Computer Science and Engineering Department, Chinese University of Hong Kong

Code Block Online IDE
Final Report

CSCI3100 Group 18

Document Version 1.0

CHEUNG Ka Wai	FENG Haolin	LEE Tsz Yan	YU Chi To
SID:1155110140	SID:1155110663	SID:1155110177	SID:1155110447

May 16, 2020

Contents

1

Introduction

1.1 Project Overview

The Code Block Online IDE provides a place for programmers to test and share their code. It is a platform for programmers to share their knowledge on coding and learn from each others on the forum. Programmers can also prototype their code on this online IDE to test out their codes, and share it with the others.

1.2 Objective

Programming skills has become more and more inportant throughout the world. And the promotion of STEM education is now worldwide and implemented in large scale. However, while the traditional text-based coding may be efficient, but they may not be able to arouse the attention to younger kids. While there are some visualised coding platforms available, but they are not well designed for older kids to ask and discuss on a problem.

Therefore, the Code Block Online IDE is designed to adapt to the problem. Beginners in programming can learn coding using the visualised online C programming IDE. For advanced programmers, they can simply type in the textbox for higher efficiency. And they can share their code for discussions or questions using the forum. We hope that this kind of forum design, that each post must be connected to a program will help creating a platform like quora, but aimed at programming, where people can exchange their thoughts on their codes.

The IDE designed may also be a tool for rapid prototyping. Although there is a current limitation on storing the visualised code, which hindered its usage on rapid prototyping, and there is not enough libraries pre-written for prototyping. But it can still be used for the purpose by drag-and-drop functions built into suitable locations thus may help when prototyping programs.

1.3 Highlights

1.3.1 Visualised Code Editor

The visualised code editor is one of the core highlights of the Code Block Online IDE. The visualised code editor allows beginners in programming to have a more vivid understanding on the meaning of each component of their program. Users of the editor drag suitable blocks of code into the coding area. And the corresponding code will be generated immediatly below the drag-and-drop area so that they can have an overview on the program written. Authenticated users can then choose to save the code on our platform and run it on our server or choose to copy it down and run on their local computer. The visualised code editor used here is adapted from repository lisa of dineshLL[1].

1.3.2 Forum

We encourage every user to share and discuss their knowledge. The forum allows authenticated users to share their code and discuss about it here. Each post must have a source code attached with it to act as an example for efficient discussion. Users can post their code and the problem they faced here, so that other users can answer and explain the problem directly and easily. Users can also share their experiences with their code here to the others. And other users interested in the topic can discuss and use the code shared to their expense. For unauthenticated guest users, they can still browse the posts in the forum to gain knowledge here.

1.3.3 Server-based Code Compilation and Execution

Like many other online program compilation and execution websites, the source code is compiled and executed at the server. This is to prevent posing security threats to client's computer by allowing an unknown excutable running on their computer. Another reason for server-based code compilation and execution, is for the ease of coding assignment verification. In programming courses, one of the biggest problemed faced by the students is the difference in output when the output of their assignment is compared with that of the standard answer's. And that is the portability problem. Server-based compilation for both standard answer from teachers and assignment hand-ins from students will solve the problem as their code will be ran on the same operating system. Therefore the protability problem is solved.

1.3.4 Search Engine

In order for efficient knowledge exchange between users, we must ensure users can find the materials they need. Therefore we provide search to all users so that they can find posts that contain the information they need. We allow users to use username, words in title, words in post content and words in reply content as search kwyword. Users can type in any keyword in the search field. The server will find the keyword in all posts for result. This way, users can have an easy access to knowledge they need.

1.3.5 Personal Account System

As we expected educational institutions to use this platform for teaching, we have classified accounts into 3 types. First, normal user accounts that are not related to any educational institutions. Second, user account for teachers, and third, student user accounts. For teacher accounts, they can check coding assignments of students using teacher accounts. They can also find out what posts their students have posted so that they can know the difficulties of students faced when studying this course. For student accounts, users of this type of accounts will can find out the posts of their teacher easily as they are also listed on their user page. Therefore, teachers can use this platform to understand the need of students by checking on the posts posted by students on different problems or as a respond on the material posted on posts by lecturers here., and release assignments or extra materials here to their student.

1.4 Project Statistics

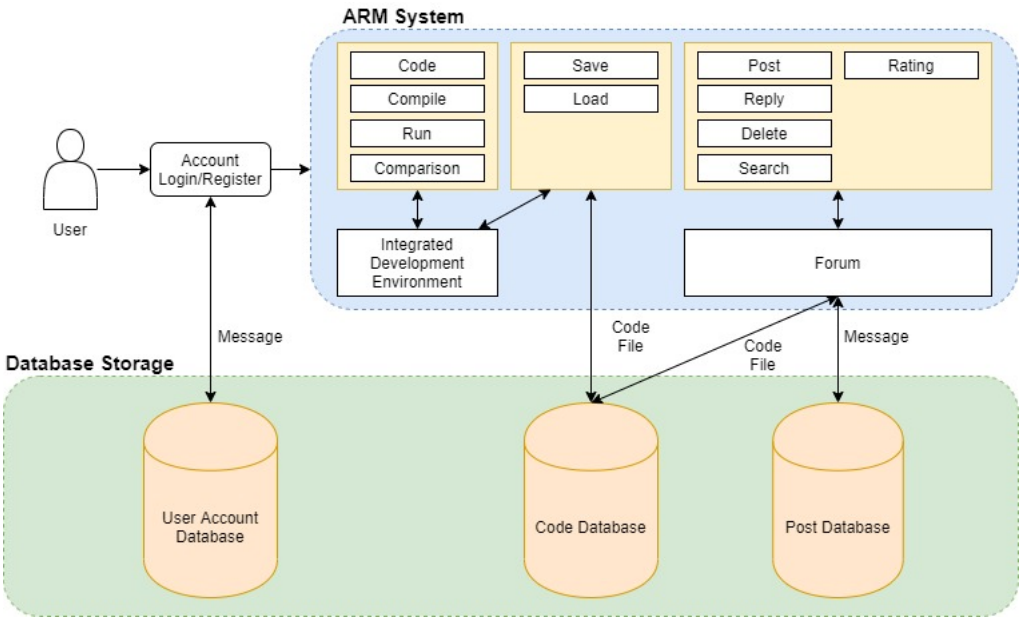
ES6-Plato is used to generate the statistics for server and website Javascripts[2].

File Name	SLOC	McCabe's Number
Server Node.js Scripts		
app.js	815	90
code.js	119	9
compAndRun.js	101	10
connect_sql.js	980	95
forum.js	140	5
php_parser	44	2
user.js	112	10
Webpage Javascripts		
code_result.js	14	1
deletePost.js	20	6
redir.js	5	1
submitUser.js	123	20
submitForm.js	29	6
EJS-Written Website Templates		
account_header.ejs	10	1
change_password.ejs	31	1
code_result.ejs	24	1
code.ejs	24	1
create_account.ejs	33	1
forum_header.ejs	12	1
forum_search.ejs	21	3
forum.ejs	25	3
login.ejs	33	1
mainpage.ejs	36	1
navbar.ejs	51	3
new_post.ejs	38	2
post.ejs	39	3
search.ejs	16	1
student_profile.ejs	33	4
student.ejs	53	7
teacher.ejs	53	7
user_profile.ejs	28	3
user.ejs	46	5
Workspace.ejs	231	1

2

System Architectural Design by DFD

2.1 Architectural Diagram



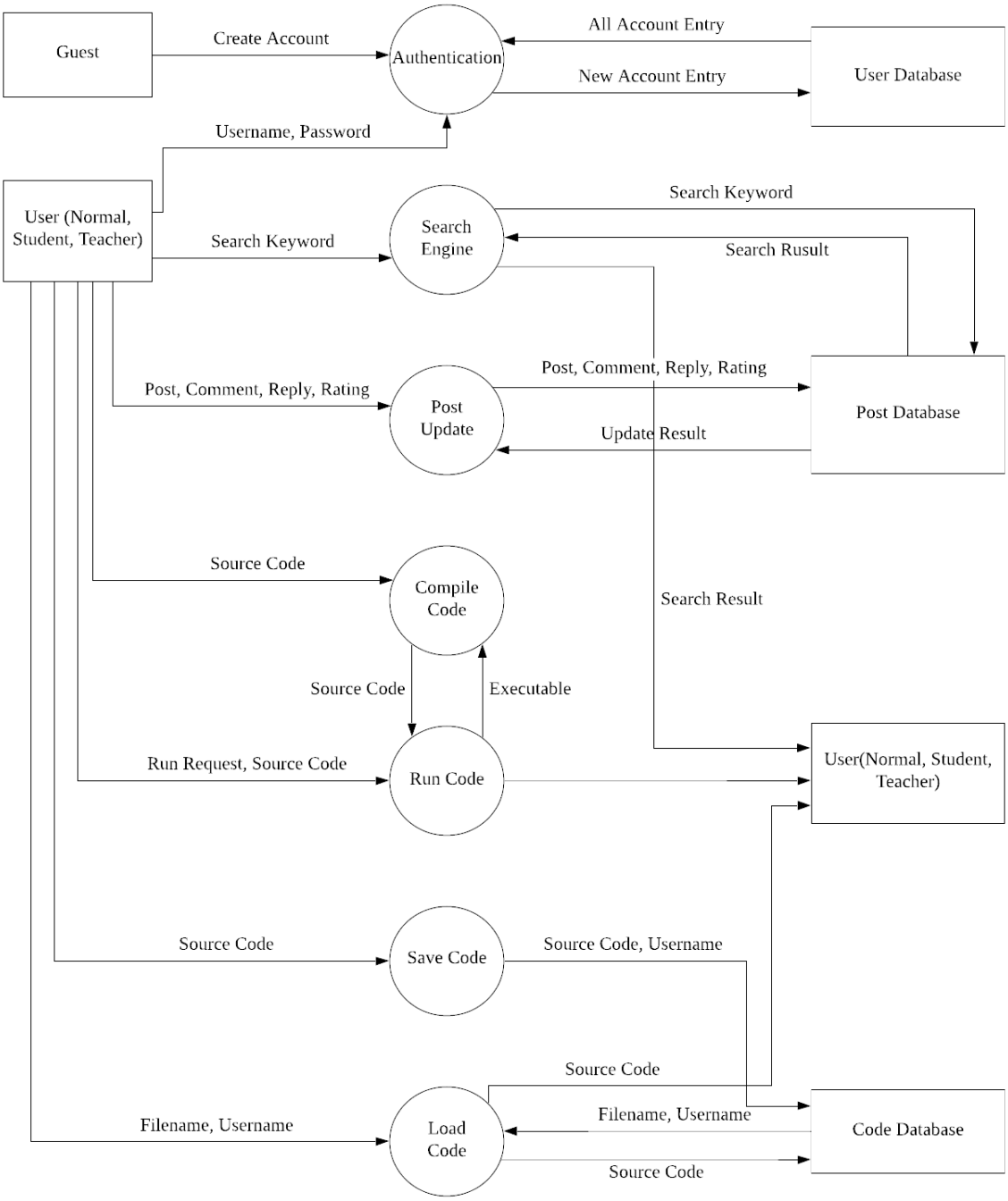
2.2 System Components

1. Personal Account System: Users Information are stored into the user database. Based on current design, users are classified into 3 class, normal user, student and teacher. They have no differences in this phase.
2. Forum System: Posts are stored into the post database. List of post threads created by users, allows users to post and comment with thier source code. Users' posts can be searched and display to other users.
3. Code Running System: Source code are stored into the code database. Source code created by users can be compiled, saved and load. Compiled program can be executed and its output will be shown.

2.3 Database

1. User Account Database includes username, account password, account type, extra permission of the account for teachers and student users to access posts only available to the class.
2. Post Database includes posts contents (contents, author, reply, source code linkage, permission) and access frequencies of posts.
3. Code Database includes source code file, author of the code, filename.

2.4 Data Flow Diagram(DFD)

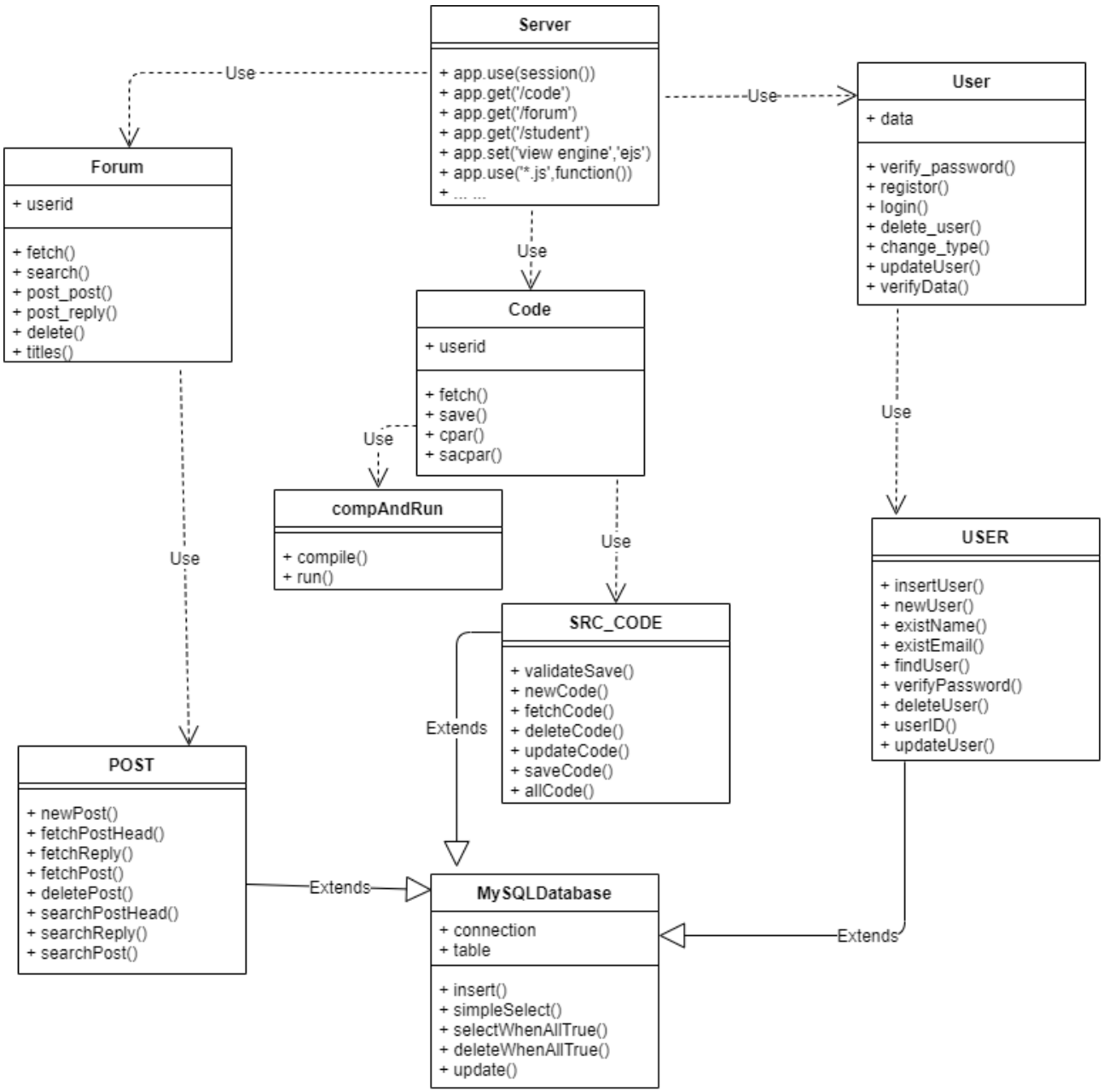


3

Detailed Description of Component by UML

3.1 General Class Diagram

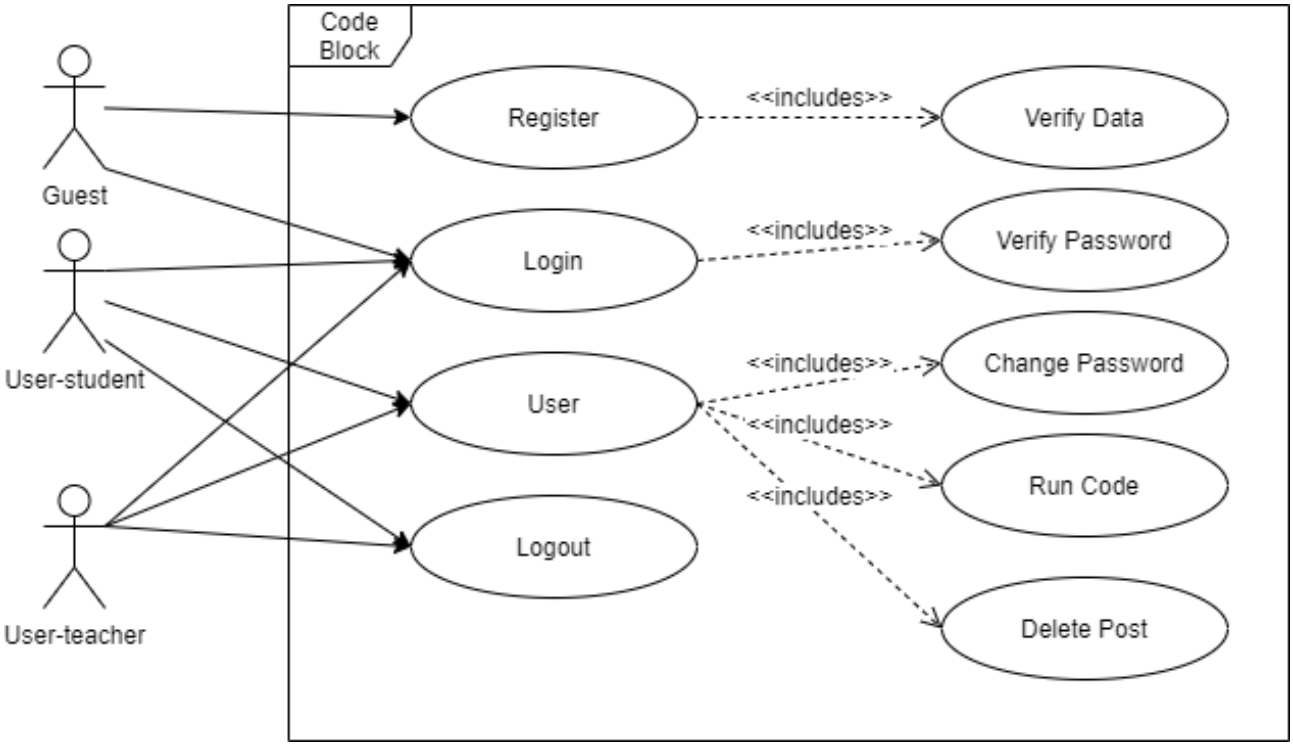
The following UML Diagram shows all classes and relations of the whole system.
The main class server uses three classes which are User, Code and Forum classes. User class uses USER class, Forum class uses POST class and Code class uses compAndRun and SRC_CODE class. USER, POST and SRC_CODE classes are extended from MySQLDatabase class


Server
+ app.use(session())
+ app.get('/code')
+ app.get('/forum')
+ app.get('/student')
+ app.set('view engine','ejs')
+ app.use('*.js',function())
+ ... ...
Use
Use
Use
Forum
+ userid
+ fetch()
+ search()
+ post_post()
+ post_reply()
+ delete()
+ titles()
User
+ data
+ verify_password()
+ registor()
+ login()
+ delete_user()
+ change_type()
+ updateUser()
+ verifyData()
Code
+ userid
+ fetch()
+ save()
+ cpar()
+ sacpar()
Use
Use
Use
compAndRun
+ compile()
+ run()
USER
+ insertUser()
+ newUser()
+ existName()
+ existEmail()
+ findUser()
+ verifyPassword()
+ deleteUser()
+ userID()
+ updateUser()
SRC_CODE
+ validateSave()
+ newCode()
+ fetchCode()
+ deleteCode()
+ updateCode()
+ saveCode()
+ allCode()
Extends
POST
+ newPost()
+ fetchPostHead()
+ fetchReply()
+ fetchPost()
+ deletePost()
+ searchPostHead()
+ searchReply()
+ searchPost()
Extends
MySQLDatabase
+ connection
+ table
+ insert()
+ simpleSelect()
+ selectWhenAllTrue()
+ deleteWhenAllTrue()
+ update()
Extends

3.2 Component-1: Personal Account System

Personal Account System is a database to provide the identity for student user or teacher user in server and provide several functionalities to user.

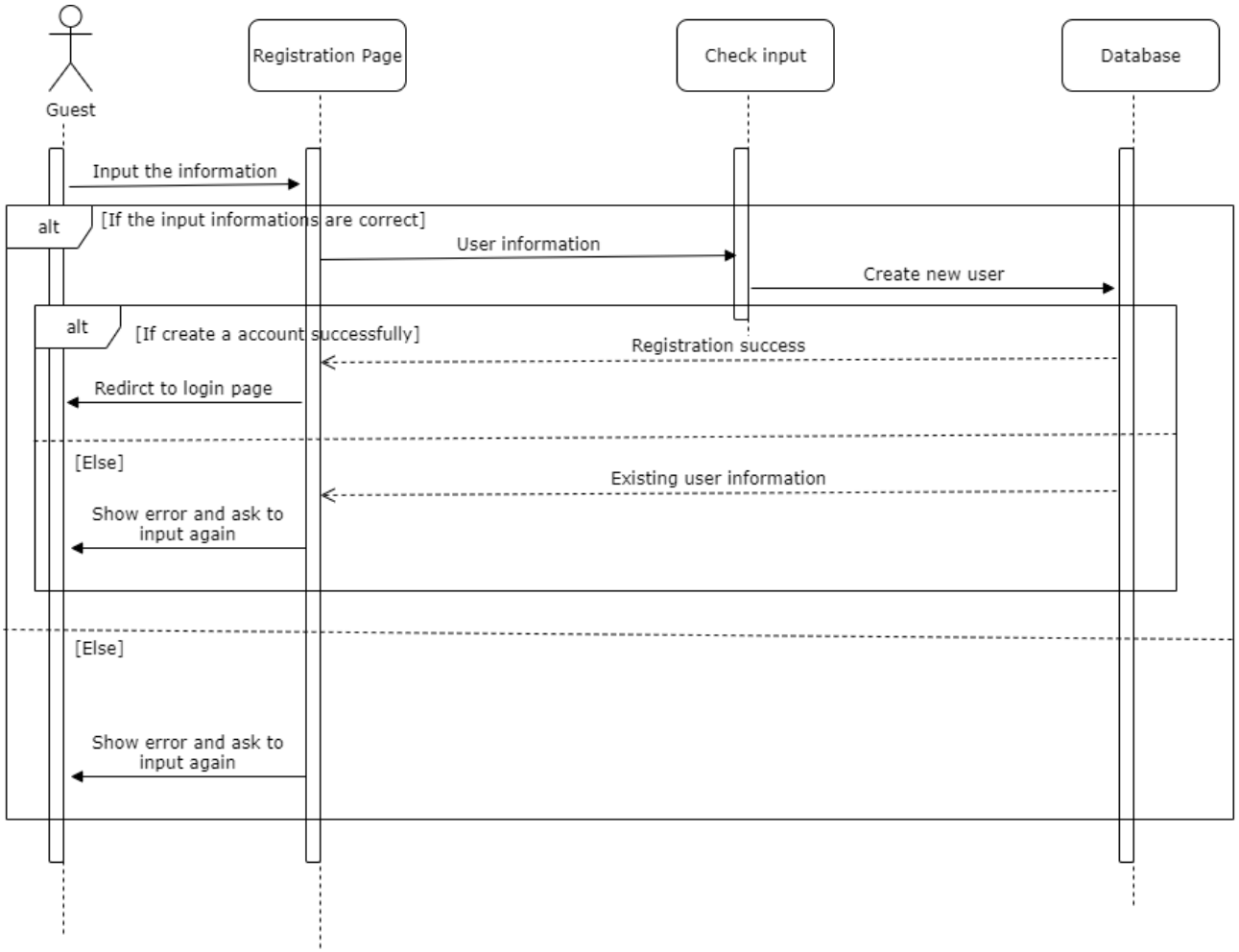
3.2.1 Use Case Diagram



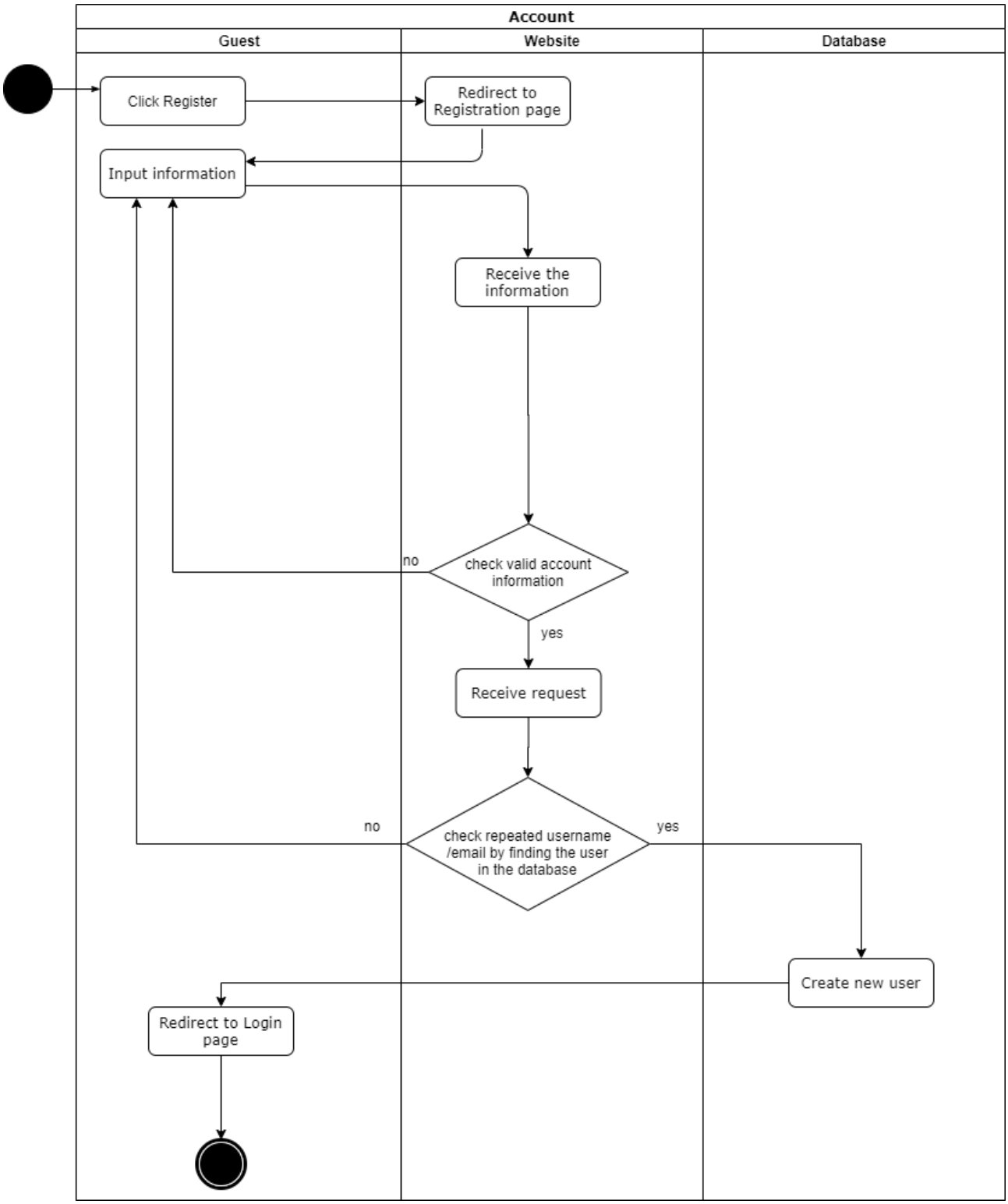
For unauthenticated users, the Personal Account System allows he/she can either go to register or login. The register function includes verifying data function to verify whether the guest's inputs are all validated. The login function includes verifying password function to verify the password input by guest by finding in the database.

For student and teacher user, Personal Account System allows he/she to login and view his/her profile page. The user function provides a page for user to change his/her password, run his/her code and delete his/her post.

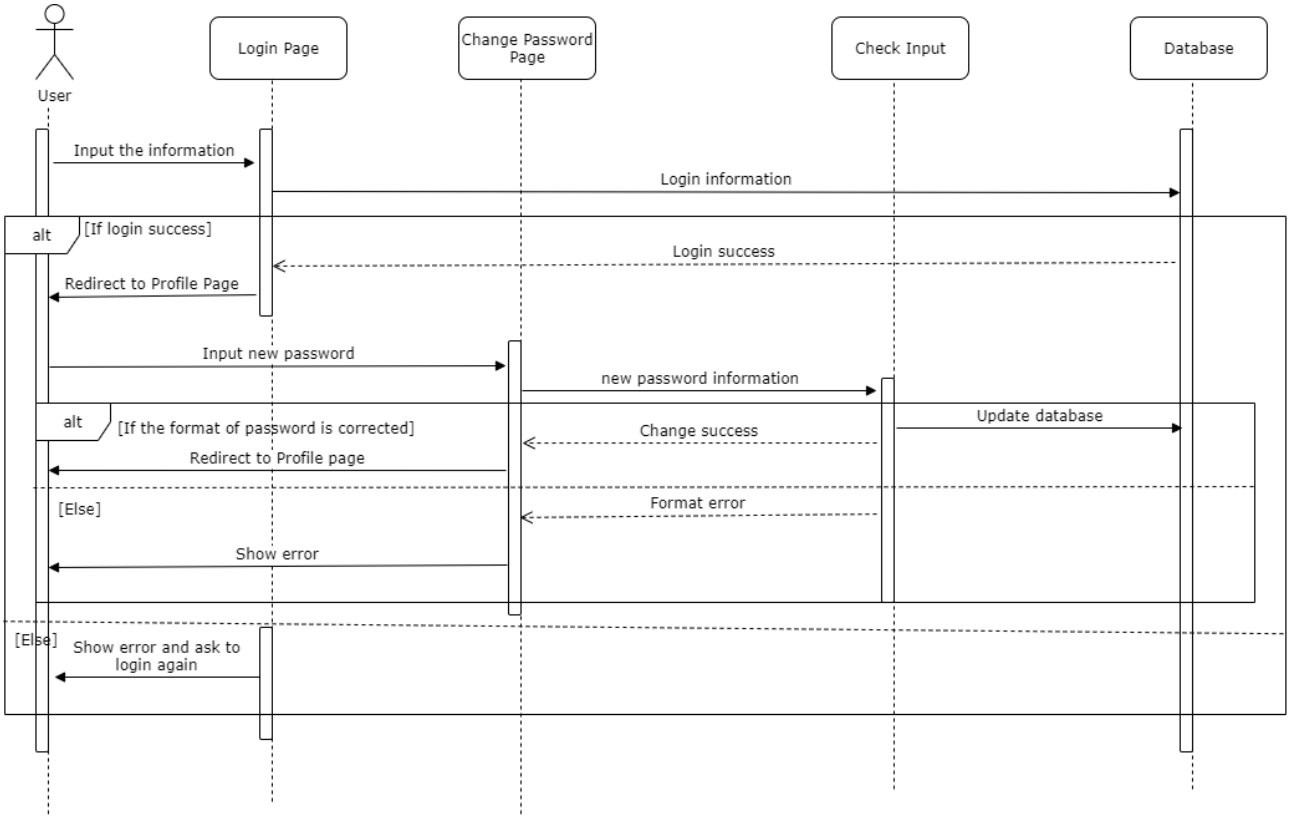
3.2.2 Sequence Diagram for Guest



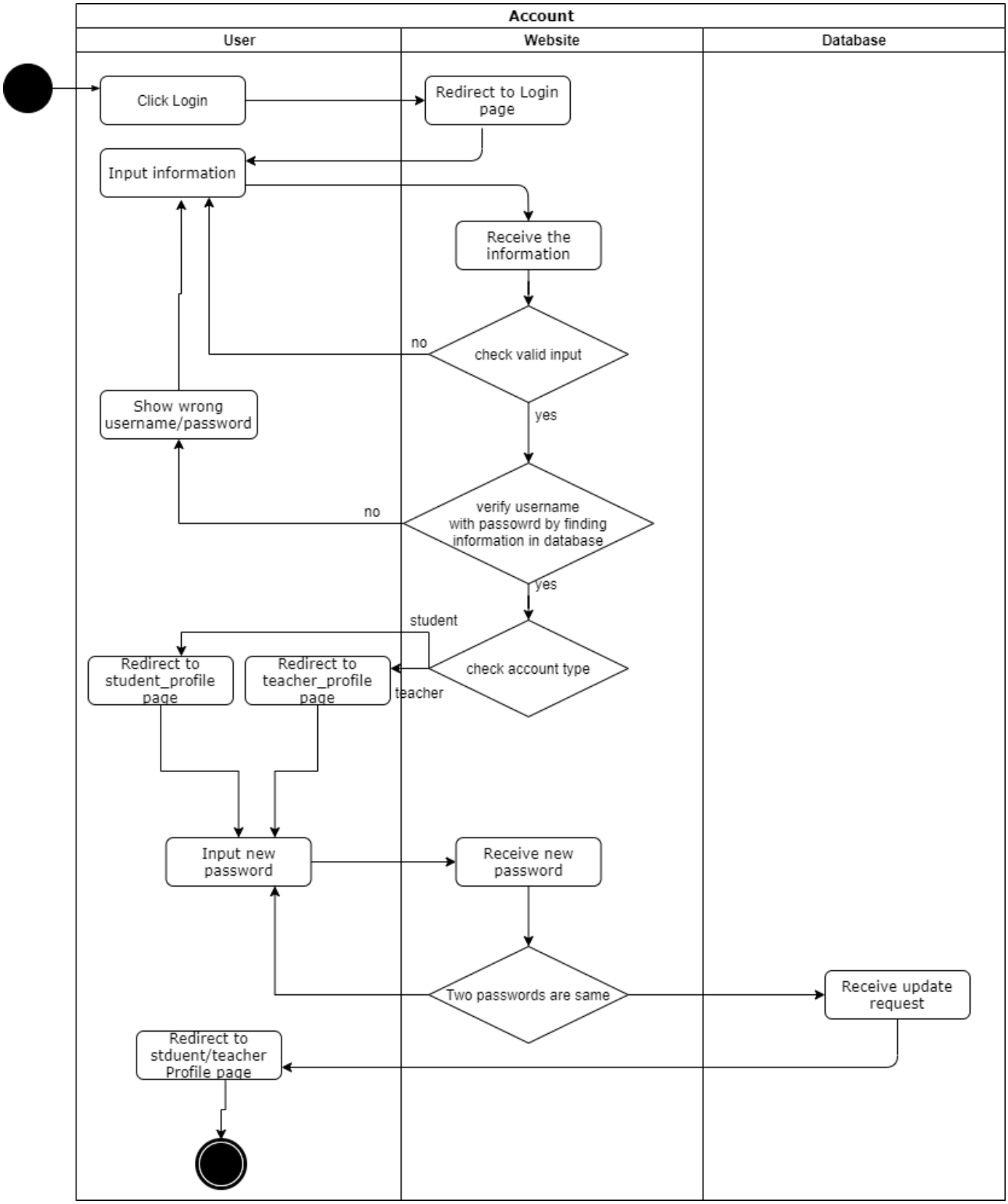
3.2.3 Activity Diagram for Guest



3.2.4 Sequence Diagram for User(login and change password for both types of user)



3.2.5 Activity Diagram for User(login and change password for both types of user)



3.2.6 Functionality

This component provided user authentication. It prevented guest without authentication to use some functions which need user identity. For example, writing code, new post and reply. It also provides an easier way for the system to record all the codes, posts and replies that created by that person and user can retrieve those information.

3.2.7 Procedures and Functions

This component provides front-webpages, which are registration page, login page and profile page.

The registration and login page provide basic input checking(format, completion, and length) before passing data request to the server. After passing the input checking, the webpages will send requests to the database, verifying the data, whether the guest input available username and verifying the password correctness, whether the user login with correct pair of username and password.

The profile page allows users to change their password, run their code and delete their posts. In this page, users can find their profile with username, codes and posts. This page will send a request to server side, getting those information. When user wants to change password, it will do the password verifying for user input, then it will send the request to the server side, changing the password.

For the backend application part, there are lots of functions with SQL statement to retrieve data or update data from the database. It will return the result to the front-webpages after retrieving or updating data, whether it is successful to get or update data.

In conclusion, this system contains the following functions:

- **Registration** , which is for the guest to register account. This function will verify the data that guest inputs is valid. If the data is valid, create a new account related to the data provided by the guest and redirect to the login page, otherwise, request the guest to input data in correct format.

- **Login** , which gets the username and password input by user, then verify the correctness from the database. If the input is valid, login to the server, otherwise request the user to input again.

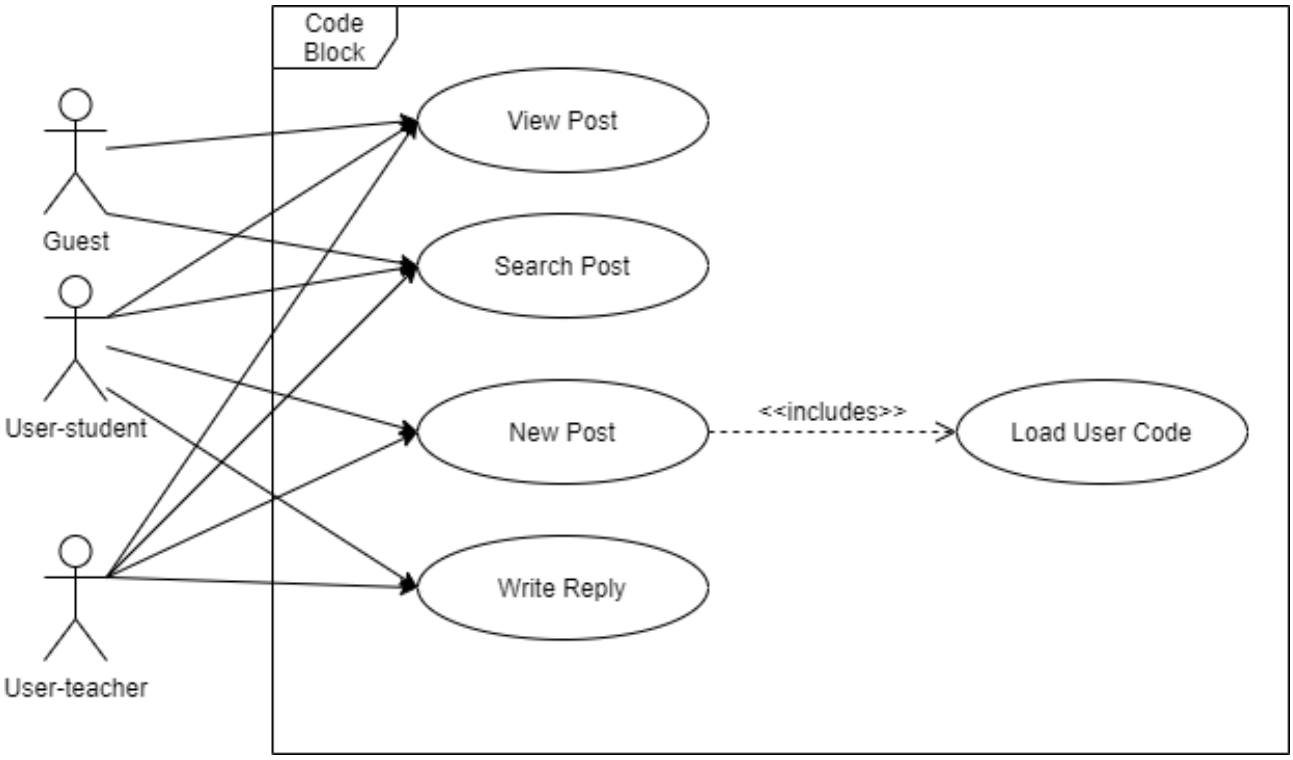
- **Change password** , which gets the two passwords input by user, then verify whether the two inputs are the same. If the inputs are valid, request server to update the information in database, otherwise request the user to input again.

- **Run code** , which allow user to choose one of his/her code and redirect to that code page for user to run.
- **Delete Post** , which allow user to delete the posts wrote by him/her.
- **Logout** , which stops the access permission with a logged in account.

3.3 Component-2: Forum

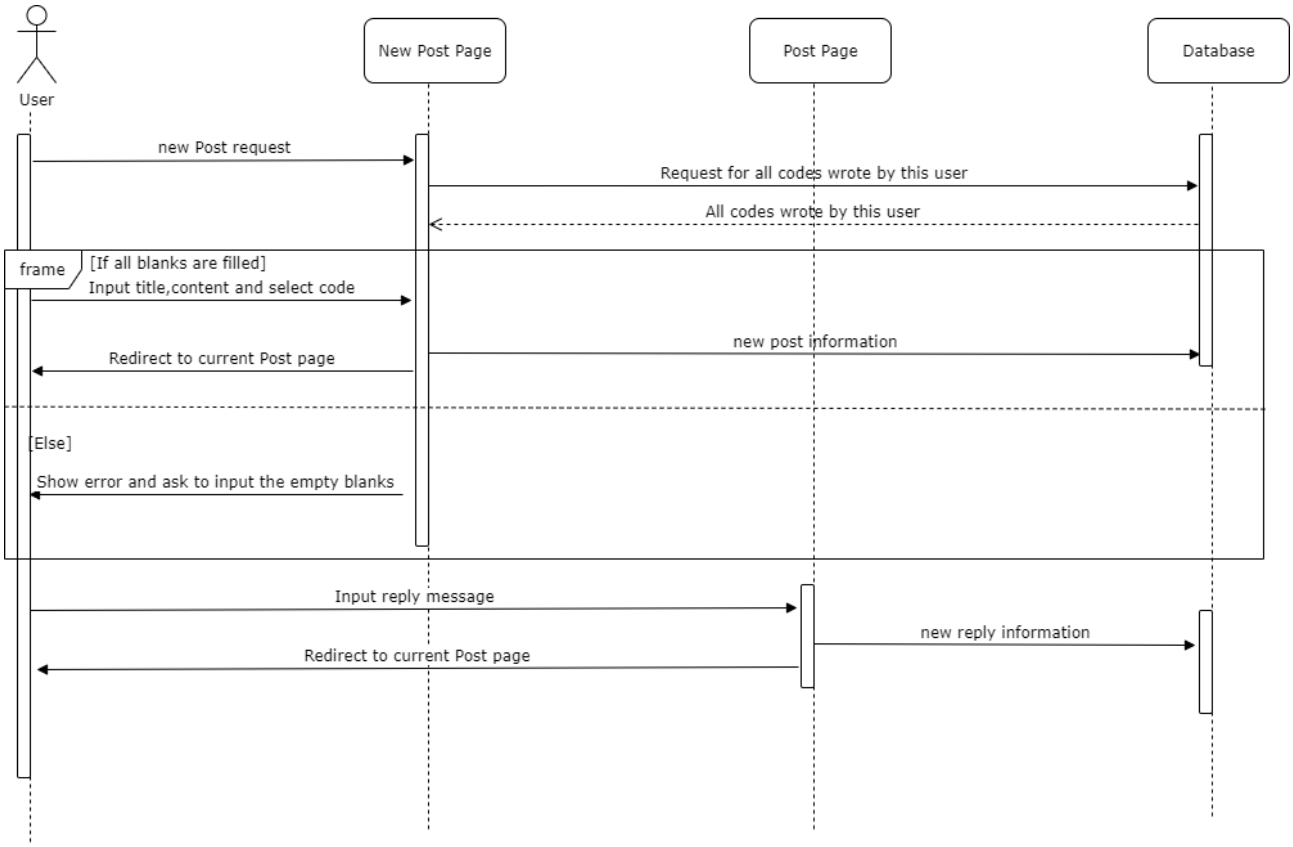
The forum system allows users to show their codes and discuss on them. The user can write new post, reply to the post, view post and search post.

3.3.1 Use Case Diagram



In this system, functions are limited to a guest, guest can only view post and search post. A user is allowed to write new post and write reply. The new post function includes load user code function to list out all the codes that user wrote by finding in the database.

3.3.2 Sequence Diagram for Writing new Post and Reply



3.3.3 Functionality

The forum component provides a platform for the user to post their code and discuss with other. It contains search post, view post, write new post, write reply functions. Everyone(guest and user)can view post and search post. Only logined users are allowed to write new post and write reply. If user wants to write a new post but he/she doesn't input title, content and select a code, the page will request user to input all the information, otherwise request the server to update a new post in database.

3.3.4 Procedures and Functions

For the front-end part, the system provided view forum, view post, search post and write new post these 3 pages. For guest, the system will only display view post without reply function, view forum and search post.

In view forum,view post and search post, guests and users are allowed to click inside, viewing the content. When they enter these pages, the server will receive the searching request, and return the required result.

In addition, users are allowed to write new post with the new post page. Users are required to fill in the title, content and select one of their codes in the page. Then the whole data set will be sent to the database, and the server will redirect user to that post page. In a post page, users are allowed to write reply, and the reply will be saved into the database.

