

Report

Report

Introduction

System Arcgitectual Design by DFD

Detailed Description of Components by UML

Component-1: Registration & Login

Component-2: Search Question

Component-3: Reply

Component-4: Online Compiler

User Interface Design

Description of the User Interface

Object and Actions

Test

Lessons Learned

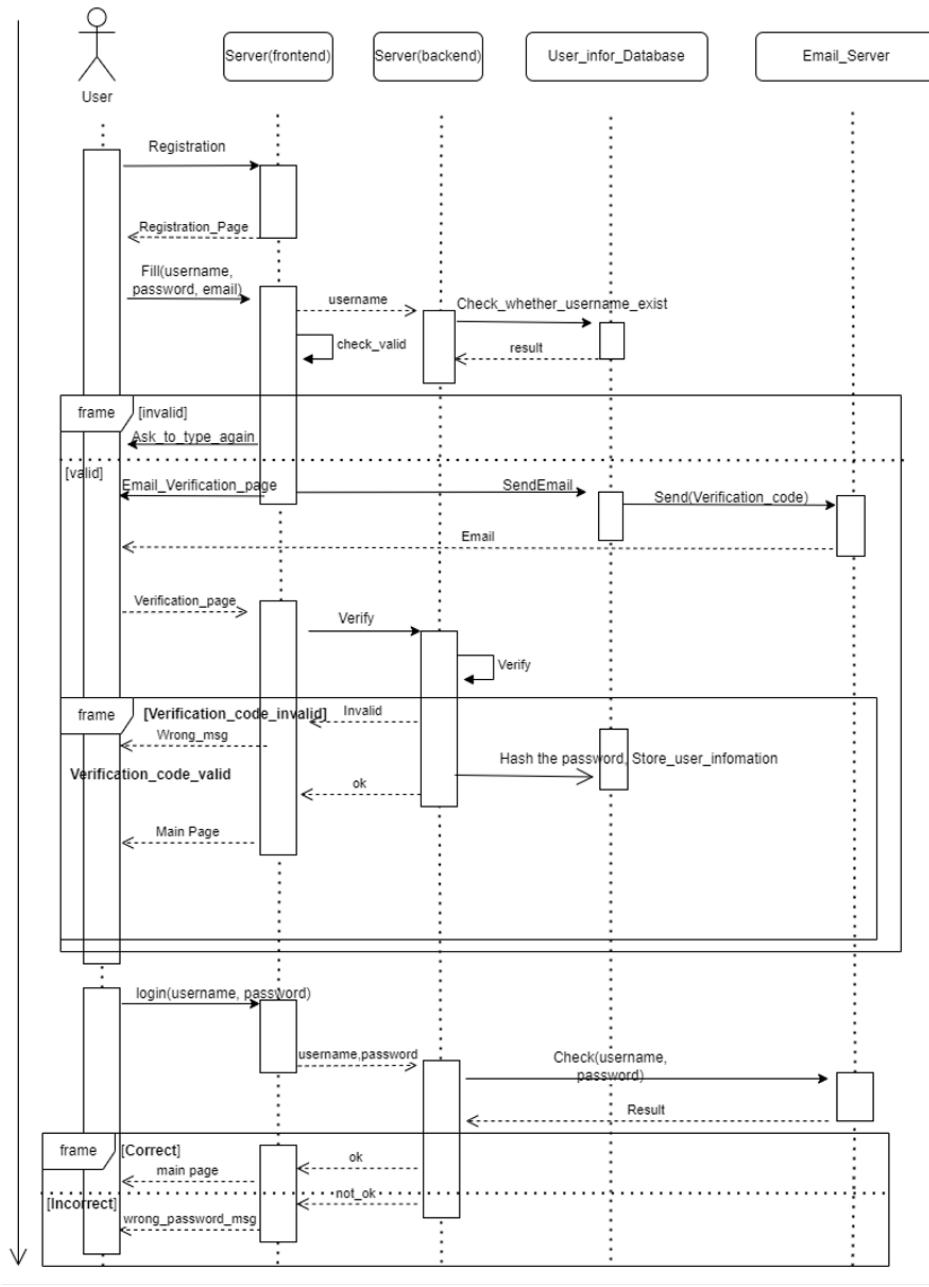
Conclusion

Introduction

System Arcgitectual Design by DFD

Detailed Description of Components by UML

Component-1: Registration & Login

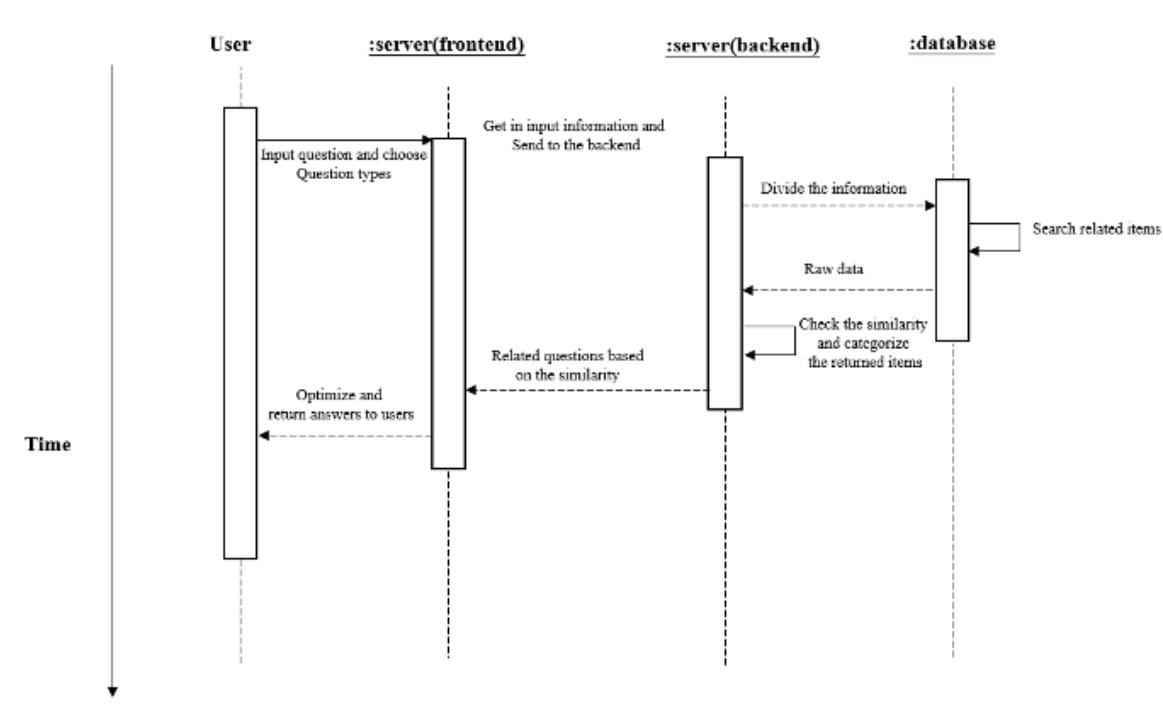


If it is the first time the user visits our website, then registration is essentially the first step. To make a new count, users must input the username, password, and email address. Based on the information, the backend will check whether the username has been taken or not. If the username is invalid, then the information will be returned to the user in order to help them choose a new valid username. If the username is valid, then an email with a randomized verification code will be sent to the input email address. The user has to type in the exact verification code to finish the verification process. If the code is invalid, the corresponding message will be sent back to the user. If the code is correct, which means the registration is legal, then all the formation about the new user will be formally recorded in the database. The most sensitive information, the password, will be hashed and encoded in the database to protect the users' privacy and keep the website safe. After storing the information, a new account is created. And the user will be redirected to his main page directly without another log-in process since the registration will promise the validation and safety of the user.

