The Hong Kong Polytechnic University
Department of Computing

COMP4913 Capstone Project
Final Report

Code Board for Programming Projection

Programme-Stream Code:	61431-FCS
Supervisor:	Dr PEI Yu Max
Co-Examiner:	Dr WU Xiaoming
2nd Assessor:	Dr WANG Dan
Submission Date:	May 23, 2020

Abstract

When teachers want to teach students how to program in class, the traditional way is using the projector to project the programming process. However, the effect of the projection by the projector is limited by many factors that cause many problems. To solve the problems of programming projection in the classroom, this project develops an online code board to provide a brand-new programming projection way for teachers. The teachers can prepare their codes on the code board before class, and program in the class. It is as same as what they do when using the projector. Therefore, they do not need to learn many new technologies. After the teacher starts the projection, the share code and a QR code will be generated. Students can use the code or scan the QR code to watch the projection of the teacher's programming on the code board on their computer or smartphone, instead of on the projection screen. In this way, the students do not need to worry about they cannot see the projection screen clearly.

Table of Contents

List of tables and figures

1 Introduction

This report will firstly state the problem that the project needs to solve. And claim the objectives and outcome of the project. After that, there is an introduction to the background of the Node.js platform and WebSocket technology used in the implementation. Then, the report will describe the design of the project and explain the reasons for the design. Some alternative technologies and designs are mentioned and compared with the final design of the project. After that, the implementation of the application is explained in detail. Finally, an evaluation is introduced, and a conclusion is made.

1.1 Problem Statement

Using a projector is the traditional way to project the programming process in class. However, it has significant limitations and problems.

The first limitation is the size of the classroom and students' vision. The programming process needs the projection screen to display more content than normal PPT. Therefore, it is difficult for the students sitting in the back of a large classroom to watch the projection clearly, especially for the students with myopia. It is unsuitable to use a projector to project the programming process in large classrooms.

The second limitation is the font size. The programmers usually prefer to use small font sizes during programming to make the screen display more codes. It means the large font size will increase the difficulty of the programming for teachers. However, decreasing the font size will make it hard for students to see the codes on the projection screen clearly. Therefore, it is really hard to find the balance of the comfortable for the teachers and students on the font-size choosing during using the projector to project the programming process.

The third limitation is the programming speed. The projector can only project limited content. Different from project a PPT, if the students fail to follow the teacher’s programming, it is impossible for them to view the previous codes by themselves to keep up to the pace. In most cases, teachers will only release the codes after the class. One traditional solution is suspending the programming on the milestones and asking students if they have any questions. However, it may slow down the course progress, especially during the programming of a multifunction program.

1.2 Objectives and Outcome

In order to solve the above problems, in this project, an online code board will be developed as a programming projection platform. The code board should have an account system for users to sign up and log in, a project management system for users to manage different projects for different classes, an online code editor with syntax highlighting for users to make their programming projection, account management system to handle different kinds of users’ behaviors and allow users to watch the programming projection without login.

2 Background

This section introduces the background knowledge of the platform, programming language, and technique this project uses. An introduction of Node.js and JavaScript will be provided. The concept of WebSocket will be discussed.

2.1 Node.js and JavaScript

As a running platform for backend JavaScript, Node.js retains those familiar interfaces in front-end browser JavaScript, without rewriting any features of the language itself, and is still based on the scope and prototype chain. The difference is that it migrates the ideas widely used in the front-end to the back-end [1]. The characteristics of Node.js compared to other languages are as follows:

1. Asynchronous I/O

   In Node.js, most operations are called asynchronously. In Node.js, we can naturally perform parallel I/O operations from the language level. There is no need to wait for the end of a previous I/O call between each call. The programming model can greatly improve efficiency [2].

   For example, when taking two file reading tasks at the same time, asynchronous I/O depends on the slowest time to read the file, while synchronous I/O is the sum of the time of the two tasks. The advantages of asynchronous here are obvious.

2. Event

With the advent of the Web 2.0 era, JavaScript has assumed more responsibilities on the front end, and events have been widely used. Node.js is not greatly influenced by Java but introduces widely used and mature events in the front-end browser to the back-end and cooperates with asynchronous I/O to expose event points to business logic [1]. Event programming has the advantages of being lightweight, loosely coupled, and focusing only on transaction points [3].

3. Callback

Compared with other web back-end programming languages, in addition to asynchronous and events, Node.js callback functions are a major feature [1]. Callback functions are also the best way to accept asynchronous calls to return data [4].

4. Single thread

A major feature of the JavaScript language is single threading, which means that you can only do one thing at a time. JavaScript's single thread is related to its purpose. As a browser scripting language, the main uses of JavaScript are to interact with users and manipulate the DOM. This determines that it can only be single-threaded, otherwise it will bring complex synchronization problems [5]. For example, suppose JavaScript has two threads at the same time, one thread adds content to a certain DOM node.js, and the other thread deletes this node.js. At this time, which thread should the browser take? So, to avoid complexity, JavaScript has been single-threaded since its inception, which has become a core feature of the language. Node.js maintains the single-threaded nature of JavaScript in the browser [6]. And in Node.js, JavaScript and other threads cannot share any state. The biggest advantage of single threading is that you don't need to worry about the synchronization of the state like multi-threaded programming. There is no deadlock here, and there is no performance overhead caused by thread context exchange. HTML5 customizes the standard of Web Workers. Web Workers can create worker threads to perform calculations to solve the problem of large JavaScript calculations blocking UI rendering. In order not to block the main thread, the worker thread passes the running result by message passing, which also makes the worker thread unable to access the UI in the main thread. Node.js adopts the same idea as Web Workers to solve the problem of large computation in a single thread - the child process [1]. The emergence of child processes means that Node.js can calmly cope with the issues of single-threaded robustness and inability to utilize multi-core CPUs. By distributing calculations to various sub-processes, a large number of calculations can be broken down, and then the results are passed through event messages between processes, which can keep the application model simple and low dependent. Through the management method of Master-Worker, Node.js can well manage each worker process to achieve higher robustness [7].

2.2 WebSocket

WebSocket is a TCP-based application layer protocol used to implement two-way communication in applications with C/S architecture [8]. Although the WebSocket protocol uses the HTTP protocol when establishing a connection, this does not mean that the WebSocket protocol is implemented based on the HTTP protocol.

There are 2 essential differences between WebSocket protocol and HTTP protocol:

1. Different communication methods: WebSocket is a two-way communication mode. The client and the server only use the "request-response" mode of the HTTP protocol during the handshake phase, and once the connection is established, the communication uses the two-way mode [9]. Both the client and server Data can be sent to the other party at any time; the HTTP protocol uses the "request-response" mode for communication from beginning to end. Because of this, the communication efficiency of the HTTP protocol is not as high as WebSocket [10].
2. Different protocol formats: The HTTP protocol is bloated, while the WebSocket protocol is lightweight. For the HTTP protocol, a data packet is a complete message. For WebSocket, the minimum unit for a client to communicate with a server is a frame, which consists of one or more frames to form a complete message [11]. That is: The client cuts the message into multiple frames and sends them to the server. The server receives the message frame and reassembles the associated frames into a complete message.

In a word, WebSocket has following advantages:

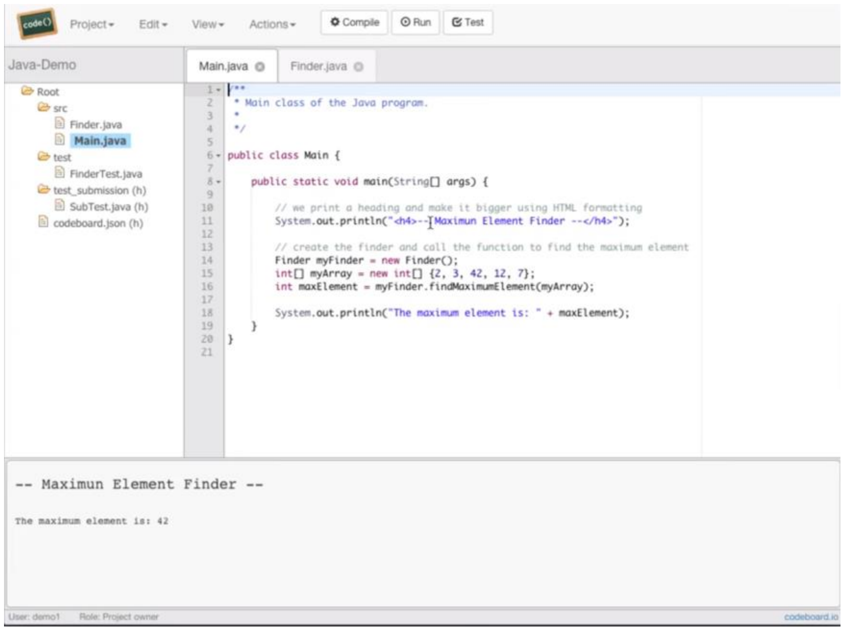
1. Push Function: WebSocket supports server-to-client push function. The server can send data directly without waiting for a client request [11].
2. Traffic Reduction: WebSocket supports two-way communication and has better real-time performance [9].