In conclusion, this system contains the following functions:

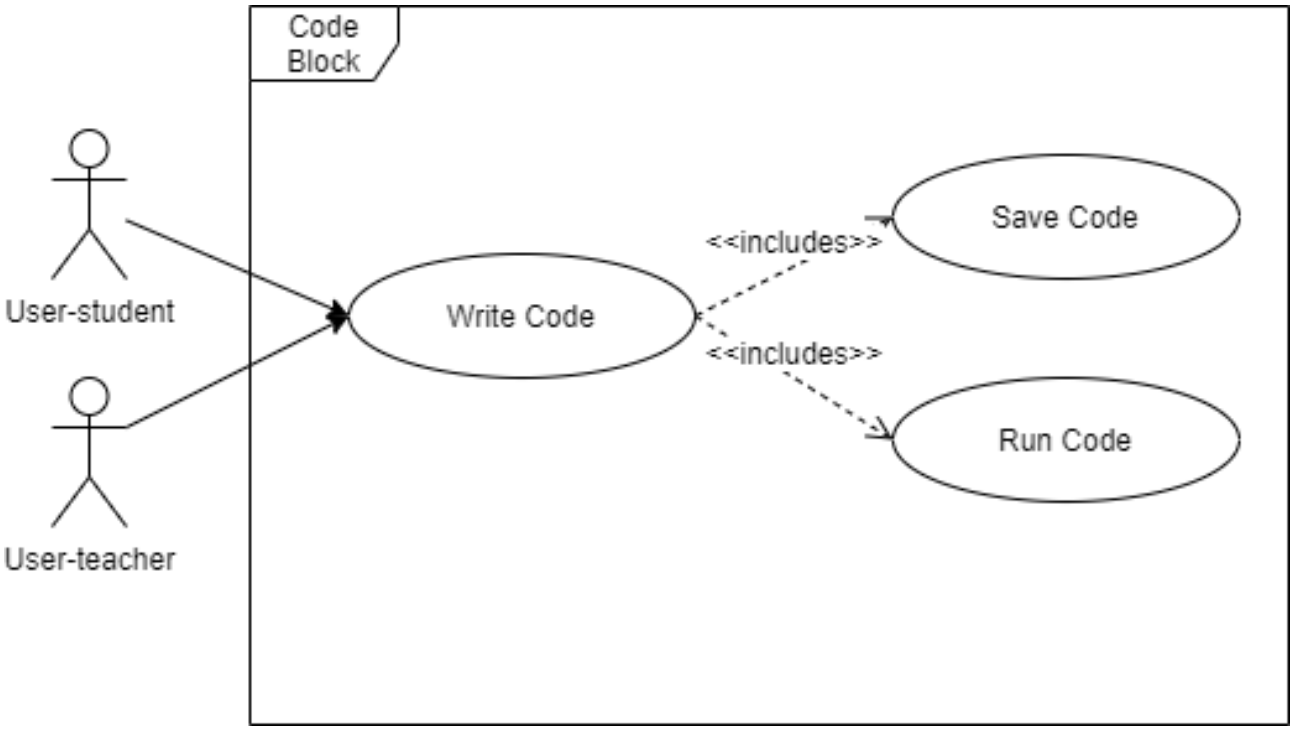
- **View Post** , which is for everyone(guest and user) to view the post and reply content.
- **Search Post** , which is a search function receive keyword input by guest or user and return all matching posts.
- **New Post** , which is for user to write new post with selecting a code and save into the database.

- **Write Reply** , which is for user to write reply in the post and save into the database.

3.4 Component-3: Code

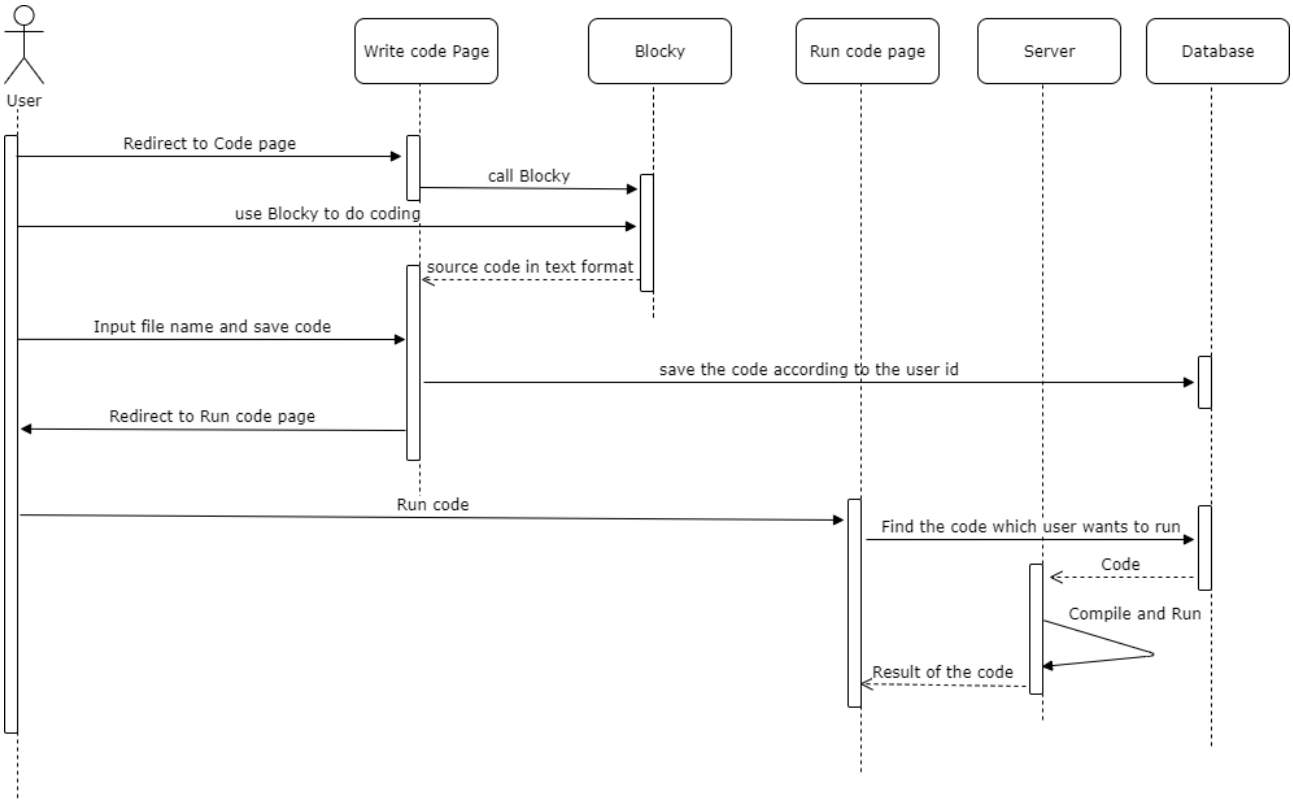
This is a workplace like a online IDE for users to write code in a simple and visualized system.User can drag the block and drop it on the workplace to do coding. The system will automatically generate the code in text format. The user can also save code and run code.

3.4.1 Use Case Diagram



In this system, functions are only provide to user, so guest should register a account first. User is allowed to write code. The write code function includes save code and run code functions.

3.4.2 Sequence Diagram for Writing Code, Saving Code and Running Code



3.4.3 Functionality

The code component provides a workplace for the user to do coding online. It contains write code, save code and run code. Only user can use all these functions.

3.4.4 Procedures and Functions

This component provides front-webpages, which are workplace page, code page and result page.

The workplace page provides a visualized coding area, a code generation area and a form to save code. User should input the file name and copy the code from the code generation area to the form, then the page will send request to server to save the code into the database according to the user identity. After that, the server will redirect the user to that code page.

The code page shows the code wrote by the user for viewing, and there is a input area is used for STDIN when the program required input from keyboard. User can run the code in this page, the page will send the request with the code and input data to server, server will do the compilation and execution. After that, the server will redirect user to result page and return the code result to that page.

For the backend application part, there are lots of functions with SQL statement to retrieve data or update data from the database. Server handles the run code request and return the result to the result page.

In conclusion, this system contains the following functions:

- **Write Code** , which is for user to do coding in a simple and visualized workplace. User can drag the block and drop it on the workplace and the system generates code in text format.

- **Save Code** , which is for user to save the code he/she writes. User should input the file name then copy and paste the code generated by the system into the form, then the page sends request to server and ask to update the database according to the user identity. After that, the server redirects user to that code page.

- **Run Code** , which is for user to run his/her code with optional STDIN input. It sends requests with code and input data to server, server handles the request then do compilation and execution. After that, the server redirects user to result page and return the code result in that page.

4

User Interface Design

4.1 Overview

The user interface of Code Block Online IDE should be simple and easy to control. The user should be able to find what they want intuitively. Therefore a header is designed for easy access to different functions in the application. Under this principle, the overall user interface design concept is "block", which is the main theme of the application. Every element in the application is in a form of a flowing block (except for the coding page). The flowing block design aims to let users focus on the content on the page.

4.1.1 Header

Header with a menu is appended on the top of every page. Only the main operations are shown to keep it clear. Under the option "user", a submenu with more options will be displayed when the mouse is on hover.

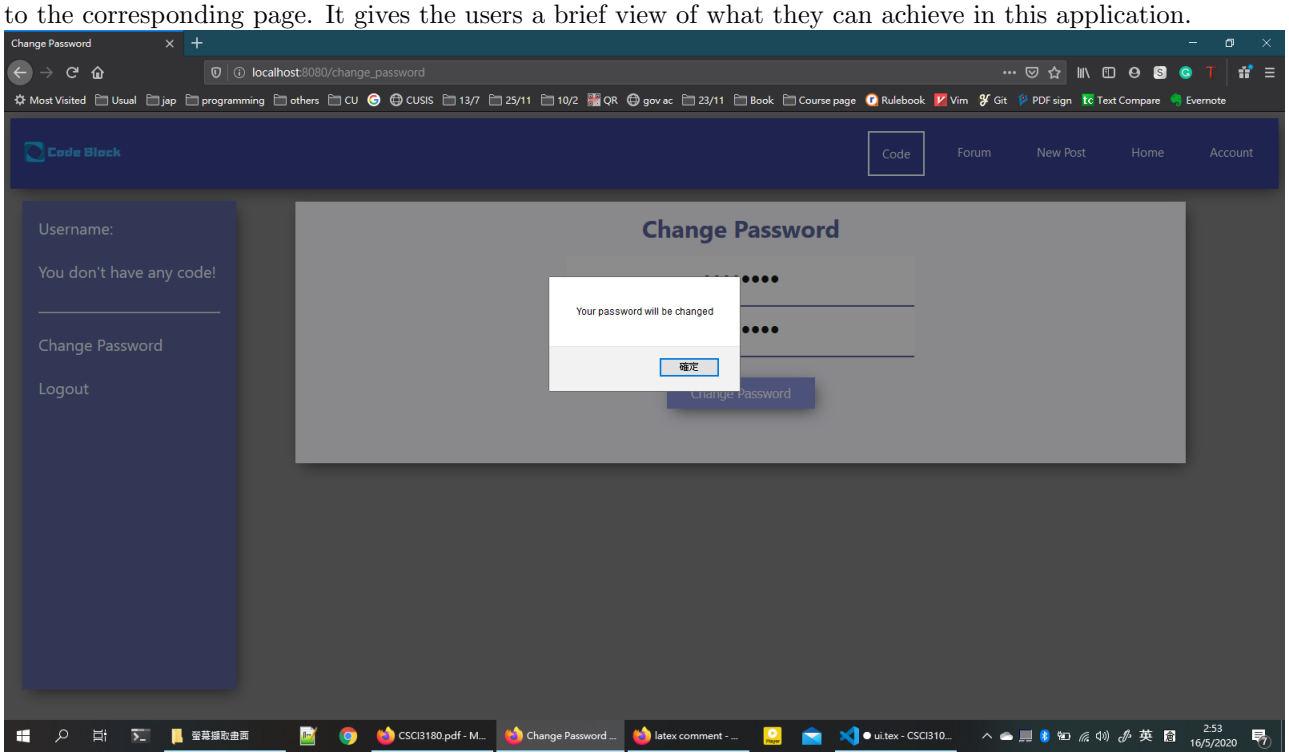
4.1.2 Form

There are in total of three filling forms for account management in the application. Instant feedback should be given to the users if the form cannot be submitted and process. For example, a warning should be displayed if the input is invalid, and the processing message should be displayed after the form is submitted.

Submittion Message

4.1.3 Home page

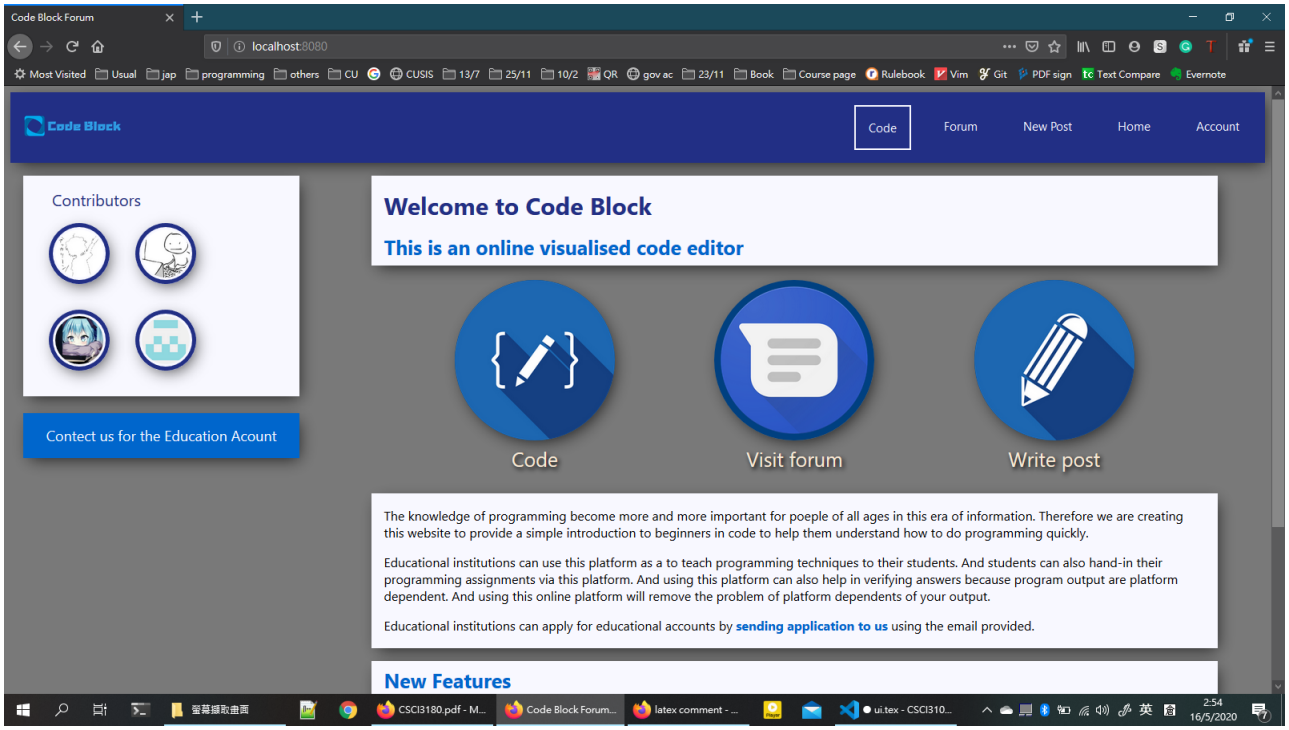
Eye-catching icons are placed on the home page. They indicate the main functions of the application and link to the corresponding page. It gives the users a brief view of what they can achieve in this application.