If the user already has a valid account, then the registration process can be ignored, but the log-in process is now required. The user has to type in a unique username and password to get into the website. Then the information will be sent to the backend of the server to validate. If neither the username nor the password is valid, the log-in request will be surely rejected, and faint hints will be sent back to the user. Specifically, the hints information goes like "Username or Password Fail", which will not provide detailed information about the error so that the privacy of the

user is protected. If the information is valid, then the user will get permission to the main page.

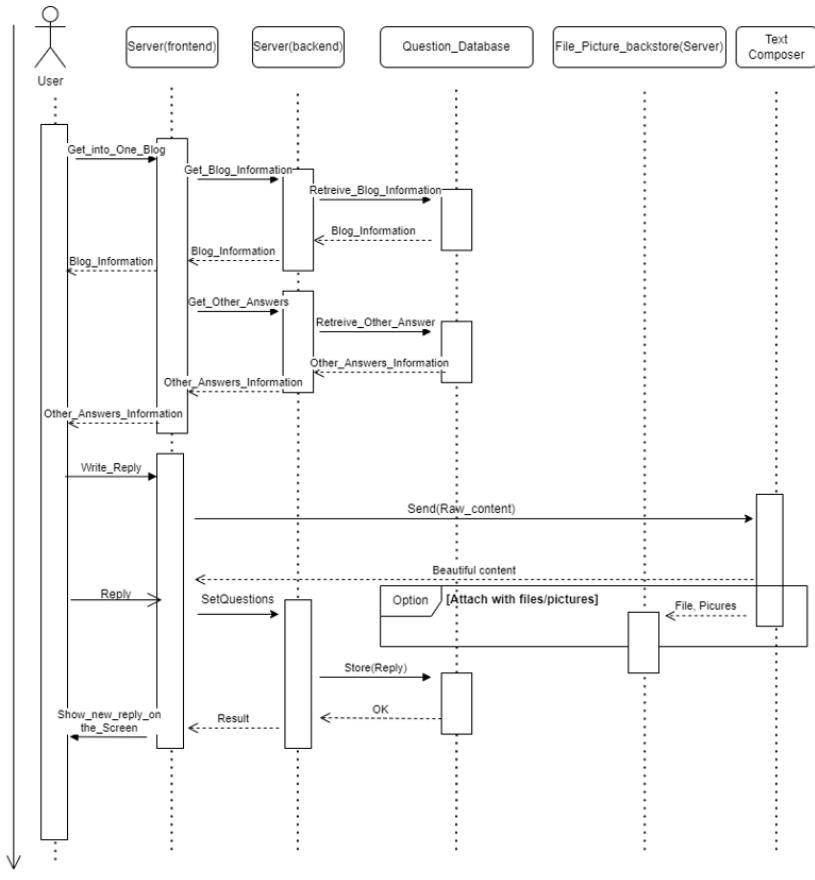
Component-2: Search Question



The search function is implemented by keyword match rather than string match. Users can search blocks related to a specific topic on the main page. Generally, the frontend will check the input and send it to the backend. The backend will split the content into different vital words based on the typed content. Then based on the derived keywords, the backend will go through the entire database and search for the blocks with the maximum similarity. Then, then related answer blocks will be returned in the order of similarity for the users to choose. For the keyword match, it is used to expand the searching scope so that the users can get more related blocks that they might be interested in.

