

James Ching-Cheong Lee



✉ machingclee@gmail.com

📁 <https://gitlab.com/machingclee>

🌐 [linkedin.com/in/chingclee](https://www.linkedin.com/in/chingclee)

Working Experience

- 01/2018 – 12/2018 **2D Game Artist**, 深圳疯点子科技有限公司
- Construct 3d accessory, create environment art and character design
- 03/2015 – 10/2015 **Research Assistant**, Department of Mathematics, Hong Kong Baptist University
- 09/2014 – 02/2015 **Research Assistant**, Department of Mathematics, Hong Kong University of Science and Technology
- 09/2012 – 08/2014 **Teaching Assistant**, Department of Mathematics, Hong Kong University of Science and Technology.

Education

- 03/2019 – 06/2019 **MicroMaster in A.I. & Programming**, Tecky Academy, Hong Kong
- 09/2012 – 08/2014 **M.Phil. in Mathematics**, The Hong Kong University of Science and Technology, Hong Kong.
- 09/2009 – 06/2012 **B.Sc. in Mathematics (First Class Honor)**, The Hong Kong University of Science and Technology, Hong Kong.

Skills

Web development

| | | |
|---------------|------------|---------------------------|
| TypeScript | React.js | Web Scraping |
| Javascript | PostgreSQL | Amazon Web Services (AWS) |
| Git | CSS | Redux |
| Express.js | Socket.io | Single Page Application |
| ElasticSearch | Solidity | |

Art and Design

| | | |
|-----------|----------|---------|
| Photoshop | Indesign | 3ds Max |
|-----------|----------|---------|

Expected Salary



Reference

Alex Lau, Lead Instructor
Tecky Academy Limited



Gordon Lau, Lead Instructor
Tecky Academy Limited



Project 3: Stand Overflow (SO)

This project is already deployed, you are welcome to our website:

<https://standoverflow.com/>

Features

SO provides a platform for “background checking” regarding to the stand of a company towards controversial issues in Hong Kong. The nature of issues can be diversified, they can be political, environmental and educational.

Structure of This Application

In frontend, we use:

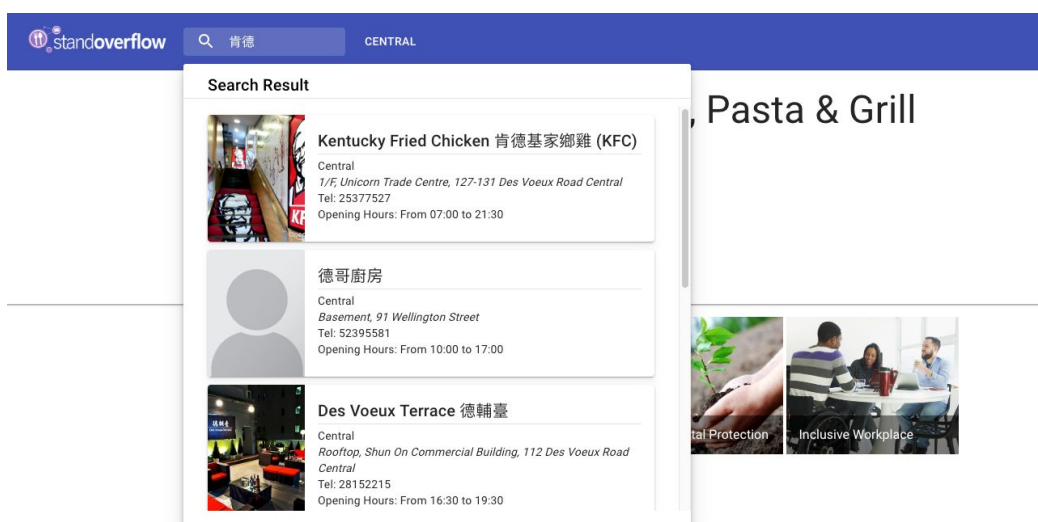
- React.js
- Redux

while in backend we use:

- Node.js + Express Framework
- PostgreSQL
- Elasticsearch
- Ethereum Blockchain Network

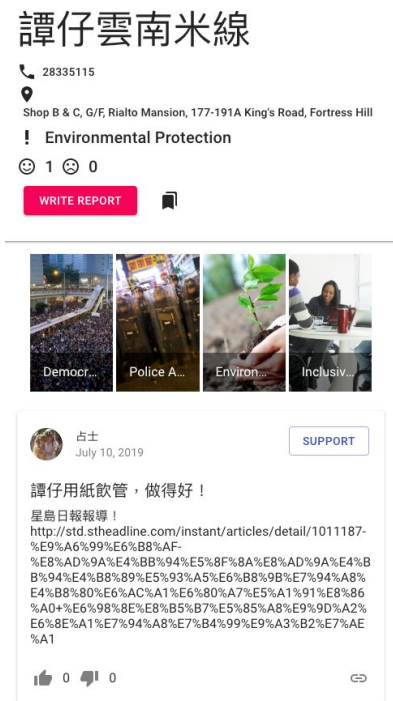
Usage

Users can search for the food store for which they want to have a review:



Users can upload a review of a company on a specific issue as in the picture on the right:

Each review is also sent to an Ethereum network and identified with an “id” called **transaction hash**, by doing so no one can modify the content any more.

