



UJOB

**DataBase-Workshop Final Project Report**

**Group（6） member:**

**李瀚生 1930026065**

**马铭泽 1930006138**

**薛劭杰 1930026143**

**冯昊轩 1930026026**

Table of Contents

[Project Description 1](#_Toc73569230)

[Data Description, Data Preprocessing and Data Collection 1](#_Toc73569231)

[Data Analysis and Visualization 2](#_Toc73569232)

[Latest Version Of ER Diagram. 4](#_Toc73569233)

[Assumption 5](#_Toc73569234)

[Functional Dependencies 5](#_Toc73569235)

[The SQL codes and the explanations 5](#_Toc73569236)

[Website Design and Feature Implementation 6](#_Toc73569237)

[Difficulties we Encountered 12](#_Toc73569238)

## 

## Project Description

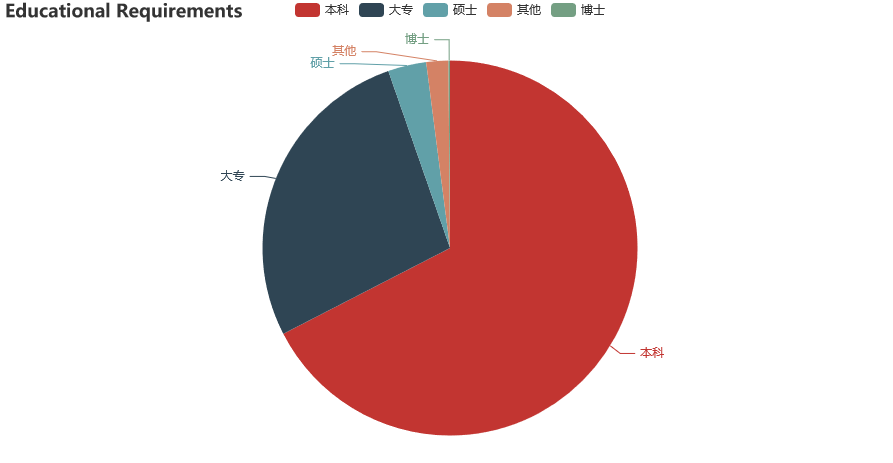
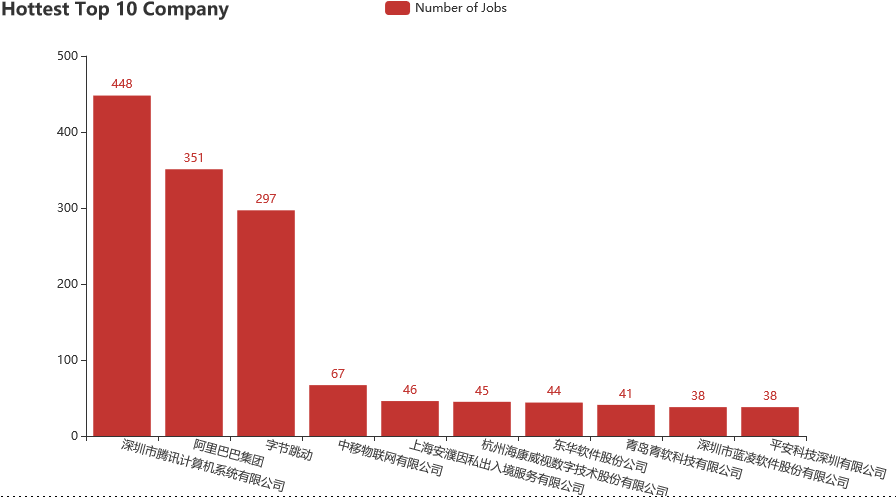
* For students, it is difficult to find internship positions and most websites offer data for a cost, often accompanied by this many advertisements. For companies: the current business is highly competitive, and there is an additional platform to publish recruitment needs. It is convenient for companies to find and explore the talents they want more widely, which is a mutual benefit and win-win result. We designed this site to make it easier for CDS students to find jobs or internships based on their professional strengths.