3. Less Control Overhead: After the connection is created, when the WebSocket client and server exchange data, the packet header controlled by the protocol is smaller [10].

3. Related Work

3.1 Codeboard.io

The first related work is a coding platform called codeboard.io. It is on https://codeboard.io/. Users can create new projects on codeboard.io, program, compile and run their codes on codeboard.io. They can share their codes with others by share links. According to its home page description, it is a web-based IDE for teachers to teach programming in the classroom. Since it focuses on the IDE functions, the code board UI of it is very complete and attractive. Figure 1 displays the screenshot of its IDE UI.



```
/**
 * Main class of the Java program.
 *
 */

public class Main {

    public static void main(String[] args) {

        // we print a heading and make it bigger using HTML formatting
        System.out.println("<h4>--Maximun Element Finder --</h4>");

        // create the finder and call the function to find the maximum element
        Finder myFinder = new Finder();
        int[] myArray = new int[] {2, 3, 42, 12, 7};
        int maxElement = myFinder.findMaximumElement(myArray);

        System.out.println("The maximum element is: " + maxElement);
    }
}
```

```
-- Maximun Element Finder --

The maximum element is: 42
```

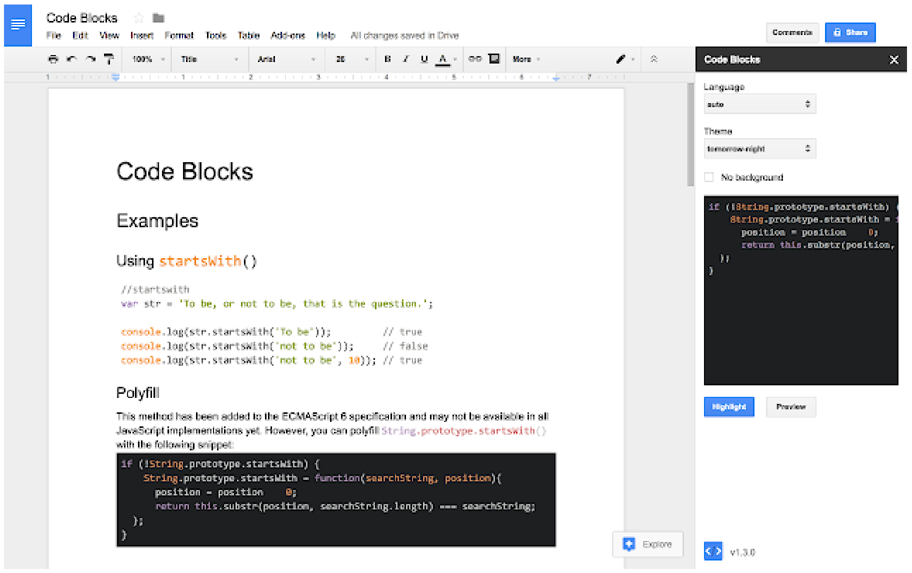
Figure 1: Codeboard IDE UI

Because of its lovely UI design, the projecting page UI design of this project also learn from its design, especially the left directory part of it, which is introduced in the implementation part of this report.

However, the codeboard.io doesn't provide the programming projecting function, which means it cannot solve the problem this report discussed. The share code function of it only supports users to read others' codes statically, which means students cannot watch the instant change of the code if the teacher is programming. But the share link mechanism of it is also valuable for this project to learn.

3.2 Google Doc with Code Blocks Add-on

As the most famous online instant collaboration tool, Google Doc provides fluent and professional collaboration experience to users. The Code Blocks Add-on is a Google Chrome plugin. It can provide syntax highlighting for Google Doc, as Figure 2 presents.



Code Blocks

Examples

Using `startsWith()`

```
//startswith
var str = 'To be, or not to be, that is the question.';

console.log(str.startsWith('To be'));         // true
console.log(str.startsWith('not to be'));     // false
console.log(str.startsWith('not to be', 10)); // true
```

Polyfill

This method has been added to the ECMAScript 6 specification and may not be available in all JavaScript implementations yet. However, you can polyfill `String.prototype.startsWith()` with the following snippet:

```
if (!String.prototype.startsWith) {
    String.prototype.startsWith = function(searchString, position){
      position = position    0;
      return this.substr(position, searchString.length) === searchString;
  };
}
```

```
if (!String.prototype.startsWith)
    String.prototype.startsWith =
      position = position    0;
      return this.substr(position,
  });
}
```

Figure 2: Google Doc with Code Blocks Add-on

A read-only Google Doc with Code Blocks Add-on seems can solve the problem discussed in this report. However, the problem of Google Doc is it can only open one document in each link. Although teachers can share a whole folder, it is hard for students to follow teachers' programming steps by clicking on files in folders and open many new tabs to watch programming. It is hard to use in programming projection for the classroom. It is because Google Doc was designed for collaboration instead of class projects. There needs to be a professional platform to implement the programming projection instead of finding a barely usable alternative.

4 Project Design

According to the preview sections introduce, the technology stack of the website is presented in Figure 3:

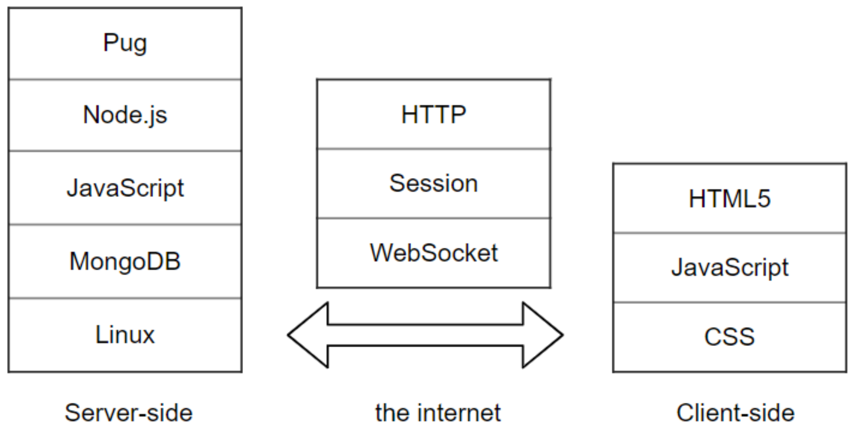


Figure 3: Technology Stack

There are alternative technology stacks that can be used for this project. The author researched the alternative web programming language PHP and Java, the alternative database MySQL and Cassandra, and the alternative push technology long polling.

- PHP and MySQL: The framework of PHP and MySQL is the most stable and mature web development framework. PHP is easy to program and can be mixed in HTML codes. As one of the oldest web programming language, PHP has a powerful library community can help programmers to implement functions with low workloads. However, the main problem is SQL database needs more system resources than the NoSQL database. [12] Since this is a small non-profit project only for education. The economics of the project are also crucial. Meanwhile, the

PHP code can be mixed with HTML code is also a disadvantage for program maintenance in the future. To solve this problem, the project needs to be designed carefully and has complete documentation. [13] It is inefficient for a one-developer small project. Using Node.js and MongoDB is much better than this framework.

- Java: Spring Boot is a mature platform for Java to implement web applications. Java also has a rich library community to support every kind of web development. However, compared with Node.js, the codes of Java web applications are more redundant, which means it will be more inefficient for implementation. And the maintenance of Java application will be more difficult than Node.js application. [14]
- Cassandra: As a NoSQL database developed by Facebook, Cassandra has professional stability and reliability. It does great jobs in handling tens of millions of data records storage. However, MongoDB is more flexible than Cassandra. There is no requirement of schema definition on MongoDB, which means its extensibility is strengthened than Cassandra. [15] The extensibility is essential for a small-scale project that requires a rapid development. Therefore, this project uses MongoDB instead of Cassandra.
- Long polling: Long polling is implemented after XMLHttpRequest, which is almost universally supported by the device. However, Long polling takes up a lot of server resources. Depending on the implementation of the server, a client's confirmation of the message may also cause the other client to not receive the expected message at all, because the server may mistakenly believe that the client has received the data it

expects. [9] Therefore, WebSocket is more reliable than long polling technology and becomes the instant communication technology used in this project.

The system architecture diagram is presented in Figure 4:

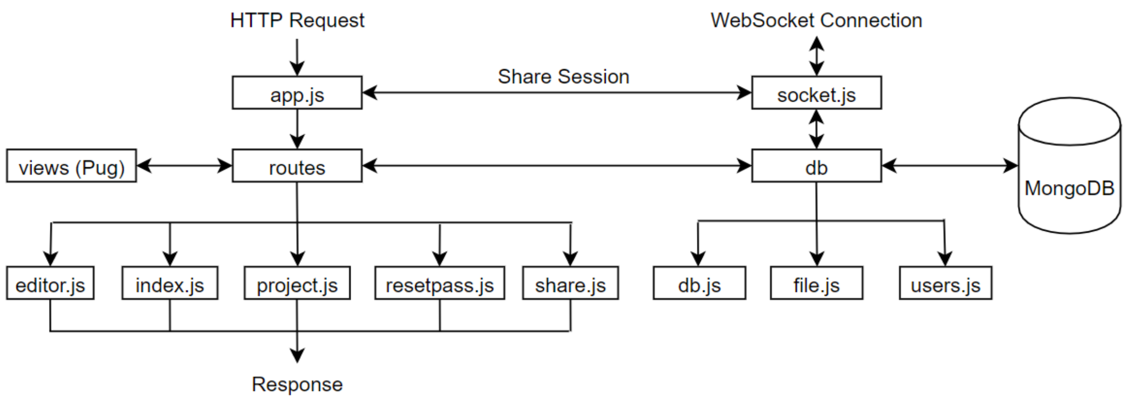


Figure 4: System Architecture Diagram

There will be 2 kinds of connections from users. The HTTP requests will be handled by the app component. The app will create and manage the session for each user. It will share the data of sessions with the socket component to help authorize the WebSocket connections. After managing the session, the app will deliver the HTTP requests to different routes according to their requirements. The routes will handle these requests and use the Pug engine to render the response pages to response to the requests. The WebSocket Connections will be handled and controlled by the socket component. Both the routes and socket need to dock with the db component to communicate with the database.

