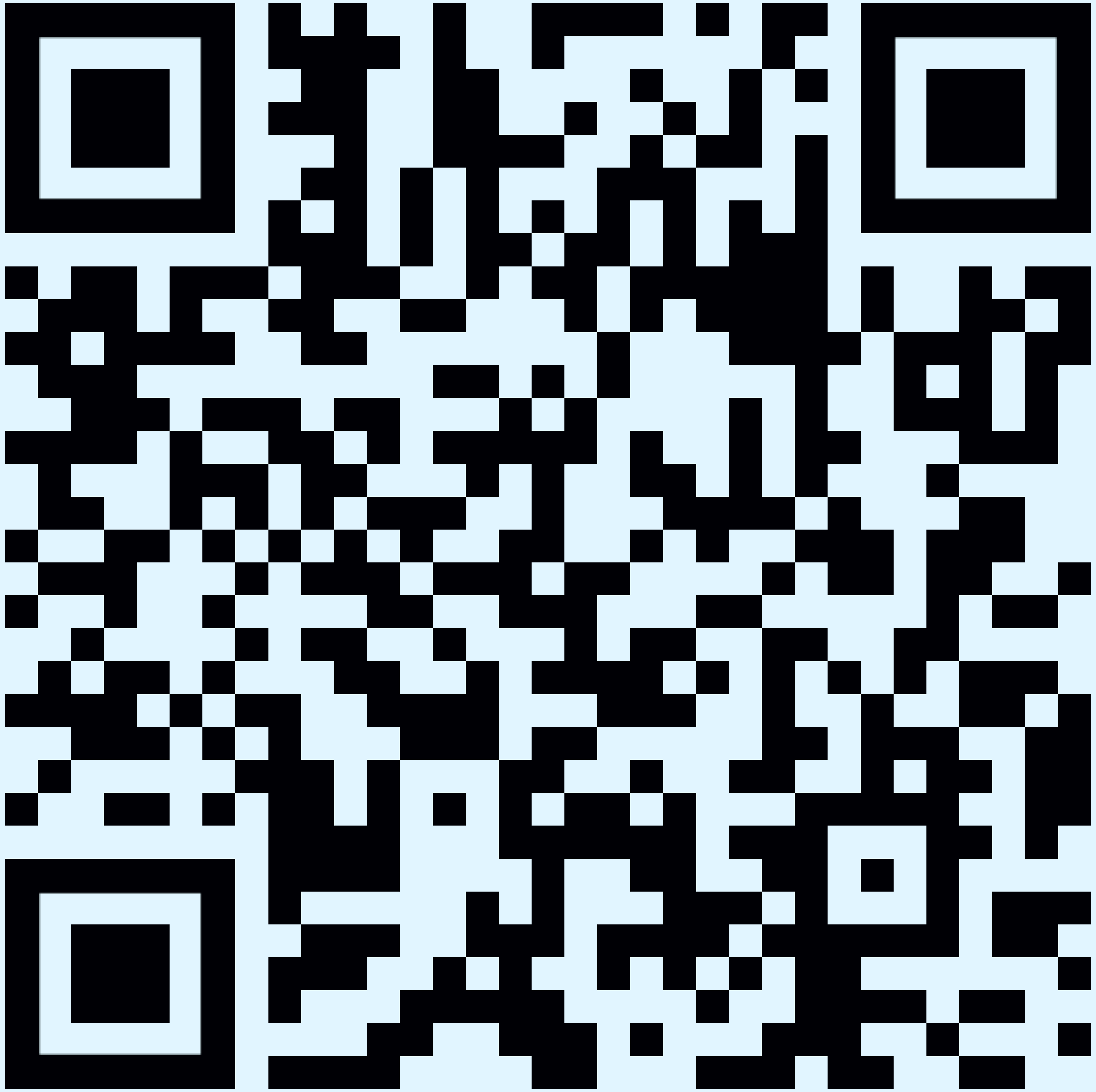
SOS



TEAM 10

Project : SOS!

Studying OS

2024.06.04

20120107 Jihwan Kim / 20102127 Seongmin Hwang / 20102102 Gwanwoo Kim / 19102094 Hyunsoo Shin

Agenda

1. **Project Overview**
   - About Project
   - Technologies Stack

2. **Implementation**
   - System Architecture
   - Key Features

3. **Problem Analysis and Debugging**
   - Issues Encountered and Solutions
   - Debugging Process

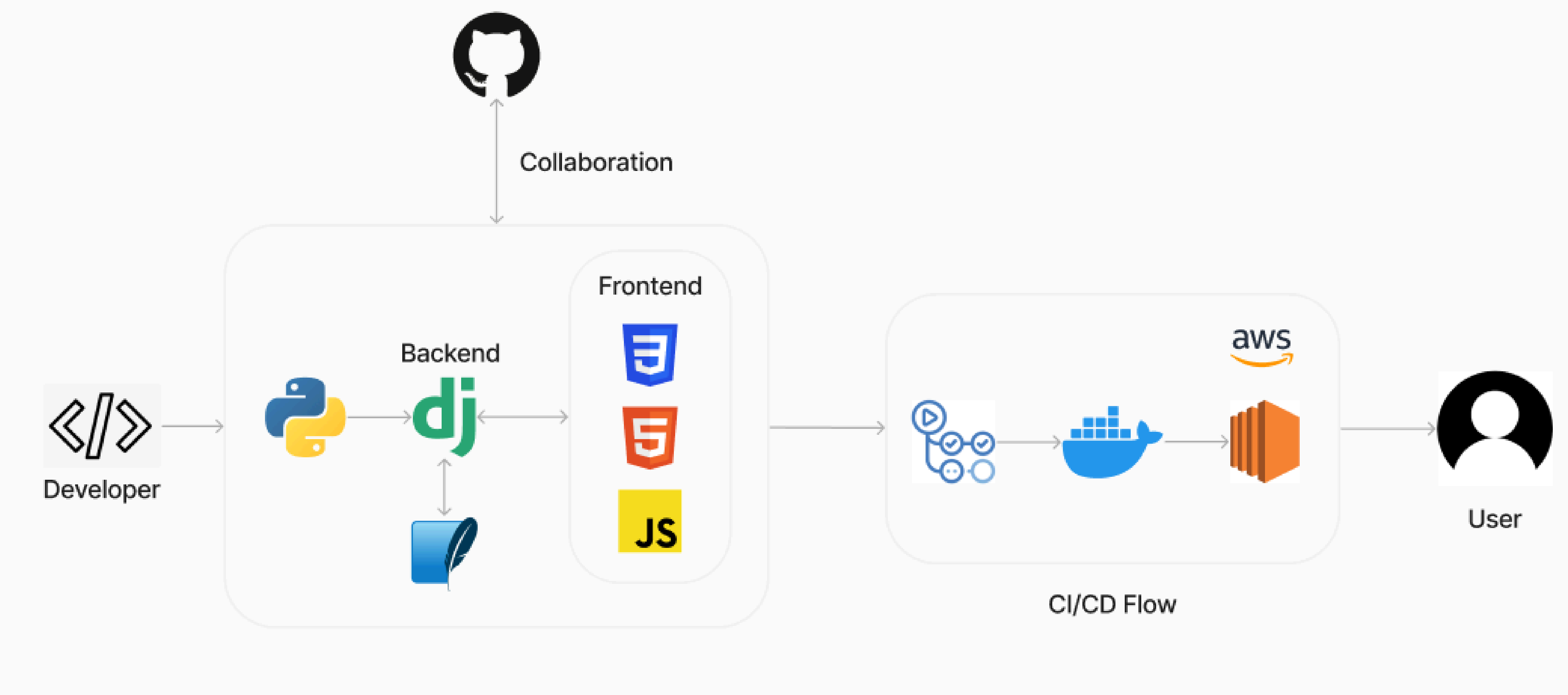
4. **Results**
   - Demonstration

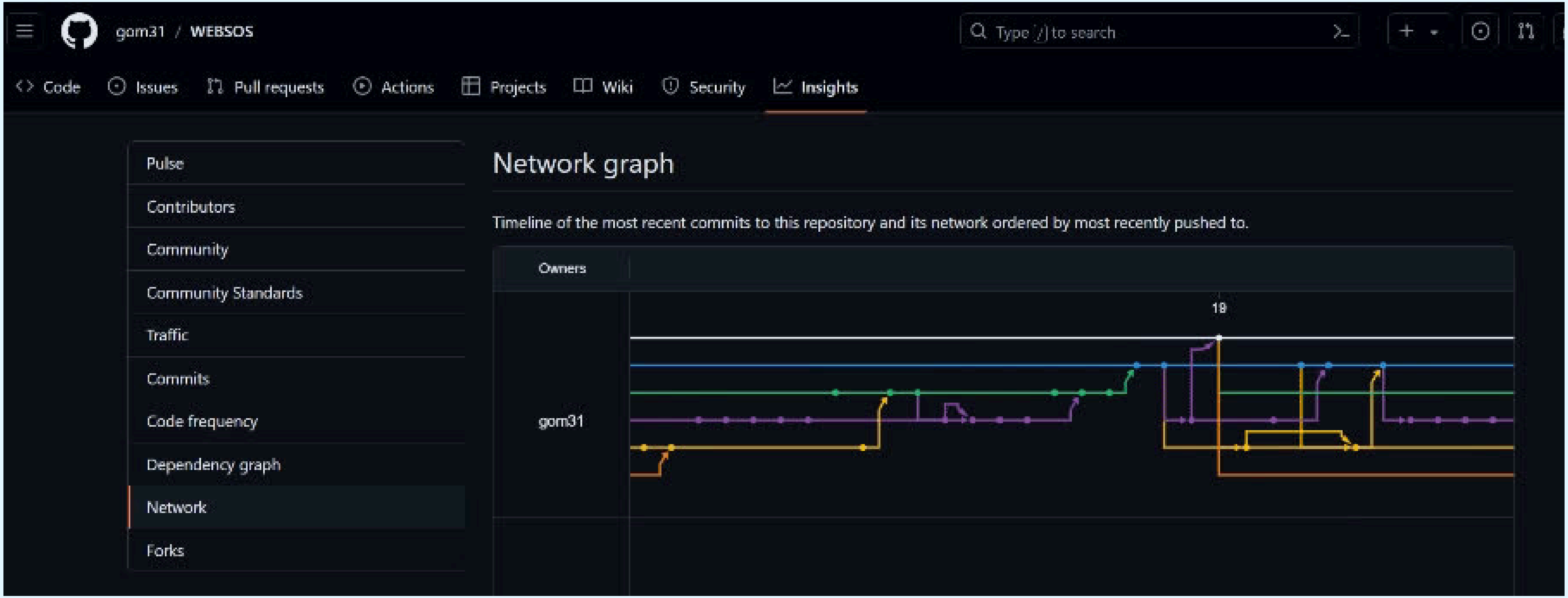
Project Overview

About the project

Prepare your OS exam!