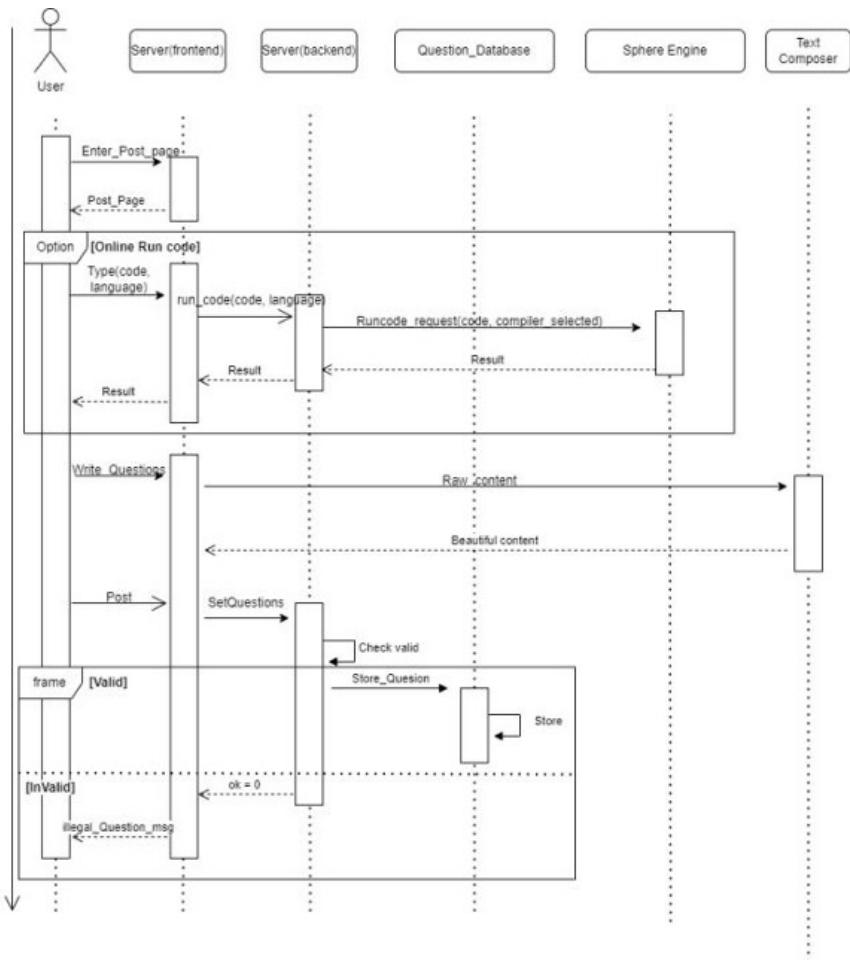
Getting lots of returned blocks is always a good idea. Sometimes, people are more interested in finding what they want in a faster way. Therefore, our search function is implemented together with the scope selection. That is, users can select the topic of the questions to narrow down the search scope. In this way, the blocks returned will be more related and concentrated.

Component-3: Reply



To reply to a block, users have to get into a block in the first place. After users get into the block, not only the block information will be shown, other answers will be returned if existing as well. All the returned information is provided for the users to better understand the blocks. If users want to offer more information and reply to the block, they should type in their answers, and the answers will be sent as raw content to the text composer. Then the text composer will process the raw content, returning and showing the content in a neat and nice way. Besides, users can also upload the files and pictures together with the answers. The uploaded answers will be stored in a unique area on the server. After the edit of the reply, the reply can be formally uploaded. Answers and their relation to different questions will also be stored in the server's database. At last, the reply will be shown to every visitor to the block.

Component-4: Online Compiler



The online compiler is on the posting page. Users can enter the posting page at first. If they want to test some codes, they should choose the programming language first. Then, they are free to program on the posting page. After they click the run button, the request will be sent to the sphere engine, and different compilers will be chosen based on which kind of program language the users are using. After that, the code will be compiled in the backend, and the result will be sent back to users.

The process of raising a question is similar to the replying procedure. In the text chat, users can type in the essential information about what they want to post, including the title, content, group type, and sub-group type. Then the information will be optimized by the text composer. Next, the backend will check the validation of the question. If there is nothing wrong, the question will be stored and posted. If not, corresponding hints will be returned to the users for them to correct.

User Interface Design

Description of the User Interface

The interface of our project is inspired by the public forum in China: Zhihu, especially the home page. It mainly consists of the search engine on the top, and different navigation bars below. Each navigation bar represents a unique order and function for all of the blogs in our database. For example, the hottest blog navigation bar represents the most popular blogs among all users, and the non-solved navigation bar represents the blogs that are not yet answered, whose layout is similar to Zhihu.

One of the characteristics of our user interface is **animation**. Our interface integrates many animations in order to make the interface more vivid and interesting to users. For example, you would notice that there is a dynamic crystal ball waving on the login page, as well as the animation text in the login page. Also, after you finish the register page, there is also another animation between two different webpages, it is not simply jump to another webpage directly in a rigid way. To make users more pleased when they use our forum, we also designed a loading

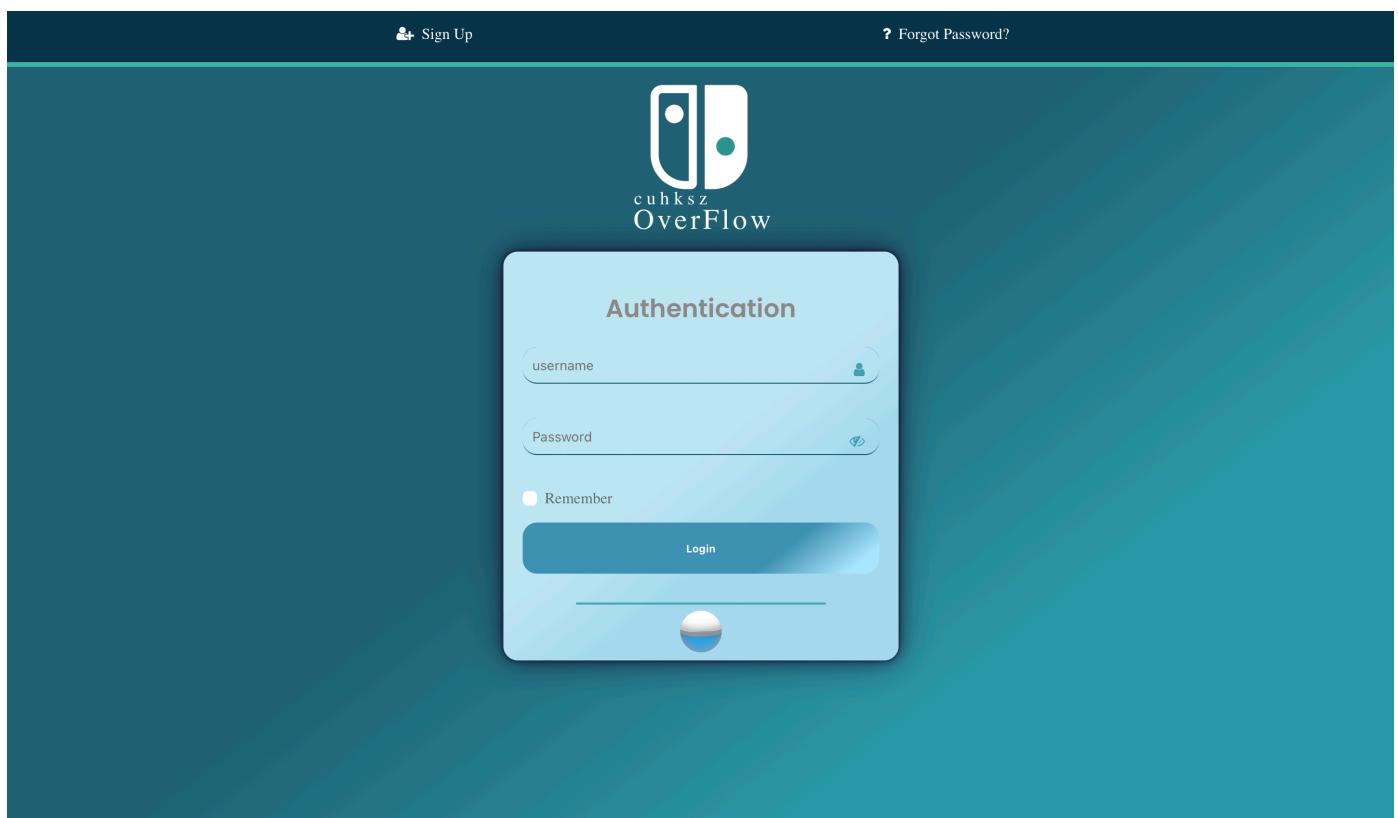
animation between home page and post blog page. In this way, user would not watch the webpage stuck in a while, or some hacker may use this time to click or type something on interface to make the webpage crash, instead, we design this loading page to protect our program but also could provide an animation for user to see to improve their experience feeling.

