

Kevin Zheng, Shun Zhang

CS-GY 6083

Spring 2022

5/15/22

## Final Project Report

---

### 1. Introduction

This report will document the design of a web-based application for a question-answering service. Users will be able to register and login to the service through a web app. After registration is complete, users can begin to ask questions they desire answers to under a variety of topics, while also being able to answer questions that other users ask. Questions posed by users will be labeled under a two-tier hierarchy of topics. Users are able to search through the aggregation of questions to find the information that they want. Questions will also be given a “best answer” based on the number of likes that it garners.

The first part of this report will outline the basic low-level view of the relationships that exist between objects in our system, how this view can then be translated into a schema usable to model the system, which will then be implemented via a database management system. Then we will go into how this model is translated into a web application with explanations for the design and feel of the system.

## 2. ER Diagram

---

To start, we need a low-level view of the entities (people, objects, or concepts) that exist within our system and the relationships between them. Below is the entity-relationship diagram of the service.

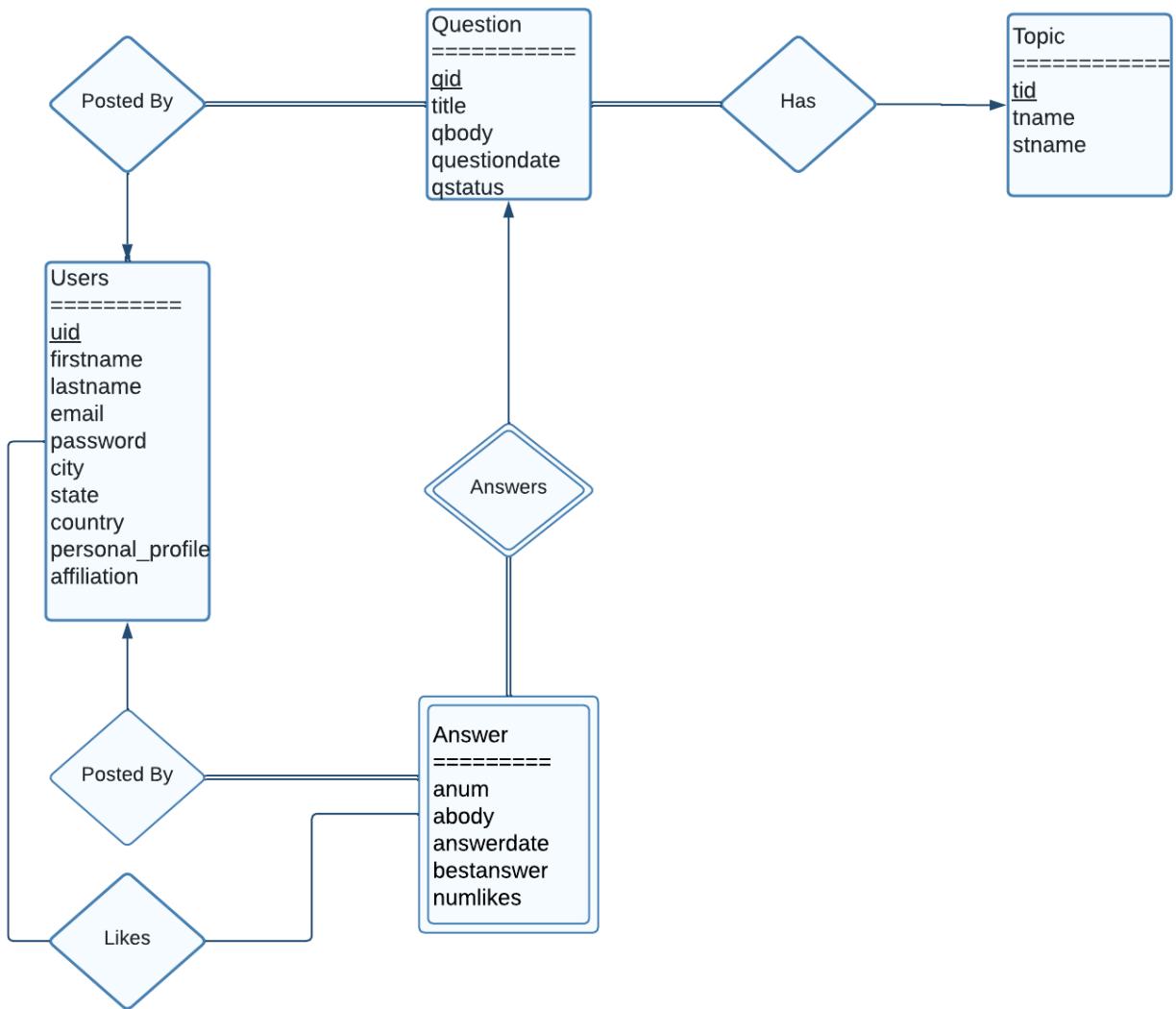
There are four essential entities that exist in this system: the user, questions, answers, and categories of topics that a question falls under. Users are identified by their user identification number, and attributes contain their name, email, password location, profile description, and affiliation. Questions can be identified by their identification number, and its attributes contain the title, body, date posted, and status. Answer cannot be identified by its own attributes, therefore it is a weak entity. It is dependent on Question as an answer cannot exist without a question it is responding to. Finally topics can be identified by an identification number and it has a main overarching topic as well as a subtopic attribute.

Relationships between these entities are denoted by a diamond, and the cardinality of the relationships are also detailed.

These cardinalities are defined under the presumption that:

- Users can ask an indefinite number of questions
- Questions can have an indefinite amount of answers
- Questions must have a topic
- Questions and answers may only be posted by one user
- Topics may be used across multiple questions

Entity-Relationship Diagram:



\*anum in Answer is partial key

Changes From Part 1:

- removed level from users

### 3. Relational Schema

---

Next, we have to translate the ER diagram into a relational schema. To translate the ER diagram into relational schema, we first translate all the entities into tables directly (Users, Questions, Answer, and Topic). Then we looked at all the cardinalities of relationships between entities. For all one-to-many relationships we insert the primary key from the one-side to the many-side. For example, since User->Question is one-to-many, we keep uid as a foreign key in Question. For the relationships that are many-many, we create a separate table for the relationship with the primary key of both tables. For example, users liking answers becomes its own table. Below is the result of converting the ER diagram into the relational schema.