- Quiz
- Study
- Examlog
- Retest

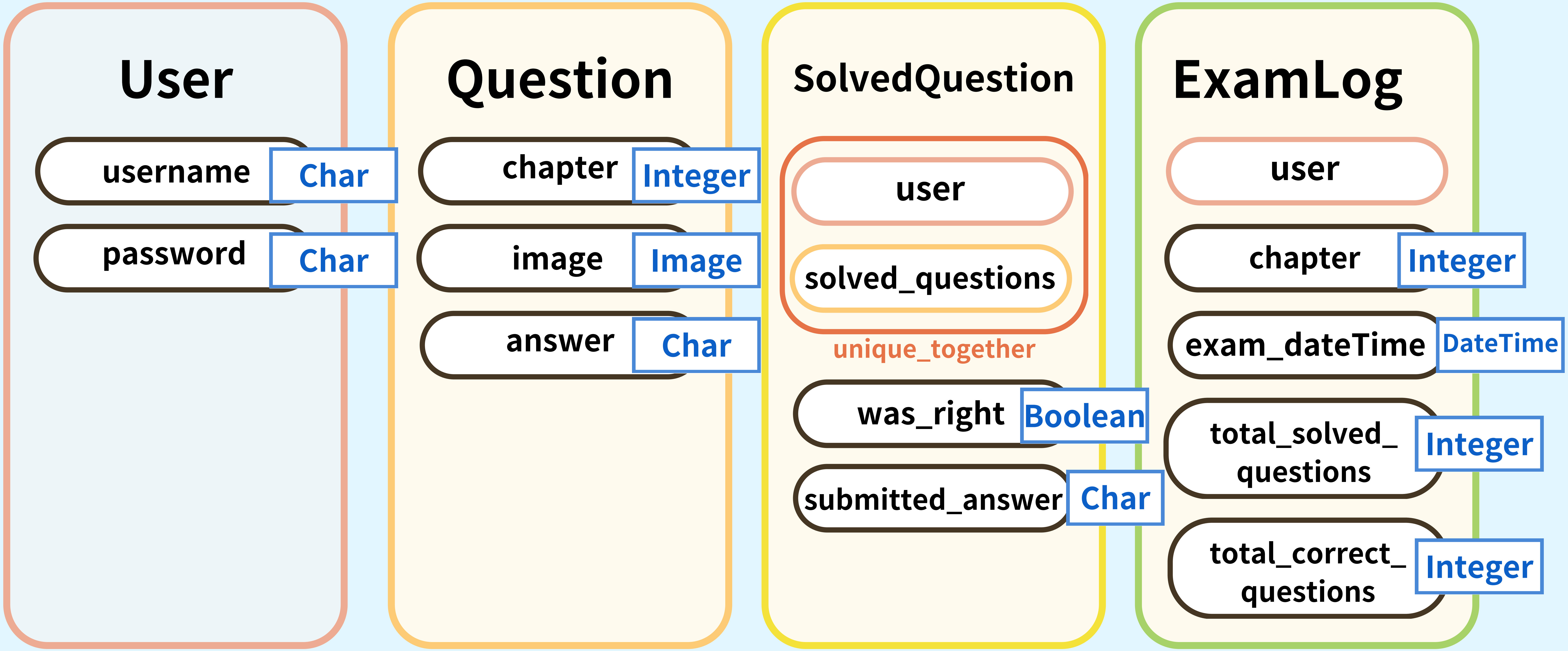




Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

DB Schema



User

username Char

password Char

Question

chapter Integer

image Image

answer Char

SolvedQuestion

user

solved_questions

unique_together

was_right Boolean

submitted_answer Char

ExamLog

user

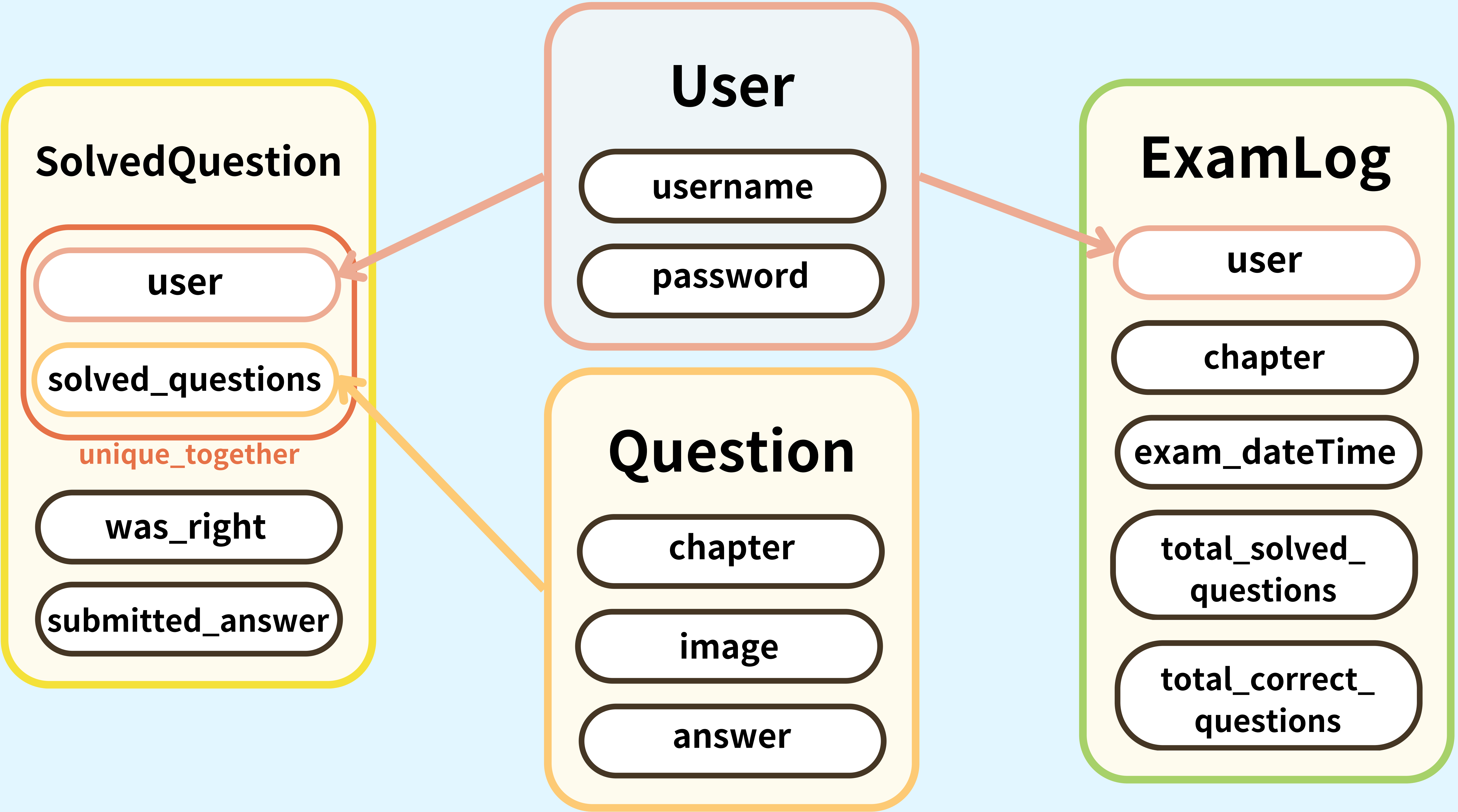
chapter Integer

exam_dateTime DateTime

total_solved_questions Integer

total_correct_questions Integer

DB Schema



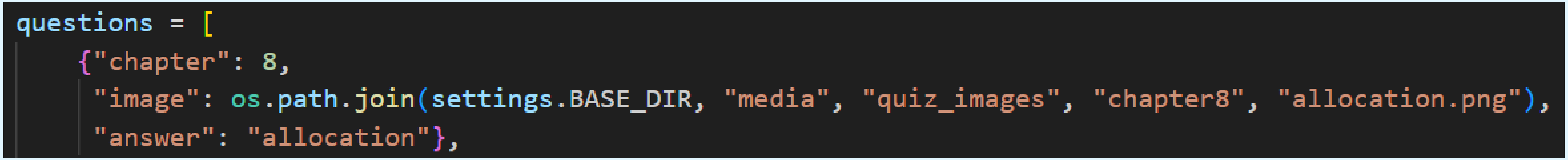
SolvedQuestion
- user
- solved_questions
- unique_together
- was_right
- submitted_answer

User
- username
- password

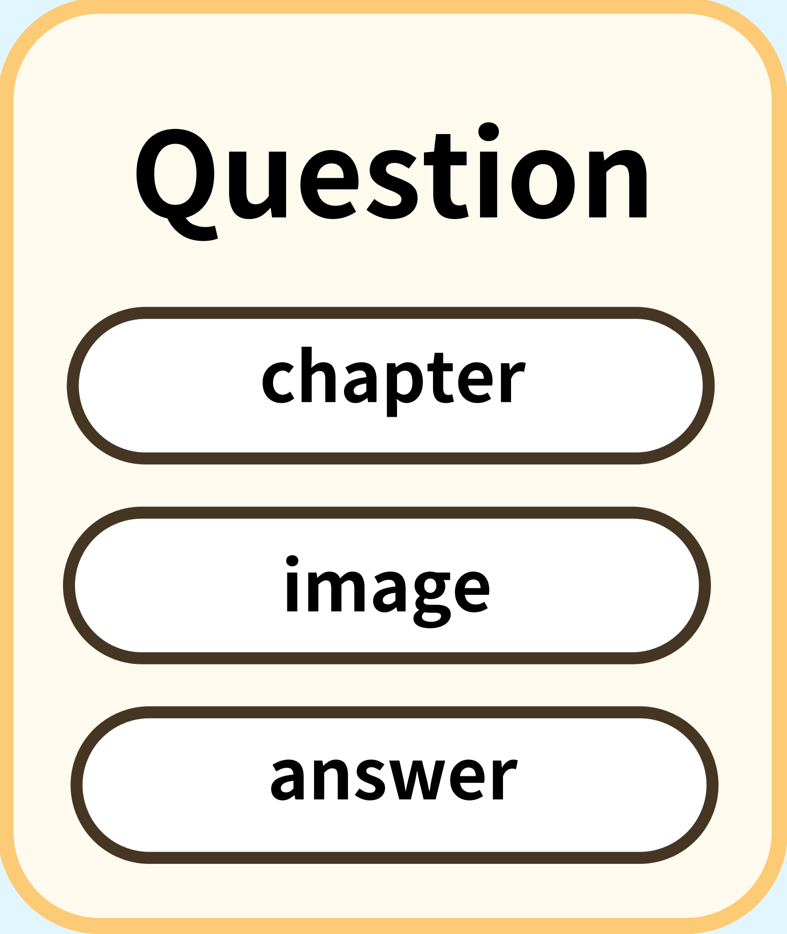
Question
- chapter
- image
- answer

ExamLog
- user
- chapter
- exam_dateTime
- total_solved_questions
- total_correct_questions

Load_questoins



```
questions = [
    {"chapter": 8,
     "image": os.path.join(settings.BASE_DIR, "media", "quiz_images", "chapter8", "allocation.png"),
     "answer": "allocation"},
```



Question

chapter

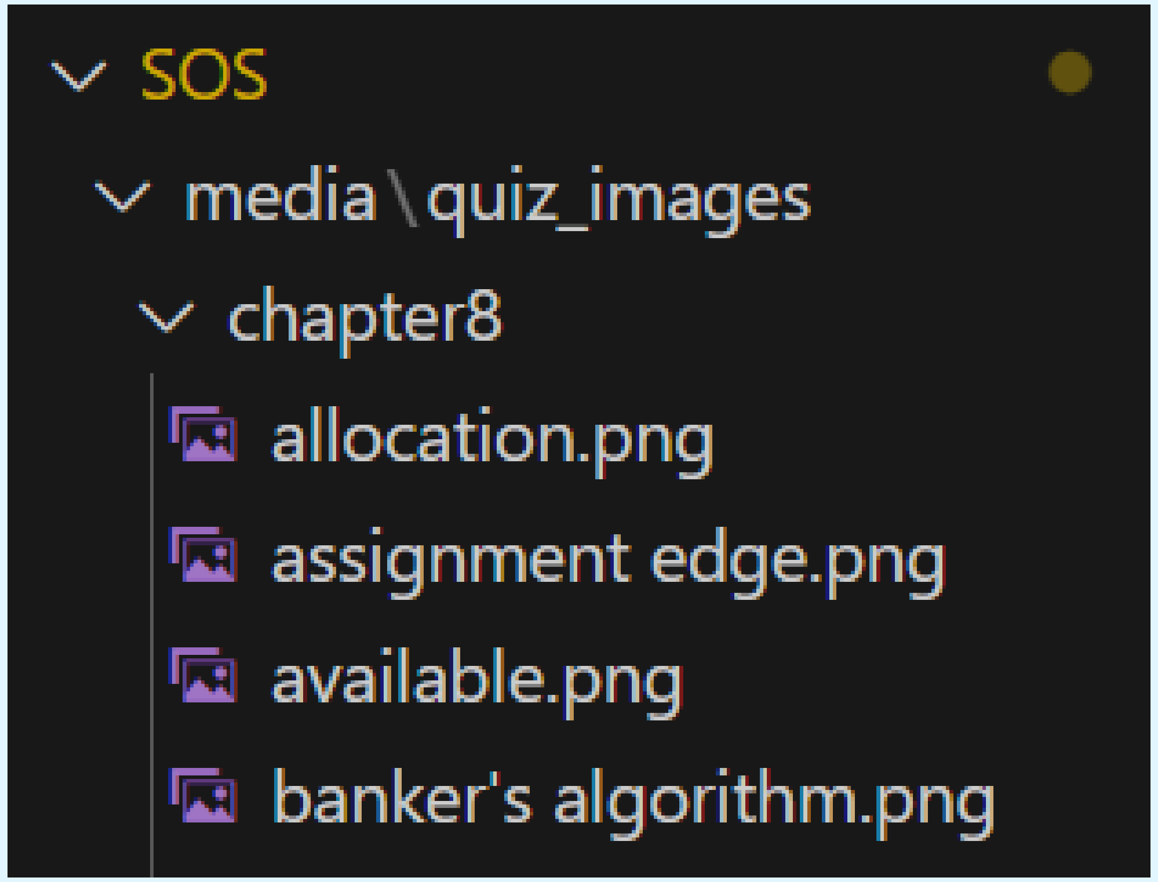
image

answer

8 (int)

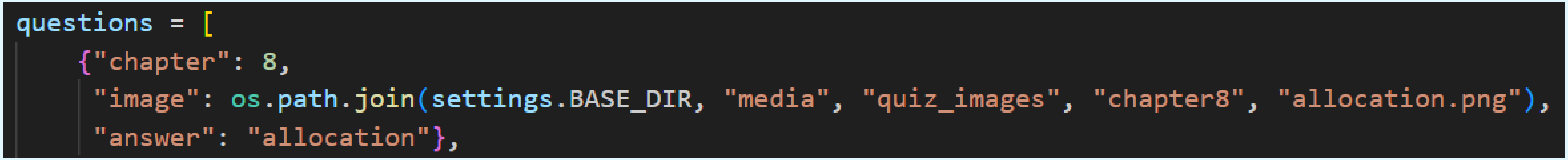
img source address

correct answer (str)

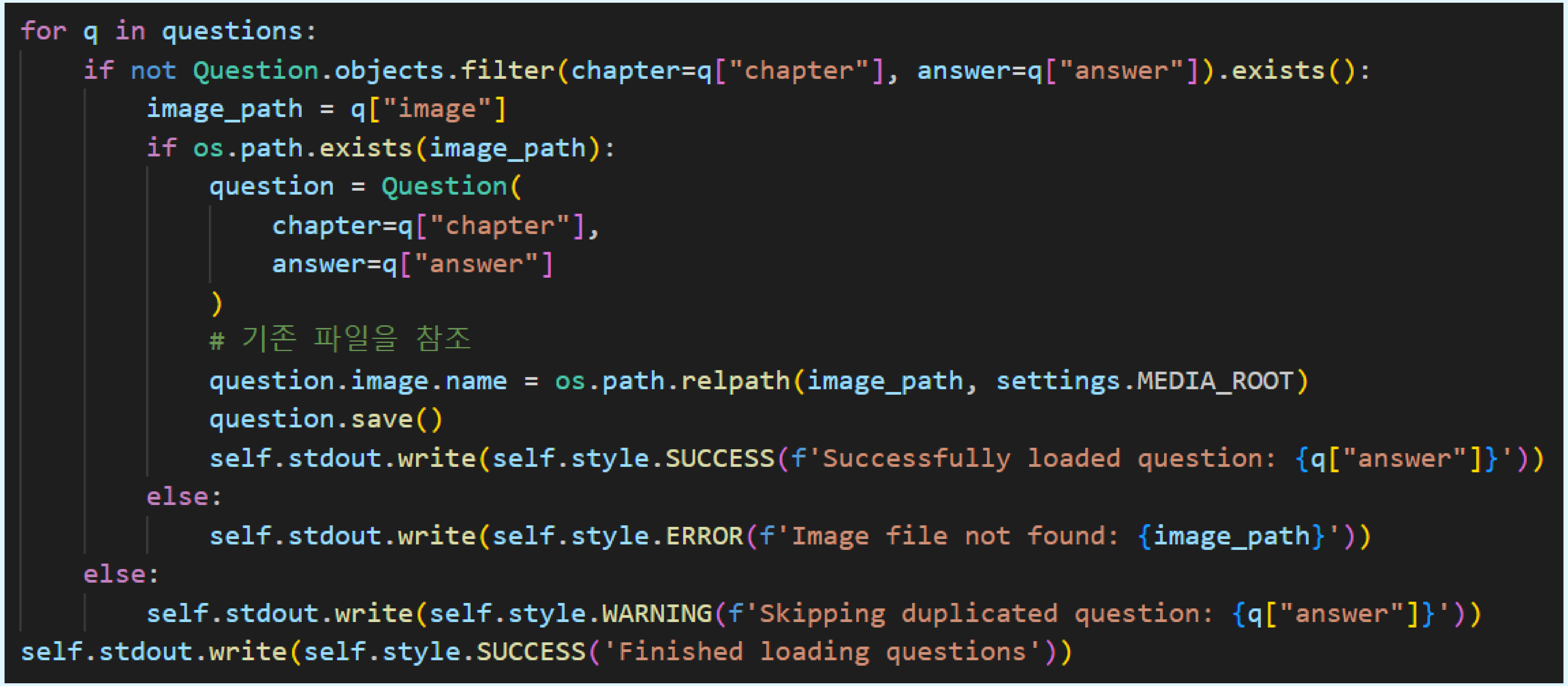


```
∨ SOS
  ∨ media \ quiz_images
    ∨ chapter8
        allocation.png
        assignment edge.png
        available.png
        banker's algorithm.png
```

