UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

CS411: DATABASE SYSTEMS

Mining Rig Assembly

Author:

Jiajun Chen (jiajunc2)
Ho Yin Au (hoyinau2)
Xiao Tang (xtang27)
Zhenye Na (zna2)

Date: April 24, 2018

Contents

1 Introduction

Mining Rig Assembly is a Web-based Application that allows the users to browse different parts of a mining rig, store rig setups, and estimate the performance of their setups on one integrated site. Users can set their expected payback periods for the setups, and our application will calculate the potential profits of them based on real-time price information, and notify the user when their expected payback periods can be achieved. Whats more, this website can visualize the performance comparison between different setups and components for specific users.

For more detailed information, please visit

- Mining Rig Assembly (Website)
- Github (for working updates)
- Mining-Rig-Assembly Github Organization (final result)
- Project Introduction Website

2 Description

1. There are several websites that help users assemble mining rigs for bit-coin, but few of them focus on Ether. Our website targets the Ether investors.
2. Users can know when their expected payback period will be met. Our website will calculate the payback period of different setups dynamically, and send an email to the users once the payback period meet the expectation.
3. Users can obtain the performance comparisons between their setups and components. We not only visualize the performance of the setups and the components, but also the comparison between setups and components. This is a practical function for users as they can view and choose their setup more intuitively.

3 Database

For this application, we used MariaDB as the Database Server (Server version: 10.0.33 - MariaDB (MariaDB Server)). In the database, we have 24 gpu records, 27 cpu records, 28 motherboard records, 27 power supply unit records and 23 RAM, leading to 129 components totally so far. Furthermore, we have tables to store setups that user created and username with corresponding password.

