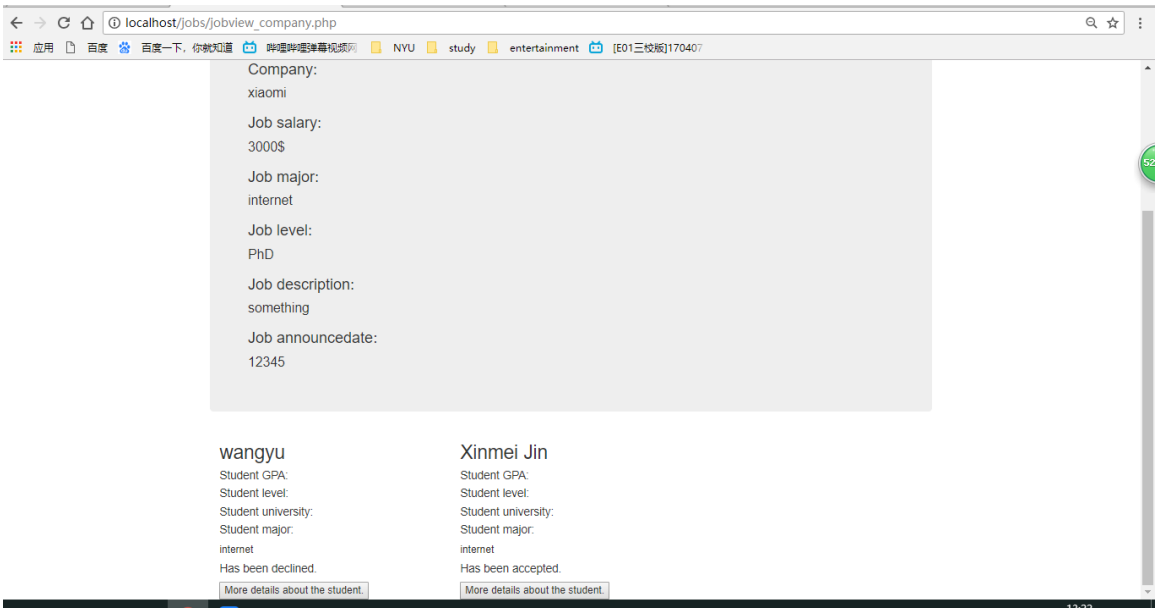
In the “My account” part, the student can edit his/her own profile and this profile can be seen by the companies and his/her friends. The student can also search for friends and add friends in this “My account” part. When receiving a friend request, the student can choose to accept or decline.



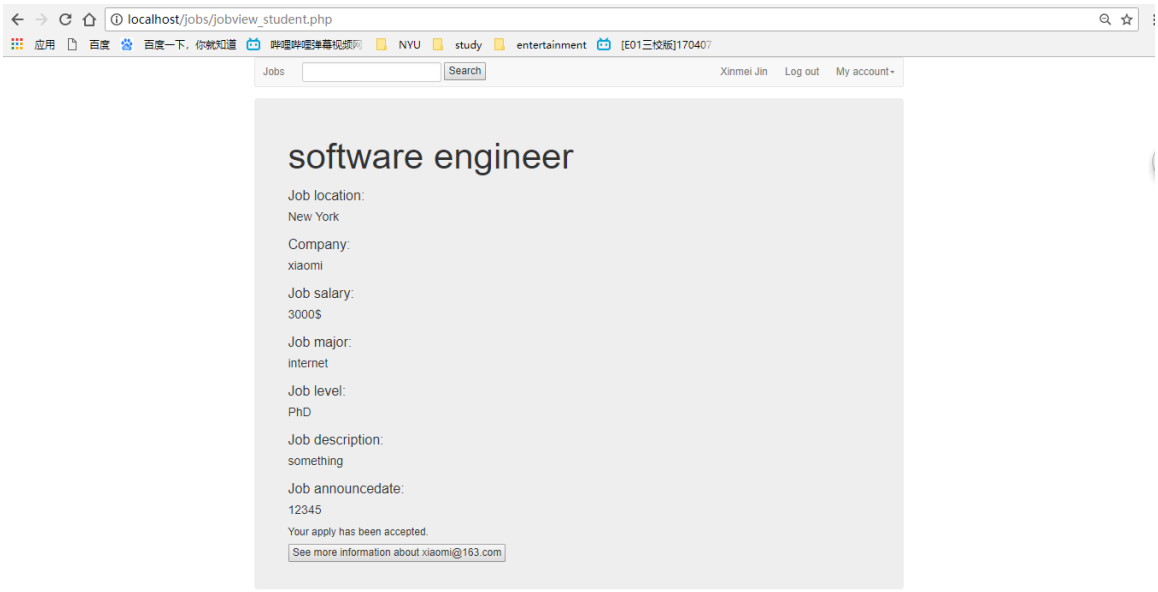


As the company part, after logging in, a company can see all the jobs they posted. When clicking into a specific job they posted, they can see all the applicants for this job and their profiles. They can decide to accept the students or decline them.

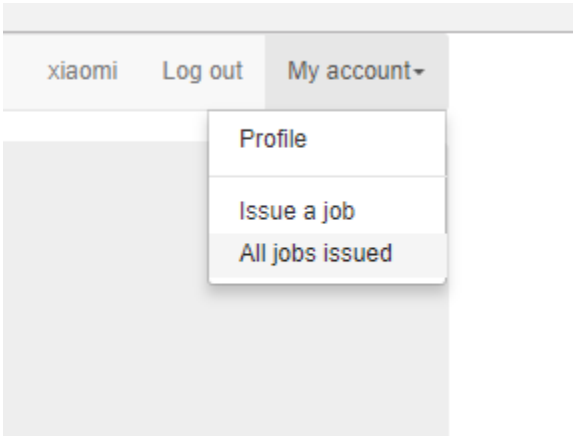




For students “Xinmei Jin”, she can see that her job apply has been accepted:



They can also edit the company profile which will be shown to the students. In the "My account" part, they can see all the jobs they posted and issue a new job which can be seen by all students.



What's more, on each page, click the "job" on the left top of each page can lead back to the main index homepage.

6. Further design prospect

In the company part, we designed the issue new job part, but we didn't consider the announcement part which can specific push to students who may be interested in this job or fit for this job. The just can be seen to all students. When a student follows a

company, he will see all the jobs posted by this company which may be more than 100 posts. We should make a more specific recommendation for students in the further completion of our project.

In the student part, we will perfect the the message function in the further design.

In this project, we used smarty for tpl (templates) model when coding, making the style of the whole project consistent and elegant.