## Data Description, Data Preprocessing and Data Collection

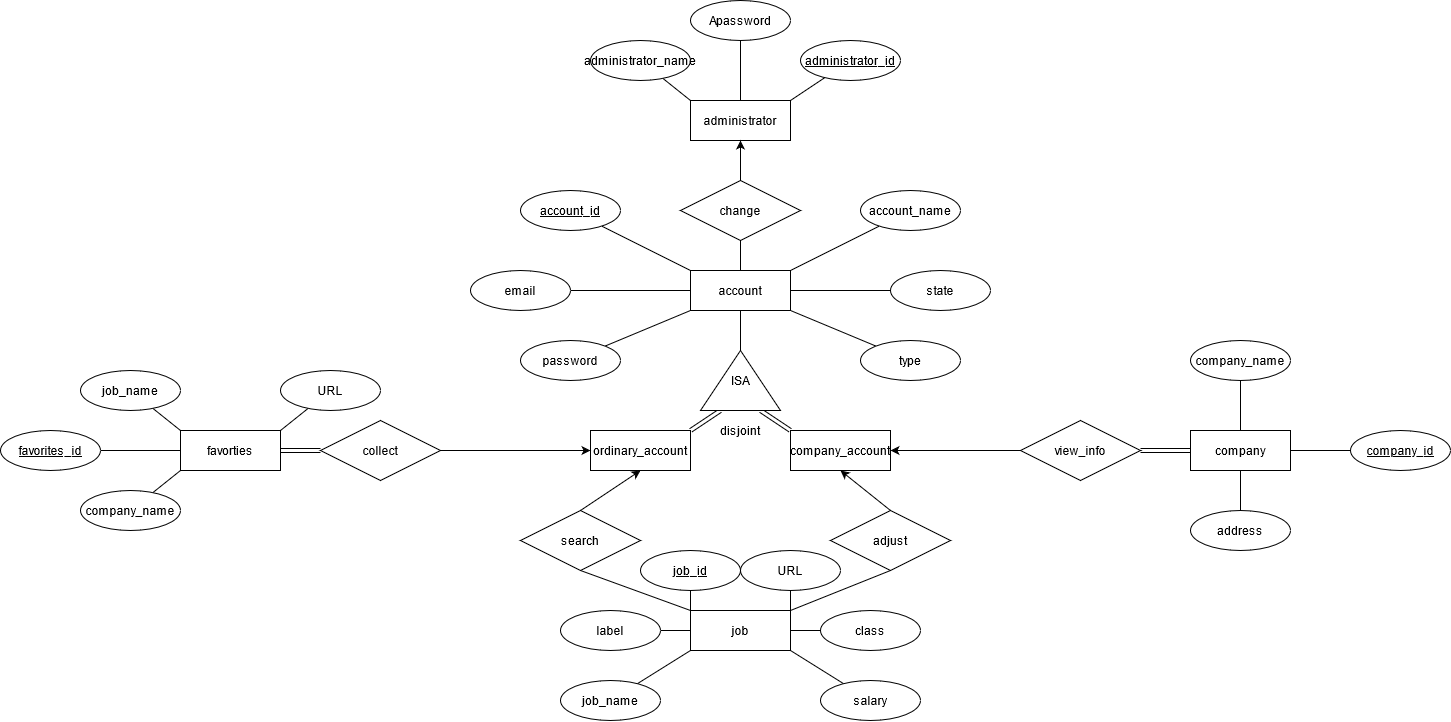
* Data Description
  + The data we crawled down is divided into 7 categories, which are company name, job name, geographic location, company welfare, salary (in tens of thousands), education, and detailed URLs.
* Data Preprocessing
  + For the job welfare part, since some companies do not offer job welfare, we set it to an empty list by default.
  + For the salary part, since the salary offered by each company is different and most of them are in a range, we just take the minimum value here. Furthermore, some companies give annual salary and some give monthly salary, so here we take monthly salary as the unit. Secondly, some companies give the unit of $1000, some companies give the unit of $10,000, here we unified by $10,000 as the unit, for some do not give the specific salary, we will set the default value -1.
  + For the education part, we divide it into Specialty, bachelor, master, doctor, postgraduate, and if not in the above education, we divide it into others.
* Data Collection
  + We used the request+selenium group and crawled the data, and saved the data of each job into a txt file, which stored the seven kinds of data we needed, that is, company name, job name, company welfare, salary, job type, required education, and detail page URL. We initially used the selenium module to call the chrome driver to crawl the data, but it was slow, so we logged into the site with our account and made sure that the cookies on the site were not changed, and then used the requests module to set a cookie to simulate sending requests. We also used a multiprocessing thread pool for asynchronous crawling to improve efficiency. The figure below shows the final crawling time we need.