Users(uid, firstname, lastname, username, email, password, city, state, country, personal\_profile, affiliation)

Questions(qid, asker, tid, title, qbody, questiondate, qstatus)

Answer(anum, answerer, uid, answerdate, abody, bestanswer, numlikes)

Likes(uid, qid, anum)

Topic(tid, tname, stname)

Foreign Keys:

- Likes(uid) is a foreign key to Users(uid)
- Likes(anum, qid) is a foreign key to Answer(anum, qid)
- Questions(assembler) is a foreign key to User(uid)
- Questions(tid) is a foreign key to Topic(tid)
- Answer(qid) is a foreign key to Question(qid)
- Answer(answerer) is a foreign key to User(uid)

Changes From Part 1:

- changed names of uid in Questions/Answers
- changed typing of uid/qid/tid in database to be auto-incremented integers

## 4. Design Logic/Justification and Constraints

---

Definitions:

- Users: a user should provide personal information about: name, for identification; username, email and passwords, for logging in or signing up; address, including city state and country, and affiliation, which are not required and can be vacant.
  - Users will also be able to edit their profiles later
- Questions: To post a question, a user will need to provide a particular topic and subtopic, a rough description as title, and clear details about the question (body). The posted time and status will be recorded automatically at the moment of posting.
  - status will be defaulted to “unanswered” at the time of submission, and will only be changed when an answer has been posted
  - a best answer will be selected based on the total number of likes
- Answers: To answer a question, a user will post an answer corresponding to that particular question. A timestamp will also be recorded when posted as well as the answer number: “anum”. We chose to use anum as the partial key as it is unique for each question but not across all questions, similar to the number of order items in orders. A best answer will be chosen based on the number of likes, with each question only allowing one best answer. Each answer can only be liked once by distinct users not including the user itself who posted the answer.
- Topic: Topics are stored uniquely by an id and stores the “Topic” as well as “Subtopic”. Topic describes more general fields such as “Math” or “Computer Science”, while “Subtopic” defines more specific topics such as “Algebra” or “C++”. The justification for using one table for topics is due to two factors: topics being a two-tier hierarchy therefore we can separate topics into a “general” and “specific”, and the assertion that questions

have to be labeled under one topic on each level of the hierarchy. The combination of these assumptions means that each question may only be associated with a “unique” set of mapping from topic to subtopic, therefore storing the topic as one table would work.

#### Additional Constraints/Notes:

- Based on uid, qid and “anum”, a like can be tracked from user to the answer num of a particular question.
  - Every user can like each answer once to prevent inflation of likes
  - Users cannot like their own posts (trigger on insert)
- Users can answer their own questions, this is to allow them to clarify something in the original post.
- We chose to separate questions and answers instead of combining them as “posts” because answers depend on questions. Without a question, an answer cannot exist independently. Because of this, answers cannot be defined by a unique identifier, so it must exist as a separate “weak” entity.

## 5. Database Setup

---

```
CREATE TABLE Users(
    uid INTEGER auto_increment PRIMARY KEY,
    firstname VARCHAR(50) NOT NULL ,
    lastname VARCHAR(50) NOT NULL,
    username VARCHAR (50) NOT NULL,
    email VARCHAR(50) NOT NULL,
    password VARCHAR(100) NOT NULL,
    city VARCHAR(30) NOT NULL,
    state VARCHAR(20) NOT NULL,
    country VARCHAR(30) NOT NULL,
    personal_profile VARCHAR(200),
    level VARCHAR (20),
    affiliation VARCHAR(30)
);

CREATE TABLE Topic(
    tid INTEGER auto_increment PRIMARY KEY,
    tname VARCHAR(20) NOT NULL,
    stname VARCHAR(20) NOT NULL
);

-- Question
CREATE TABLE Question(
    qid INTEGER auto_increment PRIMARY KEY,
    asker INTEGER NOT NULL,
    tid INTEGER NOT NULL,
    title VARCHAR(100) NOT NULL,
    qbody VARCHAR(200) NOT NULL,
    questiondate DATETIME NOT NULL,
    qstatus VARCHAR(10) NOT NULL,
    FOREIGN KEY (asker) REFERENCES Users(uid),
    FOREIGN KEY (tid) REFERENCES Topic(tid)
);

CREATE TABLE Answer(
    qid INTEGER NOT NULL,
    anum INTEGER NOT NULL,
    answerer INTEGER NOT NULL,
    abody VARCHAR(240) NOT NULL,
    answerdate DATETIME NOT NULL,
    bestanswer BOOL,
    numlikes INTEGER,
    PRIMARY KEY (qid, anum),
    FOREIGN KEY (answerer) REFERENCES Users(uid),
    FOREIGN KEY (qid) REFERENCES Question(qid)
);

-- Likes
CREATE TABLE Likes(
    uid Integer NOT NULL,
    qid Integer NOT NULL,
    anum Integer NOT NULL,
    PRIMARY KEY (uid, qid, anum),
    FOREIGN KEY (uid) REFERENCES Users(uid),
    FOREIGN KEY (qid,anum) REFERENCES Answer(qid,anum)
);
```

Above is the setup for the database with constraints, primary/foreign keys, and type definitions for all attributes for the tables discussed in the relational schema.

## 6. Useful Triggers/Procedures

---

### Searching:

We define relevance by matching the list of query words against the body of the question, the title of the question, and the answers to the question. Using Fulltext indexes, we match the keyword against each column specified earlier. When there is a match with any word, the weight will be nonzero. We decided to do a weighted sum of the individual weights with priority for title, body, then answers because the question (especially title) itself is likely what users will be searching for an answer to. All questions with zero weighted sums are ridded of because no matches were found. Below is the query and a sample output. An alternative version was also made that orders by reverse chronological order (recent questions).

```
CREATE DEFINER='root'@'localhost' PROCEDURE `Search_Chronologically` (keywords VARCHAR(1000))
BEGIN
    drop table if exists Info;

    Create table Info as Select * From Q_A_T;
    Create Fulltext Index FTqbody on Info(qbody);
    Create Fulltext Index FTtitle on Info(title);
    Create Fulltext Index FTabody on Info(abody);
    SET @KeywordQuery = keywords;
    With
    Stats as (Select qid, title as Title , qbody as Question_Body, questiondate, qstatus, tname as Topic ,stname as Subtopic,
        Match(qbody) Against (@KeywordQuery in Boolean Mode) as w1,
        Match(title) Against (@KeywordQuery in Boolean Mode) as w2,
        Match(abody) Against (@KeywordQuery in Boolean Mode) as w3
        From Info)
    Select qid, Title, Question_Body, questiondate, qstatus, Topic, Subtopic, (3*w1 + 5*w2 + Sum(w3)) as TotalWeight
    From Stats
    Group by qid
    Having TotalWeight > 0
    Order by questiondate DESC;
END
```