Another highlighted feature for our user interface is **button design**. We took amount of time to design our buttons. Since there are a lot of buttons, typically 15-20 buttons in our program, and there are all different to each other. Each button is well-designed for user. For example, to alter user to log out, we make the button highlighted when user hover this button in case user will log out by mistake. To warn user to reset username and password, we make the button in a red and a warning way. In those details, we have considered them to make sure they are user friendly as much as possible.

Overall, our UI have is **detailed, well-designed and friendly to user**. The following part will demonstrate each function and actions.

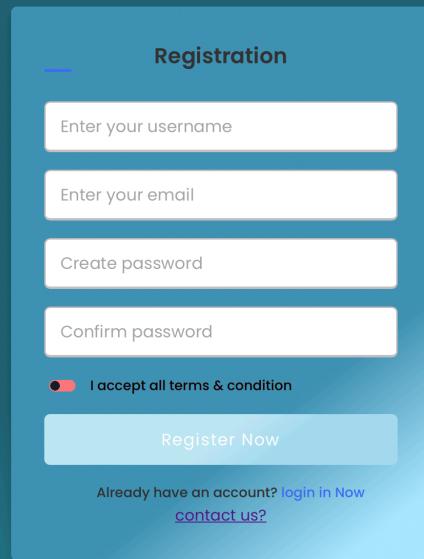
Object and Actions

- **Login**



The login page is designed for user directly use their user name and password to get into home page, user must input the correct username and password in our database in order to login. User could use click the eye icon to hide or show the password. They can also click remember to remember their password. The sign up and forget password button are on the top, user could click those icons to jump to another webpage. The animation text will be shown everytime one refreshes the page.

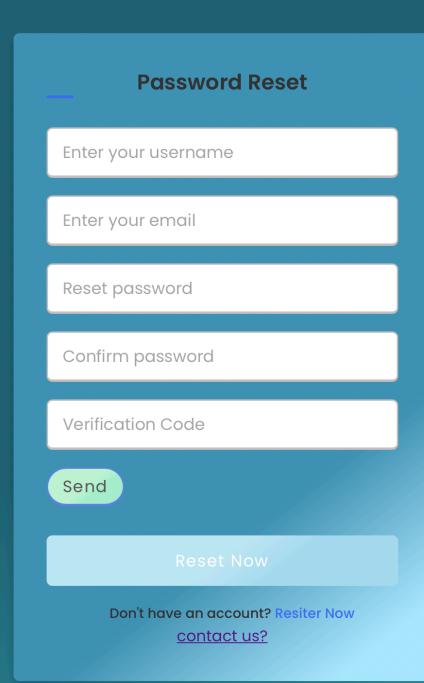
- **Register**



The registration form is titled "Registration". It contains four input fields: "Enter your username", "Enter your email", "Create password", and "Confirm password". Below these fields is a checkbox labeled "I accept all terms & condition". A large blue button at the bottom right is labeled "Register Now". At the very bottom, there is a link for users who already have an account: "Already have an account? [login in Now](#) [contact us?](#)".

In the registration part, Users are required to input their unique username and their email, they should type their password and confirm it. They should also allow all conditions to follow the rules and be responsible for their words in our forum. Notice that the username is unique, if user tries to type the username that has been used by others, which is not allowed here. User could also click the login button to jump back to the login part.

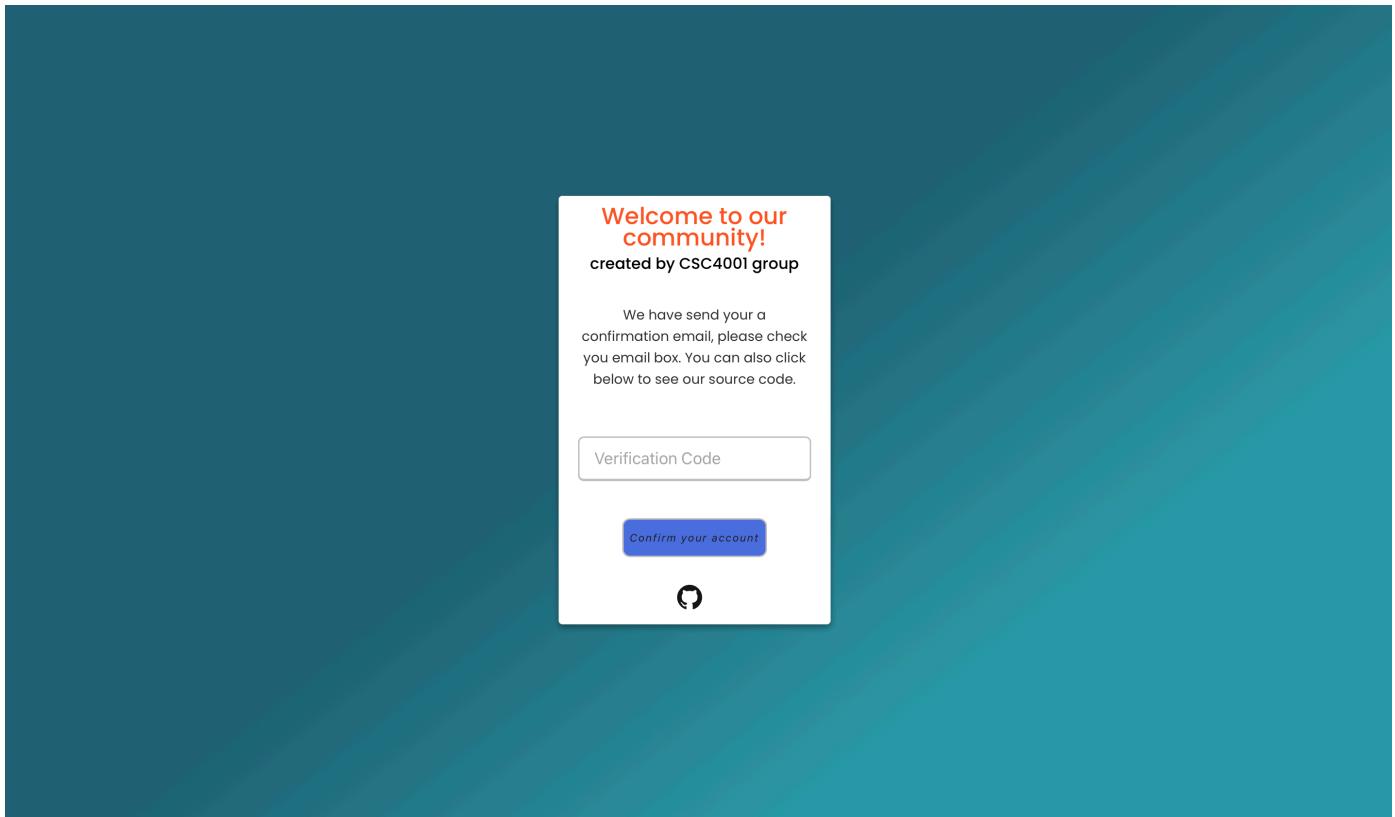
- **Forget Password**



The password reset form is titled "Password Reset". It contains five input fields: "Enter your username", "Enter your email", "Reset password", "Confirm password", and "Verification Code". Below the "Reset password" field is a green button labeled "Send". A large blue button at the bottom right is labeled "Reset Now". At the very bottom, there is a link for users who don't have an account: "Don't have an account? [Resiter Now](#) [contact us?](#)".

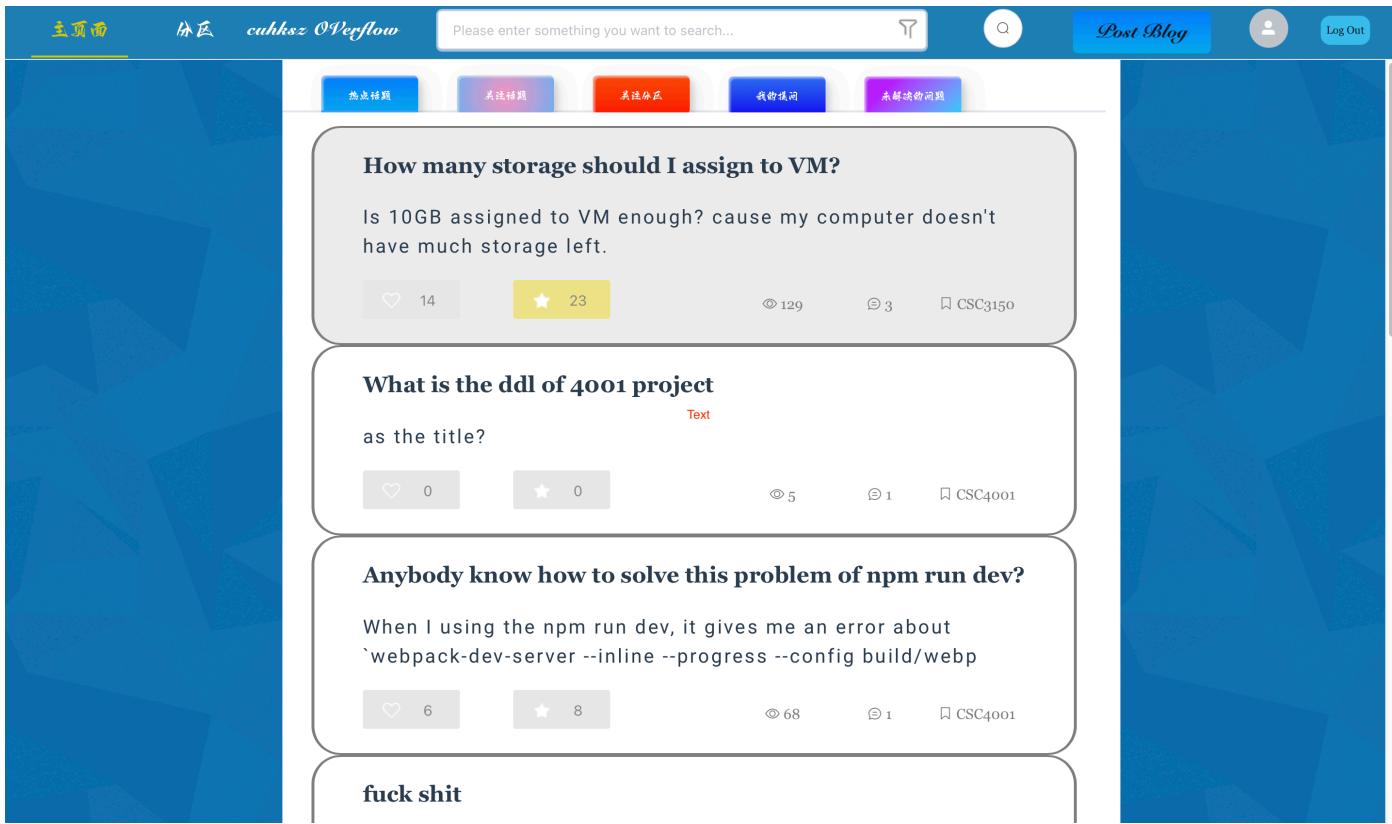
This part is used for user to find back their password and reset it. User are required to type their username, email and their new password. Notice that if the username is not correct, this reset will failed. We will send you a verification email, only and if only the verification email is correct, the password will be reset and user will directly jump to home page.

- **Verification**



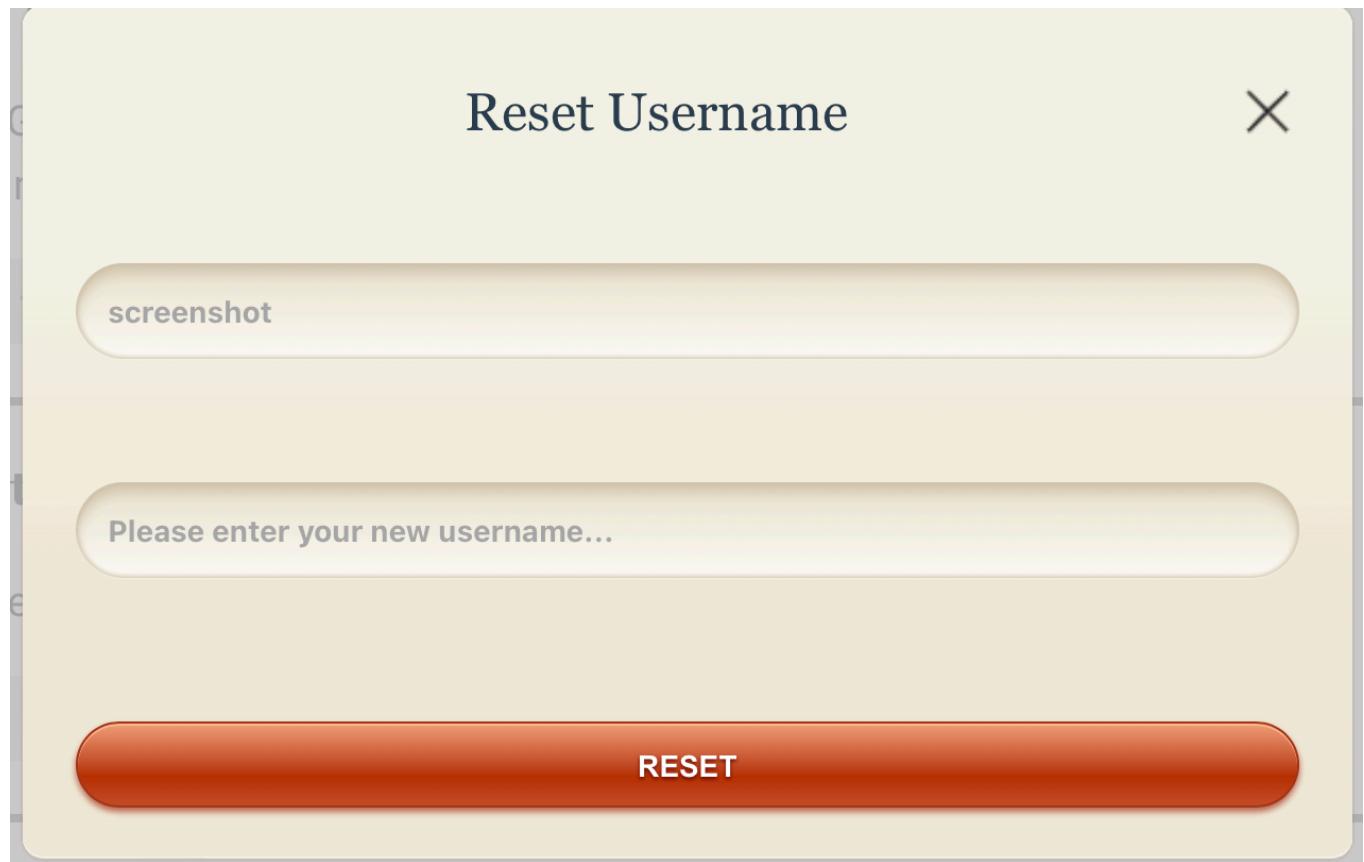
After user finish the registration part, they will receive an email from us. We have an animation in this page in order to let user feel comfortable when they are waiting since the Verification email may take a while to send. During this time, User could click the GitHub icon to see and run our source code. When the email verification code is correct, the user are allowed to go into the home page.

- **Home Page**



As for the home page, The layout of home page consist three parts. First is the menu on the top, user could click the home page icon and partition icon to go into different page, they could also click the search bar to type some content to search. And the button on the right is the post button, user could click the button to jump to the post blog part and click log out button to log out. they could click the me button to upload their unique profile, we will store their profile on our server. The second part is 5 catlogs, each catlogs contains different function and order of blogs. In hottest blogs, it mainly contain the blog that is hottest and most popular on our forum. The followed blogs represent the blogs that user click followed button, and the followed partition is the partition that user clicked follow. My blogs will show users the blogs that they have post on our server. Unsolved blogs is represented the blogs have no answer, it is designed to encourage user to answer and reply the blogs on server. The Third part is blogs part, blogs are ordered by some function in catlogs, When user hover each blog, the color will be gray. The tiltitle of blogs is bold and the content is normal. each blogs will showing user the view times, the answer amount and its partition type. User also could click the like or follwed button, the color will also changed after user click it.

- **Reset**



When User click the Reset Username or Reset password, they could reset their user name or password in the home page. When user reset their password, they are required to log in again.

- **Partition**

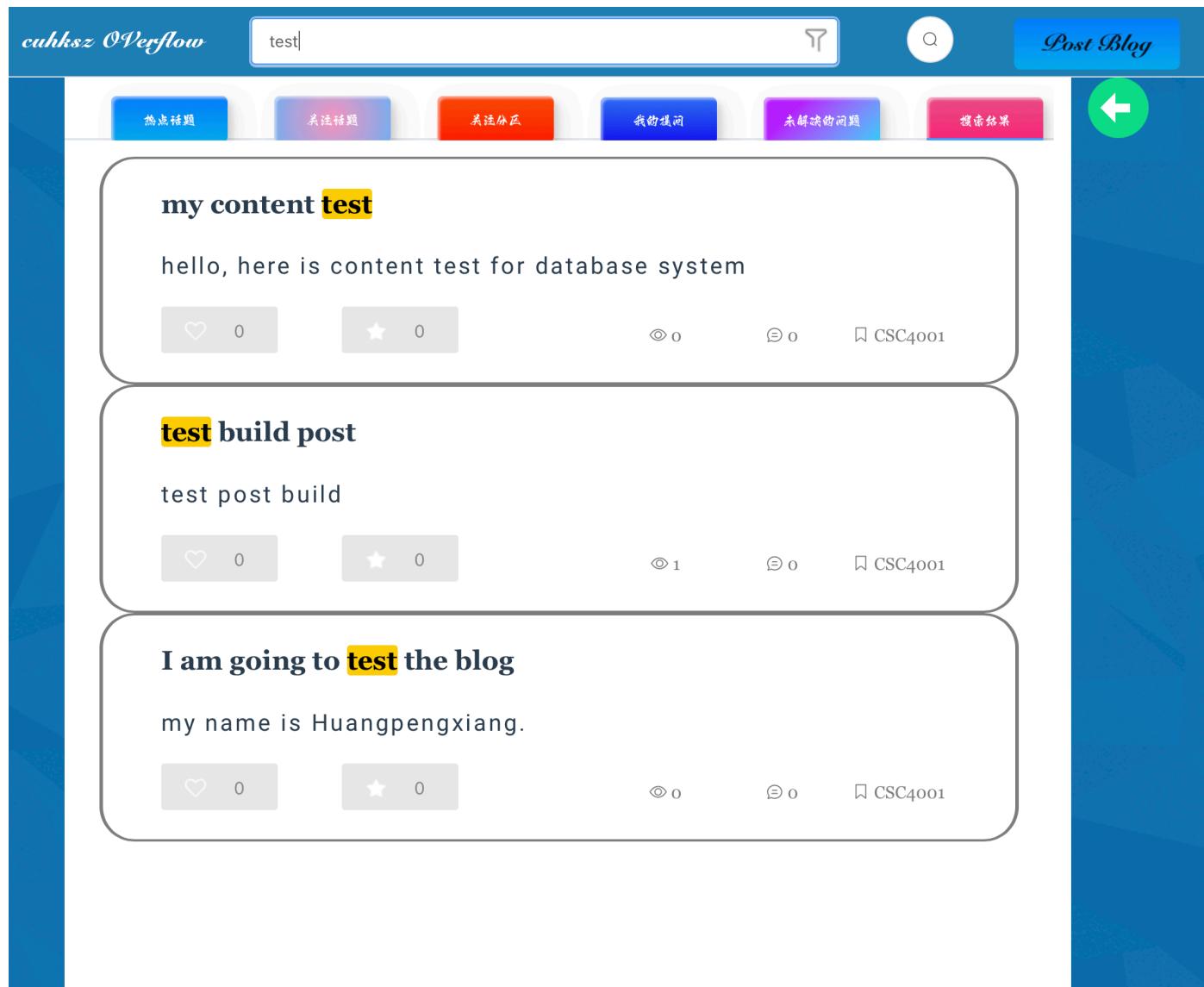
The screenshot displays a web-based course management system with a blue-themed interface. At the top, there is a navigation bar with links for "主页面", "分区", "cahksz Overflow", a search bar containing "Please enter something you want to search...", and buttons for "Post Blog", "Log Out", and a user profile icon.

The main content area lists four courses, each with a thumbnail image, the course name, and a "Sub Partitions" section:

- CSC1001 - Introduction to Computer Science**
Thumbnail: Python logo
Sub Partitions: Homework 1, Homework2, Homework3, Homework4, Midterm, Final
Star rating: 494
- CSC1002 - Computation Labotary**
Thumbnail: Snake game logo
Sub Partitions: Homework1, Homework2, Homework3, Homework4
Star rating: 400
- CSC3002 - Introduction to Computer Science: Programming Paradigm**
Thumbnail: C logo
Sub Partitions: Homework1, Homework2, Homework3, Homework4, Final
Star rating: 246
- CSC3100 - Data Structures**
Thumbnail: Data Structures logo
Sub Partitions: (partially visible)

When user click the partition button, they will jump to this part. This part is used for user to find the corresponding blogs in specific partitions. Each partition contains different sub partitions. User are allowed to click the button to see all the blogs in this subpartition. So they could find what they want in a very short time. And User are allowed to click the followed button so that they could see their followed partitions in home page, which is convienent for user to find corresponding blogs.

- **Search**



User could type the key word in the search bar to search the relavent blogs. They could type any thing they want, and we will return the most relavent blogs in the home page in order. The keyword will be highlighted so that user could find the answer very quickly. And user are also allowed to use the filters to narrow down search scope. They could click the button to choose the search scope, the search scope is based on the partition and subpartition. When User choose the filter, the search scope wont be all blogs, it will return all the content in this scope.

- **Blog**

yyy



How many storage should I assign to VM?

Is 10GB assigned to VM enough? cause my computer doesn't have much storage left.

Below is the related images:

The image shows two screenshots. On the left is the Microsoft Store page for 'Ubuntu 20.04 LTS' by Canonical Group Limited. It features the Ubuntu logo, the text 'Ubuntu 20.04.4 LTS', and a 'Get' button. Below the button are ratings: '4.1 ★' and '10 平均 评论数'. Underneath are '开发人员工具' and a download link. On the right is a screenshot of a Windows taskbar. The taskbar includes icons for File Explorer, Task View, Start, and a search bar. The 'Ubuntu' icon is visible in the taskbar, indicating it is running in the background. A context menu is open over the Ubuntu icon, showing options like 'Open', 'Run as administrator', 'Pin to taskbar', 'Unpin from taskbar', 'Switch to', 'New task (background)', and 'Close task'.

When User goes into the blogs page. They could see the blog's partition and its sub partition, and they could click the like or followed button to support this blogs. The title is in a bold form and content is in a normal form. The file will be shown as a link in the blog and the picture will be shown directly. User could click the file link to download the flie if blog contains any. They are also allowed to enlarge the picture or download the picture. When they hover the user icon, the username who created this blog will be shown. All the text form in post part will be the same here.

