Project

< Connect4>

CIS-17B

Name: Aleksandar Videv

Table Of Contents

1.Introduction………………………………….……..……………..……….……2

2. Overview………………………………….………………………….….…..….2

2.1 Project Summary…………………….………………..………………4

2.2 Versioning Scheme…………………….……………………………..3

3. Project Description…………………….…………………………...……….…4

4. Gantt Chart…………………….………………………………………........…5

5. Pseudo-code…………………….…………………………………….........…6

6. UML Diagrams…………….……………………….…………………………..9

7.Demonstrating Functionality………………..………….….……….….…….10

7.1 Proof of Concept.…..................………….….….…..…..…......…..10

7.2 Testing Results……………….………….….….….….….….……...38

**1.Introduction**

The **Connect 4** application is a web-based game that brings the classic two-player connection game to an online platform. Users can enjoy both Player vs Player and Player vs Computer modes. Developed using PHP, HTML, CSS, JavaScript, and MySQL, the application includes user authentication, game statistics tracking, and an administrative interface for user management.

**2. Overview**

**2.1 Project Summary**

* **Total Lines of Code**: Approximately 2000
* **Technologies Used**:
  + **PHP**:
    - Variables: ~50
    - Methods/Functions: ~30
  + **JavaScript**:
    - Variables: ~40
    - Functions: ~25
  + **HTML/CSS**: For structuring and styling the web pages
* **Database**:
  + **MySQL**: User authentication and statistics tracking
  + **Tables**: entity\_user

**Project Insights**: The application focuses on delivering an engaging user experience with seamless gameplay and intuitive navigation. Around 80 hours were dedicated to development and testing, ensuring both functionality and user-friendliness.

**2.2** **Versioning Scheme**

**Version 1.0**: Initial release with basic game functionality (Player vs Computer).

**Version 1.1**: Added Player vs Player mode.

**Version 1.2**: Implemented user registration and login.

**Version 1.3**: Introduced game statistics tracking.

**Version 1.4**: Developed admin interface for user management.

**Version 1.5**: Enhanced UI/UX with updated graphics.

**Version 1.6**: Improved security measures.

**Version 1.7**: Optimized database interactions.

**Version 1.8**: Added error handling and feedback.

**Version 1.9**: Implemented sorting and filtering in admin panel.

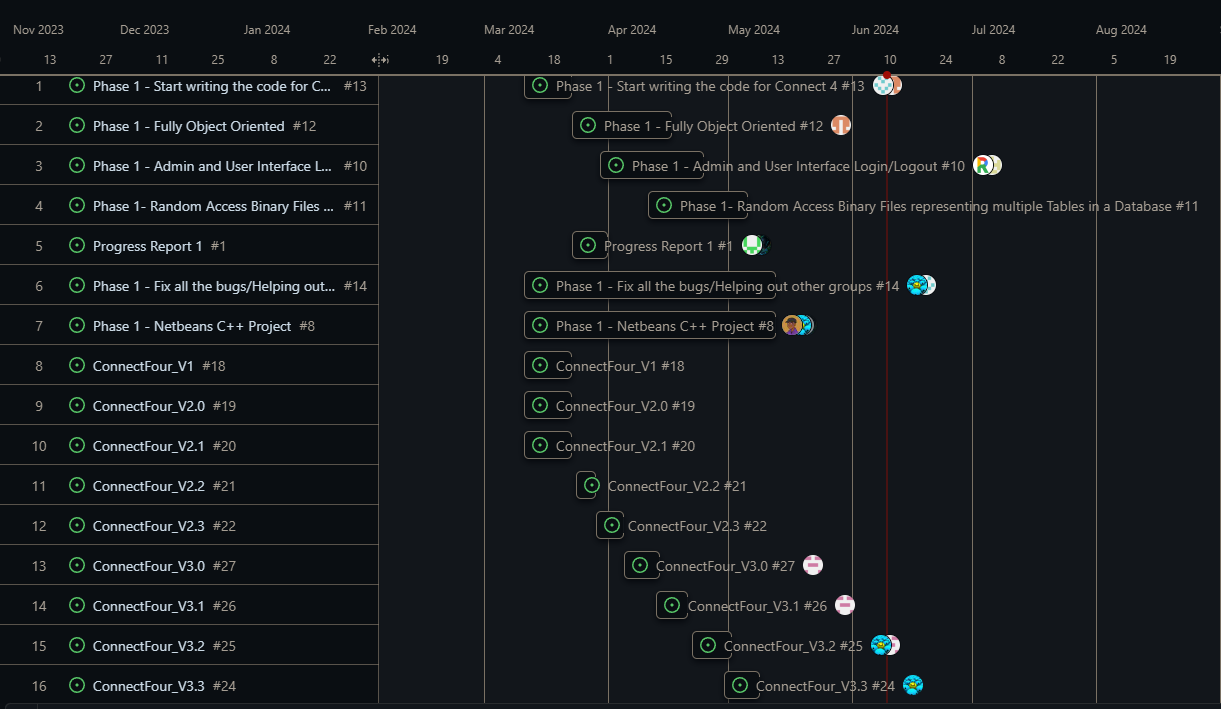
**Version 2.0**: Finalized application with documentation.

**3. Project Description**

The Connect 4 web application offers key features to provide an enjoyable gaming experience:

* **Gameplay Modes**:
  + **Player vs Player**: Two users can play against each other.
  + **Player vs Computer**: Solo play against an AI opponent.
* **User Authentication**:
  + Secure registration and login system.
  + Password management and session handling.
* **Statistics Tracking**:
  + Records wins, losses, and ties.
  + Displays user stats on profile pages.
* **Administrative Features**:
  + Admins can view, edit, and delete user accounts.
  + User list with sorting and filtering options.
* **User Interface**:
  + Responsive design suitable for various devices.
  + Interactive game board with real-time updates.
* **Security Measures**:
  + Input validation to prevent malicious activities.
  + Proper session management to protect user data.

