Application Title: Language Forest (Novel Reading Platform)

Group 1 危湘妤(B104020011), 程愷元(B104020021), 林敬原(B104020038)

1. Briefly describe what your project is about and what the project accomplished.

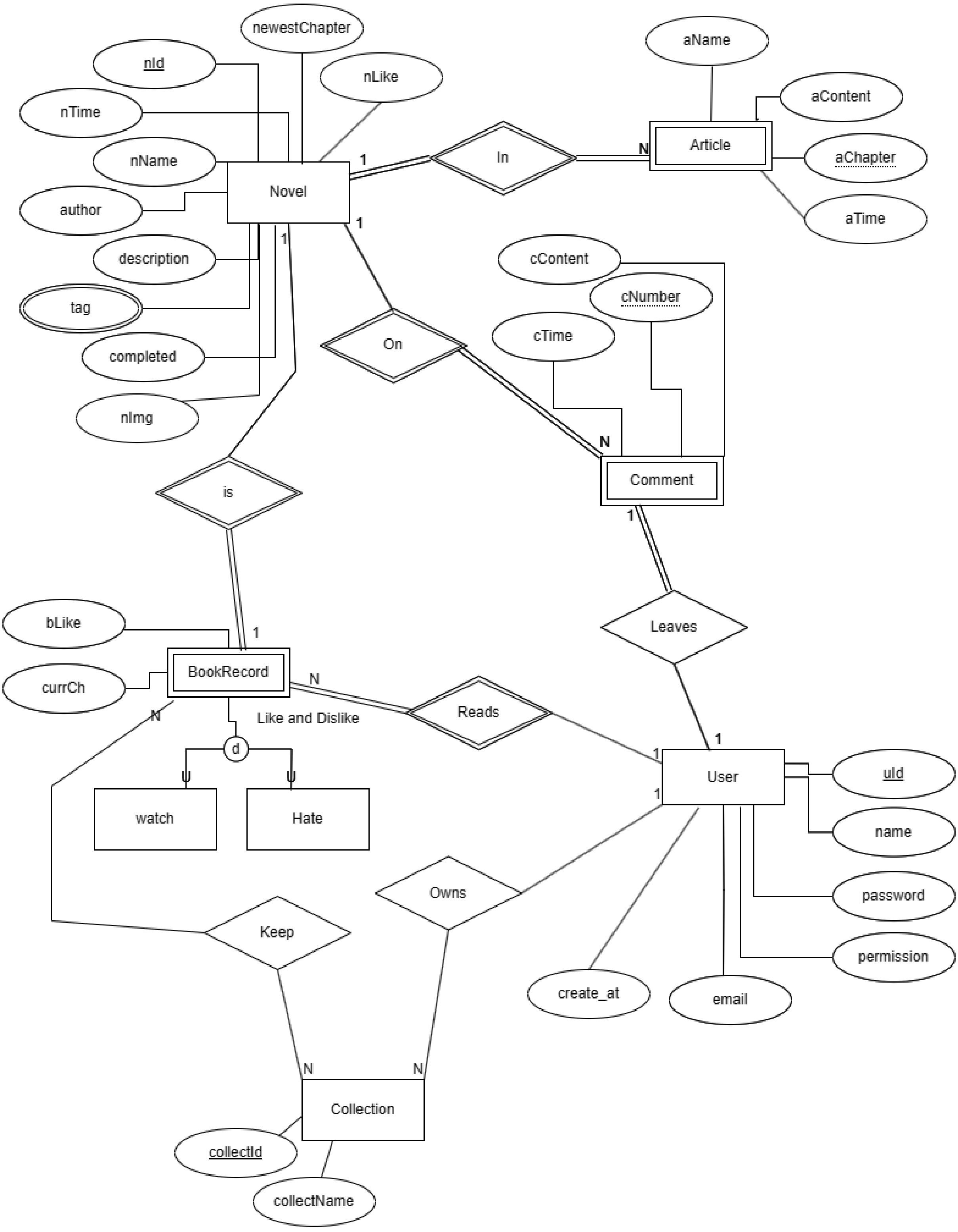
The project is a novel reader. The users can comment, read, and crawl novels on our website. When there is a novel that the website doesn't have, they can crawl it down with the help of our website and then read it in the regular form(.docx).
Also,they can use the music we provide as BGM(Background Music). The music will be matched to the emotion or mood which the article provides if we set the music class(by ai), and play default music when no music relative class (so as not to make the owner spend too much time on running AI sentiment analysis when adding a new novel or article).

2.Usefulness

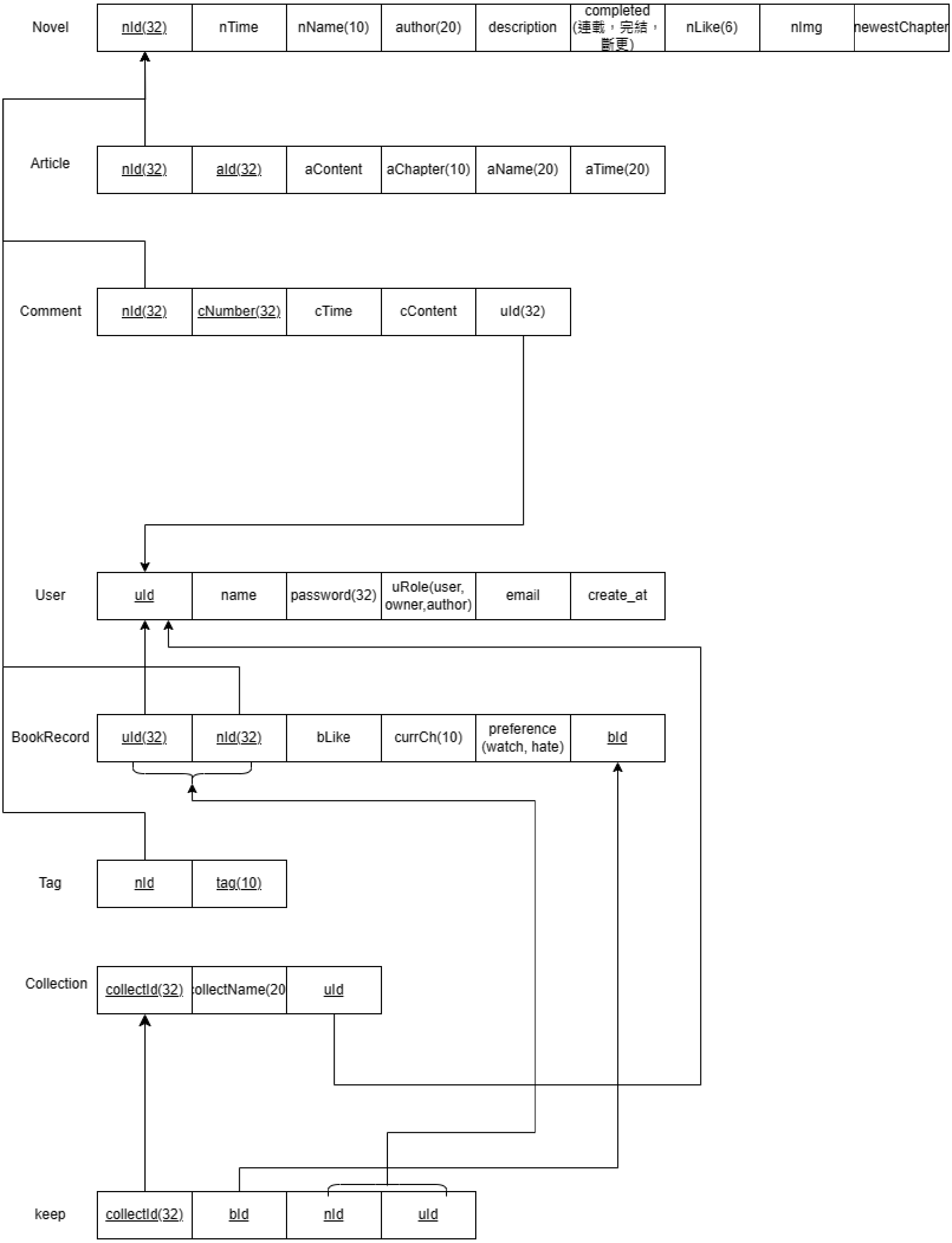
- Novel readers can help users find some new novels they may be interested in, and can record the novel they really like, so they can easily find them in the future, also out novel readers can let users ignore the novel they hate when they searching novels, which can let them not seeing the novel make they really piss off when finding things they like.
- Novel readers nowadays can't crawl data. So, we can't watch the novel if it's not on the website. Therefore, we make a website that can not only watch novels online, but crawl data from other websites.
- Besides, our website can play music based on the feeling of text, which makes users more immersed into the novel.

3. Include your ER diagram and schema

ER diagram:



Schema:



Novel: nId(32) | nTime | nName(10) | author(20) | description | completed (連載，完結，斷更) | nLike(6) | nImg | newestChapter

Article: nId(32) | aId(32) | aContent | aChapter(10) | aName(20) | aTime(20)

Comment: nId(32) | cNumber(32) | cTime | cContent | uId(32)

User: uId | name | password(32) | uRole(user, owner,author) | email | create_at

BookRecord: uId(32) | nId(32) | bLike | currCh(10) | preference (watch, hate) | bId

Tag: nId | tag(10)

Collection: collectId(32) | collectName(20 | uId

keep: collectId(32) | bId | nId | uId

4.Discuss the data in your database. Briefly discuss from where you collected data and show you did it (if crawling is automated, explain how and what tools are used)

The data we collected is from “小說狂人", “起點中文網", “晉江文學城". We used requests package to get the content of the website(unicode form), used beautifulsoup package to resolve data to html and get required data. Some novels on “晉江文學城" need to be processed by special function so we use selenium package to finish dynamic web crawler. If the text on the webpage is Simplified Chinese, use Opencc to convert it to Traditional Chinese and then save it to DB(using pymysql package) and save it as a Docx file(using docx package) to the user's local path.

5.Clearly list the functionality of your application (feature specs)

For user, they can:

- Read novel
    - They can change background colour
    - They can change the font
    - They can change the font size
    - They can decide to listen the BGM or not
- Comment on novels
    - They can also delete and edit
- Add tags certain novels
- Add novels they liked into their own collection.
    - They can create their own collection folder with the name they give personally
    - They can delete a whole collection folder at once or delete one novel from the collection one by one.
- Hide certain novels that they don’t want to see.
- Give like to certain novels, which will help the novel be seen by more people.
- Crawl novels on certain websites into their own device.
- Search novel based on novel name, author, tag, and its status(whether the novel is completed or not)

For owner, they can do everything an user can do, but when they crawl data, the data will directly inserted into database

6. Explain on basic functions

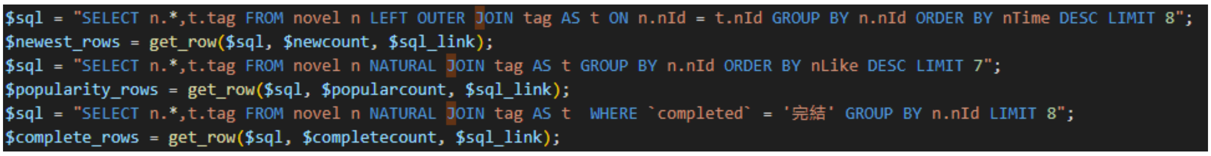
The table to explain our basic function used on the different pages. Beside the below function, we add logging in, signing up, and logging out function.

DB	Insert	Delete	Update	Query
Novel	By owner	X	By owner	O
Article	By owner	X	By owner	O
Comment	O	O	O	O
Collection	O	O	X	O
Book Record	O	X	O	O
User	O	X	O	O

Tag	O	X	X	O

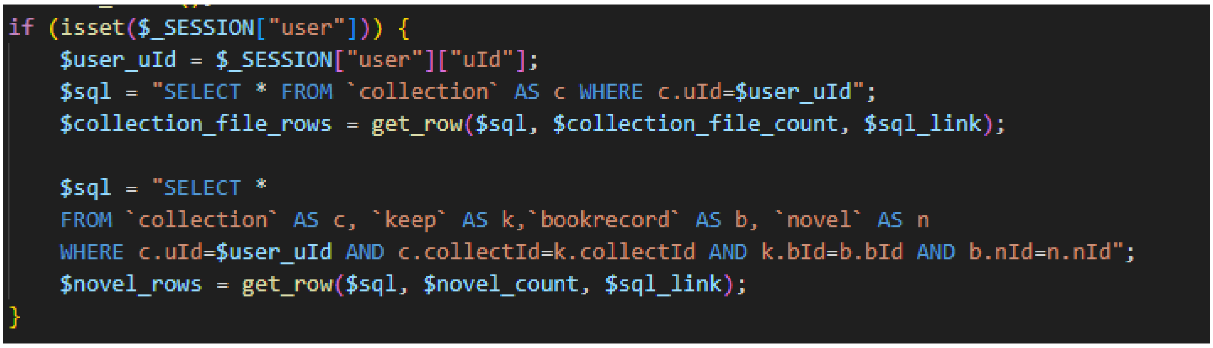
7. Show the actual SQL code snippet

For list out novel in index.php (Join two or more tables in the query)



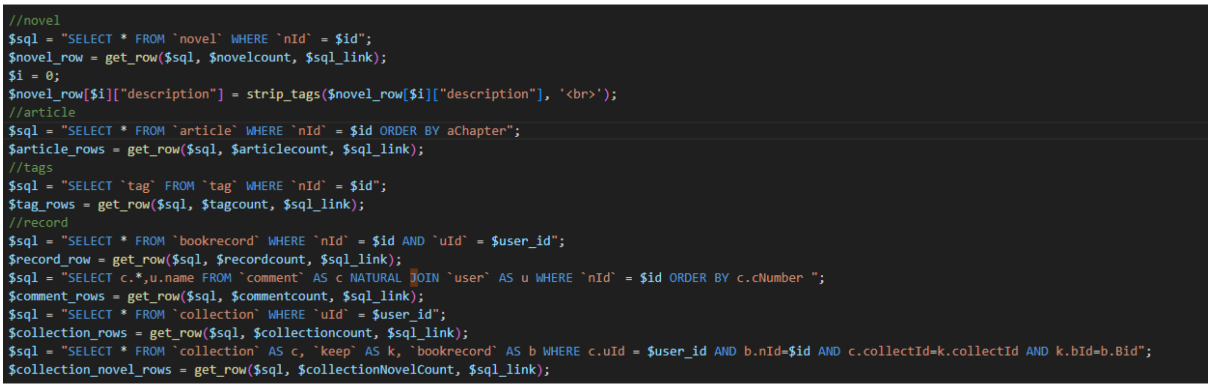
```
$sql = "SELECT n.*,t.tag FROM novel n LEFT OUTER JOIN tag AS t ON n.nId = t.nId GROUP BY n.nId ORDER BY nTime DESC LIMIT 8";
$newest_rows = get_row($sql, $newcount, $sql_link);
$sql = "SELECT n.*,t.tag FROM novel n NATURAL JOIN tag AS t GROUP BY n.nId ORDER BY nLike DESC LIMIT 7";
$popularity_rows = get_row($sql, $popularcount, $sql_link);
$sql = "SELECT n.*,t.tag FROM novel n NATURAL JOIN tag AS t  WHERE `completed` = '完結' GROUP BY n.nId LIMIT 8";
$complete_rows = get_row($sql, $completecount, $sql_link);
```

For list out the novel in collection.php: (Join two or more tables in the qery)



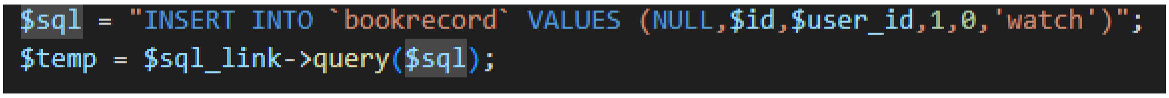
```
if (isset($_SESSION["user"])) {
    $user_uId = $_SESSION["user"]["uId"];
    $sql = "SELECT * FROM `collection` AS c WHERE c.uId=$user_uId";
    $collection_file_rows = get_row($sql, $collection_file_count, $sql_link);

    $sql = "SELECT *
    FROM `collection` AS c, `keep` AS k,`bookrecord` AS b, `novel` AS n
    WHERE c.uId=$user_uId AND c.collectId=k.collectId AND k.bId=b.bId AND b.nId=n.nId";
    $novel_rows = get_row($sql, $novel_count, $sql_link);
}
```

For the data we need to list out in novel.php

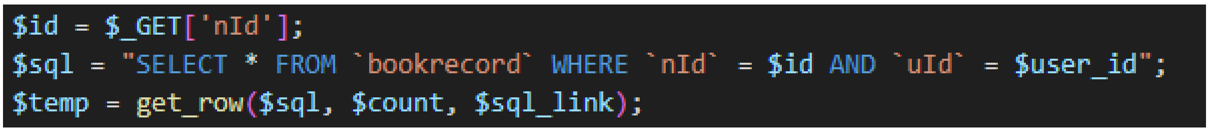


```
//novel
$sql = "SELECT * FROM `novel` WHERE `nId` = $id";
$novel_row = get_row($sql, $novelcount, $sql_link);
$i = 0;
$novel_row[$i]["description"] = strip_tags($novel_row[$i]["description"], '<br>');
//article
$sql = "SELECT * FROM `article` WHERE `nId` = $id ORDER BY aChapter";
$article_rows = get_row($sql, $articlecount, $sql_link);
//tags
$sql = "SELECT `tag` FROM `tag` WHERE `nId` = $id";
$tag_rows = get_row($sql, $tagcount, $sql_link);
//record
$sql = "SELECT * FROM `bookrecord` WHERE `nId` = $id AND `uId` = $user_id";
$record_row = get_row($sql, $recordcount, $sql_link);
$sql = "SELECT c.*,u.name FROM `comment` AS c NATURAL JOIN `user` AS u WHERE `nId` = $id ORDER BY c.cNumber ";
$comment_rows = get_row($sql, $commentcount, $sql_link);
$sql = "SELECT * FROM `collection` WHERE `uId` = $user_id";
$collection_rows = get_row($sql, $collectioncount, $sql_link);
$sql = "SELECT * FROM `collection` AS c, `keep` AS k, `bookrecord` AS b WHERE c.uId = $user_id AND b.nId=$id AND c.collectId=k.collectId AND k.bId=b.Bid";
$collection_novel_rows = get_row($sql, $collectionNovelCount, $sql_link);
```

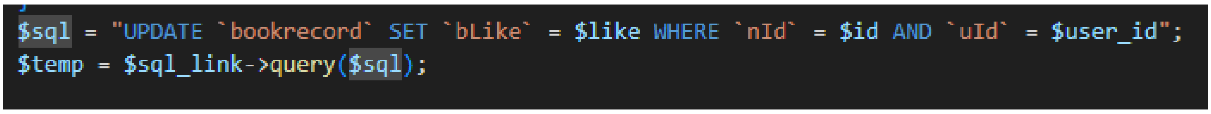
For the operation when clicking like relative buttons in novel.php (Contains Insert,Update)



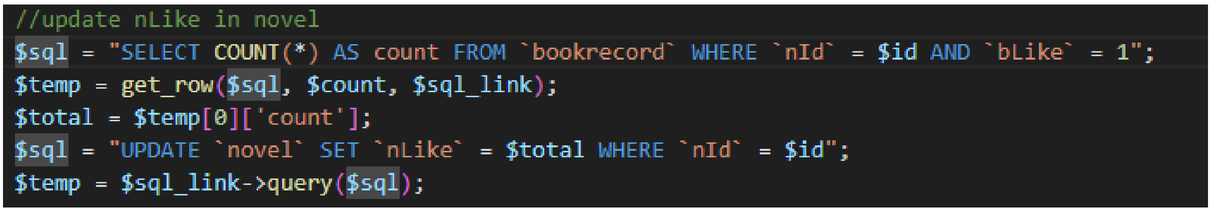
```
 $sql = "INSERT INTO `bookrecord` VALUES (NULL,$id,$user_id,1,0,'watch')";
 $temp = $sql_link->query($sql);
```



```
$id = $_GET['nId'];
$sql = "SELECT * FROM `bookrecord` WHERE `nId` = $id AND `uId` = $user_id";
$temp = get_row($sql, $count, $sql_link);
```

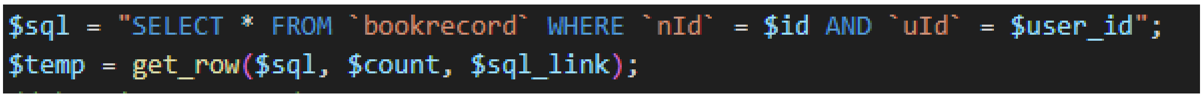


```
 $sql = "UPDATE `bookrecord` SET `bLike` = $like WHERE `nId` = $id AND `uId` = $user_id";
 $temp = $sql_link->query($sql);
```

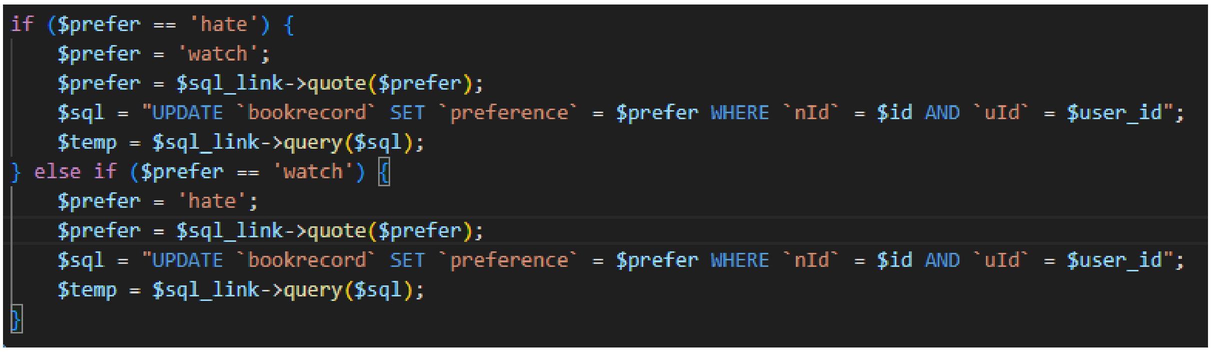


```
//update nLike in novel
$sql = "SELECT COUNT(*) AS count FROM `bookrecord` WHERE `nId` = $id AND `bLike` = 1";
$temp = get_row($sql, $count, $sql_link);
$total = $temp[0]['count'];
$sql = "UPDATE `novel` SET `nLike` = $total WHERE `nId` = $id";
$temp = $sql_link->query($sql);
```

For the operation when clicking the button about “I don’t want to see this novel again” (Contains Update)

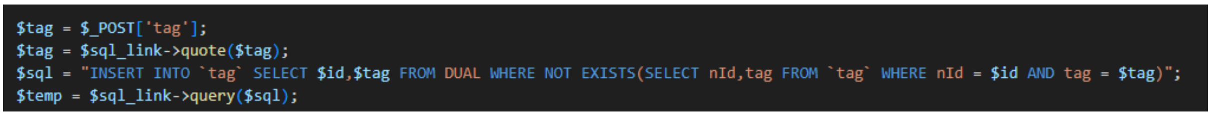


```
$sql = "SELECT * FROM `bookrecord` WHERE `nId` = $id AND `uId` = $user_id";
$temp = get_row($sql, $count, $sql_link);
```



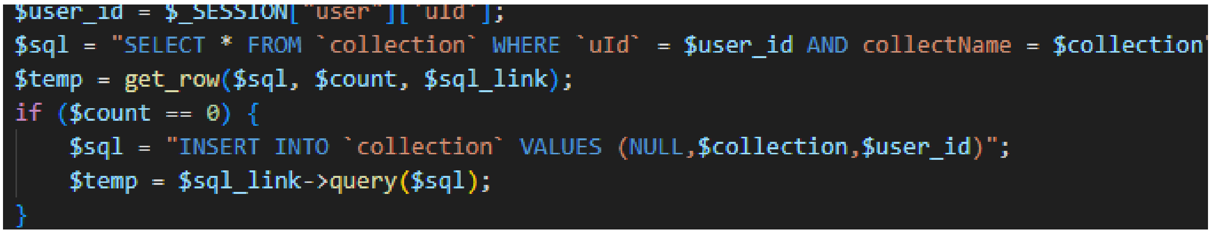
```
if ($prefer == 'hate') {
    $prefer = 'watch';
    $prefer = $sql_link->quote($prefer);
    $sql = "UPDATE `bookrecord` SET `preference` = $prefer WHERE `nId` = $id AND `uId` = $user_id";
    $temp = $sql_link->query($sql);
} else if ($prefer == 'watch') {
    $prefer = 'hate';
    $prefer = $sql_link->quote($prefer);
    $sql = "UPDATE `bookrecord` SET `preference` = $prefer WHERE `nId` = $id AND `uId` = $user_id";
    $temp = $sql_link->query($sql);
}
```

For the operation when click the button about add tag in novel.php
(Contains Insert)



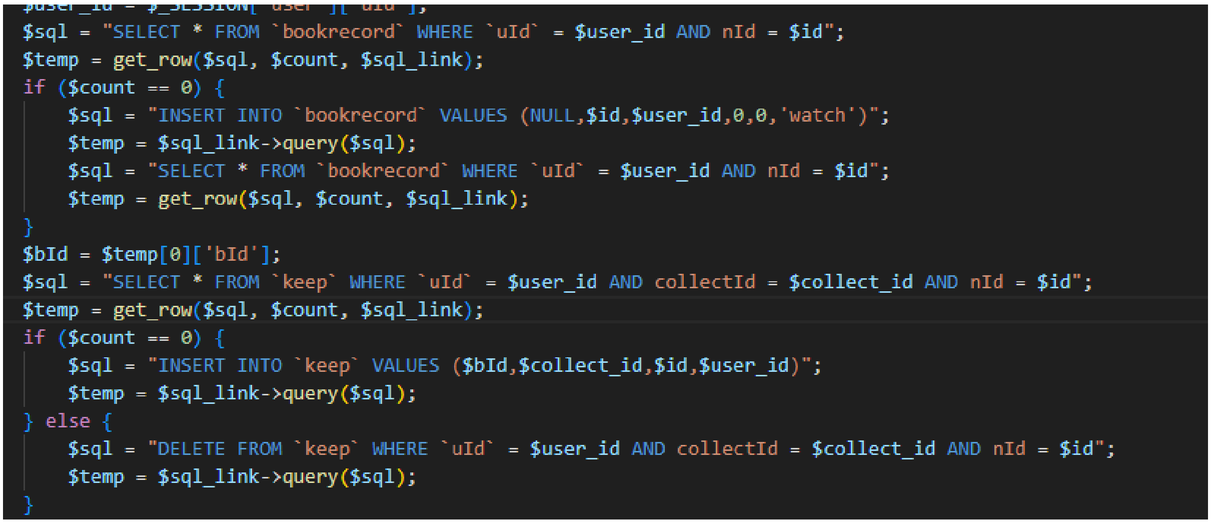
```
$tag = $_POST['tag'];
$tag = $sql_link->quote($tag);
$sql = "INSERT INTO `tag` SELECT $id,$tag FROM DUAL WHERE NOT EXISTS(SELECT nId,tag FROM `tag` WHERE nId = $id AND tag = $tag)";
$temp = $sql_link->query($sql);
```

For the operation when click the button about add collection
Add new folder
Contains(Update, Insert, Delete)



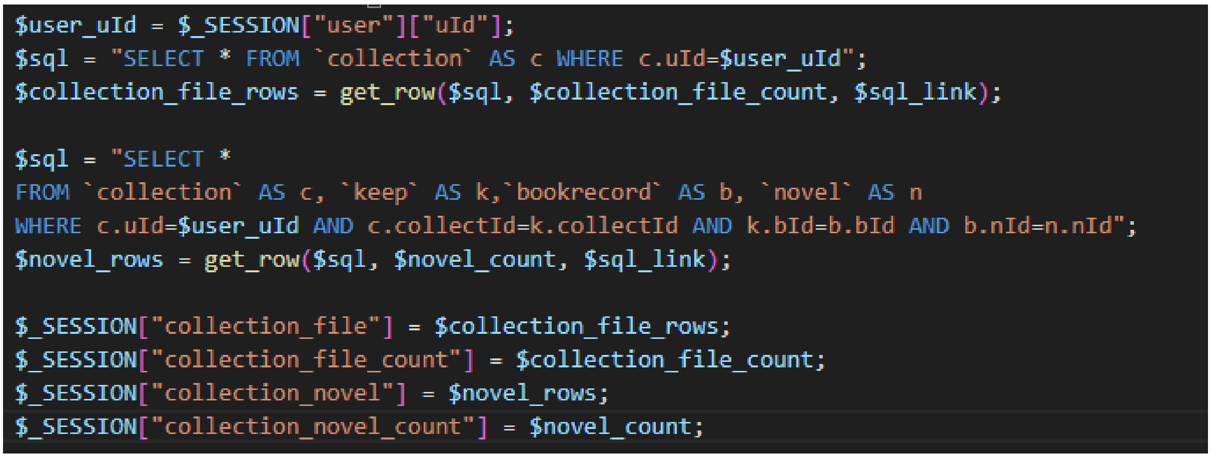
```
$user_id = $_SESSION["user"]['uId'];
$sql = "SELECT * FROM `collection` WHERE `uId` = $user_id AND collectName = $collection
$temp = get_row($sql, $count, $sql_link);
if ($count == 0) {
    $sql = "INSERT INTO `collection` VALUES (NULL,$collection,$user_id)";
    $temp = $sql_link->query($sql);
}
```

Add to a folder or remove from a folder (collection folder)



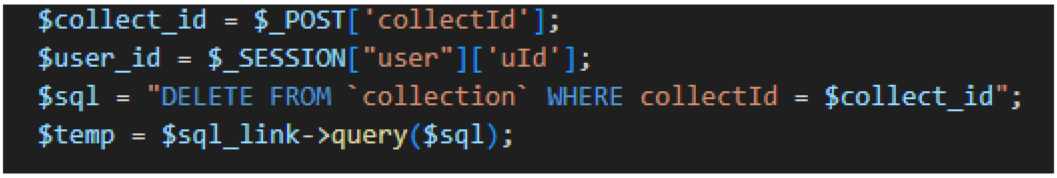
```
$user_id = $_SESSION["user"]['uId'];
$sql = "SELECT * FROM `bookrecord` WHERE `uId` = $user_id AND nId = $id";
$temp = get_row($sql, $count, $sql_link);
if ($count == 0) {
    $sql = "INSERT INTO `bookrecord` VALUES (NULL,$id,$user_id,0,0,'watch')";
    $temp = $sql_link->query($sql);
    $sql = "SELECT * FROM `bookrecord` WHERE `uId` = $user_id AND nId = $id";
    $temp = get_row($sql, $count, $sql_link);
}
$bId = $temp[0]['bId'];
$sql = "SELECT * FROM `keep` WHERE `uId` = $user_id AND collectId = $collect_id AND nId = $id";
$temp = get_row($sql, $count, $sql_link);
if ($count == 0) {
    $sql = "INSERT INTO `keep` VALUES ($bId,$collect_id,$id,$user_id)";
    $temp = $sql_link->query($sql);
} else {
    $sql = "DELETE FROM `keep` WHERE `uId` = $user_id AND collectId = $collect_id AND nId = $id";
    $temp = $sql_link->query($sql);
}
```

Get the book stat about the collection
(Join two or more tables in the query)



```
$user_uId = $_SESSION["user"]["uId"];
$sql = "SELECT * FROM `collection` AS c WHERE c.uId=$user_uId";
$collection_file_rows = get_row($sql, $collection_file_count, $sql_link);

$sql = "SELECT *
FROM `collection` AS c, `keep` AS k,`bookrecord` AS b, `novel` AS n
WHERE c.uId=$user_uId AND c.collectId=k.collectId AND k.bId=b.bId AND b.nId=n.nId";
$novel_rows = get_row($sql, $novel_count, $sql_link);

$_SESSION["collection_file"] = $collection_file_rows;
$_SESSION["collection_file_count"] = $collection_file_count;
$_SESSION["collection_novel"] = $novel_rows;
$_SESSION["collection_novel_count"] = $novel_count;
```

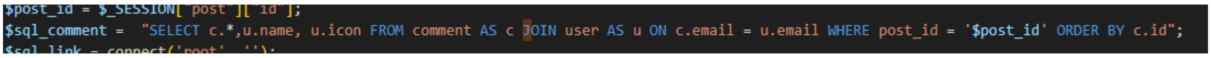
Delete the folder Contains(Delete)



```
$collect_id = $_POST['collectId'];
$user_id = $_SESSION["user"]['uId'];
$sql = "DELETE FROM `collection` WHERE collectId = $collect_id";
$temp = $sql_link->query($sql);
```

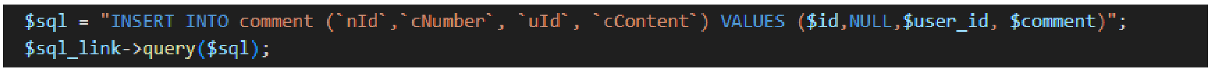
For the comment part in novel.php

Display comment (Join two or more tables in the qery)



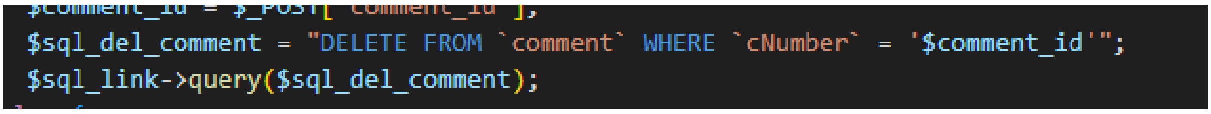
```
$post_id = $_SESSION["post"]["id"];
$sql_comment =  "SELECT c.*,u.name, u.icon FROM comment AS c JOIN user AS u ON c.email = u.email WHERE post_id = '$post_id' ORDER BY c.id";
$sql_link = connect('root', '');
```

Add comment (Insert)



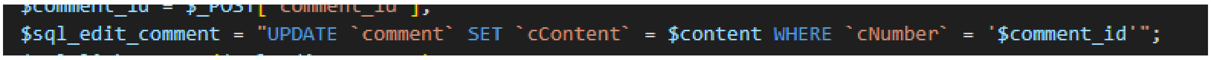
```
$sql = "INSERT INTO comment (`nId`,`cNumber`, `uId`, `cContent`) VALUES ($id,NULL,$user_id, $comment)";
$sql_link->query($sql);
```

Delete comment (Delete)



```
$comment_id = $_POST['comment_id'];
$sql_del_comment = "DELETE FROM `comment` WHERE `cNumber` = '$comment_id'";
$sql_link->query($sql_del_comment);
```

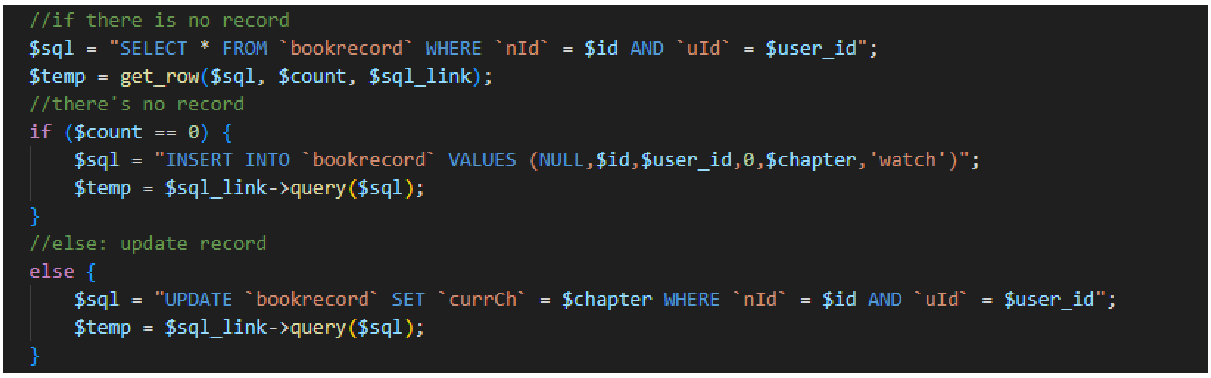
Edit comment (Update)



```
$comment_id = $_POST['comment_id'];
$sql_edit_comment = "UPDATE `comment` SET `cContent` = $content WHERE `cNumber` = '$comment_id'";
```

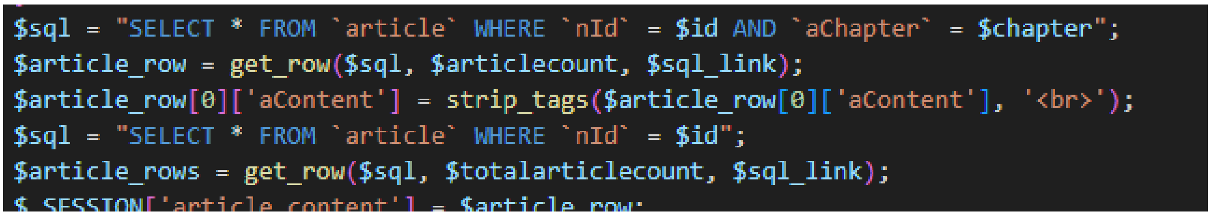
For article.php

For the book record stat (Insert, Update)



```
//if there is no record
$sql = "SELECT * FROM `bookrecord` WHERE `nId` = $id AND `uId` = $user_id";
$temp = get_row($sql, $count, $sql_link);
//there's no record
if ($count == 0) {
    $sql = "INSERT INTO `bookrecord` VALUES (NULL,$id,$user_id,0,$chapter,'watch')";
    $temp = $sql_link->query($sql);
}
//else: update record
else {
    $sql = "UPDATE `bookrecord` SET `currCh` = $chapter WHERE `nId` = $id AND `uId` = $user_id";
    $temp = $sql_link->query($sql);
}
```

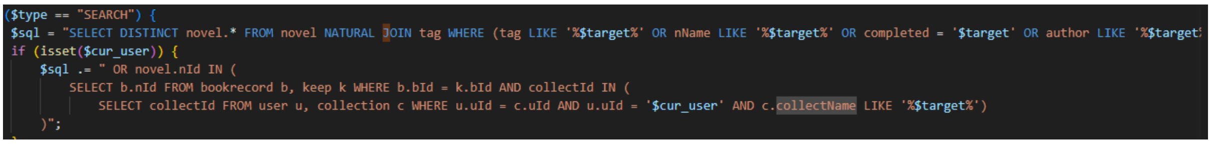
For the content of article



```
$sql = "SELECT * FROM `article` WHERE `nId` = $id AND `aChapter` = $chapter";
$article_row = get_row($sql, $articlecount, $sql_link);
$article_row[0]['aContent'] = strip_tags($article_row[0]['aContent'], '<br>');
$sql = "SELECT * FROM `article` WHERE `nId` = $id";
$article_rows = get_row($sql, $totalarticlecount, $sql_link);
$_SESSION['article_content'] = $article_row;
```

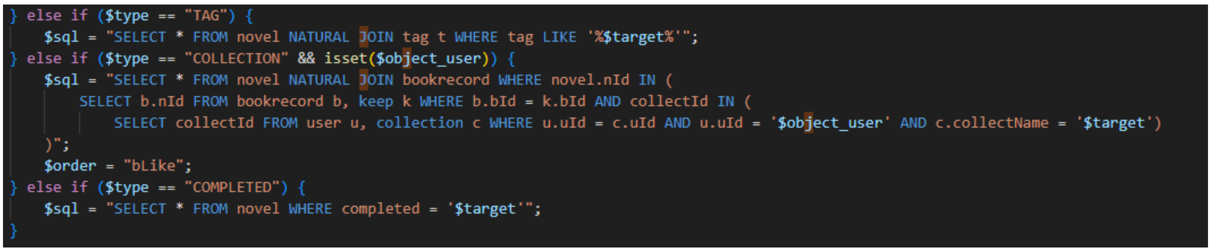
For the novel_list.php

From search bar (Join two or more table)



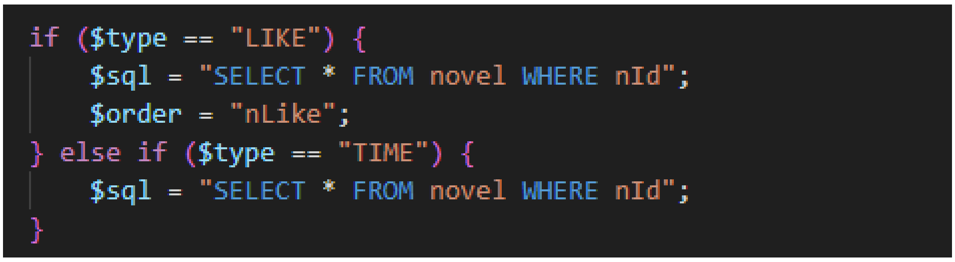
```
($type == "SEARCH") {
 $sql = "SELECT DISTINCT novel.* FROM novel NATURAL JOIN tag WHERE (tag LIKE '%$target%' OR nName LIKE '%$target%' OR completed = '$target' OR author LIKE '%$target
 if (isset($cur_user)) {
     $sql .= " OR novel.nId IN (
         SELECT b.nId FROM bookrecord b, keep k WHERE b.bId = k.bId AND collectId IN (
             SELECT collectId FROM user u, collection c WHERE u.uId = c.uId AND u.uId = '$cur_user' AND c.collectName LIKE '%$target%')
     )";
```

From novel tag button, list out collection when clicking collection title in collection.php, and from completed button (Join two or more table)



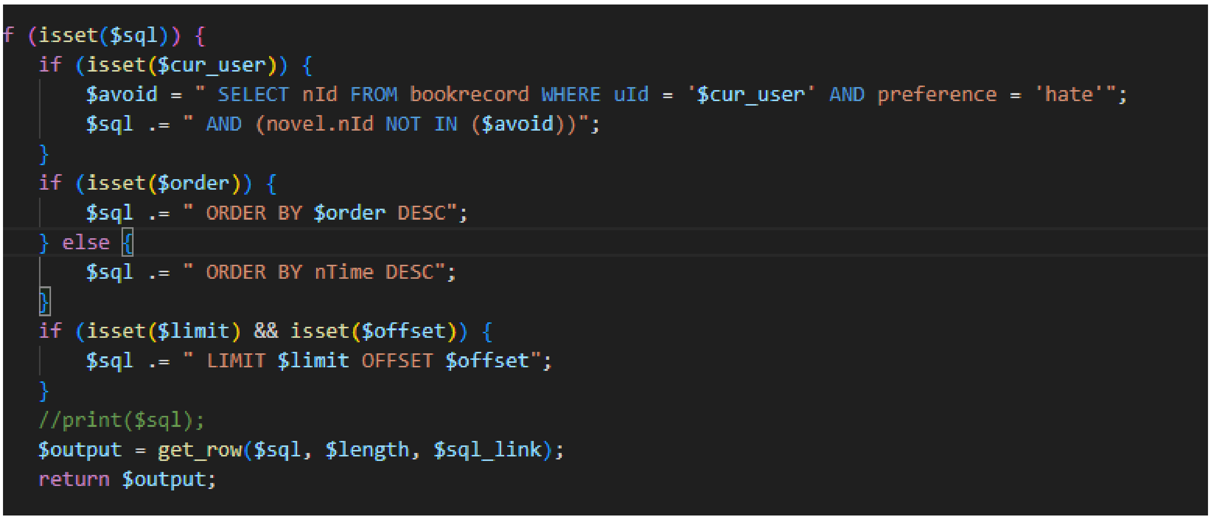
```
} else if ($type == "TAG") {
    $sql = "SELECT * FROM novel NATURAL JOIN tag t WHERE tag LIKE '%$target%'";
} else if ($type == "COLLECTION" && isset($object_user)) {
    $sql = "SELECT * FROM novel NATURAL JOIN bookrecord WHERE novel.nId IN (
        SELECT b.nId FROM bookrecord b, keep k WHERE b.bId = k.bId AND collectId IN (
            SELECT collectId FROM user u, collection c WHERE u.uId = c.uId AND u.uId = '$object_user' AND c.collectName = '$target')
    )";
    $order = "bLike";
} else if ($type == "COMPLETED") {
    $sql = "SELECT * FROM novel WHERE completed = '$target'";
}
```

From Like and TIme in index.php



```
if ($type == "LIKE") {
    $sql = "SELECT * FROM novel WHERE nId";
    $order = "nLike";
} else if ($type == "TIME") {
    $sql = "SELECT * FROM novel WHERE nId";
}
```

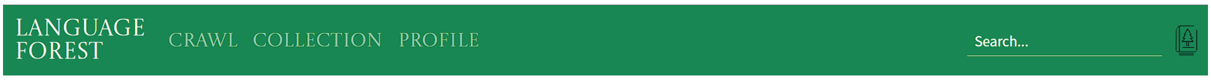
Add the avoid part, and the part to end the sql



```
f (isset($sql)) {
   if (isset($cur_user)) {
       $avoid = " SELECT nId FROM bookrecord WHERE uId = '$cur_user' AND preference = 'hate'";
       $sql .= " AND (novel.nId NOT IN ($avoid))";
   }
   if (isset($order)) {
       $sql .= " ORDER BY $order DESC";
   } else {
       $sql .= " ORDER BY nTime DESC";
   }
   if (isset($limit) && isset($offset)) {
       $sql .= " LIMIT $limit OFFSET $offset";
   }
   //print($sql);
   $output = get_row($sql, $length, $sql_link);
   return $output;
```