## Data Analysis and Visualization

You can see the screenshot below, and you can see the full page in following html document: [detailed\_image](file:///C:\Users\hp\Desktop\linear_regression\linear_regression_project\resize_render.html)

   We made three charts, the first is a pie chart, representing the educational requirements of the company, we can see that most of the companies’ educational requirements are bachelor’s degree or above. The second graph is a word cloud graph, the size of the word in the graph represents its frequency, you can see that most of the company’s benefits include five insurance and one pension, food and beverage subsidies, etc. The third chart is a bar chart, which shows the top ten companies that recruit the most number of jobs, we can see that Tencent, Ali and other such large companies have a great demand for positions

## Latest Version Of ER Diagram.



## Assumption

* Assumption 1: A company corresponds to multiple jobs, but a job can only correspond to one company
* Assumption 2: A user cannot be a company user and a student user at the same time
* Assumption 3: A job posting only corresponds to one programming language
* Assumption 4: The default wage in the data set is the minimum wage, because some jobs are given a salary range
* Assumption 5: A company can at most send 1000 jobs(Although all companies did not exceed this limit).
* Assumption 6: The classification of languages is as follows Java, Python, C++, PHP and others.

## Functional Dependencies

We have 5 relations.

1. Change(account\_id, adminstartor\_id)
2. Adjust(job\_id, account\_id)
3. Search(job\_id, account\_id)
4. Collect(favorites\_id, company\_name, job\_name, URL, account\_id)
5. View\_info(company\_id, company\_name, address, account\_id)

Explanation:

We have five relations. Each relation satisfies BCNF. In each relationship, our functional dependency is the primary key that determines every attribute. So, each of our relationships is in line with BCNF. \*\*\*

## The SQL codes and the explanations

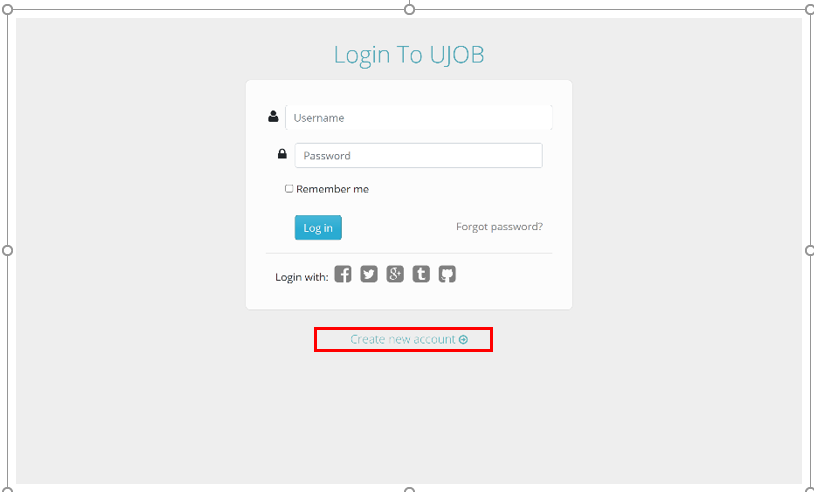
SQL Code: The sql code is in the attachment, part of it is in create\_table.sql and the other part is in django. Explanation: In order for more people to find their own suitable jobs more conveniently, we designed this database. There is a total of seven entity sets in our database. The database is designed mainly around three entity sets of accounts, jobs and administrators.  
Accounts are composed of company accounts and ordinary accounts. Company accounts can upload jobs and make changes to job information in real time. Ordinary accounts can filter or search their favorite jobs and can put their intentional jobs into their favorites. According to the needs submitted by the user, the administrator can change the user’s information. State shows the current state of the account, such as login or logout. The language at work is used to classify different languages. Languages are classified into Java, C++, PHP, Python and others. Lastly, the label has different types. For example, five social insurance and one housing fund, closed on weekends and so on.

