Platinum Collector

Wang Haiying – waha21by@student.ju.se

Web Development Fundamentals - Jönköping University - 2023

Contents

1) Introduction

Nowadays, the PlayStation platform has a large player base (Sony, 2023), in which some players are not only looking for entertainment but also want to overcome challenges and obtain in-game achievements. On the PlayStation, if a player gets all the achievements in a game, they will get the highest level of Platinum achievement. Therefore, the project called Platinum Collector was born. I was the only one who designed the user interface and code for the project. This project aims to provide a platform for PlayStation gamers to help each other achieve these achievements. The project is more than just a website, it is a community of gamers that allows each user to share their experiences and insights on how to get achievements in games. Each user will have the opportunity to write comments, tips or insights on how to get a certain achievement, and any user, including non-registered users, will be able to find other people's tips on how to get that achievement underneath the achievement.

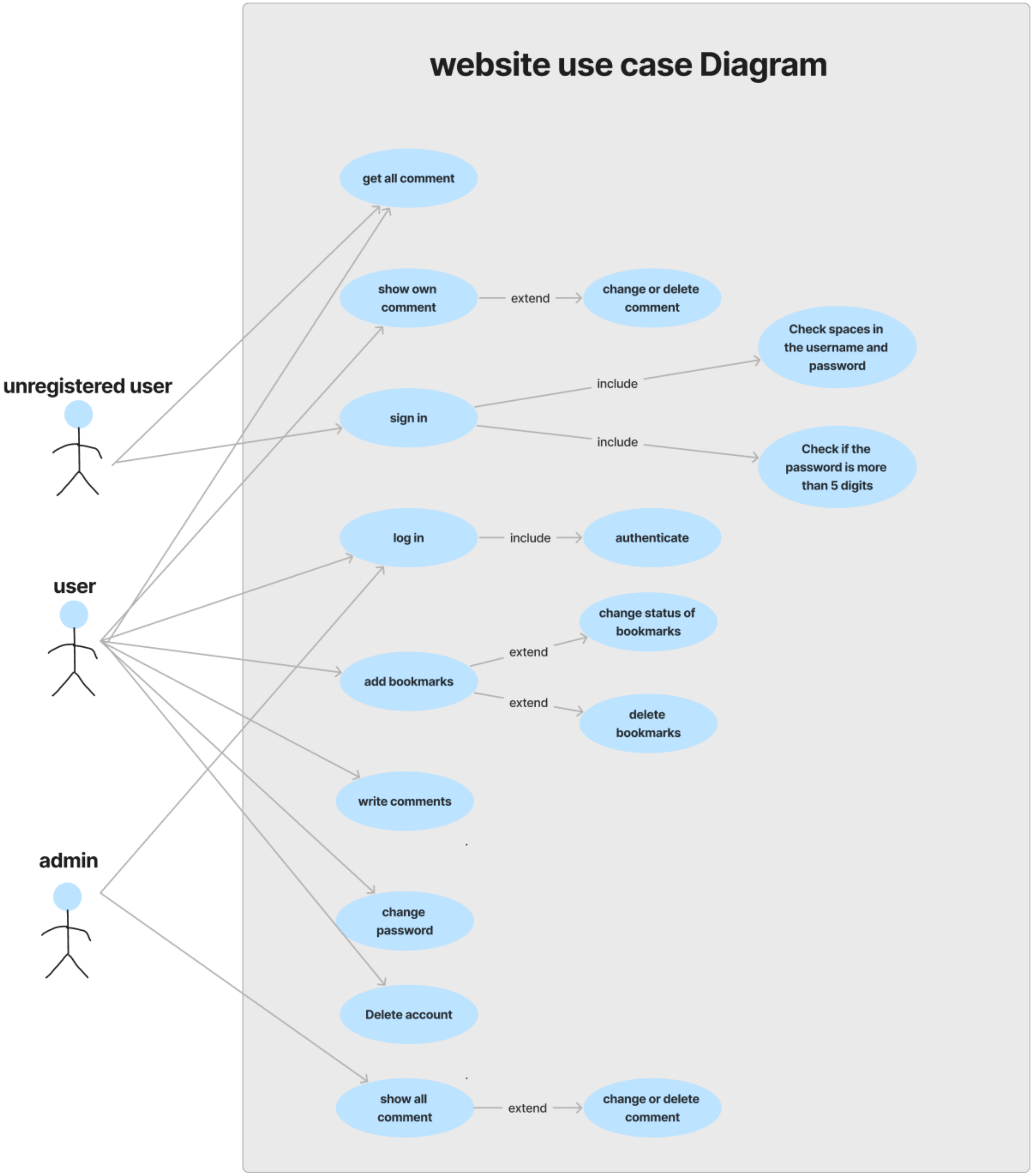

website use case Diagram
unregistered user
user
admin
get all comment
show own comment
extend
change or delete comment
Check spaces in the username and password
include
sign in
include
Check if the password is more than 5 digits
log in
include
authenticate
change status of bookmarks
extend
add bookmarks
extend
delete bookmarks
write comments
change password
Delete account
show all comment
extend
change or delete comment

Figure 1

2) Method

2.1) Architecture

This website consists of four main components: the end user, the viewer, the web application, and the database. End users access this website using a client-side viewer like Chrome to view or write comments. The browser sends an HTTP request to the server-side web application on demand. When the web application receives this request, it uses middleware to send commands to the database and returns data to the web application on demand. The web application then processes the data and responds to the browser so the end user can access the information.

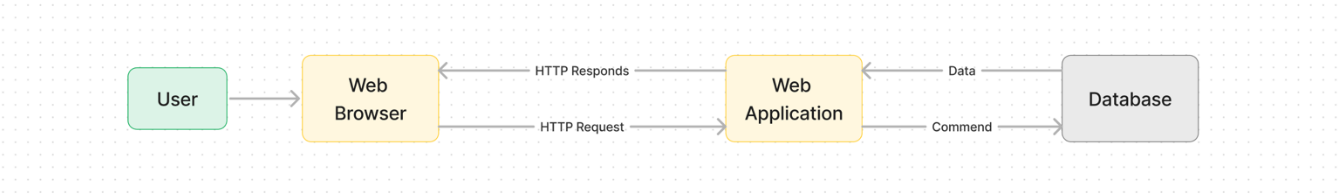


Figure 2

2.2) Database