4 ER-Diagram and Schema

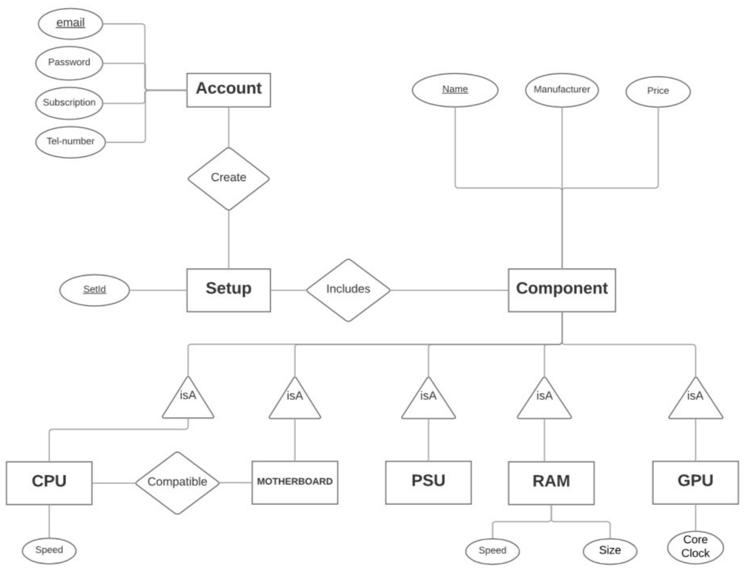


Figure 1: ER-Diagram

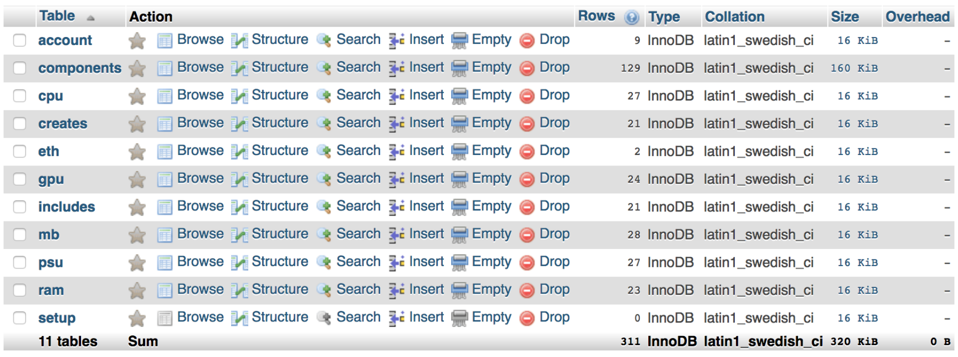
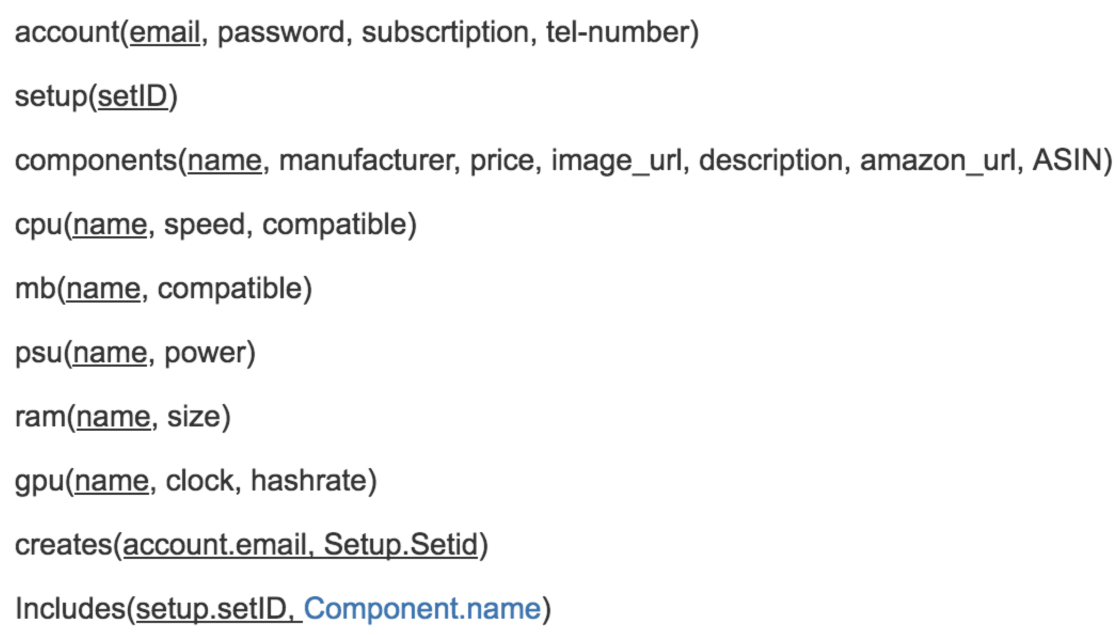


Table	Action	Rows	Type	Collation	Size	Overhead
account	Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16 KiB	-
components	Browse Structure Search Insert Empty Drop	129	InnoDB	latin1_swedish_ci	160 KiB	-
cpu	Browse Structure Search Insert Empty Drop	27	InnoDB	latin1_swedish_ci	16 KiB	-
creates	Browse Structure Search Insert Empty Drop	21	InnoDB	latin1_swedish_ci	16 KiB	-
eth	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
gpu	Browse Structure Search Insert Empty Drop	24	InnoDB	latin1_swedish_ci	16 KiB	-
includes	Browse Structure Search Insert Empty Drop	21	InnoDB	latin1_swedish_ci	16 KiB	-
mb	Browse Structure Search Insert Empty Drop	28	InnoDB	latin1_swedish_ci	16 KiB	-
psu	Browse Structure Search Insert Empty Drop	27	InnoDB	latin1_swedish_ci	16 KiB	-
ram	Browse Structure Search Insert Empty Drop	23	InnoDB	latin1_swedish_ci	16 KiB	-
setup	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
11 tables	Sum	311	InnoDB	latin1_swedish_ci	320 KiB	0 B

Figure 2: Database Schema

Database schema Detail:



account(email, password, subscrtiption, tel-number)

setup(setID)

components(name, manufacturer, price, image_url, description, amazon_url, ASIN)

cpu(name, speed, compatible)

mb(name, compatible)

psu(name, power)

ram(name, size)

gpu(name, clock, hashrate)

creates(account.email, Setup.Setid)

Includes(setup.setID, Component.name)

5 Data Collection

We use Amazon Product Advertising API to collect product data such as product name, price, description, and image. We first use the API by searching with keywords, such as CPU, GPU, motherboard, power supply and RAM. We specify the response group with Images, ItemAttributes. Then, we obtain 10 records of specified type in xml format. We parse the xml to retrieve product attributes, and use SQL commands to insert products into database. We can also set Itempage to get different records in the search.

6 Functionality

1. Display list of components in the database(CPU, GPU, motherboard, power supply, RAM).
2. Store user setup specified in user input.
3. Display specified component information.
4. Calculate total price and payback period.
5. Compare component parts between different setups with visualization.