The application emphasizes modularity and scalability, allowing for future enhancements. It provides a solid foundation for users to enjoy the classic game of Connect 4 in an online environment while maintaining secure and efficient operations.

**4. Gantt Chart** 

# 

**4. Pseudo-code**

// Main flow of the application

START

// User visits the index.php

IF user is not logged in THEN

REDIRECT to login page

ELSE

DISPLAY game options

// User selects to play a game

IF playgame.php is requested THEN

LOAD Connect4.js

INITIALIZE game board

WHILE game is not over

WAIT for user move

VALIDATE move

UPDATE game board

CHECK for win condition

END WHILE

IF game is over THEN

DISPLAY GameOver.html

END IF

END IF

// User can register

IF registerform.php is requested THEN

DISPLAY registration form

IF form is submitted THEN

VALIDATE input with regular expressions

SAVE user data to database

SET user session

REDIRECT to game options

END IF

END IF

// User can log in

IF loginhandler.php is requested THEN

VALIDATE login credentials

IF valid THEN

SET user session

REDIRECT to game options

ELSE

DISPLAY error

END IF

END IF

// User can edit their profile

IF edituser.php is requested THEN

DISPLAY user data

IF form is submitted THEN

VALIDATE input with regular expressions

UPDATE user data in database

REDIRECT to game options

END IF

END IF

// Admin can manage users

IF allusers.php or deluser.php is requested THEN

IF user is admin THEN

DISPLAY user management options

HANDLE user addition, deletion, and updates

ELSE

REDIRECT to game options

END IF

END IF

// Save game statistics

IF savestats.php is requested THEN

READ game data

SAVE game statistics to database

END IF

// Display statistics

IF displaystats.php is requested THEN

QUERY database for statistics

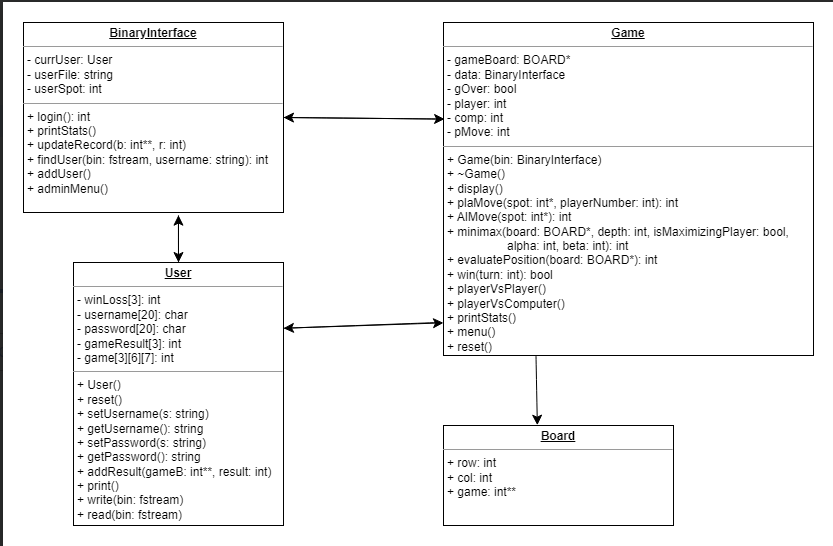
DISPLAY statistics

END IF

// Connect to the database

INCLUDE dbconnect.php

END

**5. UML**

**6.Demonstrating Functionality**

**6.1 Proof of Concept**

**Main.css:**

|  |
| --- |
| /\* main.css \*/  /\* Additional styles for form input alignment \*/  form {  **width**: **300px**;  **margin**: **0** **auto**;  **text-align**: **left**;  }  form label {  **display**: **inline**-**block**;  **width**: **100px**;  **text-align**: **right**;  }  form input[type="text"],  form input[type="password"],  form input[type="number"],  form input[type="submit"],  form button[type="button"] {  **width**: **150px**;  **margin-left**: **10px**;  }  form input[type="radio"] {  **margin-left**: **10px**;  }  form div {  **margin-bottom**: **10px**;  }  /\* Center the Admin radio buttons \*/  form div**.radio-group** {  **text-align**: **center**;  }  form div**.radio-group** label {  **text-align**: **left**;  }  form div**.submit-group** {  **text-align**: **center**;  }  form div**.submit-group** input,  form div**.submit-group** button {  **width**: **150px**;  **display**: **inline**-**block**;  **margin**: **0** **5px**;  }  **#login** {  **text-align**: **center**;  }  **#response** {  **text-align**: **center**;  }  /\* Existing styles \*/  body {  **background-color**: white;  }  div**#board** {  **position**: **relative**;  **width**: **750px**;  **height**: **600px**;  **margin**: **auto**;  **background**: blue **no-repeat**;  }  div**#messageArea** {  **width**: **750px**;  **margin**: **20px** **auto**;  **text-align**: **center**;  **color**: black;  **font-size**: **18px**;  **font-weight**: **bold**;  }  table**#gameBoard** {  **position**: **relative**;  **border-spacing**: **0px**;  **margin**: **auto**;  }  table**#gameBoard** td {  **width**: **94px**;  **height**: **94px**;  **background**: url(circle.png) **no-repeat**;  **background**-**size**: **94px**, **94px**;  }  form**#placeInput** {  **width**: **750px**;  **margin**: **20px** **auto**;  **text-align**: **center**;  }  form**#placeInput** input[type="text"] {  **background-color**: white;  **border-color**: black;  **width**: **100px**;  **font-size**: **1em**;  **padding**: **5px**;  **border**-radius: **5px**;  **margin-right**: **10px**;  }  form**#placeInput** input[type="button"] {  **background-color**: **#2f76fa**;  **border**: **none**;  **color**: white;  **padding**: **10px** **20px**;  **text-align**: **center**;  **text-decoration**: **none**;  **display**: **inline**-**block**;  **font-size**: **16px**;  **margin**: **4px** **2px**;  **cursor**: **pointer**;  **border**-radius: **12px**;  transition: **background-color** **0.3s** ease, transform **0.3s** ease;  }  form**#placeInput** input[type="button"]**:hover** {  **background-color**: **#1938ff**;  transform: scale(**1**.