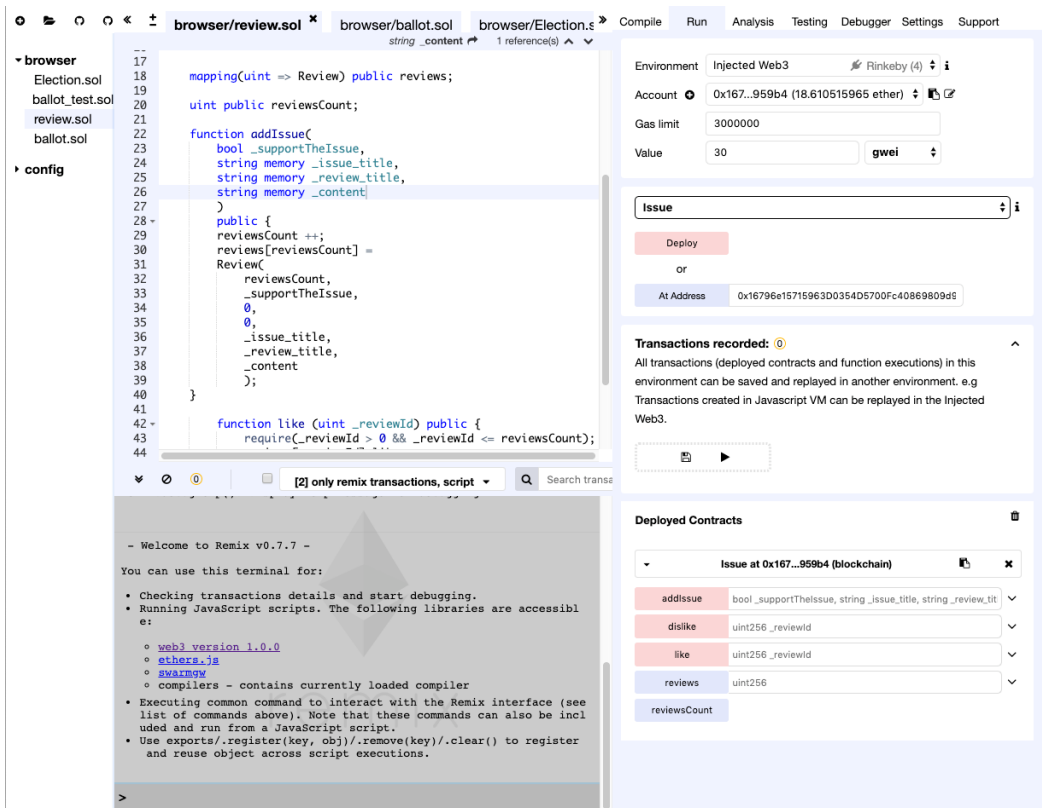


Implementation Detail of Blockchain Framework

How it Works

For the sake of better user experience, we will let users save their record to the blockchain through **our** account.

Below is a smart contract written in solidity language. They are tested and deployed to the blockchain network through Remix - Ethereum IDE. In SO we transfer data (instead of money) to this contract for record.



After a contract is deployed, we can connect to our Ethereum network account in the backend by using our private key:

```

constructor(private knex: Knex, private web3: Web3) {
  this.knex;
  this.privateKey =
    "C62D3D048C3F...";

  this.account = this.web3.eth.accounts.privateKeyToAccount(
    "0x" + this.privateKey
  );
  this.web3.eth.accounts.wallet.add(this.account);
  this.web3.eth.defaultAccount = this.account.address;
  this.contractAddress = "0xc775d626004936c70209657d6e602Fd892Ea2877";
  this.reviewsContract = new this.web3.eth.Contract(-
    this.contractAddress
  );
}

addReview = async (
  supportTheIssue: boolean,
  issueTitle: string,
  reviewTitle: string,
  content: string
): Promise<string> =>
  new Promise((resolve, reject) => {
    this.reviewsContract.methods
      .addIssue(supportTheIssue, issueTitle, reviewTitle, content)
      .send(
        {
          from: this.web3.defaultAccount,
          gasLimit: this.web3.utils.toHex(160000),
          gasPrice: this.web3.utils.toHex(this.web3.utils.toWei("30", "gwei"))
        },
        (err: string, hash: string) => {
          resolve(hash);
        }
      );
  });
};

```

then users of SO can use our API to record their review to the blockchain network, and that's all we have done for the blockchain part.