7 SQL Queries

7.1 Basic Query

When user want to create setup, a list of all components appears. User pick one component for each type of CPU, GPU, MotherBoard, Power-Supply, and RAM. Then, setup is created and stored in the database.
For example:

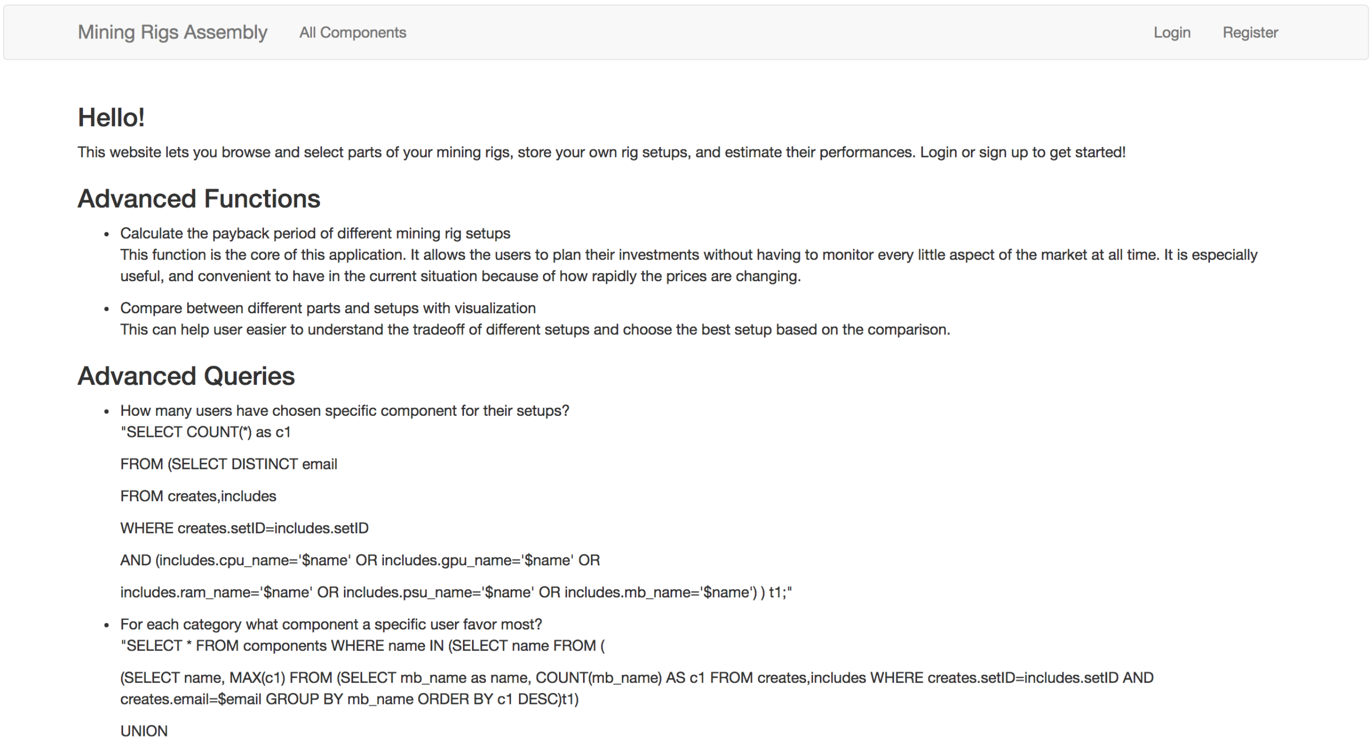
```
SELECT *
FROM components
WHERE name IN (SELECT name
               FROM cpu);
```

