ISA 681 Final Project
Chinese Checkers

4th May 2018

Collaborators:

Ashish Naik
G01095173
Dept. of Computer Science
George Mason University
anaik6@gmu.edu
Class of 2019

Rashika Koul
G01091688
Dept. of Computer Science
George Mason University
rkoul2@gmu.edu
Class of 2019

Introduction

Chinese Checkers is a strategy board game played by two, three, four or six players playing individually or with partners. Each player gets ten pieces of unique colour and the objective is to be the first to move all ten pieces across the board into the opposite starting corners. We have implemented a two-player version of Chinese Checkers also known as Diamond Chinese Checkers as the board looks like a diamond when there are two players as shown in figure 1.

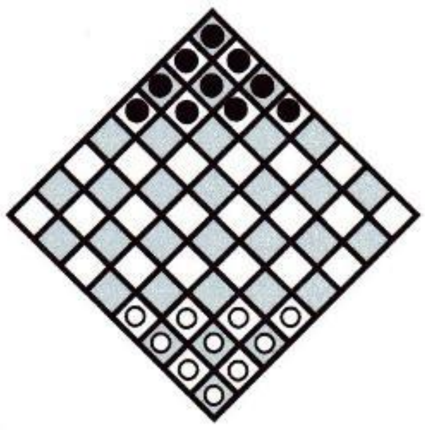


Figure 1

The web application has been developed in PHP and Node JS. XAMPP Server is used for database server and web server. This web application performs user authentication, facilitates online game playing, tracks the win, loss, draw statistics of the user.

Code Design

Following are the major components of the web application:

1) connection.php: This file contains the php code to establish a mySQL database connection.

2) server.js: This file contains the code to start a nodeJs server and establishes connection to the home.php file using socket.io and NodeJs. When user logs in, socket.io establishes a connection

3) signupform.php: This file establishes a connection to the database through connection.php. It performs input validation in the signup form and encrypts the password using SHA-256 hash before inserting it into the database.

4) signinconnection.php: This file establishes a connection to the database through connection.php. It decrypts the database entries for username and password and authenticates the user.

5) home.php: This file contains the game logic, socket connection and session timeout. It also displays the list of online players to play with and the win/loss statistics of the logged in user.

Technicalities

1) Language used:
This application has been developed in PHP and Node JS. Front end has been developed using css and bootstrap.

2) External Dependencies:
XAMPP Server: XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends. It consists of MySQL Database Server and Apache Web Server and interpreters for scripts written in PHP. Instead of XAMPP one can also download PHP, MySQL Database Server, Apache Web Server separately.

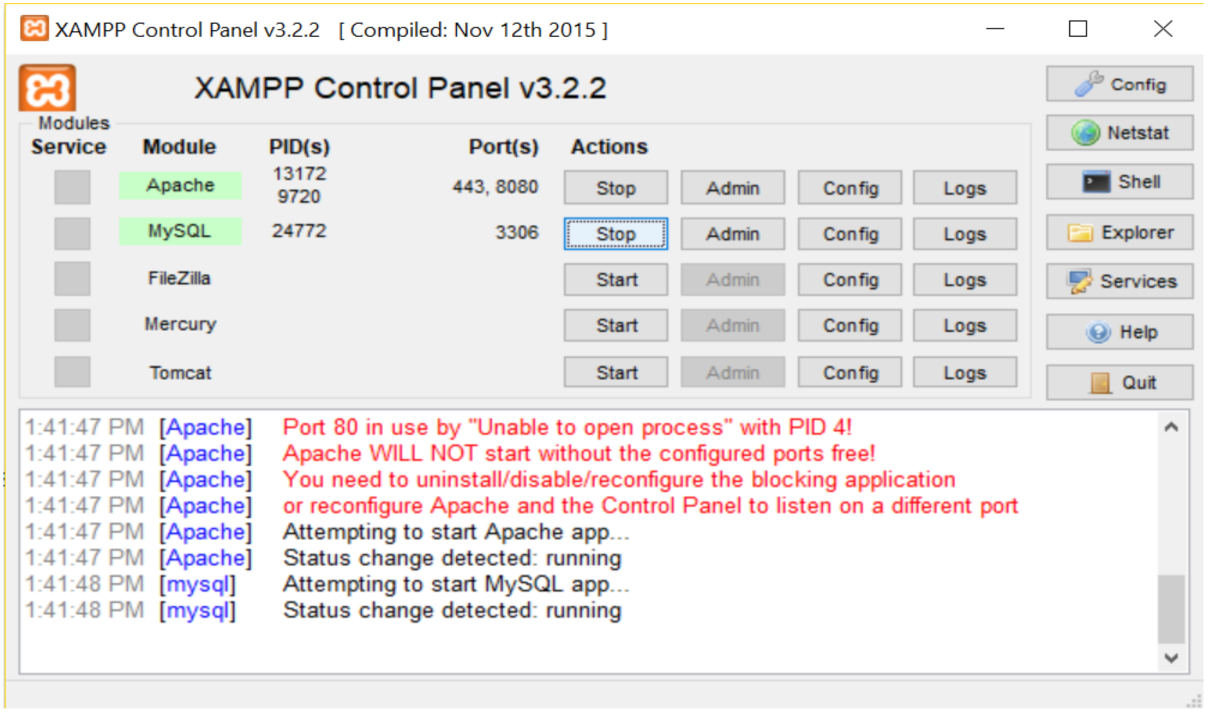
3) Installation:
Download the compressed folder from the blackboard, open it and place the 'clisev' folder in the htdocs folder inside the XAMPP folder and extract there. If one has installed PHP, MySQL Database Server, Apache Web Server separately then place the folder as follows:

1. Mac OS: Inside /Mac HD/Library/WebServer/Documents
2. Linux: Inside /var/www/html
3. Windows 10:
    1. On the start menu, type path and click Edit the system environment variables.
    2. Click Environment Variables at the bottom of the System Properties window and select PATH in the user variables list.
    3. Append your PHP Path to the PATH variable, separated by semicolon and click OK.

Instructions

One needs to follow the following procedure to get the application running on localhost:

1. Start servers: After the installation is done, start the XAMPP Server and have the Apache web server and MySQL database server running.



XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]
XAMPP Control Panel v3.2.2
Config
Netstat
Shell
Explorer
Services
Help
Quit
Modules

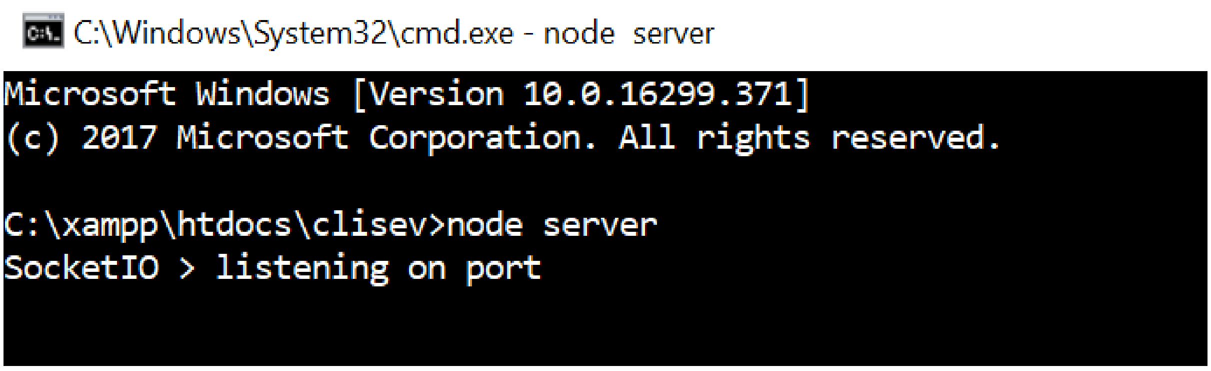
Service	Module	PID(s)	Port(s)	Actions			
	Apache	13172 9720	443, 8080	Stop	Admin	Config	Logs
	MySQL	24772	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
1:41:47 PM  [Apache]   Port 80 in use by "Unable to open process" with PID 4!
1:41:47 PM  [Apache]   Apache WILL NOT start without the configured ports free!
1:41:47 PM  [Apache]   You need to uninstall/disable/reconfigure the blocking application
1:41:47 PM  [Apache]   or reconfigure Apache and the Control Panel to listen on a different port
1:41:47 PM  [Apache]   Attempting to start Apache app...
1:41:47 PM  [Apache]   Status change detected: running
1:41:48 PM  [mysql]    Attempting to start MySQL app...
1:41:48 PM  [mysql]    Status change detected: running
```

If they are downloaded separately, run the following commands on the terminal to start them.