The database of this project is written in SQLite3, named project.db and stored in myapp-2023 directory, which is divided into 5 tables, namely popularGames , userInfo, godOfWarAchivement, comments, bookmarks. I'll introduce their contents in order.

The popularGames table is used to store popular Playstation games. This includes 3 fields, the first is the ID of the game which is the primary key of the table, the second is the name of the game, and the third is the relative path of the game's icon file.

The userInfo table is used to store the user's personal information including the user's ID (primary key) username and hashed user's password.

The godOfWarAchivement, represents the table for storing game achievement information (due to time constraints, I only created the achievement information for God of War), there are four fields in this table, the first one is the achievement ID (primary key), the second one is the name of the achievement, the third one is the relative path of the achievement icon, and the last one is the game id, which is a foreign key linked to the popularGames table to prove which game the achievement belongs to.

The comments table is used to store user comments on game achievements. The table has 6 fields, the first is the comment id which is the primary key, the second is the content of the comment, and the remaining four are the user ID, username, achievement ID, and game ID which are all foreign keys connected to popularGame, userInfo, godOfWarAchivement to ensure that a comment is correctly recorded as being written by which person under which achievement in which game.

The bookmarks table is used to record the games bookmarked by the user. The table has 7 fields, the first one is the ID of the bookmark which is also the primary key of the table, the second one is the status of the bookmark because the website provides a feature that the user's bookmarks can have two statuses, the game is playing, and the game played. The user can change the status of the bookmarks at any time. The other 5 are foreign keys, they are divided into two groups, one group is the user ID and user name they are responsible for linking to userInfo to record which user the bookmark belongs to and the other group is game ID, game name and game icon they are responsible for linking to popularGames to record which game has been added to the bookmark.