	qid	Title	Question_Body	Topic	Subtopic	TotalWeight
▶	q12	Help with syntax	Can someone explain to me why this code is giving me an error?	Computer Science	JAVA	3.5139834880828857
	q01	Need Help	As a beginner, which programming language should I learn?	Computer Science	Programming	2.1962396800518036
	q09	Integral help	How should I define the limits of an integral?	Mathematics	Calculus	2.1962396800518036
	q03	a math problem	anyone can explain Squeeze Law for me?	Mathematics	Calculus	1.3177438080310822
	q05	Hello, this is Emma	I am a graduate student in NYU. Nice to meet you!	Others	Introduction	6.98993843793869
	q13	Hi	Hey everyone, this is my first time using this site.	Others	Misc	1.3177438080310822

## Likes & Best Answer:

One of the dilemmas that we came across was how we should handle the selection of best answers while simultaneously updating the number of likes in the Answers table. Our solution was to create a trigger that automatically updates the number of likes by matching a new row's anum and qid. We wanted to have the selection of the best answer be an automated procedure, so we defined it as any answer that has the max number of likes to that particular question. If there is a tie then an arbitrary one is selected. We also defined an additional constraint that the number of likes must be non-zero for it to be selected as a best answer. This is because we wanted some sort of indication that an answer is worthy of being selected as a "best" answer, and since a user cannot like their own answer, someone different has to endorse them.

```
DELIMITER $$  
CREATE TRIGGER update_likes  
AFTER INSERT ON Likes  
FOR EACH ROW  
BEGIN  
    UPDATE Answer  
        SET numlikes = numlikes + 1  
    WHERE NEW.qid = qid AND NEW.anum = anum;  
    UPDATE Answer LEFT JOIN MaxLikes ON Answer.qid = MaxLikes.qid  
        SET bestanswer =  
            CASE WHEN Answer.anum = an AND numlikes = ML AND numlikes > 0 THEN True  
            ELSE False  
        END;  
END $$
```

\*MaxLikes is a table mapping each question's max number of likes to the answer number that corresponds to it.

## Question Status:

The question status is determined by the existence of an answer posted to that question. Since there are only two possible statuses that a question can have (answered & unanswered), we simply have to make a trigger that checks if a question has an answer posted under it when it was originally unanswered, if so, then change it to answered.

```
Drop Trigger if exists update_status;  
DELIMITER $$  
CREATE TRIGGER update_status  
AFTER INSERT ON Answer  
FOR EACH ROW  
BEGIN  
    UPDATE Question  
        SET qstatus = "Answered"  
    WHERE New.qid = Question.qid AND qstatus != "Answered";  
END $$
```

## Enforcing Constraints

There are some additional constraints that need to be enforced outside of the table definitions.

For example, we chose to not allow a user to answer a question more than once (although they can answer a question posted by themselves, perhaps to clarify something). This is to ensure that questions are not spammed by users. We also have to restrict liking answers. Users are not allowed to like their own answers because it would artificially inflate the answer's value. These two restrictions are enforced in the triggers below.

```
Drop Trigger if exists same_answerer
DELIMITER $$

CREATE TRIGGER same_answerer
Before INSERT ON Answer
FOR EACH ROW
BEGIN
    if NEW.answerer in (Select answerer From Answer Where New.qid = Answer.qid) Then
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "You can't answer the same question twice";
    end if;
END $$

Drop Trigger if exists same_like
DELIMITER $$

CREATE TRIGGER same_like
Before INSERT ON Likes
FOR EACH ROW
BEGIN
    if (NEW.uid = (Select answerer From Answer Where New.qid = Answer.qid and New.anum = Answer.anum)) Then
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "You can't like your own answer";
    end if;
END $$

DELIMITER ;
```

## Defining anum

One of the few troubles that we faced was having to deal with how to assign a new number for an answer to a question. This would normally be done with an auto increment constraint, however due to the way Answer is defined two auto-increment attributes would not be possible. To solve this problem, we created a function that automatically calculates the next anum available to assign to a new answer based on the question id.

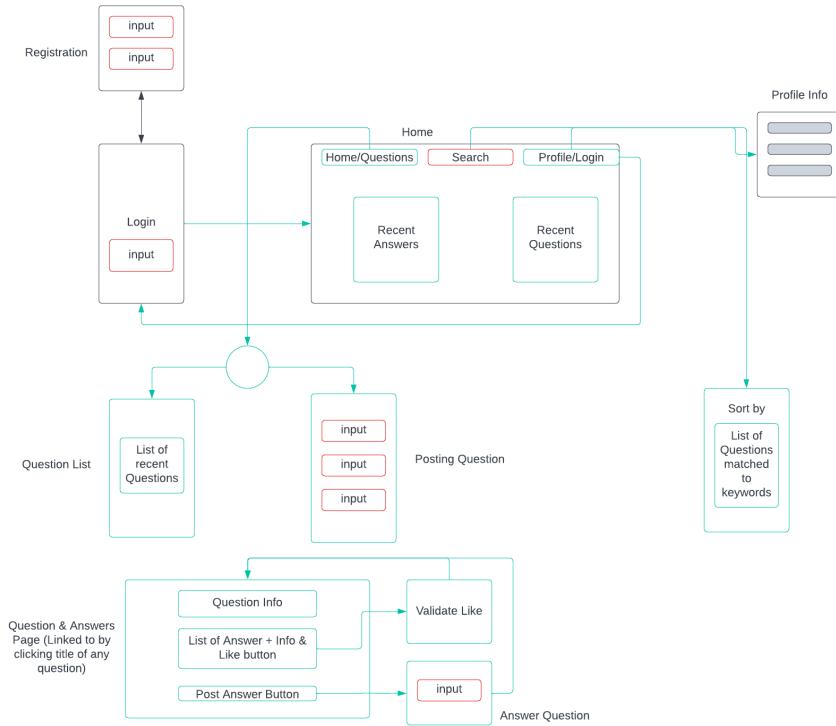
```
CREATE DEFINER='root'@'localhost' FUNCTION `next_anum`(
    questionid INTEGER
) RETURNS int
    DETERMINISTIC
BEGIN
    DECLARE anum INTEGER;
    Select COUNT(*) + 1 into anum
    FROM Answer
    Where qid = questionid;
    RETURN anum;
END
```

***Additional Views/Procedures present in submission folder***

# 7. Web Application Design

## 7.1 Outline/Initial Conceptual Design

To plan out the design for our web-application, we first consulted the requirements of the project. We listed out all requirements needed by the system and planned out a conceptual design. This design includes a login page, a registration page, a homepage and several functional pages. The design can be seen in the figure below.