7.2 Advanced Query

```
SELECT *
FROM components
WHERE name IN (SELECT name
               FROM ((SELECT name, MAX(c1)
                      FROM (SELECT mb_name as name,
                                   COUNT(mb_name) AS c1
                            FROM creates, includes
                            WHERE creates.setID=includes.setID
                              AND creates.email=$email
                            GROUP BY mb_name
                            ORDER BY c1 DESC) t1)

UNION

(SELECT name, MAX(c1)
FROM (SELECT cpu_name as name, COUNT(cpu_name) AS c1
      FROM creates, includes
      WHERE creates.setID=includes.setID
        AND creates.email=$email
      GROUP BY cpu_name
      ORDER BY c1 DESC) t2)

UNION

(SELECT name, MAX(c1)
FROM (SELECT gpu_name as name, COUNT(gpu_name) AS c1
      FROM creates,includes
      WHERE creates.setID=includes.setID
        AND creates.email=$email
      GROUP BY gpu_name
      ORDER BY c1 DESC) t3)

UNION

(SELECT name, MAX(c1)
FROM (SELECT ram_name as name, COUNT(ram_name) AS c1
      FROM creates,includes
      WHERE creates.setID=includes.setID
        AND creates.email=$email
      GROUP BY ram_name
      ORDER BY c1 DESC) t4)

UNION

(SELECT name, MAX(c1)
FROM (SELECT psu_name as name, COUNT(psu_name) AS c1
      FROM creates,includes
      WHERE creates.setID=includes.setID
        AND creates.email=$email
      GROUP BY psu_name
      ORDER BY c1 DESC) t5))

      namet);
```

8 Data-flow

Below is the procedure of a user entering this application.



Hello!

This website lets you browse and select parts of your mining rigs, store your own rig setups, and estimate their performances. Login or sign up to get started!

Advanced Functions

- Calculate the payback period of different mining rig setups
  This function is the core of this application. It allows the users to plan their investments without having to monitor every little aspect of the market at all time. It is especially useful, and convenient to have in the current situation because of how rapidly the prices are changing.
- Compare between different parts and setups with visualization
  This can help user easier to understand the tradeoff of different setups and choose the best setup based on the comparison.

Advanced Queries

- How many users have chosen specific component for their setups?
  "SELECT COUNT(*) as c1

  FROM (SELECT DISTINCT email

  FROM creates,includes

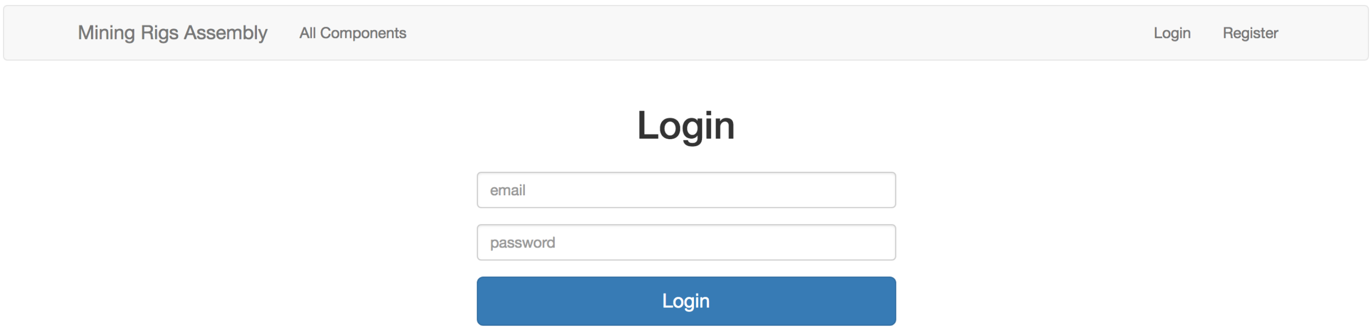
  WHERE creates.setID=includes.setID

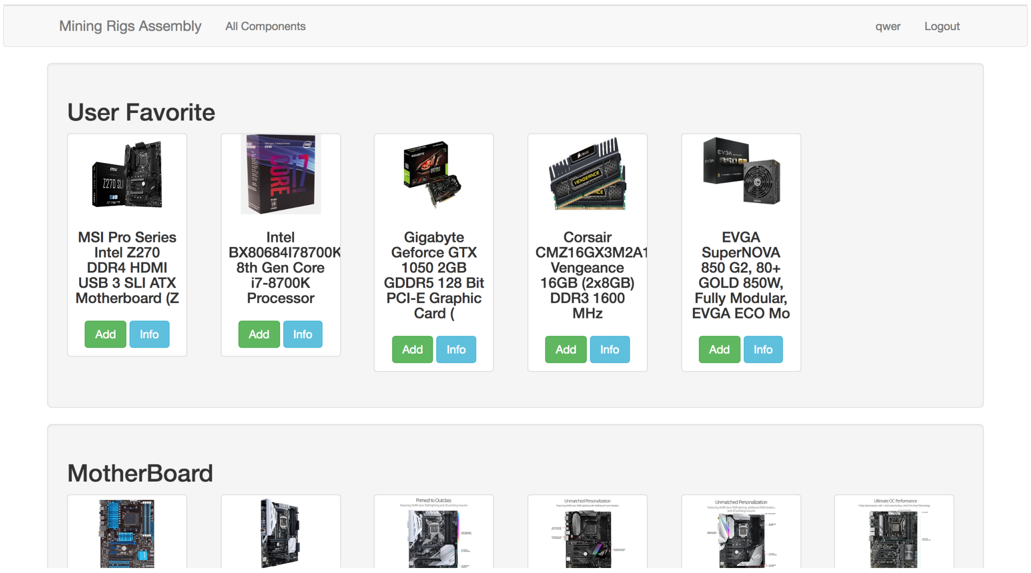
  AND (includes.cpu_name='$name' OR includes.gpu_name='$name' OR