**05**);  }  table**#gameBoard** **.red** {  **background**: url("red.png") **no-repeat** **center** **center**;  **background**-**size**: **94px**, **94px**;  }  table**#gameBoard** **.yellow** {  **background**: url("yellow.png") **no-repeat** **center** **center**;  **background**-**size**: **83px**, **83px**;  **color**: **#FFD700**; /\* Darker yellow for better visibility \*/  }  header {  **color**: white;  **background**: linear-gradient(**#2f76fa**, **#1938ff**);  **font-size**: **18px**;  **padding**: **0.5em** **1em**;  **margin-bottom**: **2em**;  **border**-radius: **0.2em**;  **display**: flex;  **justify**-**content**: space-between;  align-items: **center**;  **text-align**: **center**;  **height**: **25px**; /\* Adjust the height to your preference \*/  }  **.header-left** {  flex: **1**;  **text-align**: **left**;  }  **.header-center** {  flex: **1**;  **text-align**: **center**;  }  **.header-right** {  flex: **1**;  **text-align**: **right**;  }  header a {  **text-decoration**: **none**;  **color**: white;  **margin**: **0** **10px**;  }  header a**:hover** {  **background-color**: indianred;  }  footer {  **text-align**: **center**;  **font-size**: **10px**;  **font-style**: **italic**;  **color**: grey;  }  **#reportTable** {  **width**: **100**%;  }  **#tableButton**, **#reportTable** th {  **background**: linear-gradient(**#2f76fa**, **#1938ff**);  **color**: white;  **font-size**: **16px**;  **font-weight**: **bold**;  }  **#tableButton** {  **padding**: **2px** **50px**;  **border**-radius: **0.2em**;  **border-color**: **#1938ff**;  }  **#reportTable** table, **#reportTable** th, **#reportTable** td {  **border**: **1px** **solid** black;  **border**-radius: **0.2em**;  } |

**dbconnect.php:**

|  |
| --- |
| <?php  $dsn = "mysql:host=localhost;dbname=connect\_4";  $dbusername = "root";  $dbpassword = "";  **try**{  $pdo = **new** PDO($dsn, $dbusername, $dbpassword);  $pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);  }  **catch**(PDOException $e){  **echo** "Connection Failed" . $e->getMessage();  } |

**displayfunctions.js:**

|  |
| --- |
| **function** sortTable(col, order) {  **var** index, table, rows, switching, i, x, y, shouldSwitch;  **if** (col == "time") {  index = **2**;  } **else**  index = col;  table = document.getElementById("reportTable");  switching = **true**;  **while** (switching) {  switching = **false**;  //Get rows with data and loop through all of them  rows = table.rows;  **for** (i = **1**; i < (rows.length - **1**); i++) {  shouldSwitch = **false**;  //Get two adjacent rows, then compare  x = rows[i].getElementsByTagName("TD")[index];  y = rows[i + **1**].getElementsByTagName("TD")[index];  **if** (order == **0**) {  **if** (x.innerHTML.toLowerCase() > y.innerHTML.toLowerCase()) {  // If swap needed, mark as a switch and break the loop to switch and restart  shouldSwitch = **true**;  **break**;  }  } **else** **if** (x.innerHTML.toLowerCase() < y.innerHTML.toLowerCase()) {  // If swap needed, mark as a switch and break the loop to switch and restart  shouldSwitch = **true**;  **break**;  }  }  **if** (shouldSwitch) {  //Make switch and mark sentinel value as true to restart the sorting loop  rows[i].parentNode.insertBefore(rows[i + **1**], rows[i]);  switching = **true**;  }  }  }  **function** filter(){  **var** input,filter,tab,tr,td,rowTxt;    //Get input in filter box  input = document.getElementById("input");  filter = input.value.toUpperCase();  //Get table elements  tab = document.getElementById("reportTable");  tr = tab.getElementsByTagName("tr");  **for** (**var** i = **0**; i < tr.length; i++){    //Get each row's text  td = tr[i].getElementsByTagName("td")[**0**];  **if** (td){    rowTxt = td.textContent || td.innerText;  //Then see if filter values exist in text value  **if** (rowTxt.toUpperCase().indexOf(filter) > -**1**){  tr[i].style.display = "";    }**else** tr[i].style.display = "none"; //If not, don't display  }  }  }  **function** showEditForm(user){  section = document.getElementById('login');  section.innerHTML = '<input type="hidden" name="u" value="' + user.username +'">';  section.innerHTML += '<input type="hidden" name="p" value="' + user.password +'">';    section.innerHTML += '<label for="username">Username:</label>';  section.innerHTML += '<input type="text" name="username" value="' + user.username +'" required>';  section.innerHTML += '<br>';  section.innerHTML += '<label for="password">Password:</label>';  section.innerHTML += '<input type="text" name="password" value="' + user.password +'" required>';  section.innerHTML += '<br>';  section.innerHTML += '<label for="fname">First Name:</label>';  section.innerHTML += '<input type="text" name="fname" value="' + user.fname +'" required>';  section.innerHTML += '<br>';  section.innerHTML += '<label for="lname">Last Name:</label>';  section.innerHTML += '<input type="text" name="lname" value="' + user.lname +'" required>';  section.innerHTML += '<br>';  section.innerHTML += '<label for="wins">Wins:</label>';  section.innerHTML += '<input type="number" name="wins" value="' + user.wins +'" required>';  section.innerHTML += '<br>';  section.innerHTML += '<label for="losses">Losses:</label>';  section.innerHTML += '<input type="number" name="losses" value="' + user.losses +'" required>';  section.innerHTML += '<br>';  section.innerHTML += '<label for="ties">Ties:</label>';  section.innerHTML += '<input type="number" name="ties" value="' + user.ties +'" required>';  section.innerHTML += '<br>';  section.innerHTML += '<label for="admin">Admin:</label>';  **if**(user.admin == **1**){  section.innerHTML += 'Yes<input type="radio" name="admin" value="1" checked>';  section.innerHTML += 'No<input type="radio" name="admin" value="0">';  }**else** {  section.innerHTML += 'Yes<input type="radio" name="admin" value="1" >';  section.innerHTML += 'No<input type="radio" name="admin" value="0" checked>';  }  section.innerHTML += '<br><input type="submit" value="Save">';  section.innerHTML += '<button type="reset">Reset to current values</button> <br><br>';  } |

**footer.html:**

|  |
| --- |
| <footer>  <br>  <p>Made for CIS-17B: C++ Advanced Objects, Spring 24 by "Online Connect 4 Group"</p>  </footer>  </body>  </html> |

**gameboard.html:**

|  |
| --- |
| <div id="board">  <table id='gameBoard'>  <tr>  <td id="00"></td><td id="01"></td><td id="02"></td><td id="03"></td><td id="04"></td><td id="05"></td><td id="06"></td>  </tr>  <tr>  <td id="10"></td><td id="11"></td><td id="12"></td><td id="13"></td><td id="14"></td><td id="15"></td><td id="16"></td>  </tr>  <tr>  <td id="20"></td><td id="21"></td><td id="22"></td><td id="23"></td><td id="24"></td><td id="25"></td><td id="26"></td>  </tr>  <tr>  <td id="30"></td><td id="31"></td><td id="32"></td><td id="33"></td><td id="34"></td><td id="35"></td><td id="36"></td>  </tr>  <tr>  <td id="40"></td><td id="41"></td><td id="42"></td><td id="43"></td><td id="44"></td><td id="45"></td><td id="46"></td>  </tr>  <tr>  <td id="50"></td><td id="51"></td><td id="52"></td><td id="53"></td><td id="54"></td><td id="55"></td><td id="56"></td>  </tr>  </table>  </div>  <div id="messageArea"></div>  <form id="placeInput">  <input type="text" id="userMove" placeholder="1" maxlength="1">  <input type="button" id="getMove" value="Place">  </form>  <script src="Connect4.js"></script>  <form action='./savestats.php' method='post'>  <div id="response"></div>  </form> |

**header.php:**

|  |
| --- |
| <?php  session\_start();  ?>  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <title>Connect 4</title>  <link rel="stylesheet" href="./includes/main.css">  </head>  <body>  <header>  <div class="header-left">  <p>Online Connect 4</p>  </div>  <div class="header-center">  <?php  **if** (isset($\_SESSION["loggedIn"]) && $\_SESSION["loggedIn"] == **true**) {  **echo** "<a href='./playgame.php'>Play a Game</a>";  **echo** "<a href='./displaystats.php'>View your stats</a>";  **if** (isset($currUser) && $currUser->getAdmin()) {  **echo** "<a href='./allusers.php'>View all users</a>";  }  }  ?>  </div>  <div class="header-right">  <?php  **if** (isset($\_SESSION["loggedIn"]) && $\_SESSION["loggedIn"] == **true**) {  **echo** '<a href="./index.php">Logout</a>';  }  ?>  </div>  </header> |

**user.php:**

|  |
| --- |
| <?php  **class** **User** {  **private** $id;  **private** $username;  **private** $password;  **private** $fname;  **private** $lname;  **private** $wins;  **private** $losses;  **private** $ties;  **private** $admin;  **public** **function** **getID**() {  **return** $this->id;  }  **public** **function** **setID**($id){  $this->id = $id;  }    **public** **function** **getUser**(){  **return** $this->username;  }  **public** **function** **setUser**($u){  $this->username = $u;  }    **public** **function** **getPass**(){  **return** $this->password;  }  **public** **function** **setPass**($p){  $this->password = $p;  }    **public** **function** **getfName**(){  **return** $this->fname;  }  **public** **function** **setfName**($fn){  $this->fname = $fn;  }    **public** **function** **getlName**(){  **return** $this->lname;  }  **public** **function** **setlName**($ln){  $this->lname = $ln;  }    **public** **function** **getWins**(){  **return** $this->wins;  }  **public** **function** **setWins**($w){  $this->wins = $w;  }    **public** **function** **getLosses**(){  **return** $this->losses;  }  **public** **function** **setLosses**($l){  $this->losses = $l;  }    **public** **function** **getTies**(){  **return** $this->ties;  }  **public** **function** **setTies**($t){  $this->ties = $t;  }    **public** **function** **getAdmin**(){  **return** $this->admin;  }  **public** **function** **setAdmin**($a){  $this->admin = $a;  }    **public** **function** **toJSON**(){  **return** '{"username":"'.$this->username.'","password":"'.$this->password.'","fname":"'.$this->fname.  '","lname":"'.$this->lname.'","wins":"'.$this->wins.'","losses":"'.$this->losses.  '","ties":"'.$this->ties.'","admin":"'.$this->admin.'"}';  }    **public** **function** **printStats**(){  **echo** $this->fname . " " . $this->lname . "'s Stats <br>";  **echo** "Wins-Losses-Ties: ";  **echo** $this->wins ."-" . $this->losses . "-" . $this->ties;  }  **public** **function** **fillFromSql**($u, $p) {  **require** ('./dbconnect.php');  $query = "SELECT \* FROM `entity\_user` WHERE `username` = :u AND `password` = :p";  $statement = $pdo->prepare($query);  $statement->bindParam(":u", $u);  $statement->bindParam(":p", $p);  $statement->execute();  //Load data from query into array  $results = $statement->fetchAll(PDO::FETCH\_ASSOC);  //If there is no user with that username and pass, return -1  **if** (count($results) == **0**) {  **return** -**1**;  }  $this->id = $results[**0**]["id\_user"];  $this->username = $results[**0**]["username"];  $this->password = $results[**0**]["password"];  $this->fname = $results[**0**]["first\_name"];  $this->lname = $results[**0**]["last\_name"];  $this->wins = $results[**0**]["wins"];  $this->losses = $results[**0**]["losses"];  $this->ties = $results[**0**]["ties"];  $this->admin = $results[**0**]["admin"];  }  **public** **function** **addToSql**() {  //Connect to database  **require** ('./dbconnect.php');  //First see if username is taken  $query = "SELECT \* FROM `entity\_user` WHERE `username` = :u";  $statement = $pdo->prepare($query);  $statement->bindParam(":u", $this->username);    $statement->execute();  //Load data from query into array  $results = $statement->fetchAll(PDO::FETCH\_ASSOC);    //If there are any users with that username, return -1  **if** (count($results) != **0**) {  **return** -**1**;  }    $query = "INSERT INTO `entity\_user`(`username`, `password`, `first\_name`, `last\_name`, `wins`, `losses`, `ties`, `admin`) VALUES (:u,:p,:fn,:ln,'0','0','0',:a)";  $statement = $pdo->prepare($query);  $statement->bindParam(":u", $this->username);  $statement->bindParam(":p", $this->password);  $statement->bindParam(":fn", $this->fname);  $statement->bindParam(":ln", $this->lname);  $statement->bindParam(":a", $this->admin);  $statement->execute();  }    **public** **function** **updateSql**(){  //Connect to database  **require** ('./dbconnect.php');    $query = "UPDATE `entity\_user` SET `username`= :u,`password`= :p,`first\_name`= :fn,`last\_name`= :ln,`wins`= :w,`losses`= :l,`ties`= :t,`admin`= :a WHERE `id\_user` = :id";    $statement = $pdo->prepare($query);  $statement->bindParam(":u", $this->username);  $statement->bindParam(":p", $this->password);  $statement->bindParam(":fn", $this->fname);  $statement->bindParam(":ln", $this->lname);  $statement->bindParam(":w", $this->wins);  $statement->bindParam(":l", $this->losses);  $statement->bindParam(":t", $this->ties);  $statement->bindParam(":a", $this->admin);  $statement->bindParam(":id", $this->id);  $statement->execute();  }    **public** **function** **delFromSql**(){  //Connect to database  **require** ('./dbconnect.php');    $query = "DELETE FROM `entity\_user` WHERE `id\_user` = :id";    $statement = $pdo->prepare($query);  $statement->bindParam(":id", $this->id);    $statement->execute();  }  } |

**Connect4.js:**

|  |
| --- |
| **let** spots;  **let** gOver;  **let** player;  **let** comp;  **let** pMove;  **let** board;  **let** row;  **let** col;  **let** turn;  **var** view = {  displayMessage: **function**(message, colorPart){  **var** color = (colorPart === 'yellow') ? '#FFD700' : colorPart;  **var** messageArea = document.getElementById("messageArea");  messageArea.innerHTML = message + ` (<span style="color: ${color};">${colorPart}</span>)`;  },  displayPlayer: **function**(move, color){  **var** cell = document.getElementById(move);  cell.setAttribute("class", color);  }  };  **function** plaMove(playerNumber, user) {  **let** color;  **let** move;    **if**(playerNumber === **1**)  color = "red";  **else**  color = "yellow";    pMove = user - **1**; // Adjusting for 1-based index  **for** (**let** i = **5**; i >= **0**; i--) {  **if** (board[i][pMove] === **0**) {  move = i +""+ pMove;  board[i][pMove] = playerNumber;  view.displayPlayer(move, color);  **break**;  }  }  spots++;  **return** playerNumber;  }  **function** AIMove() {  **let** cMove = Math.floor(Math.random() \* **3**);  **let** block = Math.floor(Math.random() \* **2**);  **let** success = **false**;  **let** color = "yellow";  **let** move;    **if** (block === **0**) {  **if** (cMove === **0**) {  **for** (**let** i = **5**; i >= **0**; i--) {  **if** (board[i][pMove] === **0**) {  board[i][pMove] = comp;  move = i + "" + pMove;  view.displayPlayer(move, color);  success = **true**;  **break**;  }  }  }  **if** (cMove === **1** || success === **false**) {  **for** (**let** i = **5**; i >= **0**; i--) {  **for** (**let** j = **1**; j <= **6** - pMove; j++) {  **if** (board[i][pMove + j] === **0**) {  board[i][pMove + j] = comp;  pMove = pMove + j;  move = i + "" + pMove;  view.displayPlayer(move, color);  success = **true**;  **break**;  }  }  **if** (success) **break**;  }  }  **if** (cMove === **2** || success === **false**) {  console.log("block3");  **for** (**let** i = **5**; i >= **0**; i--) {  **for** (**let** j = **6**; j >= pMove - j; j--) {  **if** (board[i][pMove - j] === **0**) {  board[i][pMove - j] = comp;  pMove = pMove - j;  move = i + "" + pMove;  view.displayPlayer(move, color);  success = **true**;  **break**;  }  }  **if** (success) **break**;  }  }  }  **if** (block === **1** || success === **false**) {  **do** {  cMove = Math.floor(Math.random() \* **7**);  **for** (**let** i = **5**; i >= **0**; i--) {  **if** (board[i][cMove] === **0**) {  board[i][cMove] = comp;  move = i + "" + cMove;  view.displayPlayer(move, color);  success = **true**;  **break**;  }  }  } **while** (!success);  }    view.displayMessage("Player's Turn", "red");  spots++;  }  **function** win(turn)  {  /\*  \* win conditions  \* 1. if 0, then reset whole thing  \* 2. if 0 and player = 0, then mark first spot  \* 3. if spot = player, then tally win condtion  \* 4. else not, then switch over to other player filling spot  \*/  //declare and initialize variables  **let** count;    //check row wins  **for** (**let** i = **0**; i < row; i++)  {  count = **0**;  **for** (**let** j = **0**; j < col; j++)  {  **if** (board[i][j] === turn)  {  count++;  **if** (count === **4**)  **return** **true**;  }  **else**  {  count = **0**; // Reset count if not consecutive  }  }  }    //if no win reset count  count = **0**;    //check column wins  **for** (**let** j = **0**; j < col; j++)  {  **let** count = **0**;  **for** (**let** i = **0**; i < row; i++)  {  **if** (board[i][j] === turn)  {  count++;  **if** (count === **4**)  **return** **true**;  }  **else**  {  count = **0**; // Reset count if not consecutive  }  }  }    //if no win reset count  count = **0**;    //for checking diagonal win conditions  //for checking diagonal win conditions  **for** (**let** i = **0**; i < row - **3**; i++)  {  **for** (**let** j = **0**; j < col - **3**; j++)  {  //diagonals from top-left to bottom-right  **if** (board[i][j] === turn &&  board[i + **1**][j + **1**] === turn &&  board[i + **2**][j + **2**] === turn &&  board[i + **3**][j + **3**] === turn)  {  **return** **true**;  }  }  }  **for**(**let** i = **0**; i < row - **3**; i++)  {  **for** (**let** j = **3**; j < col; j++)  {  // diagonals from top-right to bottom-left  **if** (board[i][j] === turn &&  board[i + **1**][j - **1**] === turn &&  board[i + **2**][j - **2**] === turn &&  board[i + **3**][j - **3**] === turn)  {  **return** **true**;  }  }  }    //if no wins at all  **return** **false**;  }  **function** playerVsPlayer(user) {  **let** gOver;  **let** moves = **42**;    // Player 1's move  plaMove(turn, user);  gOver = win(turn);    **if** (spots === moves && !gOver) {  alert("The game ended in a tie");  reset(**3**);  }  **if** (gOver && turn === **1**) {  alert("Player 1 is the winner");  reset(**1**);  } **else** **if** (gOver && turn === **2**) {  alert("Player 2 is the winner");  reset(**2**);  }    **if**(turn === **2**)  turn = **1**;  **else** **if**(turn === **1**)  turn = **2**;    **let** color = (turn === **1**) ? "red" : "yellow";  view.displayMessage("Player " + turn + ", please put a legal column (1 to 7, left to right)", color);  }  **function** playerVsComputer(user){  **let** player = **1**; // Player 1 starts the game  **let** turn;  **let** gOver;  **let** moves = **42**;    // Player's move  turn = plaMove(player, user);  gOver = win(turn);  **if** (spots >= moves && !gOver) {  alert("The game ended in a tie");  reset(**3**);  }  **if** (gOver && turn === **1**) {  alert("You are the winner");  reset(**1**);  } **else** **if**(gOver && turn === **2**){  alert("The computer won");  reset(**2**);  }    // Computer's move  turn = AIMove();  gOver = win(turn);  **if** (spots >= moves && !gOver) {  reset(**3**);  alert("The game ended in a tie");  }  **if** (gOver && turn === **1**) {  reset(**1**);  alert("You are the winner");    } **else** **if**(gOver && turn === **2**){  reset(**2**);  alert("The computer won");  }    }  **function** game() {  row = **6**;  col = **7**;  board = [];    **for** (**let** i = **0**; i < row; i++) {  board.push([**0**]);  **for** (**let** j = **0**; j < col; j++) {  board[i][j] = **0**;  }  }  turn = **1**;  comp = **2**;  spots = **0**;    **do** {  uInput = prompt("Would you like to play \n1. Player vs Player \n2. Player vs Computer");  uInput = parseInt(uInput);  **if** (uInput === **1** || uInput === **2**) valid = **true**;  **else** alert("Invalid selection, please choose option 1 or 2");  } **while** (!valid);    view.displayMessage("Player 1, please put a legal column (1 to 7, left to right)", "red");  }  **function** reset(result){  **var** prompt = document.getElementById("response");  prompt.innerHTML += "<input type='hidden' name='result' value='" + parseInt(result) + "'>\n";  prompt.innerHTML += "<label for='replay'>Would you like to play again?</label><br>\n";  prompt.innerHTML += "Yes<input type='radio' name='replay' value='1'>\n";  prompt.innerHTML += "No<input type='radio' name='replay' value='0'> <br>\n";  prompt.innerHTML += "<input type='submit' value='Submit'>\n";  }  **function** handleMove() {  **var** userMove = document.getElementById("userMove");  user = userMove.value;  **if**(uInput === **1**)  playerVsPlayer(user);  **else**  playerVsComputer(user);  }  **function** handleKeyPress(e) {  **var** getMove = document.getElementById("getMove");  **if** (e.keyCode === **13**) {  getMove.click();  **return** **false**;  }  }    **function** init() {  game();  **var** getMove = document.getElementById("getMove");  getMove.onclick = handleMove;  **var** userMove = document.getElementById("userMove");  userMove.onkeypress = handleKeyPress;  }  window.onload = init; |

**allusers.php:**

|  |
| --- |
| <?php  // allusers.php  **include**("./includes/user.php");  session\_start();  $\_SESSION["loginErr"] = "";  $currUser = **new** User();  $result = $currUser->fillFromSql($\_SESSION["user"], $\_SESSION["pass"]);  **if** ($result == -**1**) {  // If not user found, set login error message  $\_SESSION["loginErr"] = "No user with that username and password";  // Send back to main page  header("Location: ./index.php");  } **elseif** (!$currUser->getAdmin()) {  header("Location: ./displaystats.php");  }  ?>  <?php  **include**("./includes/header.php");  ?>  <script src="includes/displayfunctions.js"></script>  <form action='./edituser.php' method='post' onsubmit="return validateEditForm()">  <div id="login">  </div>  </form>  <div id='response'>  <?php  // Fill array with all users  $userAry = [];  // Connect to database  **require** ('./dbconnect.php');  $query = "SELECT `username`, `password` FROM `entity\_user`";  $statement = $pdo->prepare($query);  $statement->execute();  $results = $statement->fetchAll(PDO::FETCH\_ASSOC);  **for** ($i = **0**; $i < count($results); $i++) {  $userAry[$i] = **new** User();  $userAry[$i]->fillFromSql($results[$i]["username"], $results[$i]["password"]);  }  // Once all users are gotten, place into table  **if** (count($userAry) > **0**) {  **echo** "<input type='text' id='input' onkeyup='filter()' placeholder='Type to filter by username'>";  **echo** "<table id='reportTable'>";  **echo** "<tr>";  **echo** "<th><button id='tableButton' value='0' onclick='sortTable(this.value,0)' >Username</button></th>";  **echo** "<th><button id='tableButton' value='1' onclick='sortTable(this.value,0)' >Password</button></th>";  **echo** "<th><button id='tableButton' value='2' onclick='sortTable(this.value,0)' >Real Name</button></th>";  **echo** "<th><button id='tableButton' value='3' onclick='sortTable(this.value,1)' >Wins</button></th>";  **echo** "<th><button id='tableButton' value='4' onclick='sortTable(this.value,1)' >Losses</button></th>";  **echo** "<th><button id='tableButton' value='5' onclick='sortTable(this.value,1)' >Ties</button></th>";  **echo** "<th><button id='tableButton' value='6' onclick='sortTable(this.value,1)' >Admin</button></th>";  **echo** "<th>Edit</th>";  **echo** "<th>Delete</th>";  **echo** "</tr>";  **for** ($i = **0**; $i < count($userAry); $i++) {  **echo** "<tr>";  **echo** "<td>" . $userAry[$i]->getUser() . "</td>";  **echo** "<td>" . $userAry[$i]->getPass() . "</td>";  **echo** "<td>" . $userAry[$i]->getfName() . " " . $userAry[$i]->getlName() . "</td>";  **echo** "<td>" . $userAry[$i]->getWins() . "</td>";  **echo** "<td>" . $userAry[$i]->getLosses() . "</td>";  **echo** "<td>" . $userAry[$i]->getTies() . "</td>";  $a = ($userAry[$i]->getAdmin()) ? "Yes" : "No";  **echo** "<td>" . $a . "</td>";  // Edit Button  **echo** "<td>";  **echo** "<input type='button' value='Edit' onclick='showEditForm(" . $userAry[$i]->toJSON() . ")'>";  **echo** "</td>";  // Delete Button  **echo** "<td>";  **echo** "<form action='deluser.php' method='post'>";  **echo** "<input type='hidden' id='u' name='u' value='" . $userAry[$i]->getUser() . "'>";  **echo** "<input type='hidden' id='p' name='p' value='" . $userAry[$i]->getPass() . "'>";  **echo** "<input type='submit' value='Delete'>";  **echo** "</form></td>";  **echo** "</tr>";  }  **echo** "</table>";  }  ?>  </div>  <?php  **include**("./includes/footer.html");  ?> |

**deluser.php:**

|  |
| --- |
| <?php  **include**("./includes/user.php");  **if** ($\_SERVER["REQUEST\_METHOD"] != "POST") {  header("Location: ./index.php");  }  $currUser = **new** User();  $currUser->fillFromSql($\_POST["u"], $\_POS |

**displaystats.php:**

|  |
| --- |
| <?php  **include**("./includes/user.php");  session\_start();  $\_SESSION["loginErr"] = "";  $currUser = **new** User();  $result = $currUser->fillFromSql($\_SESSION["user"], $\_SESSION["pass"]);  **if** ($result == -**1**){  //If not user found, set login error message  $\_SESSION["loginErr"] = "No user with that username and password";  //Send back to main page  header("Location: ../index.php");  }  ?>  <?php  **include**("./includes/header.php");  ?>  <div id="response">  <?php  **echo** $currUser->printStats();  ?>  </div>  <?php  **include**("./includes/footer.html");  ?> |

**edituser.php:**

|  |
| --- |
| <?php  **include**("./includes/user.php");  **if** ($\_SERVER["REQUEST\_METHOD"] != "POST") {  header("Location: ./index.php");  }  $currUser = **new** User();  $currUser->fillFromSql($\_POST["u"], $\_POST["p"]);  $currUser->printStats();  **echo** "<br>";  $currUser->setUser($\_POST["username"]);  $currUser->setPass($\_POST["password"]);  $currUser->setfName(ucfirst($\_POST["fname"]));  $currUser->setlName(ucfirst($\_POST["lname"]));  $currUser->setWins($\_POST["wins"]);  $currUser->setLosses($\_POST["losses"]);  $currUser->setTies($\_POST["ties"]);  $currUser->setAdmin($\_POST["admin"]);  $currUser->updateSql();  header("Location: ./allusers.php"); |

**index.php:**

|  |
| --- |
| <!-- index.php -->  <?php  session\_start();  $\_SESSION["loggedIn"] = **false**;  ?>  <?php  **include**("./includes/header.php");  ?>  <div id="login">  <form method="post" action="./loginhandler.php" onsubmit="return validateLoginForm()">  <div>  <label for="user">Username:</label>  <input type="text" name="user" id="user">  </div>  <div>  <label for="pass">Password:</label>  <input type="password" name="pass" id="pass">  </div>  <div class="submit-group">  <input type="submit" value="Login">  </div>  <div class="submit-group">  <input type="button" value="Register" onclick="window.location.href='registerform.php'">  </div>  </form>  </div>  <div id="response">  <?php  **if** ($\_SESSION['err'] ?? **false**){  **echo** $\_SESSION["err"];  }  $\_SESSION["err"] = "";  ?>  </div>  <script>  function validateLoginForm() {  var user = document.getElementById("user").value;  var pass = document.getElementById("pass").value;  if (user === "" || pass === "") {  alert("Username and Password must be filled out");  return false;  }  return true;  }  </script>  <?php  **include**("./includes/footer.html");  ?> |

**loginhandler.php:**

|  |
| --- |
| <?php  **include**("./includes/user.php");  session\_start();  $currUser = **new** User();  $result = $currUser->fillFromSql($\_POST["user"], $\_POST["pass"]);  //If user and password match database  **if** ($result != -**1**){  //Save login to session  $\_SESSION["loggedIn"] = **true**;  $\_SESSION["user"] = $\_POST["user"];  $\_SESSION["pass"] = $\_POST["pass"];  $\_SESSION["err"] = "";    //Send to user stats page  header("Location: ./displaystats.php");  }  **else**{  //If not user found, set login error message  $\_SESSION["err"] = "No user with that username and password";  //Send back to main page  header("Location: ./index.php");  } |

**playgame.php:**

|  |
| --- |
| <?php  **include**("./includes/header.php");  **include**("./includes/gameboard.html");  **include**("./includes/footer.html");  ?> |

**registerform.php:**

|  |
| --- |
| <!-- registerform.php -->  <?php  session\_start();  $\_SESSION["loggedIn"] = **false**;  ?>  <?php  **include**("./includes/header.php");  ?>  <div id="login">  <form method="post" action="./registerhandler.php" onsubmit="return validateRegisterForm()">  <div>  <label for="user">Username:</label>  <input type="text" name="user" id="user">  </div>  <div>  <label for="pass">Password:</label>  <input type="text" name="pass" id="pass">  </div>  <div>  <label for="fname">First Name:</label>  <input type="text" name="fname" id="fname">  </div>  <div>  <label for="lname">Last Name:</label>  <input type="text" name="lname" id="lname">  </div>  <div class="radio-group">  <label for="admin">Admin:</label>  Yes<input type="radio" name="admin" value="1">  No<input type="radio" name="admin" value="0" checked>  </div>  <div class="submit-group">  <input type="submit" value="Register" class="spacing">  <button type="button" class="spacing" onclick="window.location.href='./index.php'">Return to Login Menu</button>  </div>  </form>  </div>  <style>  .submit-group {  display: flex;  flex-direction: column;  align-items: center; /\* Centers the buttons horizontally \*/  }  .submit-group .spacing {  margin-bottom: 10px; /\* Adjust the spacing as needed \*/  }  </style>  <div id="response">  <?php  **echo** $\_SESSION["err"];  $\_SESSION["err"] = "";  ?>  </div>  <script>  function validateRegisterForm() {  var user = document.getElementById("user").value;  var pass = document.getElementById("pass").value;  var fname = document.getElementById("fname").value;  var lname = document.getElementById("lname").value;  if (user === "" || pass === "" || fname === "" || lname === "") {  alert("All fields must be filled out");  return false;  }  if (pass.length < 5) {  alert("Password must be at least 5 characters long");  return false;  }  return true;  }  </script>  <?php  **include**("./includes/footer.html");  ?> |

**registerhandler.php:**

|  |
| --- |
| <?php  **if** ($\_SERVER["REQUEST\_METHOD"] == "POST") {  session\_start();  **include**("./includes/user.php");  $currUser = **new** User();  $currUser->setUser($\_POST["user"]);  $currUser->setPass($\_POST["pass"]);  $currUser->setfName(ucfirst($\_POST["fname"]));  $currUser->setlName(ucfirst($\_POST["lname"]));  $currUser->setAdmin($\_POST["admin"]);    $result = $currUser->addToSql();  } **else** {  header("Location: ./index.php");  }  //If there is a user with that username already  **if** ($result == -**1**){  //Save error message and send back to register page  $\_SESSION["err"] = "Username already taken";  header("Location: ./registerform.php");  }  **else**{  //If no user with that username, set "err" to account creation confirmation and go back to index  $\_SESSION["err"] = "User Account Created!";  header("Location: ./index.php");    } |

**savestats.php:**

|  |
| --- |
| <?php  **include**("./includes/user.php");  session\_start();  $\_SESSION["loginErr"] = "";  $currUser = **new** User();  $result = $currUser->fillFromSql($\_SESSION["user"], $\_SESSION["pass"]);  **if** ($result == -**1**){  //If not user found, set login error message  $\_SESSION["loginErr"] = "No user with that username and password";  //Send back to main page  header("Location: ./index.php");  }  **switch** ($\_POST["result"]){  **case** **1**:  //Win  $currUser->setWins($currUser->getWins() + **1**);  **break**;  **case** **2**:  //Loss  $currUser->setLosses($currUser->getLosses() + **1**);  **break**;  **case** **3**:  //Tie  $currUser->setTies($currUser->getTies() + **1**);  **break**;  }  $currUser->updateSql();  **if** ($\_POST["replay"]){  header("Location: ./playgame.php");  }**else** {  header("Location: ./displaystats.php");  }  ?> |

**6.2** **Testing Results**