There is an alternative design of the system architecture, which is presented in Figure 5:

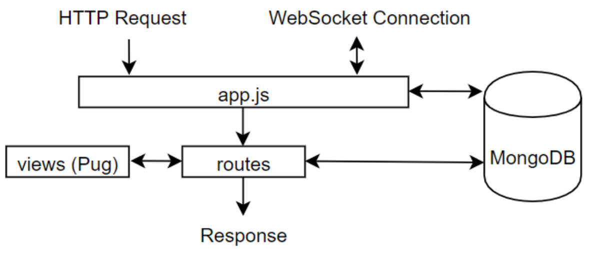


Figure 5: Alternative System Architecture Design

This is a popular Node.js architecture design for a small web application project. The app component handles all kinds of connections. Both app and routes components can communication with the database directly. This design will increase the efficiency and speed of development. However, it will increase the difficulty of maintenance of the code. The code will also be hard to understand.

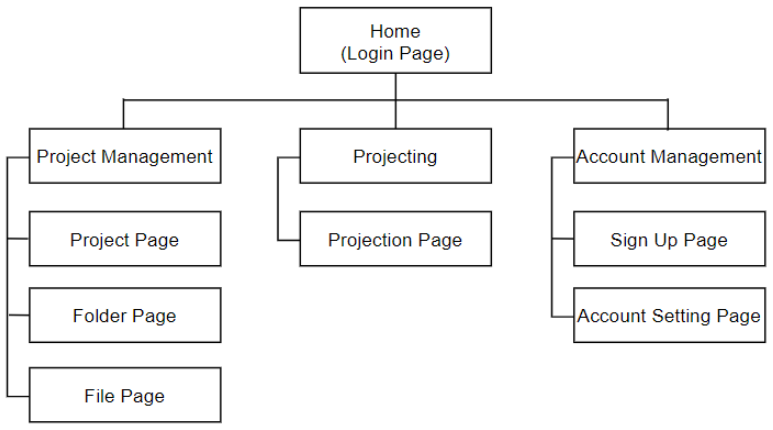


Figure 6: Website Structure Diagram

The website structure diagram is presented in Figure 6. The home page is also the login page. Users can start to watch an existed projection from the home page as well. The pages are divided into 3 kinds. The account management pages are for teachers to sign up and set their accounts. The project management pages are for teachers to create and manage projects for projecting before class. In the class, after starting projecting, the projection will be hosted and watched on the projection page.

There is an alternative design of the website structure, which is presented in Figure 7:

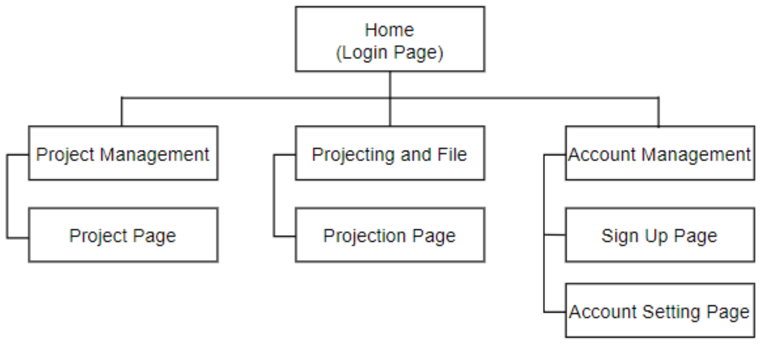


Figure 7: Alternative Website Structure Design

This design is the first design. The page of projects is also a file and folder management page. Users can make the creating, editing, and deleting operations on this page just as same as what they do on folder and file page. In this way, the workload of development can be reduced. The learning cost of users will also be condensed.

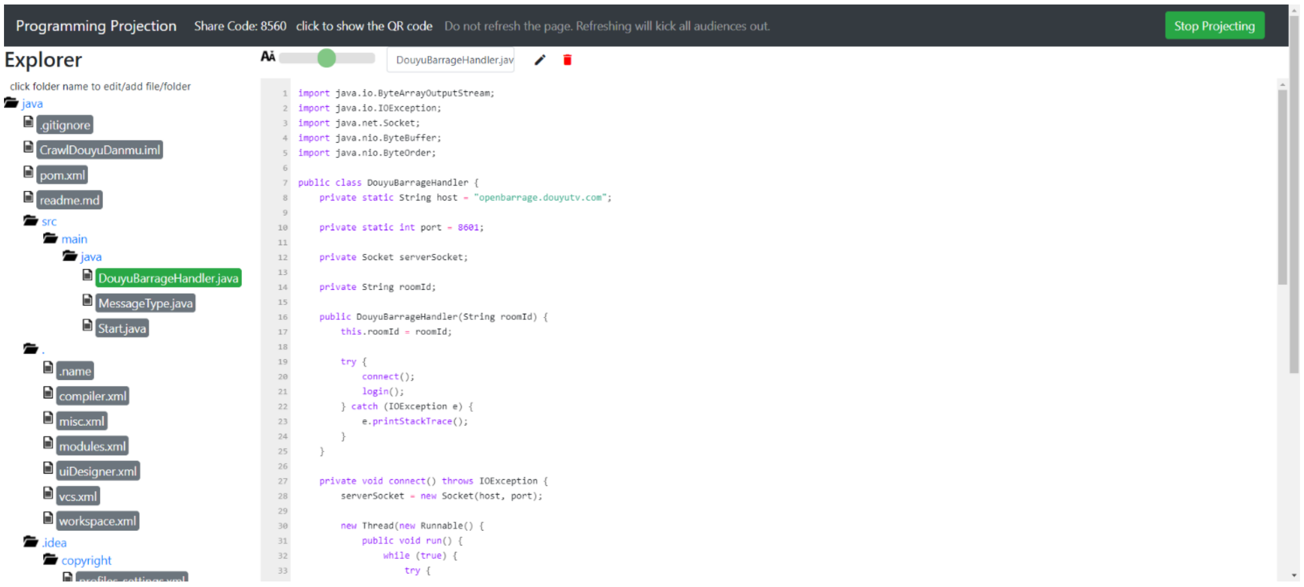
So, why is this design abandoned? Firstly, the projection page uses the WebSocket technique to communicate the data of code files and folders with the server. However, compared with a single HTTP request, the overhead of holding a WebSocket connection is larger. During the project management, users do not always need to open files. A folder page that only use single HTTP request to receive data of folders can reduce the overhead of the website. It is economical for a small project and can reduce the communication pressure on the server.

5 Implementation

Since there is no related work use WebSocket to implement instant code programming sharing, which is the main problem and challenge of this project to be implemented. This section will first describe the implementation of instant programming sharing using WebSocket. After that, there are some other explanations of the implements of other functions of this system.

5.1 Instant programming sharing implementation

The application uses the Socket.io to set up the WebSocket connections between clients and servers. Socket.io is a library that provides cross-platform real-time communication for real-time applications [16]. Socket.io is designed to "make real-time applications possible on every browser and mobile device, blurring the differences between the different



```java
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;

public class DouyuBarrageHandler {
    private static String host = "openbarrage.douyutv.com";

    private static int port = 8601;

    private Socket serverSocket;

    private String roomId;

    public DouyuBarrageHandler(String roomId) {
        this.roomId = roomId;

        try {
            connect();
            login();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private void connect() throws IOException {
        serverSocket = new Socket(host, port);

        new Thread(new Runnable() {
            public void run() {
                while (true) {
                    try {
```

Figure 8: Projecting Page UI

Figure 9 shows the beginning of WebSocket communication between the client-side and server-side. The following paragraphs will explain how the implementation will make this web page get full of the contents.