```
$ sudo service apache start
$ sudo service mysqld start
```

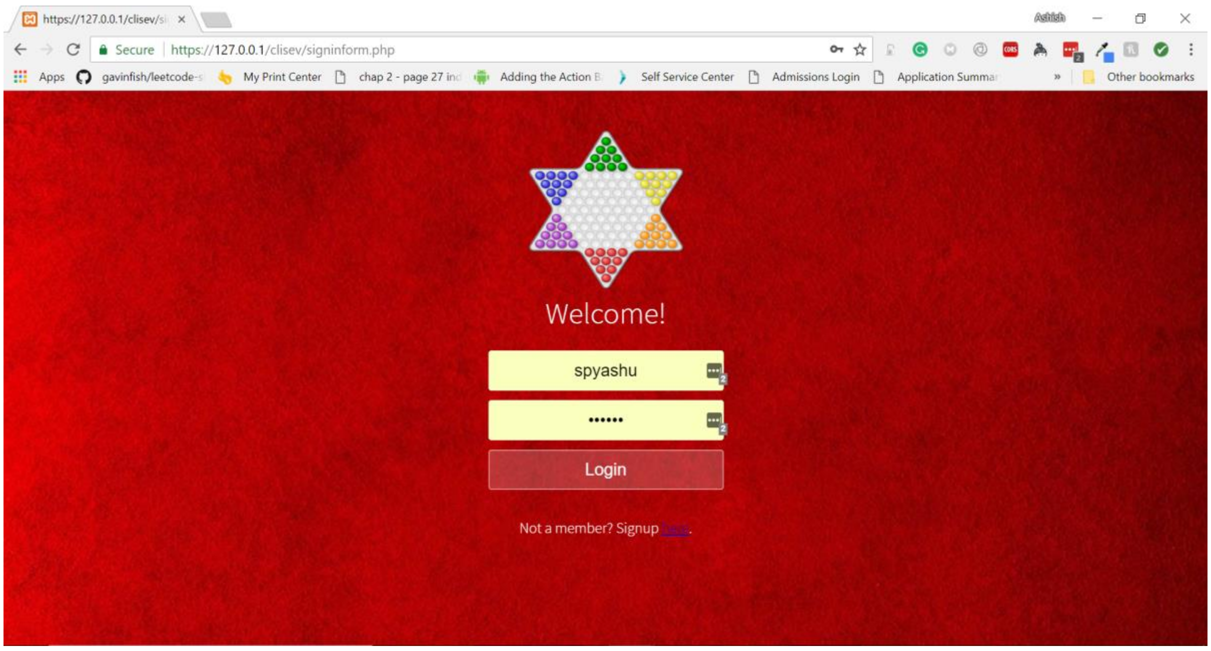
2. Start node js server: Type the command 'node server' in the command prompt.



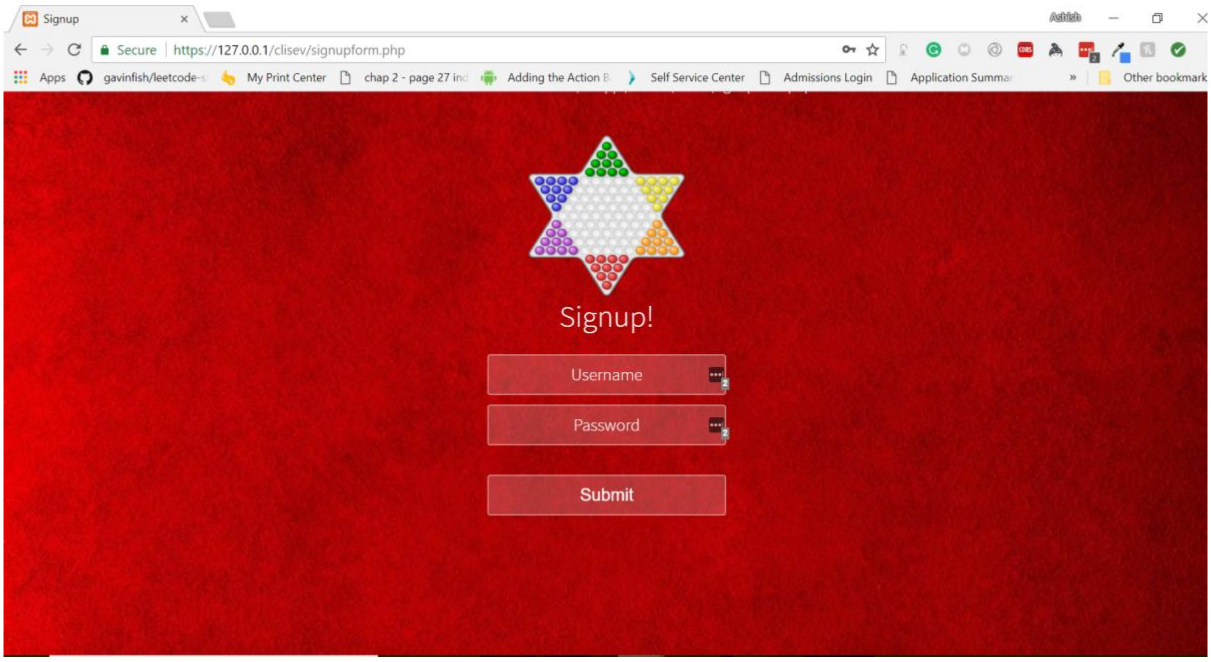
```
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\clisev>node server
SocketIO > listening on port
```

3. Signup: Go to https://localhost:8888/clisev/signinform.php and signup by clicking the 'here' button if you don't have an account already.

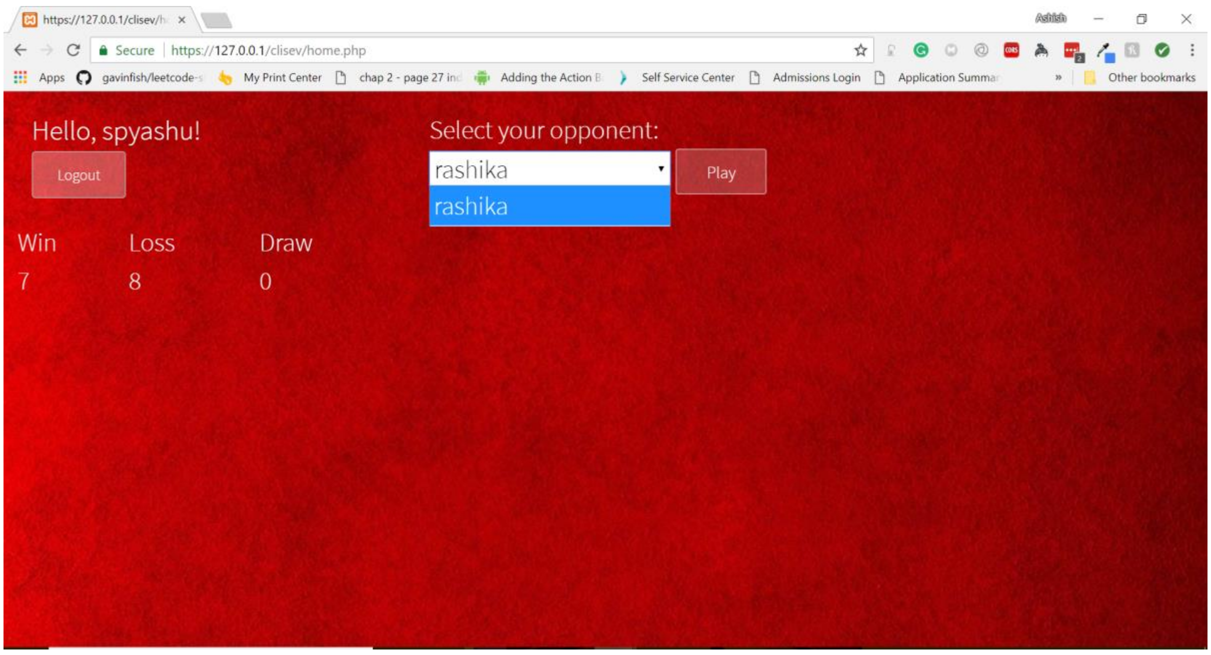

https://127.0.0.1/clisev/signinform.php
Welcome!
spyashu
••••••
Login
Not a member? Signup here.

The signup form is input validated so make sure to have only alphabets and numbers in the username and password. The password length should be between 6 to 20 character. The username length should be between 3 to 15 characters. 3.



4. Login: Once you are registered, you are required to login using the username and password that was set during registration.

5. Play: Select your opponent from the dropdown list and hit 'Play'. The drop-down list will show the players who are online. When you hit play, the Chinese checkers board appears, and you can start playing.