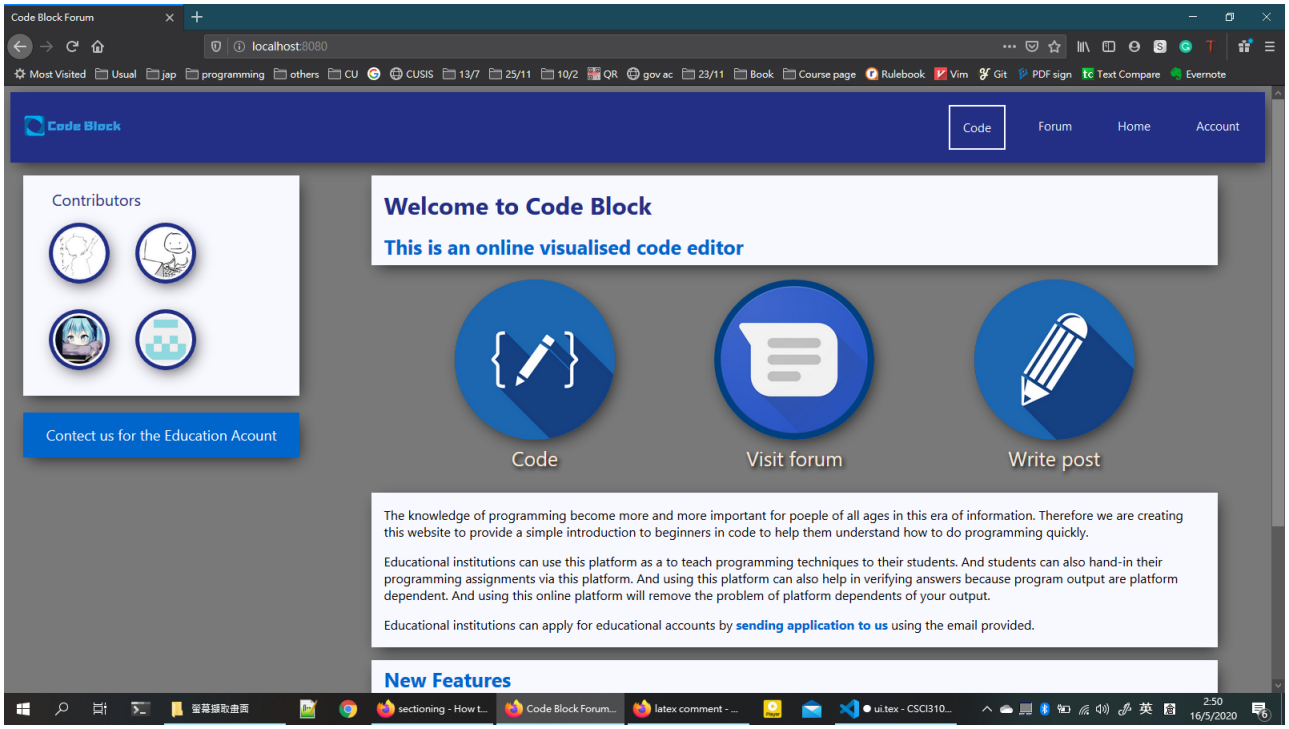
4.2 Screenshot

4.2.1 Home page

Login

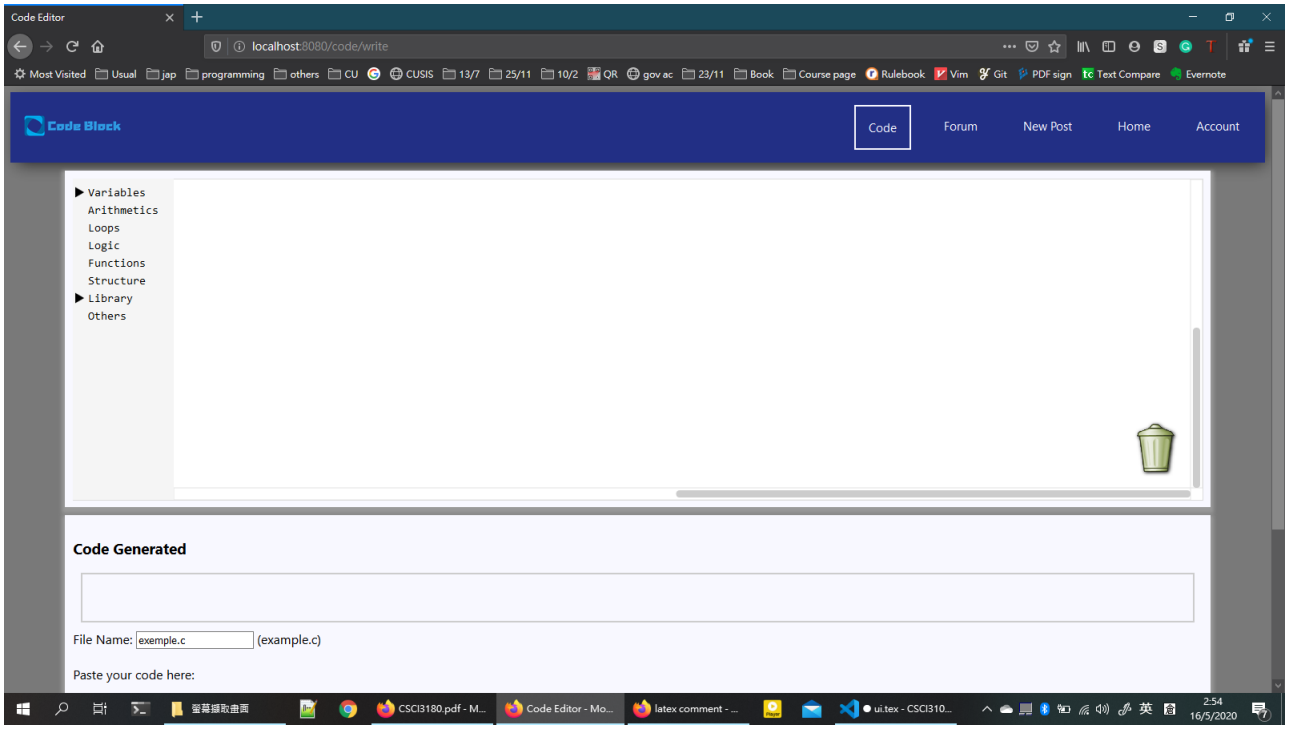


Logout

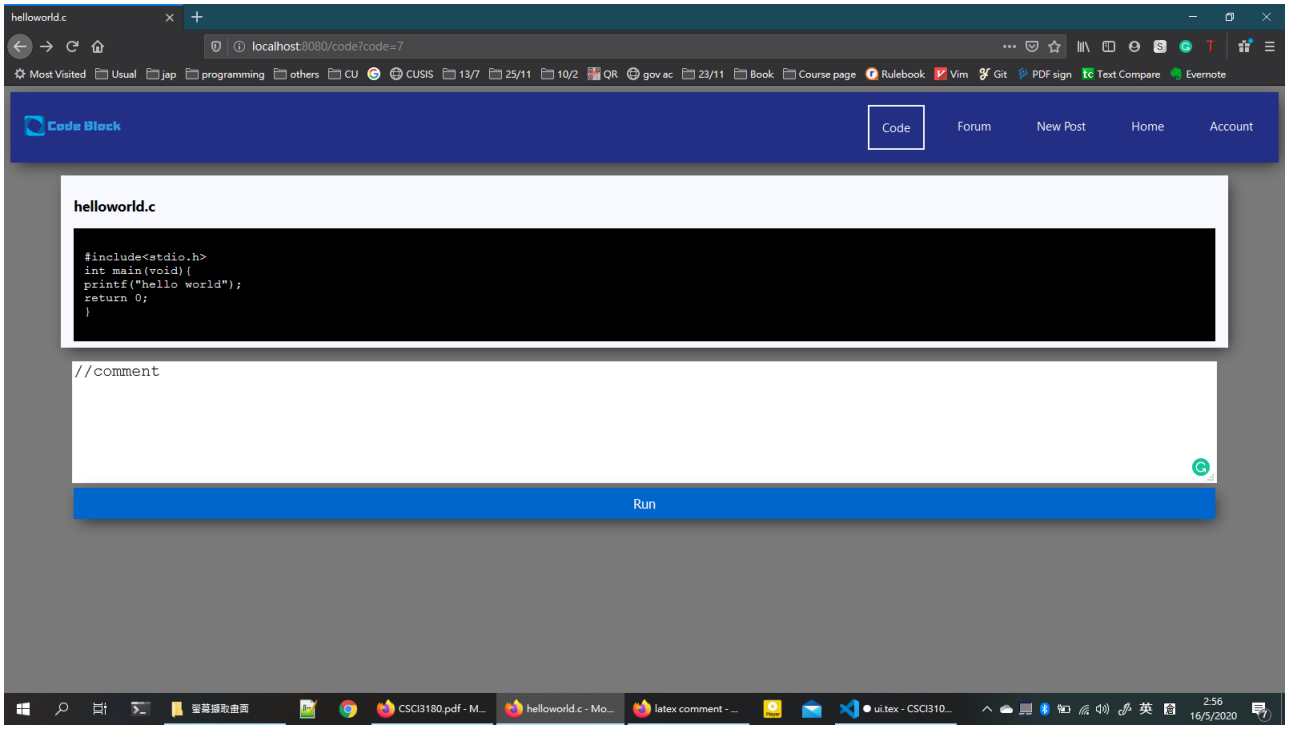


4.2.2 Code

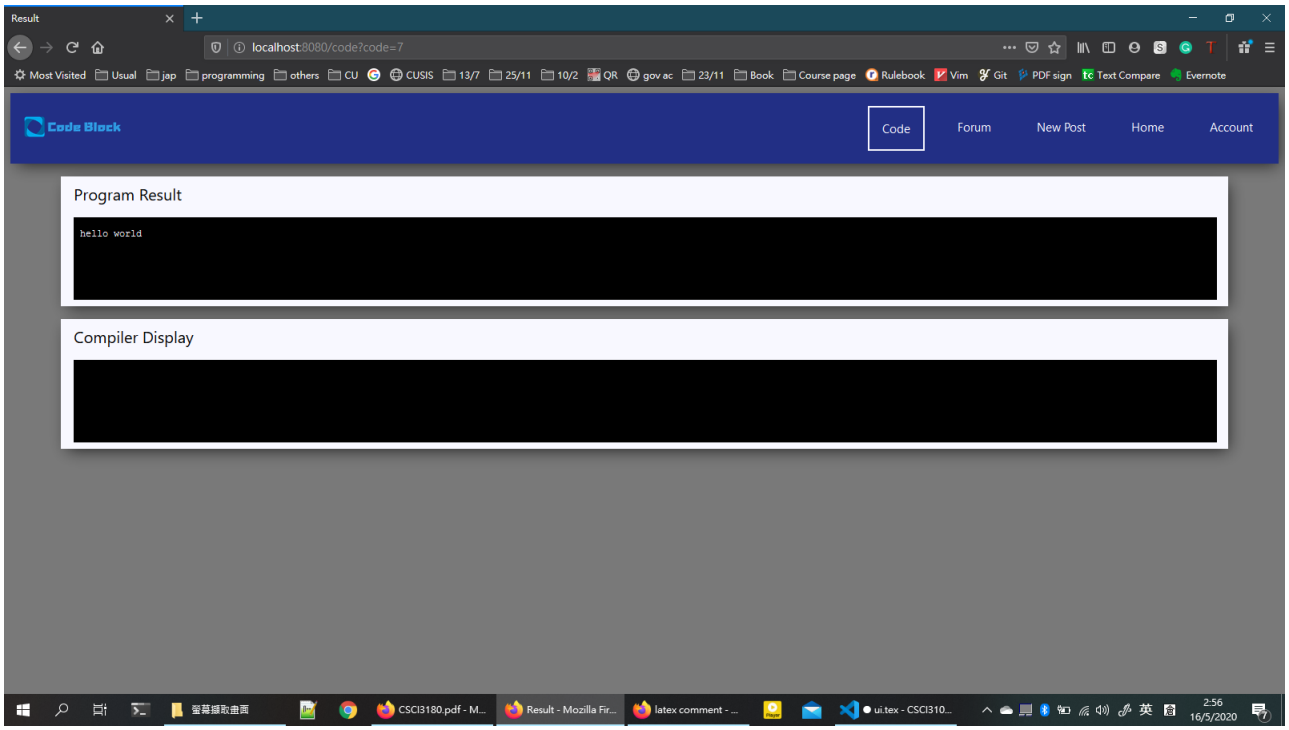
Coding Space



Generated Code

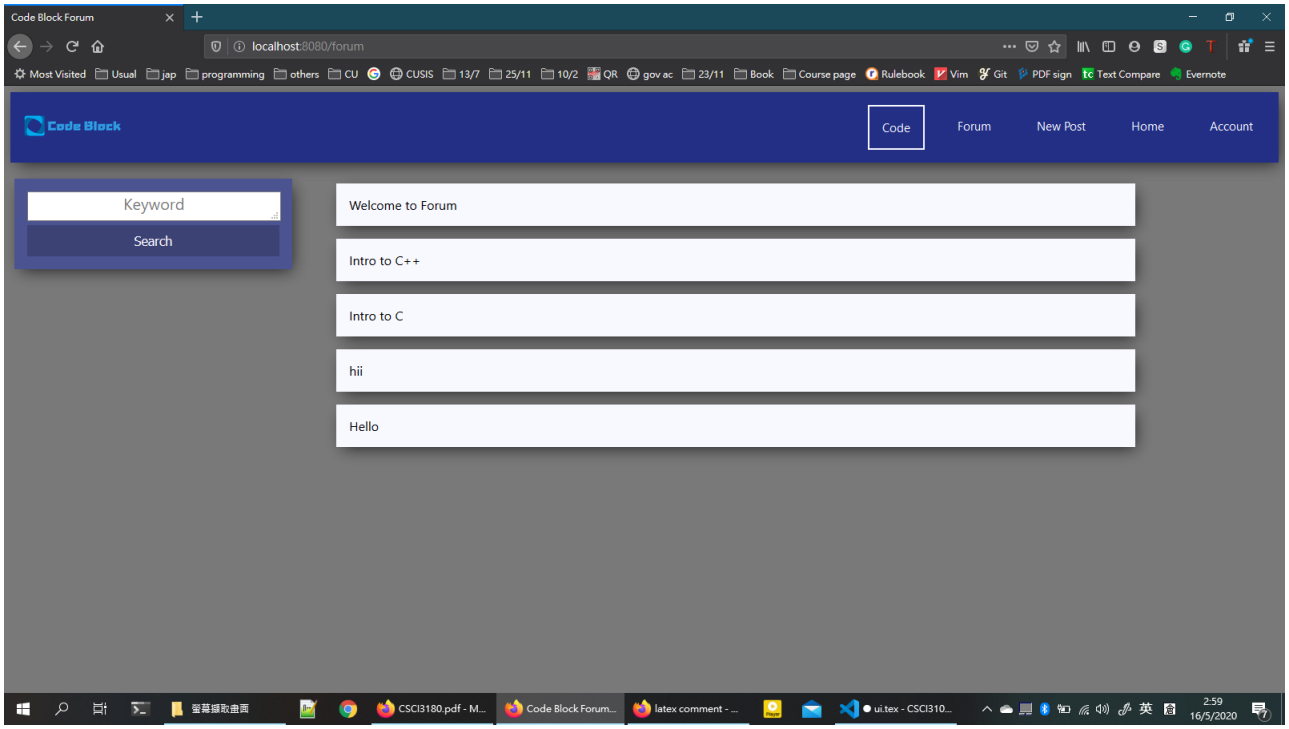


Compile Result

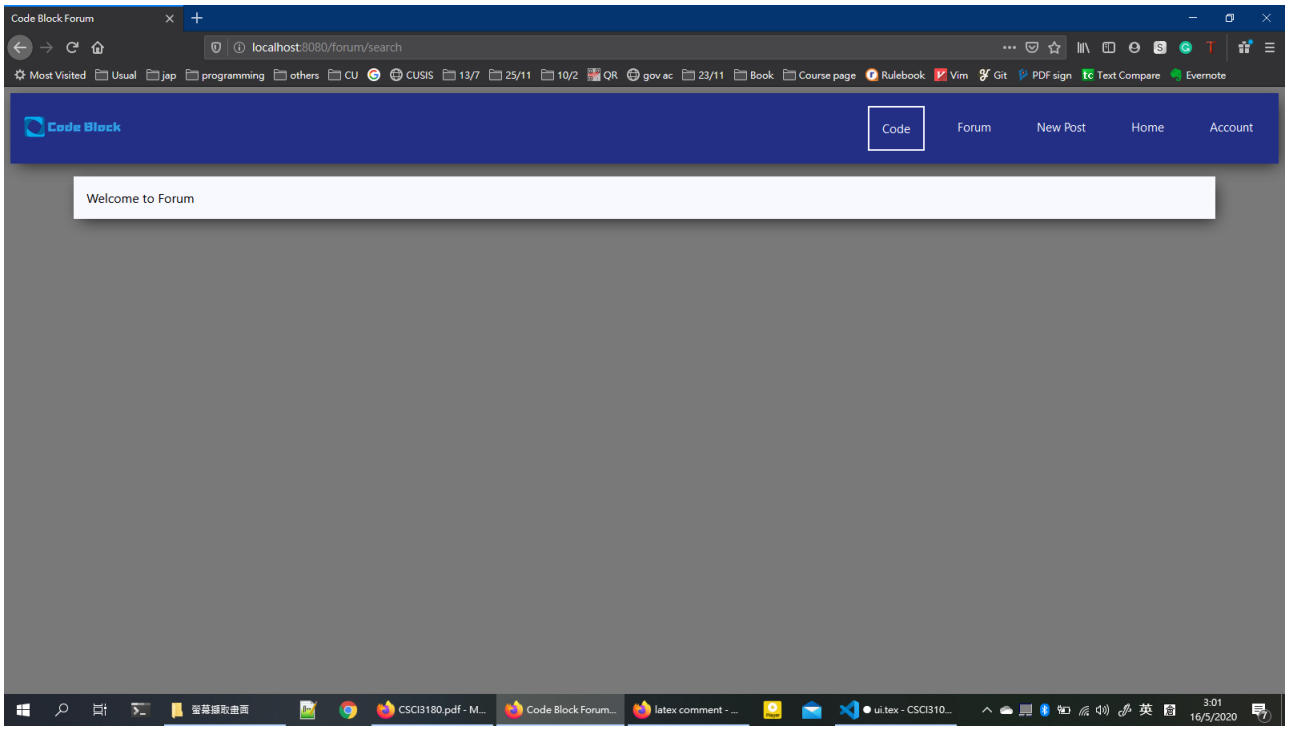


4.2.3 Forum

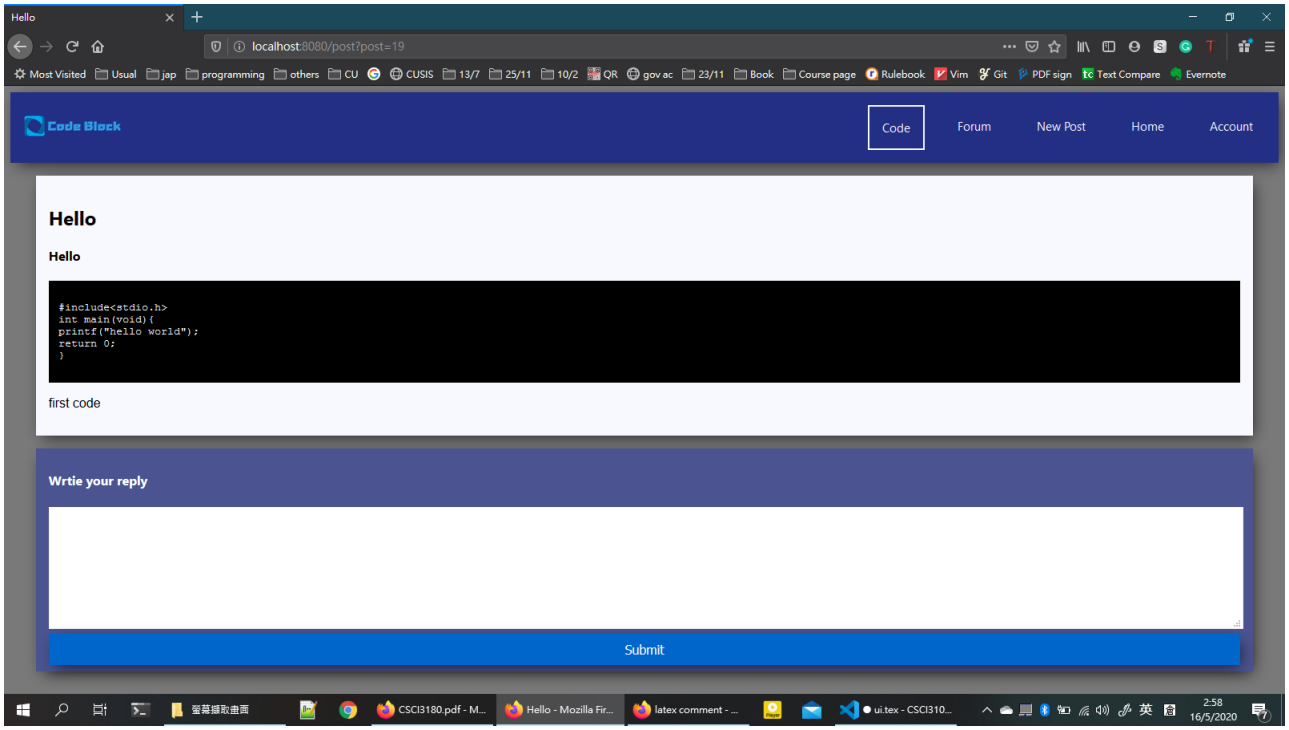
List of Posts

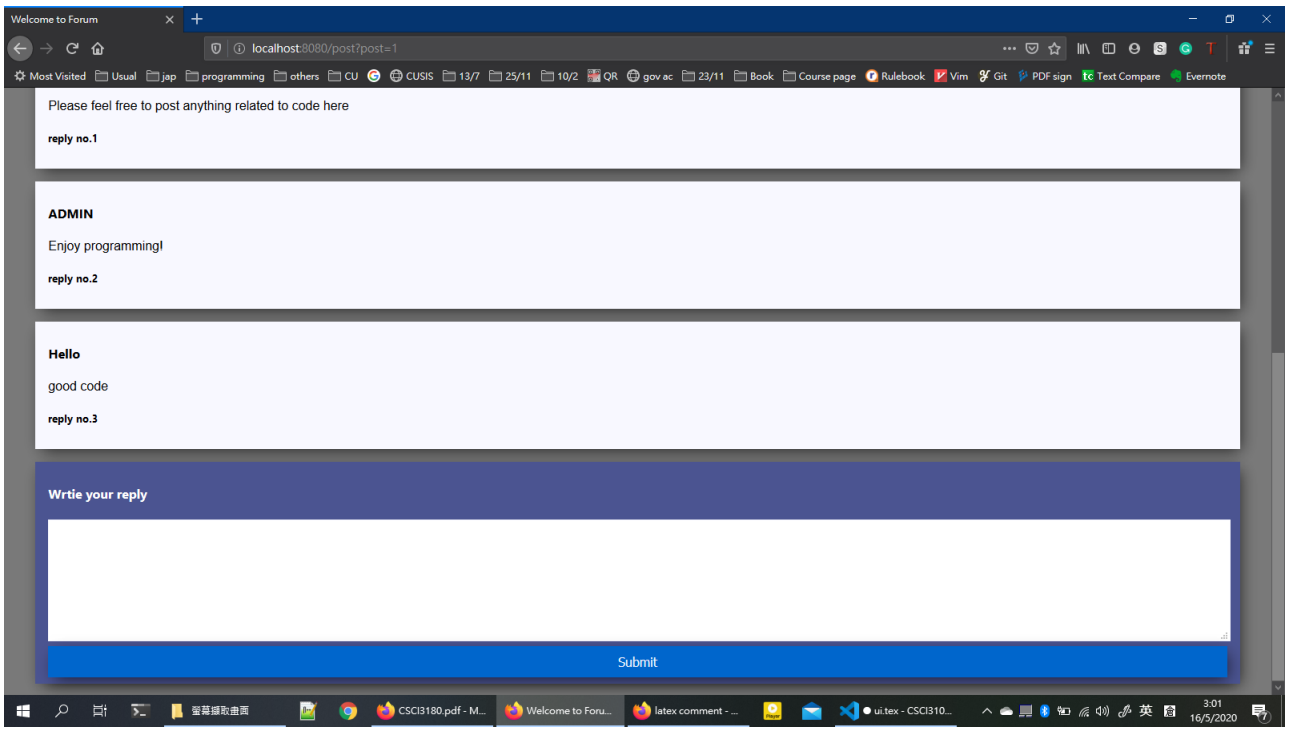


Search Result

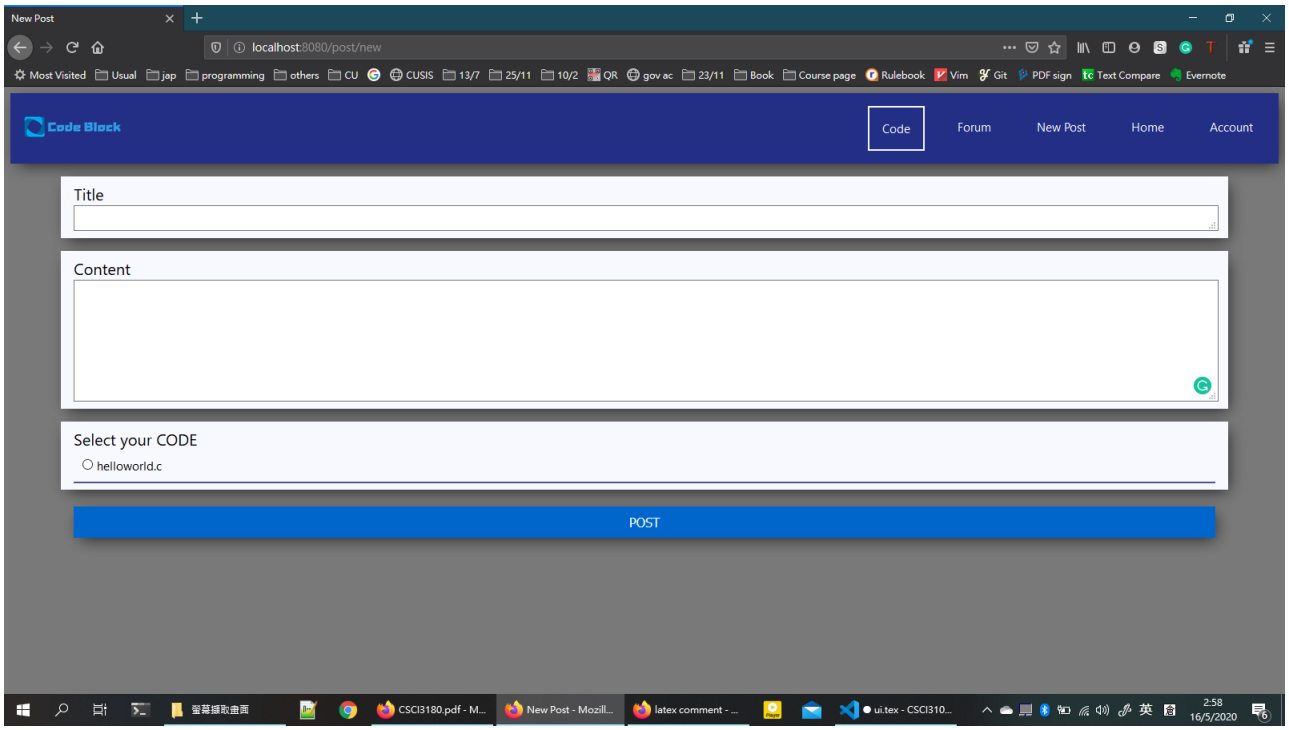


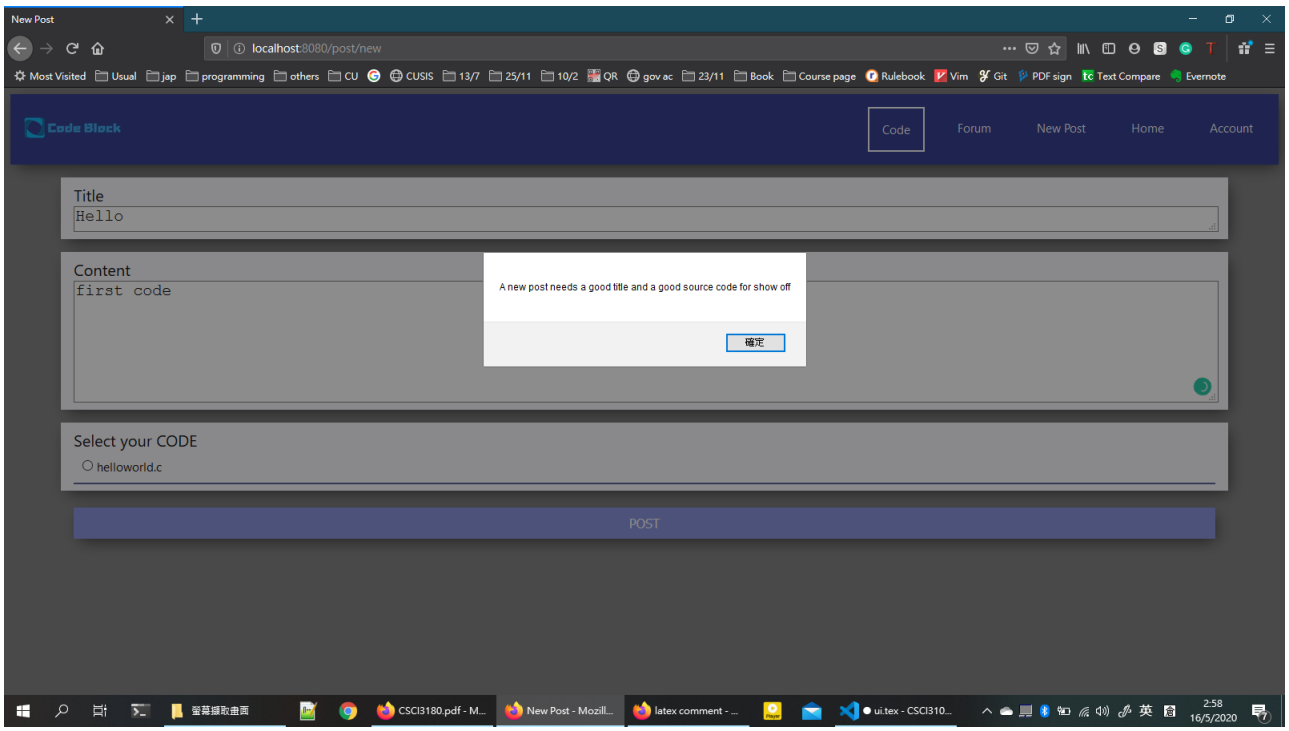
Post





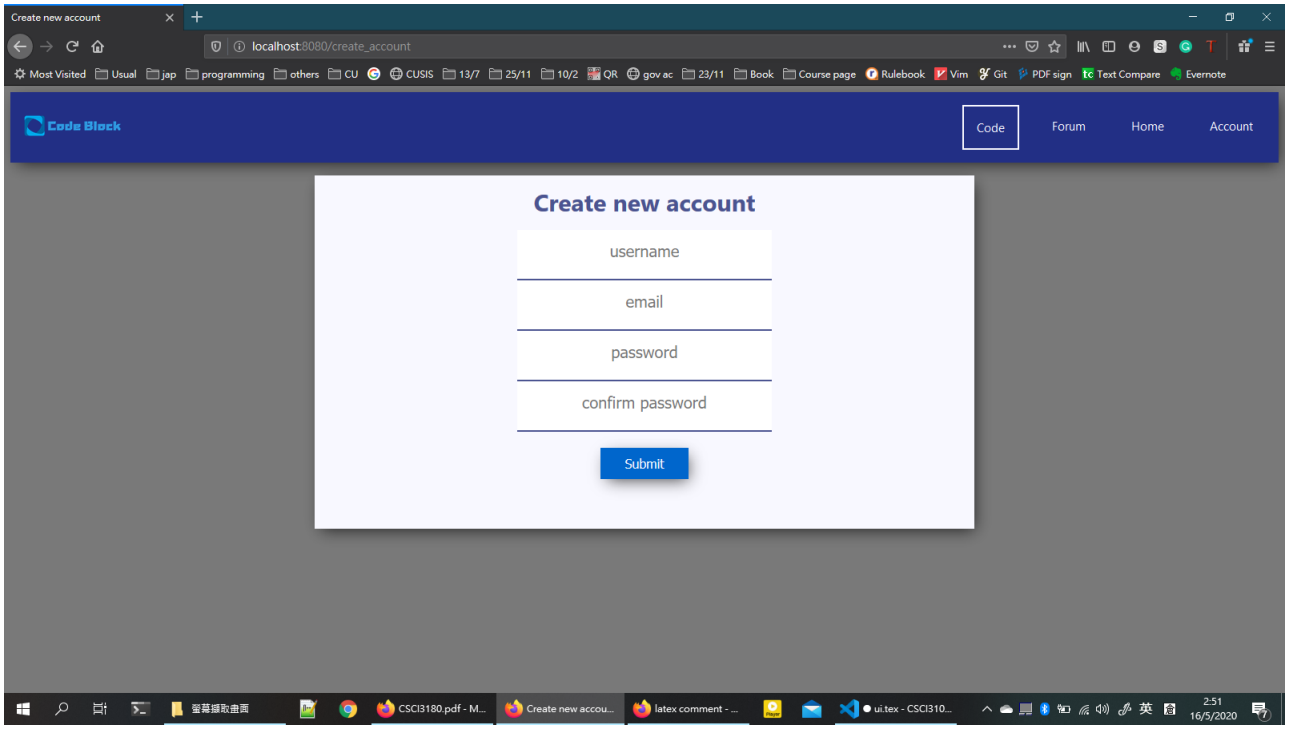
Forum - new post

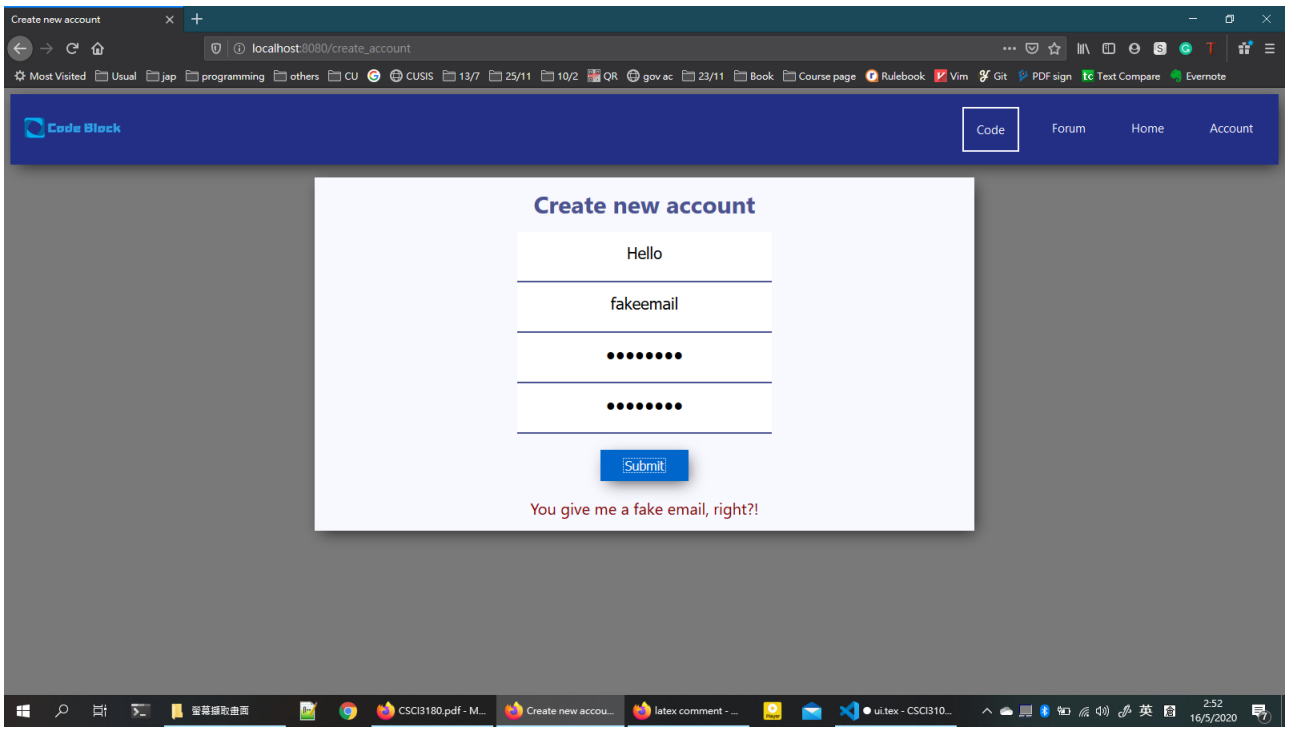




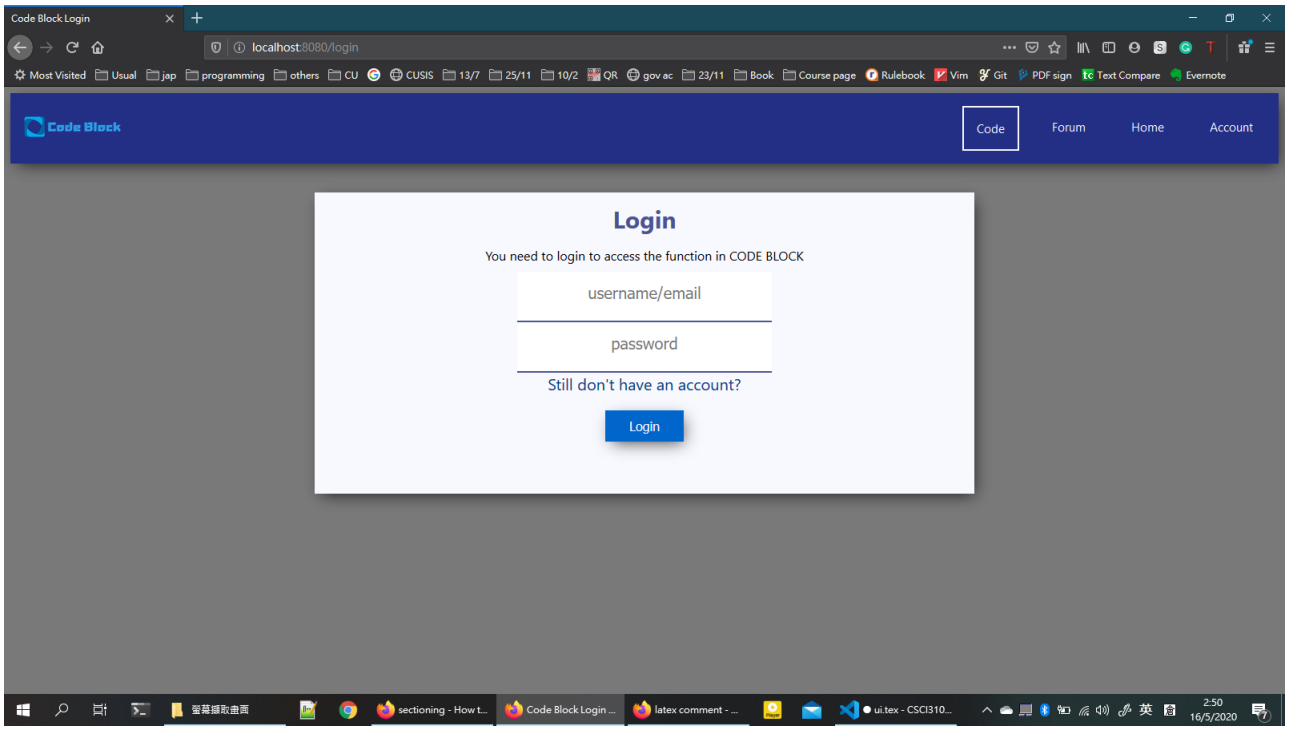
4.2.4 Account Managment

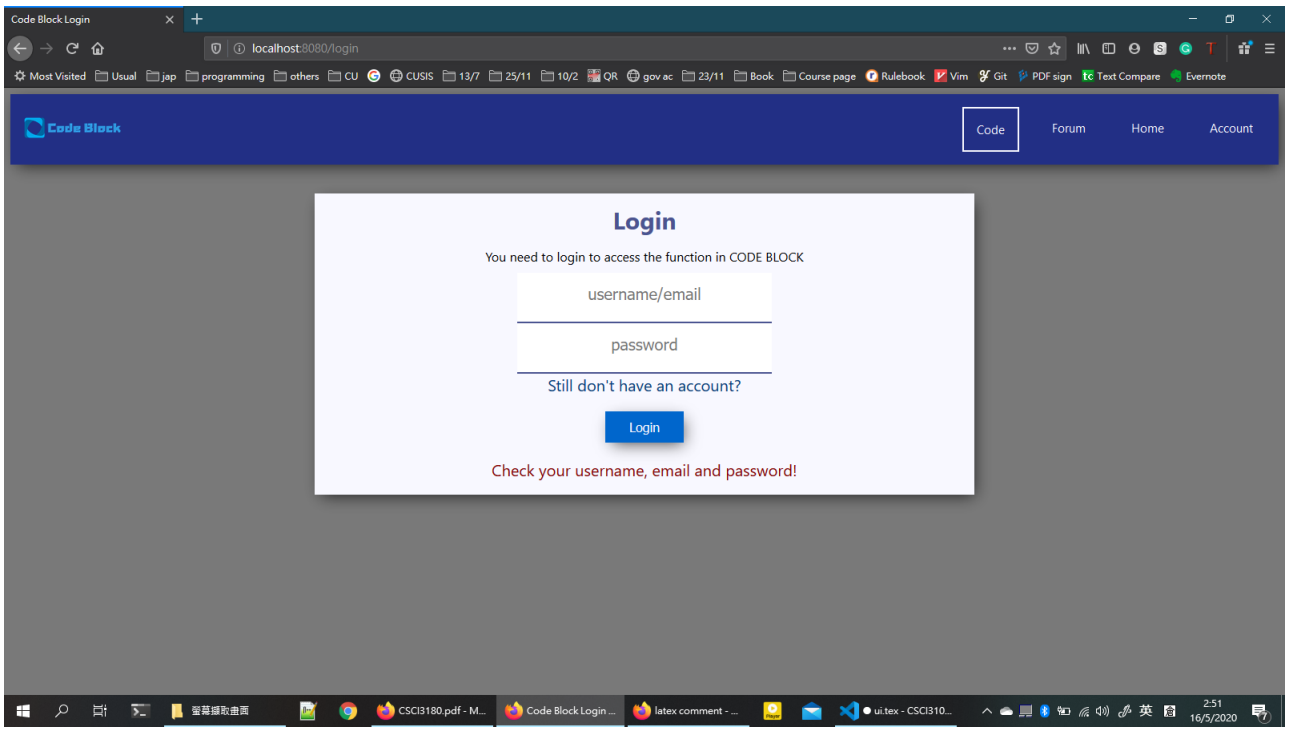
Registation



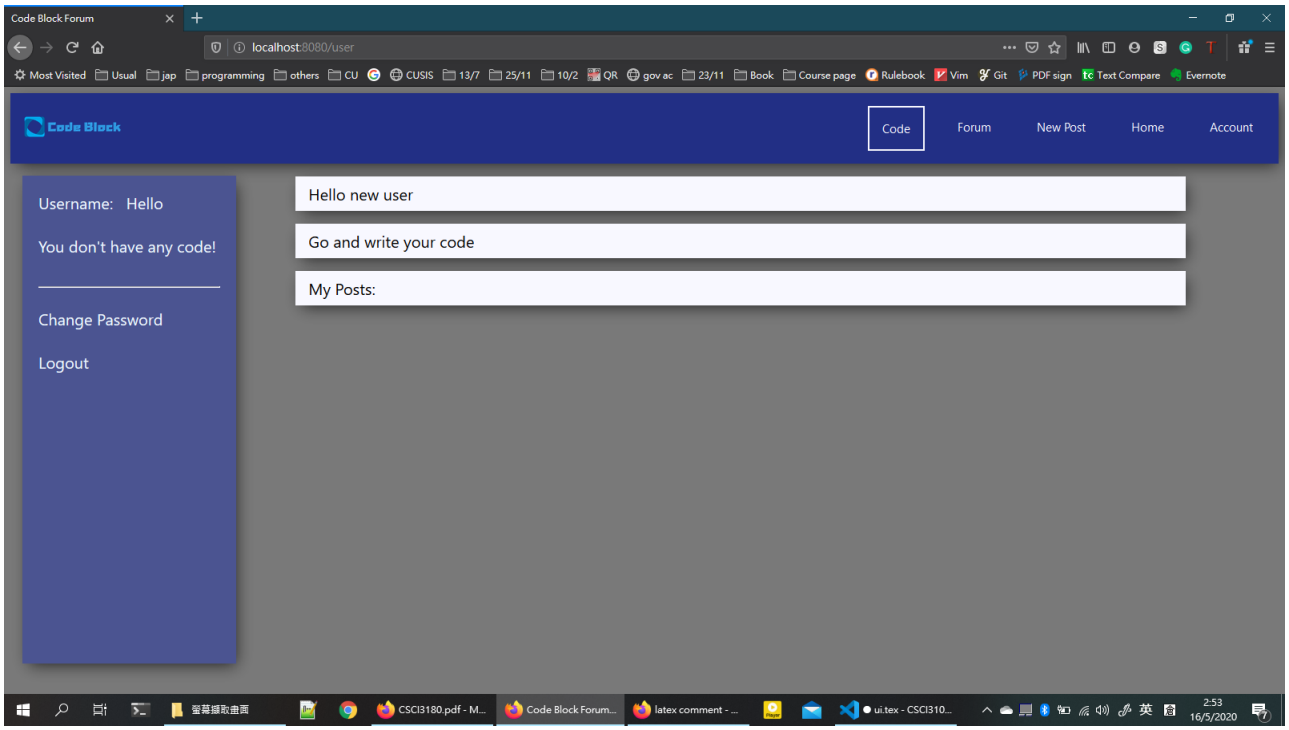

Create new account
localhost:8080/create_account
Code Block
Code
Forum
Home
Account
Create new account
Hello
fakeemail
Submit
You give me a fake email, right?!

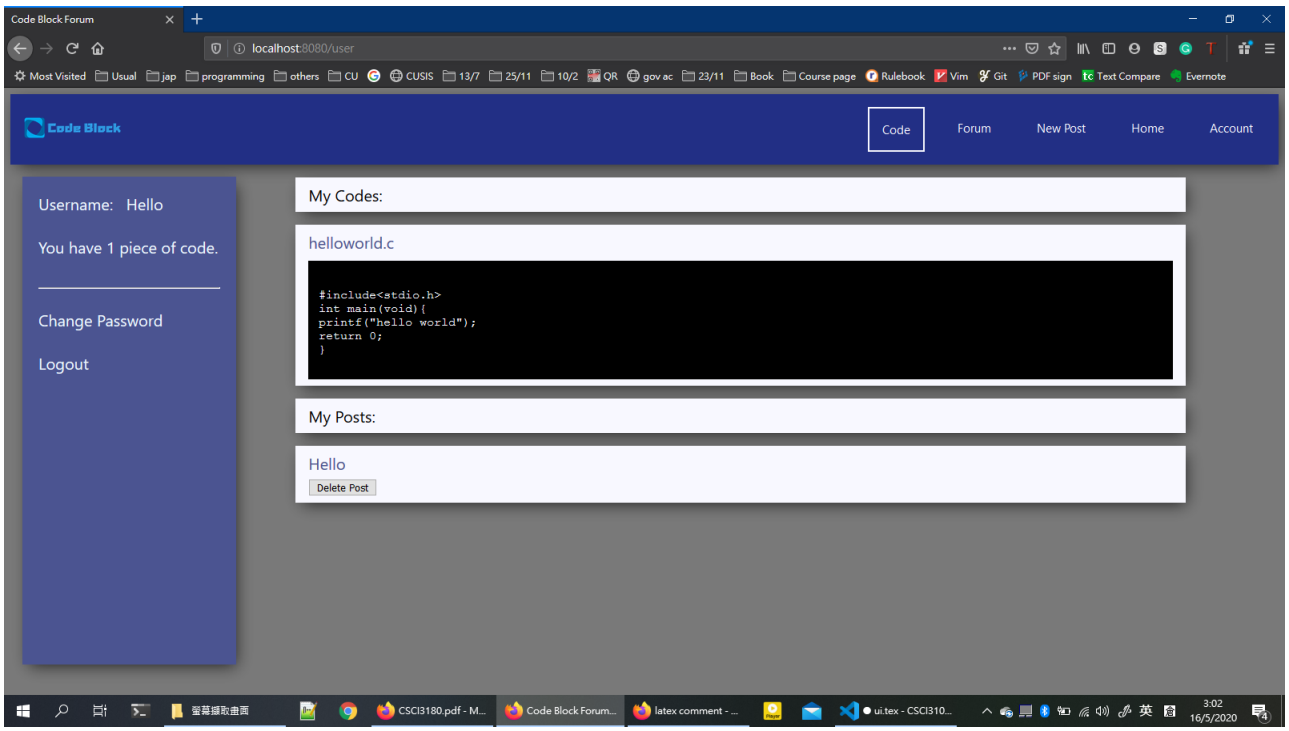
Login


Code Block Login
localhost:8080/login
Code Block
Code
Forum
Home
Account
Login
You need to login to access the function in CODE BLOCK
username/email
password
Still don't have an account?
Login

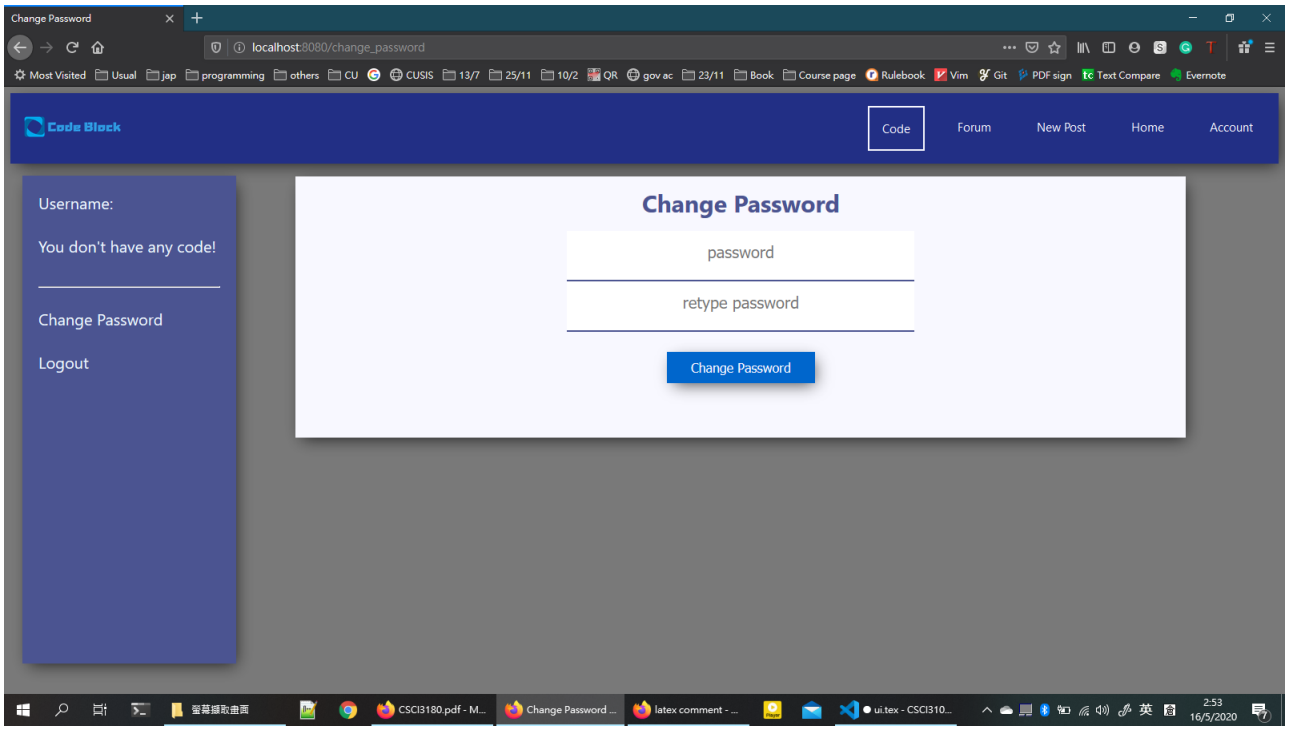


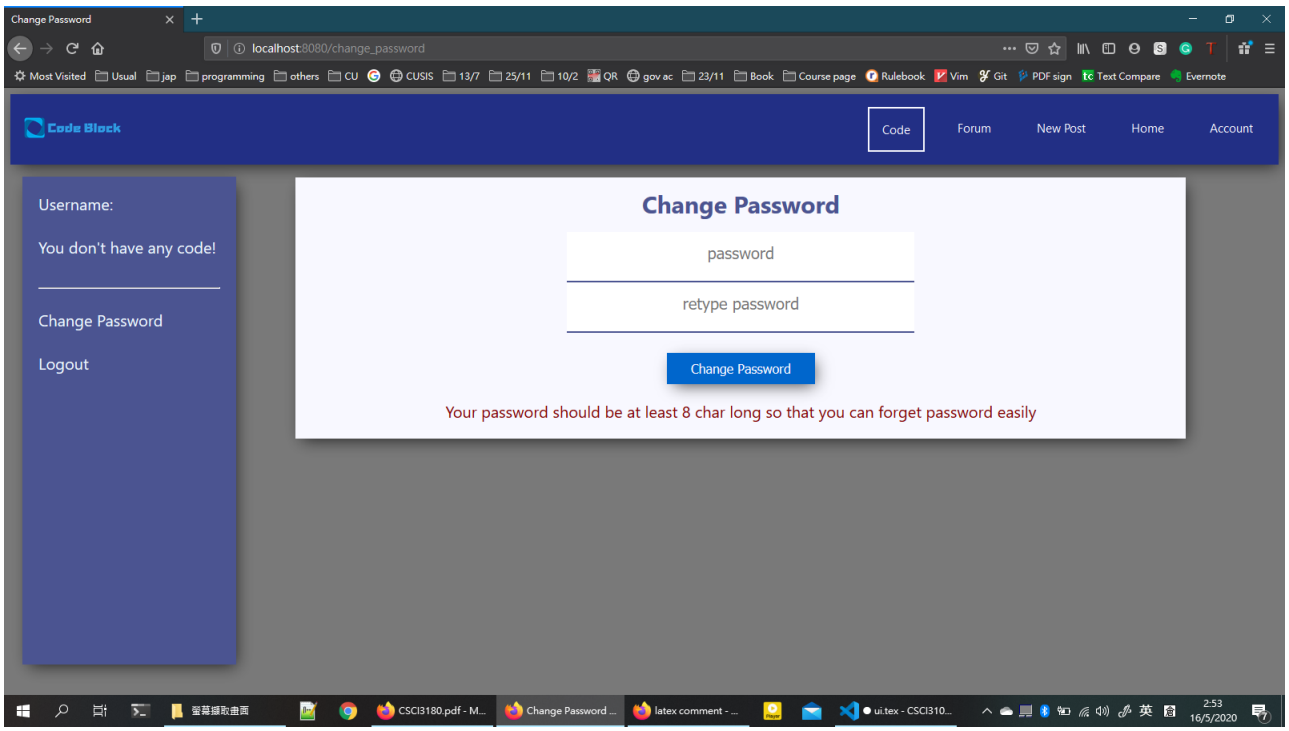
User Profile



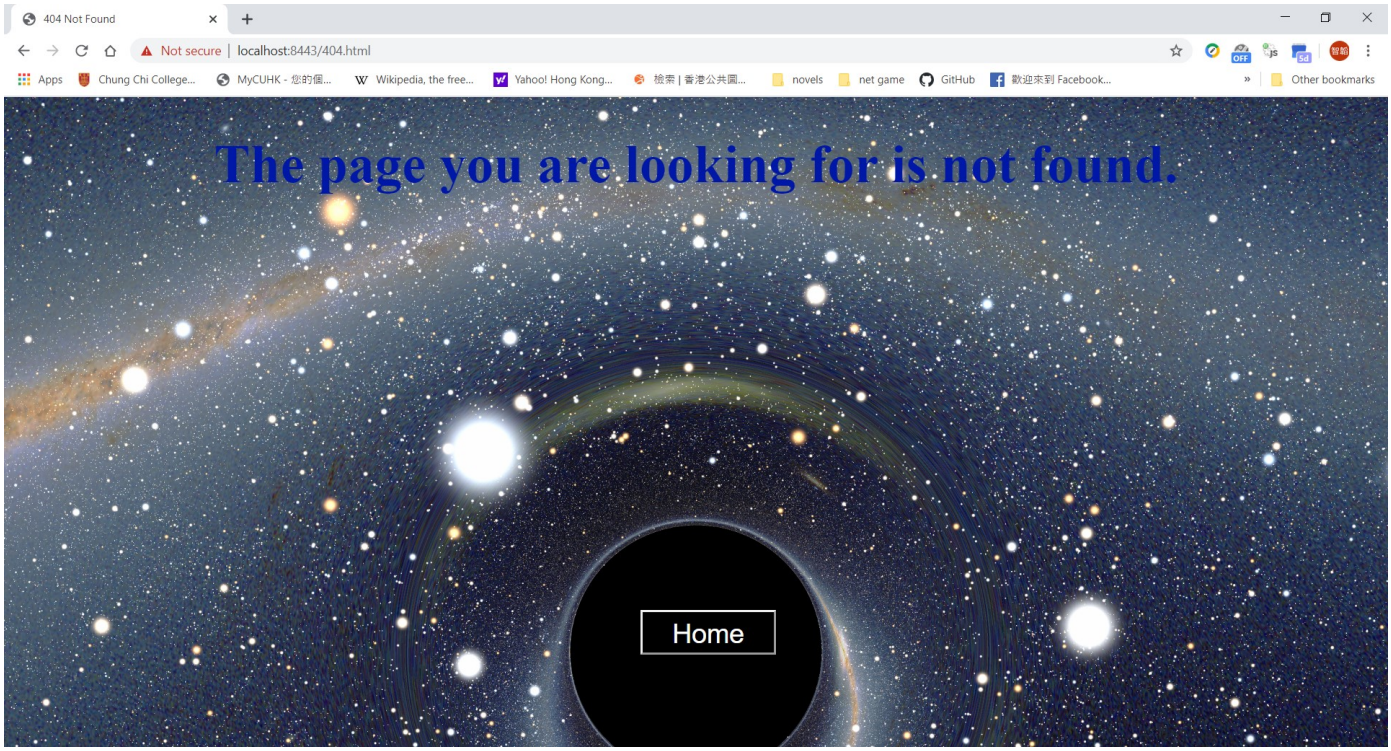


Change Password




Change Password
localhost:8080/change_password
Code Block
Code
Forum
New Post
Home
Account
Username:
You don't have any code!
Change Password
Logout
Change Password
password
retype password
Change Password
Your password should be at least 8 char long so that you can forget password easily

4.2.5 404 page


404 Not Found
Not secure | localhost:8443/404.html
The page you are looking for is not found.
Home

5

Test

5.1 Overview and Plan

The system is tested using bottom-up integration test. Bottom-up integration test is chosen for the system as the system relies heavily on two slave modules. Thus the correctness of the system relies heavily on the two slave modules. Module inputs and user activities are tested using black-box testing to ensure correctness of the system. The root server module (app.js) will be tested alongside with the website functions and its javascripts as http requests must be sent to the server to test its correctness. The test for root server module and website javascripts will be seperated according to the 4 features of the system, user account manipulations, code editing and saving, code compilation and forum.

5.2 MySQL Database Interface Module

5.2.1 Purpose

The connect_sql.js is tested in this testcase to ensure the correctness of this slave module that act as an interface to communicate between the server and the database. All child classes inherits and reuses the methods of parent class MySQLDatabase, therefore only the major functions of parent class is tested using methods of child classes

5.2.2 Inputs

Insert

Test cases of USER used (USERNAME, EMAIL, PASSWORD, ACC_TYPE):

1. newUser, newUser@localhost.net, 11111111, 0 (valid)
2. (empty), no@here, (empty), 0 (invalid empty input)
3. no, no, no, no (wrong ACC_TYPE type)

Select When All Conditions Are True

Test cases of SRC_CODE used (NAME, USER):

1. hello_world.c, 1 (valid)
2. (empty), 1 (empty name)
3. here, ADMIN (wrong USER type)

Delete When All Conditions Are True

Test cases of SRC_CODE used (NAME, USER)

1. hello-world.c, 1 (valid)
2. (empty), 1 (empty NAME)
3. here, ADMIN (wrong USER type)

Update

Test cases of USER used (PASSWORD, ID):

1. 88888888, 10 (valid)
2. (empty), 10 (empty PASSWORD)
3. hello_goodbye, newUser (wrong ID type)
4. 88888888, 0 (invalid ID)

5.2.3 Expected Outputs & Pass/Fail Criteria

The testcases demonstrates the return of the module when different input is provided, both valid and invalid queries to the MySQL database. The module should handle the exception and provide error message for unaccepted invalid testcases and return a "fail" message, and provide result for accepted valid or invalid testcases.

Insert

Case 1: (pass)

```
new_user
1
exist_name
```

Case 2: (pass)

```
new_user
0
error: ER_NO_DEFAULT_FOR_FIELD: Field 'PASSWORD' doesn't have a default value
fail
```

Case 3: (pass)

```
new_user
0
error: ER_TRUNCATED_WRONG_VALUE_FOR_FIELD: Incorrect integer value: 'no' for column '
    ACC_TYPE' at row 1
fail
```

Select When All Conditions Are True

Case 1: (pass)

```
fetch_code
[{"ID":1,"NAME":"hello_world.c","USER":1,"SRC":"#include <stdio.h>\n\nint main(void)\n{\n
    printf(\"Hello world\\n\");\n   return 0;\n}\n","SRC_SZ":130,"BLK":null,"BLK_SZ":3}]
```

Case 2: (pass)

```
fetch_code
[]
```

Case 3: (pass)

```
fetch_code
[]
```

Delete When All Conditions Are True

Case 1: (pass)

```
delete_code
success
```

Case 2: (pass)

```
delete_code
success
```

Case 3: (pass)

```
delete_code
error: ER_TRUNCATED_WRONG_VALUE: Truncated incorrect DOUBLE value: 'ADMIN'
fail
```

Update

Case 1: (pass)

```
update_user
update time
1
```

Case 2: (pass)

```
update_user
error
update time
error: ER_BAD_NULL_ERROR: Column 'PASSWORD' cannot be null
false
```

Case 3: (pass)

```
update_user
error
update time
error: ER_TRUNCATED_WRONG_VALUE: Truncated incorrect DOUBLE value: 'newUser'
false
```

Case 4: (pass)

```
update_user
update time
0
```

5.3 Source Code Compilation and Execution Module

5.3.1 Purpose

The compAndRun.js module contains 2 feature, invoke the computer the compile the C source code on the computer, and execute it if there is no error in compilation.
And the testcases observes the result of the compilation and execution.

5.3.2 Inputs

1. tmp_data/hello_world.c, 1 (valid)
2. hello_world.c, 1 (invalid path)
3. tmp_data/tmp.txt, 1 (text file instead of c source code file)
4. tmp_data/writer.c, 1 (compilation error)

5.3.3 Expected Outputs & Pass/Fail Criteria

The testcases are supposed to show the possible error message in the middle and show the compilation and execution result in an object at last.

Case 1: (pass)

```
tmp_data\hello_world.c
{"comp":"","prog":"hello world\r\n"}
```

Case 2: (pass)

```
hello_world.c
error: Command failed: gcc hello_world.c -o 1
gcc: error: hello_world.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.

stderr: gcc: error: hello_world.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.

{"comp":"gcc: error: hello_world.c: No such file or directory\ngcc: fatal error: no input
    files\ncompilation terminated.\r\n","prog":""}
```

Case 3: (pass)

```
tmp_data\tmp.txt
error: Command failed: gcc tmp_data\tmp.txt -o 1
C:/Program Files/mingw-w64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/bin/../lib/gcc/x86_64-
    w64-mingw32/8.1.0/../../../../x86_64-w64-mingw32/bin/ld.exe:tmp_data\tmp.txt: file
    format not recognized; treating as linker script
C:/Program Files/mingw-w64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/bin/../lib/gcc/x86_64-
    w64-mingw32/8.1.0/../../../../x86_64-w64-mingw32/bin/ld.exe:tmp_data\tmp.txt:1: syntax
    error
collect2.exe: error: ld returned 1 exit status

stderr: C:/Program Files/mingw-w64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/bin/../lib/gcc/
    x86_64-w64-mingw32/8.1.0/../../../../x86_64-w64-mingw32/bin/ld.exe:tmp_data\tmp.txt:
    file format not recognized; treating as linker script
C:/Program Files/mingw-w64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/bin/../lib/gcc/x86_64-
    w64-mingw32/8.1.0/../../../../x86_64-w64-mingw32/bin/ld.exe:tmp_data\tmp.txt:1: syntax
    error
collect2.exe: error: ld returned 1 exit status

{"comp":"C:/Program Files/mingw-w64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/bin/../lib/gcc
    /x86_64-w64-mingw32/8.1.0/../../../../x86_64-w64-mingw32/bin/ld.exe:tmp_data\\tmp.txt:
    file format not recognized; treating as linker script\r\nC:/Program Files/mingw-w64/
    x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/bin/../lib/gcc/x86_64-w64-mingw32
    /8.1.0/../../../../x86_64-w64-mingw32/bin/ld.exe:tmp_data\\tmp.txt:1: syntax error\r\
    ncollect2.exe: error: ld returned 1 exit status\n","prog":""}
```

Case 4: (pass)

```
tmp_data\writer.c
error: Command failed: gcc tmp_data\writer.c -o 1
tmp_data\writer.c: In function 'main':
tmp_data\writer.c:5:18: error: expected ';' before '}' token
   printf("hello")
                  ^
                  ;
 }
 ~

stderr: tmp_data\writer.c: In function 'main':
tmp_data\writer.c:5:18: error: expected ';' before '}' token
   printf("hello")
                  ^
                  ;
 }
 ~

{"comp":"tmp_data\\writer.c: In function 'main':\ntmp_data\\writer.c:5:18: error: expected
    ';' before '}' token\n   printf(\"hello\")\n                  ^\n                  ;\n
```

```
}\n ~                    \n","prog":""}
```

5.4 Code Handling Module

5.4.1 Purpose

The code.js module allows save code and compile and execution of source code. Extension names and file names are not concerned by the server.
The testcases observes the result of valid code names and invalid code names when saving and correct code and incorrect code in compilation. The correctness of code fetching will be tested with code compilation as code compilation uses code fetching for source code.

5.4.2 Inputs

Save Code

Testcases used (userid, filename, source code):

1. 10, hello.c, `#include <stdio.h>\nint main(void)\n{\nprintf("hello world\n");\nreturn 0;\n}` (valid, new code)
2. 4, hello.c, `#include <stdio.h>\nint main(void)\n{\nprintf("hello world\n");\nreturn 0;\n}` (invalid userid)
3. 10, hello.c, `#include <stdio.h>\nint main(void)\n{\nprintf("hello world\n");\nreturn 0;\n}` (valid, update code)
4. 10, (empty string), `#include <stdio.h>\nint main(void)\n{\nprintf("hello world\n");\nreturn 0;\n}` (invalid name)

Compile and Run

Testcases used (code id, stdin):

1. 12, `""` (valid)
2. 5, `"98 56\r\n"` (valid with stdin)
3. 10, `""` (nonexistent code)
4. 4, `""` (wrong code)
5. 5, `""` (valid code with invalid input)

5.4.3 Expected Outputs & Pass/Fail Criteria

Save Code

The testcases allows observation of the behaviour of the methos on valid and invalid inputs.
A number indicates a new file created, success indicates successfully updated, and fail indicates failure.

Case 1: (pass)

```
save_code
12
```

Case 2: (pass)

```
save_code
error: ER_NO_REFERENCED_ROW: Cannot add or update a child row: a foreign key constraint
    fails
fail
```

Case 3: (pass)

```
save_code
success
```

Case 4: (pass)

```
fail
```

Compile and Run

The testcases allows us to observe the behaviour of the fetch method and compile and run methos on valid and invalid inputs.
The return of the function should return an object with compilation result and program output on success, and false on failure.

Case 1: (pass)

```
fetch_code
tmp_data\10_12.c
error: Command failed: gcc tmp_data\10_12.c -o 12
tmp_data\10_12.c:1:19: warning: extra tokens at end of #include directive
 #include <stdio.h>\nint main(void)\n{\nprintf("hello world\n");\nreturn 0;\n}
                   ^
C:/Program Files/mingw-w64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/bin/../lib/gcc/x86_64-
    w64-mingw32/8.1.0/../../../../x86_64-w64-mingw32/lib/../lib/libmingw32.a(
    lib64_libmingw32_a-crt0_c.o):crt0_c.c:(.text.startup+0x2e): undefined reference to `
    WinMain'
collect2.exe: error: ld returned 1 exit status

stderr: tmp_data\10_12.c:1:19: warning: extra tokens at end of #include directive
 #include <stdio.h>\nint main(void)\n{\nprintf("hello world\n");\nreturn 0;\n}
                   ^
C:/Program Files/mingw-w64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/bin/../lib/gcc/x86_64-
    w64-mingw32/8.1.0/../../../../x86_64-w64-mingw32/lib/../lib/libmingw32.a(
    lib64_libmingw32_a-crt0_c.o):crt0_c.c:(.text.startup+0x2e): undefined reference to `
    WinMain'
collect2.exe: error: ld returned 1 exit status

{
  comp: 'tmp_data\\10_12.c:1:19: warning: extra tokens at end of #include directive\n' +
    ' #include <stdio.h>\\nint main(void)\\n{\\nprintf("hello world\\n");\\nreturn 0;\\n}\n'
     +
    '                   ^\n' +
    "C:/Program Files/mingw-w64/x86_64-8.1.0-posix-seh-rt_v6-rev0/mingw64/bin/../lib/gcc/
    x86_64-w64-mingw32/8.1.0/../../../../x86_64-w64-mingw32/lib/../lib/libmingw32.a(
    lib64_libmingw32_a-crt0_c.o):crt0_c.c:(.text.startup+0x2e): undefined reference to `
    WinMain'\r\n" +
    'collect2.exe: error: ld returned 1 exit status\n',
  prog: ''
}
```

Case 2: (pass)

```
fetch_code
tmp_data\1_5.c
{ comp: '', prog: 'GCD of 98 and 56 is 14 ' }
```

Case 3: (pass)

```
fetch_code
false
```

Case 4: (pass)

```
fetch_code
tmp_data\1_4.c
error: Command failed: gcc tmp_data\1_4.c -o 4
tmp_data\1_4.c:1:10: fatal error: iostream: No such file or directory
 #include <iostream>
```

```
          ~~~~~~~~~~
compilation terminated.

stderr: tmp_data\1_4.c:1:10: fatal error: iostream: No such file or directory
 #include <iostream>
          ~~~~~~~~~~
compilation terminated.

{
  comp: 'tmp_data\\1_4.c:1:10: fatal error: iostream: No such file or directory\n' +
    ' #include <iostream>\n' +
    '          ~~~~~~~~~~\n' +
    'compilation terminated.\r\n',
  prog: ''
}
```

Case 5: (pass, random output with invalid stdin)

```
fetch_code
tmp_data\1_5.c
{ comp: '', prog: 'GCD of 0 and 16 is 16 ' }
```

5.5 Forum Module

5.5.1 Purpose

The forum.js handles forum related requests, including fetch post, post posts and replies, search posts, removing posts by authenticated user and fetching basic data of all valid posts in the forum.
The testcases observes the result of each major feature of the module upon valid and invalid posts, and valid and invalid users.

5.5.2 Input

Fetch Post

Used testcases (post ID):

1. 1 (valid, post)
2. 2 (valid, reply)
3. 3 (nonexistent record)
4. -1 (invalid ID)
5. 0 (invalid ID)

Post Post

Used testcases (title, content, code ID):

1. new entry, `something here\r\nand hello world`, 1 (valid)
2. My first code, `here`, 1 (title collision of another post by different person)
3. My first code, `here`, 1 (title collision of another post by same person)
4. (empty string), (empty string), 1 (empty string)
5. first code, `here`, 0 (invalid code ID)
6. My first code, `here`, 1 (invalid user)

Post Reply

Used testcases (post ID, content):

1. 39, `here\r\nhere` (valid)
2. 43, `here\r\nhere` (reply pointing to a reply, but not post)
3. 3, `here\r\nhere` (reply pointing to nonexistent entry)
4. 39, (empty string) (empty reply)
5. 1, `here\r\nhere` (invalid user)

Remove Post

Used testcases (post ID):

1. 39 (valid)
2. 3 (nonexistent entry)
3. 38 (entry not owned)

Search

User testcases (keyword):

1. ADMIN (valid username)
2. C++ (valid titile keyword)
3. C++ C (valid multiple title keyword with no occurence)
4. is (valid content keyword)
5. #* (valid keyword with universal sign with no occurence)
6. C* (valid title keyword with universal sign)

Title Retrieval

This function requires no specific input

5.5.3 Expected Outputs & Pass/Fail Criteria

Fetch Post

The method should return post with details (author, content, title, code) and all its replies with its author and content

Case 1: (pass)

```
fetch_post
0
[
  {
    ID: 1,
    USER: 'ADMIN',
    TITLE: 'Welcome to Forum',
    CONTENT: 'Welcome to forum, you can post and comment on others code freely',
    CODE: '#include <stdio.h>\n' +
      '\n' +
      'int main(void)\n' +
      '{\n' +
      '   printf("Hello world\\n");\n' +
      '   return 0;\n' +
      '}\n'
```

```
  },
  {
    USER: 'ADMIN',
    CONTENT: 'Please feel free to post anything related to code here'
  },
  { USER: 'ADMIN', CONTENT: 'Enjoy programming!' },
  { USER: 'ADMIN', CONTENT: 'test\r\ntest again\r\ngood program' },
  { USER: 'ADMIN', CONTENT: '' }
]
```

Case 2: (pass)

```
fetch_post
0
[]
```

Case 3: (pass)

```
fetch_post
0
[]
```

Case 4: (pass)

```
fetch_post
0
[]
```

Case 5: (pass)

```
fetch_post
0
[
  {
    USER: 'ADMIN',
    CONTENT: 'Welcome to forum, you can post and comment on others code freely'
  },
  {
    USER: 'ADMIN',
    CONTENT: 'Website for introduction to C++\n' +
      'https://www.tutorialspoint.com/cplusplus/index.htm\n' +
      "Well, there are plenty of guides available on google also, so pls google it if you
    don't like it"
  },
  {
    USER: 'ADMIN',
    CONTENT: 'Again, tutorialspoint provide a good introduction to programming in C.\r\n' +
      'https://www.tutorialspoint.com/cprogramming/index.htm'
  },
  { USER: 'ADMIN', CONTENT: 'What comment should I write' },
  { USER: 'ADMIN', CONTENT: 'find the gcd of two int' },
  { USER: 'TEST', CONTENT: 'Hi' },
  { USER: 'helloworld', CONTENT: 'hello world' }
]
```

For case 5, although the behaviour of it was unexpected, and returns all post heads, but it follows the return format required, therefore it is considered to have the test passed

Post Post

The method should return post id upon success.

Case 1: (pass)

```
new_post
39
```

Case 2: (pass)

```
new_post
40
```

Case 3: (pass)

```
new_post
41
```

Case 4: (pass)

```
new_post
42
```

Case 5: (pass)

```
new_post
error: ER_NO_REFERENCED_ROW: Cannot add or update a child row: a foreign key constraint
    fails
fail
```

Case 6: (pass)

```
new_post
error: ER_NO_REFERENCED_ROW: Cannot add or update a child row: a foreign key constraint
    fails
fail
```

Post Reply

The method should return post id of the reply upon success

Case 1: (pass)

```
new_post
46
```

Case 2: (fail)

```
new_post
47
```

Case 3: (fail)

```
new_post
48
```

Case 4: (pass)

```
new_post
49
```

Case 5: (pass)

```
new_post
error: ER_NO_REFERENCED_ROW: Cannot add or update a child row: a foreign key constraint
    fails
fail
```

Although testcase 2, 3 failed, but that should not be considered as a bug as it will not affect the user. It should be considered as a feature which let users to leave invisible posts.

Remove Post

The method should return success or fail to indicate the result of action.

Case 1: (pass)

```
delete_post
success
```

Case 2: (pass)

```
delete_post
success
```

Case 3: (pass)

```
delete_post
fail
```

Search

The method should return related title, author and post id of related posts upon success.

Case 1: (pass)

```
search_post
0
[
  {
    ID: 1,
    USER: 'abc',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'abc',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 20,
    USER: 'abc',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  },
  {
    ID: 1,
    USER: 'ADMIN',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'ADMIN',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 13,
    USER: 'ADMIN',
    TITLE: 'Intro to C',
    CREATE_TIME: '2020-04-22T07:06:22.000Z'
  },
  {
    ID: 20,
    USER: 'ADMIN',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  },
  {
    ID: 25,
```

```
    USER: 'ADMIN',
    TITLE: 'gcd in C',
    CREATE_TIME: '2020-04-23T14:37:50.000Z'
  },
  {
    ID: 1,
    USER: 'helloworld',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'helloworld',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 20,
    USER: 'helloworld',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  },
  {
    ID: 1,
    USER: 'newUser',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'newUser',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 20,
    USER: 'newUser',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  },
  {
    ID: 1,
    USER: 'TEST',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'TEST',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 20,
    USER: 'TEST',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  }
]
```


Case 2: (pass)

```
search_post
0
[
  {
    ID: 1,
    USER: 'abc',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'abc',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 20,
    USER: 'abc',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  },
  {
    ID: 1,
    USER: 'ADMIN',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'ADMIN',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 20,
    USER: 'ADMIN',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  },
  {
    ID: 1,
    USER: 'helloworld',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'helloworld',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 20,
    USER: 'helloworld',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  },
  {
    ID: 1,
    USER: 'newUser',
    TITLE: 'Welcome to Forum',
```

```
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'newUser',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 20,
    USER: 'newUser',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  },
  {
    ID: 1,
    USER: 'TEST',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'TEST',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 20,
    USER: 'TEST',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  }
]
```

Case 3: (pass)

```
search_post
0
[
  {
    ID: 1,
    USER: 'abc',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'abc',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 20,
    USER: 'abc',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  },
  {
    ID: 1,
    USER: 'ADMIN',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
```

```
},
{
  ID: 4,
  USER: 'ADMIN',
  TITLE: 'Intro to C++',
  CREATE_TIME: '2020-04-19T16:35:34.000Z'
},
{
  ID: 20,
  USER: 'ADMIN',
  TITLE: 'CSCI3100',
  CREATE_TIME: '2020-04-22T16:57:42.000Z'
},
{
  ID: 1,
  USER: 'helloworld',
  TITLE: 'Welcome to Forum',
  CREATE_TIME: '2020-03-25T13:32:05.000Z'
},
{
  ID: 4,
  USER: 'helloworld',
  TITLE: 'Intro to C++',
  CREATE_TIME: '2020-04-19T16:35:34.000Z'
},
{
  ID: 20,
  USER: 'helloworld',
  TITLE: 'CSCI3100',
  CREATE_TIME: '2020-04-22T16:57:42.000Z'
},
{
  ID: 1,
  USER: 'newUser',
  TITLE: 'Welcome to Forum',
  CREATE_TIME: '2020-03-25T13:32:05.000Z'
},
{
  ID: 4,
  USER: 'newUser',
  TITLE: 'Intro to C++',
  CREATE_TIME: '2020-04-19T16:35:34.000Z'
},
{
  ID: 20,
  USER: 'newUser',
  TITLE: 'CSCI3100',
  CREATE_TIME: '2020-04-22T16:57:42.000Z'
},
{
  ID: 1,
  USER: 'TEST',
  TITLE: 'Welcome to Forum',
  CREATE_TIME: '2020-03-25T13:32:05.000Z'
},
{
  ID: 4,
  USER: 'TEST',
  TITLE: 'Intro to C++',
  CREATE_TIME: '2020-04-19T16:35:34.000Z'
},
```

```
  {
    ID: 20,
    USER: 'TEST',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  }
]
```

Case 4: (pass)

```
search_post
0
[
  {
    ID: 4,
    USER: 'abc',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 4,
    USER: 'ADMIN',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 4,
    USER: 'helloworld',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 4,
    USER: 'newUser',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 4,
    USER: 'TEST',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  }
]
```

Case 5: (pass)

```
search_post
0
[]
```

Case 6: (pass)

```
search_post
0
[
  {
    ID: 1,
    USER: 'abc',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
```

```
  USER: 'abc',
  TITLE: 'Intro to C++',
  CREATE_TIME: '2020-04-19T16:35:34.000Z'
},
{
  ID: 20,
  USER: 'abc',
  TITLE: 'CSCI3100',
  CREATE_TIME: '2020-04-22T16:57:42.000Z'
},
{
  ID: 1,
  USER: 'ADMIN',
  TITLE: 'Welcome to Forum',
  CREATE_TIME: '2020-03-25T13:32:05.000Z'
},
{
  ID: 4,
  USER: 'ADMIN',
  TITLE: 'Intro to C++',
  CREATE_TIME: '2020-04-19T16:35:34.000Z'
},
{
  ID: 20,
  USER: 'ADMIN',
  TITLE: 'CSCI3100',
  CREATE_TIME: '2020-04-22T16:57:42.000Z'
},
{
  ID: 1,
  USER: 'helloworld',
  TITLE: 'Welcome to Forum',
  CREATE_TIME: '2020-03-25T13:32:05.000Z'
},
{
  ID: 4,
  USER: 'helloworld',
  TITLE: 'Intro to C++',
  CREATE_TIME: '2020-04-19T16:35:34.000Z'
},
{
  ID: 20,
  USER: 'helloworld',
  TITLE: 'CSCI3100',
  CREATE_TIME: '2020-04-22T16:57:42.000Z'
},
{
  ID: 1,
  USER: 'newUser',
  TITLE: 'Welcome to Forum',
  CREATE_TIME: '2020-03-25T13:32:05.000Z'
},
{
  ID: 4,
  USER: 'newUser',
  TITLE: 'Intro to C++',
  CREATE_TIME: '2020-04-19T16:35:34.000Z'
},
{
  ID: 20,
  USER: 'newUser',
```

```
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  },
  {
    ID: 1,
    USER: 'TEST',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'TEST',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 20,
    USER: 'TEST',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  }
]
```

Title Retrieval

The method should return title, post id, author and created time of all posts, not replies upon success.

Case 1: (pass)

```
search_post_head
[
  {
    ID: 1,
    USER: 'ADMIN',
    TITLE: 'Welcome to Forum',
    CREATE_TIME: '2020-03-25T13:32:05.000Z'
  },
  {
    ID: 4,
    USER: 'ADMIN',
    TITLE: 'Intro to C++',
    CREATE_TIME: '2020-04-19T16:35:34.000Z'
  },
  {
    ID: 13,
    USER: 'ADMIN',
    TITLE: 'Intro to C',
    CREATE_TIME: '2020-04-22T07:06:22.000Z'
  },
  {
    ID: 20,
    USER: 'ADMIN',
    TITLE: 'CSCI3100',
    CREATE_TIME: '2020-04-22T16:57:42.000Z'
  },
  {
    ID: 25,
    USER: 'ADMIN',
    TITLE: 'gcd in C',
    CREATE_TIME: '2020-04-23T14:37:50.000Z'
  },
```

```
  {
    ID: 38,
    USER: 'helloworld',
    TITLE: 'hello world',
    CREATE_TIME: '2020-05-10T15:48:01.000Z'
  }
]
```

5.6 User Data Handling Module

Because the module acts as an interface which spawns process of connect_sql.js MySQL database connection interface module. Therefore the testing of this intermediate module is merged into tests of root server module as decisions are done in root server module and webpage javascripts.

5.7 User Account Feature

5.7.1 Purpose

The system provides user account registration, authentication and personal data retrieval functions to users. Account registration will be tested separately to check whether there is any unwanted behaviour such as allowing blank username or password or repetition of username or user email. User authentication will be checked with user data retrieval as the two functions are closly related.

5.7.2 Inputs

User Account Registration

Used testcases (username, email, password, password):

1. Joker, joker@dc.com, heath-ledger, heath-ledger (valid)
2. TEST, test@notest.com, 12345678, 12345678 (used username)
3. ITEM, test@test.com, 12345678, 12345678 (used email)
4. hi, hi, 12345678, 12345678 (invalid email address)
5. hihi, hihi@hihi.net, qwer, qwer (password length < 8)
6. (empty), incognito@test.com, 12345678, 12345678 (empty username)
7. hihihi, (empty), 12345678, 12345678 (empty email)
8. @#$xyz, linus@god.com, 12345678, 12345678 (invalid username)
9. hi, hi@hi.net, 12345678, 456789123 (incorrect re-type of password)

User Account Authentication and User Data Retrieval

Used testcase:

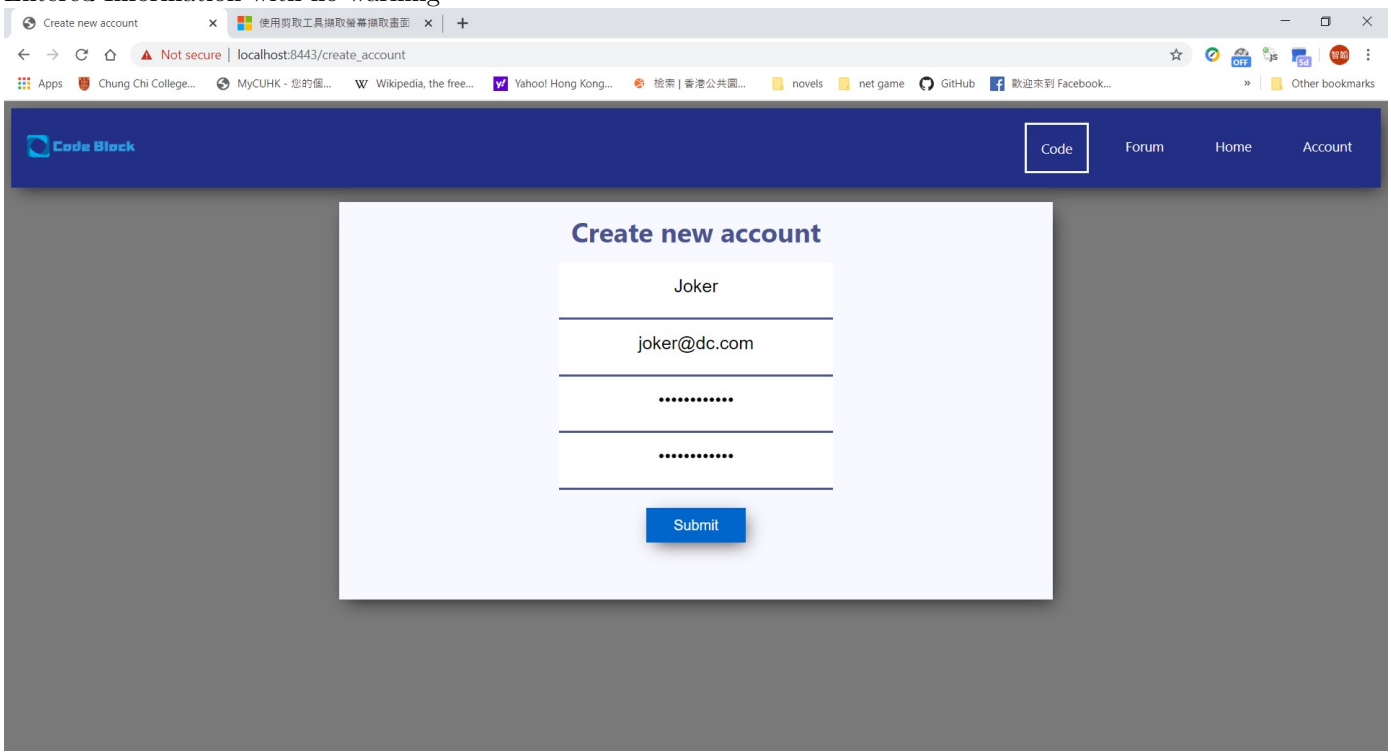
1. ADMIN, CSCI3100GRP18 (valid normal account)
2. Joker, 12345678 (wrong password)
3. AD, CSCI3100GRP18 (invalid username/email)
4. test@test.com, ABC (valid teacher account using email login)
5. abc, 11111111 (valid student account)
6. Joker, Heath-Ledger (different case)
7. “ or ”“=”, a OR 1=1 (SQL injection)
8. *c*, * (SQL injection)
9. idk, “;SELECT * FROM USER WHERE USERNAME=”ADMIN (SQL injection)

5.7.3 Expected Outputs & Pass/Fail Criteria

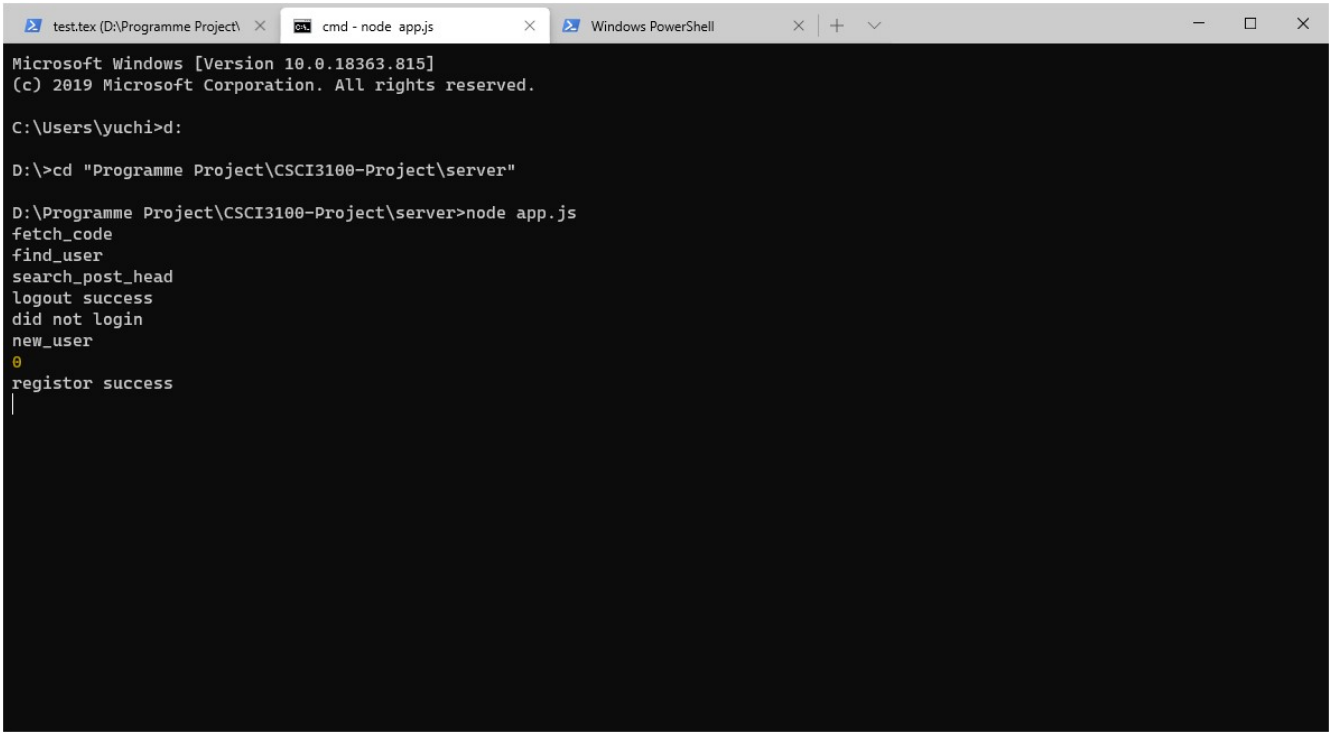
User Account Registration

The system should allow only valid users to register with valid user data. And the system should give warning when invalid data is submitted.

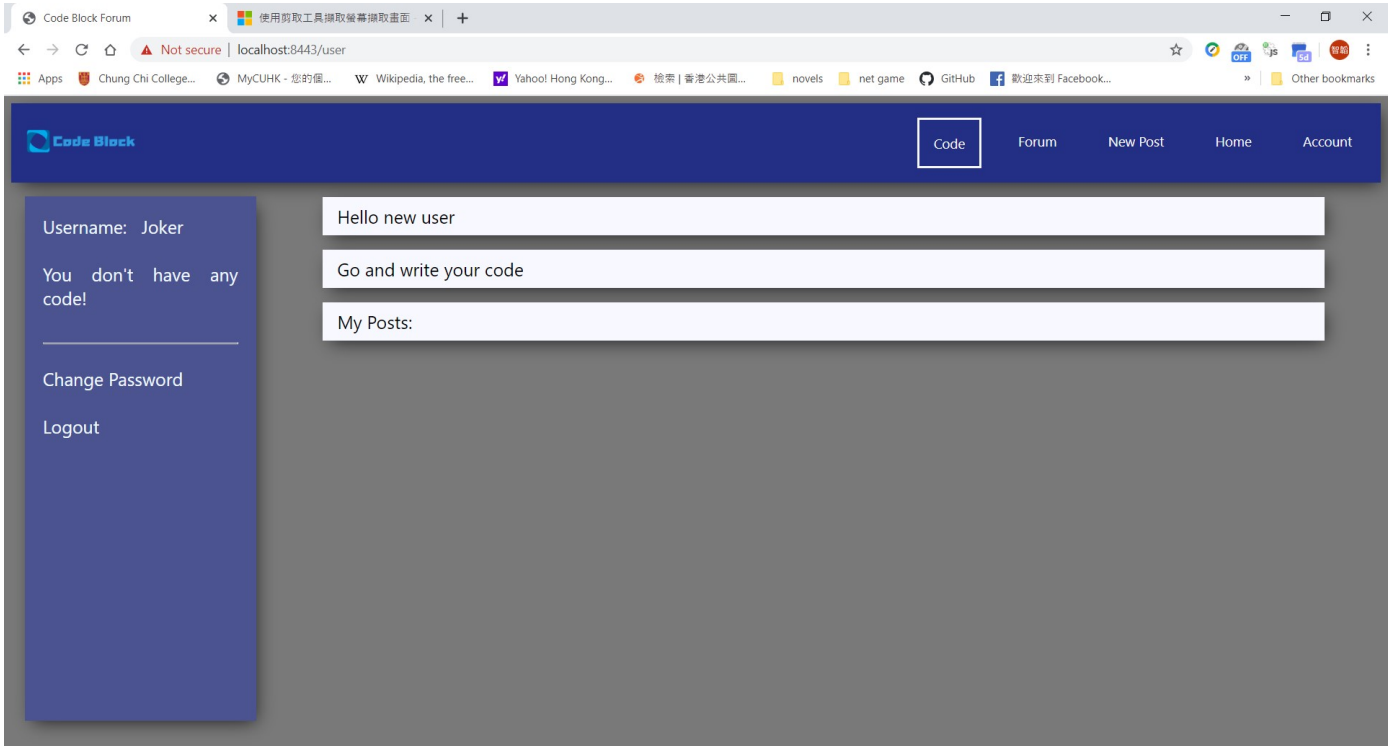
Case 1 (pass):
Entered Information with no warning



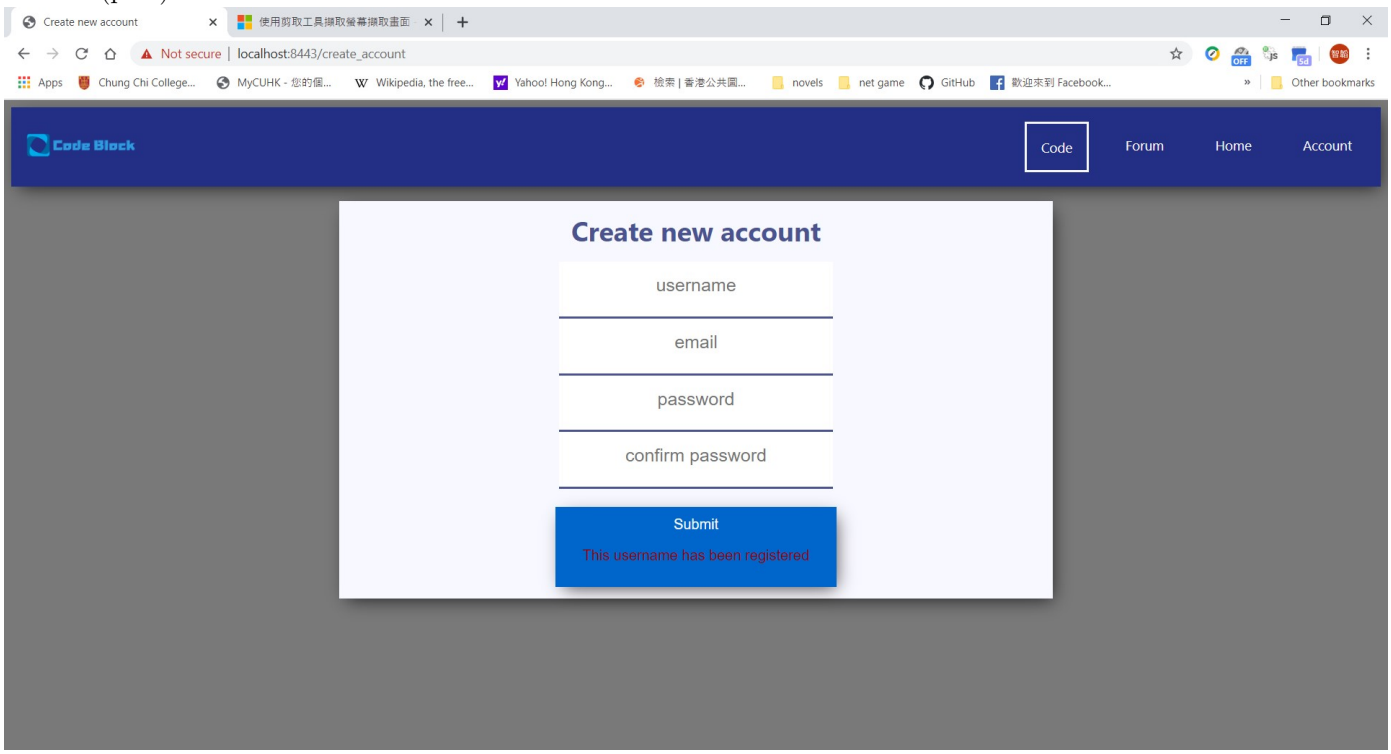
Result in shown in terminal



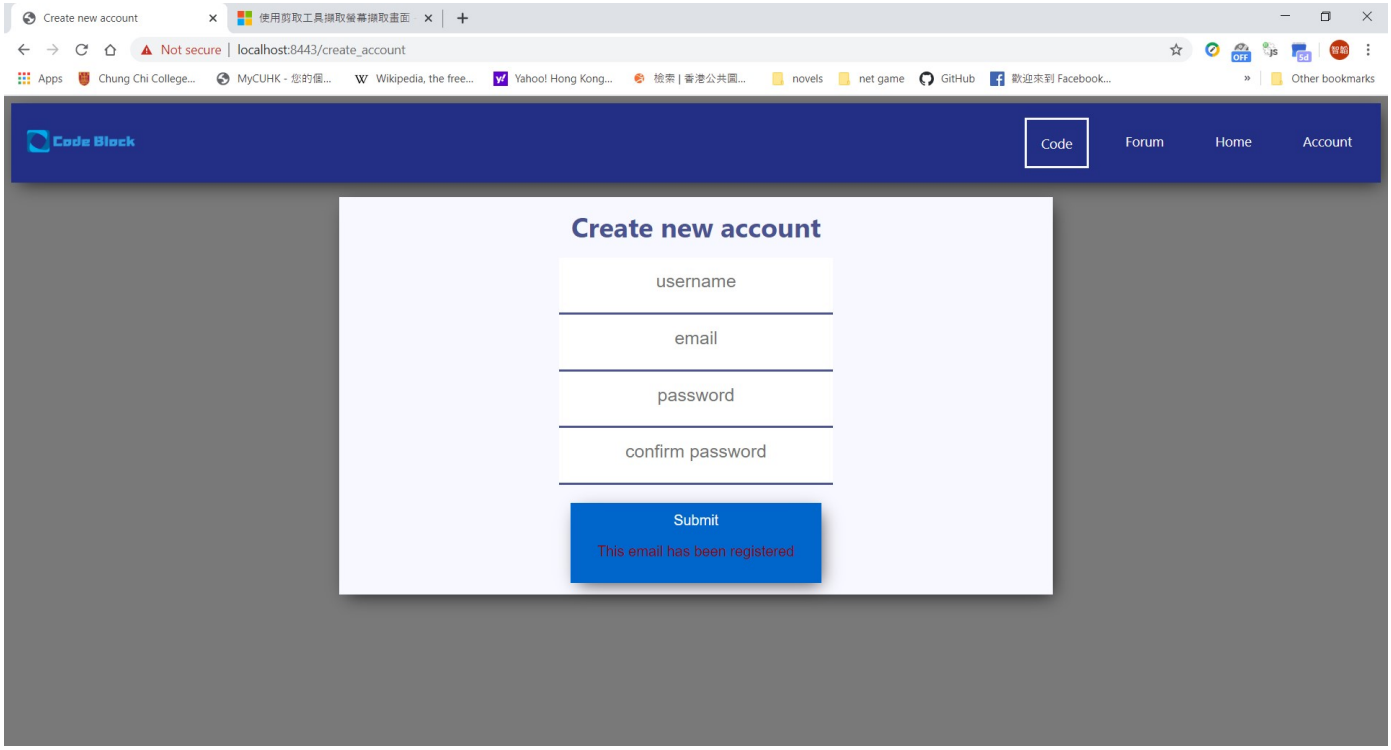
User Page



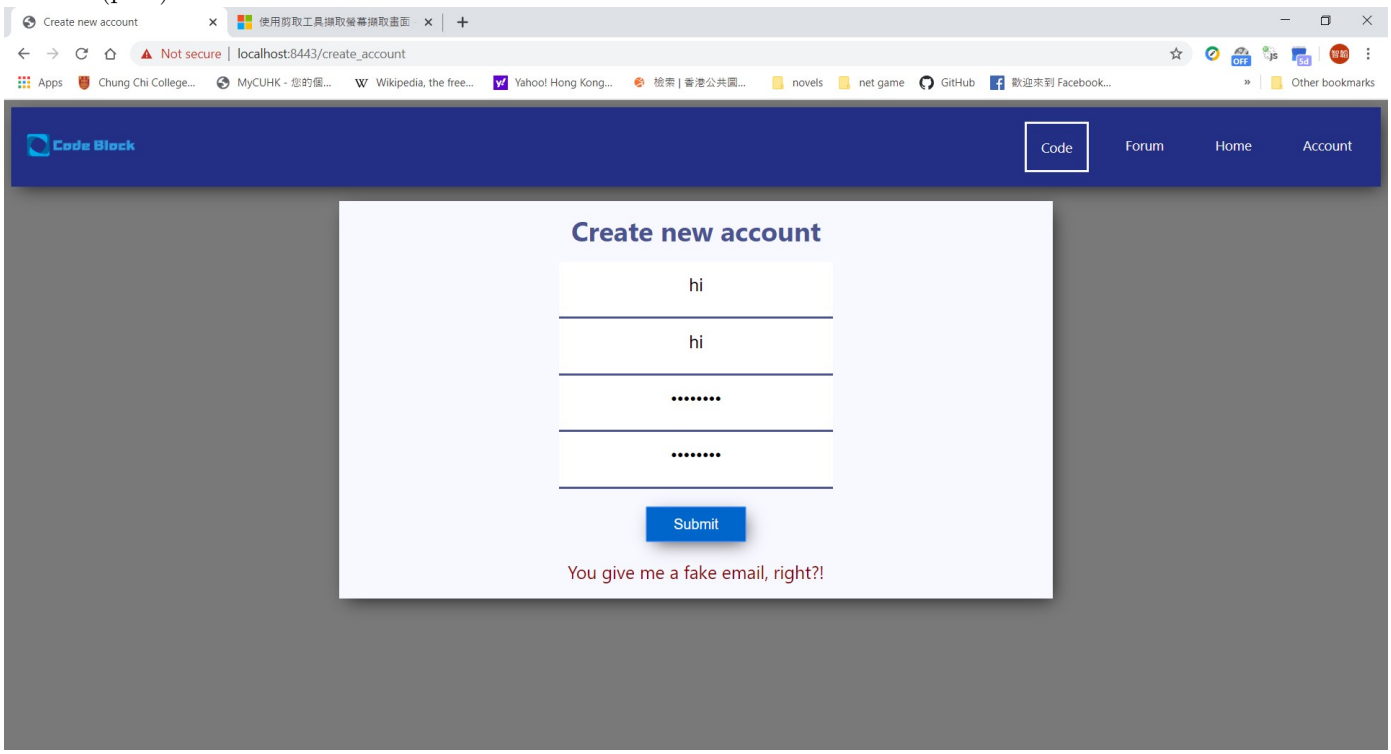
Case 2: (pass)



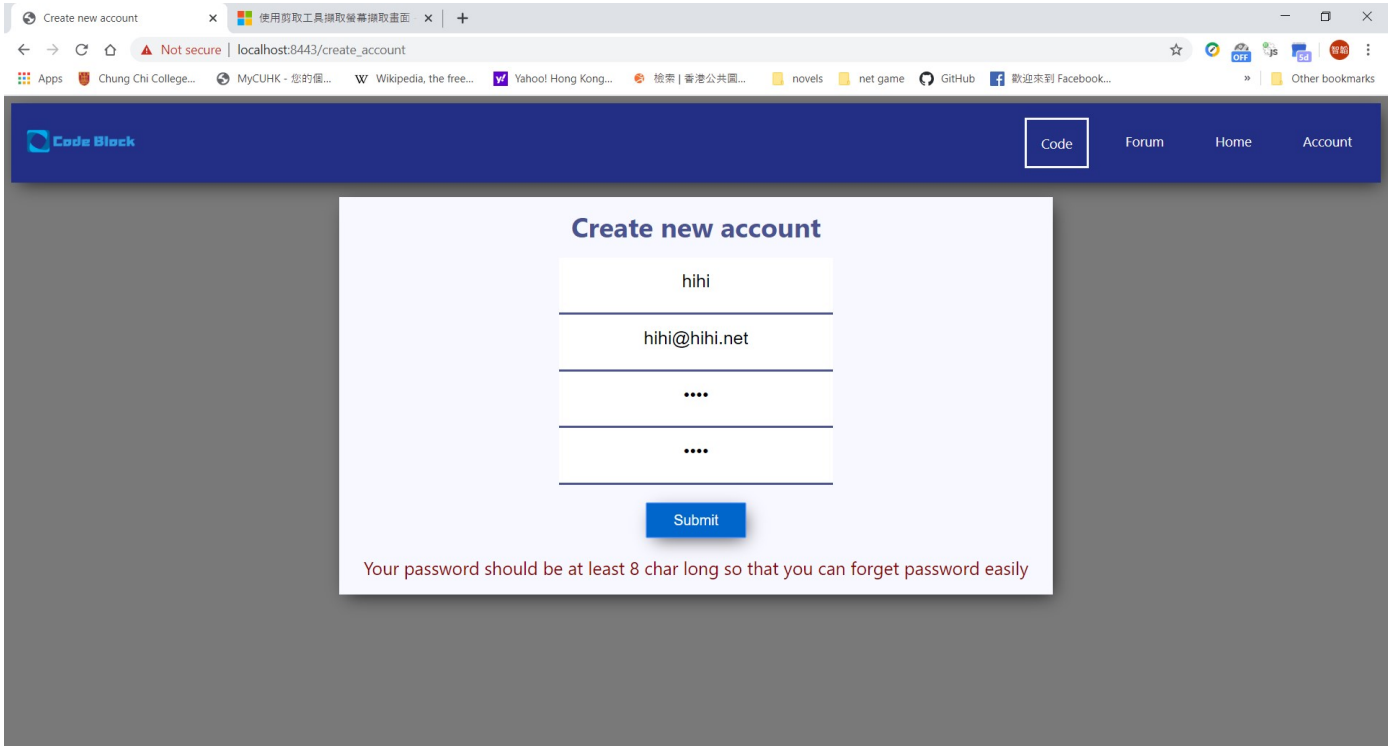
Case 3: (pass)



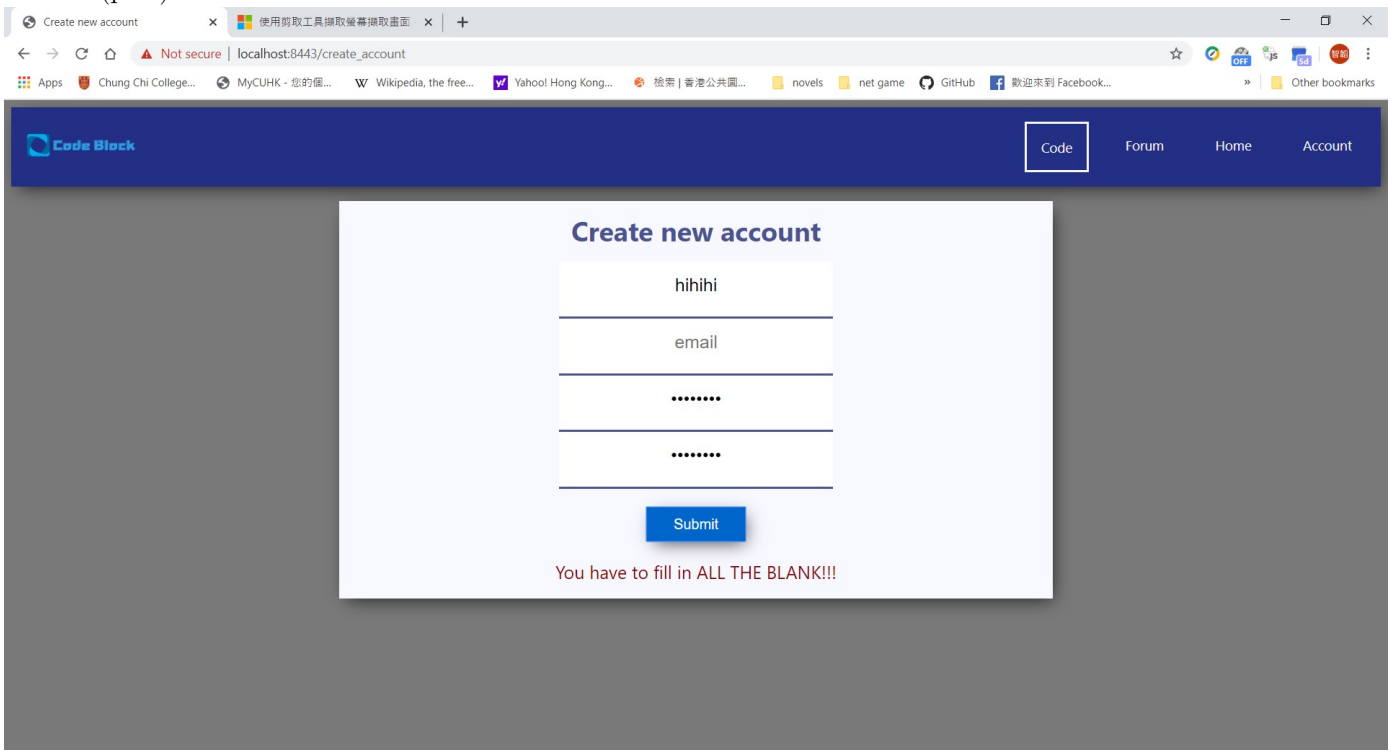
Case 4: (pass)



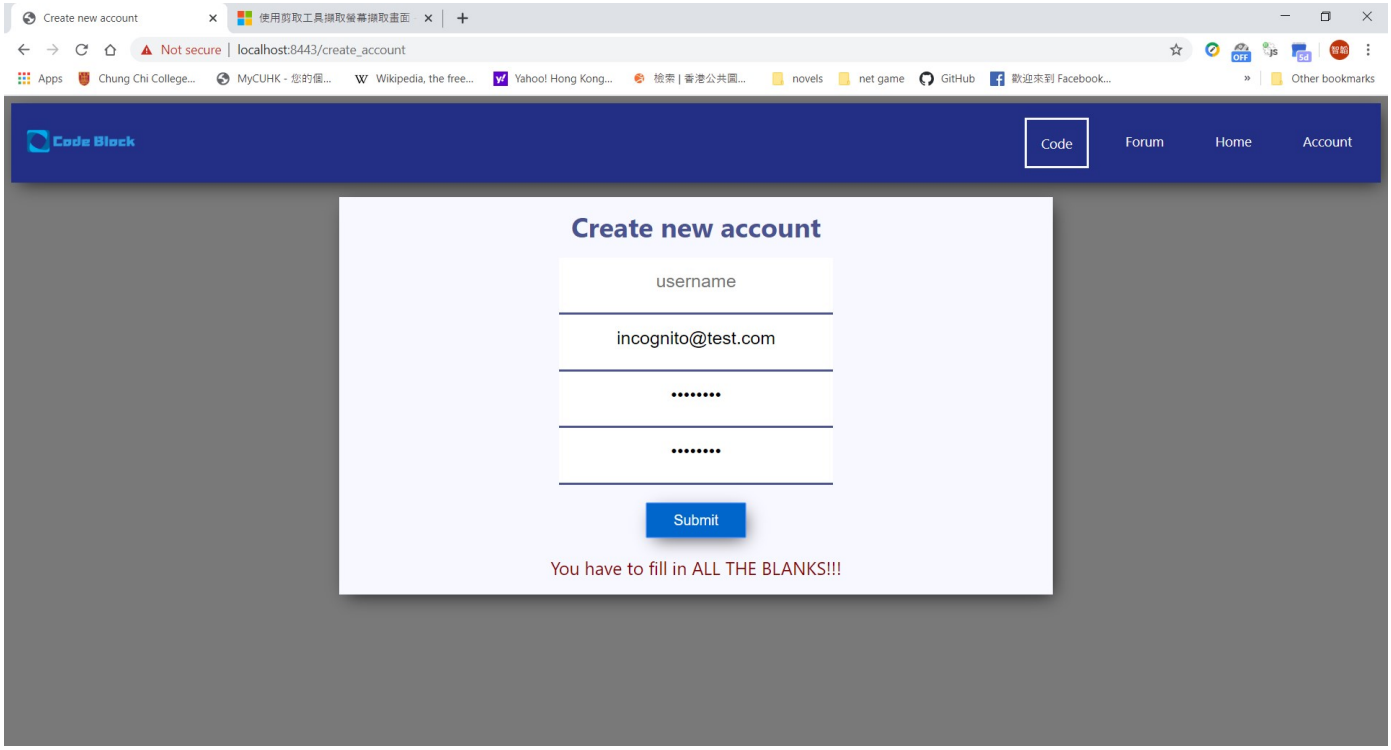
Case 5: (pass)



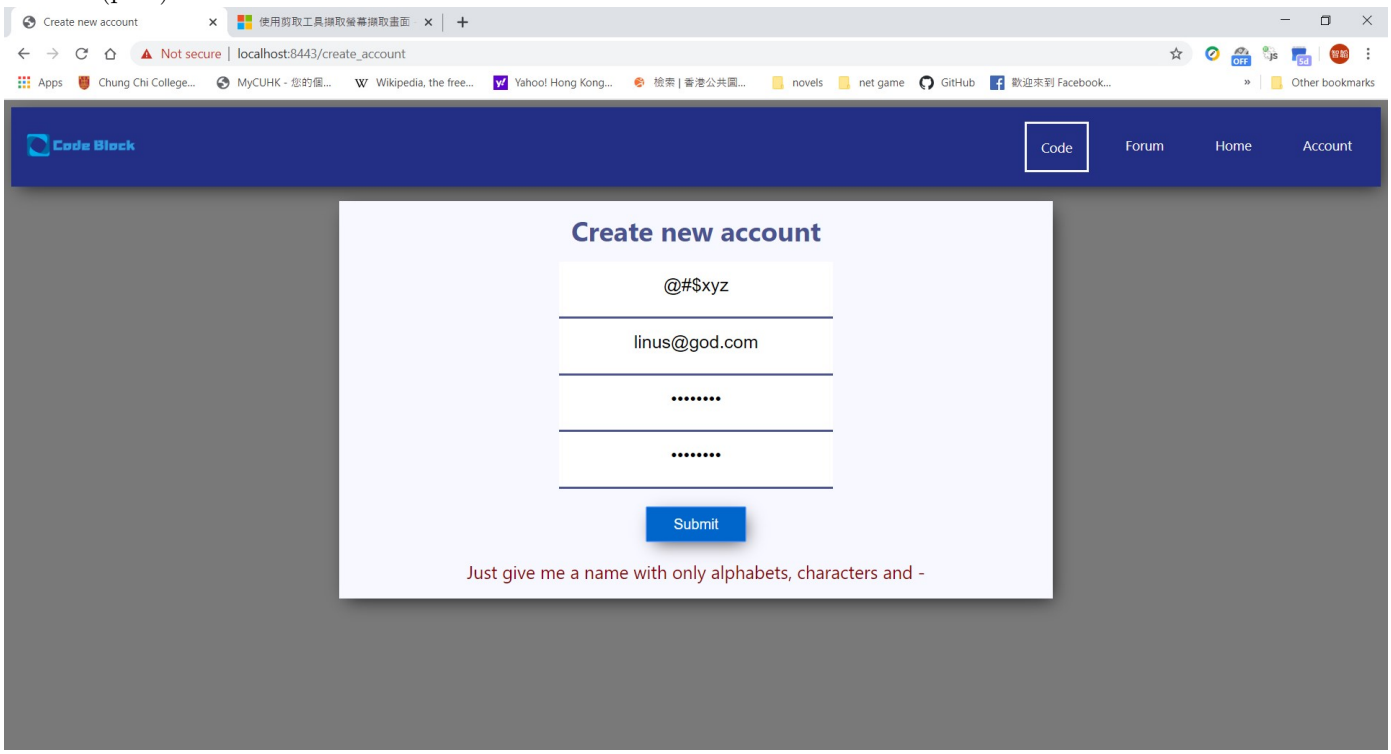
Case 6: (pass)



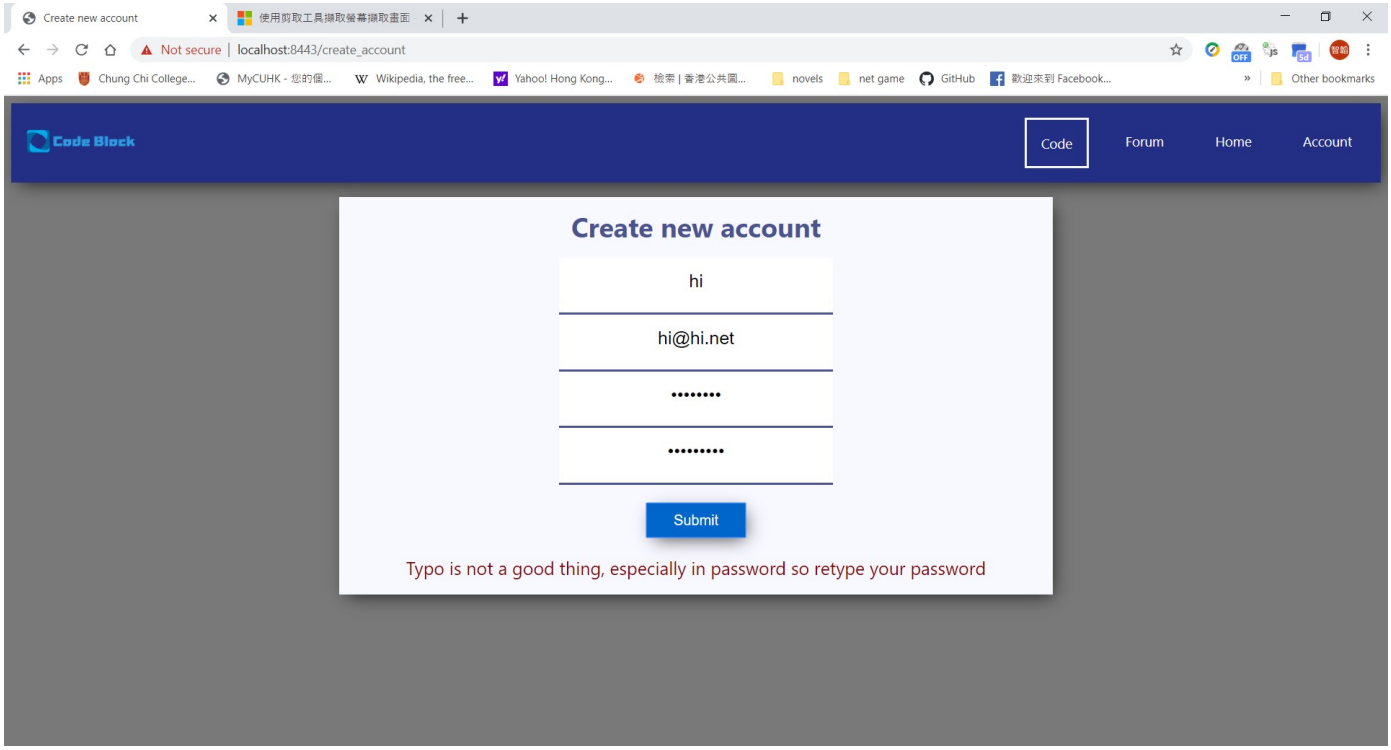
Case 7: (pass)



Case 8: (pass)



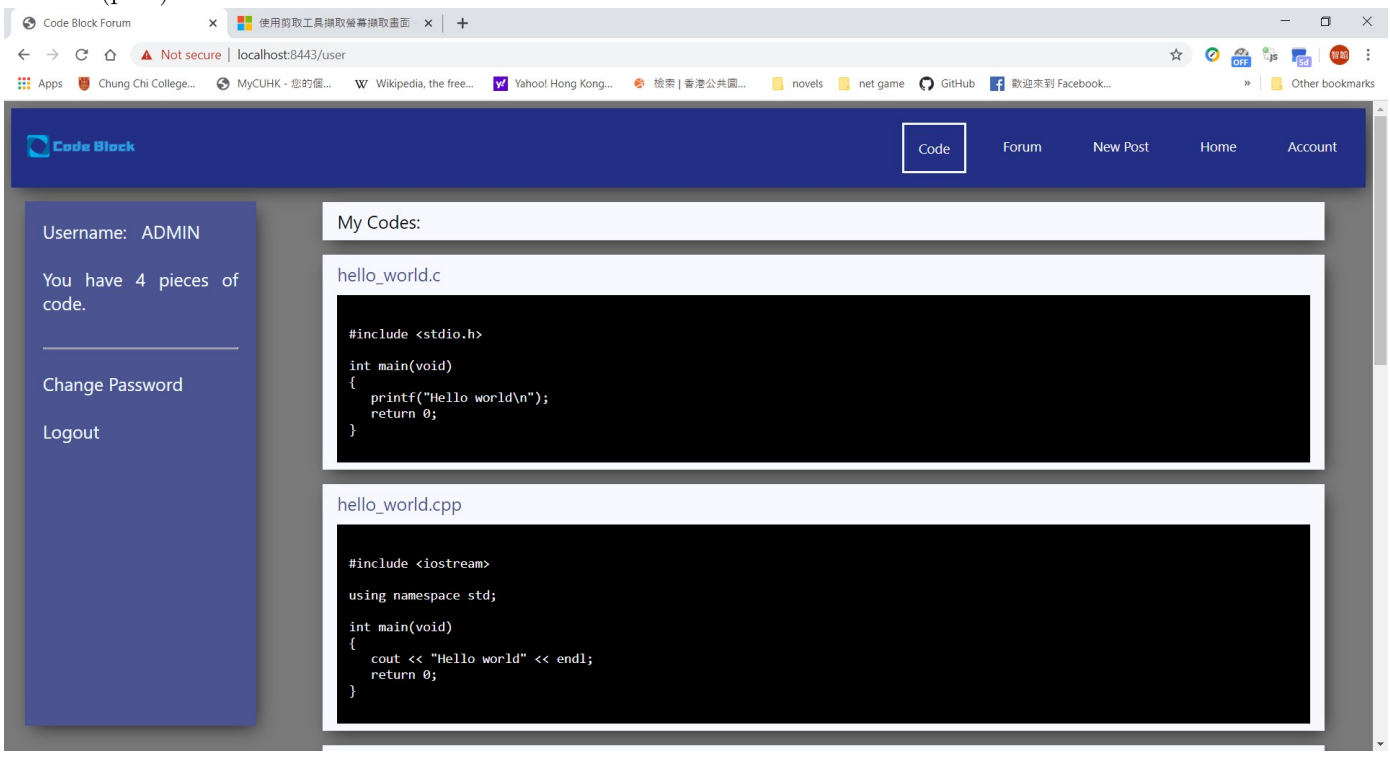
Case 9: (pass)



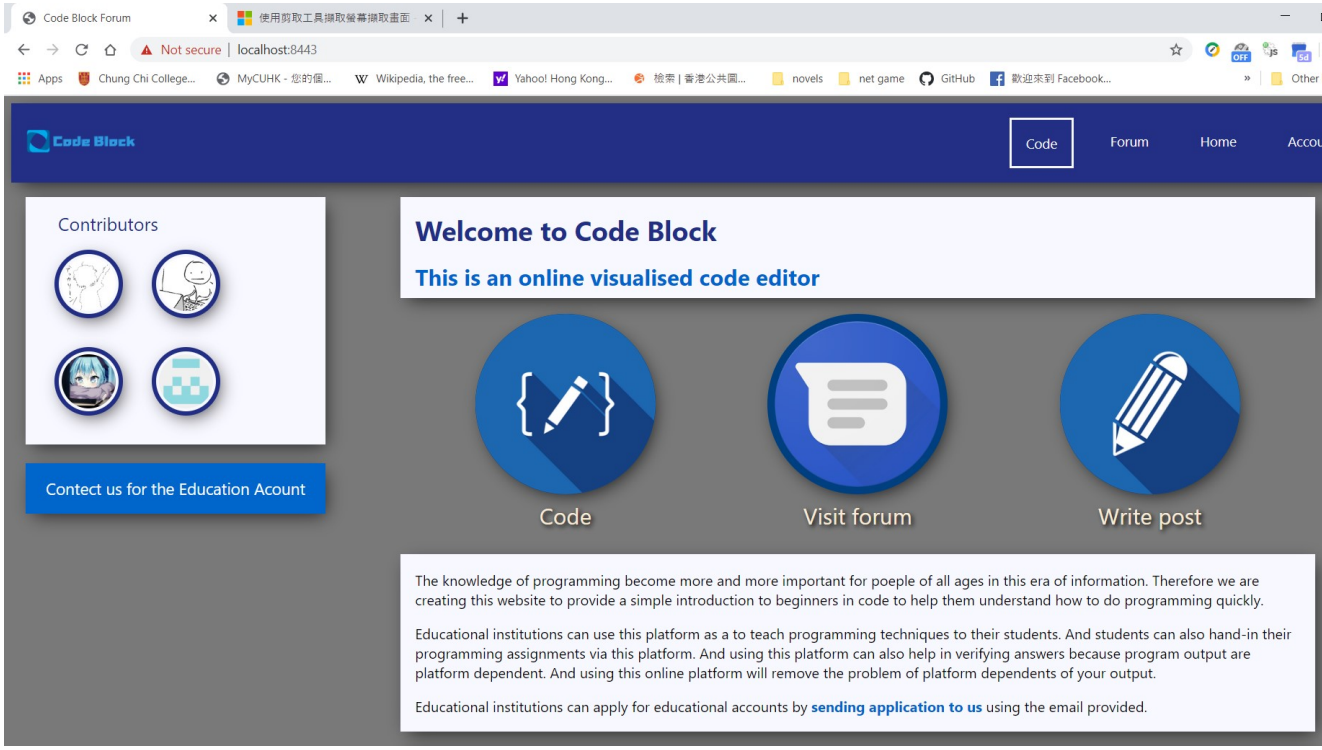
User Account Authentication and User Data Retrieval

The system should allow users to authenticate when they have submitted valid account information. And for successful authentications, the system should redirect the user to user's mainpage which lists out informations of the user in the system, including all codes and posts written by the user.
Upon unsuccessful authentications, the system should notify the user that the username or password is incorrect. And the system should be SQL injection resistant, such that when special symbols are used in username or password, the system will still perform comparison correctly for authentication.

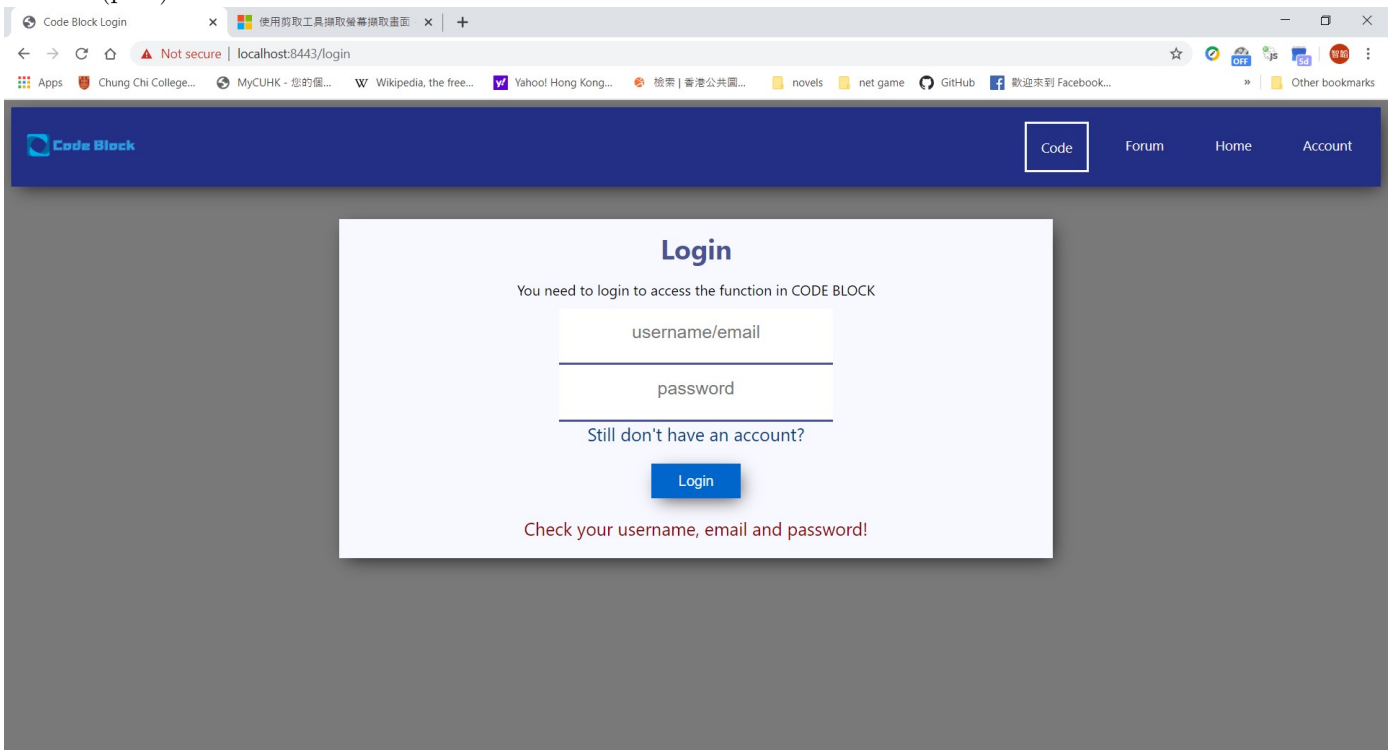
Case 1: (pass)



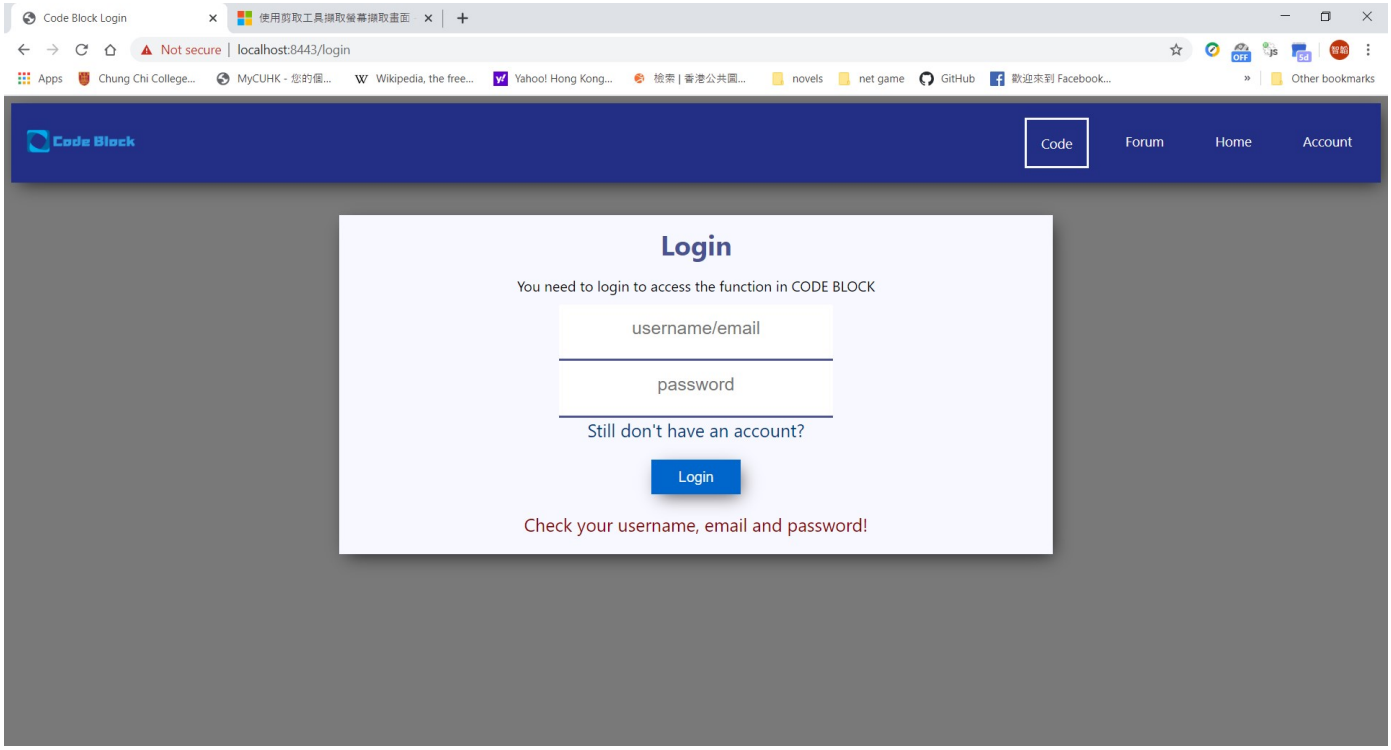
Logout



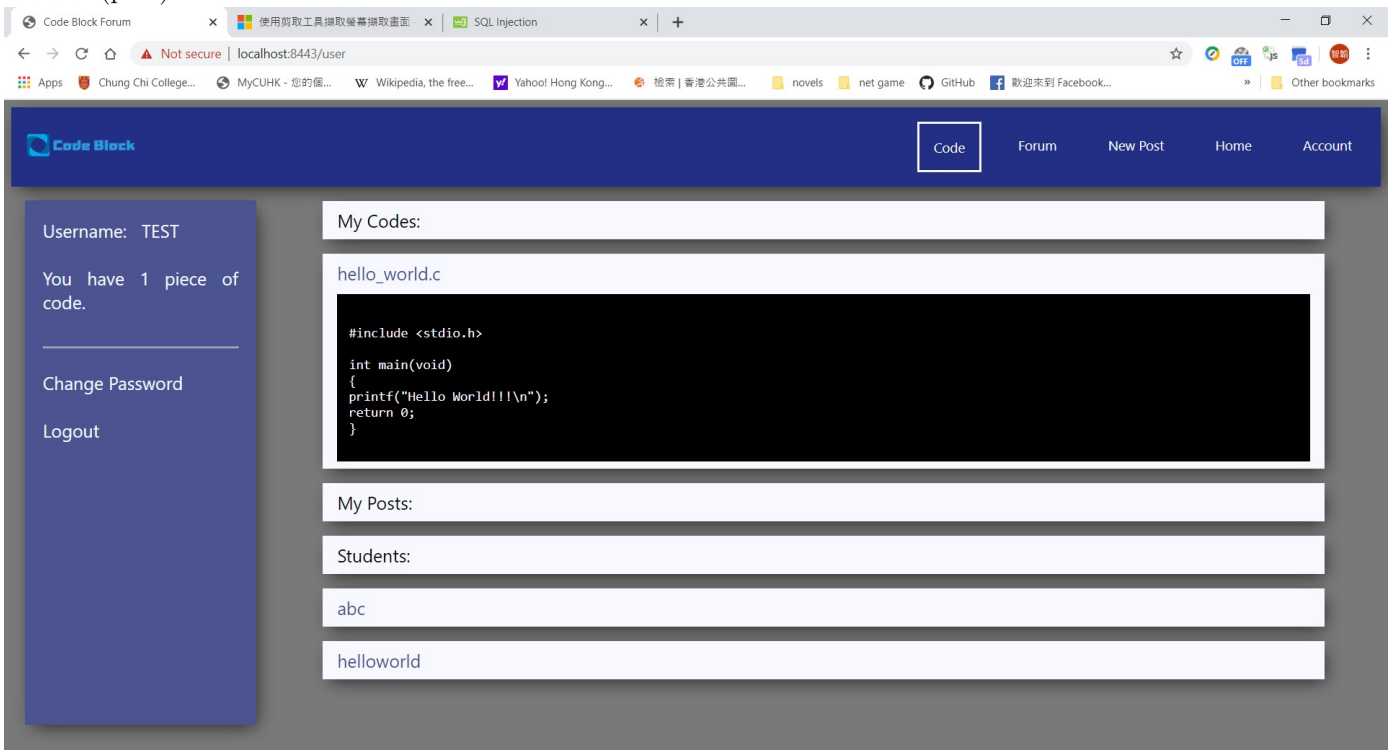
Case 2: (pass)

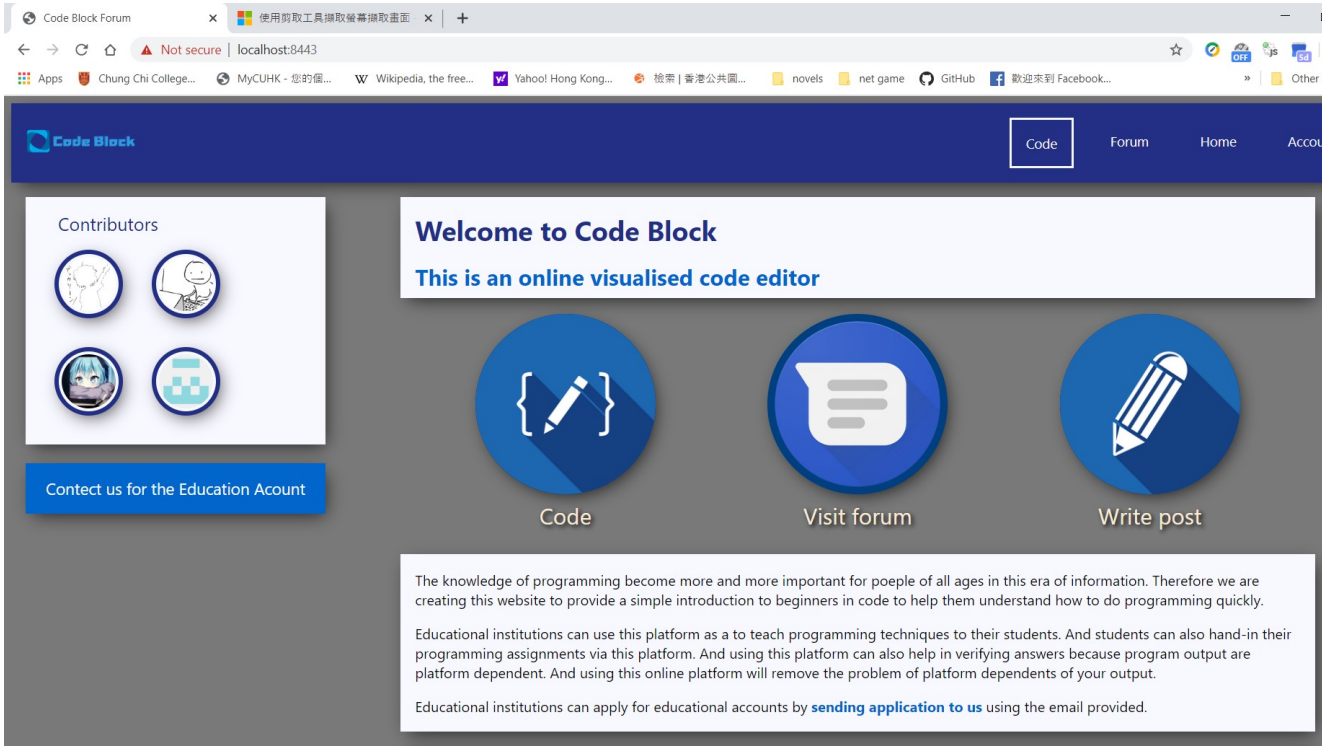


Case 3: (pass)


Code Block Login
Not secure | localhost:8443/login
Code Block
Code
Forum
Home
Account
Login
You need to login to access the function in CODE BLOCK
username/email
password
Still don't have an account?
Login
Check your username, email and password!

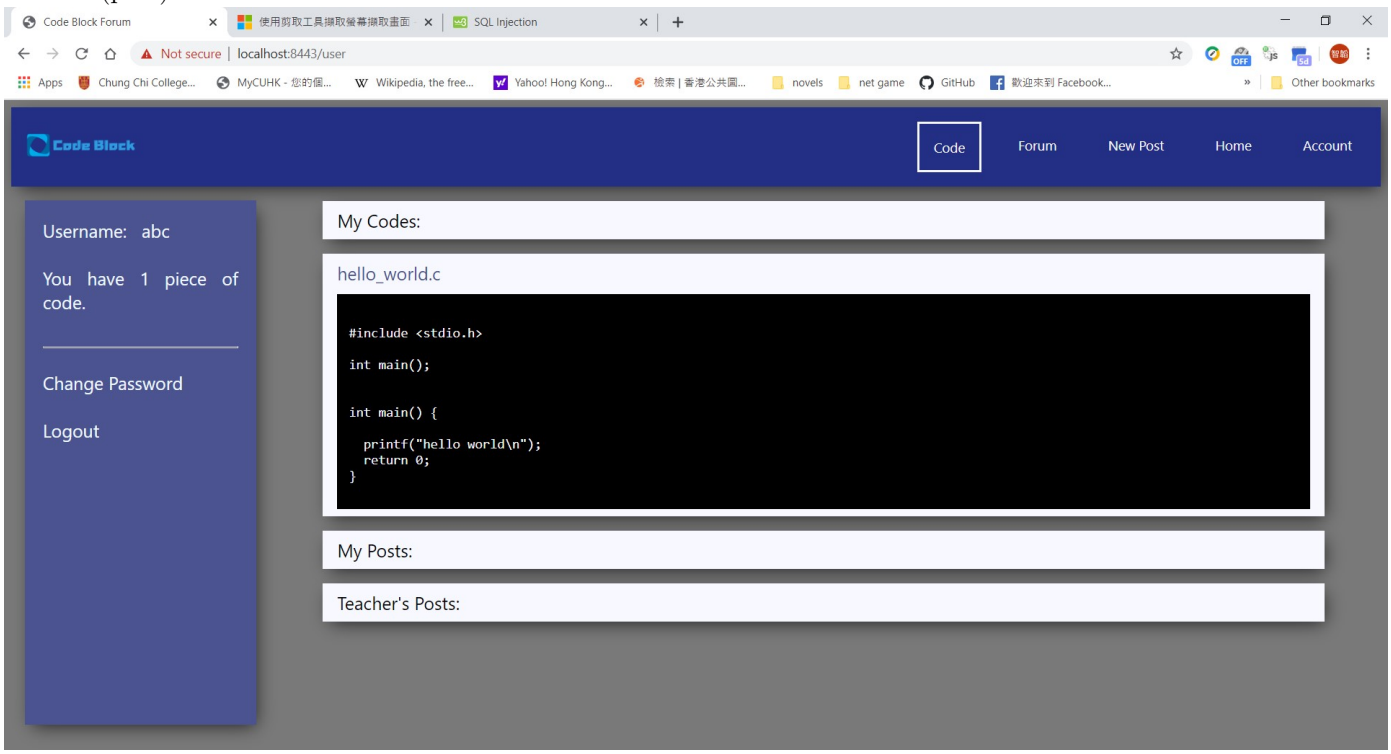
Case 4: (pass)


Code Block Forum
Not secure | localhost:8443/user
Code Block
Code
Forum
New Post
Home
Account
Username: TEST
You have 1 piece of code.
Change Password
Logout
My Codes:
hello_world.c
#include <stdio.h>
int main(void)
{
printf("Hello World!!!\n");
return 0;
}
My Posts:
Students:
abc
helloworld

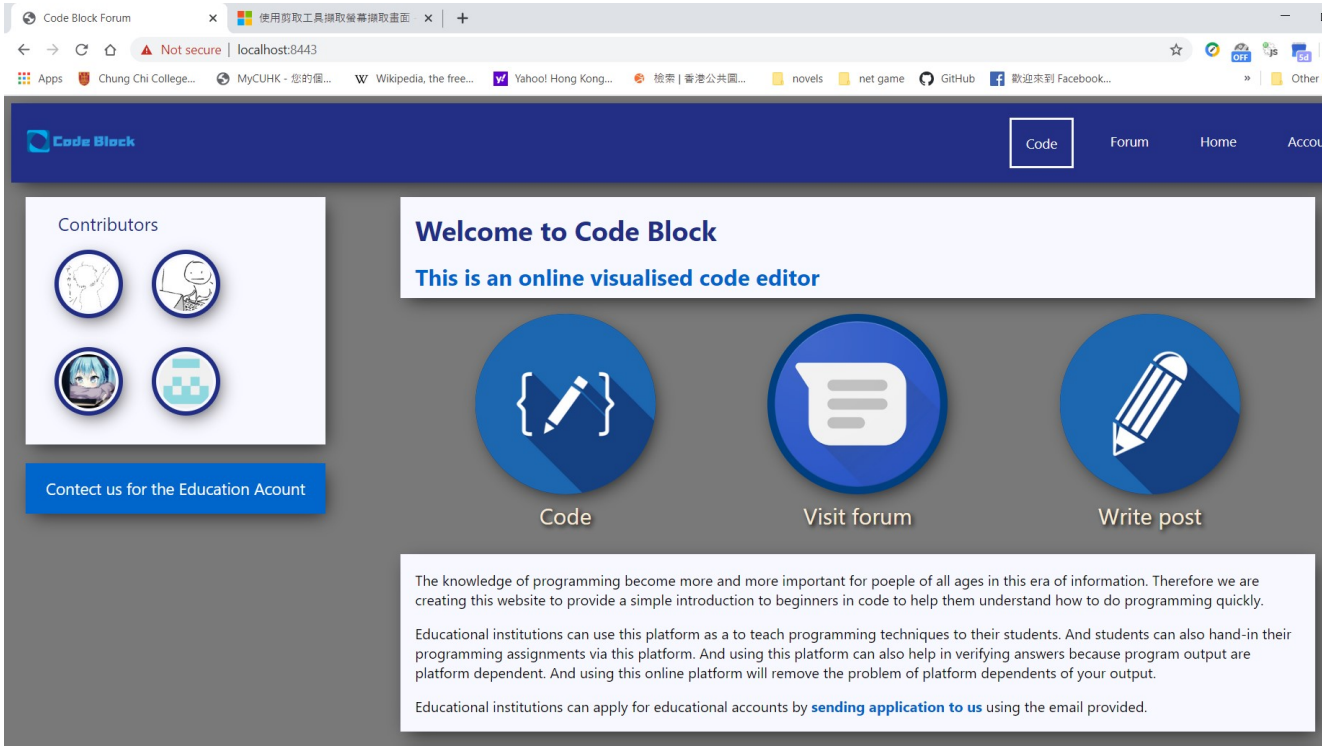


Logout

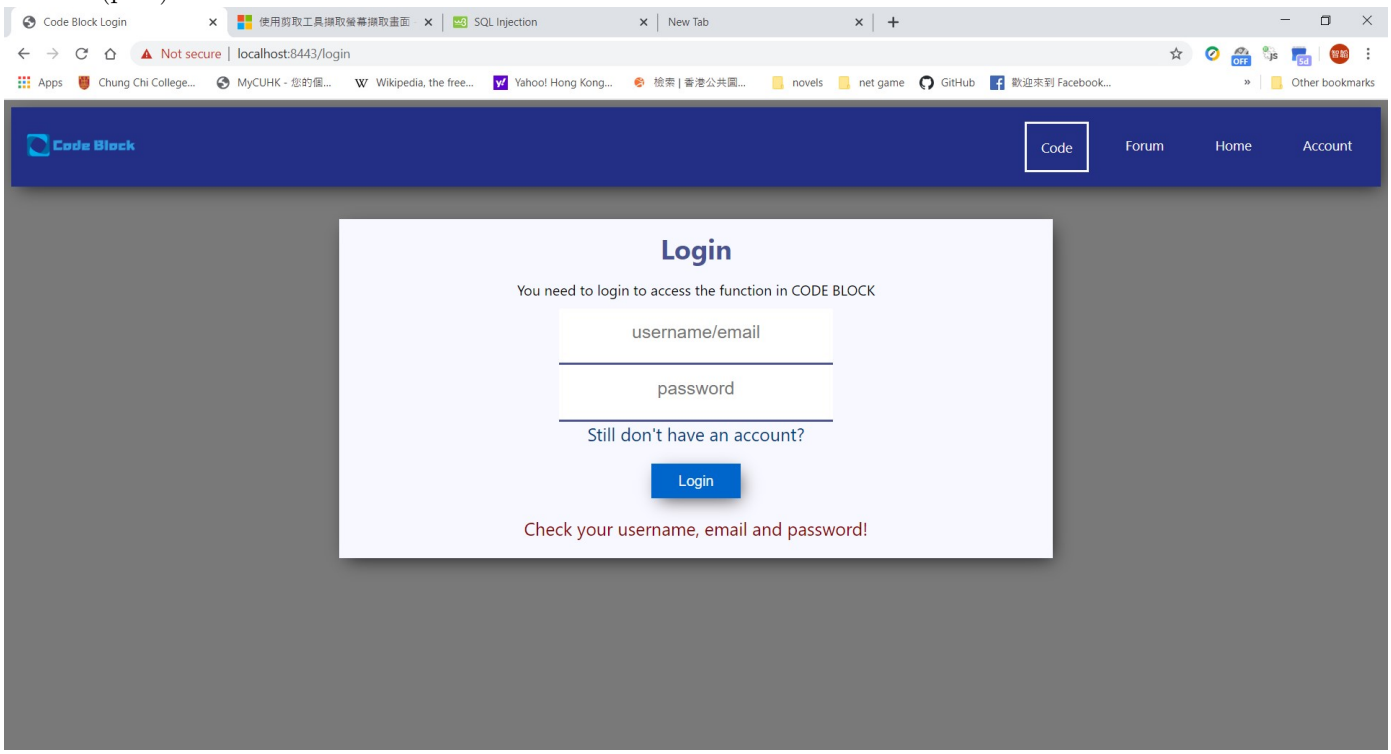
Case 5: (pass)



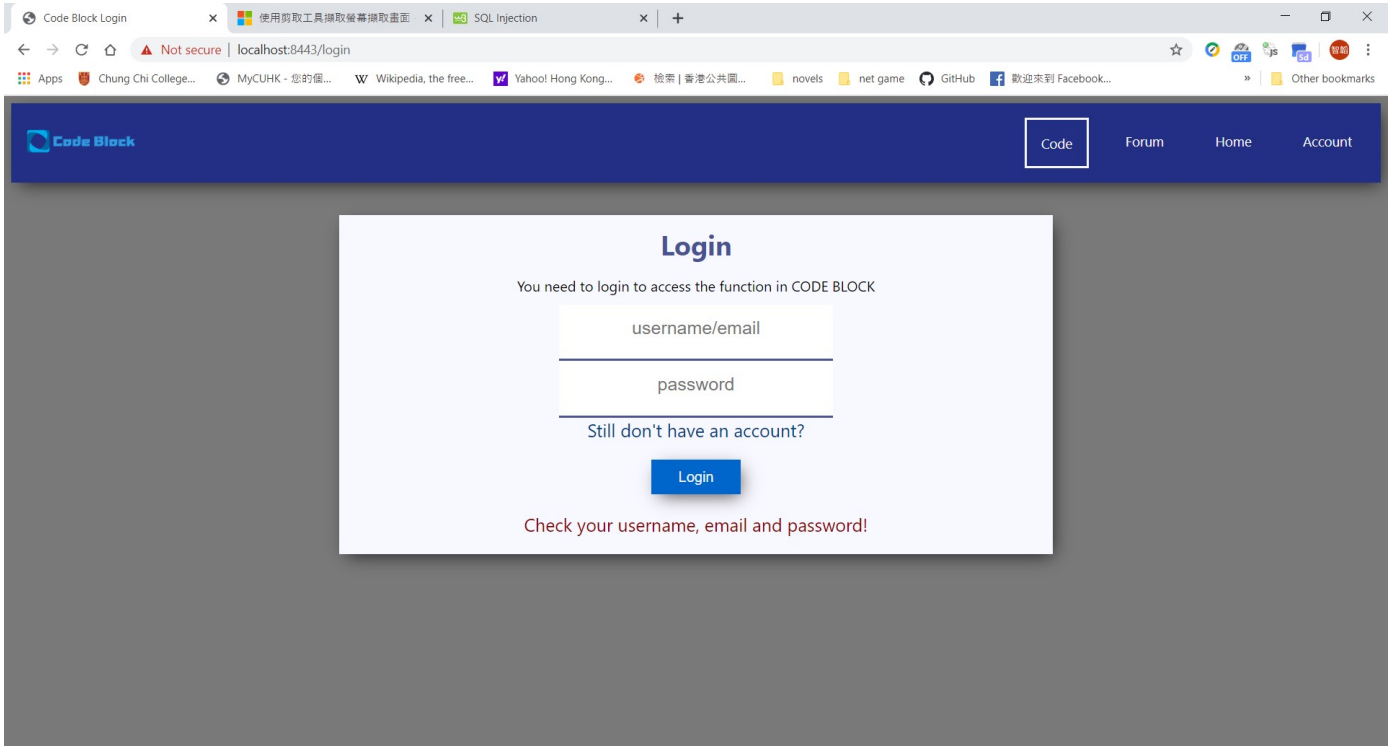
Logout



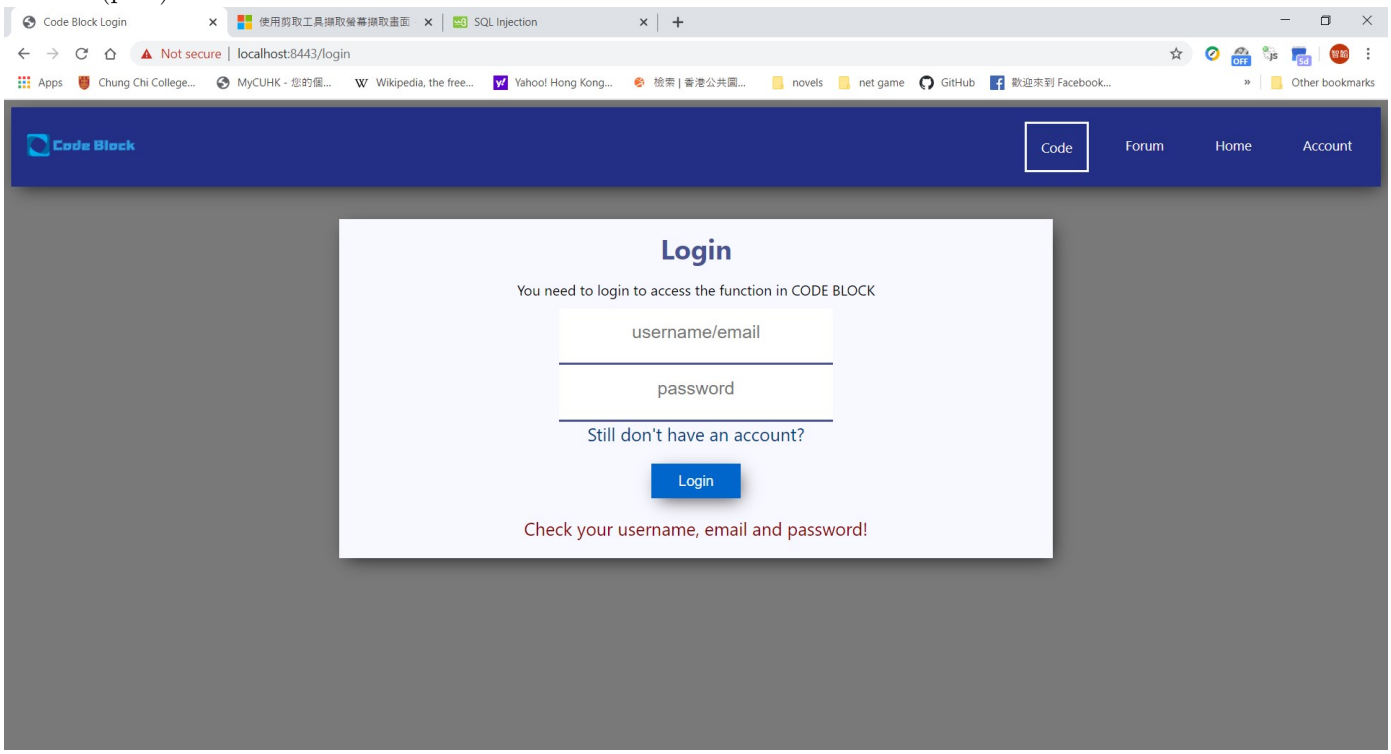
Case 6: (pass)



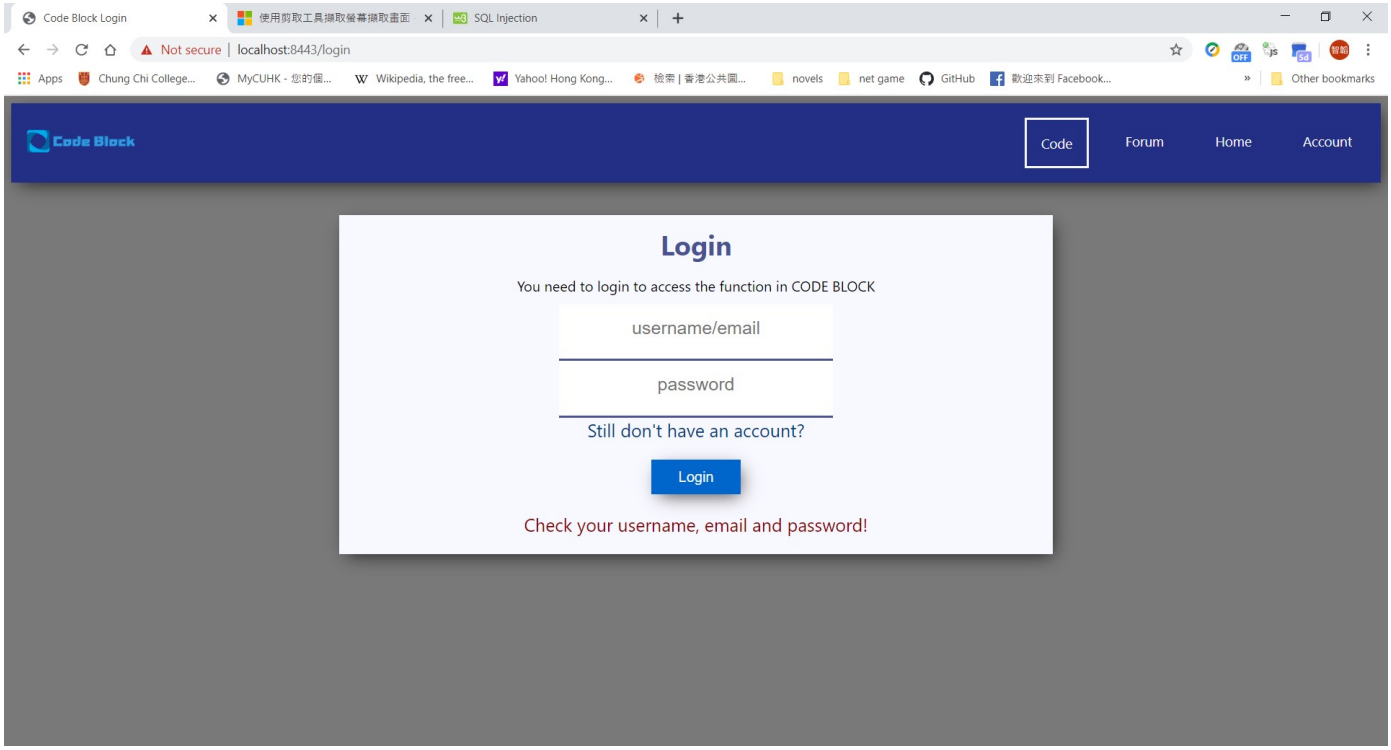
Case 7: (pass)



Case 8: (pass)



Case 9: (pass)



5.8 Code Editing and Saving

5.8.1 Purpose

The system should allow authenticated users to do code editing, using text-based or visualised method. And it should allow authenticated users to store their code written in the database for later retrieval.
The testcases will observe upon three functions of the feature, writing new code, updating code and saving code. Upon success, the system should redirect the user from code editing page to a page showing the code saved, the code saved should be shown in user page also. The test of code saving will be combined in tests of writing new code and code update.
The text-based code editing will be tested with visualised code editing since the current version does not support saving the source code generated using the visualised code editor directly.

5.8.2 Inputs

New Code Writing

1. visualised code editor with no input from stdin and text-based code submit
2. visualised code editor with stdin and text-based code submit
3. visulised code editor invalid block placement
4. text-based wrong code submission

Code Update

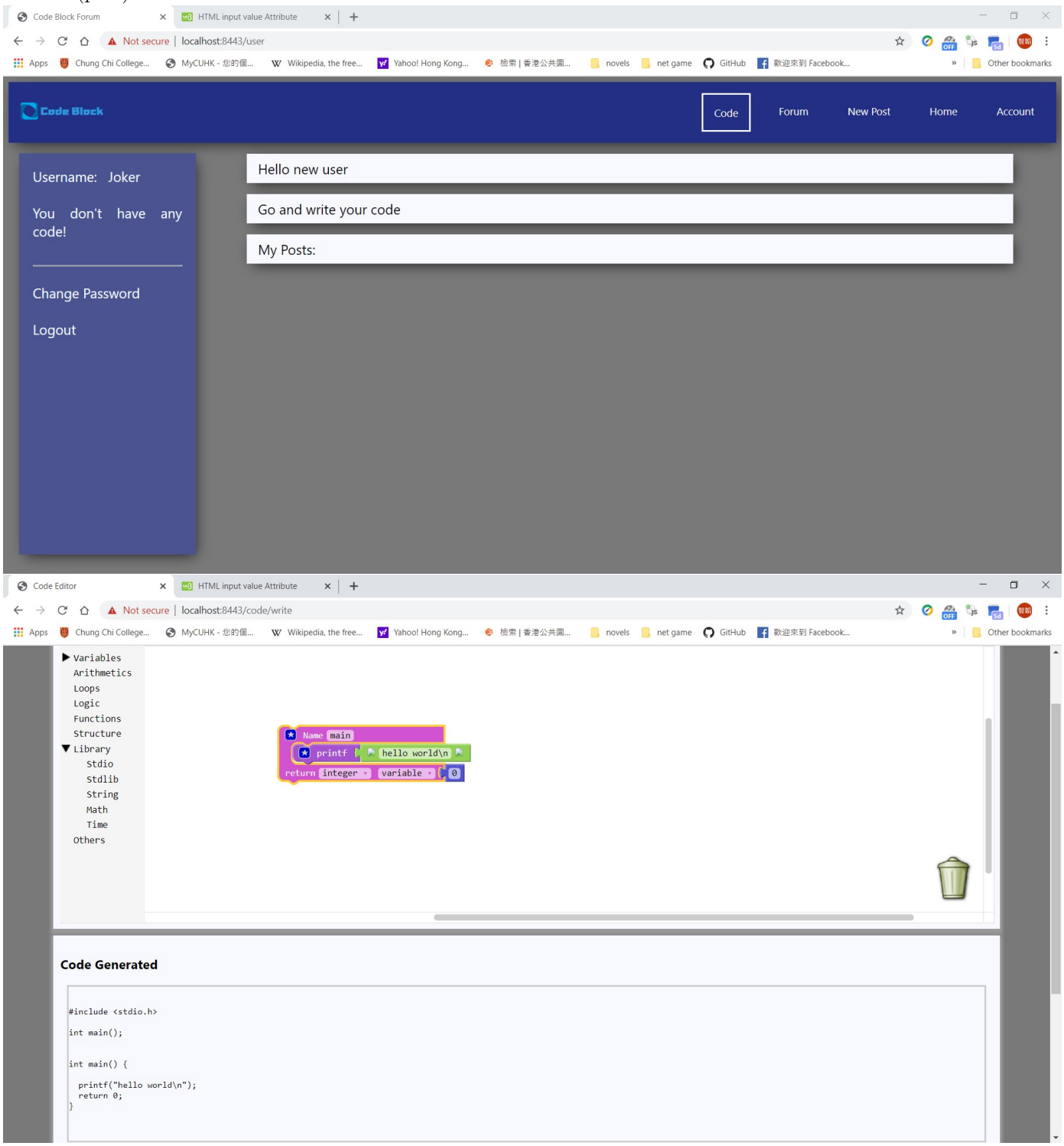
1. text-based code update upon Case 2 of New Code Writing

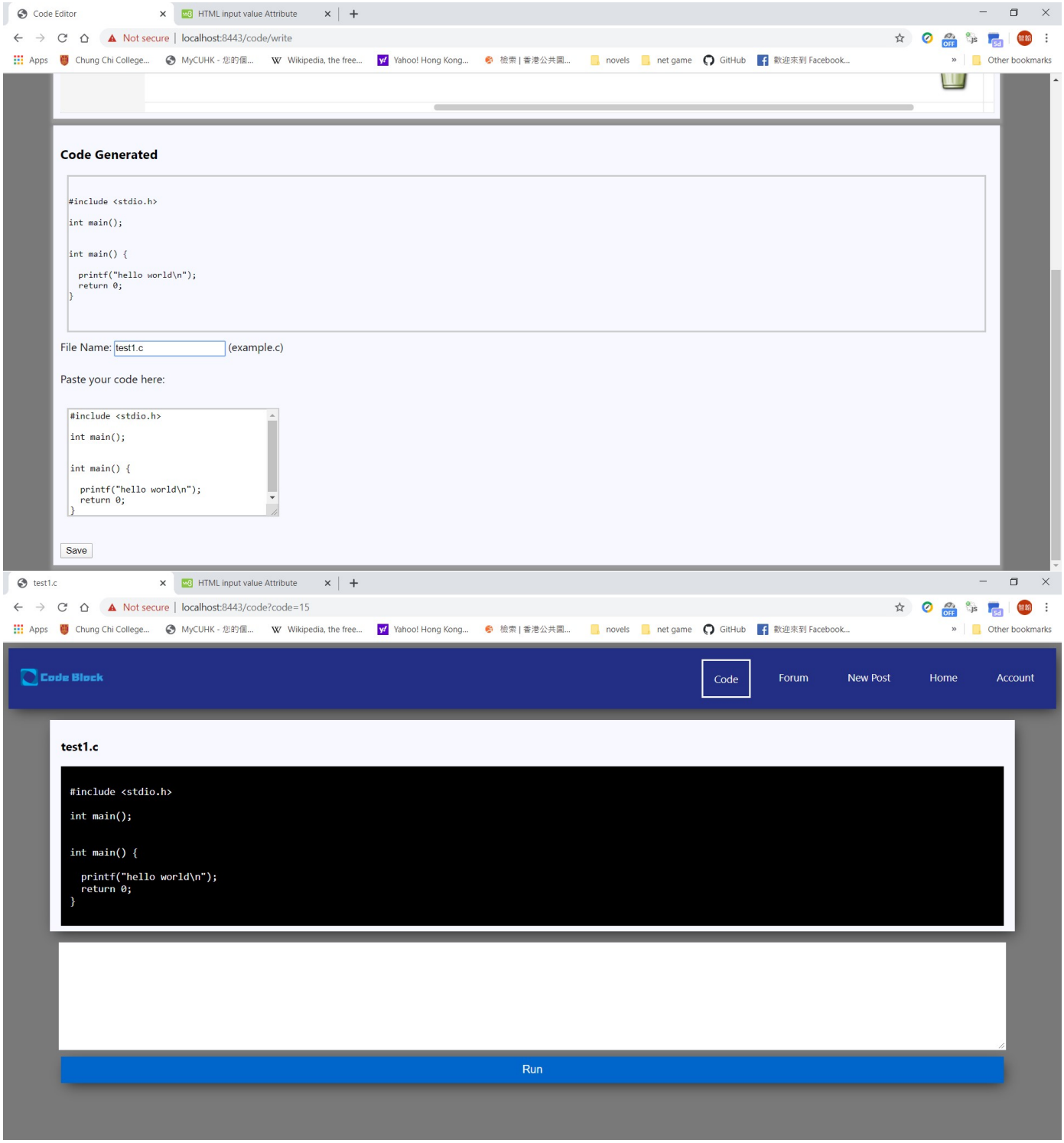
5.8.3 Expected Outputs & Pass/Fail Criteria

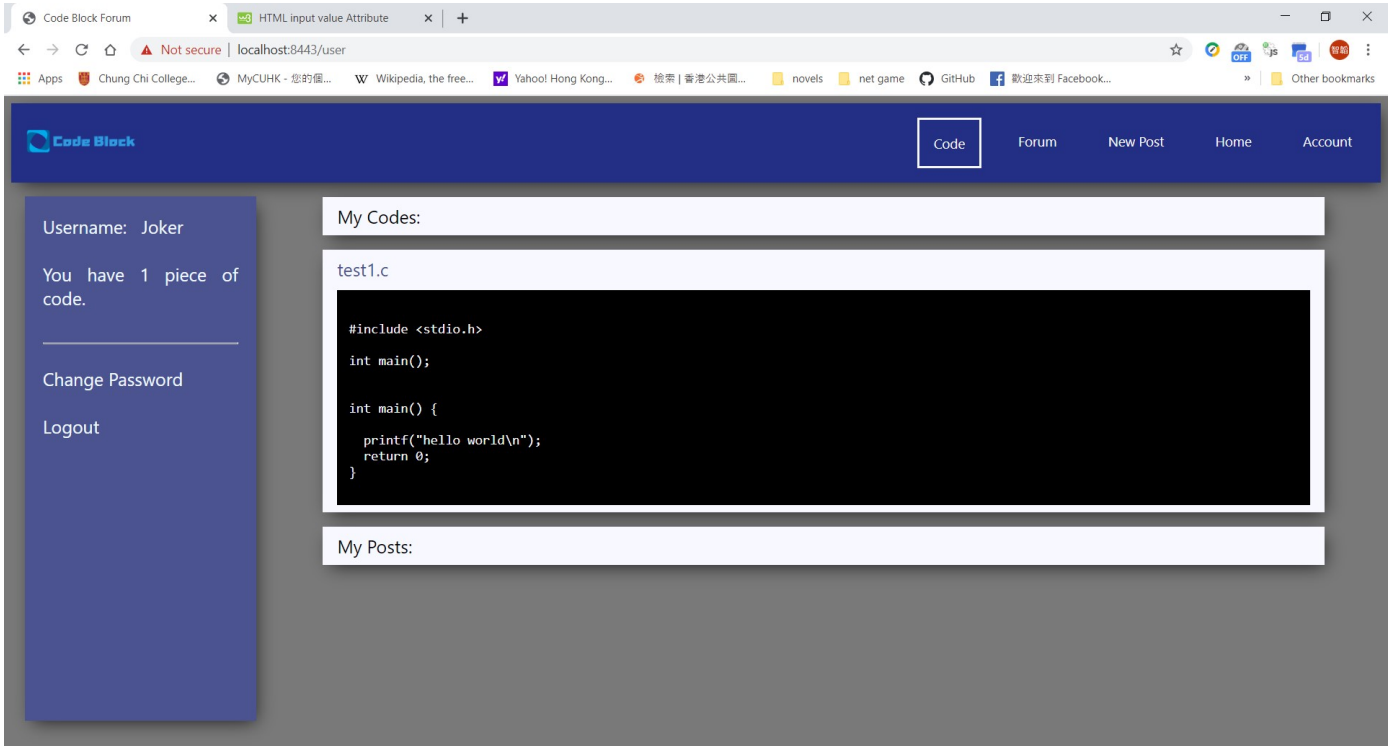
New Code Writing

For successful code saving, the system should redirect the client to a page showing the code stored. And For visualised code editor, the system should show warning when blocks are misplaced.

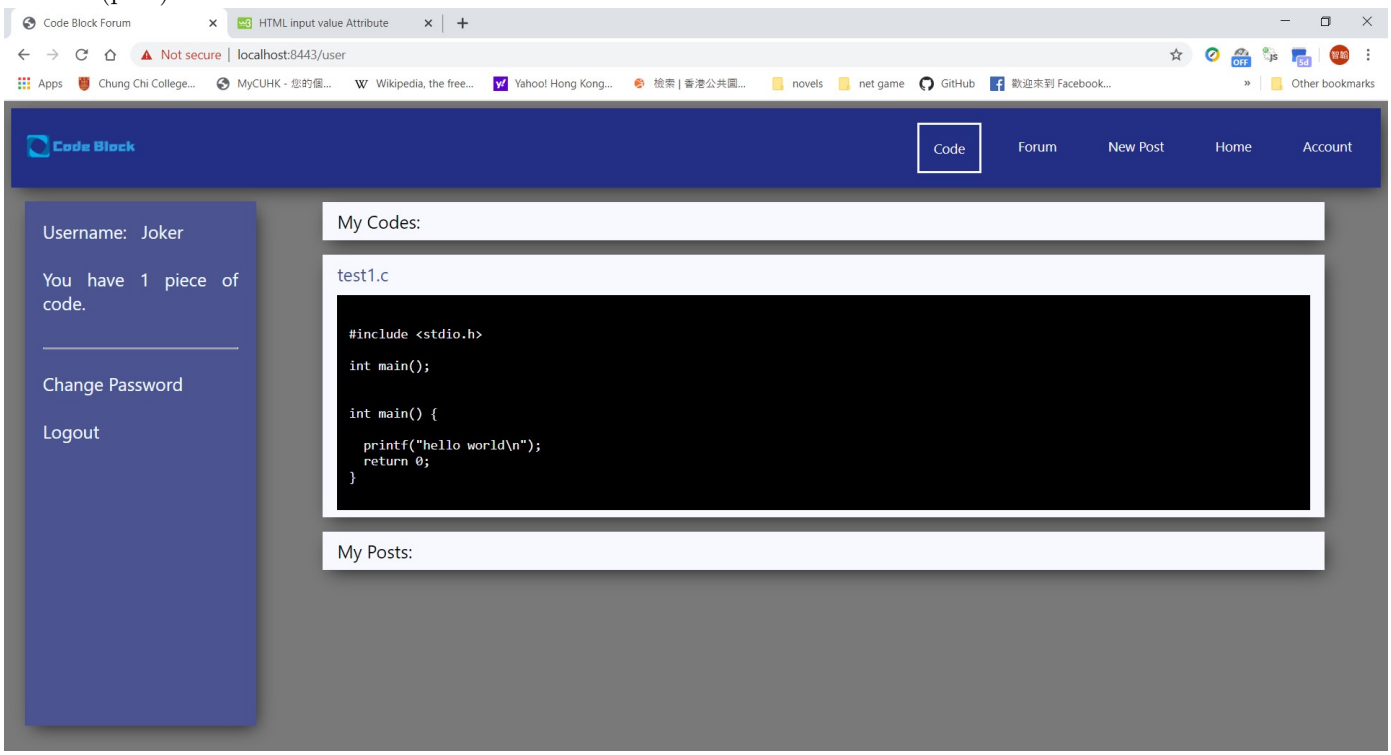
Case 1: (pass)

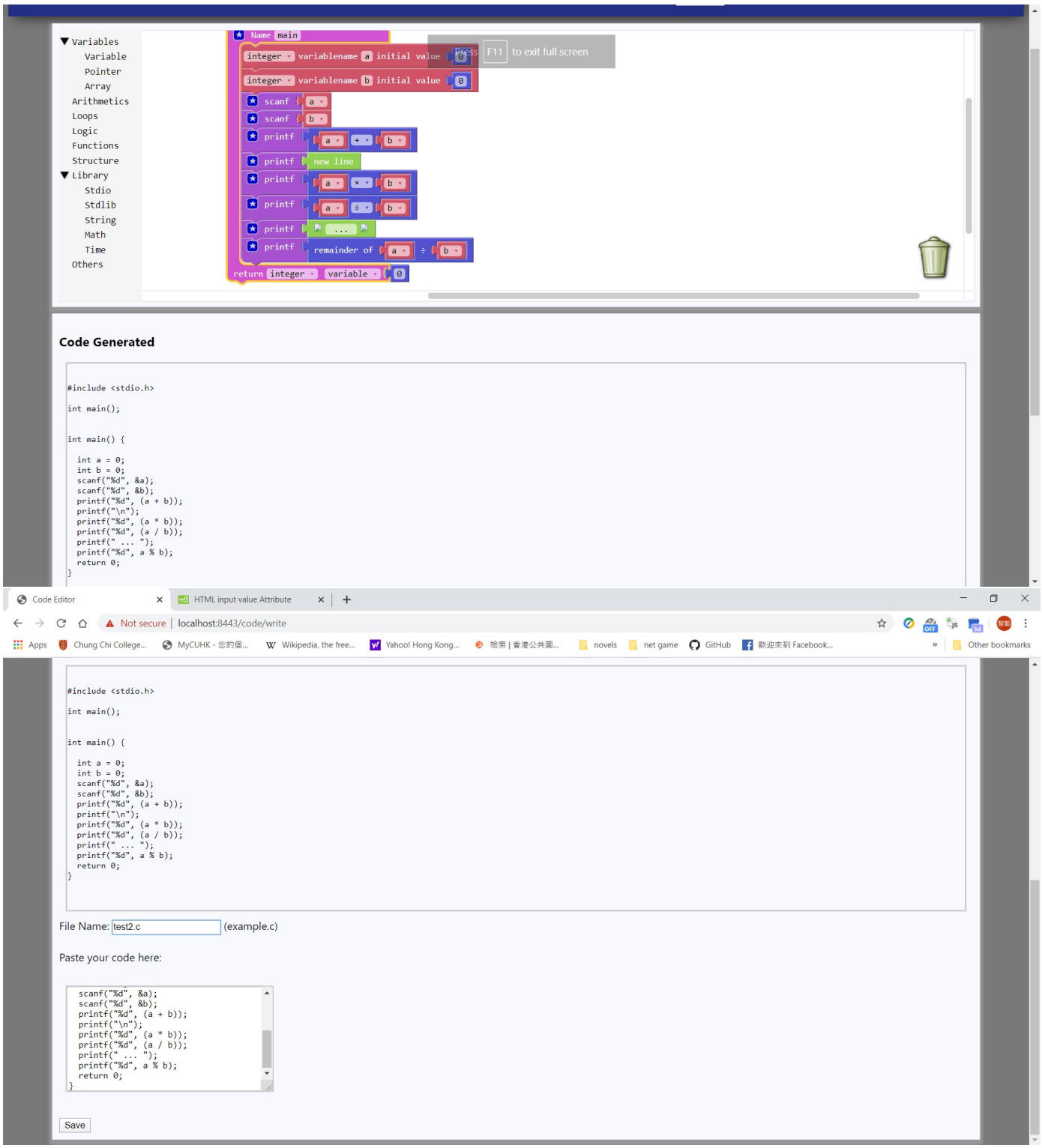

Code Block Forum
HTML input value Attribute
Not secure | localhost:8443/user
Code Block
Code
Forum
New Post
Home
Account
Username: Joker
You don't have any code!
Change Password
Logout
Hello new user
Go and write your code
My Posts:
Code Editor
HTML input value Attribute
Not secure | localhost:8443/code/write
Variables
Arithmetics
Loops
Logic
Functions
Structure
Library
Stdio
Stdlib
String
Math
Time
Others
Name main
printf hello world\n
return integer variable 0
Code Generated
#include <stdio.h>
int main();
int main() {
printf("hello world\n");
return 0;
}

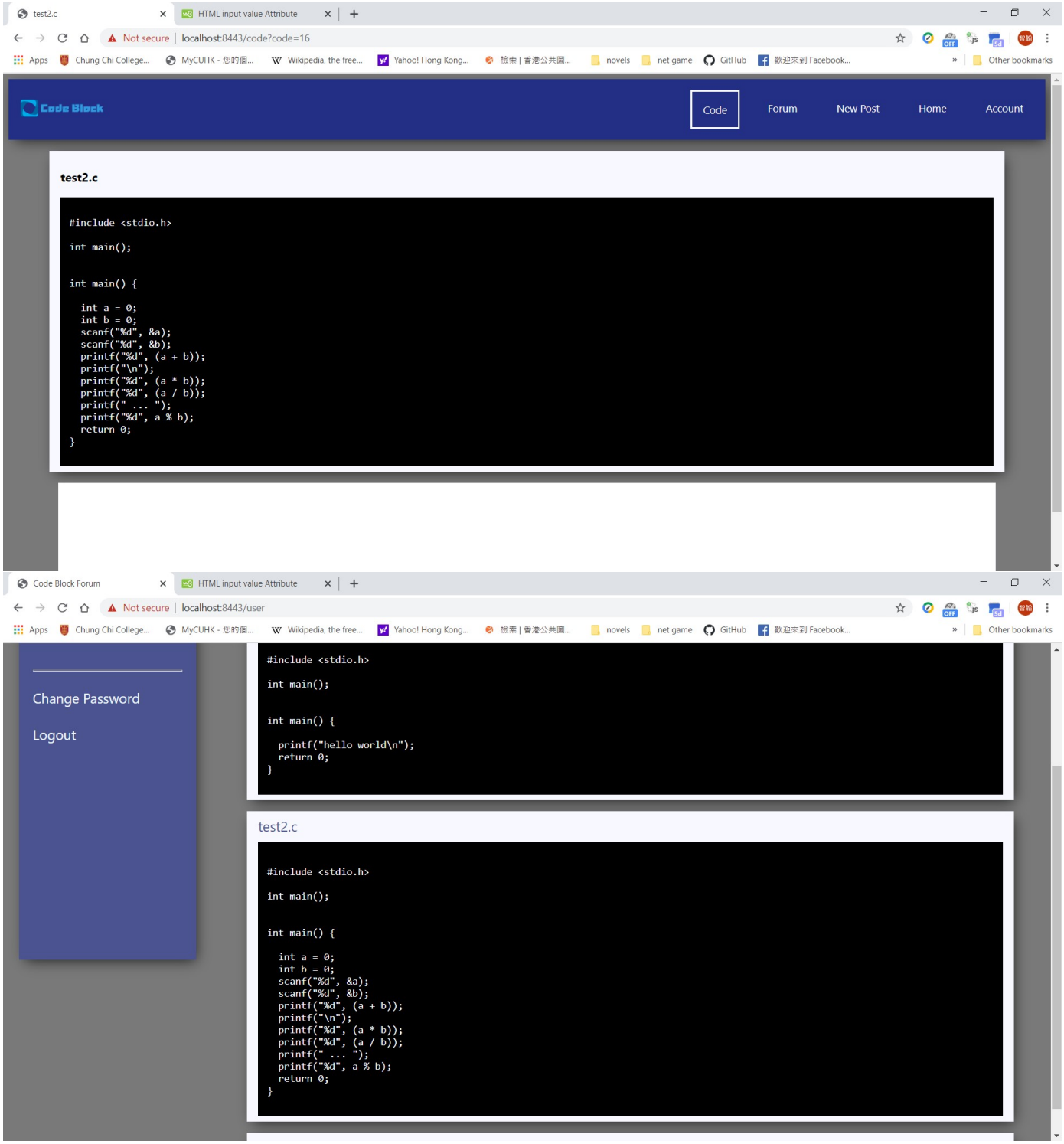

Code Editor
HTML input value Attribute
Not secure | localhost:8443/code/write
Code Generated
#include <stdio.h>
int main();
int main() {
printf("hello world\n");
return 0;
}
File Name: test1.c (example.c)
Paste your code here:
#include <stdio.h>
int main();
int main() {
printf("hello world\n");
return 0;
}
Save
test1.c
HTML input value Attribute
Not secure | localhost:8443/code?code=15
Code Block
Code
Forum
New Post
Home
Account
test1.c
#include <stdio.h>
int main();
int main() {
printf("hello world\n");
return 0;
}
Run


Code Block Forum
HTML input value Attribute
Not secure | localhost:8443/user
Code Block
Code
Forum
New Post
Home
Account
Username: Joker
You have 1 piece of code.
Change Password
Logout
My Codes:
test1.c
#include <stdio.h>
int main();
int main() {
printf("hello world\n");
return 0;
}
My Posts:

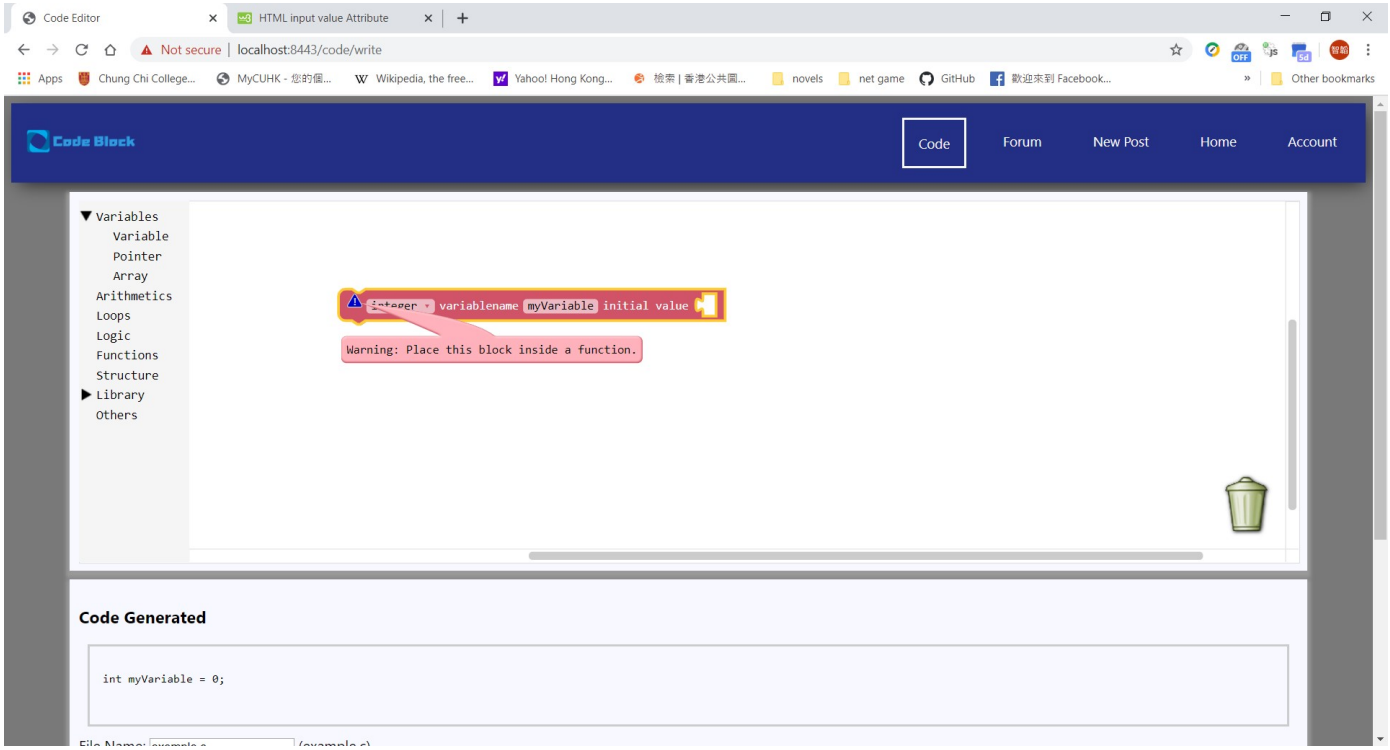
Case 2: (pass)


Code Block Forum
HTML input value Attribute
Not secure | localhost:8443/user
Code Block
Code
Forum
New Post
Home
Account
Username: Joker
You have 1 piece of code.
Change Password
Logout
My Codes:
test1.c
#include <stdio.h>
int main();
int main() {
printf("hello world\n");
return 0;
}
My Posts:

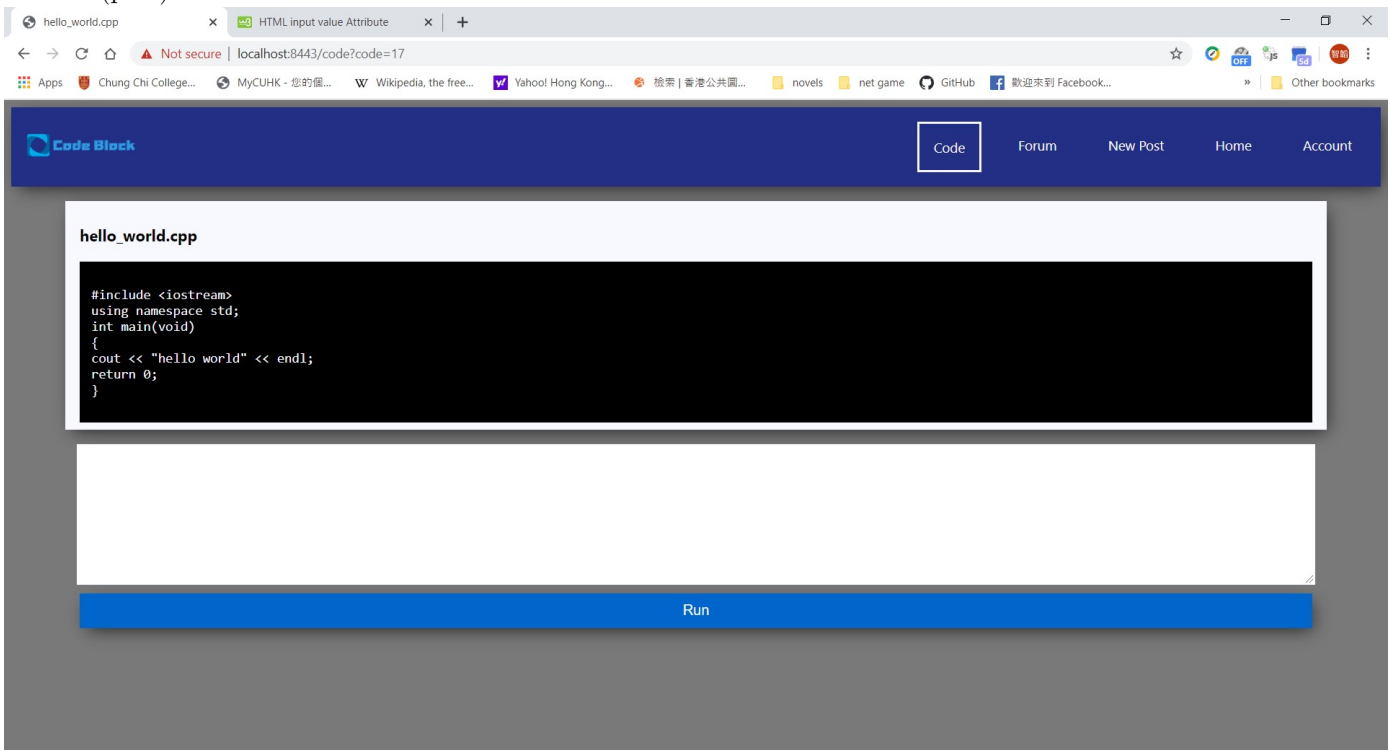

Variables
Variable
Pointer
Array
Arithmetics
Loops
Logic
Functions
Structure
Library
Stdio
Stdlib
String
Math
Time
Others
Name main
integer variablename a initial value 0
integer variablename b initial value 0
scanf a
scanf b
printf a + b
printf new line
printf a × b
printf a ÷ b
printf ...
printf remainder of a ÷ b
return integer variable 0
Press F11 to exit full screen
Code Generated
#include <stdio.h>
int main();
int main() {
int a = 0;
int b = 0;
scanf("%d", &a);
scanf("%d", &b);
printf("%d", (a + b));
printf("\n");
printf("%d", (a * b));
printf("%d", (a / b));
printf(" ... ");
printf("%d", a % b);
return 0;
}
Code Editor
HTML input value Attribute
Not secure | localhost:8443/code/write
Apps
Chung Chi College...
MyCUHK - 您的個...
Wikipedia, the free...
Yahoo! Hong Kong...
檢索 | 香港公共圖...
novels
net game
GitHub
歡迎來到 Facebook...
Other bookmarks
#include <stdio.h>
int main();
int main() {
int a = 0;
int b = 0;
scanf("%d", &a);
scanf("%d", &b);
printf("%d", (a + b));
printf("\n");
printf("%d", (a * b));
printf("%d", (a / b));
printf(" ... ");
printf("%d", a % b);
return 0;
}
File Name: test2.c (example.c)
Paste your code here:
scanf("%d", &a);
scanf("%d", &b);
printf("%d", (a + b));
printf("\n");
printf("%d", (a * b));
printf("%d", (a / b));
printf(" ... ");
printf("%d", a % b);
return 0;
}
Save


test2.c
HTML input value Attribute
Not secure | localhost:8443/code?code=16
Code Block
Code
Forum
New Post
Home
Account
test2.c
#include <stdio.h>
int main();
int main() {
int a = 0;
int b = 0;
scanf("%d", &a);
scanf("%d", &b);
printf("%d", (a + b));
printf("\n");
printf("%d", (a * b));
printf("%d", (a / b));
printf(" ... ");
printf("%d", a % b);
return 0;
}
Code Block Forum
HTML input value Attribute
Not secure | localhost:8443/user
Change Password
Logout
#include <stdio.h>
int main();
int main() {
printf("hello world\n");
return 0;
}
test2.c
#include <stdio.h>
int main();
int main() {
int a = 0;
int b = 0;
scanf("%d", &a);
scanf("%d", &b);
printf("%d", (a + b));
printf("\n");
printf("%d", (a * b));
printf("%d", (a / b));
printf(" ... ");
printf("%d", a % b);
return 0;
}

Case 3: (pass)



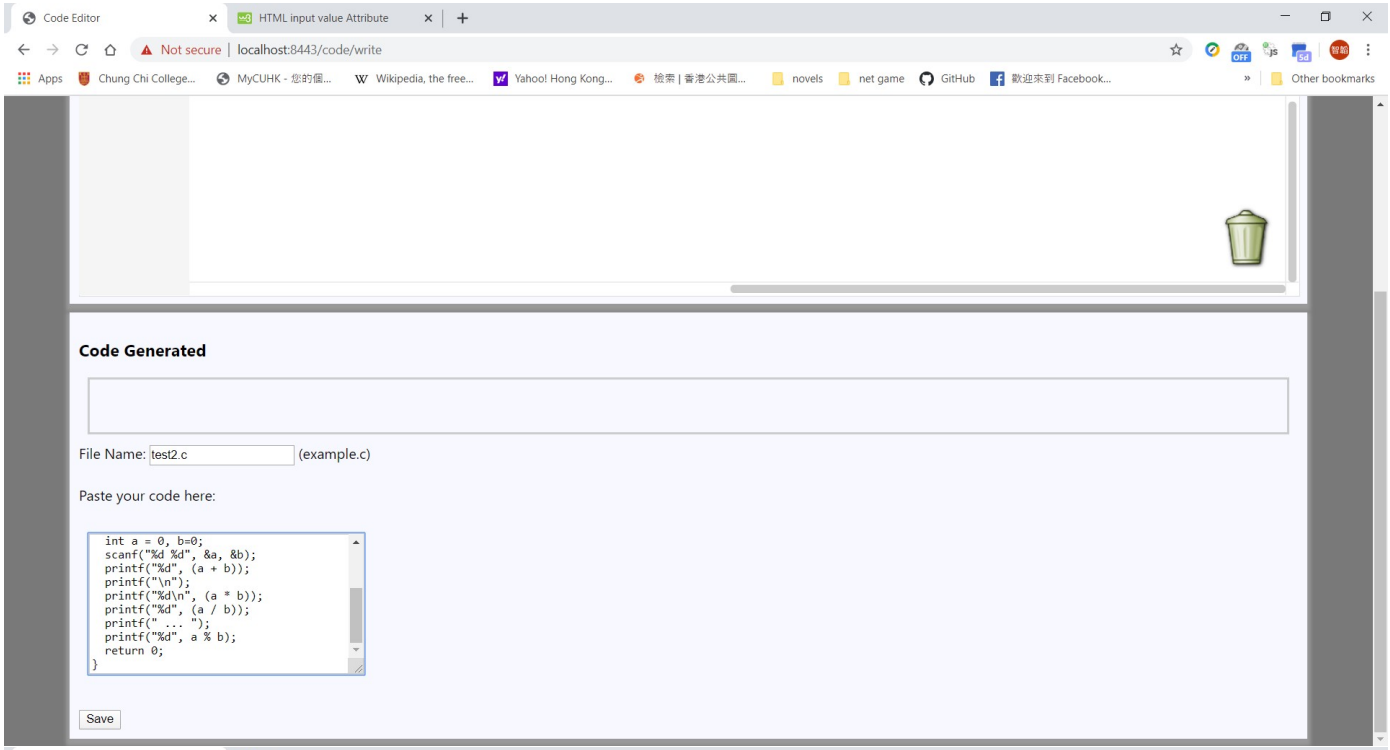
Case 4: (pass)

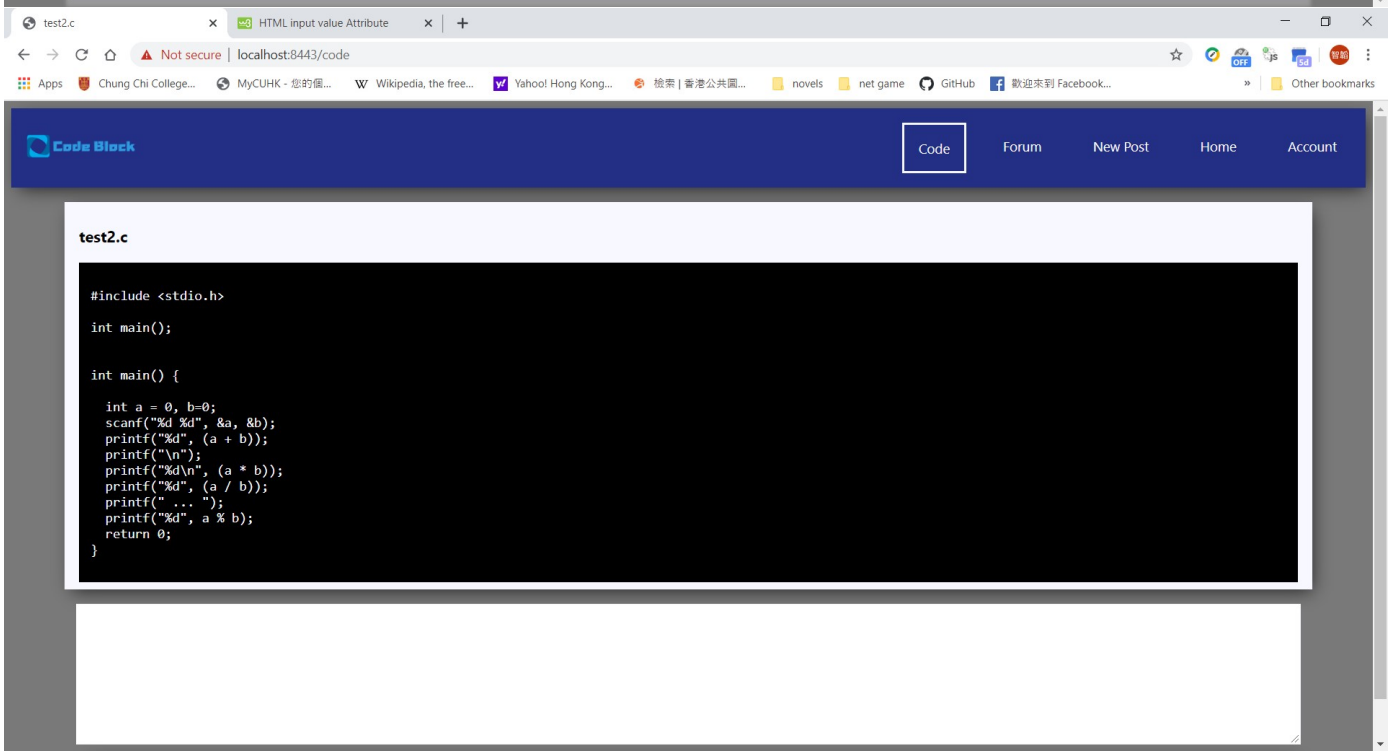


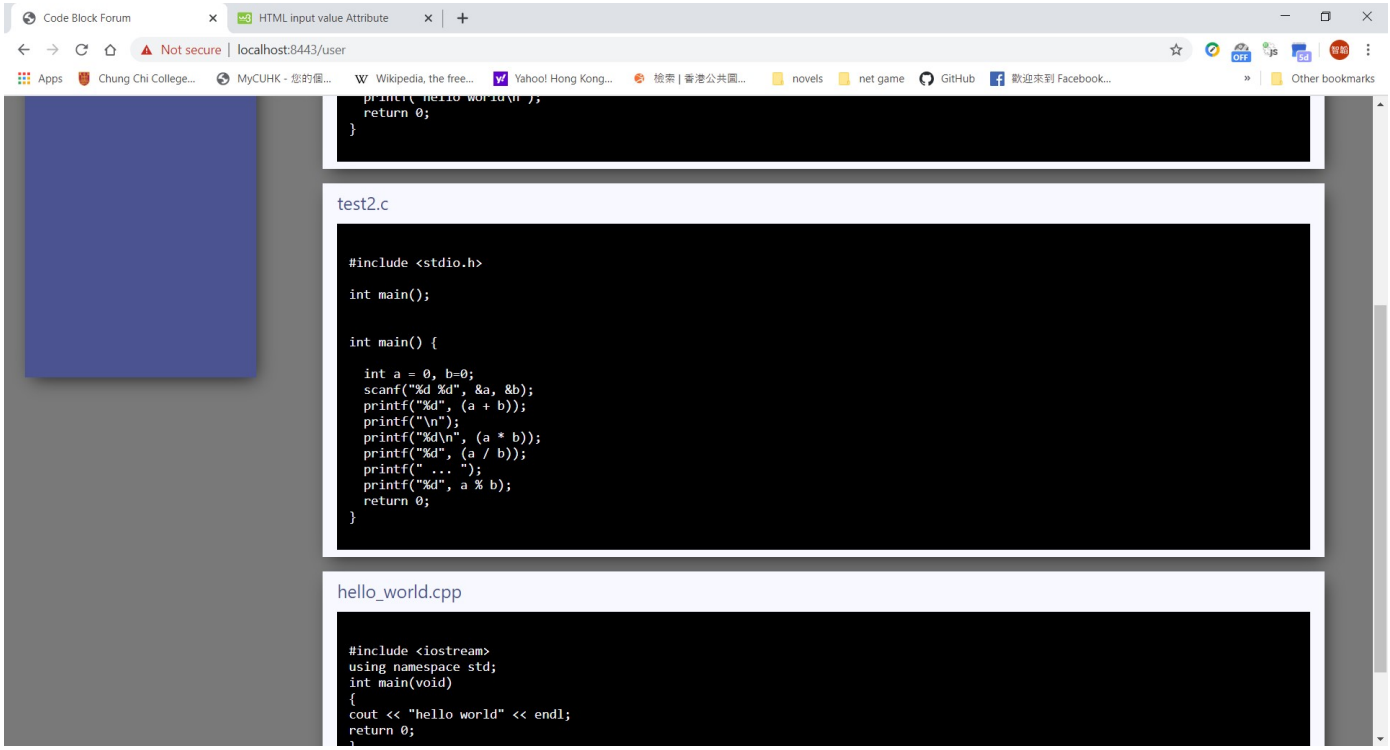
Code Update

For successful code saving, the system should redirect the client to a page showing the code stored.

Case 1: (pass)


Code Editor
HTML input value Attribute
Not secure | localhost:8443/code/write
Code Generated
File Name: test2.c (example.c)
Paste your code here:
int a = 0, b=0;
scanf("%d %d", &a, &b);
printf("%d", (a + b));
printf("\n");
printf("%d\n", (a * b));
printf("%d", (a / b));
printf(" ... ");
printf("%d", a % b);
return 0;
}
Save


test2.c
HTML input value Attribute
Not secure | localhost:8443/code
Code Block
Code
Forum
New Post
Home
Account
test2.c
#include <stdio.h>
int main();
int main() {
int a = 0, b=0;
scanf("%d %d", &a, &b);
printf("%d", (a + b));
printf("\n");
printf("%d\n", (a * b));
printf("%d", (a / b));
printf(" ... ");
printf("%d", a % b);
return 0;
}



5.9 Code Compilation and Running

5.9.1 Purpose

This feature allows user to compile and execute the code. The system should return the compilation and execution result of the program.
The testcases observes upon the result of the compilation and execution of the source code written in previous part. Code fetching is also tested in this part.