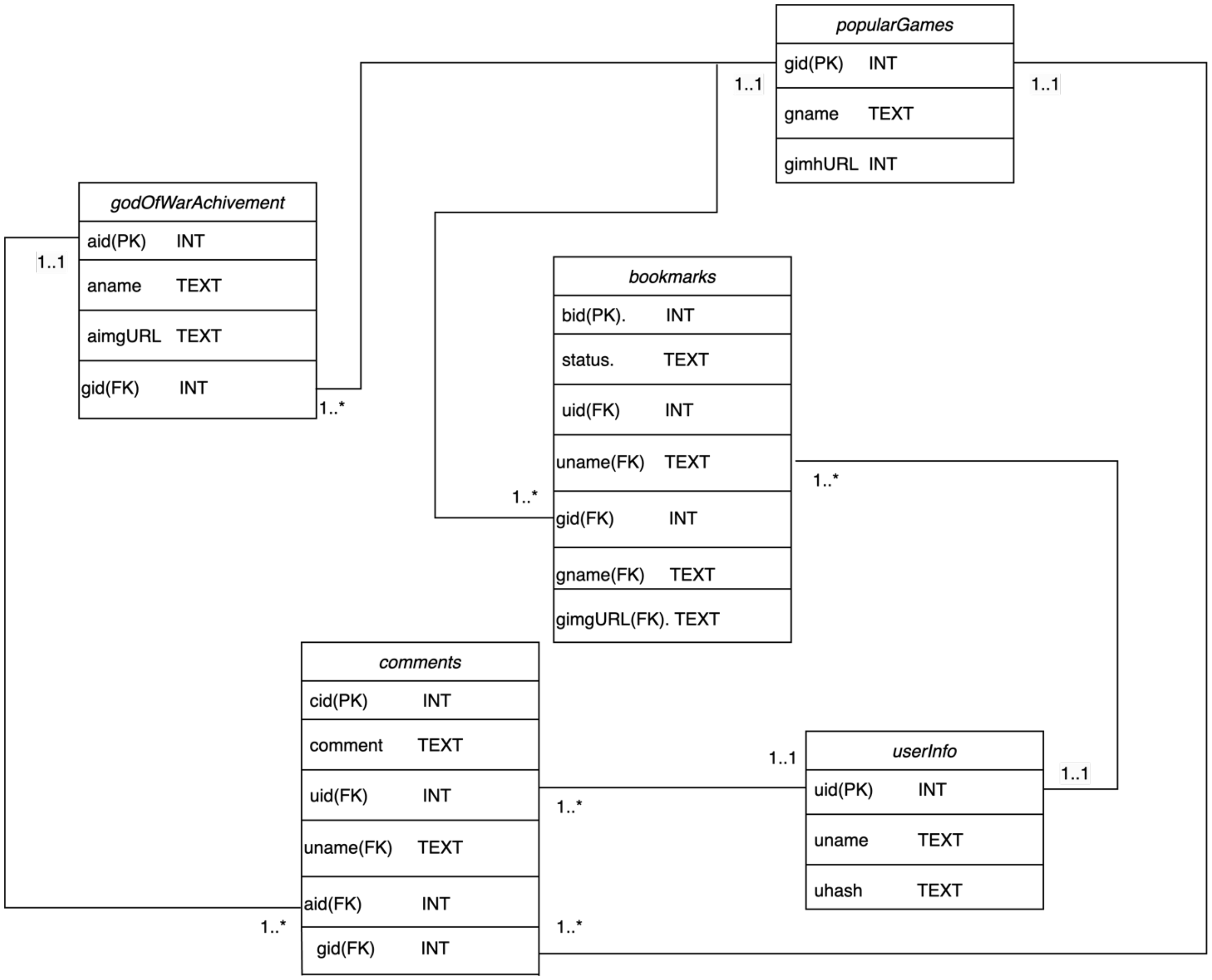


Figure 3

2.3) Graphical User Interface

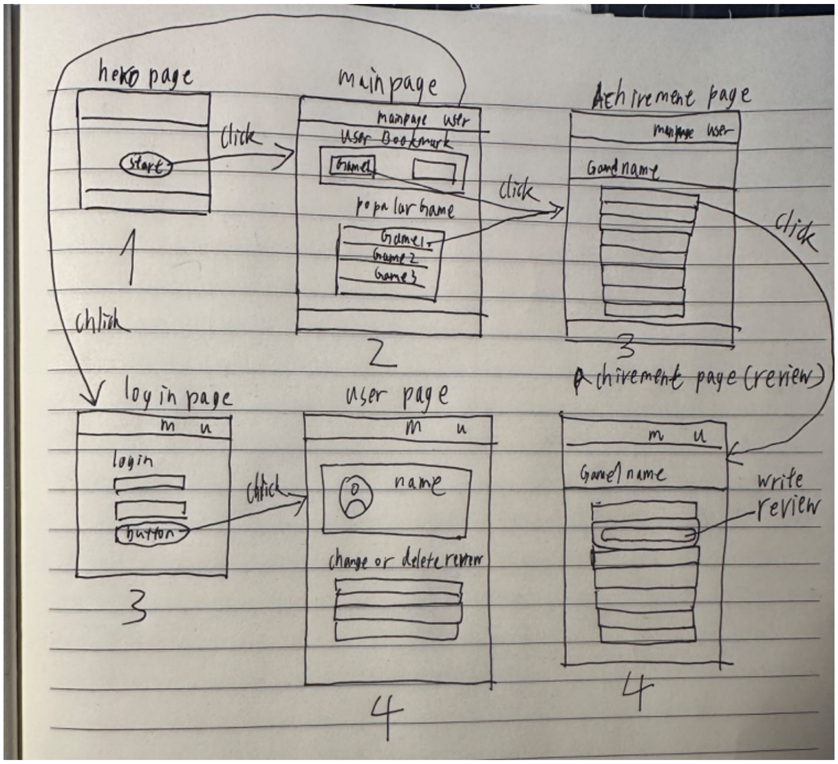
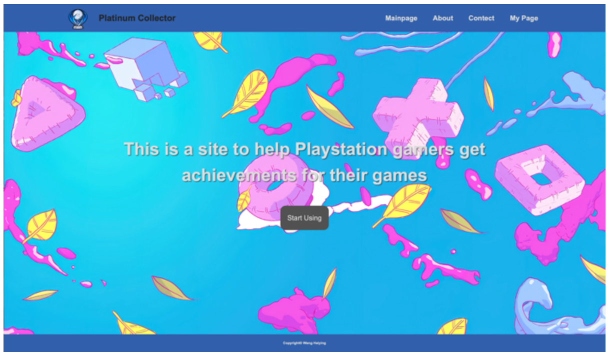


Figure 4

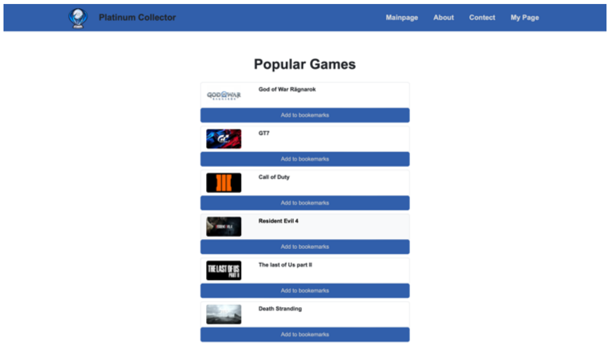
Figure 4 is the Lo-fi flow chart of this website. It briefly summarises the use process of the website. I will introduce it according to the order of the figure above and combine the screenshot of the website below

The site starts with Hero Image (Figure 5) and the user goes to the home page (Figure 6) by clicking on it. Here is all the game information. If the user is logged in (Figure 7), the user's bookmark will also be displayed, and the user can bookmark the game on this page to find it quickly the next time they use the site. In addition, when the user clicks on the game icon, the page will be redirected to the game Achievements page (Figure 8), which will display all the game's achievements. Users can click on an achievement icon to see what other people have written about the achievement or write a note for the achievement for other users to see. The second is about the user's personal page. Users can log in (Figure 9) or create an account on any page by clicking My page in the navigation bar. When a user logs in, the page will jump to his personal page (Figure 10). On this page, the user can change his password, but the length of the password must be more than 5 digits and the user can delete his account completely (the administrator account can't be deleted) by clicking Delete Account. In addition, the user can delete or edit their own comments on this page (administrator accounts have the highest level of access to delete and edit anyone's comments). Lastly, there are two static pages, about and contact. The About page includes an introduction to the site's features and a brief description of the designer to help users understand the site's functionality. The contact page contains the designer's contact information, which will help users interact with the designer.



