

BÁO CÁO ĐỒ ÁN MÔN HỌC

Lớp: IT003.CTTN.P21

Thông tin sinh viên

- Mã số sinh viên: **24520027**
- Họ và tên: **Lê Thiên Quân**
- Tên đề tài: **ICPC Assistant – Ứng dụng hỗ trợ thống kê và huấn luyện đội tuyển ICPC**

1. Giới thiệu đề tài

Đồ án xây dựng một ứng dụng web hỗ trợ đội tuyển ICPC trong việc quản lý, thống kê và phân tích dữ liệu luyện tập. Hệ thống sử dụng FastAPI (Python) làm backend và HTML, CSS, JavaScript cho phần frontend. Dữ liệu được thu thập tự động từ Codeforces thông qua API chính thức.

2. Quá trình thực hiện

Tuần 1

- Tìm hiểu kiến trúc FastAPI.
- Học cách tích hợp FastAPI với HTML/CSS/JS.

Tuần 2

- Thiết kế giao diện cơ bản.
- Viết các hàm truy vấn API từ Codeforces.

Tuần 3

- Sử dụng SQLite để lưu trữ dữ liệu tạm.
- Tối ưu quá trình gọi API, tránh trùng lặp.

Tuần 4

- Xây dựng chức năng thống kê: số bài đã giải, biểu đồ tiến bộ, xếp hạng.
- Cải thiện giao diện, bố cục dữ liệu.

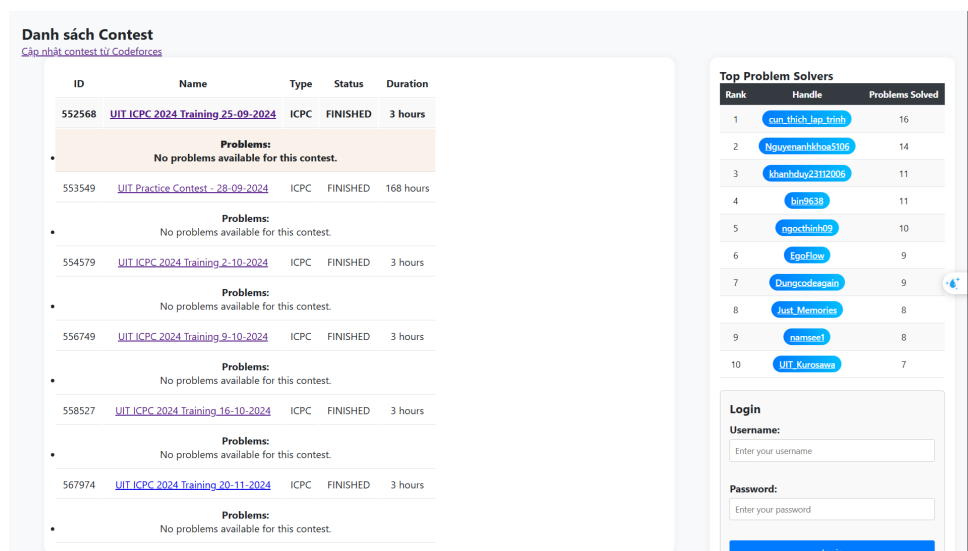
3. Kết quả đạt được

- Hoàn thành hệ thống cơ bản:
 - Kết nối Codeforces API.
 - Lưu trữ dữ liệu cục bộ.
 - Thống kê bài tập, biểu đồ xếp hạng.
- Giao diện trực quan, dễ sử dụng.
- Hạn chế: tính năng chatbot AI hỗ trợ luyện tập vẫn chưa hoàn thiện.

4. Tài liệu tham khảo

1. FastAPI Documentation – <https://fastapi.tiangolo.com>
2. Codeforces API – <https://codeforces.com/apiHelp>
3. SQLite Documentation – <https://sqlite.org>
4. MDN Web Docs – HTML, CSS, JS – <https://developer.mozilla.org>

Phụ lục 1: Demo giao diện



Hình 1: Giao diện chính của trang web

Standings for UIT ICPC 2024 Training 25-09-2024

Rank	Handle	Problems Solved
1	cun_thich_lap_trinh	6
2	khanhduy23112006	5
3	ThucLV	5
4	anhzzz	5
5	Just_Memories	5
6	anlanhounoofficial	5
7	biep638	5
8	UIT_Kurosawa	4
9	Giangcoder	4
10	omlgg	4
11	namsee1	4
12	nghiahom9999	4
13	HoangViet	4
14	64queto	4
15	EgoFlow	4
16	logm	4
17	DOANHONGBAO	4
18	babfish23	4
19	LittleNoPro	3
20	ahzzz	3
21	MrRaymond	3
22	louis97	3
23	ComTamNgoQuyen	3
24	BananaBread	3
25	urnqt	3
26	TNT_NHAT_VOI	3
27	UIT.NguyenThaiSon	3
28	Dungcodeagain	3
29	Nguyenanhkhoa5106	3
30	anhphaut	2
31	Kiffnz	2
32	ch1_kier	2

Hình 2: Giao diện của bảng xếp hạng mỗi contest



Hình 3: Biểu đồ phân tích của mỗi thành viên

Phụ lục 2: Trích xuất Docstring (ví dụ)

File: `.\login.py`

Loại: function

Tên: `authenticate_user`

Docstring:

Authenticate a user by username and password.

Args:

`username (str)`: The username to authenticate.

`password (str)`: The plain text password to verify.

Returns:

dict or None: The user dictionary if authentication is successful, otherwise None.