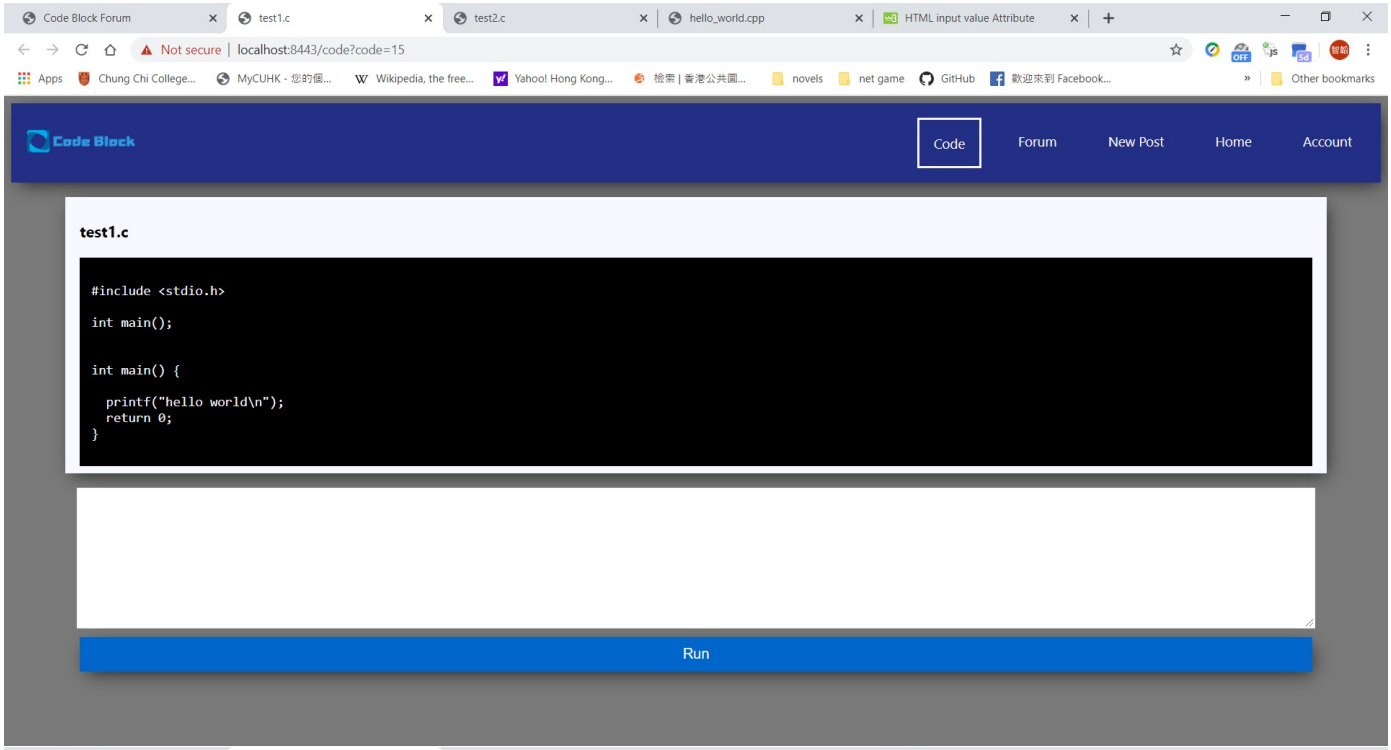
5.9.2 Inputs

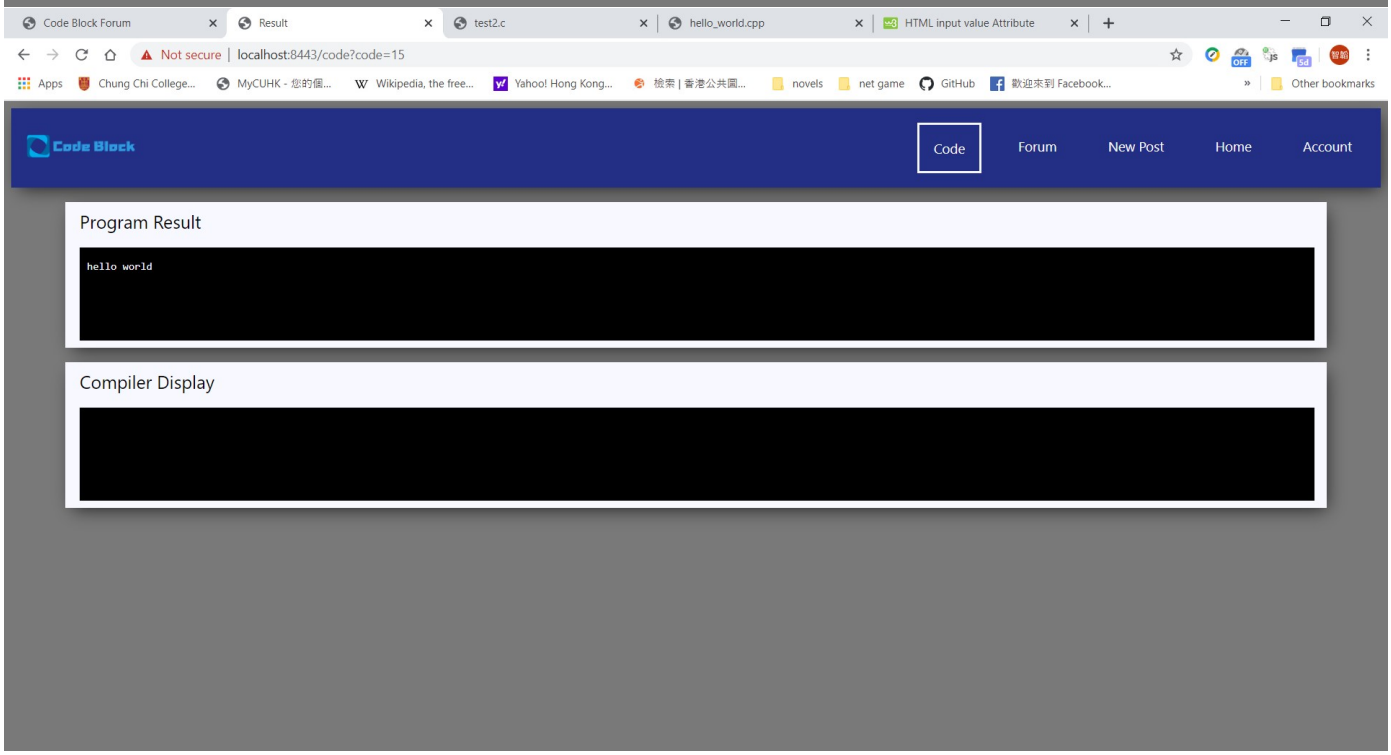
1. test1.c (valid code)
2. test2.c (valid code with proper input)
3. hello_world.cpp (compilation error)
4. gcd.c (valid code without proper input)
5. gcd.c (valid code with proper input)
6. code id = 3 (invalid code id)

5.9.3 Expected Outputs & Pass/Fail Criteria

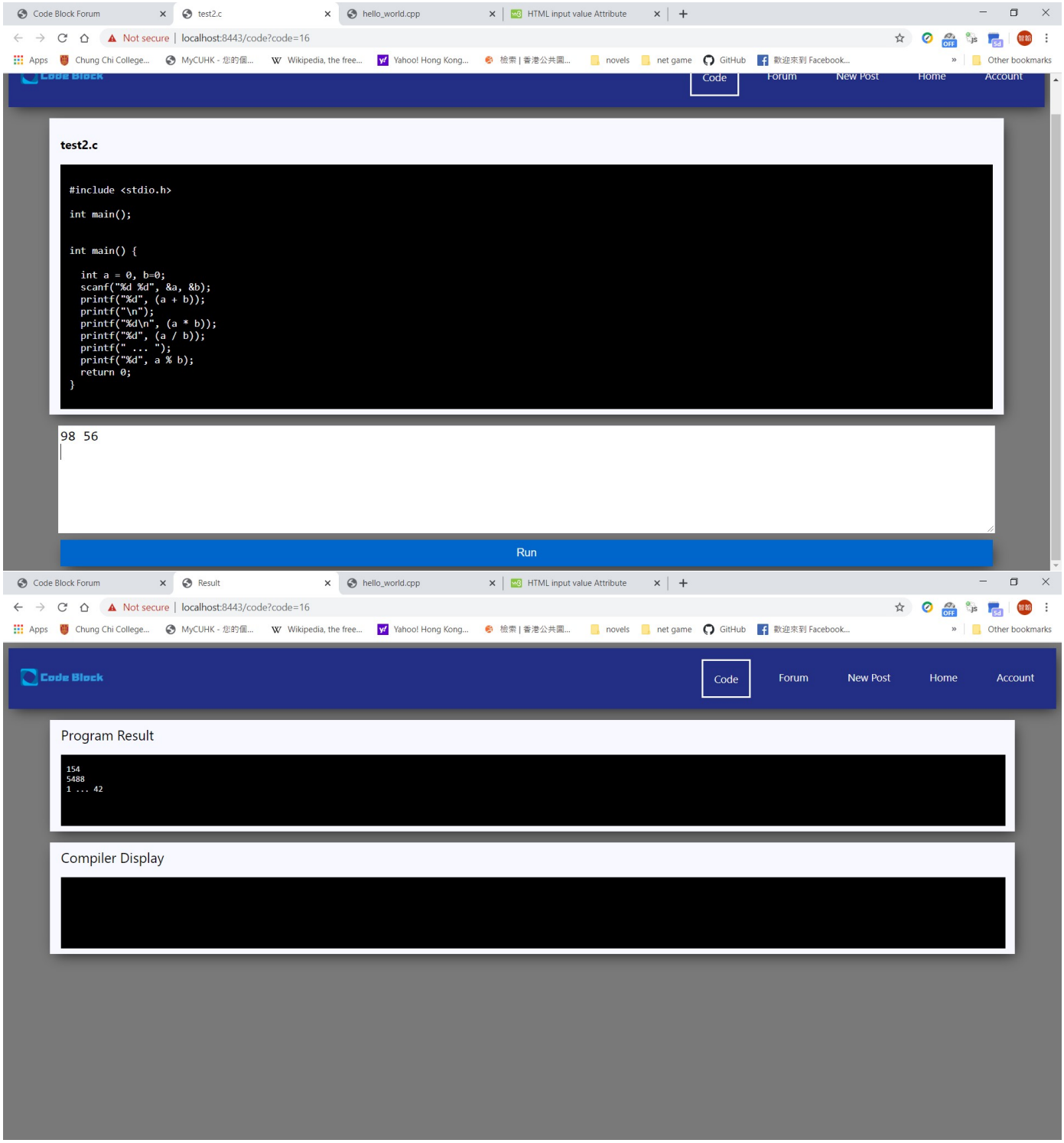
The system should return error message generated by the compiler upon compilation failure. If the compiler successfully compiled the code, the system should return the execution result of the compiler. Yet, upon invalid code requests, the system should redirect user to page 404.

Case 1: (pass)

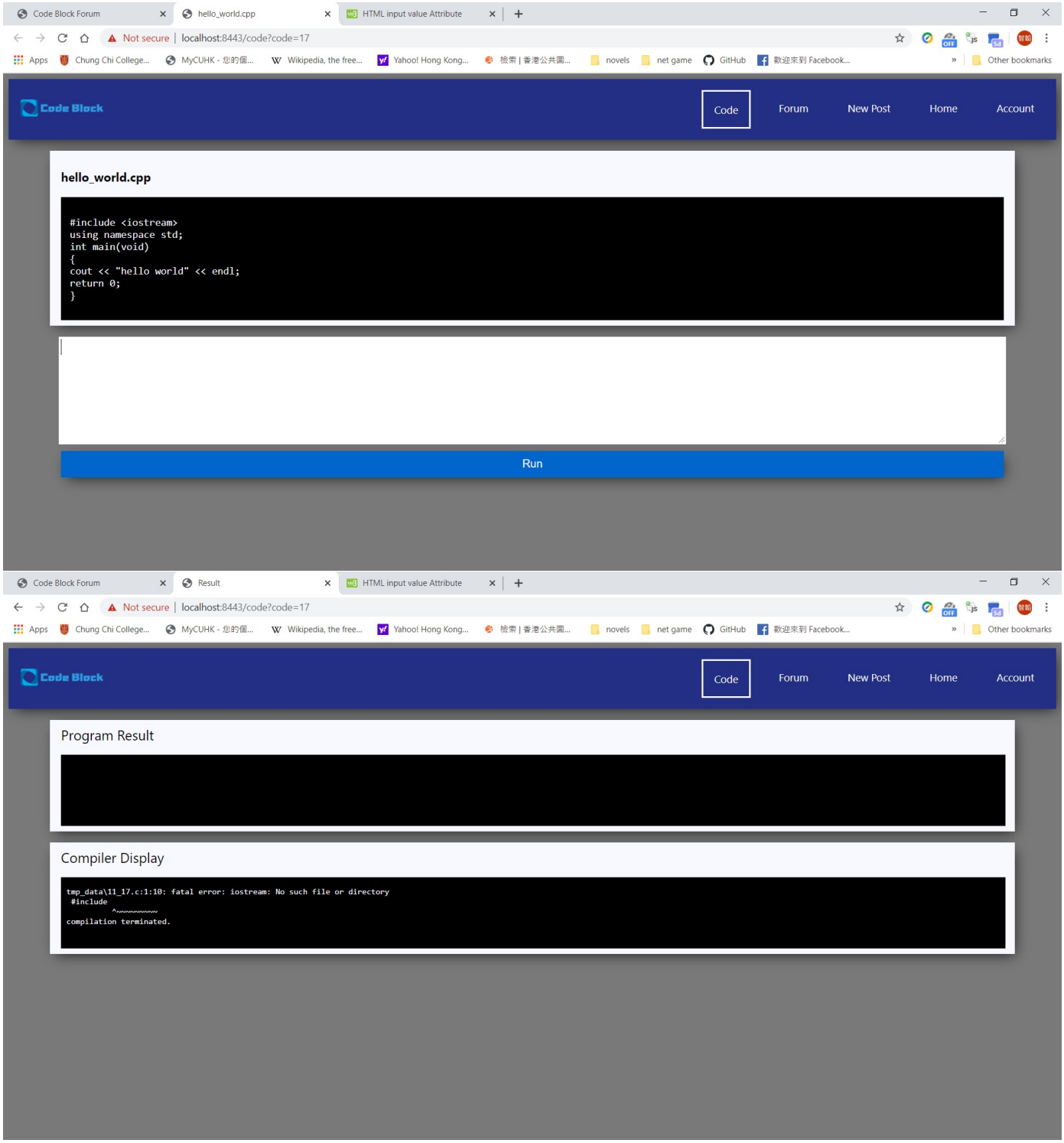

Code Block Forum
test1.c
test2.c
hello_world.cpp
HTML input value Attribute
Not secure | localhost:8443/code?code=15
Code Block
Code
Forum
New Post
Home
Account
test1.c
#include <stdio.h>
int main();
int main() {
printf("hello world\n");
return 0;
}
Run


Code Block Forum
Result
test2.c
hello_world.cpp
HTML input value Attribute
Not secure | localhost:8443/code?code=15
Code Block
Code
Forum
New Post
Home
Account
Program Result
hello world
Compiler Display

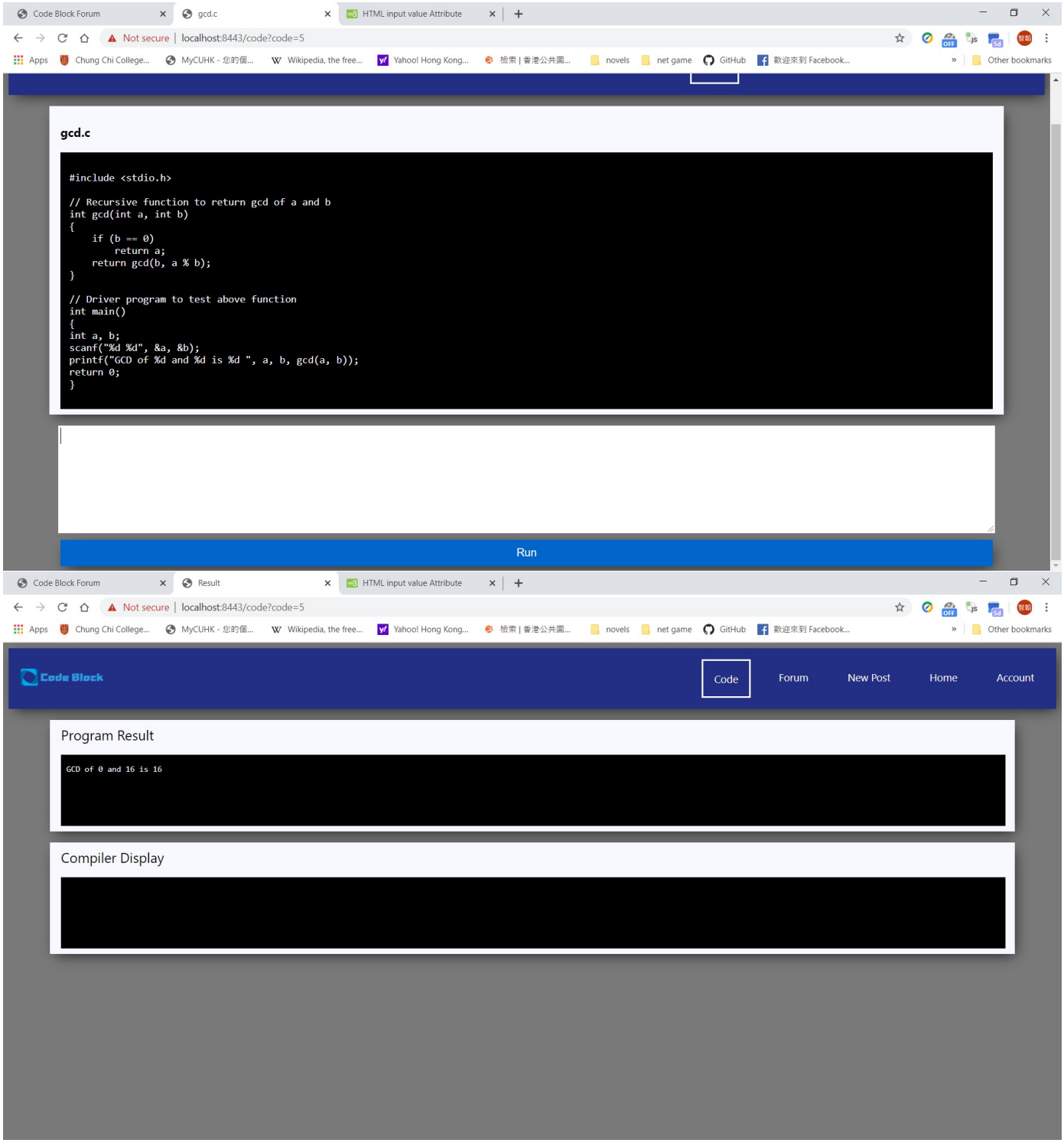
Case 2: (pass)


Code Block Forum
test2.c
hello_world.cpp
HTML input value Attribute
Not secure | localhost:8443/code?code=16
Code
Forum
New Post
Home
Account
test2.c
#include <stdio.h>
int main();
int main() {
int a = 0, b=0;
scanf("%d %d", &a, &b);
printf("%d", (a + b));
printf("\n");
printf("%d\n", (a * b));
printf("%d", (a / b));
printf(" ... ");
printf("%d", a % b);
return 0;
}
98 56
Run
Result
Program Result
154
5488
1 ... 42
Compiler Display

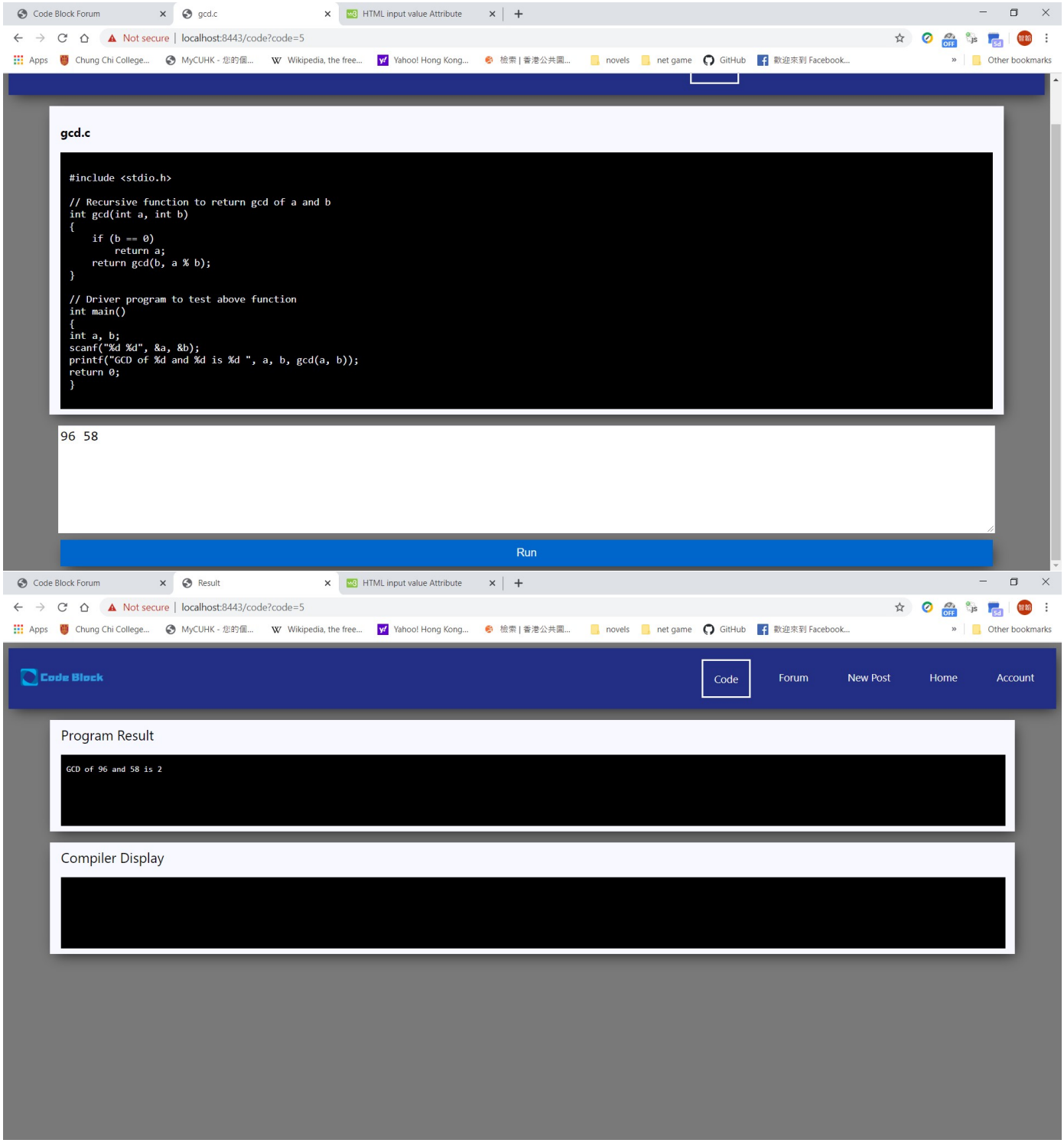
Case 3: (pass)


Code Block Forum
hello_world.cpp
HTML input value Attribute
Not secure | localhost:8443/code?code=17
Code Block
Code
Forum
New Post
Home
Account
hello_world.cpp
#include <iostream>
using namespace std;
int main(void)
{
cout << "hello world" << endl;
return 0;
}
Run
Code Block Forum
Result
HTML input value Attribute
Not secure | localhost:8443/code?code=17
Code Block
Code
Forum
New Post
Home
Account
Program Result
Compiler Display
tmp_data\11_17.c:1:10: fatal error: iostream: No such file or directory
#include
^~~~~~~~~~
compilation terminated.

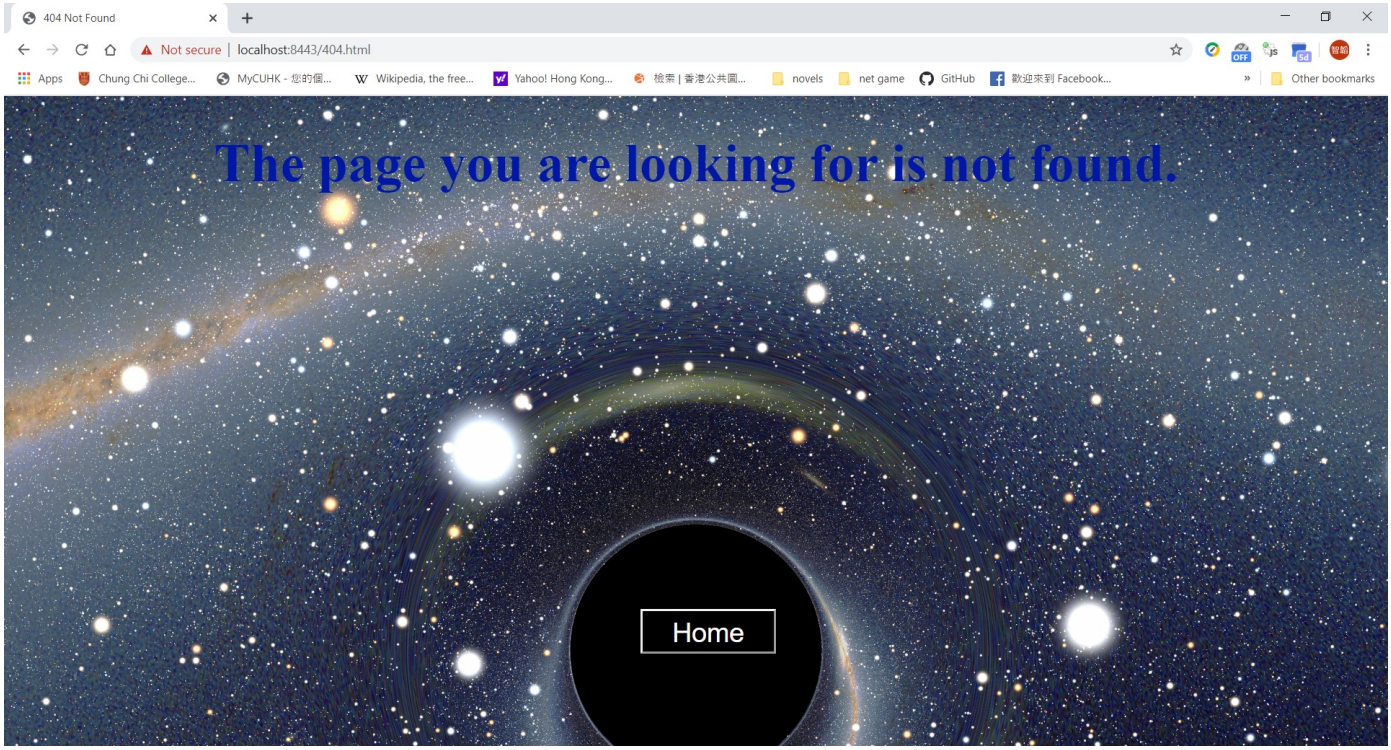
Case 4: (pass)


Code Block Forum
gcd.c
HTML input value Attribute
Not secure | localhost:8443/code?code=5
gcd.c
#include <stdio.h>
// Recursive function to return gcd of a and b
int gcd(int a, int b)
{
if (b == 0)
return a;
return gcd(b, a % b);
}
// Driver program to test above function
int main()
{
int a, b;
scanf("%d %d", &a, &b);
printf("GCD of %d and %d is %d ", a, b, gcd(a, b));
return 0;
}
Run
Code Block Forum
Result
HTML input value Attribute
Not secure | localhost:8443/code?code=5
Code Block
Code
Forum
New Post
Home
Account
Program Result
GCD of 0 and 16 is 16
Compiler Display

Case 5: (pass)


Code Block Forum
gcd.c
HTML input value Attribute
Not secure | localhost:8443/code?code=5
gcd.c
#include <stdio.h>
// Recursive function to return gcd of a and b
int gcd(int a, int b)
{
if (b == 0)
return a;
return gcd(b, a % b);
}
// Driver program to test above function
int main()
{
int a, b;
scanf("%d %d", &a, &b);
printf("GCD of %d and %d is %d ", a, b, gcd(a, b));
return 0;
}
96 58
Run
Code Block Forum
Result
HTML input value Attribute
Not secure | localhost:8443/code?code=5
Code Block
Code
Forum
New Post
Home
Account
Program Result
GCD of 96 and 58 is 2
Compiler Display

Case 6: (pass)



5.10 Forum

5.10.1 Purpose

This feature allows authenticated users to share their code for discussion here, and authenticated users can also reply to other's posts. Unauthenticated users should be able to read the posts here without reply function. And users can also use the search function to find posts using specific keywords.
Testcases should observe upon the result of fetching posts when using the forum feature. And tests will be carried upon post directing from forum, reply, new posts and forum searching.

5.10.2 Inputs

Post Fetching

Used testcases (title, post id):

1. gcd in C, 25 (valid post, Unauthenticated user)
2. gcd in C, 25 (valid post, authenticated user)
3. (unavailable), 29 (nonexistent)
4. (reply), 2 (reply)

New Post

1. Unauthenticated user
2. Hello World, `Something here`, test1.c
3. Arithmetics(newline) Here, `a+b\r\n a/b`, test2.c (newlines testing)
4. (empty), (empty), test1.c
5. 12345678910, (empty), (empty)

Reply

1. Unauthenticated user
2. gcd in C
3. gcd in C (empty response)

Search

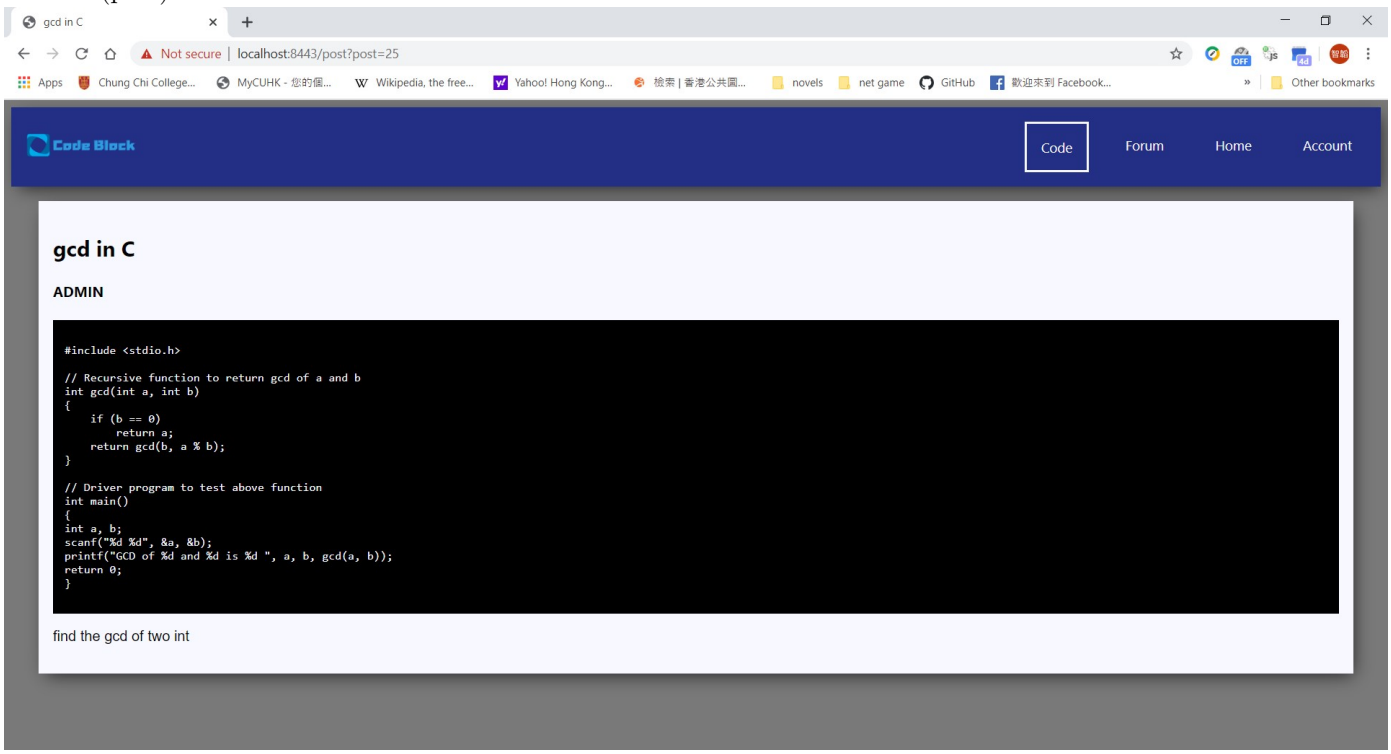
1. C++ C (multiple keyword search)
2. ADMIN (username search)
3. gcd (single keyword search)
4. “OR 1=1” (SQL injection)

5.10.3 Expected Outputs & Pass/Fail Criteria

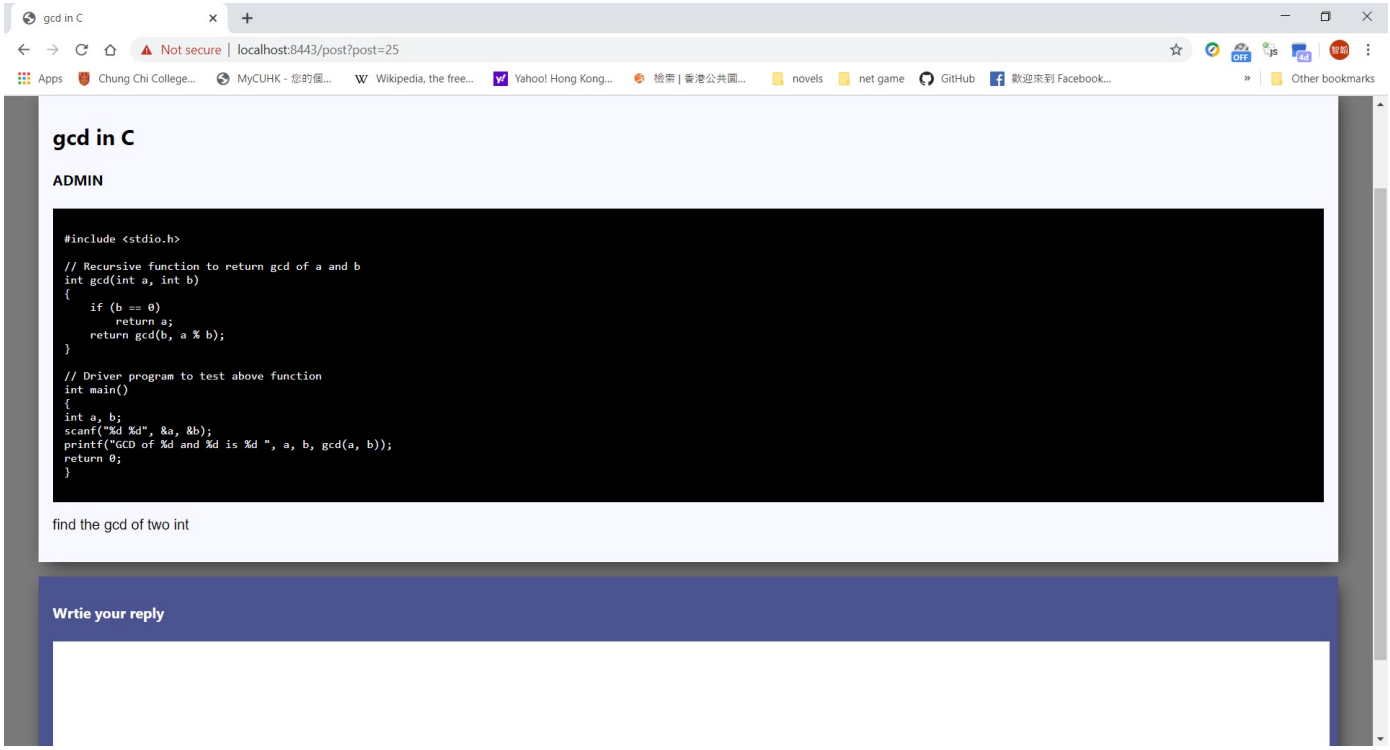
Post Fetching

The system should return the post with post title, content, code, replies and users related in the post upon a valid post request. For invalid post requests, it should redirect the user to page 404.

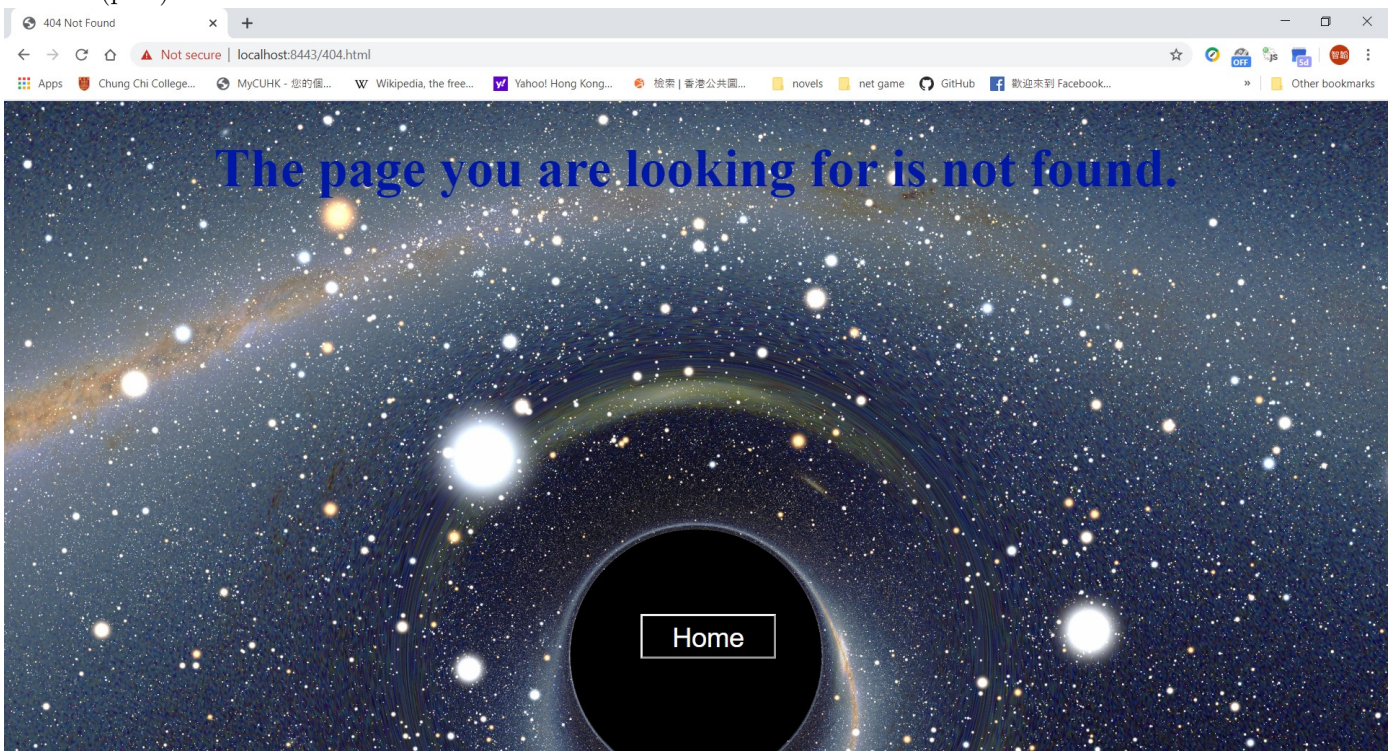
Case 1: (pass)



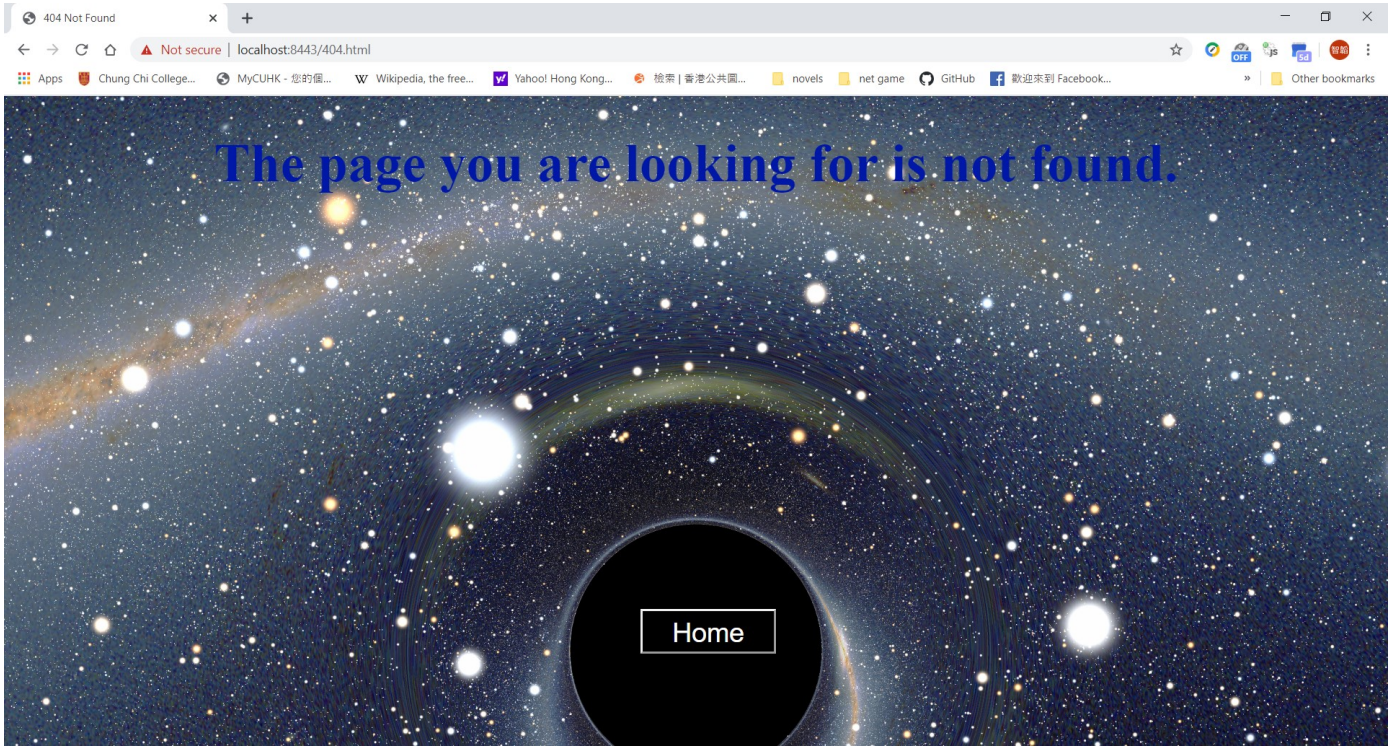
Case 2: (pass)



Case 3: (pass)

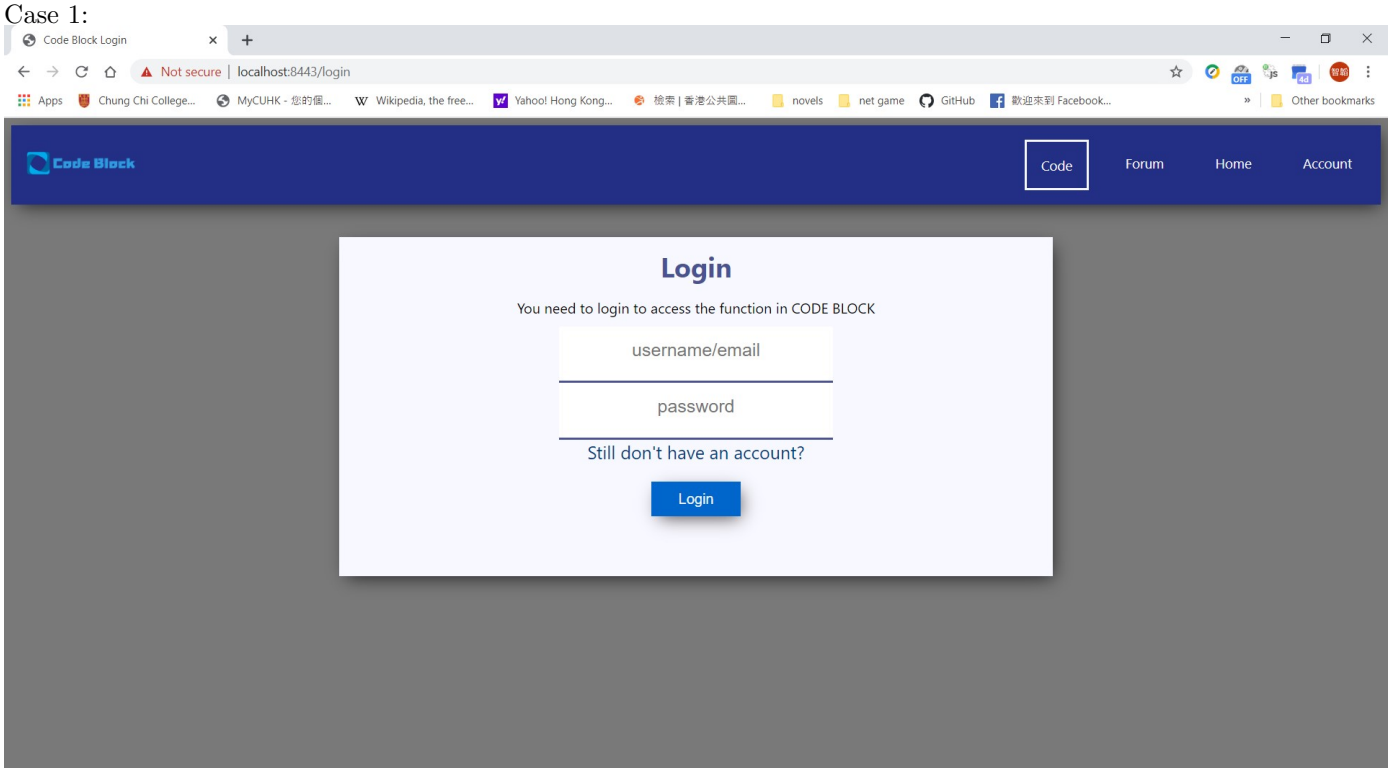


Case 4: (pass)