Load_questoins

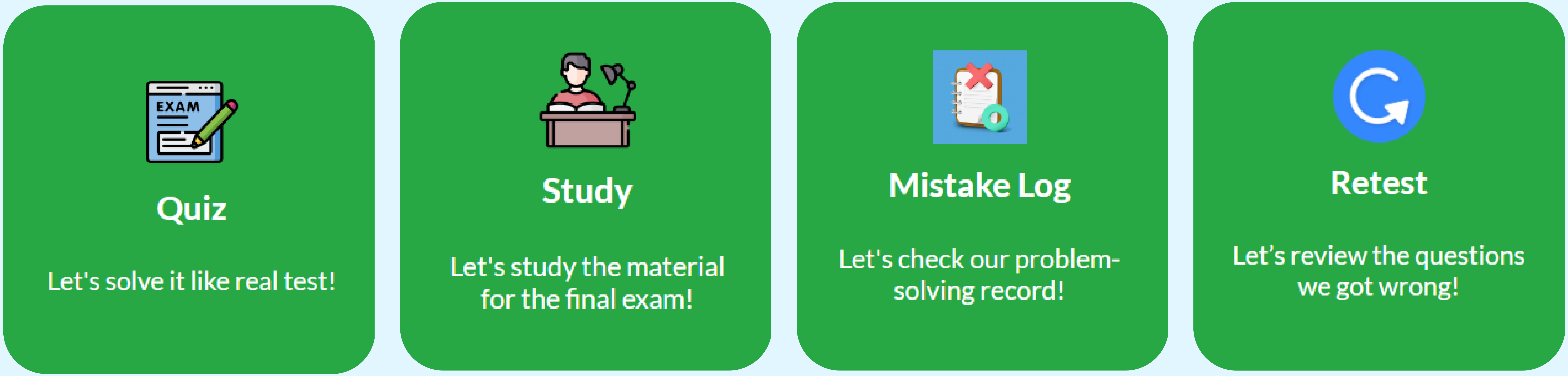


```python
questions = [
    {"chapter": 8,
     "image": os.path.join(settings.BASE_DIR, "media", "quiz_images", "chapter8", "allocation.png"),
     "answer": "allocation"},
```



```python
for q in questions:
    if not Question.objects.filter(chapter=q["chapter"], answer=q["answer"]).exists():
        image_path = q["image"]
        if os.path.exists(image_path):
            question = Question(
                chapter=q["chapter"],
                answer=q["answer"]
            )
            # 기존 파일을 참조
            question.image.name = os.path.relpath(image_path, settings.MEDIA_ROOT)
            question.save()
            self.stdout.write(self.style.SUCCESS(f'Successfully loaded question: {q["answer"]}'))
        else:
            self.stdout.write(self.style.ERROR(f'Image file not found: {image_path}'))
    else:
        self.stdout.write(self.style.WARNING(f'Skipping duplicated question: {q["answer"]}'))
self.stdout.write(self.style.SUCCESS('Finished loading questions'))
```

Views in Django



Quiz

Let's solve it like real test!

Study

Let's study the material for the final exam!

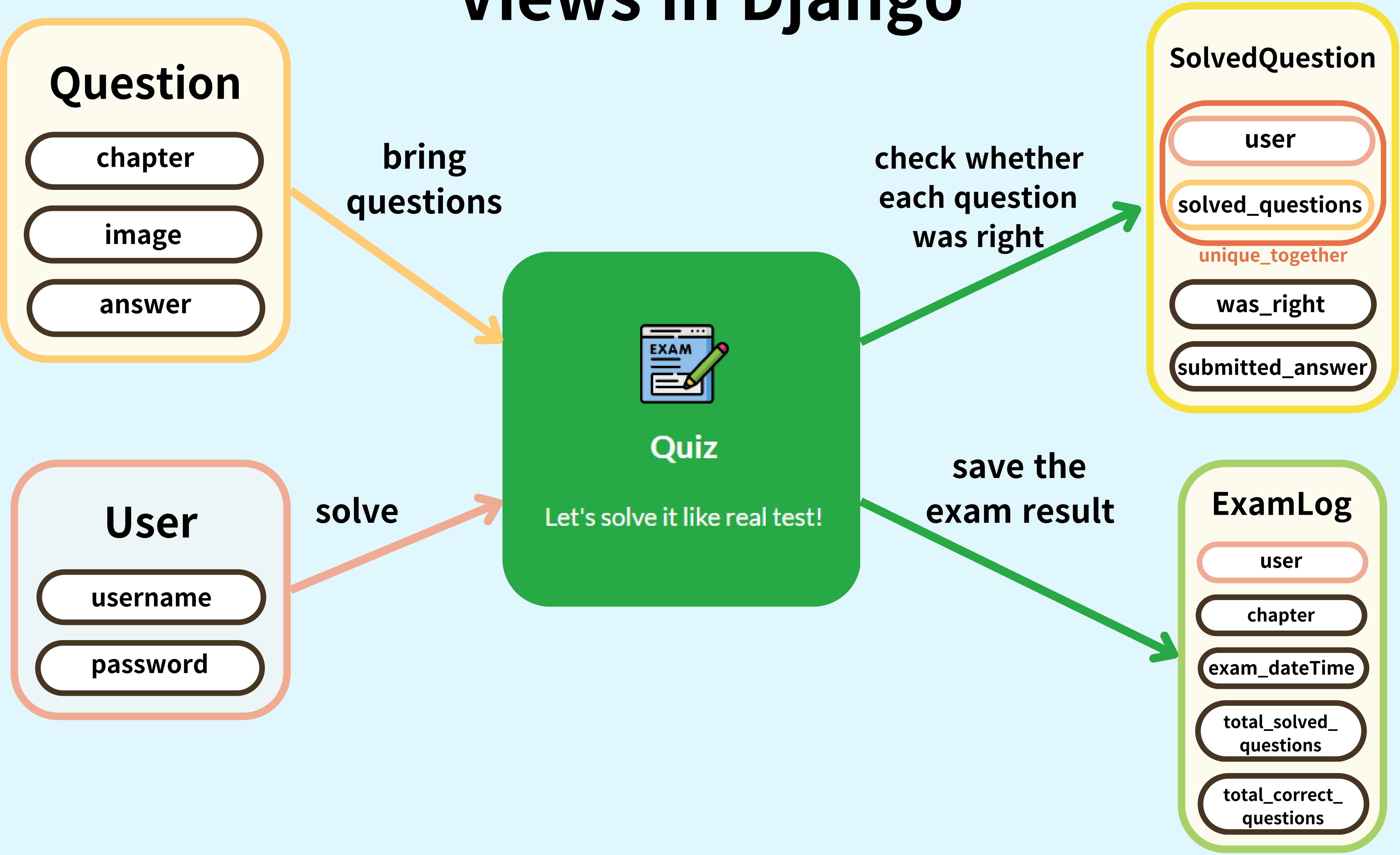
Mistake Log

Let's check our problem-solving record!

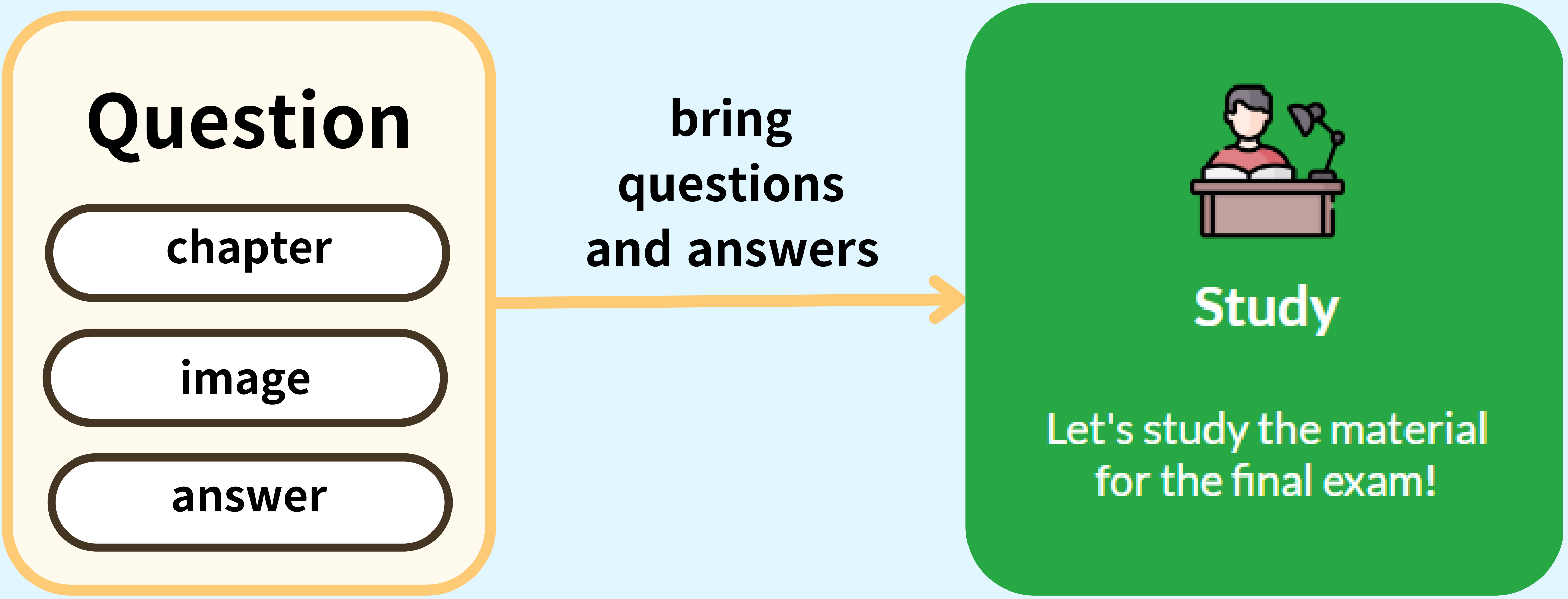
Retest

Let’s review the questions we got wrong!

Views in Django



Views in Django



Question

chapter

image

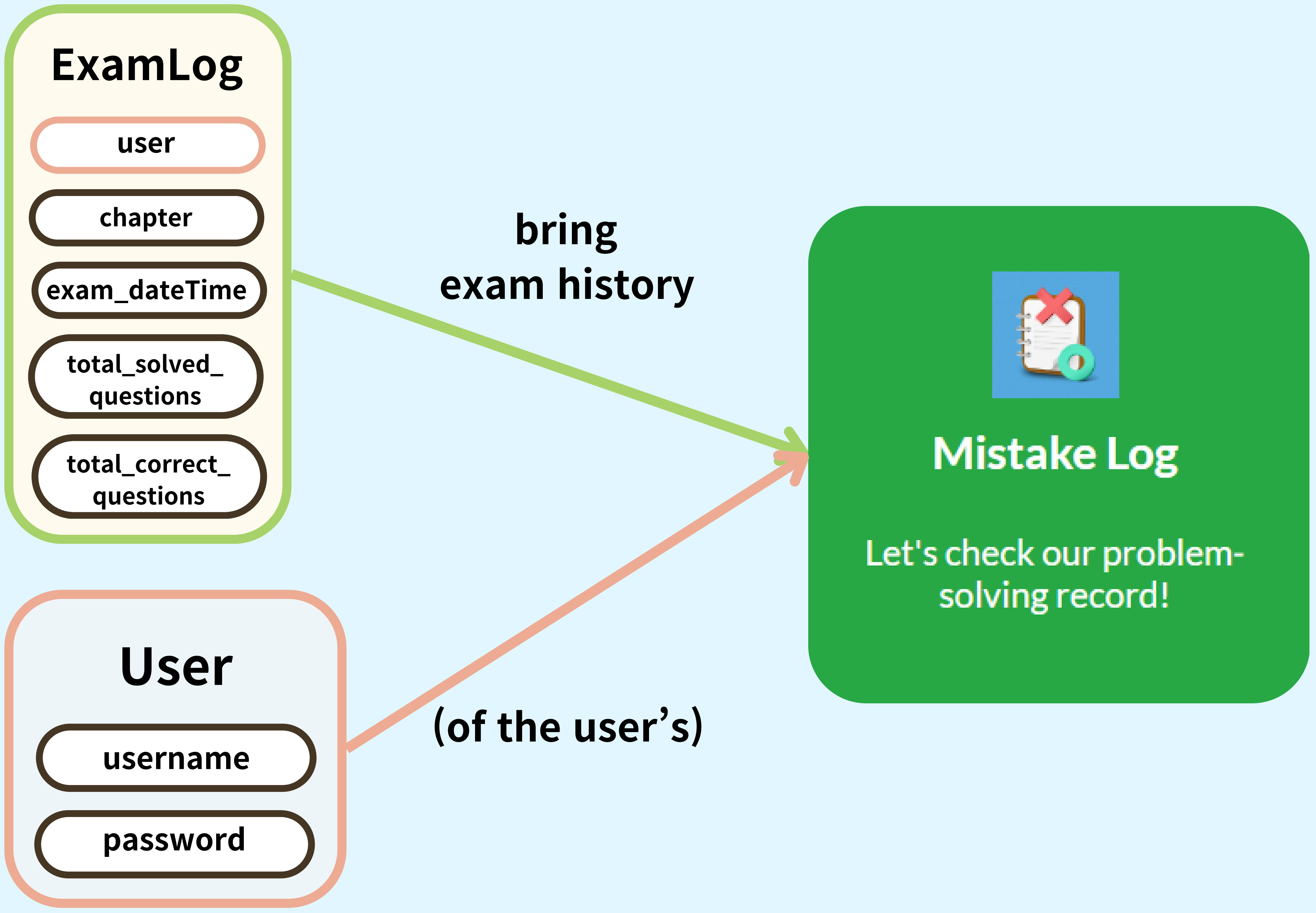
answer

bring questions and answers

Study

Let's study the material for the final exam!