Project 2: BridgeTama (橋魂)

This project is already deployed, you are welcome to the website:

<http://www.artxiv.space/user/login>

Features

This website is dedicated to building a platform for players to enjoy the card game “Contact Bridge”, which grows into quite a number of variant forms and the gameplay in this website is just among one of them.

Structure of This Application

For the frontend:

- Pure HTML + JavaScript file
- Socket.io

while for the backend:

- Express.js
- Knex
- PostgreSQL
- Socket.io

Usage

On login, users can click the button to queue for the next game:

You may click the following for queuing!

BridgeTama Start!

Number of People Queuing: **0**

You may click the following for queuing!

BridgeTama Start!

Number of People Queuing: **1**

The number will automatically increase to 2, 3 when there are other players queuing for the game. Number of queuing players is taken from the table in our database:

```
pppoker=# select * from queuing
;
 id | user_id |          created_at          |          updated_at          |
-----+-----+-----+-----+
 26 |      22 | 2019-07-17 04:38:03.270451+00 | 2019-07-17 04:38:03.270451+00 |
(1 row)
```

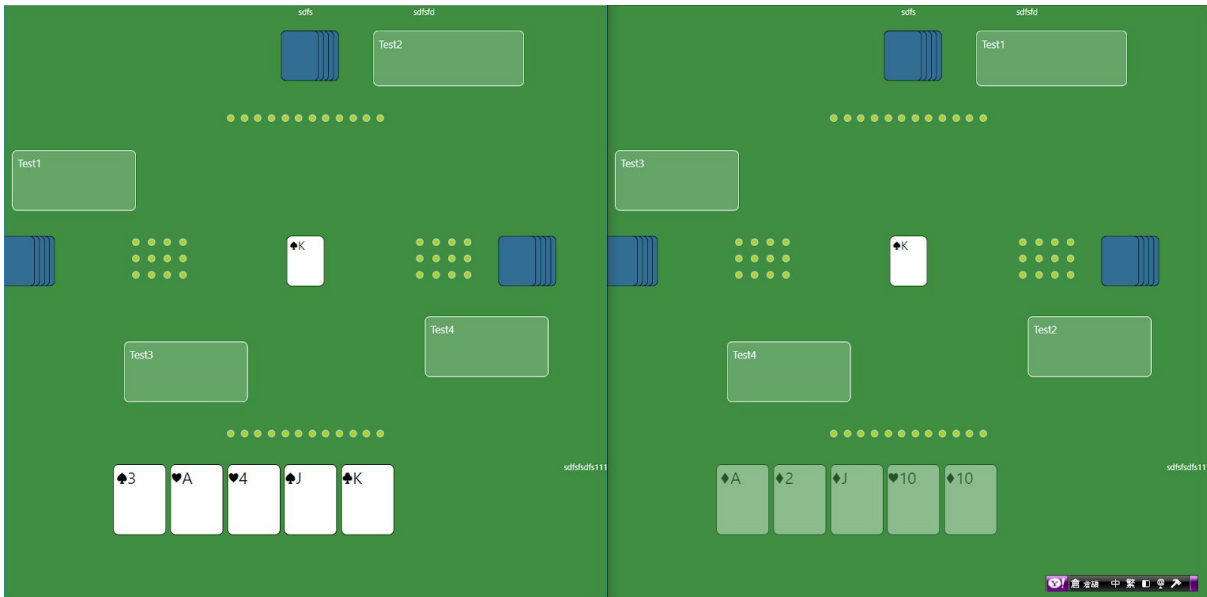
Once the game is started, a newly permuted deck of 52 cards will be saved to our database, and the whole game of this room of players will depend on this deck:

```

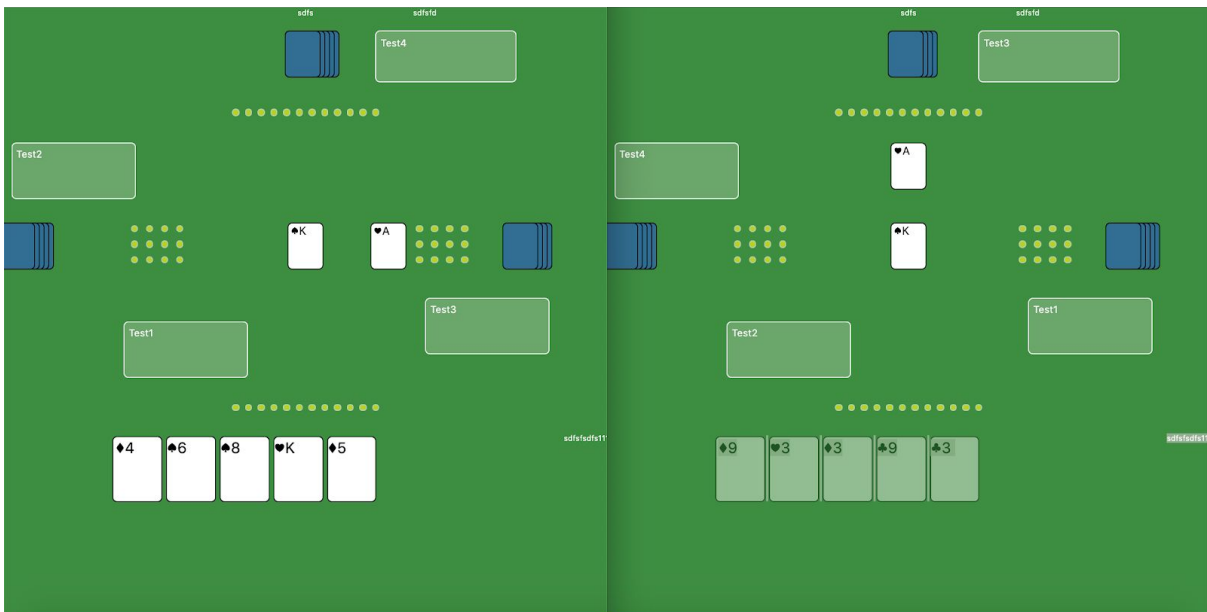
1032 | Test2_Test1_Test3_Test4 | ♠A | 2019-07-17 04:54:07.692924+00 | 2019-07-17 04:54:07.692924+00 |
1033 | Test2_Test1_Test3_Test4 | ♠Q | 2019-07-17 04:54:07.694407+00 | 2019-07-17 04:54:07.694407+00 |
1034 | Test2_Test1_Test3_Test4 | ♠5 | 2019-07-17 04:54:07.696442+00 | 2019-07-17 04:54:07.696442+00 |
1035 | Test2_Test1_Test3_Test4 | ♠5 | 2019-07-17 04:54:07.697865+00 | 2019-07-17 04:54:07.697865+00 |
1036 | Test2_Test1_Test3_Test4 | ♠4 | 2019-07-17 04:54:07.699256+00 | 2019-07-17 04:54:07.699256+00 |
1037 | Test2_Test1_Test3_Test4 | ♠2 | 2019-07-17 04:54:07.700618+00 | 2019-07-17 04:54:07.700618+00 |
1038 | Test2_Test1_Test3_Test4 | ♥K | 2019-07-17 04:54:07.70203+00 | 2019-07-17 04:54:07.70203+00 |
1039 | Test2_Test1_Test3_Test4 | ♠9 | 2019-07-17 04:54:07.703422+00 | 2019-07-17 04:54:07.703422+00 |
1040 | Test2_Test1_Test3_Test4 | ♠8 | 2019-07-17 04:54:07.704855+00 | 2019-07-17 04:54:07.704855+00 |
(52 rows)