*\*this is not representative of our final design, this diagram served as a guide for our eventual design\**

We decided to have a design that was centered around the homepage, with it wiring to all other pages in the application. To implement this behavior, the homepage has a navigation bar to organize and navigate to different functional login pages (user's profile, posting question,

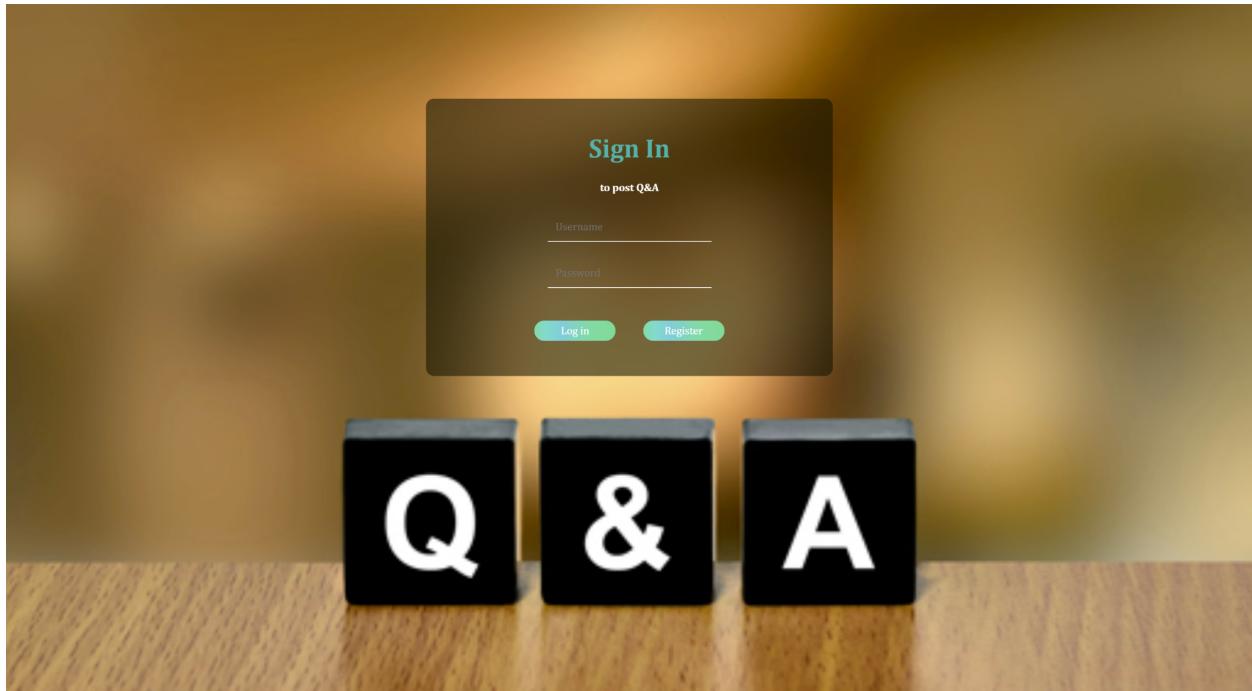
question-list, question-details, recent-answers, search&search-result). The navigation bar also serves its purpose as the way for users to browse (search).

Since users must have an account to perform actions, many actions such as posting answers, questions, or liking will be restricted if the user has not logged in. However, the user will still be able to browse by searching for keywords and seeing matched questions. This is designed so all possible users logged in or not will be able to look up questions and answers.

We wanted this design to feel as connected and coherent as possible. Therefore, much of our design is interconnected. What this means is that you can get to the homepage from almost anywhere, and you can access any question's details as long as you are on a page that lists the question. We took inspiration from other question-answering systems such as stack overflow to design our system. We also took effort in making our system as user-friendly as possible, this meant much use of icons and minimal clutter of information.

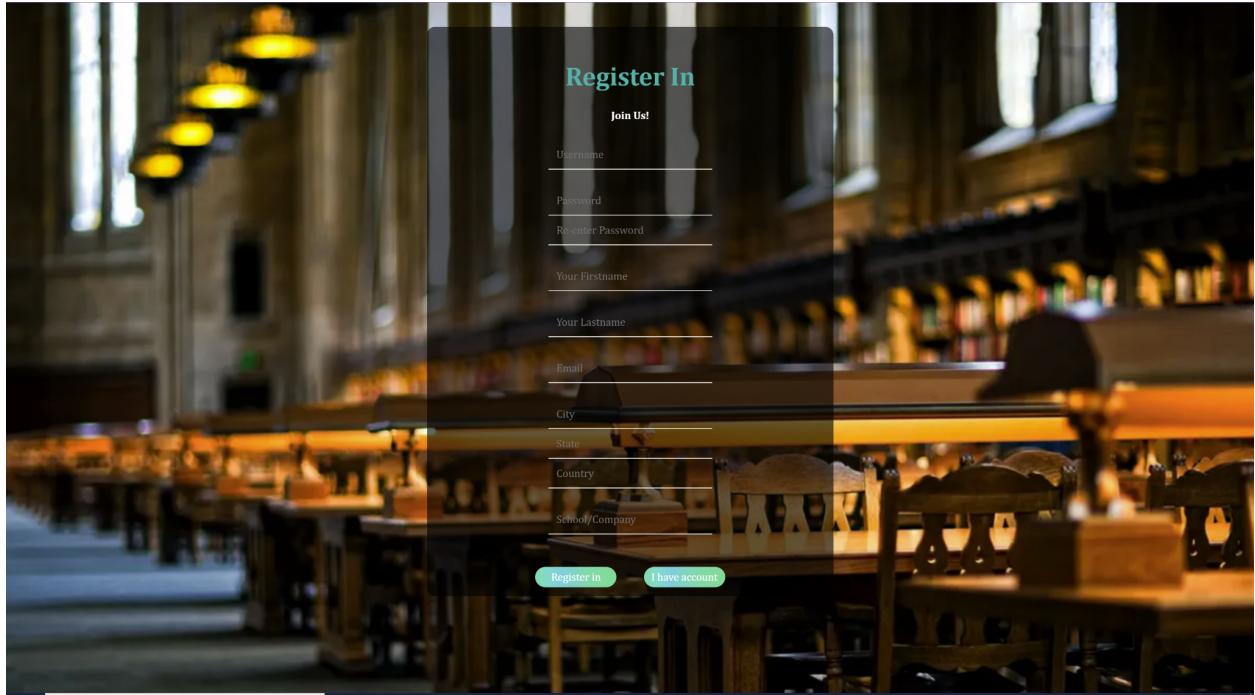
The next few sections will go into more detail about each individual page's design detail and justification.