Views in Django



ExamLog

user

chapter

exam_dateTime

total_solved_questions

total_correct_questions

bring exam history

User

username

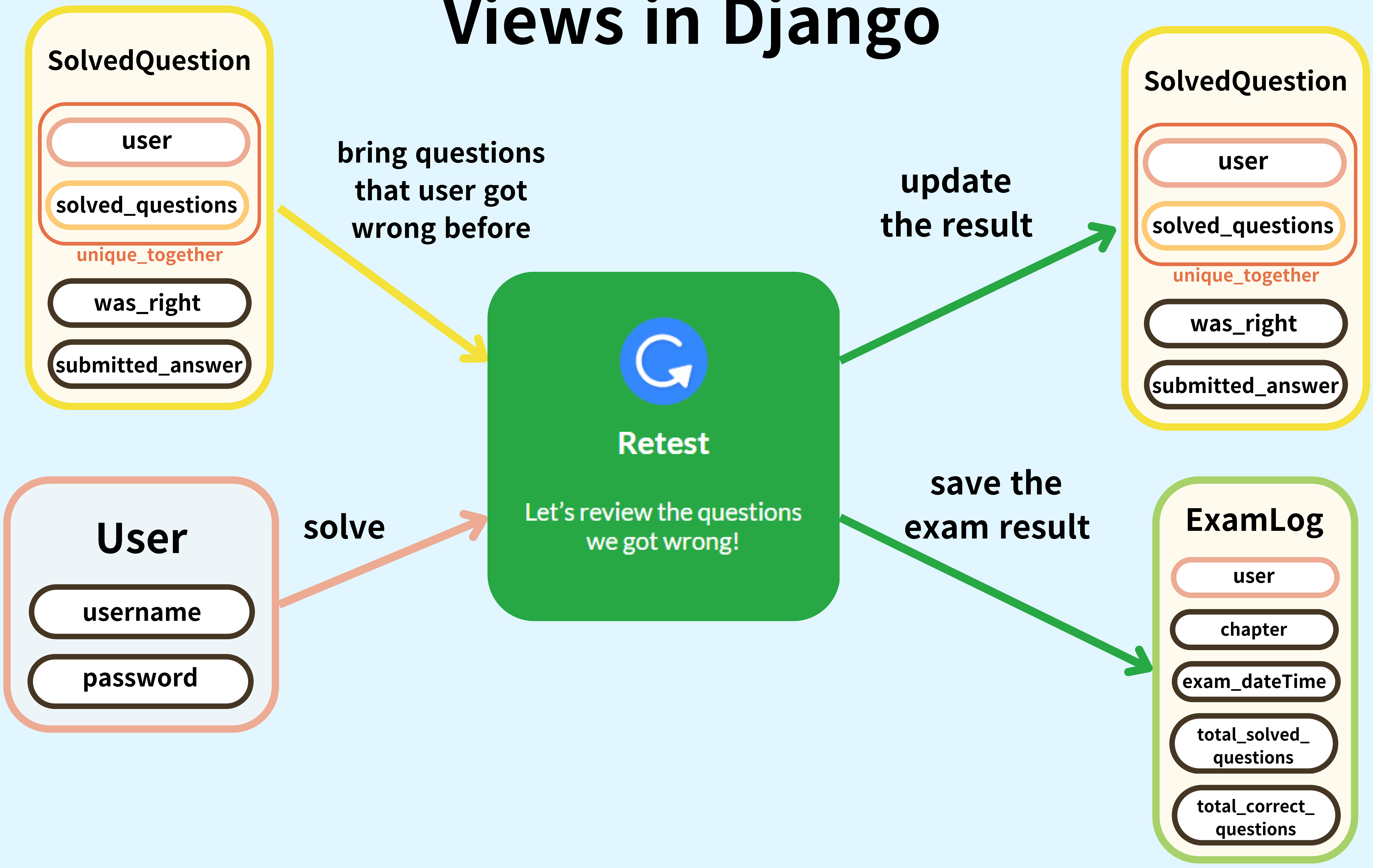
password

(of the user’s)

Mistake Log

Let's check our problem-solving record!

Views in Django



SolvedQuestion

user

solved_questions

unique_together

was_right

submitted_answer

bring questions that user got wrong before

User

username

password

solve

Retest

Let’s review the questions we got wrong!

update the result

SolvedQuestion

user

solved_questions

unique_together

was_right

submitted_answer

save the exam result

ExamLog

user

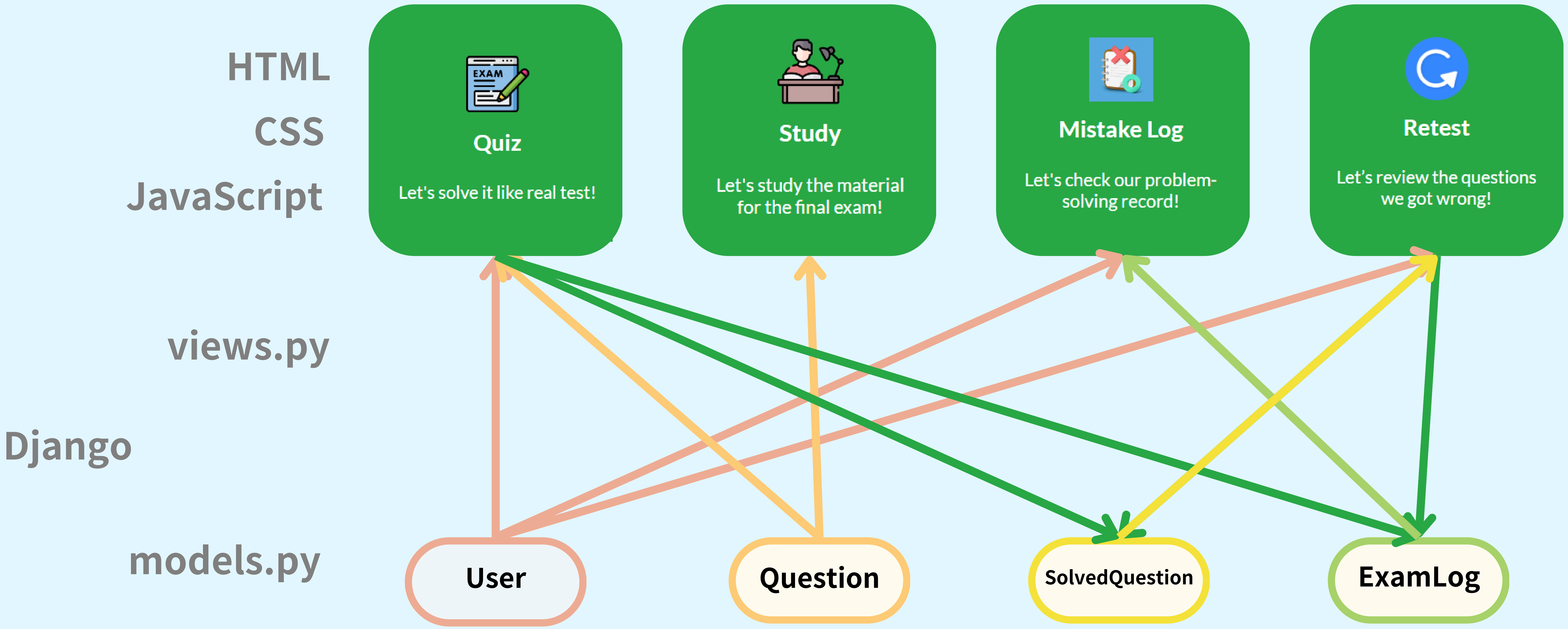
chapter

exam_dateTime

total_solved_questions

total_correct_questions

Django (MTV)



HTML

CSS

JavaScript

Quiz

Let's solve it like real test!

Study

Let's study the material for the final exam!

Mistake Log

Let's check our problem-solving record!

Retest

Let's review the questions we got wrong!

views.py

Django

models.py

User

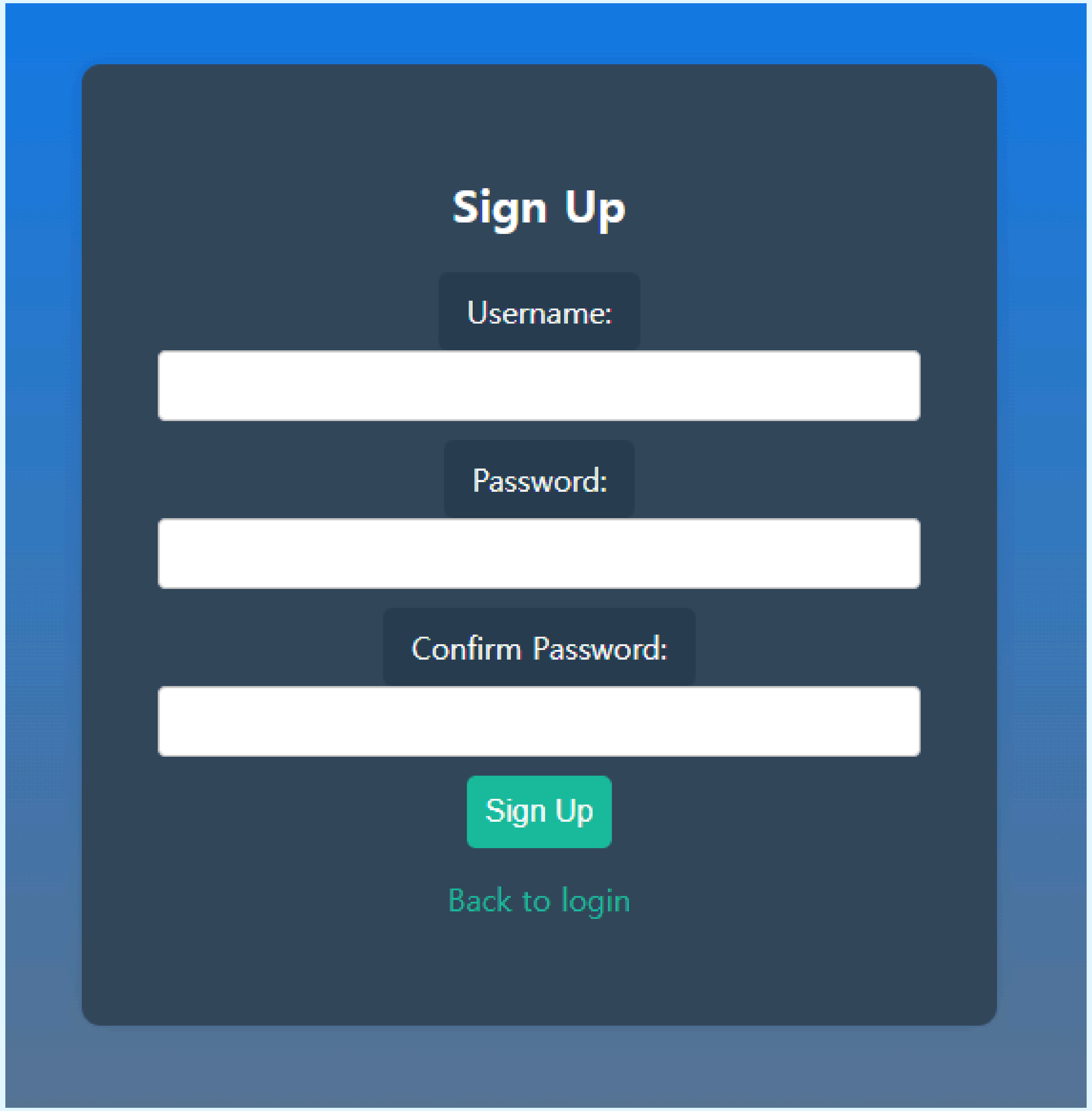
Question

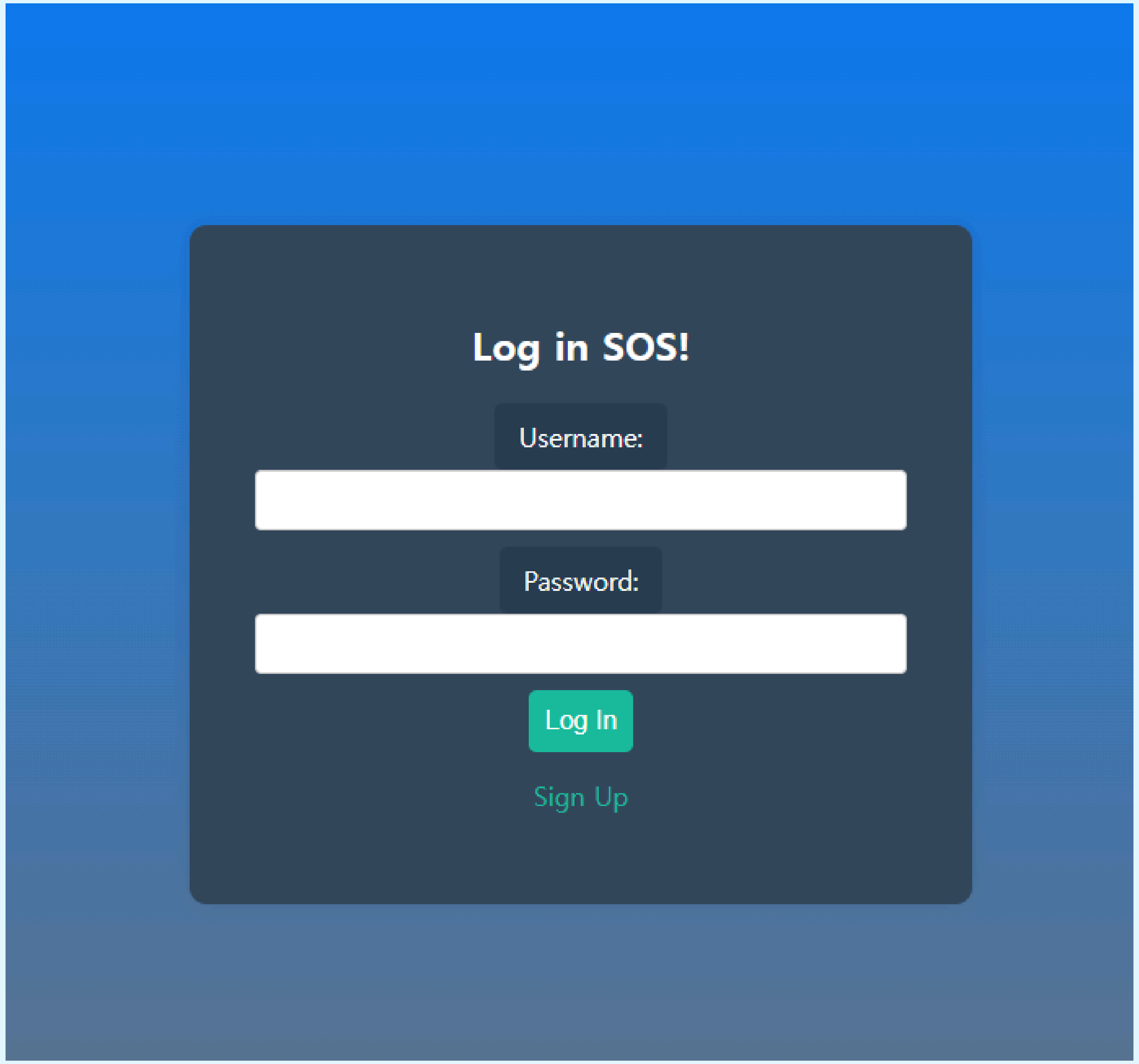
SolvedQuestion

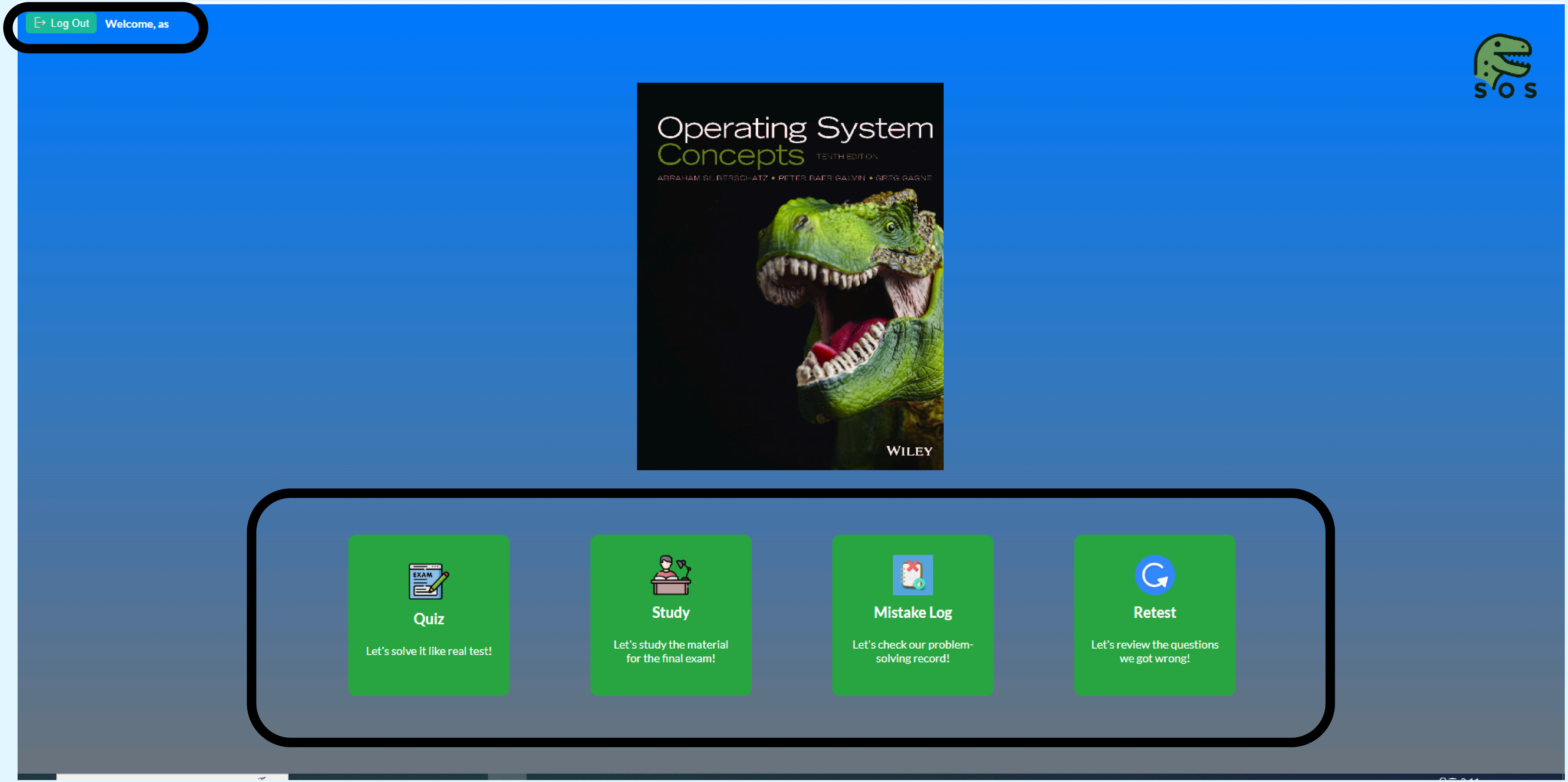
ExamLog

SQLite

Implementation


Sign Up
Username:
Password:
Confirm Password:
Sign Up
Back to login


Log in SOS!
Username:
Password:
Log In
Sign Up


Log Out
Welcome, as
S O S
Operating System Concepts
TENTH EDITION
ABRAHAM SILBERSCHATZ • PETER BAER GALVIN • GREG GAGNE
WILEY
EXAM
Quiz
Study
Mistake Log
Retest

Let's solve it like real test!

Let's study the material for the final exam!

Let's check our problem-solving record!

Let's review the questions we got wrong!

Question

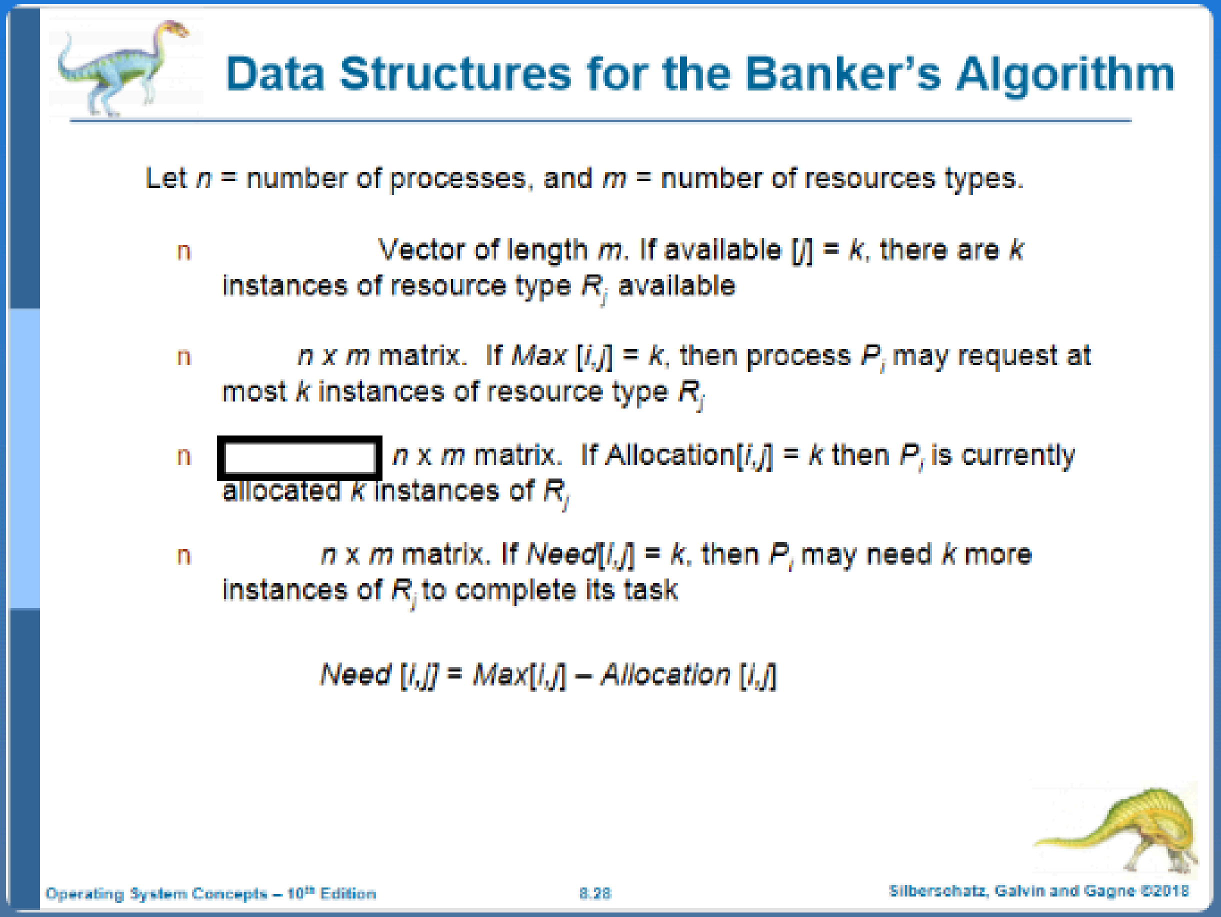
chapter

image

answer

Chapter 8 Study

<Question 1>



Data Structures for the Banker's Algorithm

Let n = number of processes, and m = number of resources types.

n Vector of length m. If available [j] = k, there are k instances of resource type R_j available

n n x m matrix. If *Max* [i,j] = k, then process P_i may request at most k instances of resource type R_j

n □ n x m matrix. If Allocation[i,j] = k then P_i is currently allocated k instances of R_j

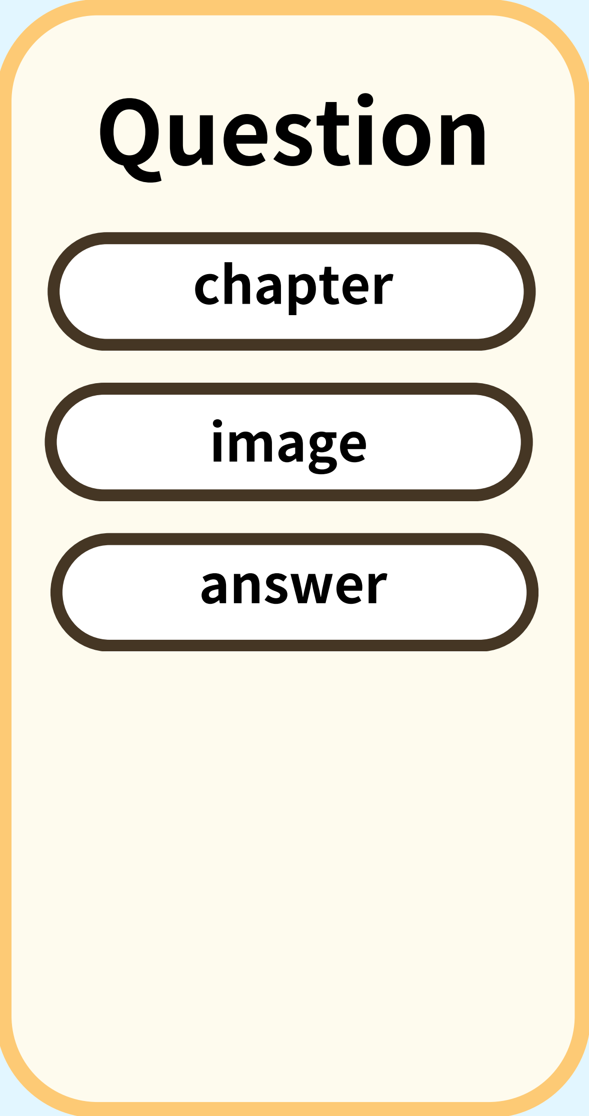
n n x m matrix. If *Need*[i,j] = k, then P_i may need k more instances of R_j to complete its task

Need [i,j] = *Max*[i,j] – *Allocation* [i,j]

Operating System Concepts – 10th Edition 8.28 Silberschatz, Galvin and Gagne ©2018

allocation

⌂ →

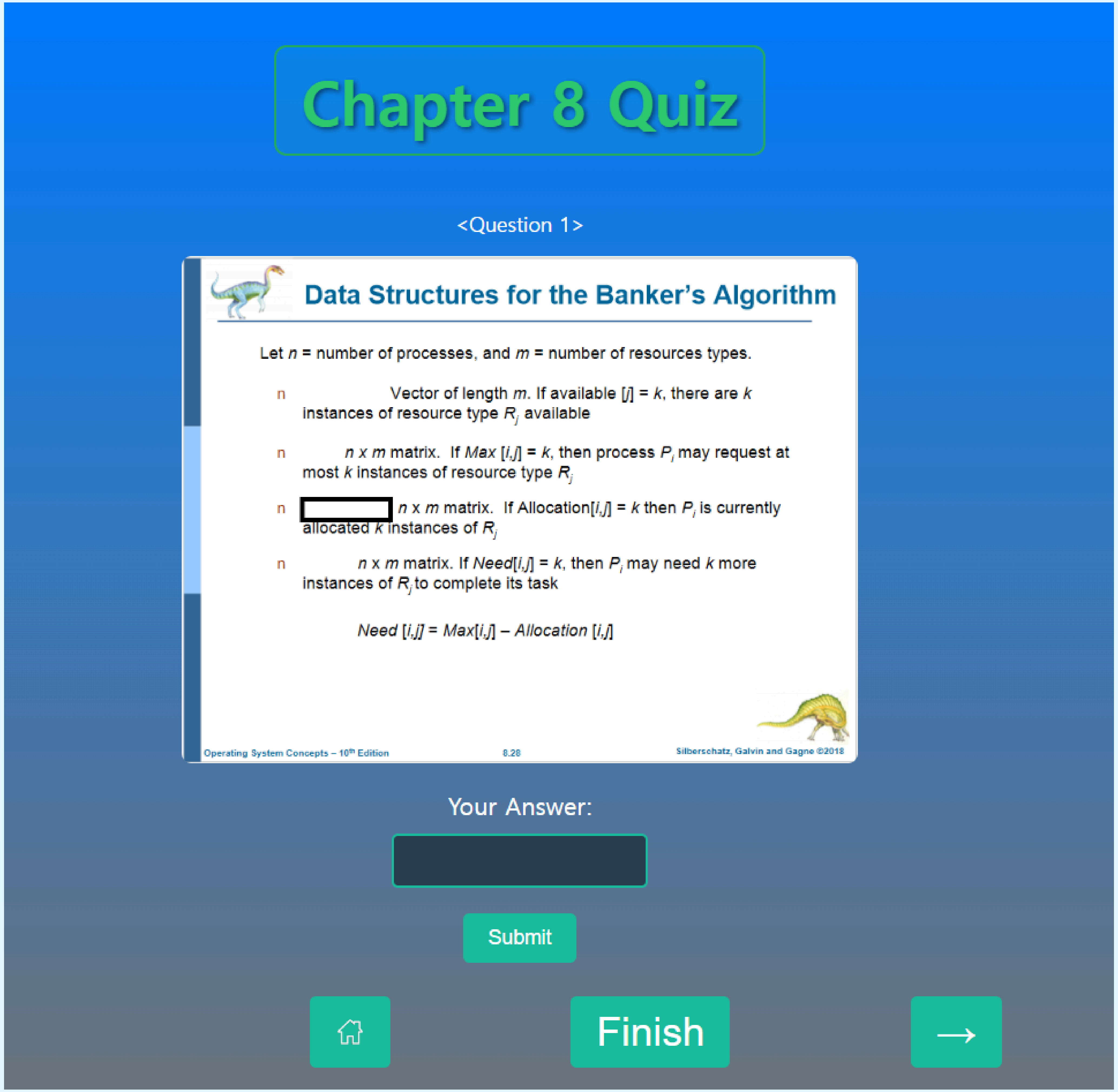


Question

chapter

image

answer



Chapter 8 Quiz

<Question 1>

Data Structures for the Banker's Algorithm

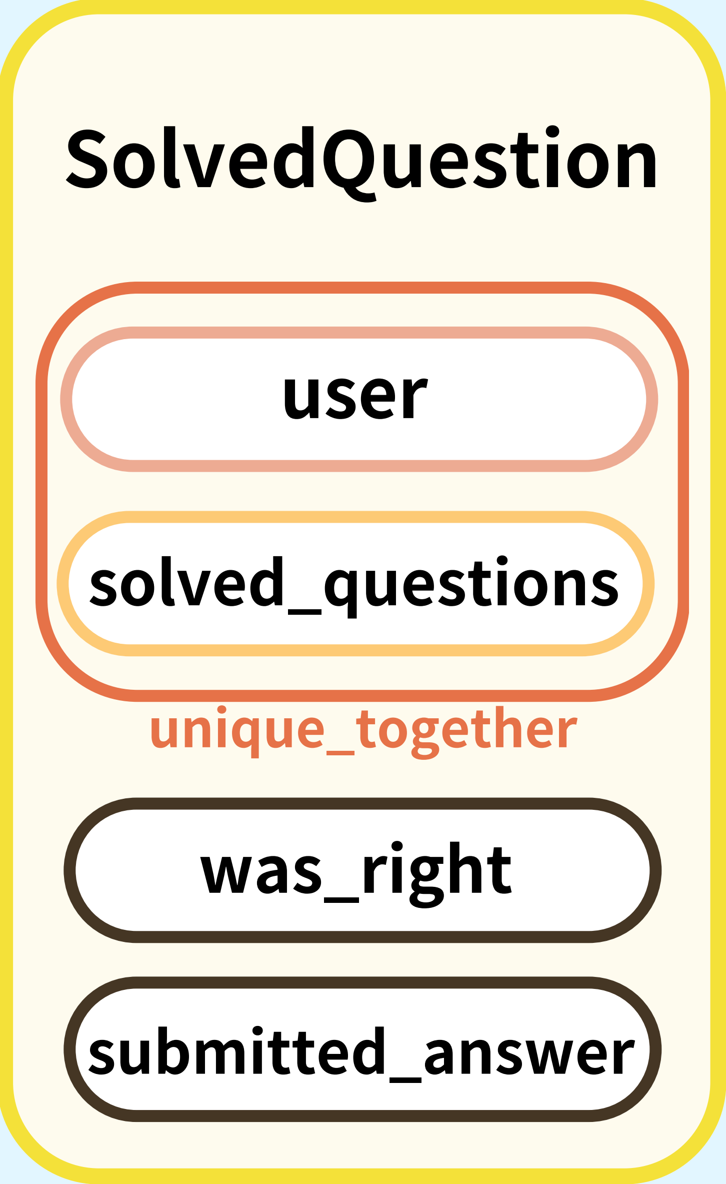
Let *n* = number of processes, and *m* = number of resources types.

- Vector of length *m*. If available [*j*] = *k*, there are *k* instances of resource type R_j available
- *n x m* matrix. If *Max* [*i,j*] = *k*, then process P_i may request at most *k* instances of resource type R_j
- □ *n* x *m* matrix. If Allocation[*i,j*] = *k* then P_i is currently allocated *k* instances of R_j
- *n* x *m* matrix. If *Need*[*i,j*] = *k*, then P_i may need *k* more instances of R_j to complete its task

Need [*i,j*] = *Max*[*i,j*] – *Allocation* [*i,j*]

Operating System Concepts – 10th Edition 8.28 Silberschatz, Galvin and Gagne ©2018

Your Answer:



SolvedQuestion

user

solved_questions

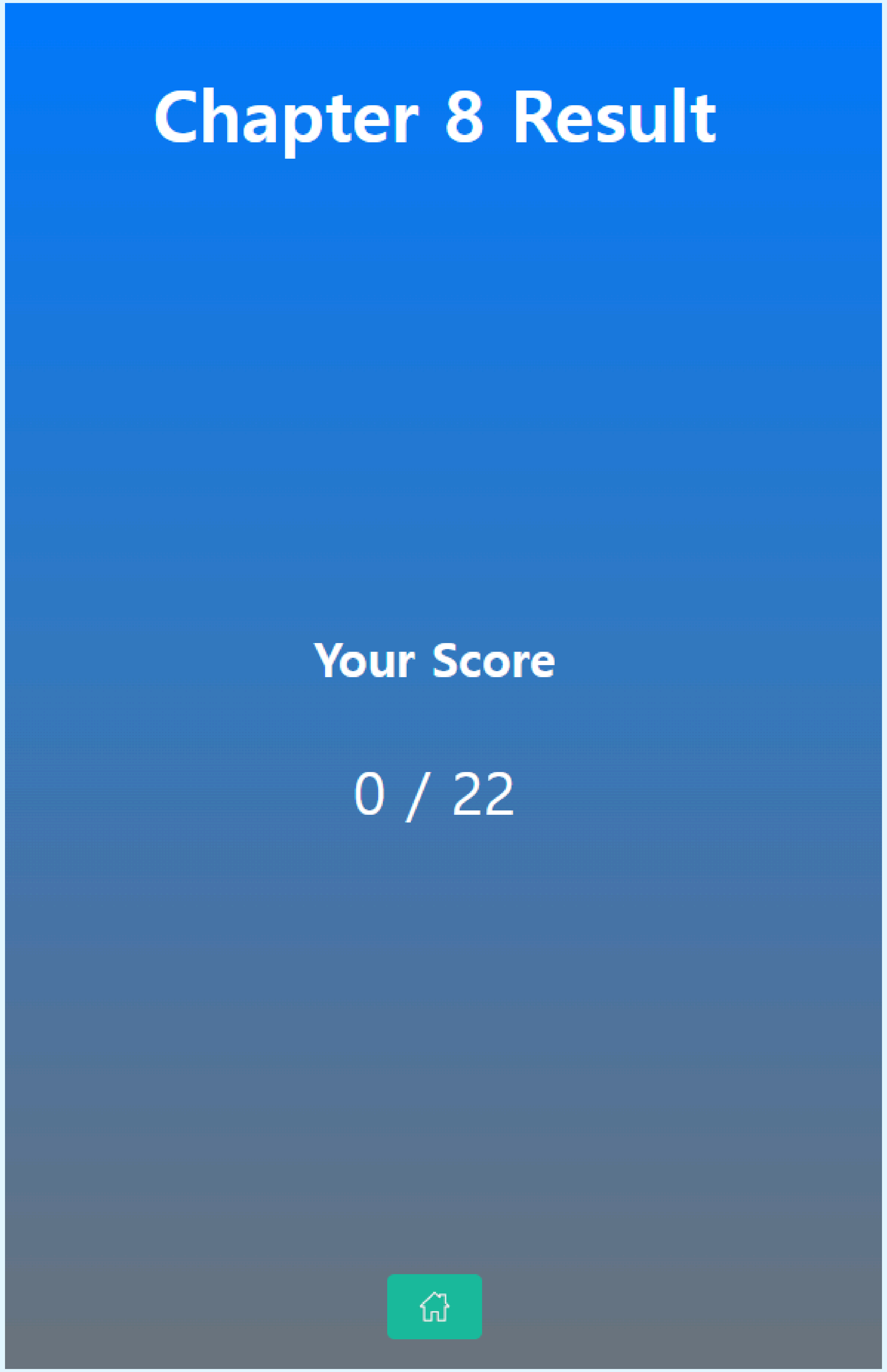
unique_together

was_right

submitted_answer

ExamLog

- user
- chapter
- exam_dateTime
- total_solved_questions
- total_correct_questions



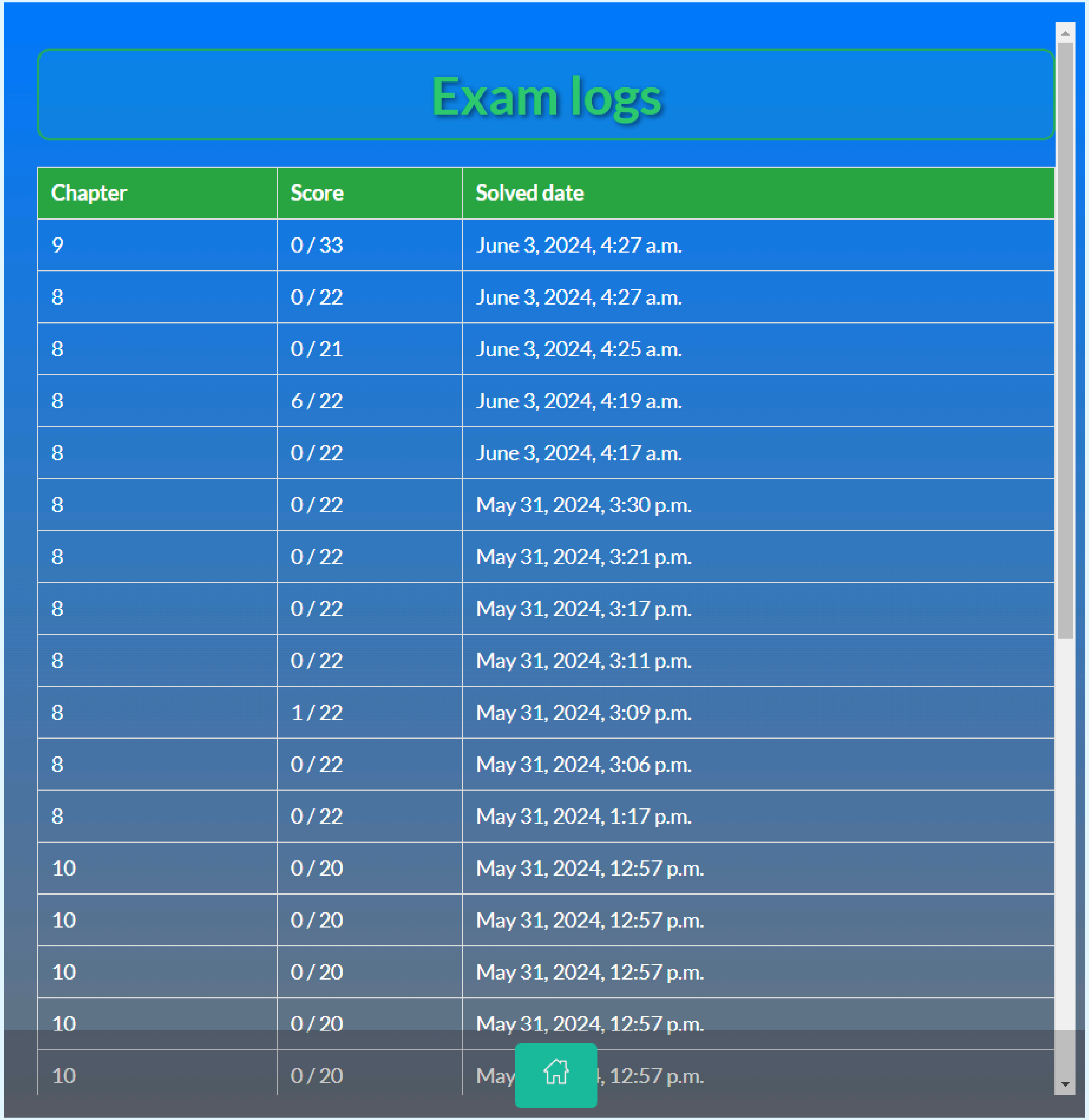
Chapter 8 Result

Your Score

0 / 22

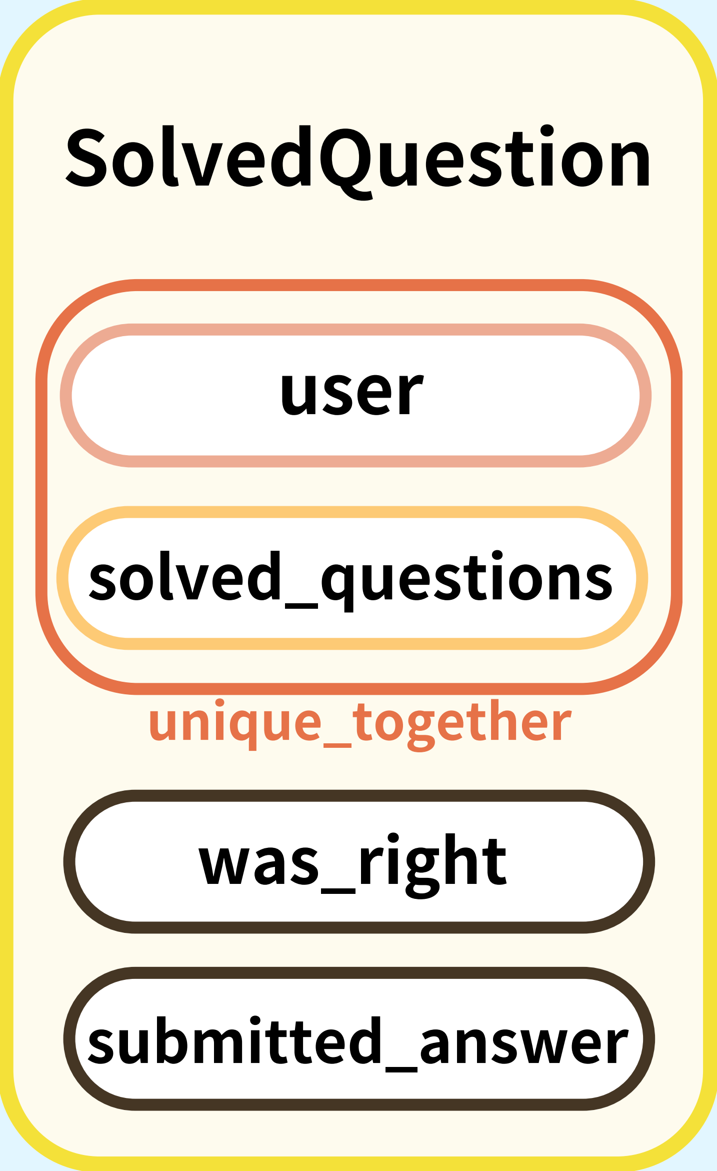
ExamLog

- user
- chapter
- exam_dateTime
- total_solved_questions
- total_correct_questions



Exam logs

Chapter	Score	Solved date
9	0 / 33	June 3, 2024, 4:27 a.m.
8	0 / 22	June 3, 2024, 4:27 a.m.
8	0 / 21	June 3, 2024, 4:25 a.m.
8	6 / 22	June 3, 2024, 4:19 a.m.
8	0 / 22	June 3, 2024, 4:17 a.m.
8	0 / 22	May 31, 2024, 3:30 p.m.
8	0 / 22	May 31, 2024, 3:21 p.m.
8	0 / 22	May 31, 2024, 3:17 p.m.
8	0 / 22	May 31, 2024, 3:11 p.m.
8	1 / 22	May 31, 2024, 3:09 p.m.
8	0 / 22	May 31, 2024, 3:06 p.m.
8	0 / 22	May 31, 2024, 1:17 p.m.
10	0 / 20	May 31, 2024, 12:57 p.m.
10	0 / 20	May 31, 2024, 12:57 p.m.
10	0 / 20	May 31, 2024, 12:57 p.m.
10	0 / 20	May 31, 2024, 12:57 p.m.
10	0 / 20	May , 12:57 p.m.



SolvedQuestion

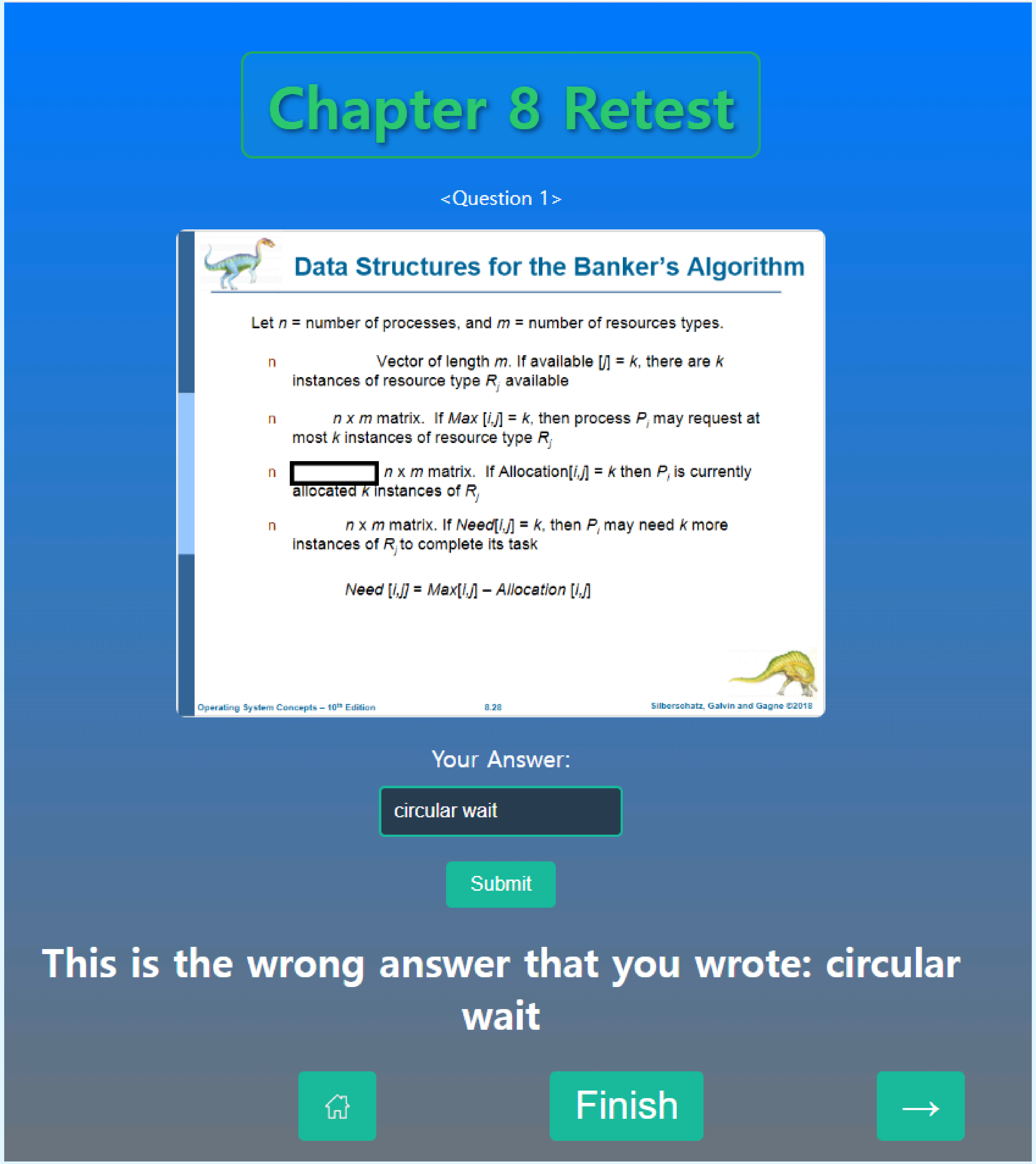
user

solved_questions

unique_together

was_right

submitted_answer



Chapter 8 Retest

<Question 1>

Data Structures for the Banker's Algorithm

Let n = number of processes, and m = number of resources types.

n Vector of length m. If available $[j] = k$, there are k instances of resource type R_j available

n $n \times m$ matrix. If $Max\ [i,j] = k$, then process P_i may request at most k instances of resource type R_j

n □ $n \times m$ matrix. If Allocation$[i,j] = k$ then P_i is currently allocated k instances of R_j

n $n \times m$ matrix. If $Need[i,j] = k$, then P_i may need k more instances of R_j to complete its task

$$Need\ [i,j] = Max[i,j] – Allocation\ [i,j]$$

Operating System Concepts – 10th Edition 8.28 Silberschatz, Galvin and Gagne ©2018

Your Answer:

circular wait

This is the wrong answer that you wrote: circular wait yml

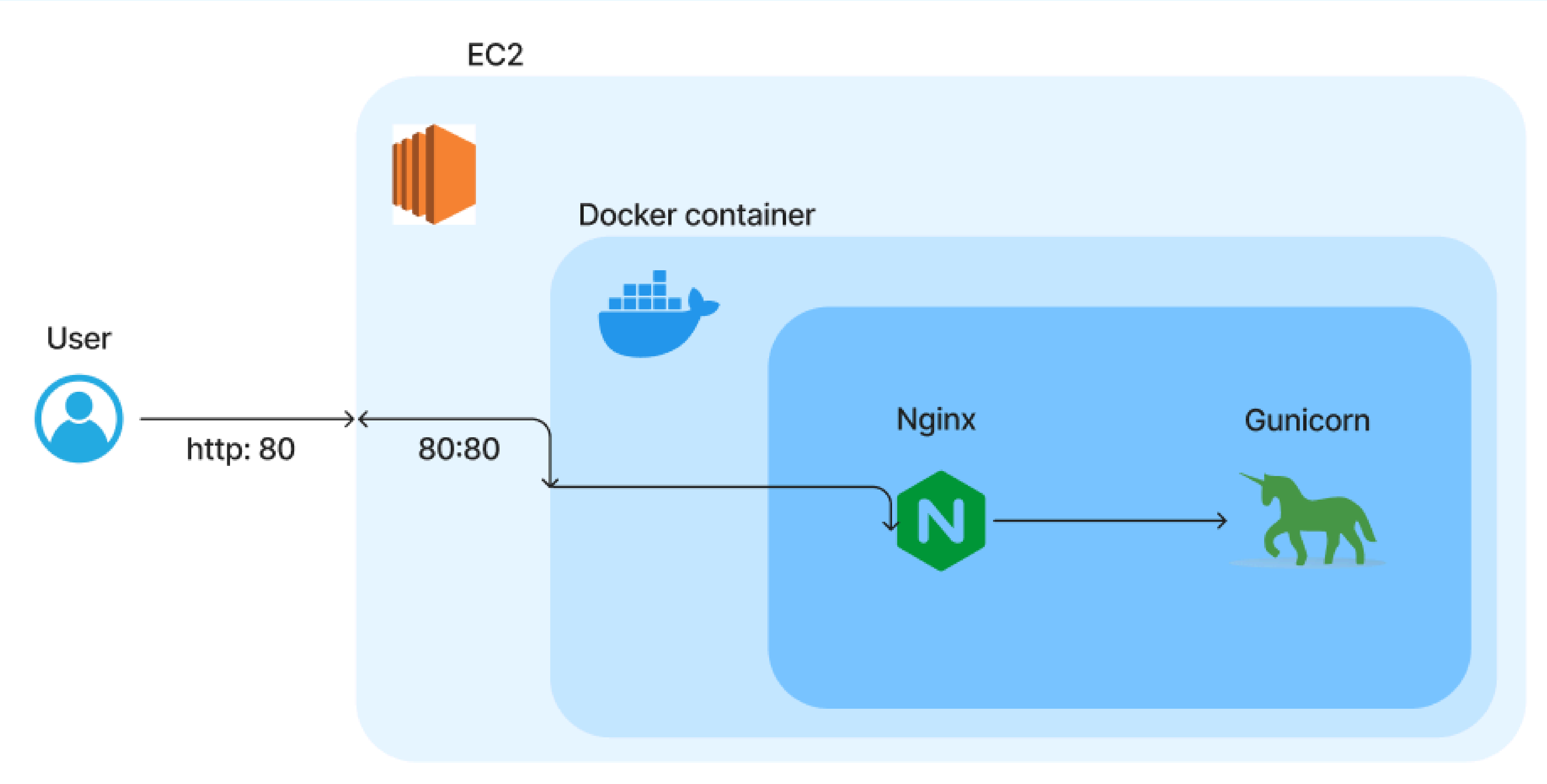
Event
->Runner
->Workflow(Job, Action)

Dockerfile

build
->push
->pull(ec2 server)