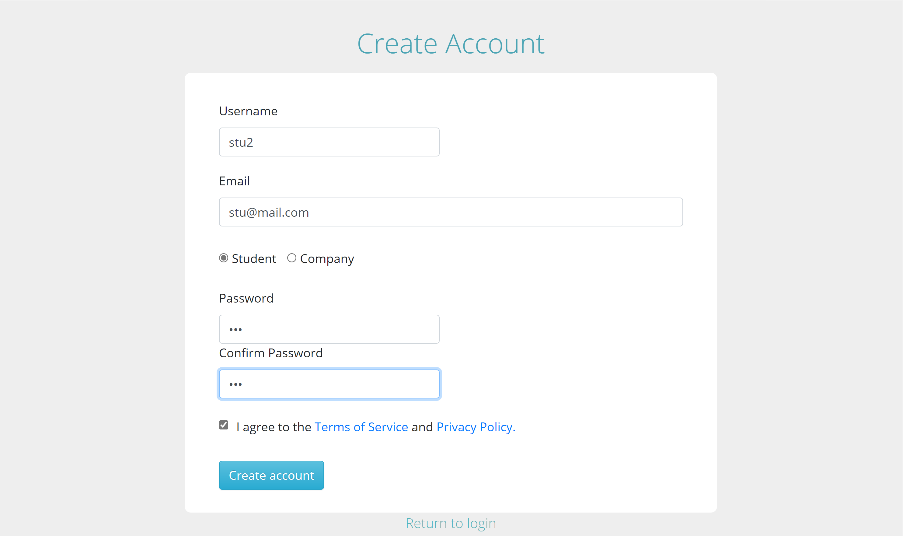
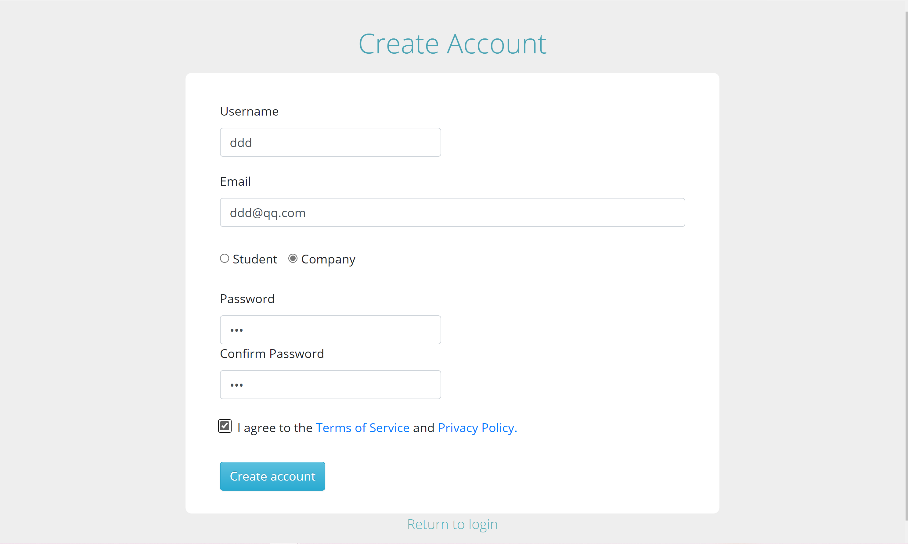
## 