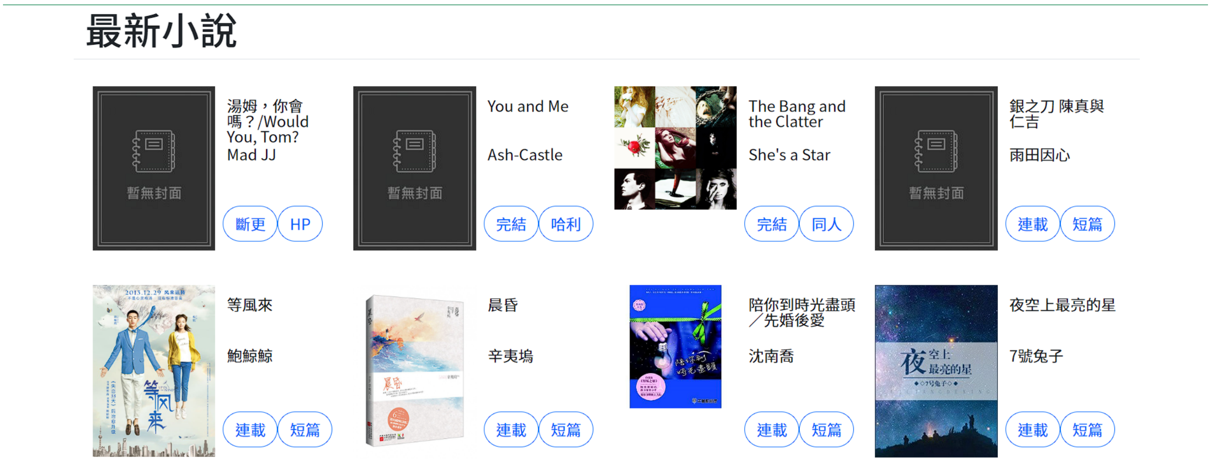
8. List and briefly explain the dataflow

- Navbar



- Use the icon on the top left corner to go to the main page.
- Use the “Crawl” button to go to the crawler page.
- Use the “Collection” button to go to the collection page.(If not login,then this button will redirect to the login modal)
- Use the “Profile” button to watch the user’s profile in modal form.(If not login,then this button will redirect to the login modal)
- Use the search bar to search.
- Use the icon on the top right corner to login/logout.

- Main page



最新小說

湯姆，你會嗎？/Would You, Tom?
Mad JJ

You and Me
Ash-Castle

The Bang and the Clatter
She's a Star

銀之刀 陳真與仁吉
雨田因心

等風來
鮑鯨鯨

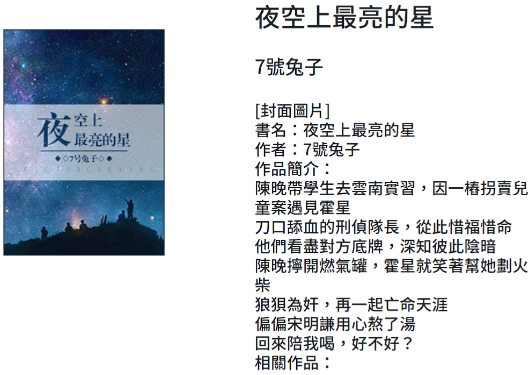
晨昏
辛夷塢

陪你到時光盡頭／先婚後愛
沈南喬

夜空上最亮的星
7號兔子

- Click on the title of each section (For example,”最新小說” in this case) to view every data in particular order(This will list every novel from the newest to the oldest in this case)
- Click on the picture of each novel to go to the corresponding novel page.
- Click on the tag(blue buttons) to search the corresponding tag.

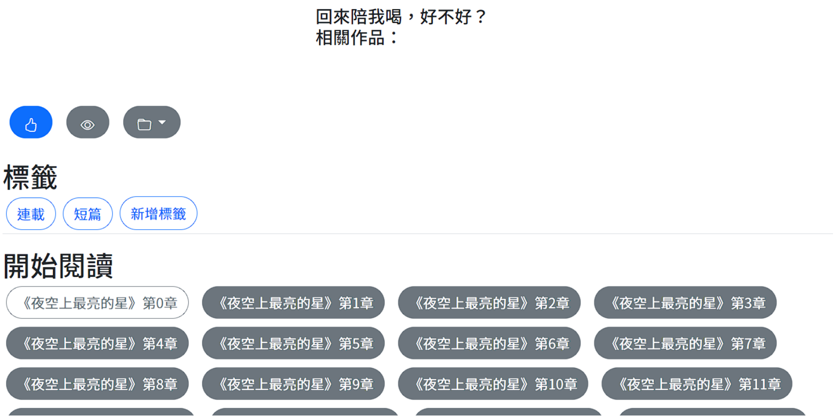
- Novel page- novel description



夜空上最亮的星

7號兔子

[封面圖片]
書名：夜空上最亮的星
作者：7號兔子
作品簡介：
陳晚帶學生去雲南實習，因一樁拐賣兒童案遇見霍星
刀口舔血的刑偵隊長，從此惜福惜命
他們看盡對方底牌，深知彼此陰暗
陳晚擰開燃氣罐，霍星就笑著幫她劃火柴
狼狽為奸，再一起亡命天涯
偏偏宋明謙用心熬了湯
回來陪我喝，好不好？
相關作品：



回來陪我喝，好不好？
相關作品：

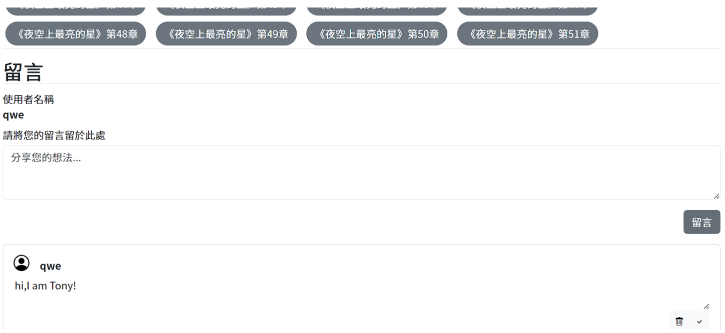
標籤

連載 短篇 新增標籤

開始閱讀

《夜空上最亮的星》第0章 《夜空上最亮的星》第1章 《夜空上最亮的星》第2章 《夜空上最亮的星》第3章
《夜空上最亮的星》第4章 《夜空上最亮的星》第5章 《夜空上最亮的星》第6章 《夜空上最亮的星》第7章
《夜空上最亮的星》第8章 《夜空上最亮的星》第9章 《夜空上最亮的星》第10章 《夜空上最亮的星》第11章

- Click on the like button to leave a like.
- Click on the eye button to dislike this novel.
- Click on the file button to add a novel to my collection.
- Click on the tag(blue buttons) to search the corresponding tag.
- Click on the article page button to article page.



《夜空上最亮的星》第48章 《夜空上最亮的星》第49章 《夜空上最亮的星》第50章 《夜空上最亮的星》第51章

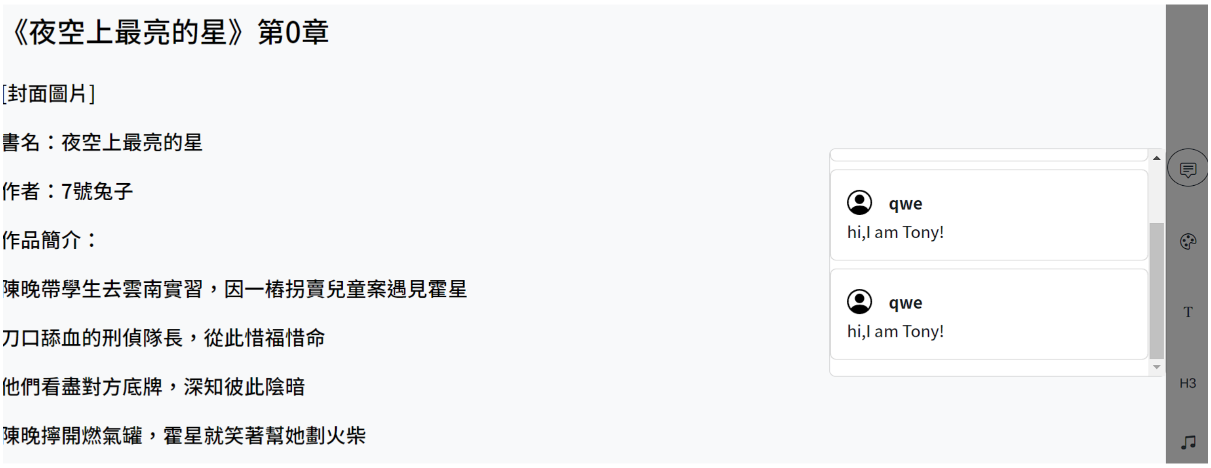
留言

使用者名稱
qwe
請將您的留言留於此處
分享您的想法...

qwe
hi,I am Tony!

- Click on the “留言" button to submit text in text box.

- Article page



《夜空上最亮的星》第0章

[封面圖片]

書名：夜空上最亮的星

作者：7號兔子

作品簡介：

陳晚帶學生去雲南實習，因一樁拐賣兒童案遇見霍星

刀口舔血的刑偵隊長，從此惜福惜命

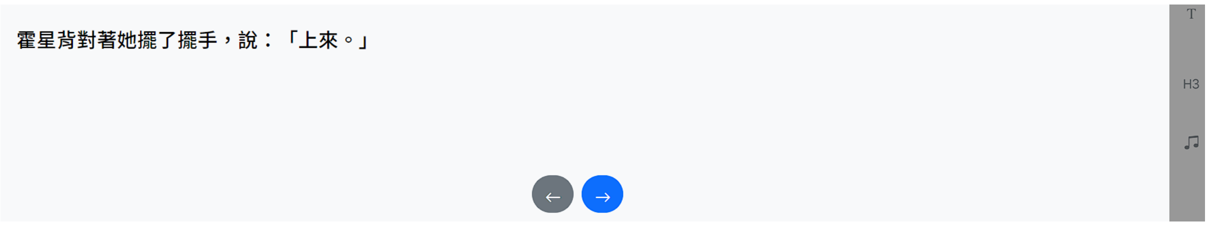
他們看盡對方底牌，深知彼此陰暗

陳晚擰開燃氣罐，霍星就笑著幫她劃火柴

qwe
hi,I am Tony!

qwe
hi,I am Tony!

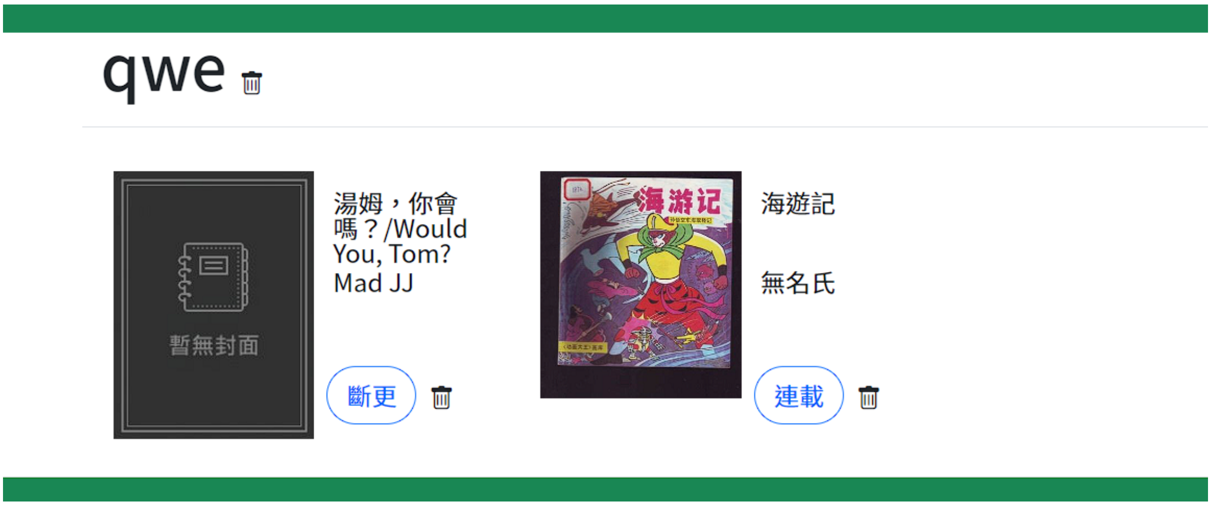
- Click on the first button in the sidebar to pop up the message box.
- Click on the second button in the sidebar to pop up the background modification box.
- Click on the third button in the sidebar to pop up the font modification box.
- Click on the fourth button in the sidebar to pop up the font size modification box.
- Click on the fifth button in the sidebar to pop up the music box.



霍星背對著她擺了擺手，說：「上來。」

- Click on the button at the button to the previous or next page.

- Collection page (click on “collection” button on the navbar)



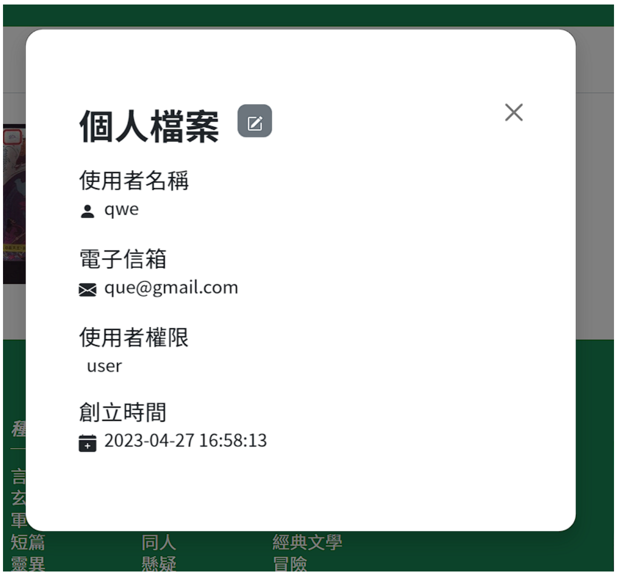
qwe

湯姆，你會嗎？/Would You, Tom?
Mad JJ

海遊記
無名氏

- Click on the trash can button next to the collection name to delete the whole collection.
- Click on the trash can button in the novel block to delete the relation between collection and novel.
- Click on the tag(blue buttons) to search the corresponding tag.
- Click on the novel picture to the novel page.

- Profile modal(click on “profile” button on the navbar)



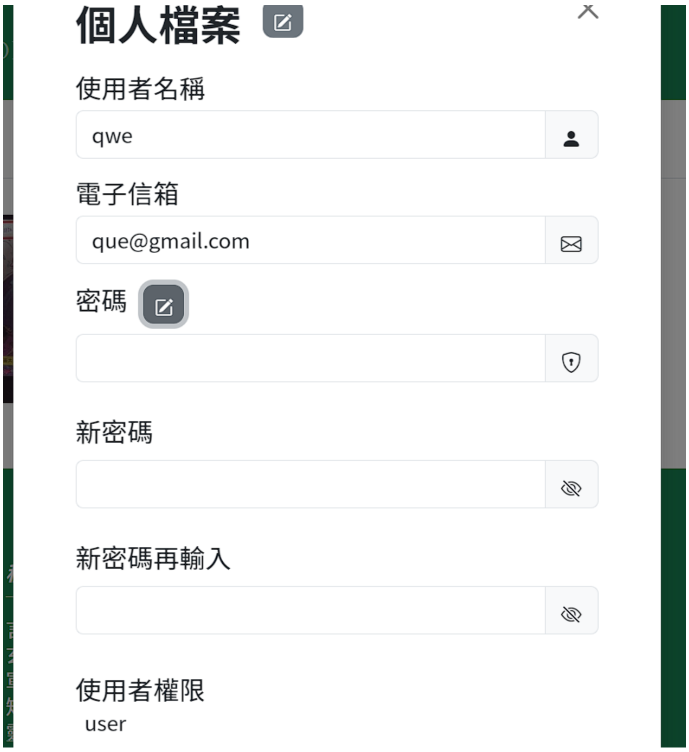
個人檔案

使用者名稱
qwe

電子信箱
que@gmail.com

使用者權限
user

創立時間
2023-04-27 16:58:13



個人檔案

使用者名稱: qwe

電子信箱: que@gmail.com

密碼

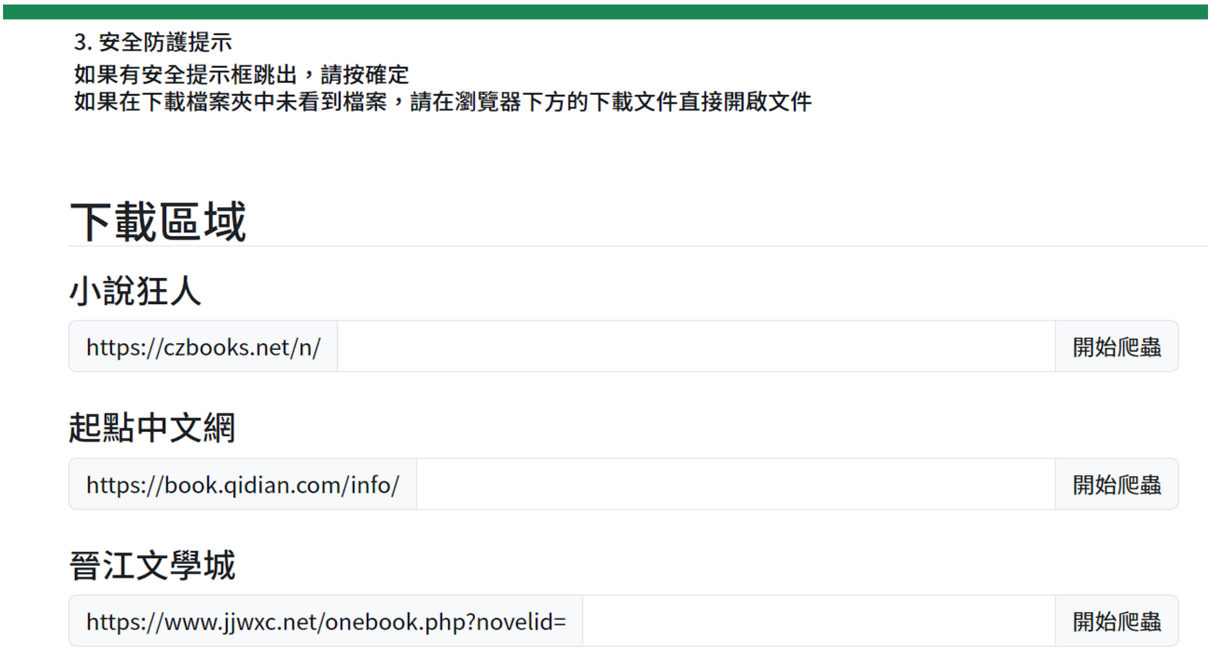
新密碼

新密碼再輸入

使用者權限: user

- Click on the edit button to edit profile.
- Click on the edit button next to the “密碼” to edit password.

- Crawler page(click on “crawler” button on the navbar)



3. 安全防護提示
如果有安全提示框跳出，請按確定
如果在下載檔案夾中未看到檔案，請在瀏覽器下方的下載文件直接開啟文件

下載區域

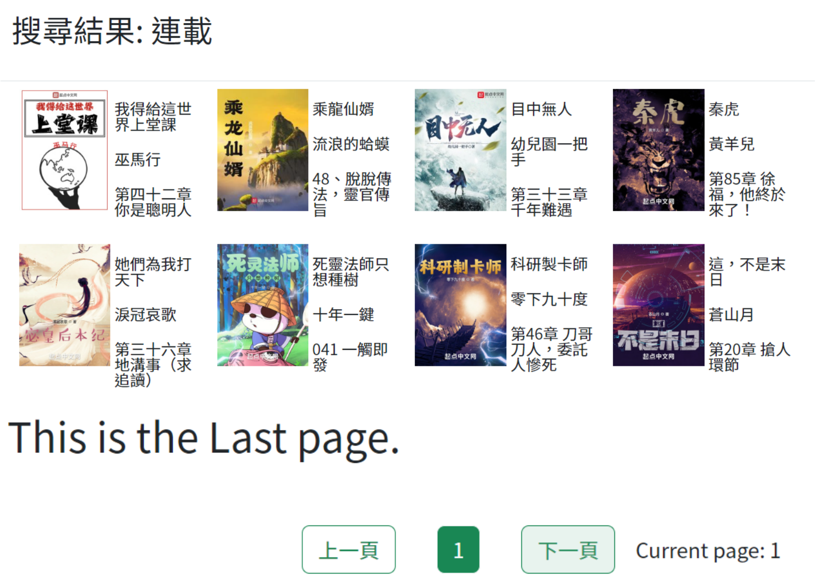
小說狂人: https://czbooks.net/n/

起點中文網: https://book.qidian.com/info/

晉江文學城: https://www.jjwxc.net/onebook.php?novelid=

- Click on the “開始爬蟲” button to fetch docx file related to novel whose url you input.

- Search page(click on “crawler” button on the navbar)



搜尋結果: 連載

我得給這世界上堂課
巫馬行
第四十二章 你是聰明人

乘龍仙婿
流浪的蛤蟆
48、脫脫傳法，靈官傳旨

目中無人
幼兒園一把手
第三十三章 千年難遇

秦虎
黃羊兒
第85章 徐福，他終於來了！

她們為我打天下
淚冠哀歌
第三十六章 地溝事（求追讀）

死靈法師只想種樹
十年一鍵
041 一觸即發

科研製卡師
零下九十度
第46章 刀哥刀人，委託人慘死

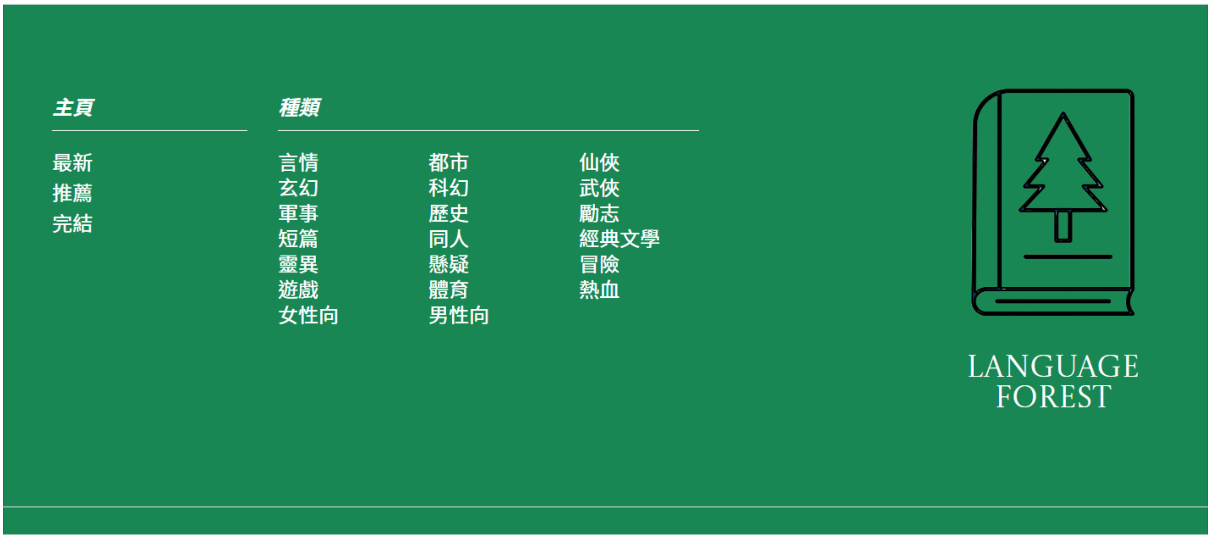
這，不是末日
蒼山月
第20章 搶人環節

This is the Last page.

Current page: 1

- Click on the novel block to the novel page.
- Click on the "上一頁","下一頁" button to the previous and next page..

- Footer



主頁

最新
推薦
完結

種類

言情
玄幻
軍事
短篇
靈異
遊戲
女性向

都市
科幻
歷史
同人
懸疑
體育
男性向

仙俠
武俠
勵志
經典文學
冒險
熱血

- Click on the button under the "主頁" to redirect to the position on the main page.
- Click on the button under the "種類" to search related genres/tags on the search page

9. Explain your advanced function and why it is considered as advanced.

For Crawler function:

We use the crawler function to allow users to crawl novels from other webpages to the local path, and also allow administrators to crawl novels from other webpages to DB on our webpage. Because the web scraping process for different websites requires adjustments based on specifications, formats, and attributes. For example, when loading article pages on "起點小說網", two different scraping methods are needed. "晉江文學城" has redirected pages during the article loading process, requiring the use of dynamic page loading. Formatting the scraped data into a fixed format and saving it as a Docx file also takes time to adjust. Initially, we envisioned making the locally stored data accessible for reading on our website without storing it in a database. However, we later discovered that due to server security features, web pages cannot actively retrieve local files, leading us to abandon this functionality. It would be impractical to require users to upload the file every time they want to view it.

For BGM function :

We will let the owner optionally add music relative tags (BGM tags) to an article content when crawling a novel to a database. When user opens the article page, it will play the music which has already been decided by those BGM tags we added, and if there is no BGM tags, we will play default music we set.cThe method we add BGM tags is through a pre-trained chinese and english text classifier(emotion e.g. joy, love, anger, sadness, fear) model, nanaaaa/emotion_chinese_english, which can help us classify the article we input into the emotion we wanted. The way we use it is through the api from hug face to access it, the reason why not import it is because the speed of our computer makes it take a pretty long time to analyse those articles.As for the music part, we use the music from free-stock-music (https://www.free-stock-music.com/), where music has a different mood and theme. And the music to play when reading a novel will be the random mp3 file we save in the right folder and when the website detects that kind of music it is able to pick a random song in that folder if we have multiple songs in it.

We first thought we could train our own model, but after doing some research, we found out that the Chinese label data for different emotions is hard to find, so we changed to find some pretrained model. And luckily, we find some, and knowing that there are some model have the ability to zero shot classification text which is really surprising us, because we can use some label such as novel_type through that model, however, it can only access english text, so we then translate our text and input to it, but at last the translate package turns out some bug, so we give up this method and just apply chinese emotion analyse. And the way we play music is by some javascript codes, which can using ajax to know the music file in a specific folder and make us able to make it, however after some time of reflecting after finishing the code, we found out it will be much easier to reach the goal of random play audio through database not by get the content of that folder.All in all, we think we have detoured in the building of this function.

10. Describe one technical challenge that you encountered

- It will return errors in some pages if we enter it directly instead of entering through handles, therefore, we will redirect our user to the main page if the user enters certain pages directly.
- The AI model for Chinese classification for novel types is rare, therefore we try to translate our text and input into an English classification model.
- The AI model environment is hard to build up and needs a better computer to execute the analysis we need(even pretty slow in colab), therefore we use API to access.
- The translation package has a word limit, therefore we go back to use a chinese emotion analyse model instead (but lose the ability to analyse novel type)
- The policy about audio play made by google means the audio cannot be auto played.

- The safety guardian stopped me from visiting the local file initiative, so we can’t read local files to our website.
-

11. State if everything went according to the initial development plan and proposed specifications, if not - why?!

There are some points that we didn’t go according to the initial development plan and proposed.Initially, we want to read the local file users crawl down from the website and have the same experience as reading novels on the website, but after knowing the safety policy, we know this feature might be impossible to complete. And, the edit novel content function, which needs the function to read local files on the website be completed. Also, we initially want the user to vote for the song they like,however, we want to try something else to decide the music to play so we change to using ai to give the article a proper song, and when using ai to analyse we want to make it can also classify the novel type of an article as the theme of the music, but there comes some problem to the translate package when dealing with many text, so we give up for it(initially, we want to translate it to english and classify) . In addition, we initially want the music of the article to change through different paragraphs or some big emotion turning, but some users read pretty fast, we are not sure the music can play normally or not if there are multiple types of music in an article.

12. Provide a link to your video

Link to cloud:

https://drive.google.com/file/d/1vl2HmtCTvr-A0qurITTW3sVGDjbhj3eD/view?usp=sharing