  includes.ram_name='$name' OR includes.psu_name='$name' OR includes.mb_name='$name')) t1;"
- For each category what component a specific user favor most?
  "SELECT * FROM components WHERE name IN (SELECT name FROM (

  (SELECT name, MAX(c1) FROM (SELECT mb_name as name, COUNT(mb_name) AS c1 FROM creates,includes WHERE creates.setID=includes.setID AND creates.email=$email GROUP BY mb_name ORDER BY c1 DESC)t1)

  UNION





User Favorite

MSI Pro Series Intel Z270 DDR4 HDMI USB 3 SLI ATX Motherboard (Z
Add Info

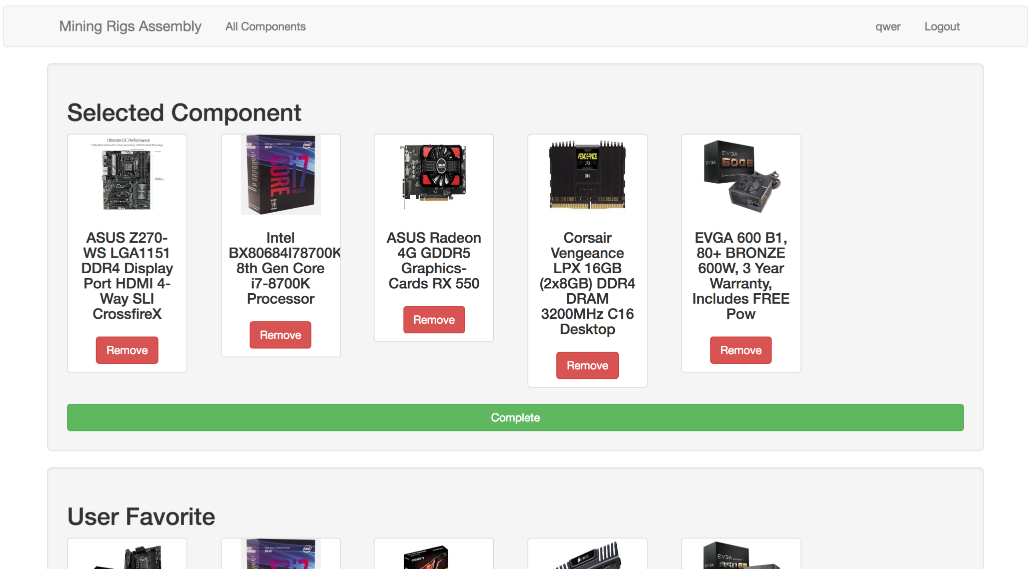
Intel BX80684I78700K 8th Gen Core i7-8700K Processor
Add Info