```

The following is the scenario of the first turn of the 5th round. Player3 is about to choose ♥A:



The following shows the view of player Test1 and Test2 after ♥A is chosen:



The game is supposed to continue until the 12th round. Winner of each round depends on a complicated rule, and each round will start with the winner of the previous round.

Project 1: Artxiv

Why Artxiv?

In mainland China there have been web applications like 米画师 and 画萌 where artists can upload their portfolio, and project-initiator can recruit artist there on a project basis. Unfortunately they have no counterpart in Hong Kong, my first project on Tecky Academy is dedicated to building such an application for artists in Hong Kong.

Due to limited time and ability at that time, the project is incomplete.

Structure of This Application

In frontend:

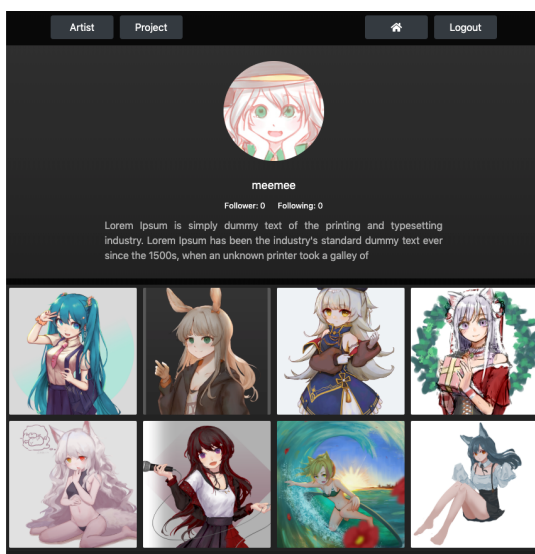
- HTML with JavaScript files

In backend:

- JSON file as a database
- Node.js + Express Framework

Usage

Artist can upload their arts which are to be shown below:



Users can click the toggle “Manage Artwork” to reorder their work as follows:

