**INTRODUCTION**

TACTICAL Chess is designed with the traditional rules of the games along with some modifications done in order to play the game better. In the normal chess games be it manual, offline or online it isn’t always possible to view the previous games to make the present game better. Thus, in TACTICAL Chess it would become possible to review and rate the previous games. It would then be possible to decide what the best move is and hence play it to win the game. Also, the best move can be generated, to facilitate the best possible move in the current game to receive the best possible output. Customisation of the game based on the current standings generate an improved look of the program.

Chess is a two-player strategy board game played on a chessboard. The game is played by millions of people worldwide. Each player begins with 16 pieces: one king, one queen, two rooks, two knights, two bishops, and eight pawns. Each of the six-piece types moves differently, with the most powerful being the queen and the least powerful the pawn. The objective is to checkmate the opponent's king by placing it under an inescapable threat of capture. To this end, a player's pieces are used to attack and capture the opponent's pieces, while supporting each other. In addition to checkmate, the game can be won by voluntary resignation of the opponent, which typically occurs when too much material is lost or checkmate appears unavoidable. There are also several ways a game can end in a draw.

Chess is believed to have originated in India sometime before the 7th century. The game was derived from the Indian game chaturanga, which is also the likely ancestor of the Eastern strategy games xiangqi, janggi, and shogi. A minority view holds that chess originated in China. The pieces assumed their current powers in Spain in the late 15th century the rules were finally standardized in the 19th century.

**DEFINITION OF THE PROBLEM**

The issue with normal chess games available online is that there is no possibility to rewind or review the game. The review module is an added advantage that this web-based application has as the previous moves generated by a player can be viewed at a later stage and sufficient clarifications regarding moves can be done. This is particularly useful in the case of learning or teaching the game. The weak point of the given player is projected and hence leads to fast correction. Moreover, the best move made by the player can also be analysed at a later-stage. The time constraints and characteristics of the game can then be cross-analysed to find out the best player in the future.

**FUNCTIONAL REQUIREMENT**

1. **Register: -** To play on the website with other users or with their friends they need to register first or Either they need to see their previous saved games.

* **Input:** Add personal details such as name, email id and password.
* **Output:** User id and password for login

1. **Login: -** To get into the main page they need to enter the user’s credentials and it should match with the database.

* **Input:**  User need to enter their credentials to login.
* **Output:**  users will redirect to the index page if users are credentials are match with the database or it will display an error.

1. **Forward and backward Move: -** Each player is given authority to do move one step forward or backward in the middle of the game.

* **Input:**  Click the forward or backward option
* **Output:**  One move forward or one move backward while playing in the middle of the game

1. **Replay game:** Each player is given authority to review and analyze the games that he has made. Thus, it’s possible to find out the best possible player and the move.

Select the rewind option: Allows player to rewind and review the game

* **Output:** The entire previous game is made viewable by the player.

### **NON-FUNCTIONAL REQUIREMENTS**

**Secured Login**: The user will be able to access the system only using authorized user id and password on.

**Data Usage Control**: The data visibility of the logged user will be limited to his/her role (access rights).

### **SOFTWARE AND HARDWARE REQUIREMENTS**

This section of the document contains a brief study of the hardware required for the system. As the proposed system is to be made into a computerized functional system, a computer is a must. The hardware requirements are listed below:

* P4 Process or above
* 512MB RAM or above
* 20GB HDD or above

Along with the hardware, it requires software to make a system as well as to run a system with the computer hardware. As the system is a web-based one, so the most important requirement is an internet connection and a web browser that can support trending UI frameworks.

**DESIGN MODEL**

1. **Data Flow Diagram**

* **Level 0:**

User

Tactical Chess

Chess

Database

* **Level 1 DFD:**

Moves

Login

User

Authorization

Replay

Saved games

games

New Game

Rewind

Move

Forward

Store Moves

**IMPLEMENTATION**

**Index.php**

<?php include "includes/header.php" ?>

<div class="game\_sec" >

<button class="startgamebtn btn" onclick="startgame(showIdx)" > Start Game</button>

<div class="left\_sec">

<h5 style="padding-left: 12%;">Users List</h5>

<?php

if(isset($\_COOKIE['userid']))

setcookie("userid", (string)$\_SESSION['uid'],time() + (86400 \* 30));

setcookie("username", $\_SESSION['username'],time() + (86400 \* 30));

?>

<hr style="border: 1px solid lightgrey;">

<ul id="usrdisplay"></ul>

</div>

<div class="mid\_sec">

<h1 id="gameidx"></h1>

<div class="chessboard">

<div class="white box" id="r8c1"><img id="b-r1" class="black-piece" src="Images/b-rook.png"></div>

<div class="black box" id="r8c2"><img id="b-k1" class="black-piece" src="Images/b-knight.png"></div>

<div class="white box" id="r8c3"><img id="b-b1" class="black-piece" src="Images/b-bishop.png"></div>

<div class="black box" id="r8c4"><img id="b-q1" class="black-piece" src="Images/b-queen.png"></div>

<div class="white box" id="r8c5"><img id="b-ki" class="black-piece" src="Images/b-king.png"></div>

<div class="black box" id="r8c6"><img id="b-b2" class="black-piece" src="Images/b-bishop.png"></div>

<div class="white box" id="r8c7"><img id="b-k2" class="black-piece" src="Images/b-knight.png"></div>

<div class="black box" id="r8c8"><img id="b-r2" class="black-piece" src="Images/b-rook.png"></div>

<!--2nd row !-->

<div class="black box" id="r7c1"><img id="b-p1" class="black-piece" src="Images/b-pawn.png"></div>

<div class="white box" id="r7c2"><img id="b-p2" class="black-piece" src="Images/b-pawn.png"></div>

<div class="black box" id="r7c3"><img id="b-p3" class="black-piece" src="Images/b-pawn.png"></div>

<div class="white box" id="r7c4"><img id="b-p4" class="black-piece" src="Images/b-pawn.png"></div>

<div class="black box" id="r7c5"><img id="b-p5" class="black-piece" src="Images/b-pawn.png"></div>

<div class="white box" id="r7c6"><img id="b-p6" class="black-piece" src="Images/b-pawn.png"></div>

<div class="black box" id="r7c7"><img id="b-p7" class="black-piece" src="Images/b-pawn.png"></div>

<div class="white box" id="r7c8"><img id="b-p8" class="black-piece" src="Images/b-pawn.png"></div>

<!-- 7th row !-->

<div class="white box" id="r2c1"><img id="w-p1" class="white-piece" src="Images/w-pawn.png"></div>

<div class="black box" id="r2c2"><img id="w-p2" class="white-piece" src="Images/w-pawn.png"></div>

<div class="white box" id="r2c3"><img id="w-p3" class="white-piece" src="Images/w-pawn.png"></div>

<div class="black box" id="r2c4"><img id="w-p4" class="white-piece" src="Images/w-pawn.png"></div>

<div class="white box" id="r2c5"><img id="w-p5" class="white-piece" src="Images/w-pawn.png"></div>

<div class="black box" id="r2c6"><img id="w-p6" class="white-piece" src="Images/w-pawn.png"></div>

<div class="white box" id="r2c7"><img id="w-p7" class="white-piece" src="Images/w-pawn.png"></div>

<div class="black box" id="r2c8"><img id="w-p8" class="white-piece" src="Images/w-pawn.png"></div>

<!-- 8th row !-->

<div class="black box" id="r1c1"><img id="w-r1" class="white-piece" src="Images/w-rook.png"></div>

<div class="white box" id="r1c2"><img id="w-b1" class="white-piece" src="Images/w-bishop.png"></div>

<div class="black box" id="r1c3"><img id="w-k1" class="white-piece" src="Images/w-knight.png"></div>

<div class="white box" id="r1c4"><img id="w-q1" class="white-piece" src="Images/w-queen.png"></div>

<div class="black box" id="r1c5"><img id="w-ki" class="white-piece" src="Images/w-king.png"></div>

<div class="white box" id="r1c6"><img id="w-k2" class="white-piece" src="Images/w-knight.png"></div>

<div class="black box" id="r1c7"><img id="w-b2" class="white-piece" src="Images/w-bishop.png"></div>

<div class="white box" id="r1c8"><img id="w-r2" class="white-piece" src="Images/w-rook.png"></div>

</div>

</div>

<!-- .left\_sec -->

<div class="right\_sec">

<div>

<form method="post" id="gameidform">

<input type="text" palceholder="Enter Game ID">

<button type="submit">Sub</button>

</form>

</div>

<div class="stepsbox">

<div class="timer">

<span class="gametime" id="blacktime">00:30</span>

</div>

<div class="stepsContainer" >

<div class="stepsControlBtns">

<button type="button" value="backward" onclick="return forward()">

<i class="material-icons">skip\_previous</i>

</button>

<button type="button" value="forward" onclick="return backward()">

<i class="material-icons">skip\_next</i>

</button>

<?php if (isset($\_GET['gameid'])): ?>

<button type="button" value="forward" onclick="return rewindgame(<?= $\_GET['gameid'] ?>)">

Replay Game

</button>

<?php endif; ?>

</div>

<hr style="margin:0">

<div class="steps" id="stepsshow">

<table>

<tr>

<th>Drag From</th>

<th>Drop To</th>

<th>Drag Ele</th>

</tr>

</table>

<table id="stepstable">

</table>

</div>

</div>

<div class="timer">

<span class="gametime" id="whitetime">00:30</span>

</div>

</div>

<!-- .stepsbox -->

</div>

<!-- .right\_sec -->

</div>

<!-- .game\_sec -->

<script>

// $(document).ready(()=>{

document.addEventListener("DOMContentLoaded", function(){

$ = jQuery;

showIdx = function(id){

$("#gameidx").html(id)

}

$("form").on("submit", function(ev){

ev.preventDefault();

gameId = $("#gameidform input").val();

prepare\_pieces();

})

data = [];

moveInt = setInterval(function() {

console.log("AllidMoves")