=====

File: .\login.py
Loại: function
Tên: create_access_token
Docstring:
Create a JWT access token.

Args:
data (dict): The data to encode in the token (e.g., user info).
expires_delta (timedelta, optional): The time until the token expires. Defaults to 15

Returns:
str: The encoded JWT token as a string.

=====

File: .\login.py
Loại: function
Tên: login_page
Docstring:
Render the login page.

Args:
request (Request): The incoming HTTP request.

Returns:
TemplateResponse: The rendered login.html template.

=====

File: .\login.py
Loại: function
Tên: login
Docstring:
Authenticate user and return an access token.

Args:
form_data (OAuth2PasswordRequestForm): The form data containing username and password.

Raises:
HTTPException: If authentication fails.

Returns:
dict: A dictionary containing the access token and token type.

=====

File: .\login.py
Loại: function
Tên: get_current_user
Docstring:
Retrieve the current user from the JWT token.

Args:

token (str): The JWT token from the Authorization header.

Raises:

HTTPException: If the token is invalid or missing required fields.

Returns:

dict: A dictionary containing the username and role of the user.

=====

File: .\login.py

Loại: function

Tên: admin_required

Docstring:

Dependency to ensure the current user is an admin.

Args:

current_user (dict): The current user dictionary, injected by Depends(get_current_us

Raises:

HTTPException: If the user is not an admin.

Returns:

dict: The current user dictionary if the user is an admin.

=====

File: .\main.py

Loại: function

Tên: fetch_contest_info

Docstring:

Fetch contest information from the Codeforces API for a given contest ID.

Args:

contest_id (int): The ID of the contest to fetch.

Returns:

dict or None: The contest information dictionary if successful, otherwise None.

=====

File: .\main.py

Loại: function

Tên: update_contests

Docstring:

Update contests and standings in the database from the Codeforces API.

Returns:

dict: A message indicating how many contests were added or updated.

=====

File: .\main.py

Loại: function

Tên: update_contests_get

Docstring:

HTTP GET endpoint to trigger updating contests and standings.

Returns:

dict: A message indicating how many contests were added or updated.

=====

File: .\main.py

Loại: function

Tên: contest_standings

Docstring:

Render the standings page for a specific contest.

Args:

request (Request): The incoming HTTP request.

contest_id (int): The ID of the contest.

Returns:

TemplateResponse: The rendered standings.html template.

=====

File: .\main.py

Loại: function

Tên: fetch_contest_standings

Docstring:

Fetch standings for a given contest from the database.

Args:

contest_id (int): The ID of the contest.

Returns:

list: A list of dictionaries containing handle, rank, and problems solved.

=====

File: .\main.py

Loại: function

Tên: member_info

Docstring:

Render the statistics page for a specific member (user).

Args:

request (Request): The incoming HTTP request.

handle (str, optional): The handle (username) of the member.

Returns:

TemplateResponse: The rendered member.html template with member statistics.

=====

File: .\main.py

Loại: function

Tên: fetch_problem_names

Docstring:

Fetch all problem names for a given contest from the database.

Args:

contest_id (int): The ID of the contest.

Returns:

list: A list of problem names for the contest.

=====

File: .\main.py

Loại: function

Tên: home

Docstring:

Render the home page with contest list, top solvers, and problems for each contest.

Args:

request (Request): The incoming HTTP request.

Returns:

TemplateResponse: The rendered home.html template.

=====

File: .\backend\models.py

Loại: class

Tên: Contest

Docstring:

ORM model for a programming contest.

Attributes:

id (int): Primary key, unique contest ID.

name (str): Name of the contest.

type (str): Type of the contest (e.g., ICPC, CF).

phase (str): Current phase of the contest.

durationSeconds (int): Duration of the contest in seconds.

standings (list[Standing]): List of standings for this contest.

problems (list[Problem]): List of problems for this contest.

=====

File: .\backend\models.py

Loại: class

Tên: Solver

Docstring:

ORM model for a solver (participant).

Attributes:

id (int): Primary key, unique solver ID.

handle (str): Unique handle/username of the solver.

solved_count (int): Total number of problems solved by the solver.

=====

File: .\backend\models.py

Loại: class

Tên: Problem

Docstring:
ORM model for a problem in a contest.

Attributes:
id (int): Primary key, unique problem ID.
name (str): Name of the problem.
topics (str): Topics/tags associated with the problem.
contest_id (int): Foreign key to the contest this problem belongs to.
contest (Contest): Relationship to the Contest model.

=====

File: .\backend\models.py
Loại: class
Tên: Standing
Docstring:
ORM model for a contest standing (result for a solver in a contest).

Attributes:
id (int): Primary key, unique standing ID.
handle (str): Handle/username of the solver.
rank (int): Rank achieved by the solver in the contest.
problems_solved (int): Number of problems solved by the solver in the contest.
contest_id (int): Foreign key to the contest.
contest (Contest): Relationship to the Contest model.

=====

File: .\backend\sign_url.py
Loại: function
Tên: generate_random_string
Docstring:
Generate a random string of fixed length.

Args:
length (int, optional): The length of the random string. Defaults to 6.

Returns:
str: A random alphanumeric string of the specified length.

=====

File: .\backend\sign_url.py
Loại: function
Tên: build_signed_url
Docstring:
Generate a signed URL for the Codeforces API.

Args:
method_name (str): The API method name (e.g., 'contest standings').
params (dict): The parameters to include in the API request.

Returns:
str: The signed URL ready for use with the Codeforces API.

Link github của dự án:

Link ở đây