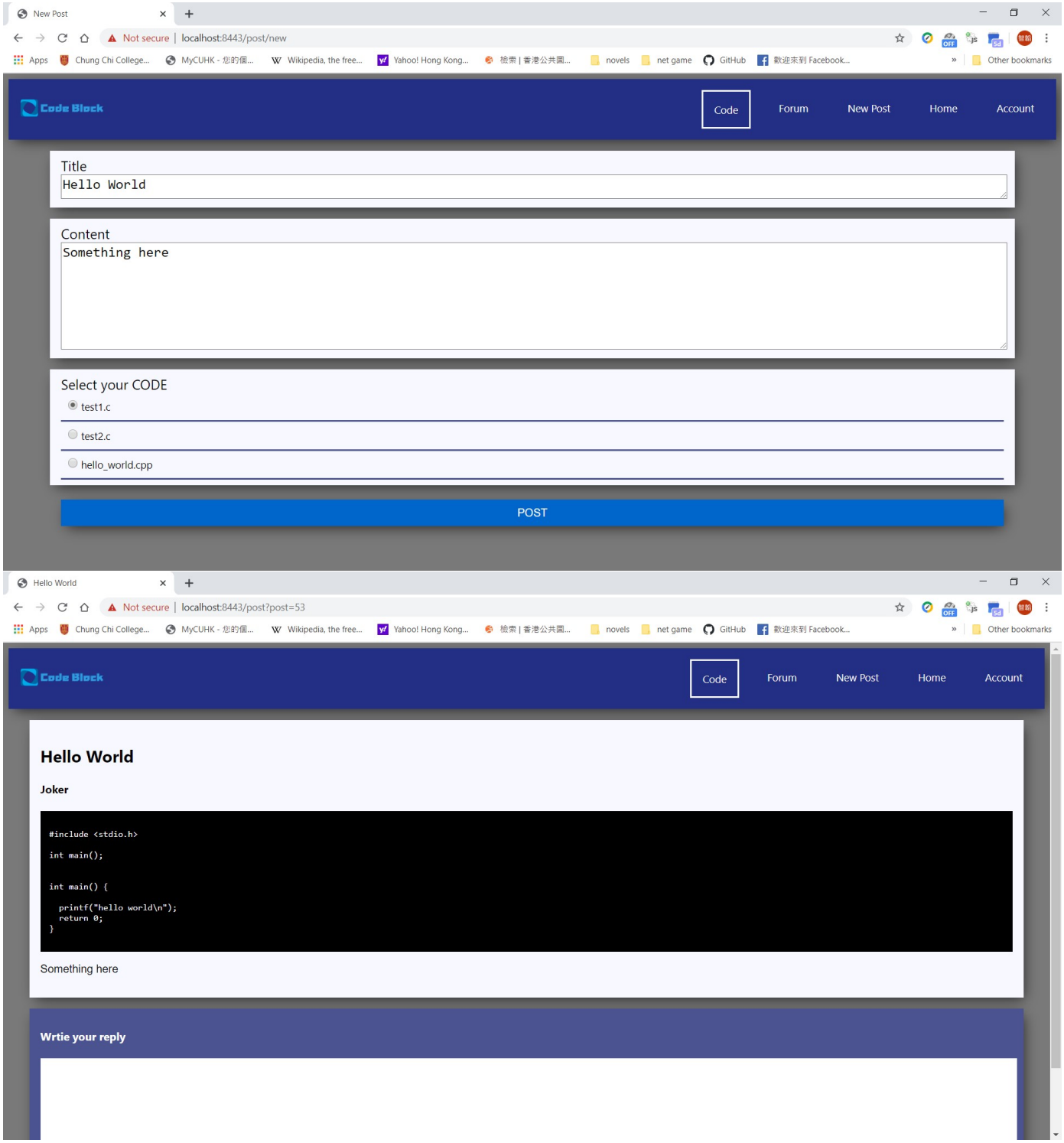


New Post

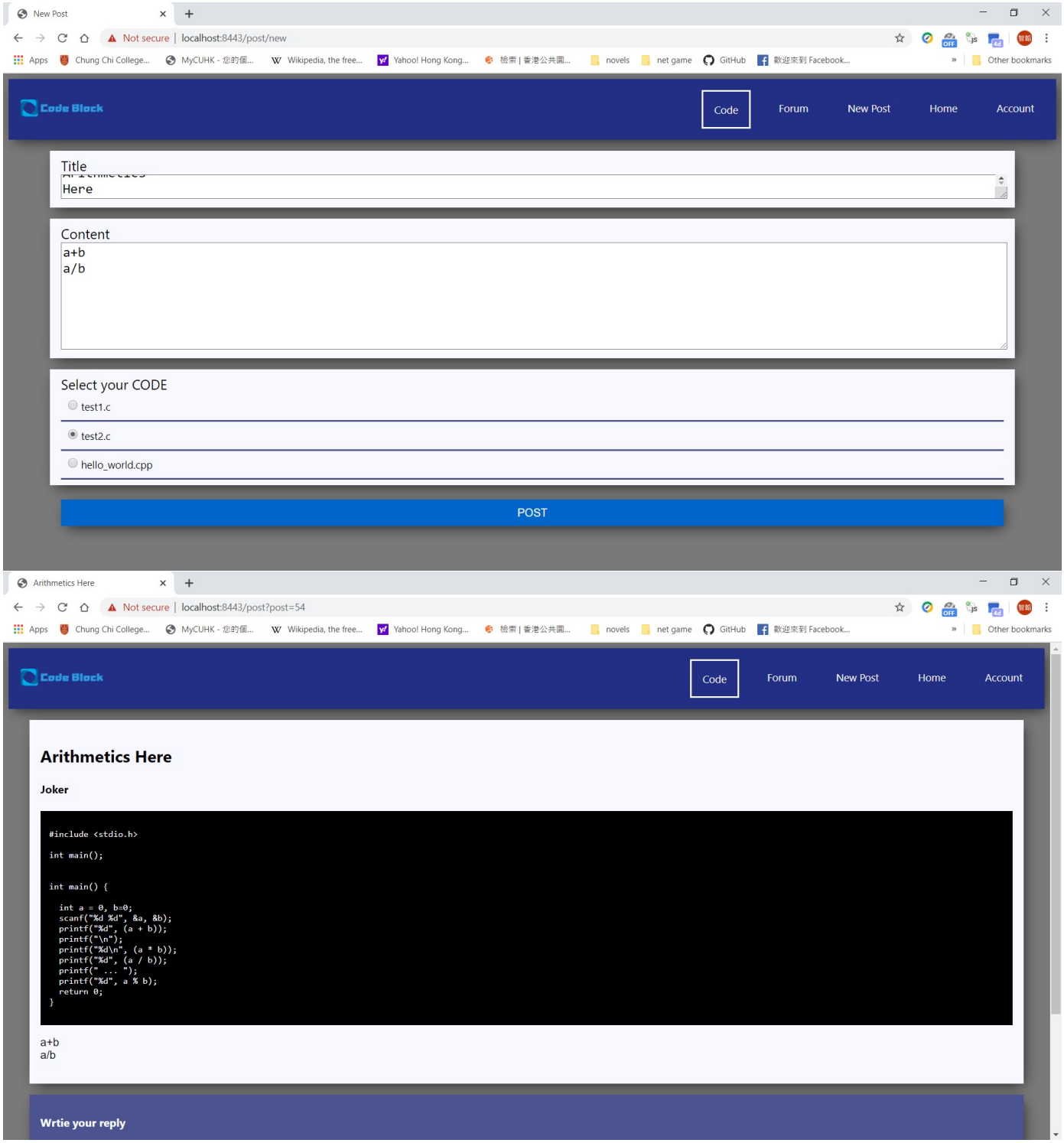
Case 1:



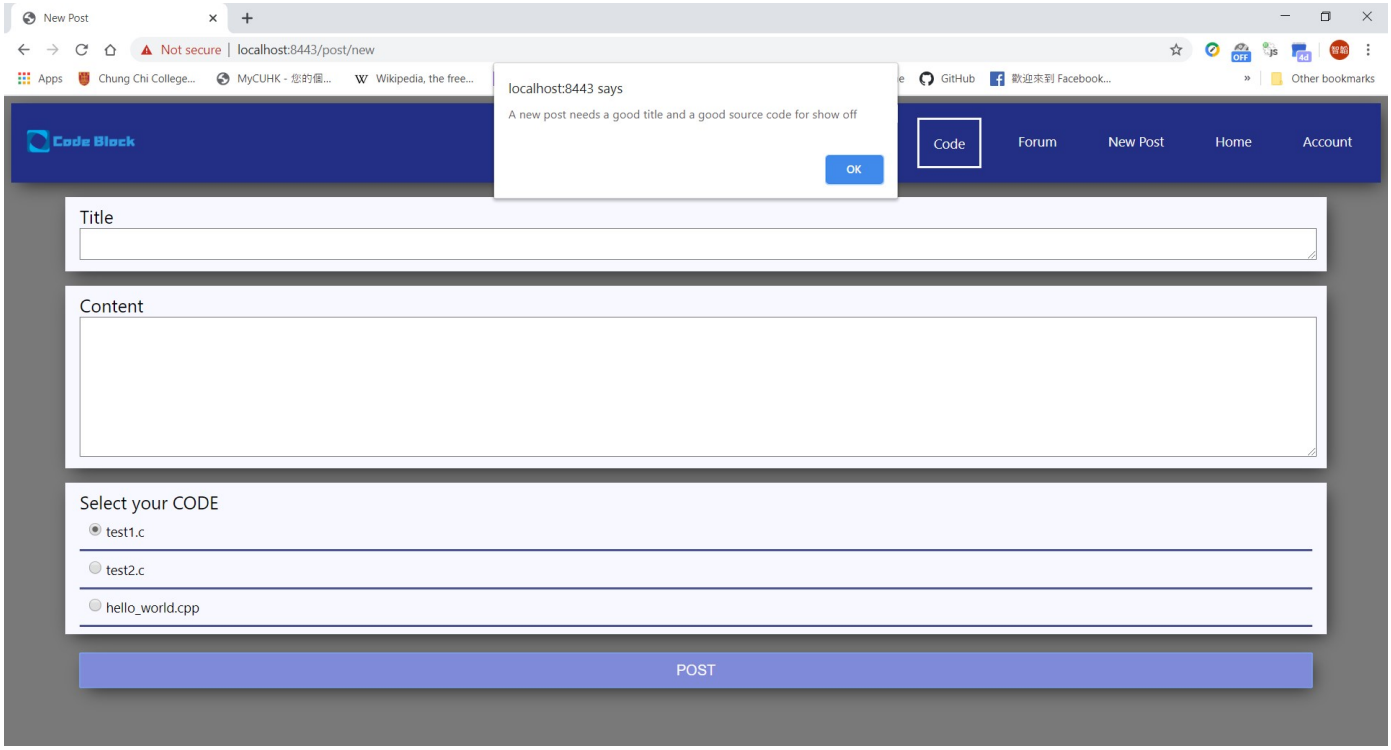
Case 2: (pass)


New Post
Not secure | localhost:8443/post/new
Code Block
Code
Forum
New Post
Home
Account
Title
Hello World
Content
Something here
Select your CODE
test1.c
test2.c
hello_world.cpp
POST
Hello World
Not secure | localhost:8443/post?post=53
Code Block
Code
Forum
New Post
Home
Account
Hello World
Joker
#include <stdio.h>
int main();
int main() {
printf("hello world\n");
return 0;
}
Something here
Wrtie your reply

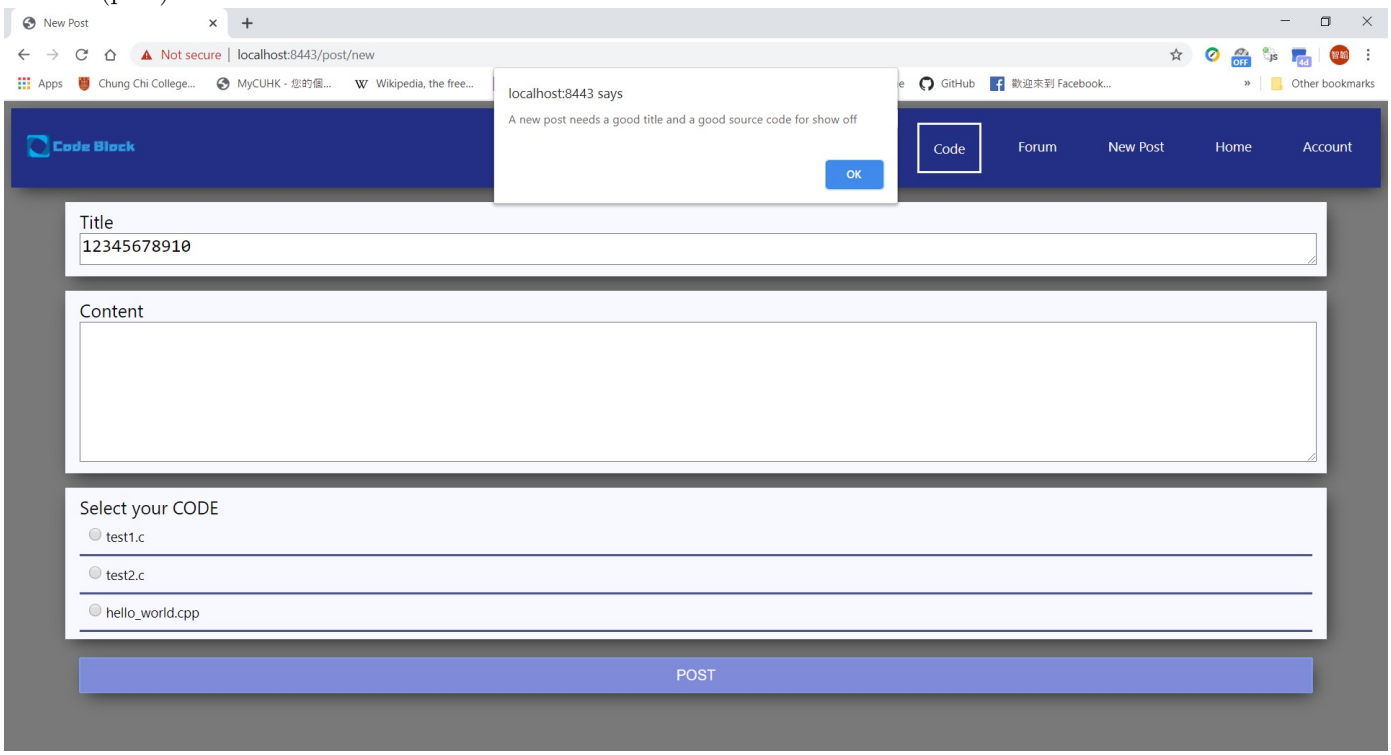
Case 3: (pass)


New Post
Not secure | localhost:8443/post/new
Code Block
Code
Forum
New Post
Home
Account
Title
Here
Content
a+b
a/b
Select your CODE
test1.c
test2.c
hello_world.cpp
POST
Arithmetics Here
Not secure | localhost:8443/post?post=54
Code Block
Code
Forum
New Post
Home
Account
Arithmetics Here
Joker
#include <stdio.h>
int main();
int main() {
int a = 0, b=0;
scanf("%d %d", &a, &b);
printf("%d", (a + b));
printf("\n");
printf("%d\n", (a * b));
printf("%d", (a / b));
printf(" ... ");
printf("%d", a % b);
return 0;
}
a+b
a/b
Wrtie your reply

Case 4: (pass)



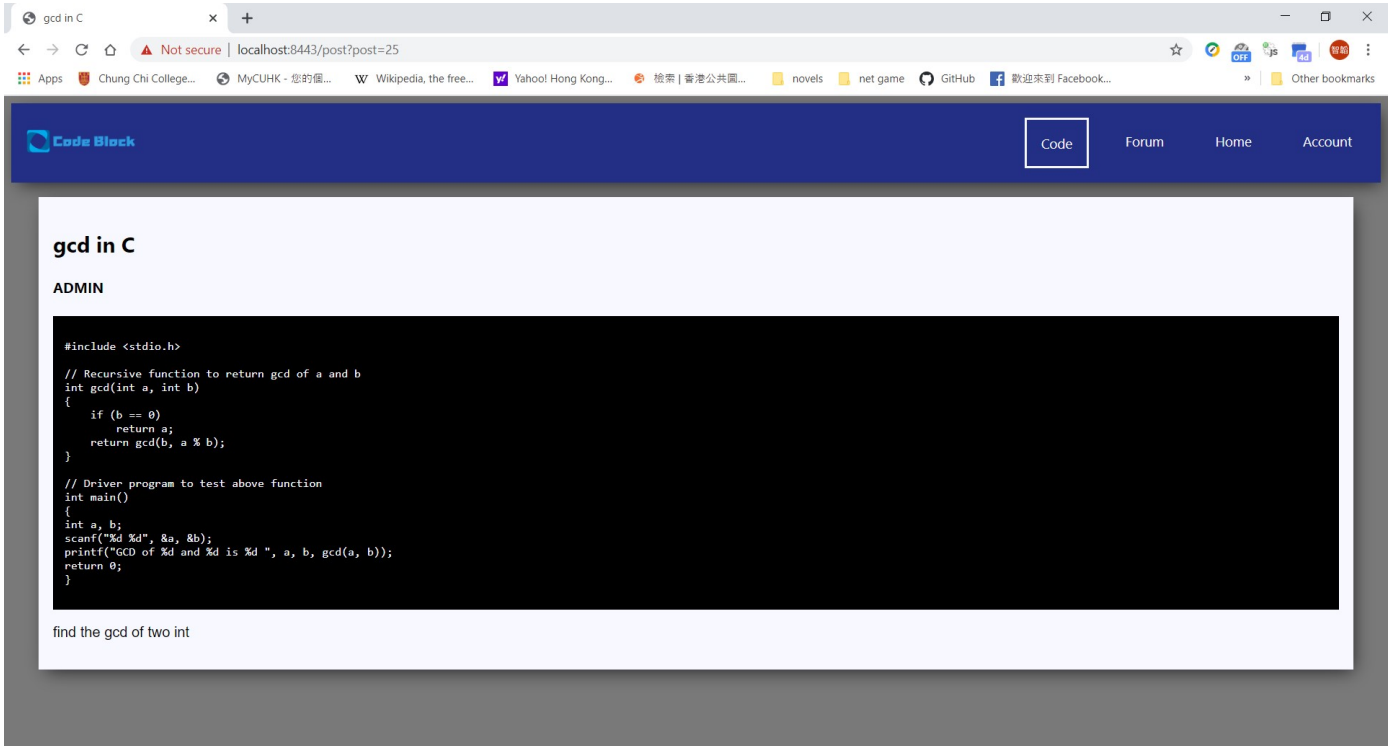
Case 5: (pass)



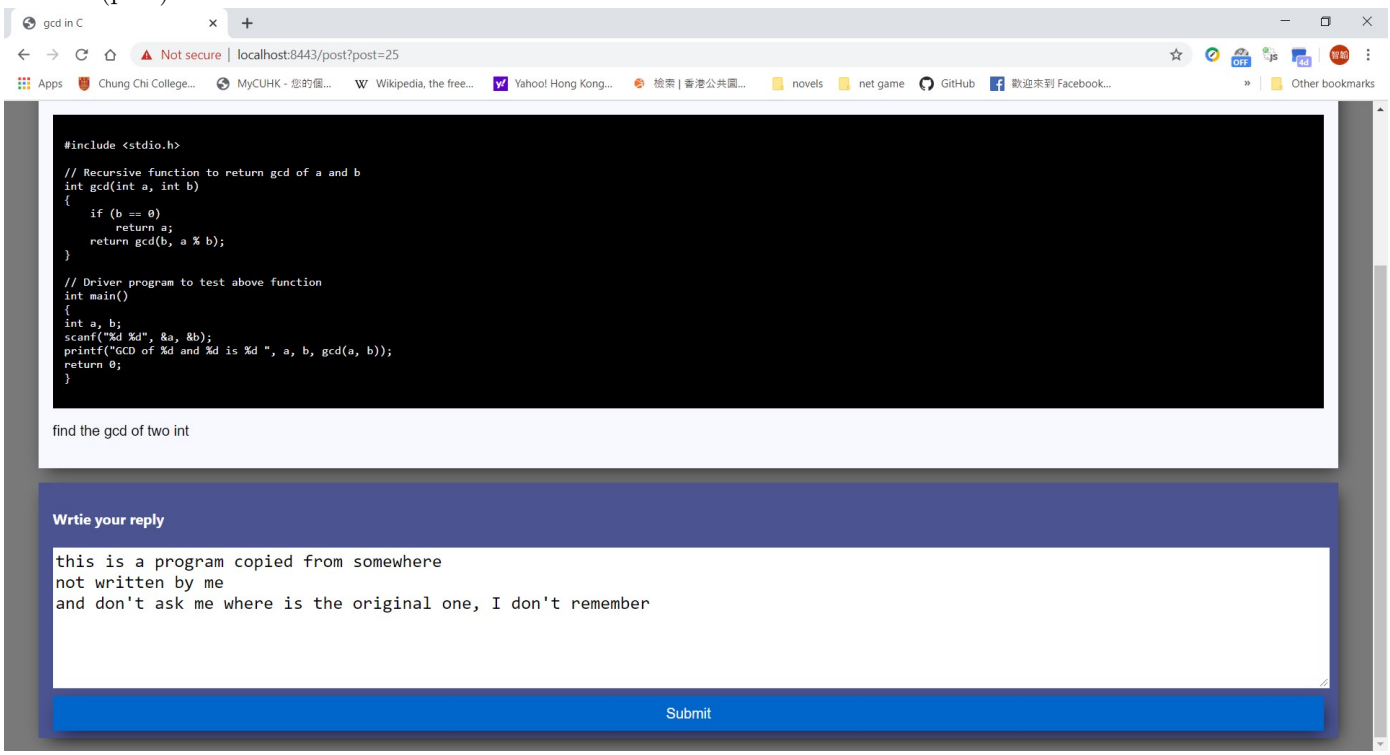
Reply

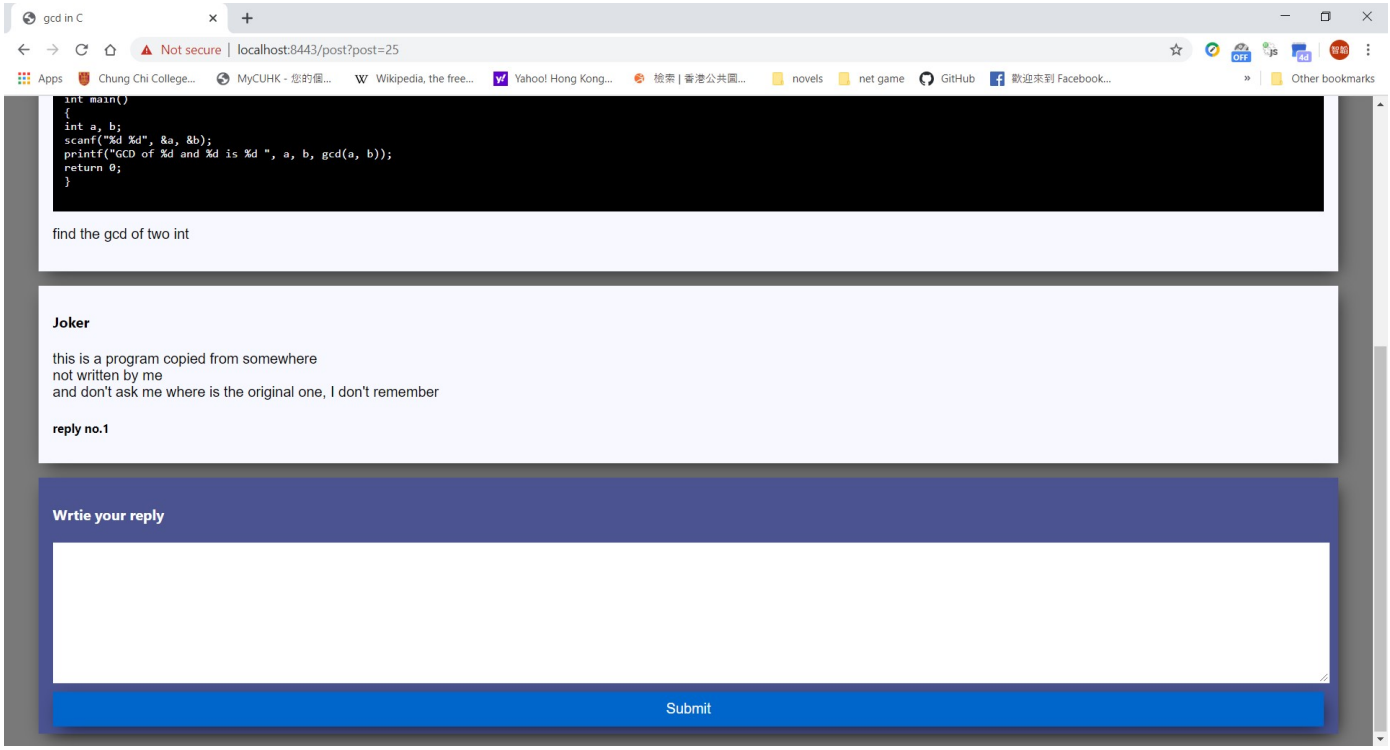
There should exist a textarea for authenticated users to write their reply, for unauthenticated users, there should not be any textarea for reply. And upon successfully added the reply, the system should refresh the page for the new reply added for the client.

Case 1: (pass)

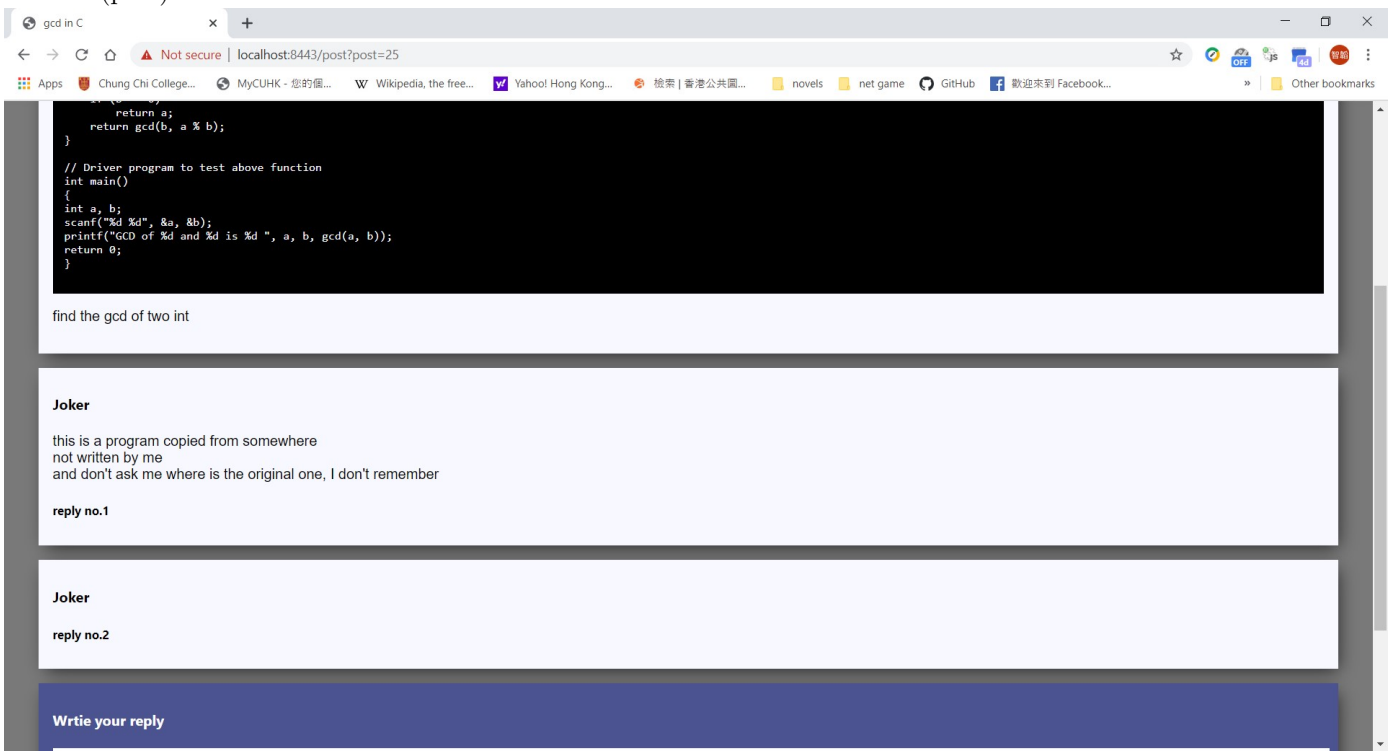

gcd in C
Not secure | localhost:8443/post?post=25
Code Block
Code
Forum
Home
Account
gcd in C
ADMIN
#include <stdio.h>
// Recursive function to return gcd of a and b
int gcd(int a, int b)
{
if (b == 0)
return a;
return gcd(b, a % b);
}
// Driver program to test above function
int main()
{
int a, b;
scanf("%d %d", &a, &b);
printf("GCD of %d and %d is %d ", a, b, gcd(a, b));
return 0;
}
find the gcd of two int

Case 2: (pass)


gcd in C
Not secure | localhost:8443/post?post=25
#include <stdio.h>
// Recursive function to return gcd of a and b
int gcd(int a, int b)
{
if (b == 0)
return a;
return gcd(b, a % b);
}
// Driver program to test above function
int main()
{
int a, b;
scanf("%d %d", &a, &b);
printf("GCD of %d and %d is %d ", a, b, gcd(a, b));
return 0;
}
find the gcd of two int
Wrtie your reply
this is a program copied from somewhere
not written by me
and don't ask me where is the original one, I don't remember
Submit



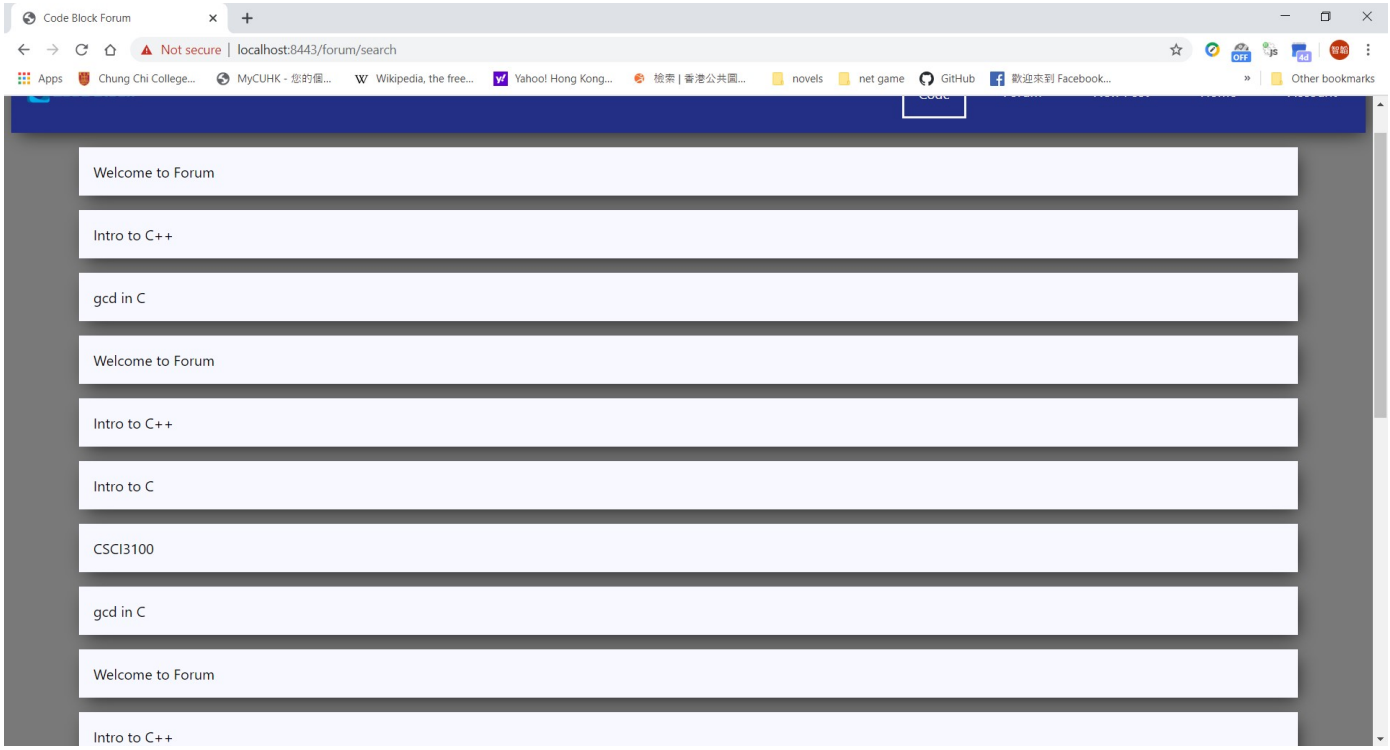
Case 3: (pass)



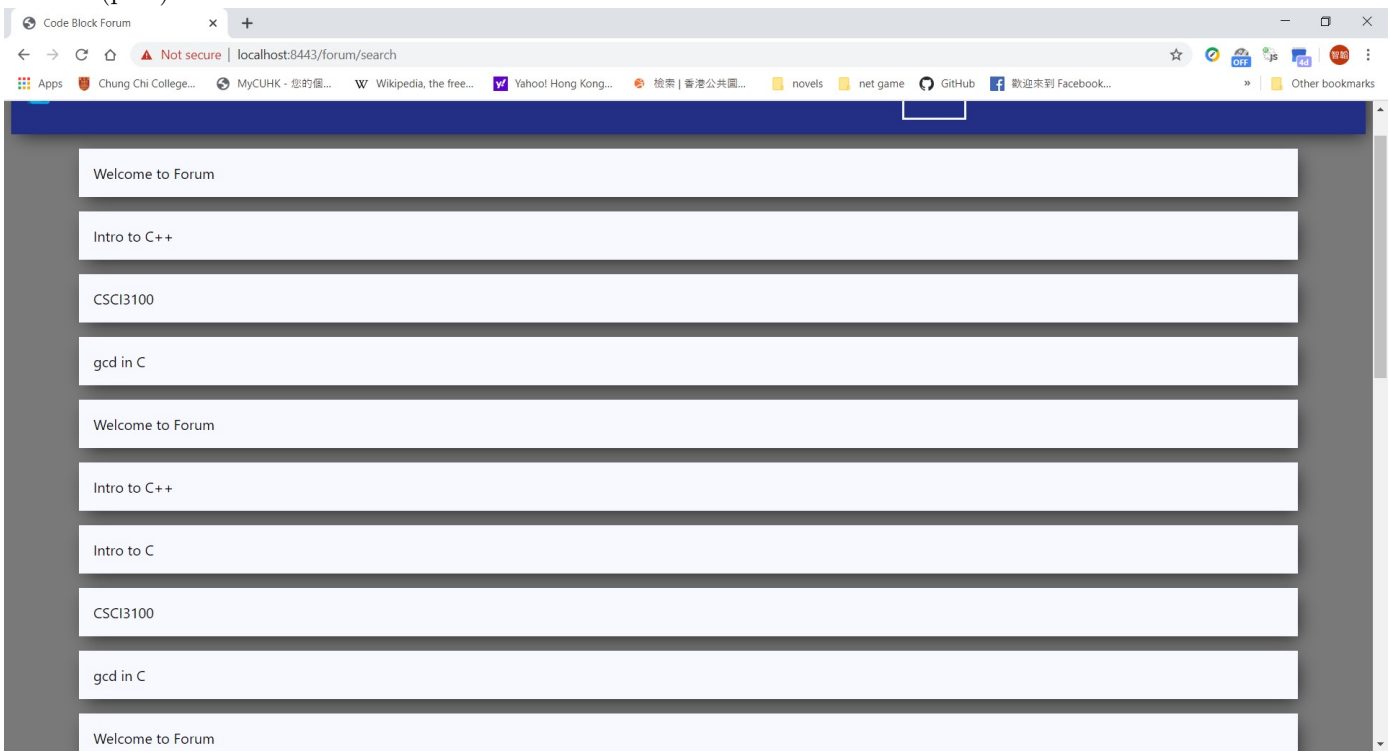
Search

The system should list all relatde post titles with links to the posts.

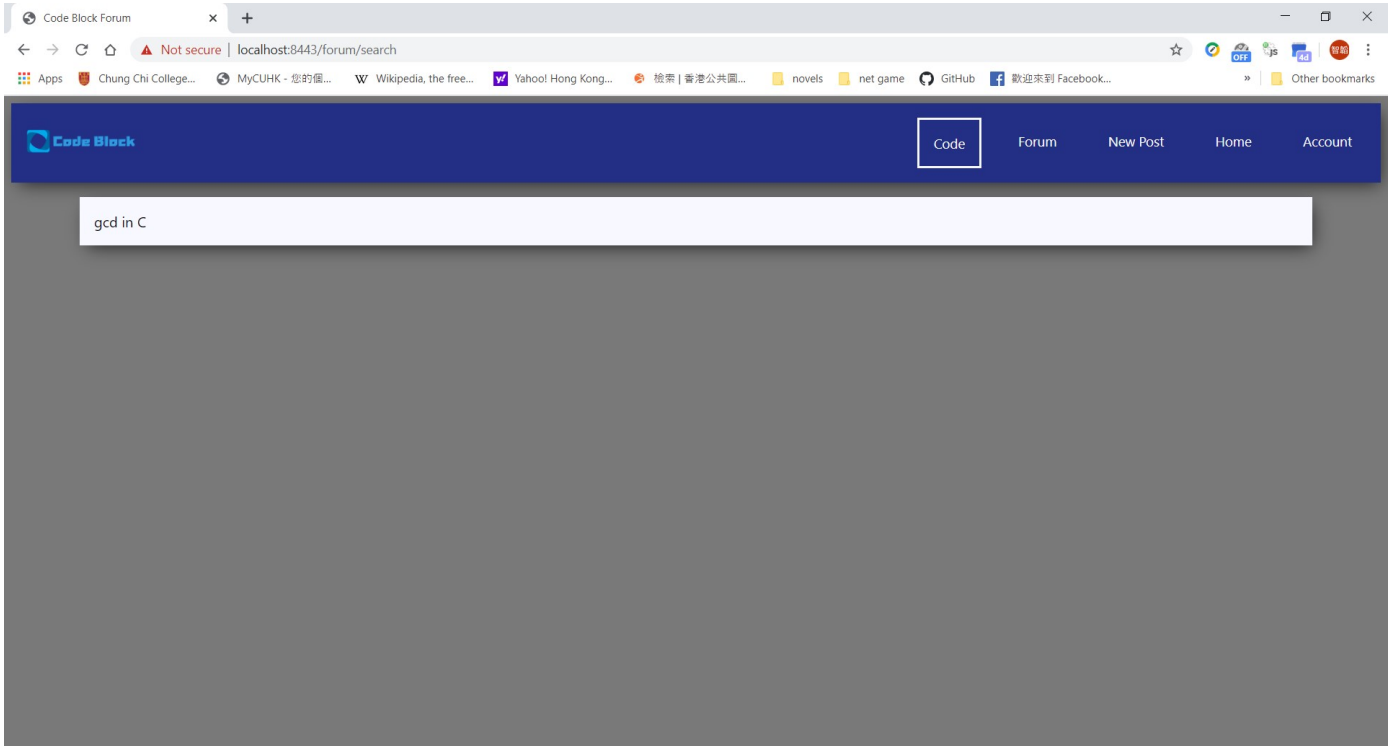
Case 1: (pass)



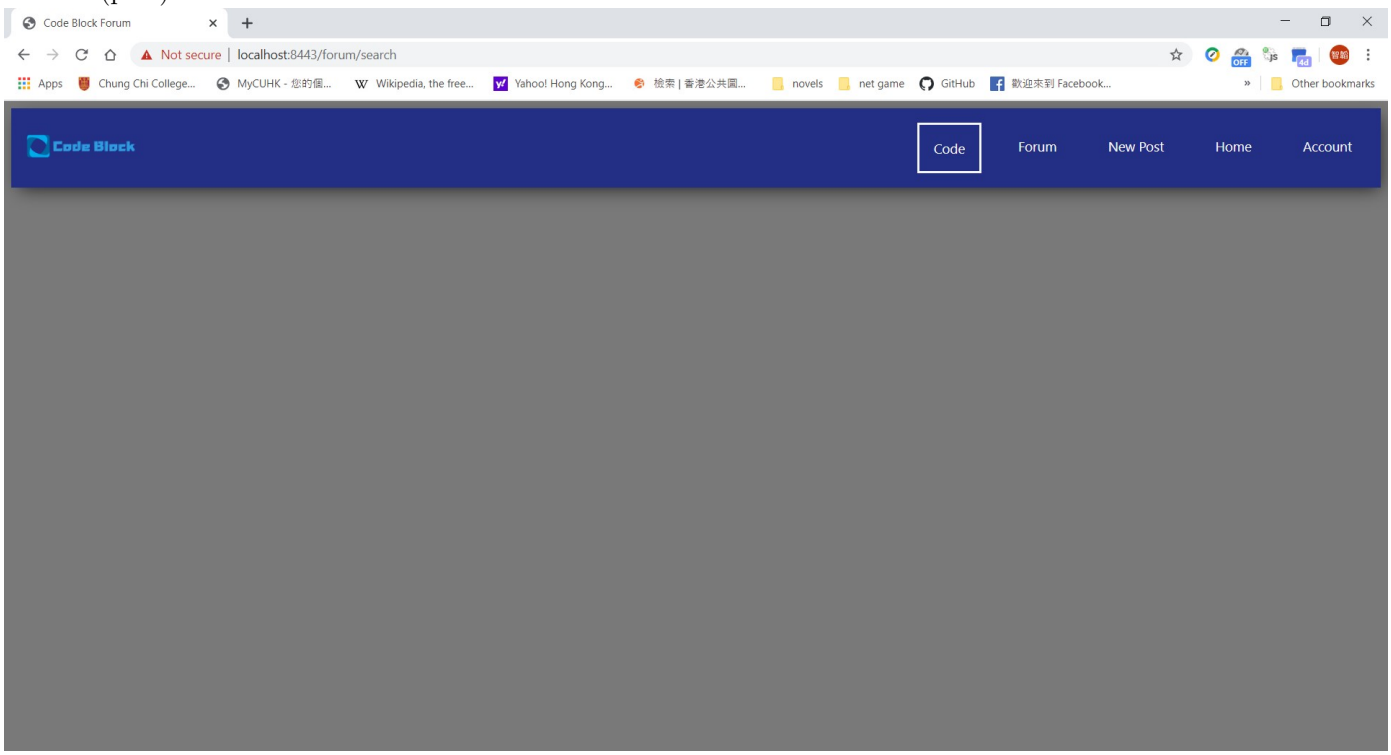
Case 2: (pass)



Case 3: (pass)


Code Block Forum
Not secure | localhost:8443/forum/search
Apps
Chung Chi College...
MyCUHK - 您的個...
Wikipedia, the free...
Yahoo! Hong Kong...
檢索 | 香港公共圖...
novels
net game
GitHub
歡迎來到 Facebook...
Other bookmarks
Code Block
Code
Forum
New Post
Home
Account
gcd in C

Case 4: (pass)


Code Block Forum
Not secure | localhost:8443/forum/search
Apps
Chung Chi College...
MyCUHK - 您的個...
Wikipedia, the free...
Yahoo! Hong Kong...
檢索 | 香港公共圖...
novels
net game
GitHub
歡迎來到 Facebook...
Other bookmarks
Code Block
Code
Forum
New Post
Home
Account

6

Lesson Learned

In this project, we have followed the waterfall software development model in the start, but we have gradually changed to agile and kanban model. In the coding stage, the back-end team have finished different modules and verified their correctness. Yet, the a part of the front-end team failed to finish their part according to schedule. This hindered the process, and we would not be able to proceed if we continued following the waterfall model as we should not proceed to integration stage if we have not finished the last module. In this project, we have found a major concern of waterfall model, is that waterfall model is not so flexible when only a part of the project is delayed. However, when a hybrid of kanban and agile model is adopted in later stage, the testing and integration advanced faster as the system allows more flexibility and enhancement without re-evaluating the requirements.

Another lesson learnt is that the deadline of schedule should be set much earlier than the actual deadline. Because it is likly that a part of your project delayed or even have no sign that it has been worked on by the person whom are supposed to do. By setting an earlier deadline in schedule, it is more likly that we can mend and finish the unworked part on time. Thus a higher software quality can be acheived.

If we can re-do the project, we would rewrite the whole schedule with earlier deadline of each phase of development. Moreover, instead of using waterfall model, kanban development model will be adopted from the start of the project. This is to handle the problem of inflexibility of waterfall model and the progress of each task is visualised and open. Therefore if we have adopted kanban development model, tasks will be handled in a better pace as we can know which tasks is being worked on.

Currently, the system is currently on version 1.0.1, yet there are many features designed was not implemented. For example, peer rating of posts, ordering of posts according to hit rate and tag system. Direct loading of code generated by visualised code editor, and loading of pre-written libraries for customized functions for rapid prototyping. Another feature planned but not implemented is code result comparison with expected output provided. This implies that there are still a lot of room for improvement as we have a lot of advance features not implemented yet. If we have more time, and a better planned schedule, we would continue on implementing the features and improve user experiences.

7

Conclusion

The Block Code Online IDE is designed to provide a dedicated online platform for programmers to exchange knowledge and source code. We believe only with exchange of ideas will allow a better result. Thus this dedicated platform is designed. The platform has a forum for programmers to share their code, and discuss about it. The platform also allows users to run codes on this platform, so that people can use other's codes more easily. A visualised code editor is also implemented in the system, so that programming beginners can learn coding using it more easily. The implementation and system architecture of the Block Code Online IDE followed the UML diagrams described above, thus with its OOP programming architecture and highly modularized design, it is highly adaptable to future changes. And the system is thoroughly tested, with protection against SQL injection. Therefore, our Block Code Online IDE is a user-friendly, reliable, secure, adaptative and functional system for programmers to use.

8

Reference

[1] dineshLL, AchalaDias, JudeNirosan, and maxeau, *Lisa*, Project Lisa is an online tool which teach Object Oriented concepts to the users. This intuitive tool also helps the users be familierize with C++ syntaxes while they learn OOP concepts., 2018.

[2] J. Harlin, J. Overson, C. Davis, and D. Linse, *Es6-plato 1.2.3*, Visualize JavaScript source complexity with plato. Based on the older es5 plato, this is a port to es6 and eslint, 2019.