$.post("./database/allidmoves.php", {gameid: gameId})

.always(function(d){

// console.log(d);

d = JSON.parse(d);

// console.log(d.data)

if(d.data.length%2)

turn = 1;

else

turn = 0;

random\_move();

if(data.length != d.data.length){

// strMove = curMove.dragFrom + curMove.dropto;

for(i = data.length-1; i < d.data.length; i++){

if(i==-1)i=0;

curMove = d.data[i];

theAmazingMover(curMove.dragfrom, curMove.dropto)

}

console.log(curMove.dragfrom, curMove.dropto)

{"id":"1048","uid":"8","gameid":"24","dragfrom":"r2c6","dropto":"r4c6","dragele":"w-p6"}

}

data.push( curMove);

})

}, 500);

})

function theAmazingMover(from, to){

console.log(from)

var from\_img = document.getElementById(from);

var to\_img = document.getElementById(to);

console.log(from\_img, to\_img)

var piece\_id = document.getElementById(from\_img.childNodes[0].id);

console.log("to\_img:" +to\_img);

console.log(from\_img);

console.log(piece\_id);

piece\_id.remove(piece\_id);

var fur\_node = to\_img.appendChild(piece\_id);

console.log("fur\_node:" +fur\_node);

var parent = document.getElementById(to\_img.id);

console.log("parent moving: " +parent.id);

if(parent)

child = document.getElementById(parent.childNodes[0].id)

if(parent != null)

{

parent.replaceChild(fur\_node,child);

}

console.log("delet: "+piece\_id);

}

</script>

<?php include "includes/footer.php"

?>

**Creating Cookies**

<?php

if (isset($\_COOKIE['userid']) && isset($\_COOKIE['username'])) {

$\_SESSION['username'] = $\_COOKIE['username'];

$\_SESSION['uid'] = (int)$\_COOKIE['userid'];

exit(header("location: index.php"));}?>

**Creating Session**

<?php

session\_start();

setcookie("username", "", time() - 3600);

require "./database/dataconnect.php";

$con->query("update login set active='not active' where uid='" . $\_SESSION['uid'] . "'");

unset($\_SESSION['username']);

unset($\_SESSION['uid']);

session\_destroy();

header("Location: login.php");

?>

**//moves.js**

var turn = 0

var row,col;

var node;

var black=16,white=16 ;

var validMoves=[];

var dragfrom, dropto, dragele;

var whitetimeout, blacktimeout;

var select=0;

var stopfun;

var gameId;

function refresh() {

return "Your custom message.";

}

**//AJAX CODE FOR DISPLAY THE MOVES ONTO THE SAME PAGE**

var xhttp = new XMLHttpRequest();

function active() {

var display = document.getElementById('usrdisplay');

xhttp.onreadystatechange = function() {

if(this.readyState == 4 && this.status == 200){

display.innerHTML = this.responseText;

}

};

xhttp.open("GET", "./database/useractive.php", true);

xhttp.send();

return false;

}

setInterval(function() {

active();

},1000);

function rewind(rewindgame\_id) {

console.log("foward moving");

var steps = document.getElementById("stepstable");

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

var temp = "";

var data = JSON.parse(this.responseText);

var tmpdata = data.data;

console.log(tmpdata);

tmpdata.forEach(element => {

temp += element;

});

console.log("Data coming: "+temp)

var from = temp.slice(select,select+4);

var to = temp.slice(select+4,select+8);

select=select+8;

console.log("select : " + select);

console.log("from : " + from + " to : " + to);

var from\_img = document.getElementById(from);

var to\_img = document.getElementById(to);

var piece\_id = document.getElementById(from\_img.childNodes[0].id);

console.log("to\_img:" +to\_img);

console.log(from\_img);

console.log(piece\_id);

piece\_id.remove(piece\_id);

var fur\_node = to\_img.appendChild(piece\_id);

console.log("fur\_node:" +fur\_node);

var parent = document.getElementById(to\_img.id);

console.log("parent moving: " +parent.id);

if(parent)

child = document.getElementById(parent.childNodes[0].id)

if(parent != null)

{

parent.replaceChild(fur\_node,child);

}

console.log("delet: "+piece\_id);

}

};

xhttp.open("POST", "./database/rewindmove.php");

xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded")

xhttp.send("rewindgame\_id=" + rewindgame\_id);

}

function rewindgame(rewindgame\_id) {

console.log("rewind game id : " + rewindgame\_id);

xhttp.open("POST", "./database/rewindmove.php");

xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded")

xhttp.send("rewindgame\_id=" + rewindgame\_id);

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

var data = JSON.parse(this.responseText);

console.log(data);

var datacount = data.rowcount;

console.log("count: "+datacount);

var intervalID = setInterval(function () {

if (--datacount <= -1) {

window.clearInterval(intervalID);

alert("no more rows");

}

rewind(rewindgame\_id);

}, 1000);

}

}

}

function backward() {

console.log("backward moving");