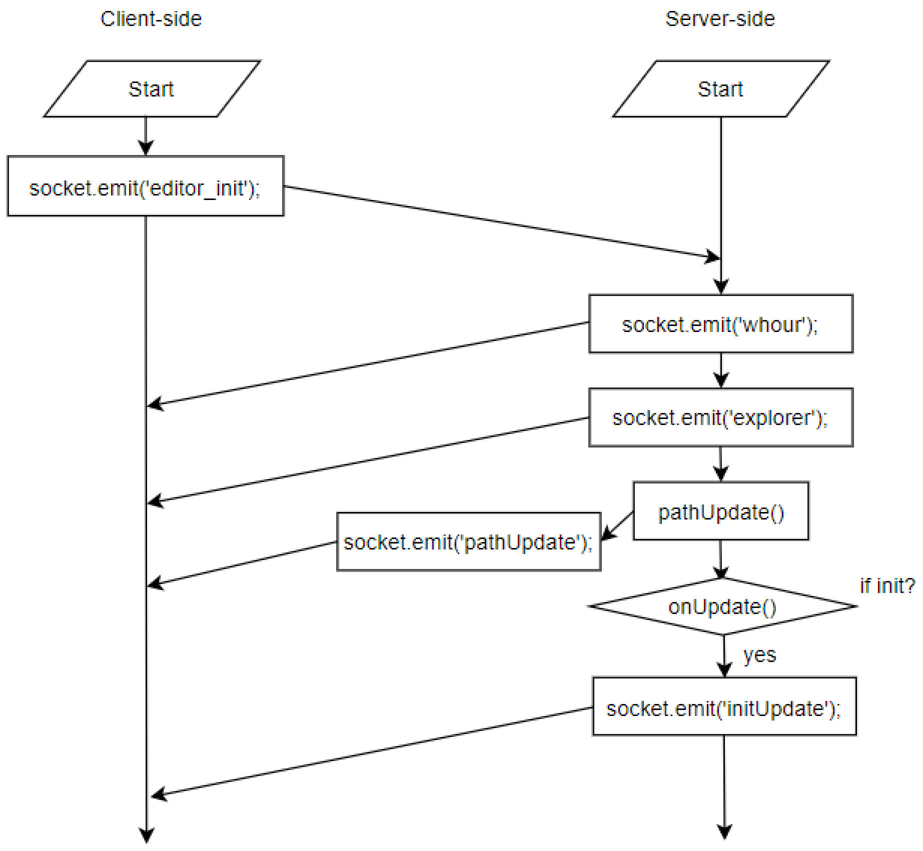


Figure 9: Beginning WebSocket Communication

The client-side will emit an "editor_init" message to the server-side. The server-side will first read the session data of the client from the sharing data by app.js to figure out the identification of the user. Then it will emit a "whour" message with the identification data to the client to make the browsers know it should display as a host or audience. Meanwhile, the server will read the directory of the project that opened by this projecting, then emit to the client as well. This message will bring the content to fill the left Explorer part of the UI.

The server will call a function pathUpdate() to emit the path of the file under projecting to the client with "pathUpdate" message. This message will make the file under projecting turn to green on the Explorer. The server will also call a function onUpdate() to read the content of the file under projecting and emit by message "initUpdate" to fill the right part of the UI.

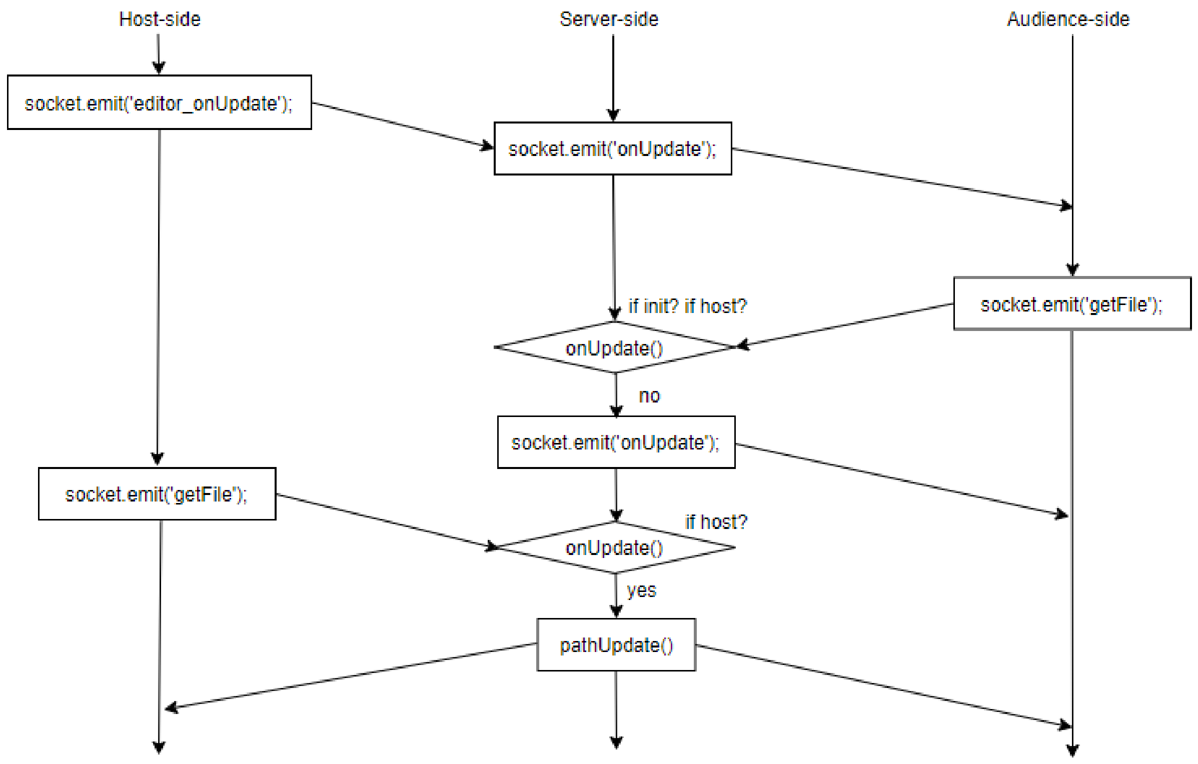


Figure 10: Instant Projecting Implement

How does instant projecting work? Figure 10 displays the process of the host edit the code and the host and audiences change the file selected. Once the host client makes any changes to his code, the change will be emitted to the server by "editor_onUpdate" message. Then the server will deliver this change to every audience in this projecting room. If an audience wants to change to view another file that is not under programming by the host, his request will be emitted by “getFile” message. And the server will judge if he is the host. Since he is not, the change will only be sent to the audience himself. But if a host wants to do the same operation, the server will not only send the change to the host, but also call the pathUpdate() function to deliver this change to all audiences in the room.

Figure 11 shows the process of host creates, edit, or delete a file or folder.

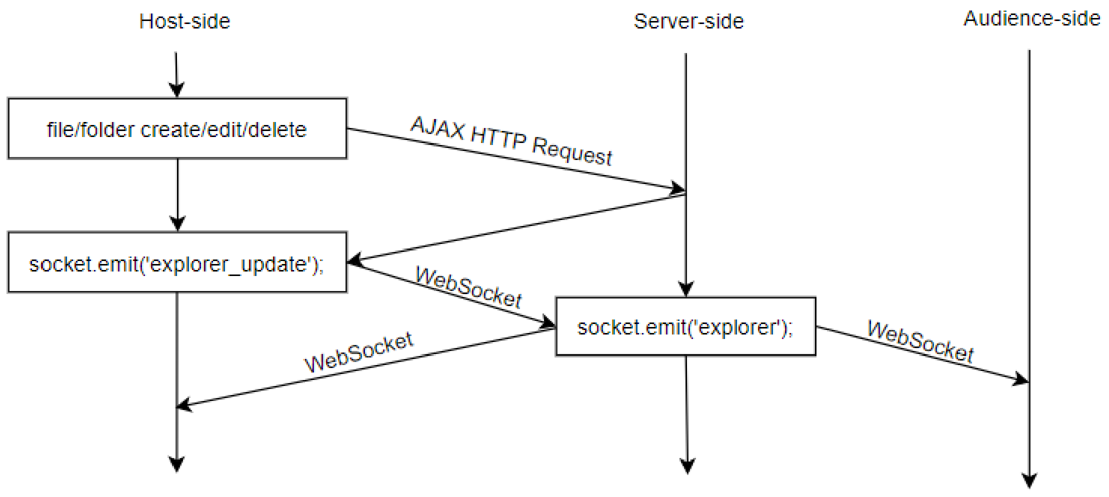


Figure 11: Create, Edit or Delete a File or Folder

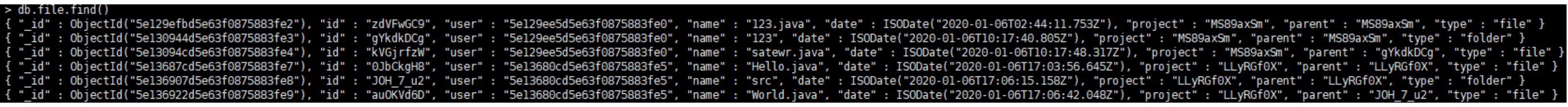
The changes of files and folders themselves, instead of codes, will not be submitted by WebSocket. They can be emitted by WebSocket, but there is not necessary to use it in this case. Since there is no requirement of instant for this function, compare with WebSocket, it is easier to implement this function by Ajax. Therefore, here the request will only be sent by a single HTTP request. After the server handled the request, it will respond to the result. The response will make the client emit a message to the server via WebSocket. Then the server will deliver this change to everyone in the room using WebSocket. This design reduces the complexity of the code. Enhanced the reliability of the program.

5.2 Code Reusability

The reusability of code is essential for a web application with many functions. The author tried to make the code as reusable as possible in this project. The way to make code reusability is to make duplicate codes into functions. Then call the functions in the places that use the codes of them. In the preview descriptions and figures, the functions and parts socket.emit('explorer'), pathUpdate(), onUpdate(), socket.emit('getFile') are all reusable and have been used in more than 2 places. Due to space limitations, this report cannot show all reusable functions in this project. The functions in db components are all reusable functions, which greatly reduces code redundancy. That is the reason why the system architecture design divides db component from the socket and routes components and requires all components to visit the database though db component.

5.3 Data Structure of File System

How to implement the file system is another essential problem of this project. In early-stage implement, the file system was implemented as a file storage system, which means all



```
> db.file.find()
{ "_id" : ObjectId("5e129efbd5e63f0875883fe2"), "id" : "zdVFwGC9", "user" : "5e129ee5d5e63f0875883fe0", "name" : "123.java", "date" : ISODate("2020-01-06T02:44:11.753Z"), "project" : "MS89axSm", "parent" : "MS89axSm", "type" : "file" }
{ "_id" : ObjectId("5e130944d5e63f0875883fe3"), "id" : "gYkdkDCg", "user" : "5e129ee5d5e63f0875883fe0", "name" : "123", "date" : ISODate("2020-01-06T10:17:40.805Z"), "project" : "MS89axSm", "parent" : "MS89axSm", "type" : "folder" }
{ "_id" : ObjectId("5e13094cd5e63f0875883fe4"), "id" : "kVGjrfzW", "user" : "5e129ee5d5e63f0875883fe0", "name" : "satewr.java", "date" : ISODate("2020-01-06T10:17:48.317Z"), "project" : "MS89axSm", "parent" : "gYkdkDCg", "type" : "file" }
{ "_id" : ObjectId("5e13687cd5e63f0875883fe7"), "id" : "0JbCkgH8", "user" : "5e13680cd5e63f0875883fe5", "name" : "Hello.java", "date" : ISODate("2020-01-06T17:03:56.645Z"), "project" : "LLyRGf0X", "parent" : "LLyRGf0X", "type" : "file" }
{ "_id" : ObjectId("5e136907d5e63f0875883fe8"), "id" : "JOH_7_u2", "user" : "5e13680cd5e63f0875883fe5", "name" : "src", "date" : ISODate("2020-01-06T17:06:15.158Z"), "project" : "LLyRGf0X", "parent" : "LLyRGf0X", "type" : "folder" }
{ "_id" : ObjectId("5e136922d5e63f0875883fe9"), "id" : "auOKVd6D", "user" : "5e13680cd5e63f0875883fe5", "name" : "World.java", "date" : ISODate("2020-01-06T17:06:42.048Z"), "project" : "LLyRGf0X", "parent" : "JOH_7_u2", "type" : "file" }
```

Figure 12: Data Structure of Files

storage system is too heavy for a small project server to handle. Meanwhile, it is hard to maintain this system. The best solution is using a database to store the directories of each file and folders, then store all files at the same directory on the server.

Figure 12 displays the data structure of files. The directory of the file is stored by 2 parameters, they are “project” and “parent”. This design is like the reverse linked list in C language, whose items are connected by pointers as Figure 13 presents.

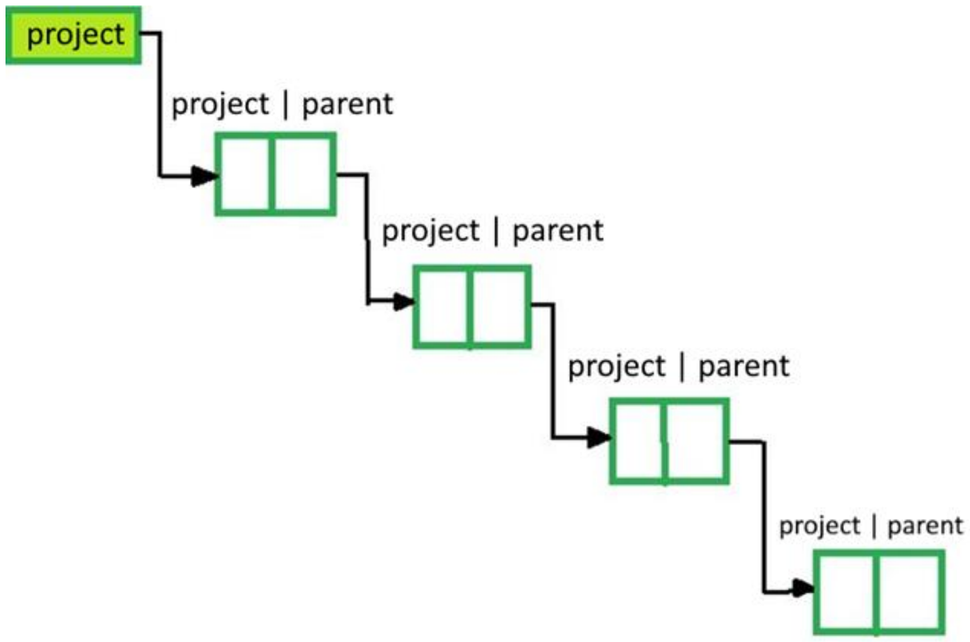
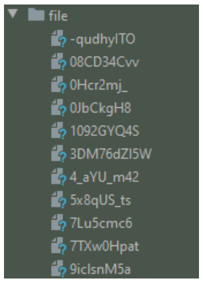


Figure 13: Linked List in C language

The “parent” is the pointer and the “project” is the header of the list. Since every folder may be removed from the list, all files and folders after this removed folder should be found and removed as well. This design is more efficient than store the directory as a parameter directly. It is easy to use recursion to find all files and folders under a folder from a database in this data structure design. This recursion will be introduced in the following report. After using this data structure, all files under different folders from different projects can be stored under the same path on the server. Figure 14 displays how they are stored on the server now.



```
file
    -qudhyITO
    08CD34Cvv
    0Hcr2mj_
    0JbCkgH8
    1092GYQ4S
    3DM76dZI5W
    4_aYU_m42
    5x8qUS_ts
    7Lu5cmc6
    7TXw0Hpat
    9iclsnM5a
```

Figure 14: File Storage

5.4 Implementation of Seeking Path by Recursion

This section presents how the program seeks the path of file or folder using recursion.

Figure 15 display the diagram of the recursion.

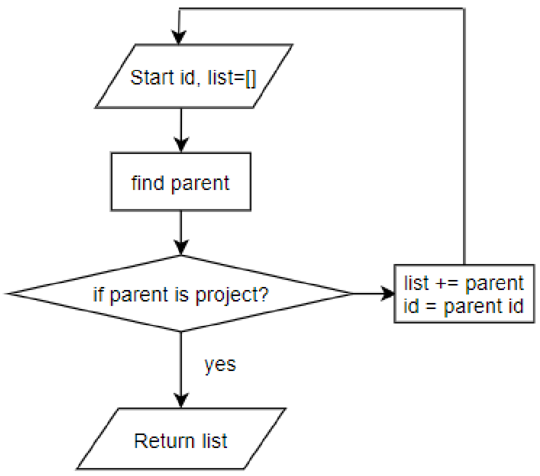


Figure 15: Seeking Path by Recursion

The start of recursion has the id of the file and an empty list. The recursion will first find the parent of the file. If the parent is the project, which means it reaches the root of the directory, the recursion will be terminated and return to the list. But if not, the parent id will be added into the list and also become the id. Then the new list and id will go to the start of this loop until the parent projects.

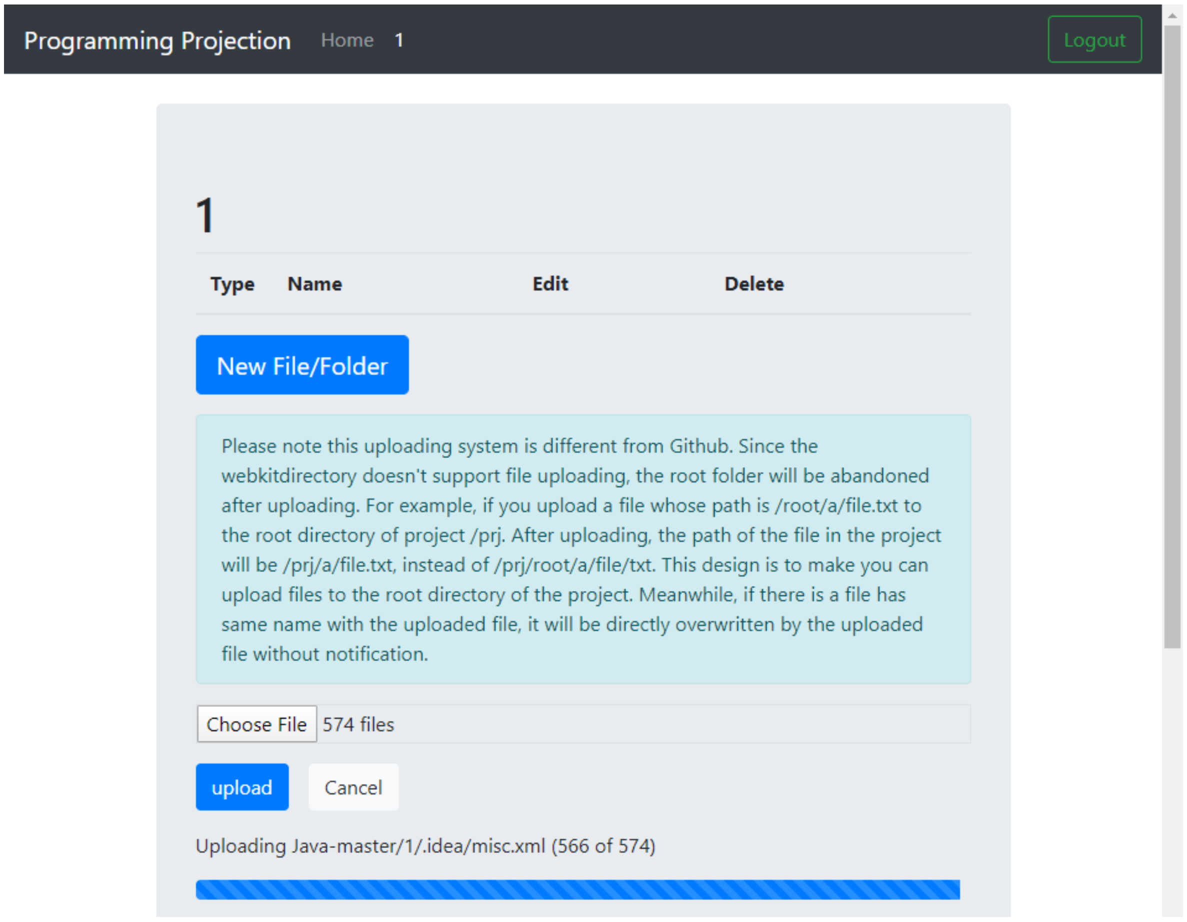
Recursion solves problems that loops are difficult to solve. The overhead of using recursion to find the path is insignificant. And the data structure of using this algorithm can reduce database redundancy significantly.

5.5 Implementation of Folder Uploading

The folder uploading function was implemented by webkitdirectory and FormData format. However, this design has a problem with the limitation of file amount. Since the form only submitted by one HTTP request, it is easy to exceed the size limitation of the request.

The solution is submitting one request for one file. Firstly, use webkitdirectory to read the file list that required to submit. Then use for loop to submit files one by one. This design can make the uploading system can upload an unlimited number of files. Even there are files exceed the size limitation, the whole upload progress will not be terminated. However, this design has another problem. Since Node.js handles requests asynchronously, some files' requests may run faster than others. And if a child file runs faster than its parent folder, the duplicate name folders may be created.

To solve this problem, the async await technique should be used. The implement of this function first used the recursion to replace the loops to prevent the asynchronous concurrent delivery on the front-end. Then used the Promises library of Node.js to create await mechanism to make sure every file and folder be stored one by one.



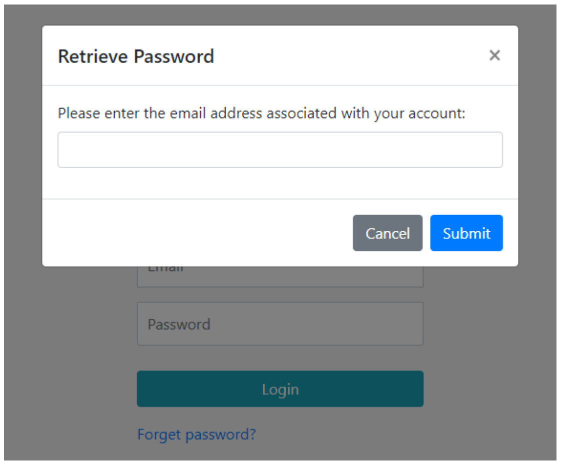
Please note this uploading system is different from Github. Since the webkitdirectory doesn't support file uploading, the root folder will be abandoned after uploading. For example, if you upload a file whose path is /root/a/file.txt to the root directory of project /prj. After uploading, the path of the file in the project will be /prj/a/file.txt, instead of /prj/root/a/file/txt. This design is to make you can upload files to the root directory of the project. Meanwhile, if there is a file has same name with the uploaded file, it will be directly overwritten by the uploaded file without notification.

Uploading Java-master/1/.idea/misc.xml (566 of 574)

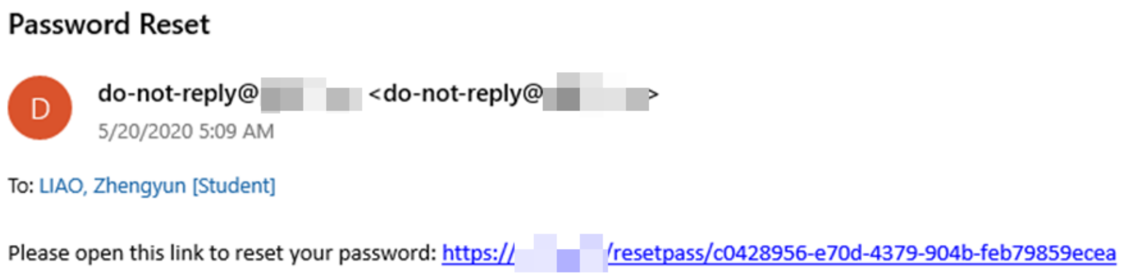
Figure 16: Folder Uploading can Upload Unlimited Number of Files

5.6 Implementation of other functions

Due to space limitations, there are many implemented functions that cannot be introduced in detail. In this section, they will be listed.


Retrieve Password

Please enter the email address associated with your account:

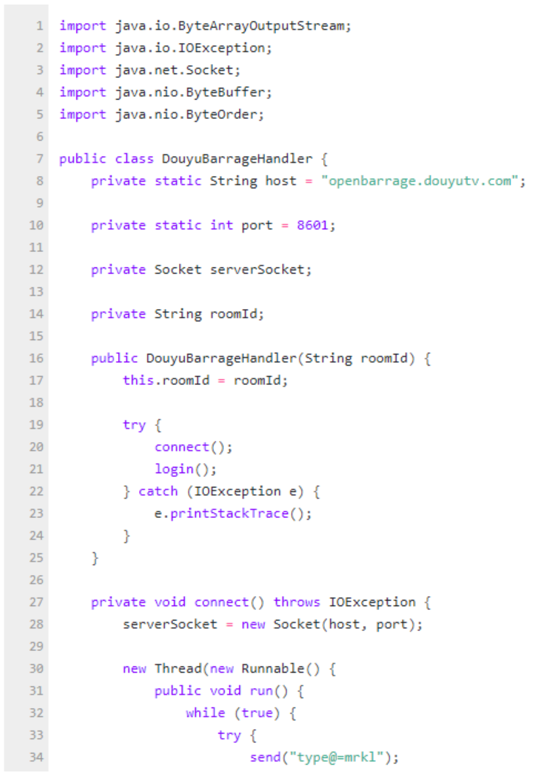

Password Reset

do-not-reply@[redacted] <do-not-reply@[redacted]>
5/20/2020 5:09 AM

To: LIAO, Zhengyun [Student]

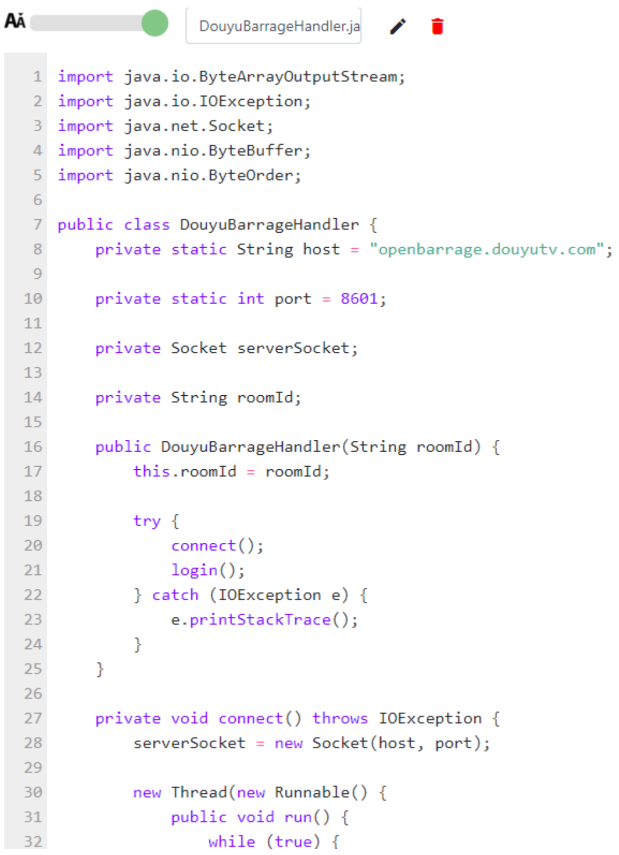
Please open this link to reset your password: https://[redacted]/resetpass/c0428956-e70d-4379-904b-feb79859ecea

- Password retrieving function. If a user forgets his password, he can use the password retrieving function. An email with a link will be sent to his mailbox. He can open the link to retrieve his password.

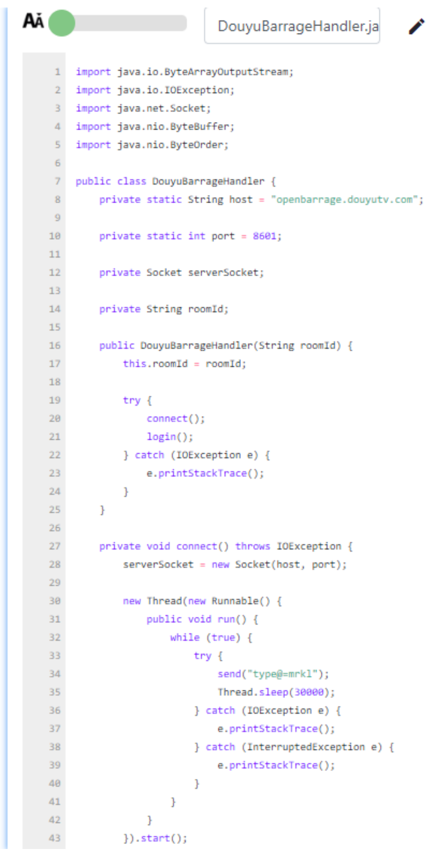


```java
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;

public class DouyuBarrageHandler {
    private static String host = "openbarrage.douyutv.com";

    private static int port = 8601;

    private Socket serverSocket;

    private String roomId;

    public DouyuBarrageHandler(String roomId) {
        this.roomId = roomId;

        try {
            connect();
            login();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private void connect() throws IOException {
        serverSocket = new Socket(host, port);

        new Thread(new Runnable() {
            public void run() {
                while (true) {
                    try {
                        send("type@=mrkl");
```

- Syntax highlighting function.

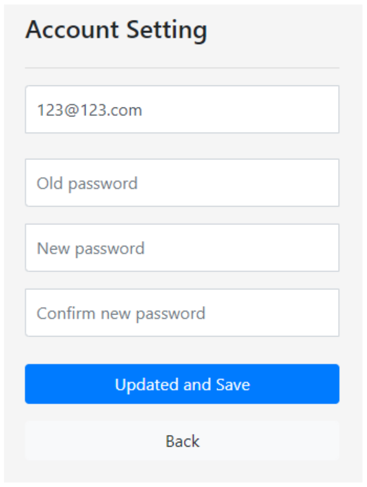


```java
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;

public class DouyuBarrageHandler {
    private static String host = "openbarrage.douyutv.com";

    private static int port = 8601;

    private Socket serverSocket;

    private String roomId;

    public DouyuBarrageHandler(String roomId) {
        this.roomId = roomId;

        try {
            connect();
            login();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private void connect() throws IOException {
        serverSocket = new Socket(host, port);

        new Thread(new Runnable() {
            public void run() {
                while (true) {
```



```java
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;

public class DouyuBarrageHandler {
    private static String host = "openbarrage.douyutv.com";

    private static int port = 8601;

    private Socket serverSocket;

    private String roomId;

    public DouyuBarrageHandler(String roomId) {
        this.roomId = roomId;

        try {
            connect();
            login();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private void connect() throws IOException {
        serverSocket = new Socket(host, port);

        new Thread(new Runnable() {
            public void run() {
                while (true) {
                    try {
                        send("type@=mrkl");
                        Thread.sleep(30000);
                    } catch (IOException e) {
                        e.printStackTrace();
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            }
        }).start();
```

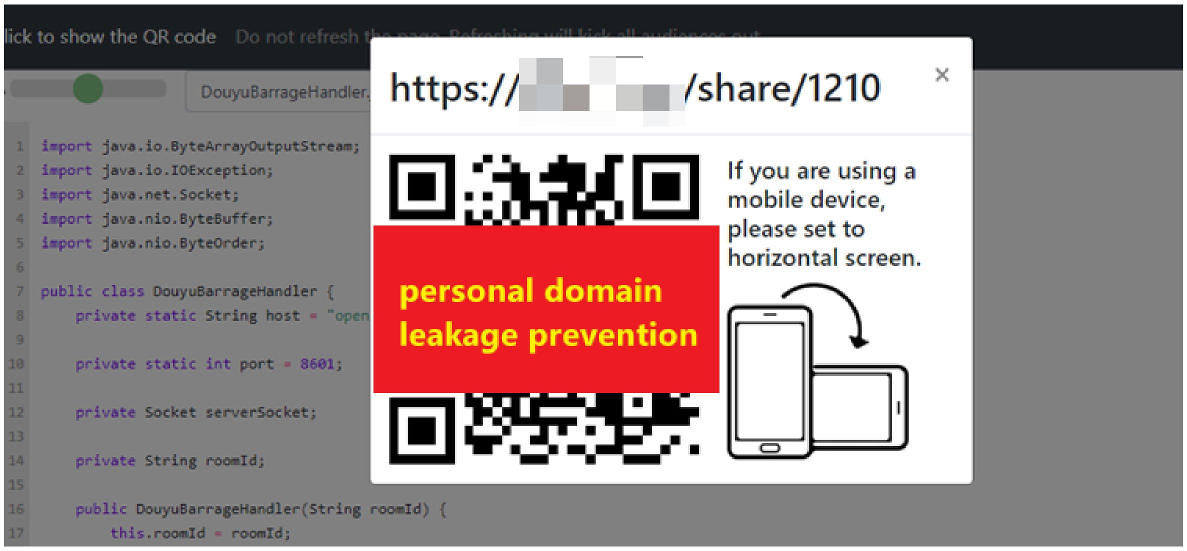
- Font size changing function



Account Setting

123@123.com

- Account setting function. Users can change their passwords and email addresses.



lick to show the QR code   Do not refresh the page. Refreshing will kick all audiences out

DouyuBarrageHandler.

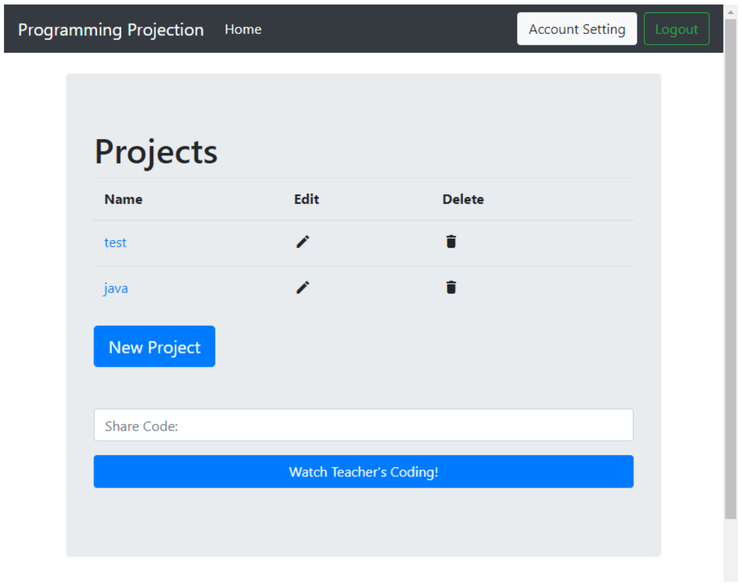
```java
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;

public class DouyuBarrageHandler {
    private static String host = "open

    private static int port = 8601;

    private Socket serverSocket;

    private String roomId;

    public DouyuBarrageHandler(String roomId) {
        this.roomId = roomId;
```

https://[redacted]/share/1210 ×

personal domain leakage prevention

If you are using a mobile device, please set to horizontal screen.

- QR code sharing function. QR code will be generated for each projection to be scanned by smartphone users to watch programming projection on their phones.



Programming Projection   Home   Account Setting   Logout

Projects

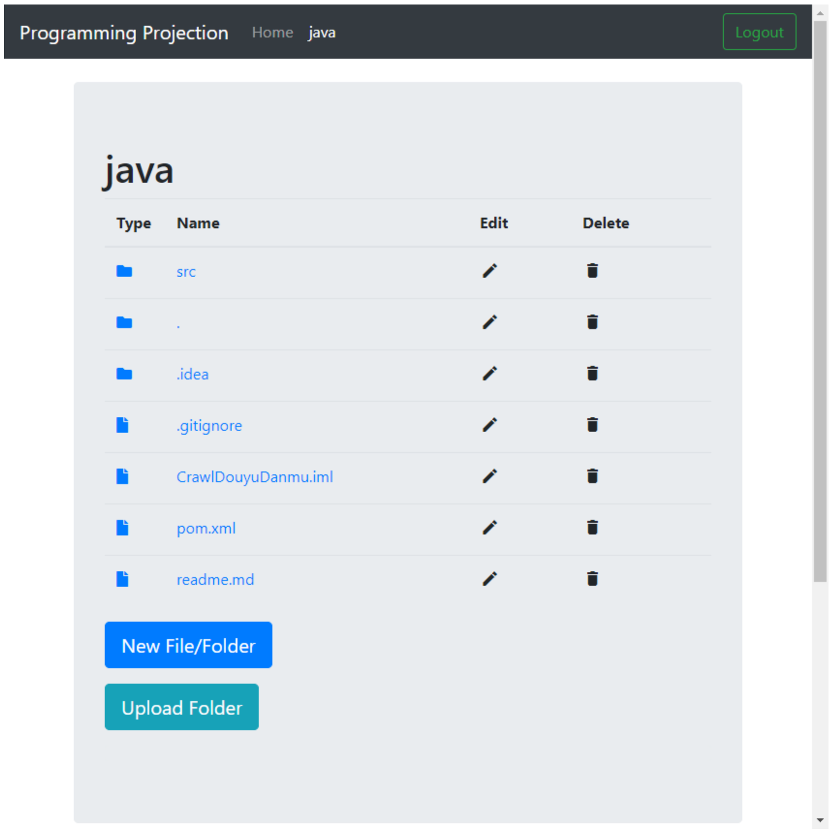
Name	Edit	Delete
test	✎	🗑
java	✎	🗑

New Project

Share Code:

Watch Teacher's Coding!

- Project Management Page



Programming Projection Home java Logout

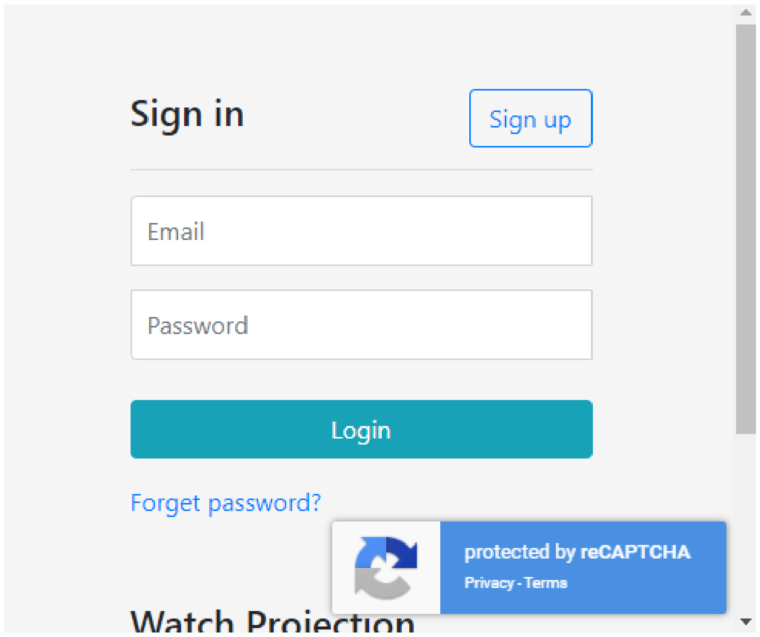
java

Type	Name	Edit	Delete
	src		
	.		
	.idea		
	.gitignore		
	CrawlDouyuDanmu.iml		
	pom.xml		
	readme.md		

New File/Folder

Upload Folder

- Folder Management Page



Sign in

Sign up

Email

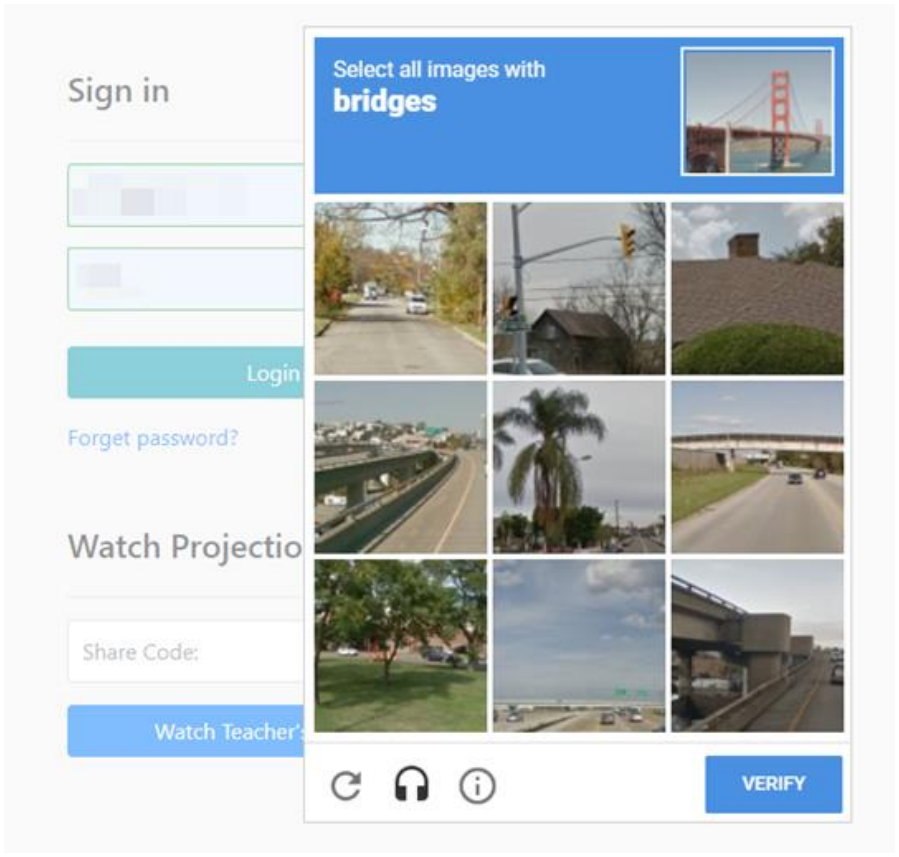
Password

Login

Forget password?

protected by reCAPTCHA
Privacy - Terms

Watch Projection



- Login and Sign up pages are protected by reCAPTCHA V2 invisible

6 Conclusion

This report describes the researches and implementations did in this project. The report firstly introduces the main technologies Node.js and WebSocket used in this project, describes the researches of the alternative technologies PHP, Java, Cassandra, and long polling, and explains why those alternative technologies are not suitable for this project. Then the report describes the implementations and evaluation in detail. This project implements a fully functional programming projection platform. All major requirements from the supervisor are implemented successfully. The problem of programming projection in the classroom has been solved by this platform completely. There are some suggestions from surveys in evaluation. The UI design and projecting functions can be improved in the future.

7 Reference

[1] M. Cantelon, M. Harter, T. Holowaychuk, and N. Rajlich, *Node. js in Action*. Manning Greenwich, 2014.

[2] M. Madsen, F. Tip, and O. Lhoták, "Static analysis of event-driven Node. js JavaScript applications," in *ACM SIGPLAN Notices*, 2015, vol. 50, no. 10, pp. 505-519: ACM.

[3] O. Etzion, P. Niblett, and D. C. Luckham, *Event processing in action*. Manning Greenwich, 2011.

[4] A. I. Khan and N. A. Patel, "Callback function for messaging platform in public telephone system," ed: Google Patents, 2007.

[5] D. Flanagan, *JavaScript: the definitive guide*. " O'Reilly Media, Inc.", 2006.

[6] R. Dahl, "Node.js online documentation," *https://nodejs.org/en/*.

[7] I. K. Chaniotis, K.-I. D. Kyriakou, and N. D. J. C. Tselikas, "Is Node. js a viable option for building modern web applications? A performance evaluation study," vol. 97, no. 10, pp. 1023-1044, 2015.

[8] A. Lombardi, *WebSocket: lightweight client-server communications*. " O'Reilly Media, Inc.", 2015.

[9] V. Pimentel and B. G. J. I. I. C. Nickerson, "Communicating and displaying real-time data with websocket," vol. 16, no. 4, pp. 45-53, 2012.

[10] Y. J. I. Furukawa, "Web-based control application using WebSocket," pp. 673-675, 2011.

[11] J.-P. J. A. U. S. o. S. Erkkilä, "Websocket security analysis," pp. 2-3, 2012.

[12] R. Lawrence, "Integration and virtualization of relational SQL and NoSQL systems including MySQL and MongoDB," in *2014 International Conference on Computational Science and Computational Intelligence*, 2014, vol. 1, pp. 285-290: IEEE.

[13] K. Lei, Y. Ma, and Z. Tan, "Performance comparison and evaluation of web development technologies in php, python, and node. js," in *2014 IEEE 17th international conference on computational science and engineering*, 2014, pp. 661-668: IEEE.

[14] T. Åkesson and R. Horntvedt, "Java, Python and Javascript, a comparison," ed, 2019.

[15] V. Abramova and J. Bernardino, "NoSQL databases: MongoDB vs cassandra," in *Proceedings of the international C* conference on computer science and software engineering*, 2013, pp. 14-22.

[16] G. Rauch, "Socket.IO online documentation," *https://socket.io/docs/*.

[17] K. Ma, A. Abraham, B. Yang, and R. Sun, "Intelligent Web Data Management of WebSocket-Based Real-Time Monitoring," in *Intelligent Web Data Management: Software Architectures and Emerging Technologies*: Springer, 2016, pp. 105-124.