Gigabyte Geforce GTX 1050 2GB GDDR5 128 Bit PCI-E Graphic Card (
Add Info

Corsair CMZ16GX3M2A1 Vengeance 16GB (2x8GB) DDR3 1600 MHz
Add Info

EVGA SuperNOVA 850 G2, 80+ GOLD 850W, Fully Modular, EVGA ECO Mo
Add Info

MotherBoard



Selected Component

ASUS Z270-WS LGA1151 DDR4 Display Port HDMI 4-Way SLI CrossfireX
Remove

Intel BX80684I78700K 8th Gen Core i7-8700K Processor
Remove

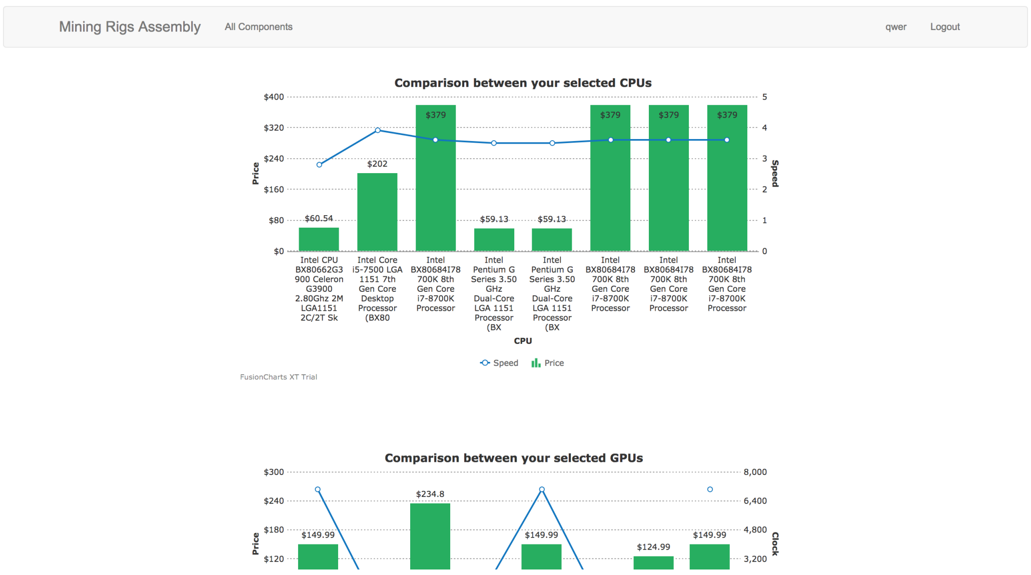
ASUS Radeon 4G GDDR5 Graphics-Cards RX 550
Remove

Corsair Vengeance LPX 16GB (2x8GB) DDR4 DRAM 3200MHz C16 Desktop
Remove

EVGA 600 B1, 80+ BRONZE 600W, 3 Year Warranty, Includes FREE Pow
Remove

Complete

User Favorite



9 Advanced Functions

1. **Calculate payback period of the mining rig investment**
   This function is the core of this application. It allows the users to plan their investments without having to monitor every little aspect of the market at all time. It is especially useful, and convenient to have in the current situation because of how rapidly the prices are changing. This function is advanced because it requires additional data from GPU such as hash rate and power consumption. Also, it update ETH price and component price in the database automatically and send email notification to user if they subscribe the set they created.

2. **Comparisons between different parts/setups with visualization**
   This can help user easier to understand the trade-off of different setups and choose the best setup based on the comparison. This function is advanced because it needs to summarize data into useful information such as total price of the setup and the payback period for each user setups for comparison.

10 Technical Challenge

When we implemented the first advanced function, we need to update all prices for components. And we found out the API that was used to obtain price blocked our requests as we requested too many times. we solved this problem by using sleep() function in Php.

11 Development Plan

Generally, everything went well according to the development plan, but some ideas of the project have been replaced for more practical reasons. For example, we started to build this website for bit-coin users. Nevertheless, we found that there are few websites for ETH investors, so we switched to focus on a more specific market. Whats more, when we need to realize the advanced functions, we found that more data should be stored in the database, so we add more attributes and entities to the database and it is more complex than the draft.

12 Labor Division

1. **Ho Yin Au**

   - Use Amazon Product Advertising API to crawl data.
   - Advanced query 2.
   - Access to Amazon API part and calculate total price part in automatic update price function.

- Display components part, user favorite part and selected components part in *add_setup.php*, *submit_setup.php*.
- Manage that data modification in database from website queries works according to the functionality.

2. **Jiajun Chen**

- Implement the basic outlook of website.
- Implement the *home.php* and *components.php*.
- Implement the periodical run function in advanced function1.
- Midterm presentation and demo video making.
- Arrange the team work and coordinate the meeting.

3. **Xiao Tang**

- Database initial setup.
- Registration and *login* page implementations and related database manipulations.
- Using different APIs to update ethereum price and mining difficulty which are used in.
- Advanced function 1 calculations.

4. **Zhenye Na**

- Backup codes powering final project website to Github periodically.
- Implemented Advanced Function 2 in php with FushionCharts.
- Implemented *info.php* and completed similar items comparison.
- Finished Project Introduction Website.
- Formated Final Report.