This is a site to help Playstation gamers get achievements for their games

Figure 5



Popular Games

God of War Rägnarok

GT7

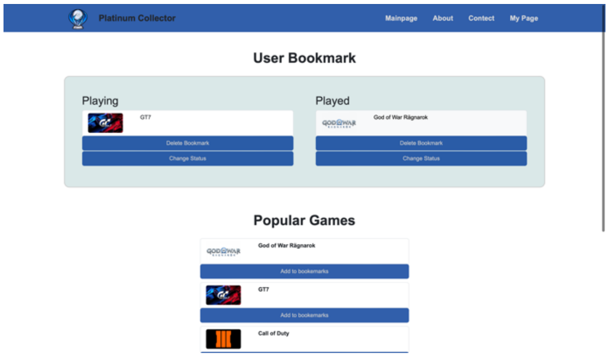
Call of Duty

Resident Evil 4

The last of Us part II

Death Stranding

Figure 6



User Bookmark

Playing

GT7

Played

God of War Rägnarok

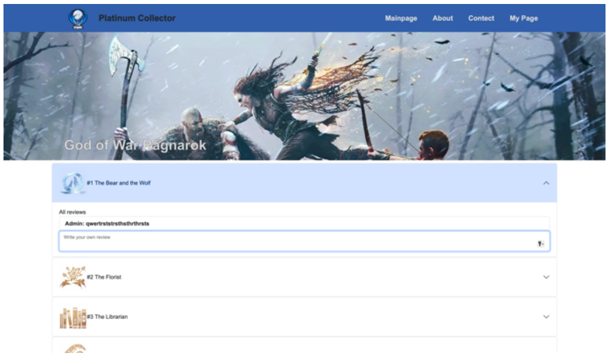
Popular Games

God of War Rägnarok

GT7

Call of Duty

Figure 7



#1 The Bear and the Wolf

All reviews

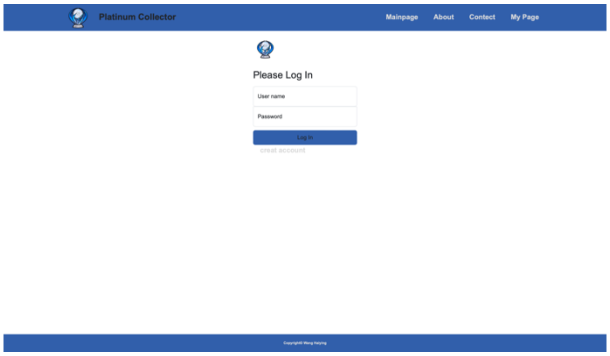
Admin: qwertrststrsthsthrthrsts

Write your own review

#2 The Florist

#3 The Librarian

Figure 8

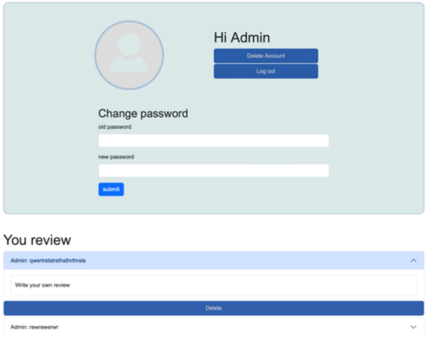


Please Log In

User name

Password

Figure 9



Hi Admin

Change password

old password

new password

You review

Admin: qwertrststrsthsthrthrsts

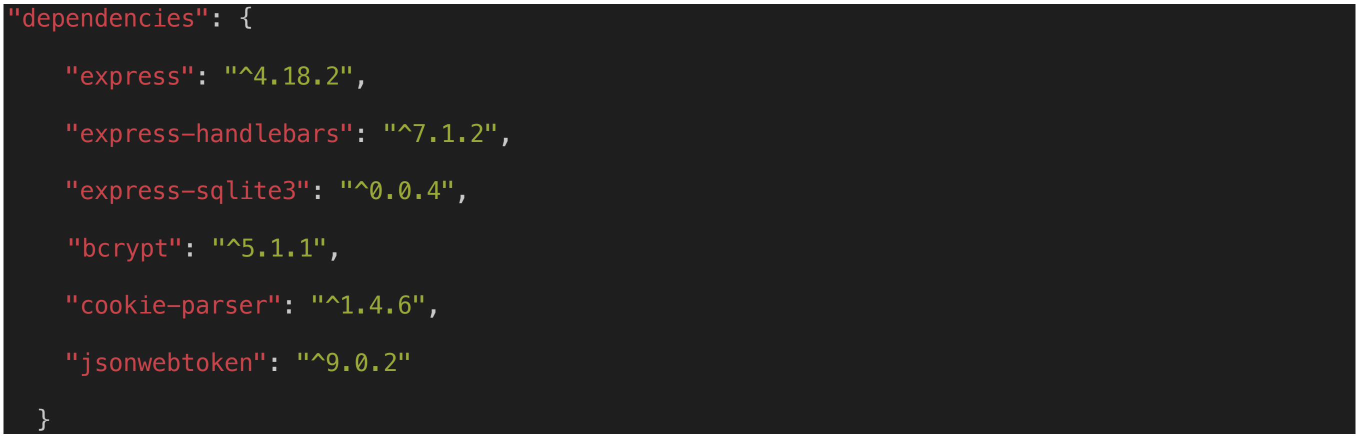
Write your own review

Admin: rewrewerwr

Figure 10

2.4) Web Application

The code for the web application is all stored in app.js in the myapp-2023 directory. It is based on Node.js and uses the express framework and I have installed 6 packages using npm.



```
"dependencies": {

    "express": "^4.18.2",

    "express-handlebars": "^7.1.2",

    "express-sqlite3": "^0.0.4",

     "bcrypt": "^5.1.1",

    "cookie-parser": "^1.4.6",

    "jsonwebtoken": "^9.0.2"

  }
```

express is a popular Node.js web application framework that enables designers to develop efficiently (Wilson, 2018). Therefore, I chose this framework for my website development.

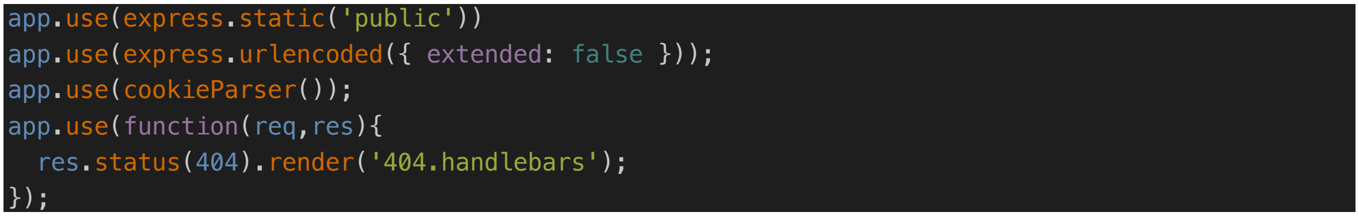
express-handlebars is a view engine for Express that allows the server to send data to the Handlebars template engine through res.render() to generate dynamic content in web pages.

express-sqlite3 is Express's SQLite3 database middleware for connecting to and manipulating SQLite3 databases. It allows me to manipulate databases in node.js.

cookie-parser is Express middleware that helps servers parse cookie data sent by clients.

bcrypt it is a password hash library that hashes the user's password so that the database stores the hashed password instead of the plaintext password. When a user logs in, the server uses bcrypt to hash the plaintext password submitted by the client and then compares it with the hash stored in the database to authenticate the user.

jsonwebtoken is a library for generating and validating JSON Web Tokens. JWT generates encrypted tokens and puts it in a cookie, which is sent to the server with a cookie on each request, allowing the server to authenticate the user with the token.



```
app.use(express.static('public'))
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(function(req,res){
  res.status(404).render('404.handlebars');
});
```

In addition, I registered four middleware. The first one serves static files in the public directory such as css files or images when the client requests it. The second one helps the server to parse the form data in the HTTP POST request. The third one is to apply the cookie-parser middleware mentioned above to the server program and the last one is to respond to a 404-error page to the client when there is a page that cannot be found