- [Reply](#)

The image displays three vertically stacked blog comment cards, each with a user profile picture, a timestamp, and a text area. The first card is from 'QinPR' replying to 'yyy' as the author, with 5 likes. The second card is from 'hands' replying to 'QinPR', with 0 likes. The third card is from 'Paul' replying to 'hands', with 0 likes. All cards mention 'Powered by Froala Editor' at the bottom.

```
graph TD; QinPR[QinPR] -- reply --> yyy[yyy]; QinPR -- author --> QinPR; QinPR -- 5 --> QinPR; QinPR --- text1[亲测10GB也是没问题的]; hands[hands] -- reply --> QinPR; hands -- 0 --> hands; hands --- text2[+1]; Paul[Paul] -- reply --> hands; Paul -- 0 --> Paul; Paul --- text3[a];
```

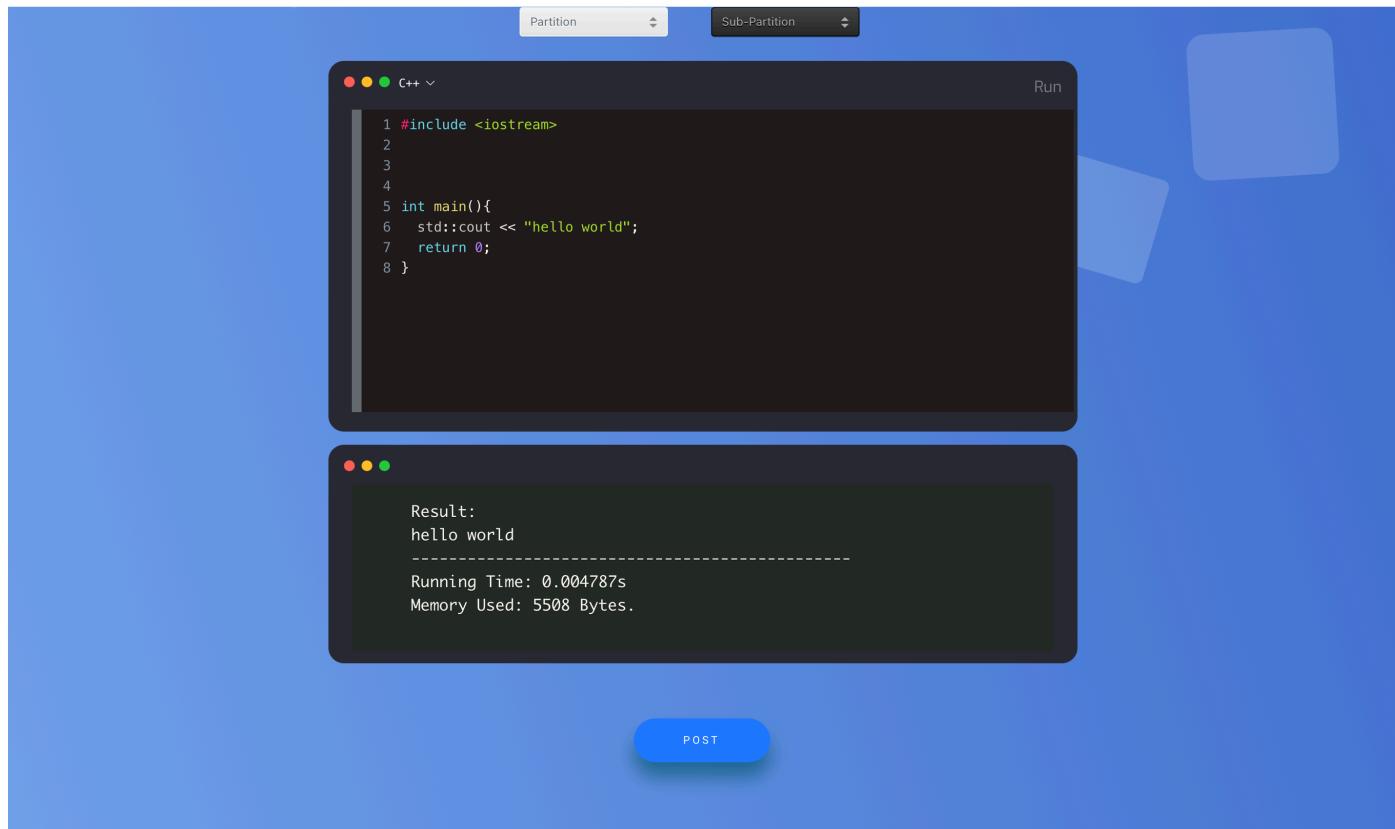
User could click the Answer button to answer each blog. Notice that the indentation will be different when user answer in a different situation. When they answer author, they will be classified into first level, otherwise they are second level. The first and second level has different identations so user could identify the blog answer quickly. User are also allowed to cilck like button to supprt this answer, The answer blog will be ordered in time.

- **Post**

The image shows a screenshot of a web application titled 'Post Your Blog Here ...'. It features a blue header bar with the title and a 'GO BACK' button. Below the header is a search bar with the placeholder 'Hello, this is blog title'. The main area contains a Froala editor window with a toolbar at the top and a contenteditable area below it. The contenteditable area has a placeholder 'feel free to input any content here'. At the bottom of the editor window, it says 'Powered by Froala' and 'Characters : 35'. Below the editor are two dropdown menus: 'Partition' and 'Sub-Partition'. The entire interface is set against a blue background.

When user click post button, they could create their own blogs. They are required to type the blog title and content, partition and their sub partitions in order to classify. The content input will be finished in a rich editor, the rich editor support many forms input and file uploading. User could upload their file or just insert a file link. Our sever will receive this file and store it into server. User could use the rich editor to create the blog forms they prefer. And all the blogs in home page will be decode as the same as they post in this webpage.

- **Compile**



In post part, User are also allowed to write their code as a supplement material. they could also run the code to see the output. Typically, User are allowed to write the code without environmental configuration, they could also write code in an ipad or a phone. The online compiler support many language include C, C++, Python, Rust... The running time and memory used will be shown on webpage just as an open Jude system. The code will be highlighted due to different language, different language has different highlighted method, and they will also be used here to improve user coding feeling.

Test

Lessons Learned

Conclusion