Win	Loss	Draw
7	8	0

6. Logout: Hit the ‘Logout’ button.

Game Rules

Chinese Checkers is a strategy board game played by two, three, four or six players playing individually or with partners. Either player may go first and the players take turns. Each player gets ten pieces of unique colour and the objective is to be the first to move all ten pieces across the board into the opposite starting corners, also referred to as “home”. As shown in the figure there are only two allowed moves that is, rolling and hopping. Rolling means simply moving one step in any direction to an adjacent empty space and hopping stands for jumping over an adjacent piece into a vacant space. Multiple continuous hops are allowed in one turn and the hops need not be in a straight line. A player may hop over his or her own pieces or the other player's pieces, one piece at a time. The first player to move all his or her pieces to the opposite starting position wins. We have implemented a two-player version of Chinese Checkers also known as Diamond Chinese Checkers.

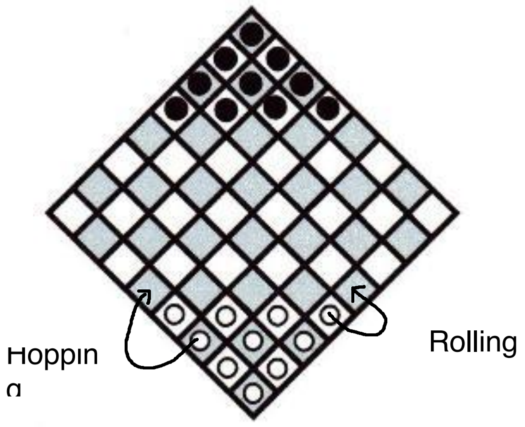


Figure 1

Thus, a very basic strategy can be to create or find the longest hopping path that leads closest to home, or immediately into it as multiple-jump moves are obviously faster to advance pieces than step-by-step moves. Since either player can make use of any hopping 'ladder' or 'chain' created, a more advanced strategy involves hindering an opposing player in addition to helping oneself make jumps across the board.

Assurance Case

We will now explain how the design implementation of this game is "secure". The following features make the game secure:

1) Prevents SQL Injection:
SQL injection is one of the most common web hacking techniques in which malicious code is placed in SQL statements, via web page input. This can destroy the database. We prevented SQL injections in all possible web page inputs in our website namely the sign in form and the signup form with the help of prepare() function which helps to bypass the meta characters in the input. The implementation is as follows...
signinconnection.php:

```
...
$query = $pdo->prepare('select * from member where uname=? and password=?');
$query->bindValue(1, $uname);
$query->bindValue(2, $hash);
$query->execute();
...
```

signupform.php:

```
...
$query = $pdo->prepare('select * from member where uname=?');
$query->bindValue(1, $uname);
$query->execute();
…
```

2) Performs server-side Input Validation:
Input validation also aids in preventing SQL Injection and buffer overflow attack and cross site scripting. We have performed input validation using regular expressions and the user inputs are validated against whitelist of allowed entries. Moreover, the length of the input is also validated. The password length should be between 6 to 20 characters. The username length should be between 3 to 15 characters. The implementation is as follows...
signupform.php:

```
...
$patternPassword = preg_match("@[A-Za-z0-9]+@", $pass);
```

```
$patternUname = preg_match("@[A-Za-z0-9]+@", $uname);
…
```

3) Prevents URL Injection:

Our application passes the parameters like username and password via a POST request. Thus, the parameters are not displayed in the URL thereby preventing URL Injection. POST is safer than GET because the parameters are not stored in the browser history or in web server logs. The implementation is as follows...

signupform.php:

```
…
If (!isset($_POST["uname"]) || !isset($_POST['password'])) {
echo "<script> alert('All fields must be filled out!');</script>";
…
```

signinconnection.php:

```
…
If (isset( $_POST['uname']) && isset($_POST['password'])) {
$uname = $_POST['uname'];
$password = $_POST['password'];
…
```

4) Prevents Buffer overflow attack:

A buffer overflow is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations. This must be prevented as it can be used by attackers to overwrite the passwords and usernames. We prevented the buffer overflow attack by checking the length of the input username and password string before inserting it into the database.

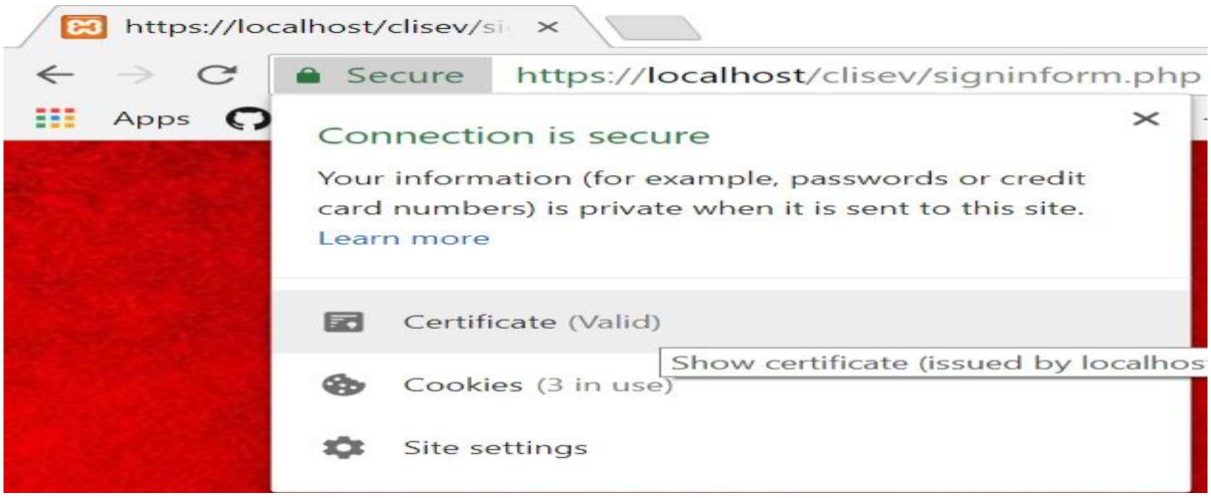
The implementation is as follows…

signupform.php:

```
…
if(!$patternUname||!$patternPassword||strlen($pass)<6||strlen($pass)>2
0||strlen($uname)<3||strlen($uname)>15)
…
```

5) SSL Certificate:

We authenticated the server using self-signed certificate generated by openSSL library which ensure that unauthorised access is prevented. HTTPS offers an extra layer of security because it uses the SSL protocol to move the data.


https://localhost/clisev/si

Secure https://localhost/clisev/signinform.php

Connection is secure

Your information (for example, passwords or credit card numbers) is private when it is sent to this site. Learn more

Certificate (Valid)

Show certificate (issued by localhos

Cookies (3 in use)

Site settings

6) Database Encryption:

Our application saves the password of the user in the encrypted form. This is done first by adding a salt value to the password and then performing password hashing using the SHA256 algorithm. This is great for protecting passwords, because we want to store passwords in a form that protects them even if the password file itself is compromised, but at the same time, we need to be able to verify that a user's password is correct while he or she is logging into the system. The implementation is as follows…

Encrypting in signupform.php:

```
…
$uname = $_POST['uname'];
$pass = $_POST['password'];
…
$options = ['cost'=> 12, 'salt' => "qazxswedcvfrtgbnhyujmkiolp",];
$hash = password_hash($pass, CRYPT_SHA256, $options);
$query= $pdo->prepare("insert into member(uname,password) values(?,?)");
$query->bindValue(1, $uname);
$query->bindValue(2, $hash);
$query->execute();
header('location:./signinform.php');
…
```

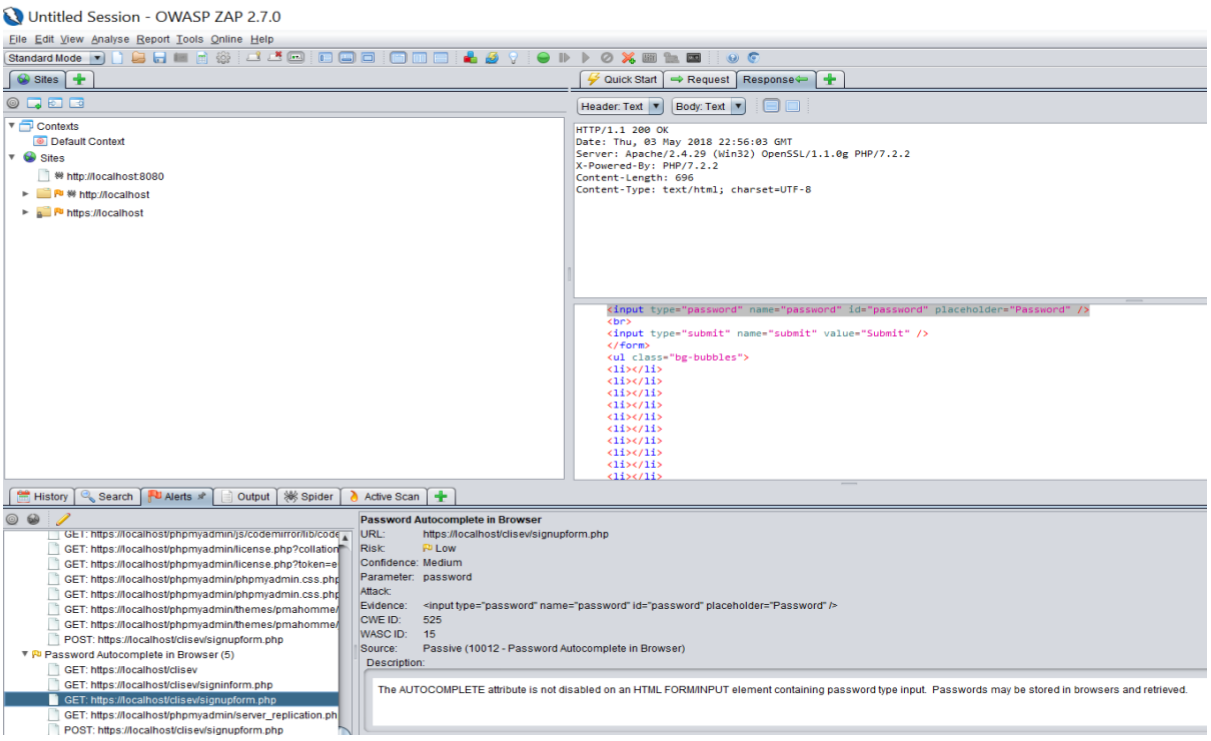
Decrypting in signinconnection.php:

```
…
$uname = $_POST['uname'];
$password = $_POST['password'];
```

```
$options = ['cost'=> 12, 'salt' => "qazxswedcvfrtgbnhyujmkiolp",];
$hash = password_hash($password, CRYPT_SHA256, $options);
$query = $pdo->prepare('select * from member where uname=? and password=?');
$query->bindValue(1, $uname);
$query->bindValue(2, $hash);
$query->execute();
…
```

7) Vulnerability tool:

We also used a dynamic analysis tool called OWASP ZAP and as per the analysis, almost all the results shown were categorized under low risk as we managed to eliminate most of the high risk vulnerabilities caused by not using post instead of get wherever required, prepare function, etc.



```
HTTP/1.1 200 OK
Date: Thu, 03 May 2018 22:56:03 GMT
Server: Apache/2.4.29 (Win32) OpenSSL/1.1.0g PHP/7.2.2
X-Powered-By: PHP/7.2.2
Content-Length: 696
Content-Type: text/html; charset=UTF-8
```

```
<input type="password" name="password" id="password" placeholder="Password" />
<br>
<input type="submit" name="submit" value="Submit" />
</form>
<ul class="bg-bubbles">
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
```

```
GET: https://localhost/phpmyadmin/js/codemirror/lib/code
GET: https://localhost/phpmyadmin/license.php?collation
GET: https://localhost/phpmyadmin/license.php?token=e
GET: https://localhost/phpmyadmin/phpmyadmin.css.php
GET: https://localhost/phpmyadmin/phpmyadmin.css.php
GET: https://localhost/phpmyadmin/themes/pmahomme/
GET: https://localhost/phpmyadmin/themes/pmahomme/
POST: https://localhost/clisev/signupform.php
Password Autocomplete in Browser (5)
    GET: https://localhost/clisev
    GET: https://localhost/clisev/signinform.php
    GET: https://localhost/clisev/signupform.php
    GET: https://localhost/phpmyadmin/server_replication.ph
    POST: https://localhost/clisev/signupform.php
```

Password Autocomplete in Browser

URL:	https://localhost/clisev/signupform.php
Risk:	Low
Confidence:	Medium
Parameter:	password
Attack:	
Evidence:	<input type="password" name="password" id="password" placeholder="Password" />
CWE ID:	525
WASC ID:	15
Source:	Passive (10012 - Password Autocomplete in Browser)

Description:

The AUTOCOMPLETE attribute is not disabled on an HTML FORM/INPUT element containing password type input. Passwords may be stored in browsers and retrieved.

8) Prevents Denial of Service attack:

If a player doesn't make a move for a very long time the opponent should not face a denial of service attack. We have prevented this by using the concept of inactive sessions where in if there is no activity by the player for a certain amount of time, he or she will automatically get logged out from the system. And, when a player is logged out during the game he or she loses the game. The implementation is as follows…

home.php:

```
…
$_SESSION['last_activity'] = time();
$_SESSION['expire_time'] = 1*60;
if( $_SESSION['last_activity'] < time()-$_SESSION['expire_time']) {
echo '<script type="text/javascript">',
'pleaseonLogout();',
'</script>';
}else $_SESSION['last_activity'] = time();
…
```

9) Prevents cheating:

A player cannot move its opponent's pieces and can only move its own pieces during its own turn. A player cannot make illegal moves either. This is demonstrated in the demo video.