According to Tichy (1997), MVC is one of the most widely used software design patterns today and makes code easier to maintain. Therefore, this network application uses the MVC design pattern. The program is generally divided into 3 parts. First is Model by SQLite3 to write the project.db database file processing data. Secondly is the View which contains all the handlebar files in the view directory that render the user page using HTML code generated by the handlebar template engine. The final is the Controller controlled by the app.js server file written by the express framework to send Model to the View according to the client's request And corresponding to the client

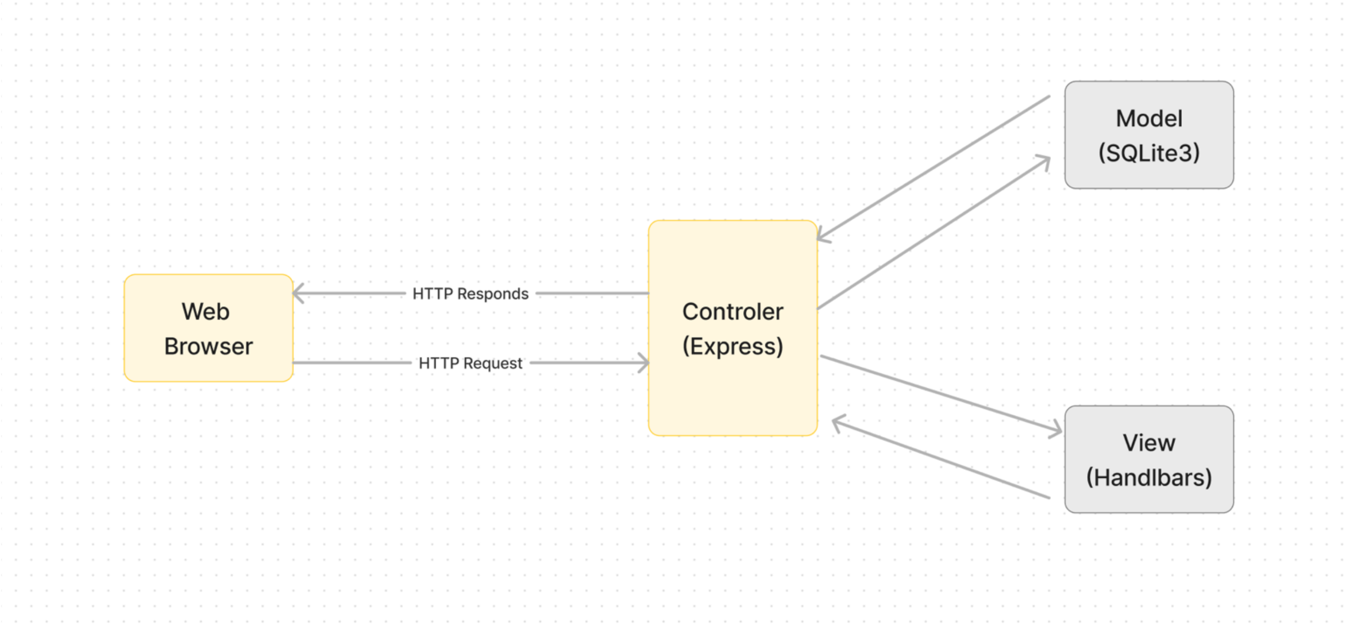


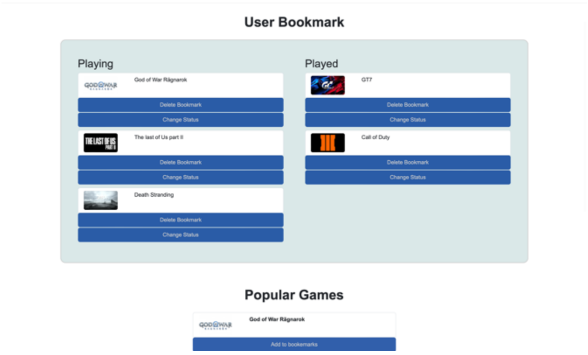
Figure 11

2.5) Security issues

The web application will also face a number of security issues. The first one is password storage. If nothing else, the user's plaintext password will be saved to the database, which will lead to user information leakage (Huseby, 2020). However, this can be used to hash the password to ensure that the password is not leaked. Another is the security of cookies. Because cookies store data in the client side, security problems will occur when cookies send user authentication information to the server. This is because sometimes a hacker may tamper with the data in the cookie to simulate the user to log in, but this can be prevented by putting the encryption token generated by the JWT into the cookie. To ensure the security of the website