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

var temp = this.responseText;

console.log("Data coming: "+temp)

var from = temp.slice(0,4);

var to = temp.slice(4,8);

console.log("from : " + from + " to : " + to);

var to\_img = document.getElementById(to);

var from\_img = document.getElementById(from);

var piece\_id = document.getElementById(to\_img.childNodes[0].id);

console.log("to\_img:" +to\_img);

console.log(from\_img);

console.log(piece\_id);

piece\_id.remove(piece\_id);

from\_img.appendChild(piece\_id);

console.log("delet: "+piece\_id);

}

};

xhttp.open("GET", "./database/forward.php", true);

xhttp.send();

return false;

}

function forward() {

console.log("foward moving");

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

var temp = this.responseText;

console.log("Data coming: "+temp)

var from = temp.slice(0,4);

var to = temp.slice(4,8);

console.log("from : " + from + " to : " + to);

var from\_img = document.getElementById(from);

var to\_img = document.getElementById(to);

var piece\_id = document.getElementById(from\_img.childNodes[0].id)

console.log("to\_img:" +to\_img);

console.log(from\_img);

console.log(piece\_id);

piece\_id.remove(piece\_id);

to\_img.appendChild(piece\_id);

console.log("delet: "+piece\_id);

}

};

xhttp.open("GET", "./database/forward.php", true);

xhttp.send();

return false;

}

//DRAG FROM AND DROP DOWN SHOWN IN DIV

function send(dragfrom,dropto) {

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

}

else {

console.log(this.statusText);

}

};

xhttp.open("POST", "./database/storemoves.php", true);

xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

xhttp.send("drag\_from="+dragfrom+"&drop\_to="+dropto+"&drag\_ele="+dragele);

setTimeout(function(){ display(); }, 200);

}

**//DISPLAY FUNCTION**

function display() {

console.log("displaying");

var steps = document.getElementById("stepstable");

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

steps.innerHTML = this.responseText;

}

};

xhttp.open("GET", "./database/display.php", true);

xhttp.send();

}

**//SETTING ATTRIBUTE 64 BOX**

function startgame(show\_my\_id){

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200)

gameId = this.responseText;

show\_my\_id(gameId);

}

prepare\_pieces();

}

function prepare\_pieces(){

random\_move()

for (var i=0; i<64; i++) {

document.getElementsByClassName('box')[i].setAttribute("ondragover", "allowDrop(event)"); document.getElementsByClassName('box')[i].setAttribute("ondrop","drop(event)");

if(i<16)

{

document.getElementsByClassName('black-piece')[i].setAttribute("ondragstart","drag(event)");

document.getElementsByClassName('black-piece')[i].setAttribute("draggable","true");

document.getElementsByClassName('white-piece')[i].setAttribute("draggable","true");

document.getElementsByClassName('white-piece')[i].setAttribute("ondragstart","drag(event)");

}

}

timeout(30)

}

function allowDrop(ev) {

ev.preventDefault();

var node = ev.target.id;

}

function drag(ev) {

console.log("dragged from : " + ev.target.parentNode.id);

drag\_from = ev.target.parentNode.id;

var parent = document.getElementById(ev.target.parentNode.id);

ev.dataTransfer.setData("text", ev.target.id);

console.log("dragged element : " + ev.target.id);

drag\_ele = ev.target.id;

dragele = drag\_ele;

dragfrom = drag\_from;

if(ev.target.id.includes("w-p")||ev.target.id.includes("b-p"))

{

validMoves=pawnMove(ev.target.parentNode.id)

}

else if (ev.target.id.includes("w-k")||ev.target.id.includes("b-k"))

{

validMoves=kingMoves(ev.target.parentNode.id)

}

else

{

validMoves=elephantMove(ev.target.parentNode.id)

}

console.log(validMoves)

}

function drop(ev) {

if(validMoves.includes(ev.target.id) || validMoves.includes(ev.target.parentNode.id))

{

ev.preventDefault();

var drop\_to = (ev.target.parentNode.id == "") ? ev.target.id : ev.target.parentNode.id;

console.log("dropp to : " + drop\_to + " drop\_to\_parent : " + ev.target.parentNode.id);

var data = ev.dataTransfer.getData("text");

console.log("drop data " + data);

var droppedImg = document.getElementById(data);

console.log("dropped img : " + droppedImg);

var move = ev.target.appendChild(document.getElementById(data));

var further\_node = document.getElementById(ev.target.parentNode.id);

console.log("move ele " + move);

var child\_node;

if(further\_node)

child\_node = document.getElementById(further\_node.childNodes[0].id);

console.log("child node " + child\_node);

if(further\_node != null)

{

var cap\_pawn=further\_node.childNodes[0].id;

further\_node.replaceChild(move,child\_node)

if(cap\_pawn.includes("w-ki"))

{

alert("Black Wins");

}

else if(cap\_pawn.includes("b-ki"))

{

alert("White Wins")

}

}

if(turn==0)

{ turn=1;

}

else

{ turn=0;

}

dropto = drop\_to;

send(dragfrom,dropto,dragele);

random\_move();

timestop();

timeout(30);

}

else

{

alert("play valid move" +ev.target.id);

}

}

function random\_move()

{ if(turn==0)

{ for(var i=0;i<16;i++)

{ document.getElementsByClassName('black-piece')[i].setAttribute("draggable","false");

document.getElementsByClassName('white-piece')[i].setAttribute("draggable","true");

}

}

else

{ for(var i=0;i<16;i++)

{ document.getElementsByClassName('white-piece')[i].setAttribute("draggable","false");

document.getElementsByClassName('black-piece')[i].setAttribute("draggable","true");

}

}

}

**//chkusername**

function checkUsername() {

var xhttp = new XMLHttpRequest();

var username = document.getElementById("user").value;

console.log("username :" + username);

xhttp.onreadystatechange = function () {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("usernameError").innerHTML = this.responseText;

// console.log(this.responseText);

} };

xhttp.open("POST","./database/chkdata.php",true);

xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

xhttp.send("uname="+username);

}

**//chkuser.php**

<?php

session\_start();

require "dataconnect.php";

$user = $\_POST['user'];

$pswd = $\_POST['pswd'];

$search\_data = "select \*from login where username = '$user' and password = '$pswd'";

$search\_result = $con->query($search\_data);

if($search\_result->num\_rows > 0)

{

while($user = $search\_result->fetch\_assoc()) {

$\_SESSION['username'] = $user['username'];

$\_SESSION['uid'] = $user['uid'];

$con->query("update login set lastlogin='" . $user['userlog']. "' where uid='" . $\_SESSION['uid'] . "'");

}

$update\_data = "update login set active='active' where uid='" . $\_SESSION['uid'] . "'";

$con->query($update\_data);

setcookie("username", "some value");

echo "true";

}

else

{

echo 'User name does not exist';

}

?>

**//datatransfer.php**

<?php

session\_start();

require "dataconnect.php";

$user = $\_POST['user'];

$pswd = $\_POST['pswd'];

$search\_data = "select \*from login where username = '$user' and password = '$pswd'";

$search\_result = $con->query($search\_data);

if($search\_result->num\_rows > 0)

{

while($user = $search\_result->fetch\_assoc()) {

$\_SESSION['username'] = $user['username'];

$\_SESSION['uid'] = $user['uid'];

$con->query("update login set lastlogin='" . $user['userlog']. "' where uid='" . $\_SESSION['uid'] . "'");

}

$update\_data = "update login set active='active' where uid='" . $\_SESSION['uid'] . "'";

$con->query($update\_data);

setcookie("username", "some value");

echo "true";

}

else

{

echo 'User name does not exist';

}

?>

**//display.php**

<?php

session\_start();

require "dataconnect.php";

$uid = $\_SESSION['uid'];

$show\_moves = "select \*from moves where uid=" . $uid . " ORDER BY id DESC";

$rslt = $con->query($show\_moves);

if($rslt->num\_rows > 0)

{

while ($row=$rslt->fetch\_assoc()) {

echo "<tr>";

echo "<td>";

echo "" . $row['dragfrom'];

echo " </td>";

echo "<td>";

echo "" . $row['dropto'];

echo " </td>";

echo "<td>";

echo "" . $row['dragele'];

echo " </td>";

echo "</tr>";

}

}

?>

**//storemoves.php**

<?php

session\_start();

require "dataconnect.php";

$dragfrom = $\_POST['drag\_from'];

$dropto = $\_POST['drop\_to'];

$dragele = $\_POST['drag\_ele'];

$uid = $\_SESSION['uid'];

$gameid=$\_SESSION['gameId'];

$insert\_data = "insert into moves(uid,dragfrom,dropto,dragele, gameid) values('$uid','$dragfrom','$dropto','$dragele', '$gameid')";

if ($con->query($insert\_data) == false)

echo "error in insert" . $con->error;

?>

**//countgame.php**

<?php

session\_start();

require "dataconnect.php";

$uid = $\_SESSION['uid'];

$show\_moves = "select \*from game where uid=" . $uid;

$rslt = $con->query($show\_moves);

if($rslt->num\_rows > 0)

{

while ($row=$rslt->fetch\_assoc()) {

echo "<br> uid = " . $row['uid'] . " and game id : " . $row['gameid'];

}

}

?>

**//login.js**

function login(e){

e.preventDefault();

var user = document.getElementById('user').value;

var pswd = document.getElementById('pswd').value;

console.log("user ab: "+user);

var xhttp = new XMLHttpRequest();

if(user =="" || pswd =="")

{

console.log("Please Enter the data")

}

else

{

console.log("user: "+user)

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

if (this.responseText == "true")

window.location.replace("./index.php");

else

document.getElementById("show\_error").innerHTML = this.responseText;

}

};

xhttp.open("POST", "./database/chkuser.php", true);

xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

xhttp.send("user="+user+"&pswd="+pswd);

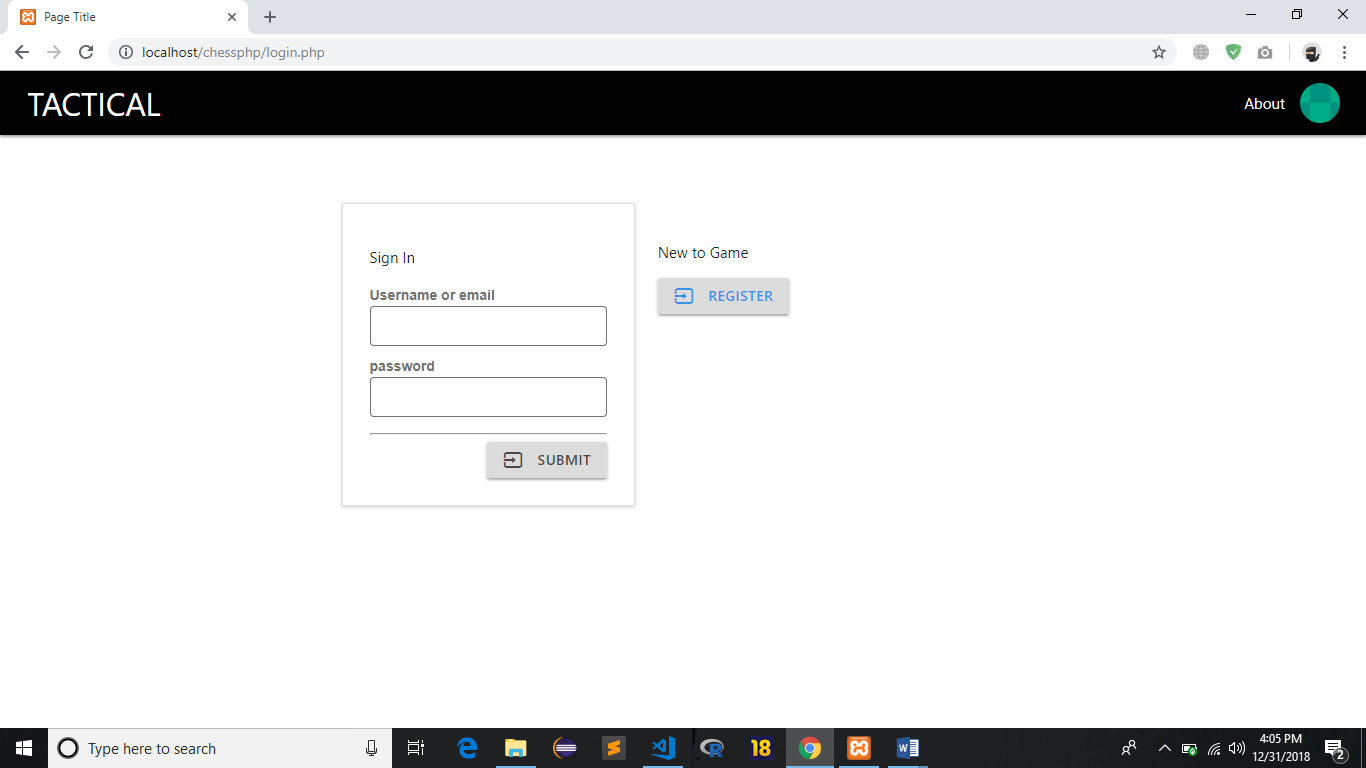
}

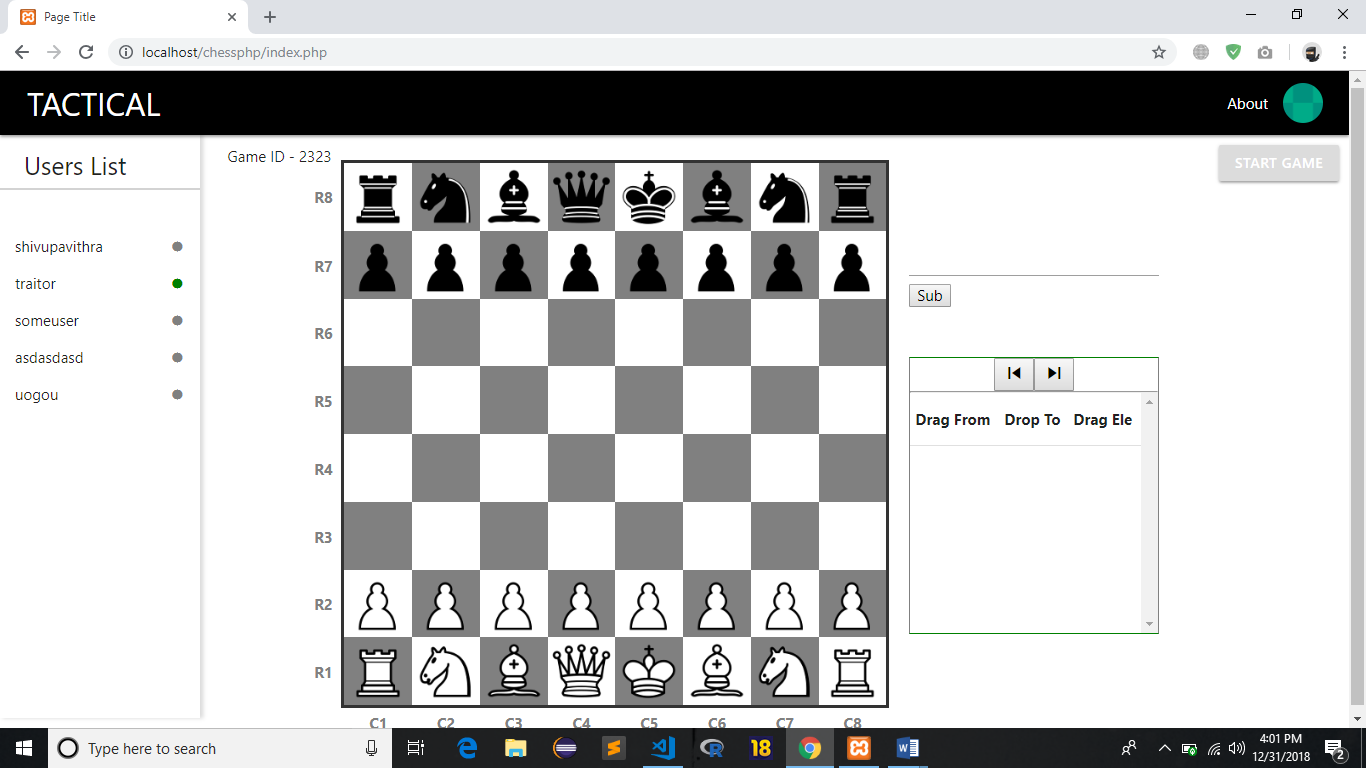
return false;

}

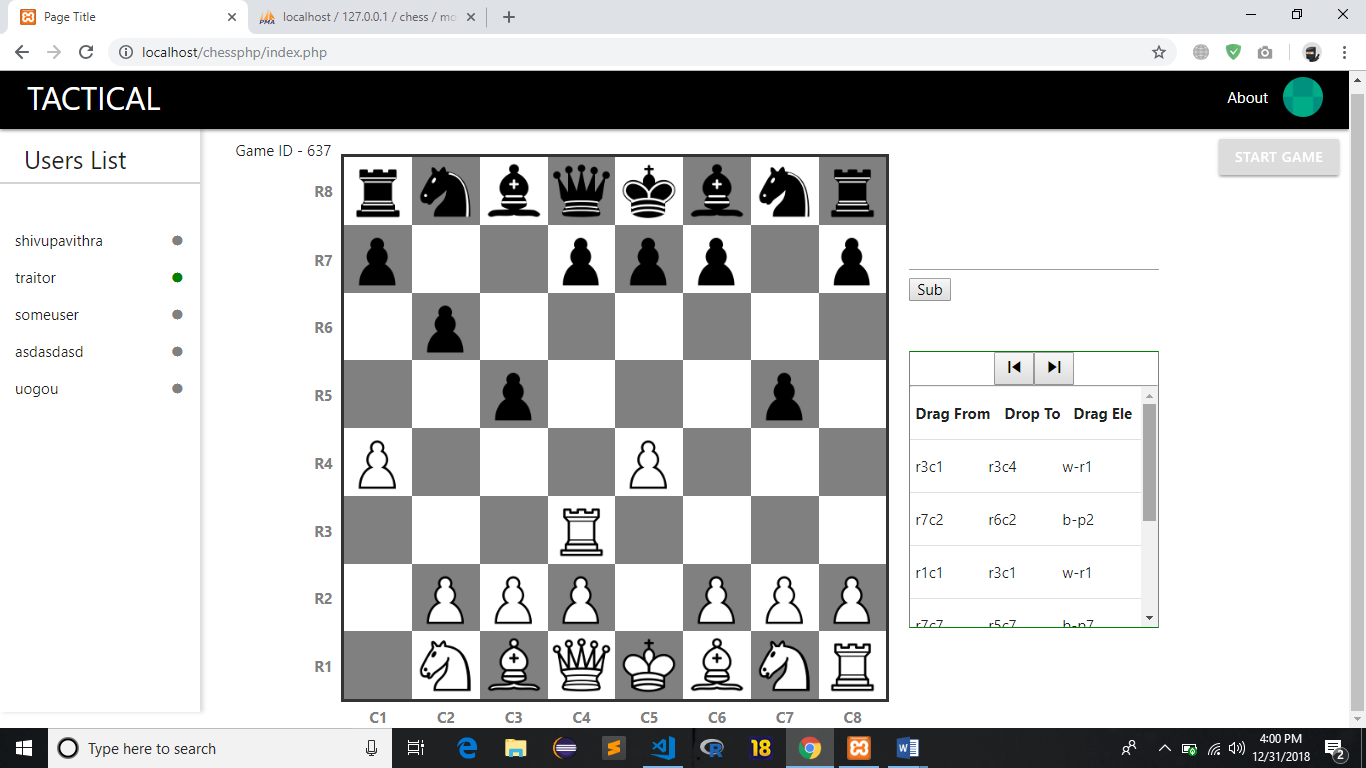
**SCREENSHOTS**

**Login Page**

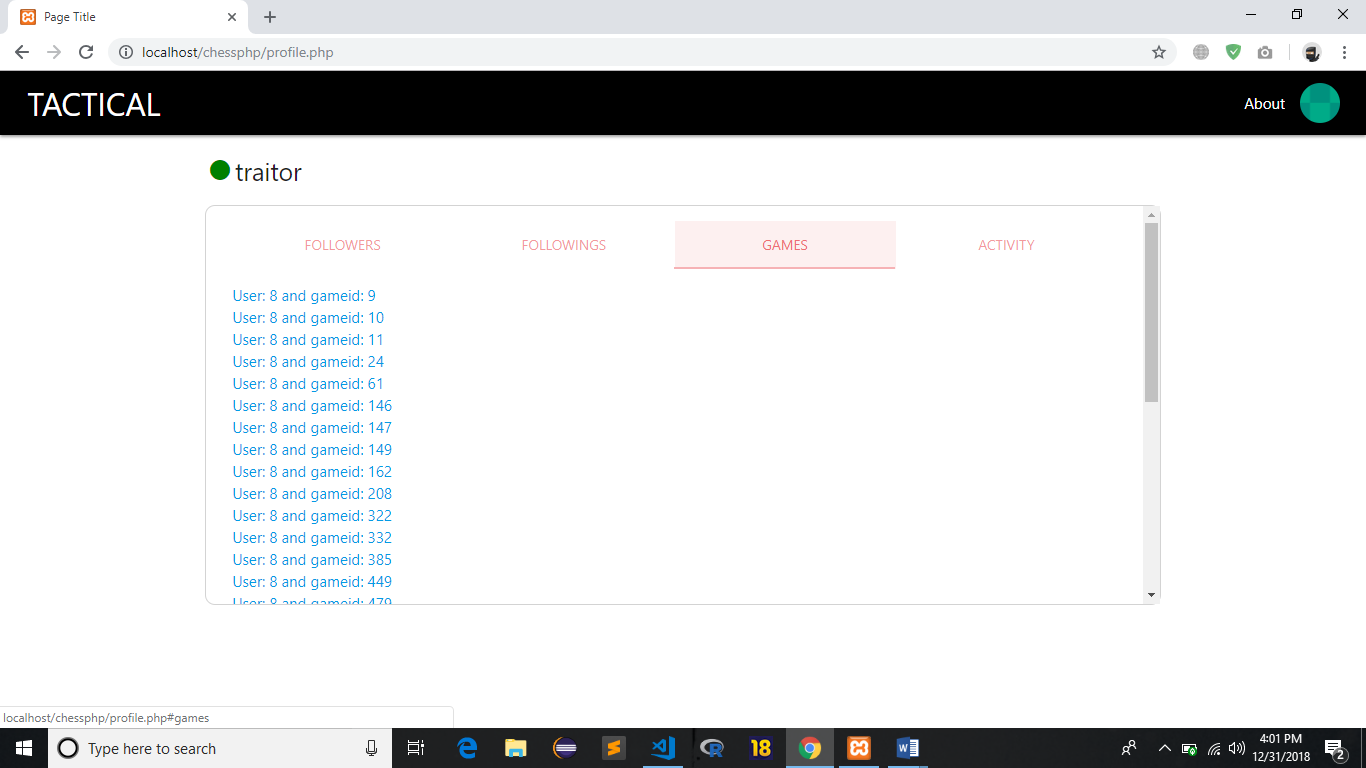
****

**Index Page**

**Displaying Move**

****

**Profile Page**



# **CONCLUSION & FUTURE ENHANCEMENT**

In the normal chess games be it manual, offline or online it isn’t always possible to view the previous games to make the present game better. Thus, in Tactical Chess it would become possible to review and rate the previous games. It would then be possible to decide what the best move is and hence play it to win the game. Also, the best move can be generated, to facilitate the best possible move in the current game to receive the best possible output. Customisation of the game based on the current standings generate an improved look of the program Chess is an attempt to review, analyse and re-evaluate games.

Every time a game is played, only the scores are collected and stored, if ever the player wants to review his game, based on the FEN-String notation made, then that would become a greed. It’s in this current scenario that it would be necessary to deal with games whose moves are all stored. The existing system, is inefficient to track the moves made by the player and it also becomes difficult to customize the game and to find out the best possible move that the user has made.

Thus, the Tactical Chess game is most efficient in finding the best game and best possible move made by the player. Thus, Tactical Chess unlike the other chess games is an efficient methodology to make effective the game of chess.

As a future enhancement, it can made possible to find out the best move made by a player. The best player can also be analyzed using the reduced time taken by the player to make the best possible move. This is seen as the next level of the project that has enhanced functionalities with artificial intelligence wherein the game can be played by players from different locations from around the globe.

**REFERENCES**

[1],Code Miles, <https://www.codemiles.com/finished-projects/java-chess->

[2]Yemista,Java Forums,10 November,2011

[3][www.github.com](http://www.github.com)

[4][www.materialized.com](http://www.materialized.com)

[5] [www.w3schools.com](http://www.w3schools.com)