

怎么运行

1. 首先创建数据库和表

```
CREATE DATABASE "ITECH3108_30407213_a2";
```

切换到这个数据库

```
\c "ITECH3108_30407213_a2";
```

然后创建数据表

```
-- 创建用户表

create table users
(
    username varchar(32) not null
        constraint users_pk
            primary key,
    fullname varchar(64) not null,
    password varchar(255) not null
);

-- 创建分享链接表

create table share_links
(
    id          serial
        constraint share_links_pk
            primary key,
    title       varchar(32) not null,
    link        varchar(255) not null,
    description varchar(255),
    total_score integer default 0
);

-- 创建链接评分表
create table scores
(
    username      varchar(32) not null
        constraint scores_users_username_fk
            references users,
    share_link_id integer not null
        constraint scores_share_links_id_fk
            references share_links,
    score         integer not null
);

-- 创建隐藏链接的表
create table hidden_links
(
    username      varchar(32) not null
        constraint hidden_links_users_username_fk
```

```
        references users,
        share_link_id integer      not null
        constraint hidden_links_share_links_id_fk
        references share_links
    );
```

2. 然后修改 server.js 文件，修改数据库的连接地址，端口号，用户名，密码，数据库名
3. 运行服务端API

```
deno run --allow-net --allow-read .\server.js
```

4. 运行前端

```
file_server
```

5. 浏览器打开

http://127.0.0.1:4507/demo_24_A_4/web/index.html

即可访问本网页了

后端服务

使用的deno进行后端服务开发，服务端的端口为 8050

数据库采用 **PostgreSQL**

后端接口API的跨域支持，参考自：<https://github.com/oakserver/oak/issues/154#issuecomment-640657239>

后端接口保持登录状态采用的JWT，参考自老师课件：week_09\jwt\oak_bearer_jwt.js

后端接口返回的数据格式都是 **JSON**

前端

前端使用 HTML JavaScript CSS

界面

注意：这里的UI界面使用了**Bootstrap5**，官网地址：<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

Bootstrap 是一个开源的前端框架，旨在帮助开发者快速创建响应式、移动设备优先的网站和Web应用程序。它由Twitter的设计师Mark Otto和Jacob Thornton于2011年发布，现在已经是全球最流行的前端框架之一。

说简单点，就是让界面更好看点，他帮我们写了很多CSS样式，我们只需要在HTML中引用即可！

网络请求

还是 fetch 进行网络请求

登录保持

使用的是 JWT，详细可见你们的课件 `week_09/jwt/oak_bearer_jwt.js`

也就是如果登录了，请求的Headers中会携带这个Token

```
headers: {  
  'authorization': `Bearer ${jwt}`  
}
```