3) Results

Based on the above methodology. I have successfully created Platinum Collector, a website for PlayStation gamers to share their experiences with each other and get achievements for their games. The site includes a hero page, a home page, an about page, a contact page, and a user page. On the home page, users can add any game in the list to their bookmarks (Figure 11) and change the status of the bookmarks. Users can click on the game's logo to enter the game's achievement information page to see what other people have written about the game's achievements or to upload their own comments about the achievements. On this page, since there are usually 30 - 40 achievements in a game, I added a paging feature (Figure 12) to make it easier for the user to read. In addition, the site also provides a login or registration feature after logging in the user is allowed to go to the user's personal page where the user can go to change the password or delete the account as well as modify their own comments.



User Bookmark

Playing

God of War Rágnarok

The last of Us part II

Death Stranding

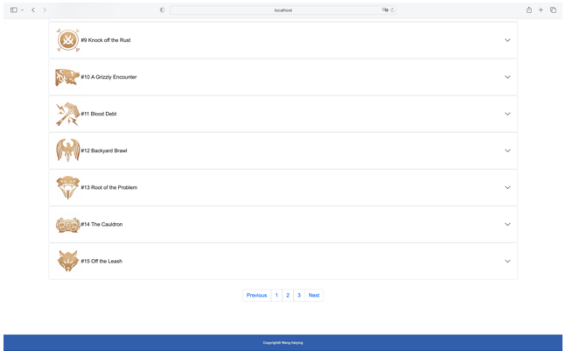
Played

GT7

Call of Duty

Popular Games

God of War Rágnarok



localhost

#9 Knock off the Rust

#10 A Grizzly Encounter

#11 Blood Debt

#12 Backyard Brawl

#13 Root of the Problem

#14 The Cauldron

#15 Off the Leash

Figure 11 **Figure 12**

There are two main things I did to improve the security of the site. Firstly, I used hashing to convert the user's plaintext password into an irreversible hash value, which prevents the user's plaintext password from being saved in the database and avoids leakage of the password. Secondly, I used JWT to generate tokens to help with authentication. Although I use a cookie in the authentication process, I put the encrypted token into a cookie, thus avoiding the security issues caused by human modification of cookie data. Also, according to Bugliesi et al. (2015), a cookie could add httpOnly to prevent cross-site scripting (XSS) attacks. So, I added (httpOnly: true) when generating the cookie to avoid this issue.

4) Discussion

Throughout the project, I learned how to complete a website project independently as a developer. First, before the website project starts, I need to plan, which includes making, and thinking about what functions should the website have, how many pages and how many tables should I need in the database, and what are the relationships between them. Secondly, I learned a lot of new programming knowledge in the process of website program writing, such as how to use Node.js and express framework, how to use npm installation package and registration middleware, and how to use MVC design pattern to divide the program into three parts to increase the maintainability of the code I pay attention to conducting security tests on the code to detect whether there are common security loopholes and ensure the security of the website. This project provided me with a useful experience to help me develop the website better in the future.

5) References

Bugliesi, M., Calzavara, S., Focardi, R., & Khan, W. (2015). CookiExt: Patching the browser against session hijacking attacks. Journal of Computer Security, 23(4), 509–537. https://doi.org/10.3233/JCS-150529

Huseby, S. H. (2004). Innocent code a security wake-up call for Web programmers.

Sony. (2023). Disability. www.sony.com https://www.sony.com/en/SonyInfo/IR/library/presen/er/archive.html

Tichy, W. F. (1997). A catalogue of general-purpose software design patterns. Proceedings of TOOLS USA 97. International Conference on Technology of Object Oriented Systems and Languages, 330–339. https://doi.org/10.1109/TOOLS.1997.654742

Wilson, E. (2018). MERN quick start guide : build web applications with MongoDB, Express.js, React, and Node (1st edition). Packt.

6) Appendices

Administrator name: Admin password:12345

User1 name: qwe   password:12345

User2 name: qwer   password:12345