## 7.2 Login, Registration & Profile Pages



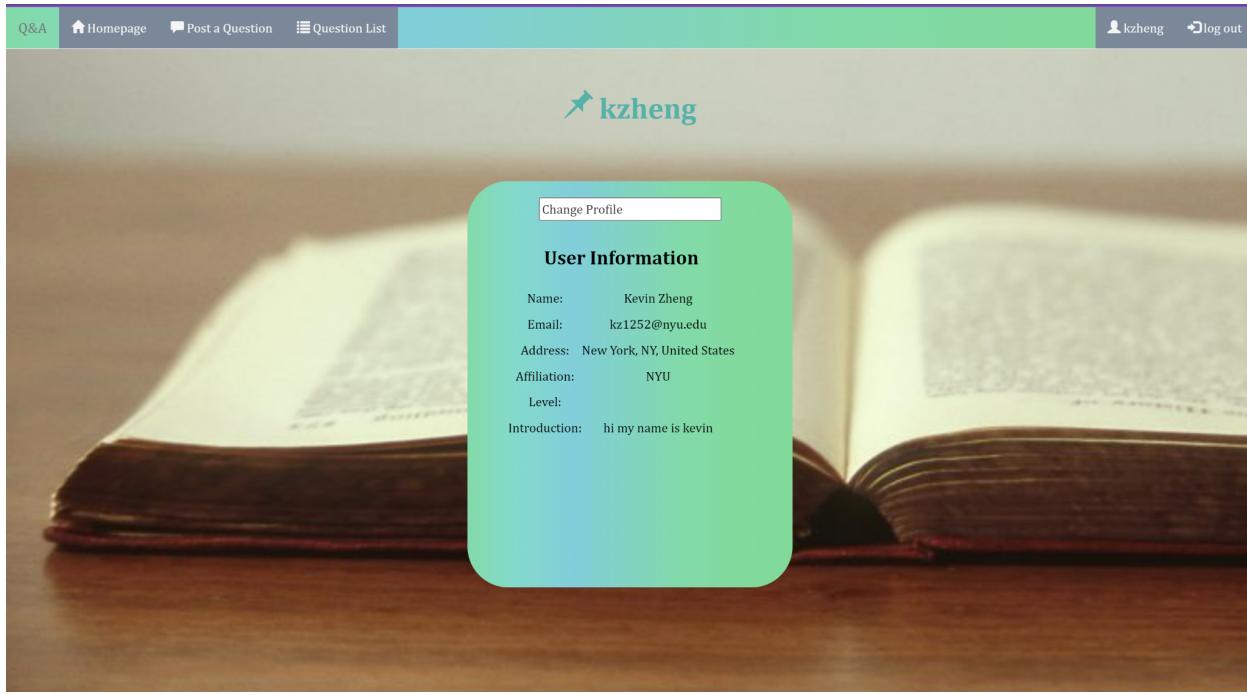
The page above is the login page. It has two input boxes for inputting user info and options to login/register. This page serves as the gateway into our main system.

Using error detection, the username and password input boxes cannot be empty, or it will give an error message. Logging in will check the username and password against the database to see if there is a match. If the username does not exist or password is invalid, an error message “Your Login Name or Password is Invalid” will show up and take the user back to the login page. This page was designed to be minimalistic, as the only functionalities needed are to login/register.



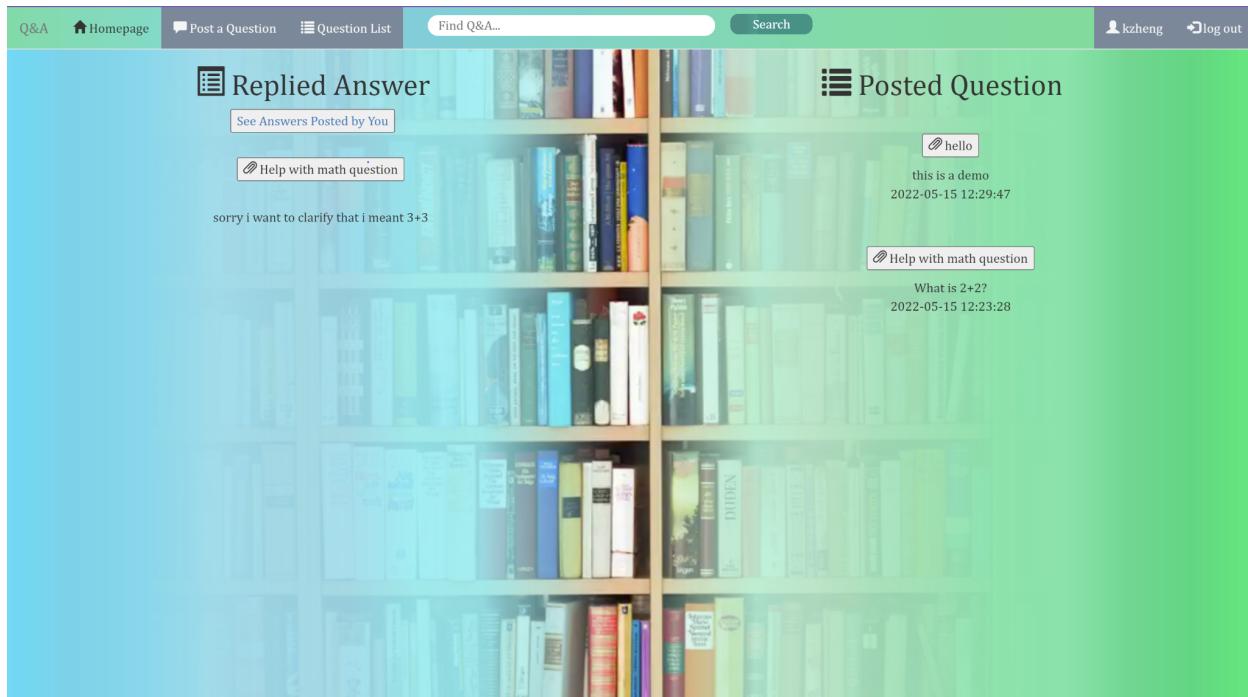
The registration page is firmly adhered to the login page, which means users can only access it through the login page. After completing registration it will jump to the login page automatically for the user to login.

The registration page will ask the user for their name, password, password confirmation, email, and address (city, state, country). The collected information will be listed on the user's profile page. Duplication of usernames is detected when the register button is clicked. If duplication happens, the user will be required to re-enter another valid username to finish the registration. The passwords must also match, or else an error will be displayed and the user will be prompted to re-register. Lastly, the email address must be valid and contain an "@", or it will be rejected. If the user desires to login, they can click "I have an account" to be redirected back to the login page.



The profile page is minimal and has the username at the top to indicate profile. The profile page lists all the information inputted by the user upon registration (except password for security reasons), and also has a text box for the user to change their profile introduction.

### 7.3 Homepage



Upon successful login, users are led to the homepage. The body and the navigation bar are two essential components. In the navigation bar, the search textbox is present in the middle as a divider between the left and right sections. The left section contains all functional options and the right section contains all options associated with user actions.

The profile page option (identified by the username) in the right hand section will not show up if a user is not logged in, instead being replaced by the login prompt.

The body of the homepage is divided into two parts. When the user is not logged in, both parts will display the message “please log in first”. If a user is logged in, the left section displays recent answers to any of the user’s questions, while the right section gives a reverse chronological list of user’s posted questions. In the body of the left section, there is a button that can jump to another page which shows answers posted by the user, as shown in the image below.

sheesh

on 2022-05-15 12:24:53

0

[See Question](#)

Enter whatever your heart desires to know

on 2022-05-15 12:27:21

0

[See Question](#)

sorry i want to clarify that i meant  $3+3$

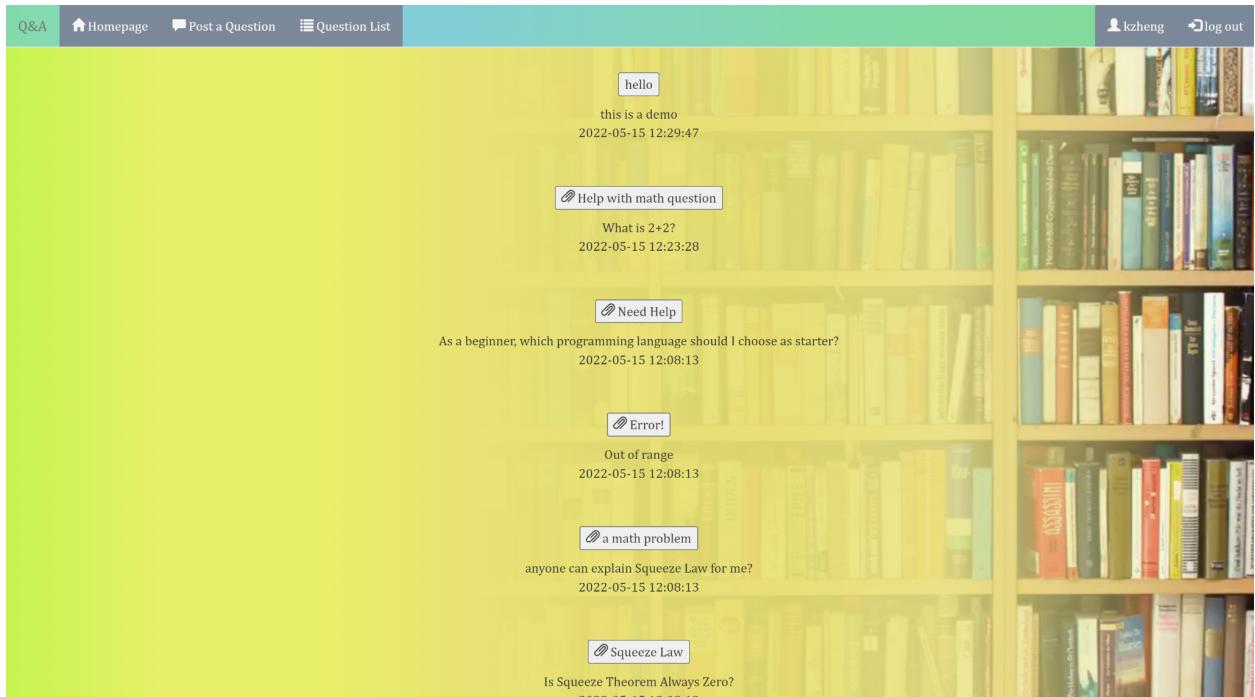
on 2022-05-15 12:28:10

0

[See Question](#)

## 7.4 Question List & Search

Q&A [Homepage](#) [Post a Question](#) [Question List](#) [kzheng](#) [log out](#)



hello

this is a demo  
2022-05-15 12:29:47

Help with math question

What is  $2+2$ ?  
2022-05-15 12:23:28

Need Help

As a beginner, which programming language should I choose as starter?  
2022-05-15 12:08:13

Error!

Out of range  
2022-05-15 12:08:13

a math problem

anyone can explain Squeeze Law for me?  
2022-05-15 12:08:13

Squeeze Law

Is Squeeze Theorem Always Zero?  
2022-05-15 12:08:13

To find a question directly, two methods are provided. The first method is browsing a question list ordered from the newest to oldest. Recently posted questions are accessed by clicking on the “Question List” option in the navigation bar and is also one of the fastest ways to see up-to-date posted questions. Each question is separated into three parts: the title, body, and date posted. To access more details about the question, the title functions as a button that can

be pressed to be redirected. Additionally, each question will have an indication by the title (paperclip icon) that will indicate whether or not the question has had an answer provided, thus making it easier for the user to visually filter out questions with no answers. The page was designed to be an intermediary to find questions that pertain to the user's interest, hence it has less information.

The screenshot shows a web-based Q&A application. At the top, there is a navigation bar with links for 'Homepage', 'Post a Question', 'Question List', 'Find Q&A...', 'Search', and user account information ('kzheng' and 'log out'). Below the navigation bar is a search bar labeled 'Sort Questions By: relevance' with a dropdown arrow and a 'Select' button. The main content area displays a list of questions:

- Hello, this is Emma**  
I am a graduate student in NYU. Nice to meet you all!  
2022-05-15 12:08:13
- Help with math question**  
What is  $2+2$ ?  
2022-05-15 12:23:28
- a math problem**  
anyone can explain Squeeze Law for me?  
2022-05-15 12:08:13
- Math**  
What is the dimension of real anti-symmetric  $4 \times 4$  matrices?  
2022-05-15 12:08:13

Another way to find a question if users know what content they are looking for is using the search bar. They can search with one or more keywords. This brings them to a similar page to the question list, except you can sort the results by either relevance or time (recency). The standard of sorting can be switched at any time. The limit length of the keywords in the search bar is 200 characters. This page will only show questions that have matched keywords in any of the title, body, or answers.

## 7.5 Questions & Answers



a math problem

anyone can explain Squeeze Law for me?

asked by KevinInNYU on 2022-05-15 12:08:13

The squeeze (or sandwich) theorem states that if  $f(x) \leq g(x) \leq h(x)$  for all numbers, and at some point  $x=k$  we have  $f(k)=h(k)$ , then  $g(k)$  must also be equal to them.

by SuperMario on 2022-05-15 12:08:13

↳ 2

[Like]

if  $f(x) \leq g(x) \leq h(x)$  for all numbers, at some point  $x=k$   $f(k)=h(k)$ , then  $g(k)$  should be equal to them.

by Snowfish on 2022-05-15 12:08:13

↳ 0

[Like]

Maybe you could try khan academy

by ROOF on 2022-05-15 12:08:13

↳ 3

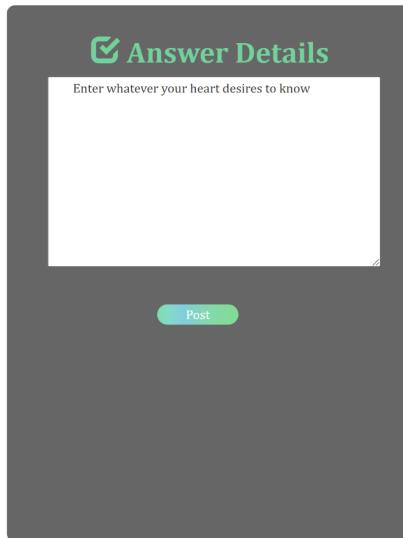
B

[Like]

+ Post a Answer

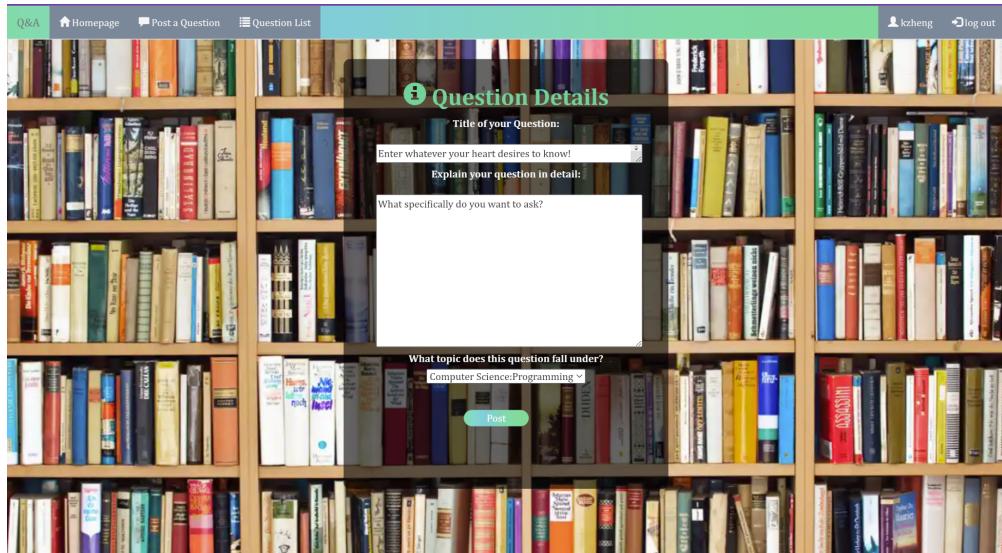
By clicking on a title of a question, users can enter the interface displaying question details and all affiliated answers. At the top of the page, the title, body, user, and date posted are listed for the question. Beneath that, affiliated answers are listed in rows. For each answer, the body, user who posted the answer, date posted, and number of likes are listed. For each answer, there is also an associated like button. If the user performs an illegal action (liking own answer or liking twice), an error message will show up indicating it. The answer with the most likes will be selected as the best answer, with a “B” icon to illustrate this.

At the bottom of the page, a button is provided for the user to post an answer to the question, which will redirect the user to the posting answer page.



On the answer posting page, users will be able to input their answers to the question with a max character length of 200 characters. If the user has already answered the question, a notice will appear at the top indicating so. If not, after submission the user will be redirected back to the question and answers page.

Lastly, to ask a question, users can click on the “Post a Question” link in the navigation bar. The interface for posting questions is similar to posting answers. The difference is that a title and topic needs to be provided for the question.



## 8. Sample Session Walkthrough

This section will detail a sample walkthrough of actions a user will take.

### 8.1 Posting a Question

We will assume we are a new user and want to register with a username called “Walkthrough”.

After registration we will login with the information provided and the homepage will be displayed.

The registration form is titled "Register In". It has a "Join Us!" button at the top. Below it are several input fields:

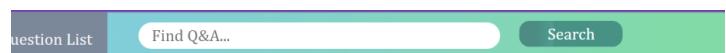
- Username: Walkthrough
- ... (two additional input fields)
- Name: Kevin
- Name: Zheng
- Email: kevin.z@nyu.edu
- Location: New York
- Location: New York
- Location: United States
- Location: NYU

At the bottom are two buttons: "Register in" and "I have account".



We will now click the “Post a Question” at the top left navigation bar to post a question. We will be brought to the page on the left and be prompted to input the details. After submitting the question, we will be brought to the question page with our newly posted question.