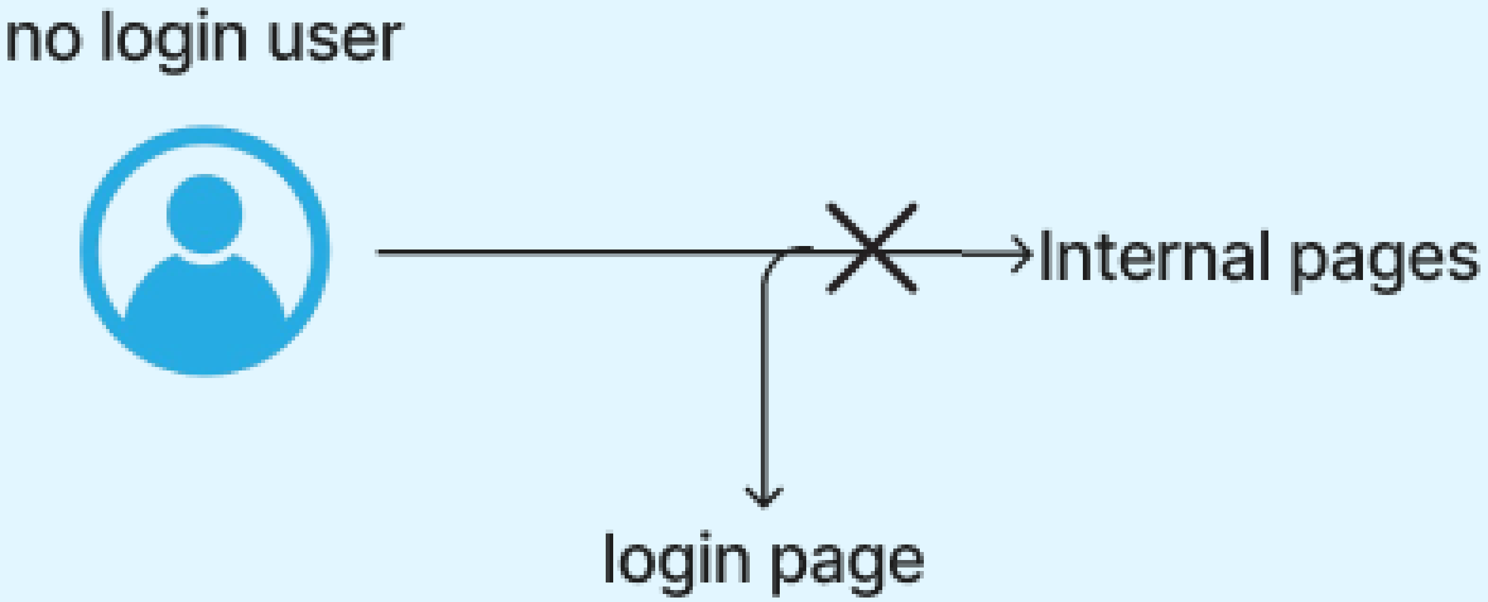
nginx.conf

serve static&media file


EC2
Docker container
User
http: 80
80:80
Nginx
Gunicorn

Problem Analysis and Debugging

Problem 1: User Validation

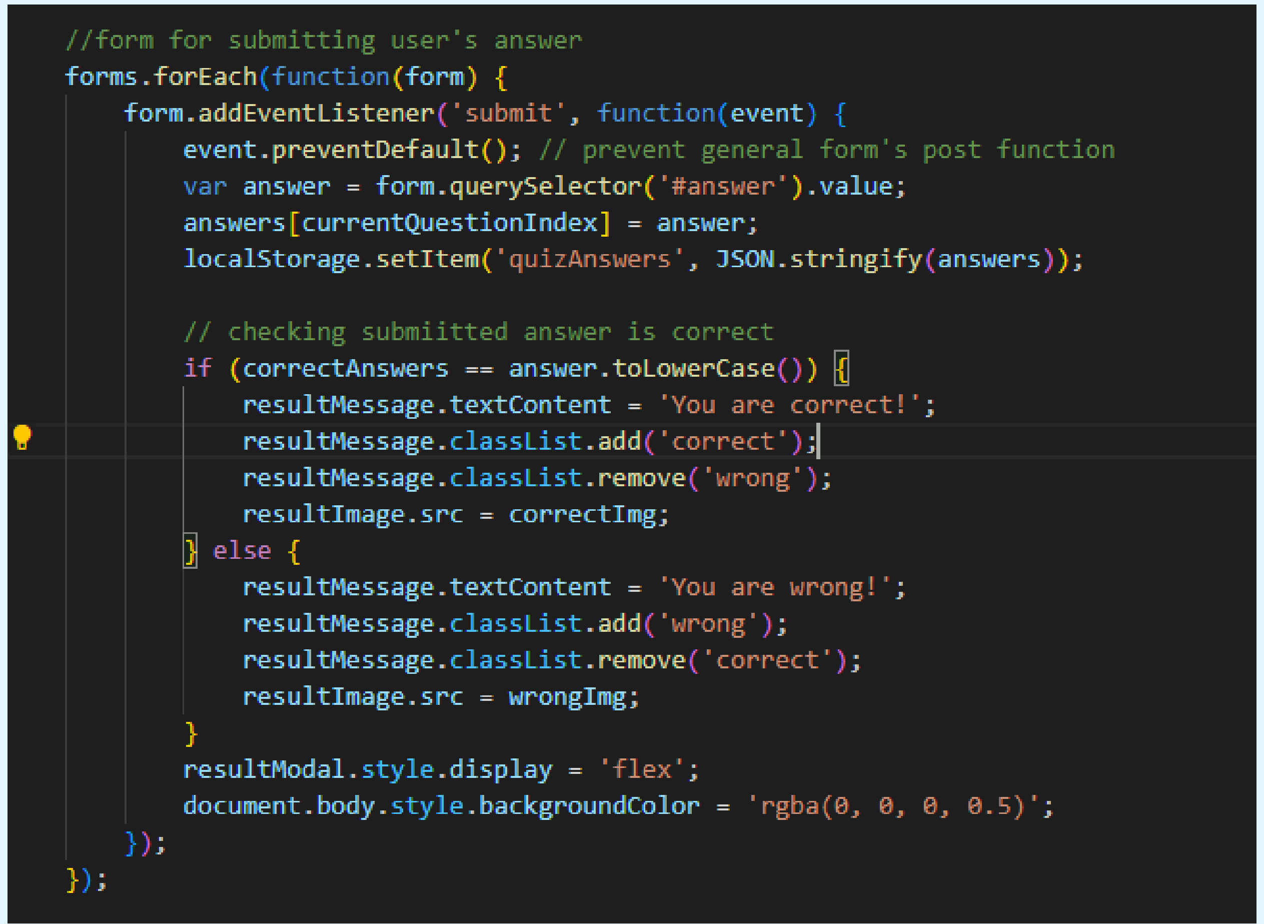


no login user

Internal pages

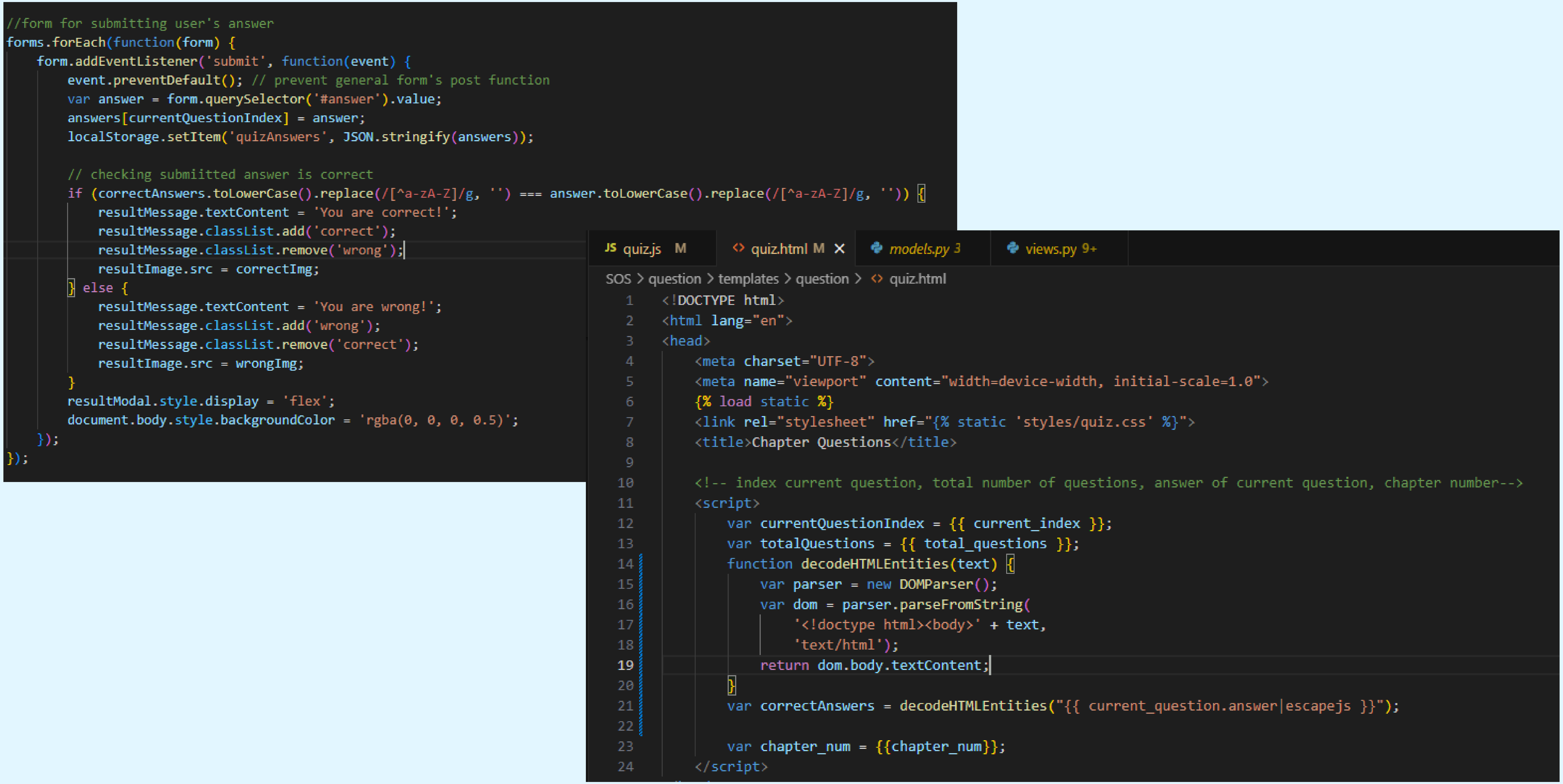
login page

Problem 2: Answer contains a special symbol



```
//form for submitting user's answer
forms.forEach(function(form) {
    form.addEventListener('submit', function(event) {
        event.preventDefault(); // prevent general form's post function
        var answer = form.querySelector('#answer').value;
        answers[currentQuestionIndex] = answer;
        localStorage.setItem('quizAnswers', JSON.stringify(answers));

        // checking submiitted answer is correct
        if (correctAnswers == answer.toLowerCase()) {
            resultMessage.textContent = 'You are correct!';
            resultMessage.classList.add('correct');
            resultMessage.classList.remove('wrong');
            resultImage.src = correctImg;
        } else {
            resultMessage.textContent = 'You are wrong!';
            resultMessage.classList.add('wrong');
            resultMessage.classList.remove('correct');
            resultImage.src = wrongImg;
        }
        resultModal.style.display = 'flex';
        document.body.style.backgroundColor = 'rgba(0, 0, 0, 0.5)';
    });
});
```

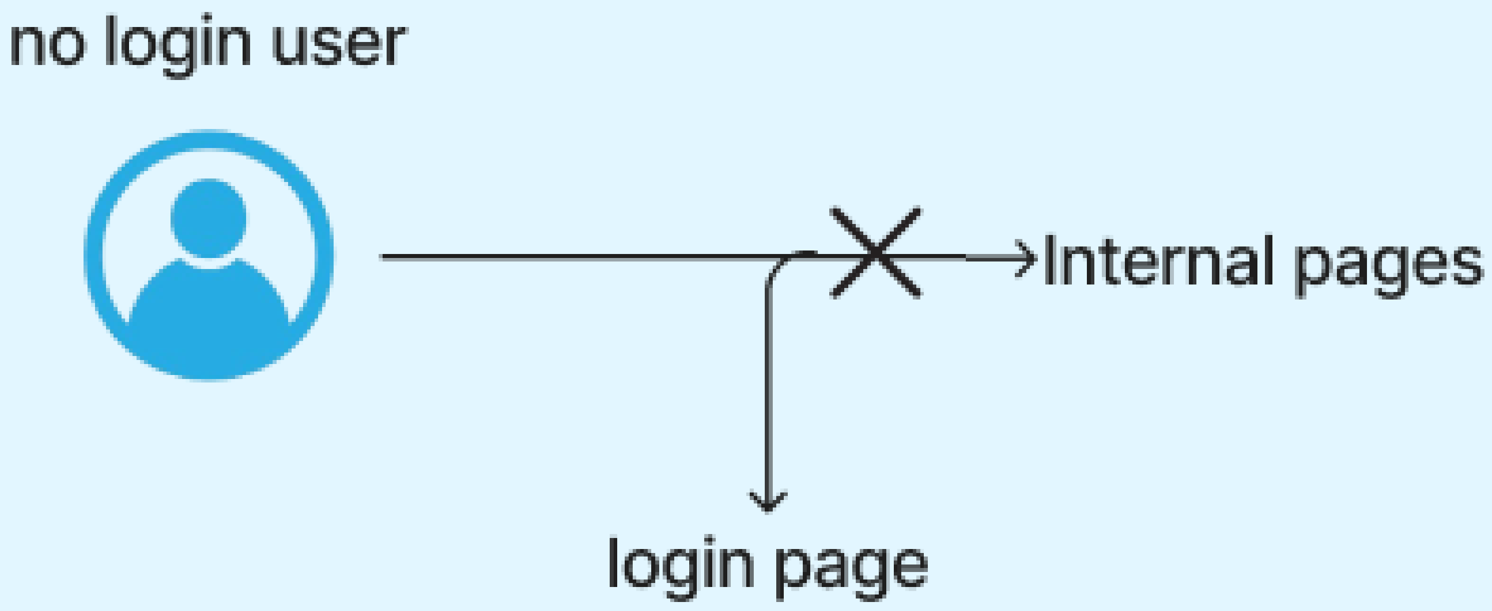
Problem 2: Answer contains a special symbol



```
//form for submitting user's answer
forms.forEach(function(form) {
    form.addEventListener('submit', function(event) {
        event.preventDefault(); // prevent general form's post function
        var answer = form.querySelector('#answer').value;
        answers[currentQuestionIndex] = answer;
        localStorage.setItem('quizAnswers', JSON.stringify(answers));

        // checking submitted answer is correct
        if (correctAnswers.toLowerCase().replace(/[^a-zA-Z]/g, '') === answer.toLowerCase().replace(/[^a-zA-Z]/g, '')) {
            resultMessage.textContent = 'You are correct!';
            resultMessage.classList.add('correct');
            resultMessage.classList.remove('wrong');
            resultImage.src = correctImg;
        } else {
            resultMessage.textContent = 'You are wrong!';
            resultMessage.classList.add('wrong');
            resultMessage.classList.remove('correct');
            resultImage.src = wrongImg;
        }
        resultModal.style.display = 'flex';
        document.body.style.backgroundColor = 'rgba(0, 0, 0, 0.5)';
    });
});
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    {% load static %}
    <link rel="stylesheet" href="{% static 'styles/quiz.css' %}">
    <title>Chapter Questions</title>

    <!-- index current question, total number of questions, answer of current question, chapter number-->
    <script>
        var currentQuestionIndex = {{ current_index }};
        var totalQuestions = {{ total_questions }};
        function decodeHTMLEntities(text) {
            var parser = new DOMParser();
            var dom = parser.parseFromString(
                '<!doctype html><body>' + text,
                'text/html');
            return dom.body.textContent;
        }
        var correctAnswers = decodeHTMLEntities("{{ current_question.answer|escapejs }}");

        var chapter_num = {{chapter_num}};
    </script>
```

Problem 1: User Validation



Problem 2: Answer contains a special symbol

Result

SOS