## Website Design and Feature Implementation

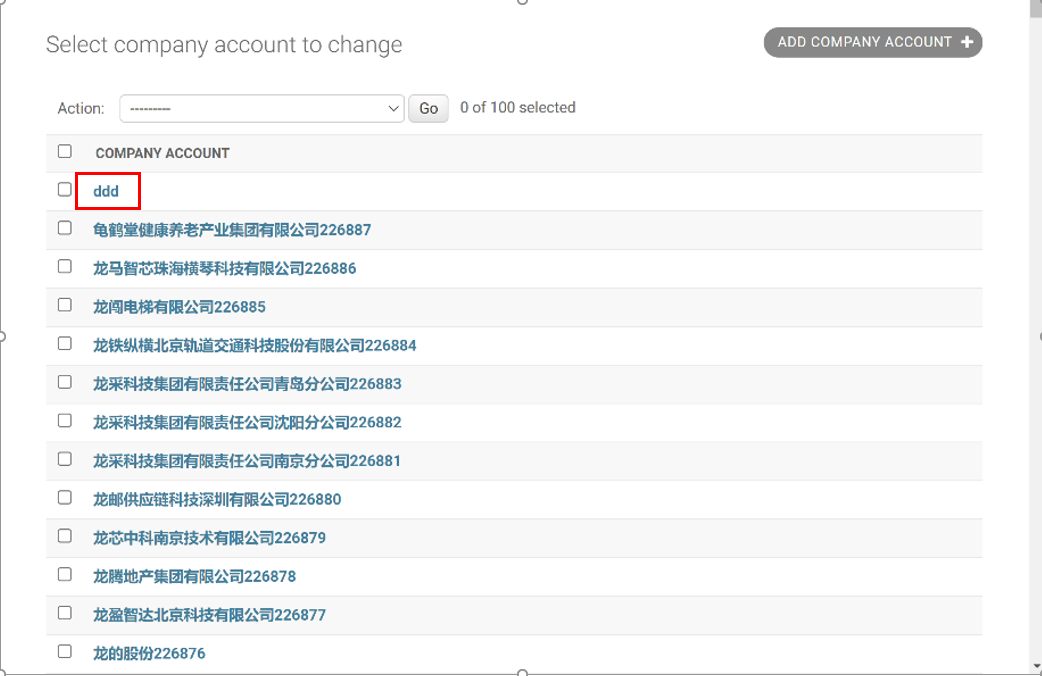
Function realization:

1. Account login and logout
2. account registration
3. Choose different types of accounts
   1. For company account:
      1. Upload job
      2. Delete job
      3. Adjust job information
      4. View account information
      5. Change account information
      6. Change password
   2. For ordinary account:
      1. Find jobs based on keywords
         1. Keywords include:
            1. Job Name
            2. Language Type
            3. Area
      2. Follow the link to jump to the job details page
      3. View account information
      4. Change account information
      5. Change password
4. Administrator:
   1. Increase user
   2. Delete user
   3. Unable to change the user password (the password on the database side is encrypted)

Demo Test:  The homepage of our website needs to be logged in to enter. If the wrong information is entered, the page will not be redirected.

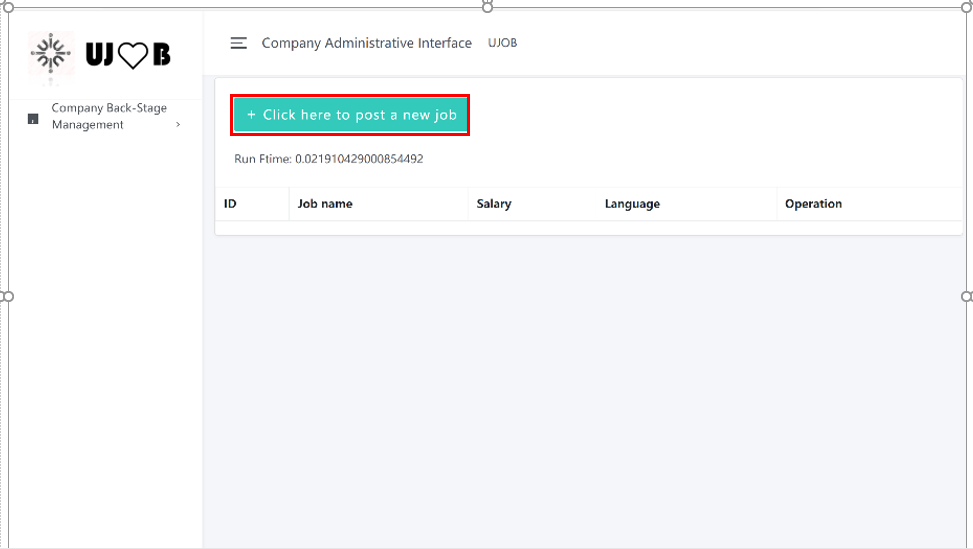
 

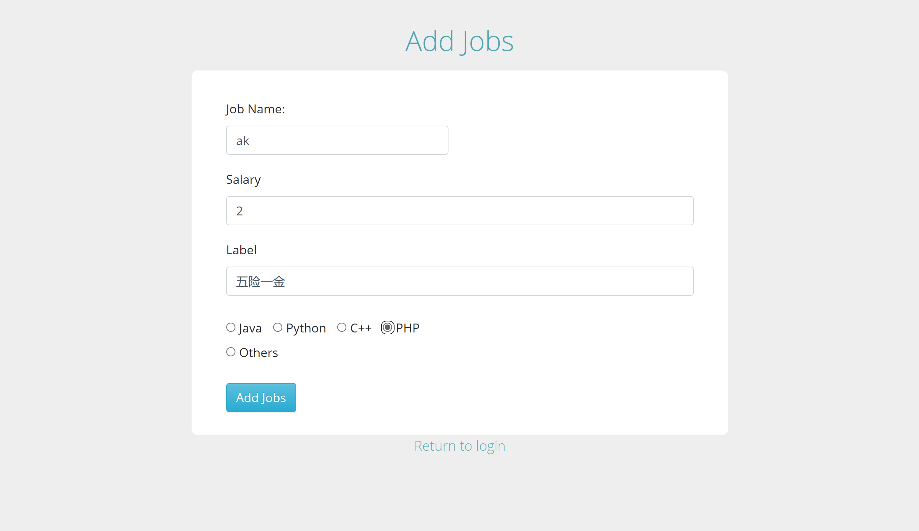
For new users, we can create a new account. In the registration interface, we need to fill in the user’s name, email password and other information. In the mailbox column, if you enter an incorrectly formatted mailbox, the system will prompt the user that there is a problem with the format of the mailbox entered. In the account selection type, the user can choose whether to create a company account or an ordinary account. In the fields for entering password and confirming password, if two different passwords are entered, the system will prompt the user that the passwords entered twice are not the same.



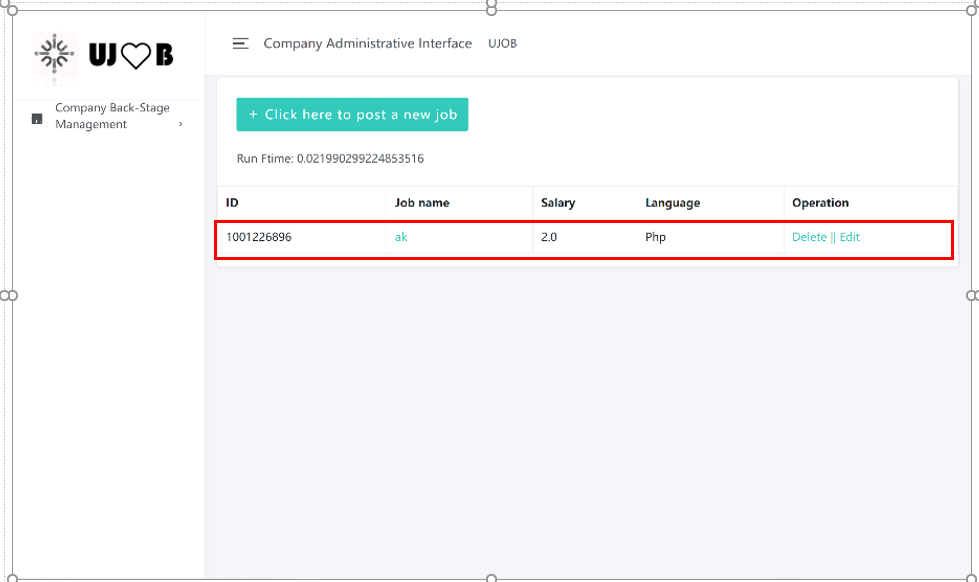
In addition, in the manager interface, we can see whether the database records and stores user requests. These requests include account registration and subsequent company users’ additions, deletions, and modifications to jobs.