The "Question Details" page features a title "Title of your Question:" with the text "Why is Python superior to C++?". Below it is a "Post a Question" button. A text area labeled "Explain your question in detail:" contains the placeholder "I need an answer to this question!". At the bottom, there is a dropdown menu for "What topic does this question fall under?" with "Computer Science:Python" selected, and a "Post" button.



Why is Python superior to C++?  
I need an answer to this question  
asked by Walkthrough on 2022-05-15 18:06:43  
Did not find any replied answers.

[+ Post a Answer](#)

## **8.1 Posting an Answer**

Say you now want to post an answer to a question. From the home page, we can use the search bar to look at questions. Let's suppose our keywords are "python math error"

The screenshot shows the Q&A homepage. At the top, there are navigation links: 'Homepage', 'Post a Question', 'Question List', 'Find Q&A...', 'Search', 'Walkthrough', and 'Log out'. Below the navigation, there are two main sections: 'Replied Answer' on the left and 'Posted Question' on the right. The 'Replied Answer' section has a heading 'Replied Answer' and a button 'See Answers Posted by You'. It displays the message 'No recent answers found.' The 'Posted Question' section has a heading 'Posted Question' and a question card for 'Why is Python superior to C++?'. The card includes the text 'I need an answer to this question', the date '2022-05-15 18:06:43', and a small icon.

After searching for that input, we see a list of questions that match the query we just made, we can then select any one of them to view the details.

The screenshot shows the search results page for the query "python math error". The top navigation bar is identical to the homepage. Below it, there is a search filter 'Sort Questions By: relevance' with a 'Select' button. The main content area displays several questions in a list format. The first question is 'Need Help' with the text 'As a beginner, which programming language should I choose as starter?' and the date '2022-05-15 12:08:13'. The second question is 'Why is Python superior to C++?' with the text 'I need an answer to this question' and the date '2022-05-15 18:06:43'. The third question is 'Error!' with the text 'Out of range' and the date '2022-05-15 12:08:13'. The fourth question is 'a math problem' with the text 'anyone can explain Squeeze Law for me?' and the date '2022-05-15 12:08:13'. The fifth question is 'Math' with the text 'What is the dimension of real anti-symmetric  $4 \times 4$  matrices?' and the date '2022-05-15 12:08:13'. Each question card includes a small icon.

For this sample walkthrough we will use the first question. After clicking on the title, we will be brought to the question page with answers.

The screenshot shows a user interface for a Q&A website. At the top, there is a navigation bar with links for "Homepage", "Post a Question", "Question List", "Find Q&A...", "Search", "Walkthrough", and "Log out". The main content area has a title "Need Help" and a question: "As a beginner, which programming language should I choose as starter?". Below the question is the timestamp "asked by AlexJ on 2022-05-15 12:08:13". Three answers are listed:

- sheesh**  
by kzheng on 2022-05-15 12:24:53  
0 likes  
[Like]
- I recommand Java and C; Python is easy to learn but not appropriate for beginner to learn data structure.  
by Jenflower on 2022-05-15 12:08:13  
10 likes  
B  
[Like]
- C++ IS THE BEST WITH NO REASON  
by Funplus\_Harry on 2022-05-15 12:08:13  
2 likes  
[Like]

We can now answer this question by clicking the “Post an answer” button at the bottom (offscreen in this screenshot). It will then bring you to the posting answer page. After the user is done writing their answers, they can post it and will be redirected to the question and answers page, with the new answer posted by the user now available.

The screenshot shows a modal window titled "Answer Details" with a checked checkbox icon. The text input field contains the message: "I think you should use python, I really like python". Below the input field is a "Post" button. To the right of the modal, the original question and its answers are visible.

**Need Help**

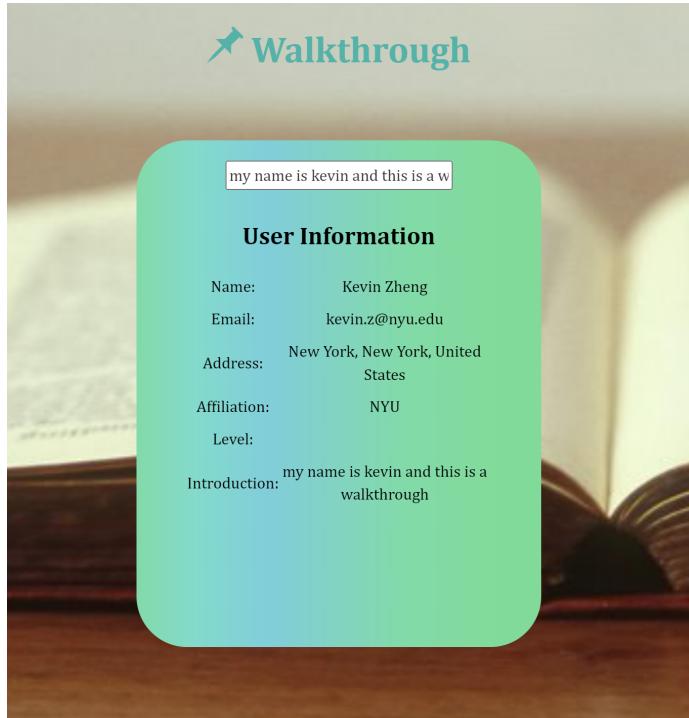
As a beginner, which programming language should I choose as starter?

asked by AlexJ on 2022-05-15 12:08:13

I think you should use python, I really like python  
by Walkthrough on 2022-05-15 18:10:14  
0 likes  
[Like]

### **8.3 Miscellaneous Pages**

This is the result of going into our profile information and changing the profile description.



This is the result of recent answers posted by user, with the answer we just provided.

I think you should use python, I really like python

on 2022-05-15 18:10:14

0

[See Question](#)

This is the result of going to the question list page, which now lists our question at the top

The screenshot shows a list of three posts on a question list page. The background is yellow with a bookshelf pattern.

- Why is Python superior to C++?**  
I need an answer to this question  
2022-05-15 18:06:43
- hello**  
this is a demo  
2022-05-15 12:29:47
- Help with math question**  
What is  $2+2$ ?  
2022-05-15 12:23:28

Lastly, say we liked an answer so much we wanted to give it a like, we can do so which will result in the like counter increasing.

sheesh

by kzheng on 2022-05-15 12:24:53

1

Like