For company accounts:

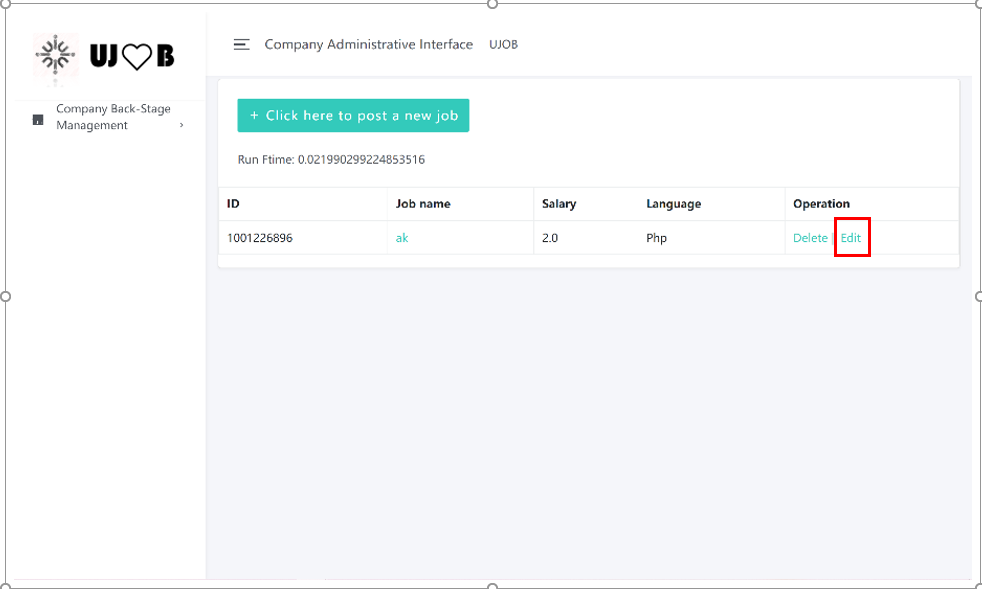
 After the company accounts log in, you will enter such an interface. In this interface, company users can view, upload, change, and delete jobs. After adding work, the interface will add corresponding work information. Complete the additional work by filling in the name of the job, salary, label and language.



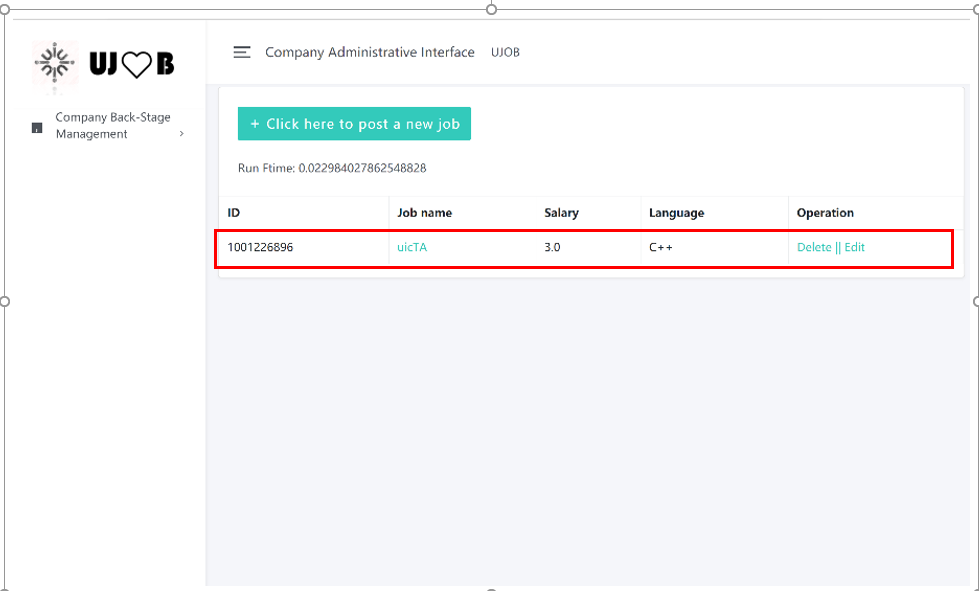
After adding:



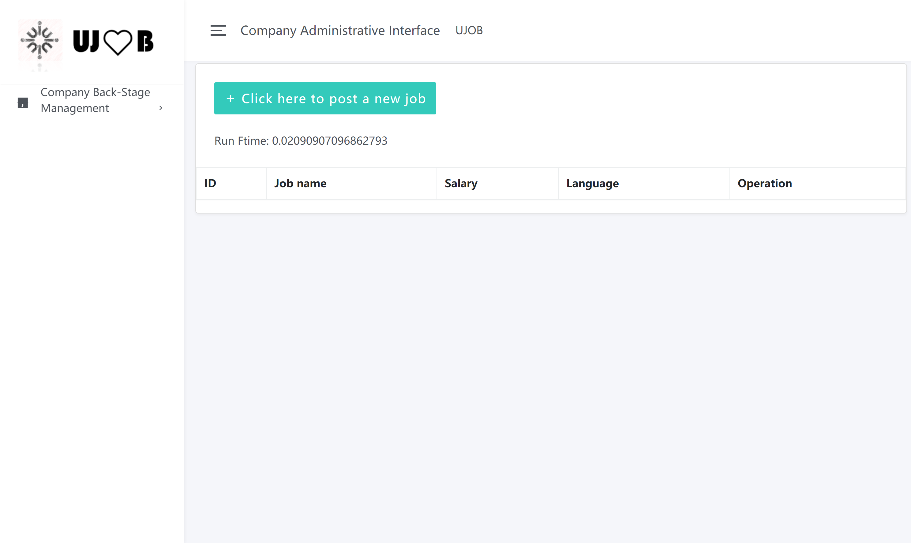
If you need to change the job information, the user can change the job information by clicking the edit button.



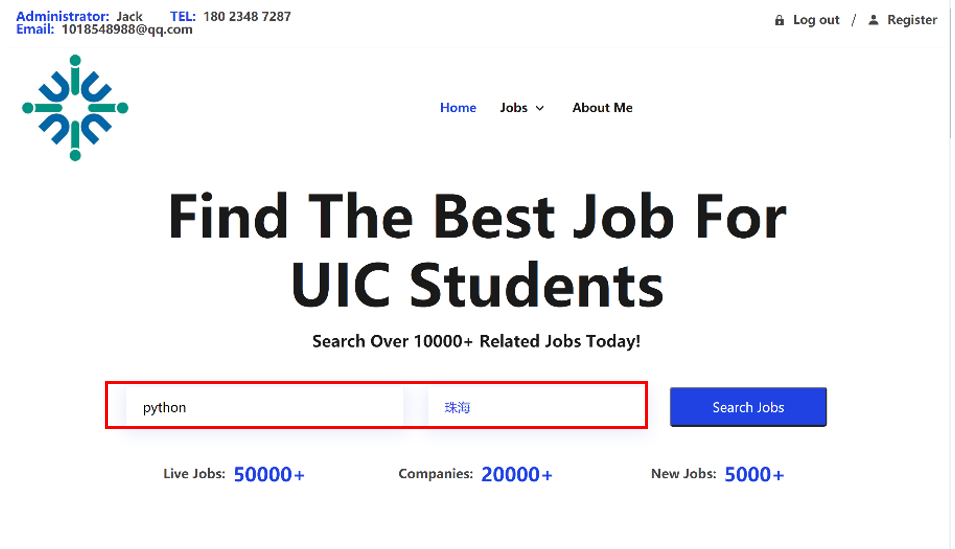
After editing:



If the job is full, the company can delete the job. After deleting:



For ordinary accounts: Users can search according to their needs. In the search bar, users can search based on the name of the job, the language type of the job, and the address of the job. For example, now, I use the Zhuhai Python job to test it.



After searching, the page will jump to the page with the corresponding job. Users can click on the ‘Browse Job’ next to it to jump to the company-related details page and learn more about it.



## Difficulties we Encountered

1. Database connection problem: The relationship between database migration mapping and inverse mapping (inspectdb), foreign key constraints may also cause data loss and import failure.
2. Routing calls between websites: Many of the url format and grammar rules are relatively unfamiliar regular expressions, and may not get when the parameters in the url are adjusted.
3. POST request and GET request are easier to report errors, because the default value of the front-end (js, html) may cause errors in your POST request, so what you get is a GET request, which is more difficult to handle.
4. Speed optimization problem: When faced with large data, the search speed will be relatively slower, and the search efficiency will be twice the result with half the effort when a good indexing method and algorithm can be used. But you need to keep trying to make changes to get a better conclusion