Al Imam Muhammad ibn Saud Islamic University
College of Computer and Information System
Computer Science Department
2nd Semester 1443 H – 2022 G

CS 438 – Internet Technologies

Project

Penguin run

BY

STUDENT NAME	ID
ABDULAZIZ ALBARRAK	439015815
OMAR YAHYA MUSAYRI	439011747
ABDULAZIZ ALOTHMAN	439013755

5/12/2022

In this project, we have built an imitation of the classic dino game of google chrome, with a database that is connected to the login, registration of accounts and score pages. Using htm, javascript and php. It was a fruitful experience and a learning leap in our knowledge in web development.

1. Flow Chart

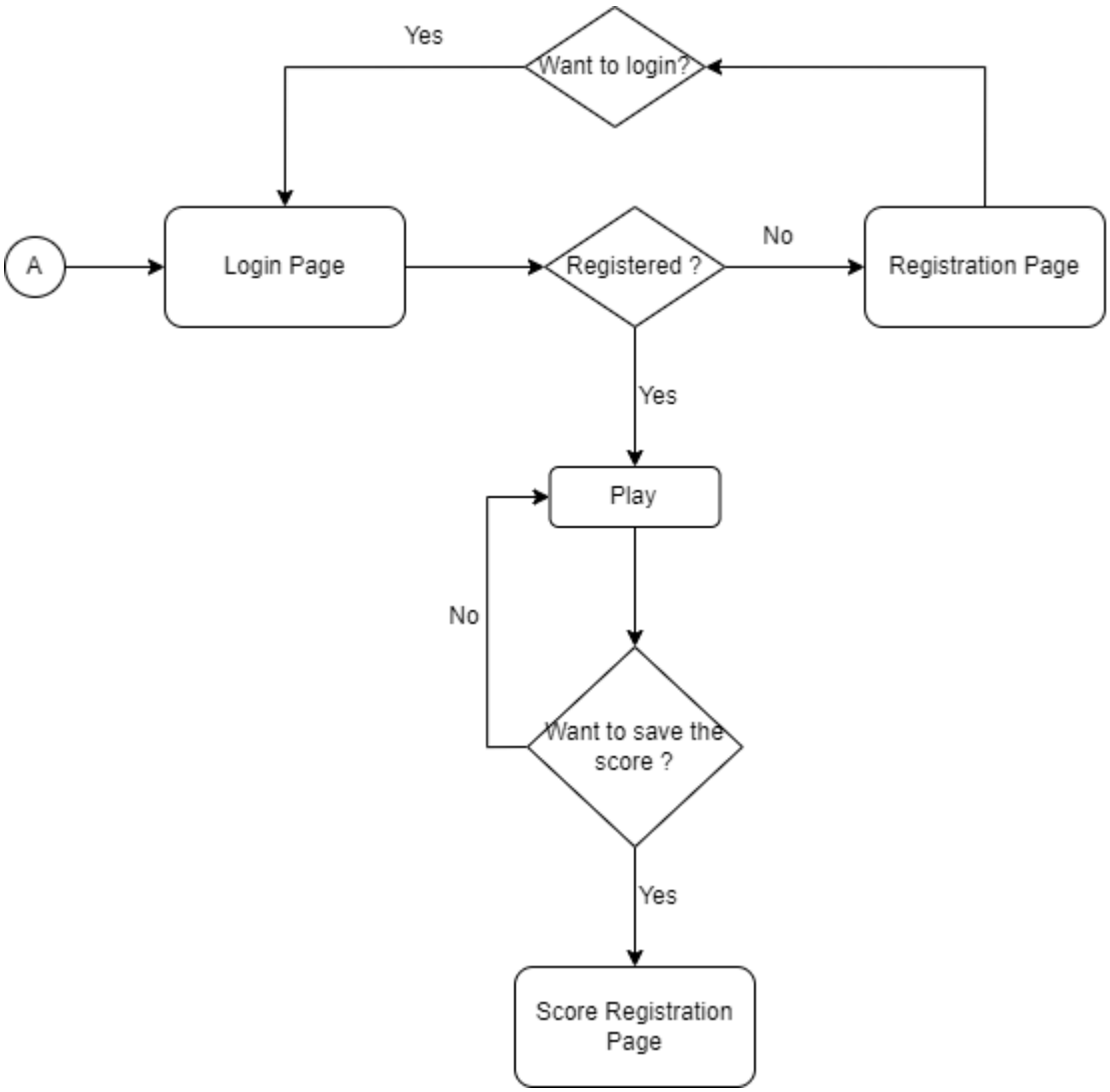

Figure 1 the game diagram

2. Look & Feel

The design of the loging page was designed to make it easier for the user to choose between login or register, if the register was chosen the user is transferred to the registration page after filling the spaces the user gets registered in the database. Inside the game, the character we chose is a Penguin from a game we played with a cloud in the background and an ice-looking land.

3. Dynamic Components

On the web page main.js, a (jump) function was created for the jumping of the Penguin character in the game, a (check) Function checks if any collision occurs if not the score keeps increasing. a (restart_btn) is used to restart the game after dying.

```
function jump(){
    if(jumped) return
    dino.classList.add('jump')
    jumped = true
    setTimeout(function(){
        dino.classList.remove('jump')
        jumped = false
    }, 800)
}

window.addEventListener('space', function(){
    jump()
})
window.addEventListener('keyup', function(e){
    if(e.key === " "){
        jump()
    }
})

```

```

restartBtn.addEventListener('click', function(){
    document.querySelector('.game').classList.remove('lost')
    restartBtn.style.display = 'none'
    score = 0
    scoreBoard.innerText = score
    hiscoreBoard.innerText = hiscore
    check()
})

```

```
function check(){
    if(cactus.offsetLeft <= 10 + dinoWidth && cactus.offsetLeft > 10){
        if(dinoHeight + dino.offsetTop > cactus.offsetTop){
            document.querySelector('.game').classList.add('lost')
            restartBtn.style.display = "block"
            storeScoreBtn.style.display = "block"
            return
        }
    }
    score++
    if(hiscore < score){
        hiscore = score
    }
    scoreBoard.innerText = score
    hiscoreBoard.innerText = hiscore

    requestAnimationFrame(check)
}
```

4. Business Logic

The database is set up as a list to save the players' information, including usernames, email, score and passwords. Furthermore, the player's score is preserved as a zero after registering, allowing any new player to register with this information and be added to the database. And the player's score is revealed after the user enters the score registration page.

In terms of a database call, after the player fills out the form on the (login.php) page, the data is sent to (login.php) via , and the data is retrieved from the database and verified. The player is transported to the beginning of the game displayed on the web page (index.php) based on the data supplied, and if the data sent is correct. For further information, please see the following image:

```
session_start();
include('config.php');
if (isset($_POST['login'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    $query = $connection->prepare("SELECT * FROM users WHERE username=:username");
    $query->bindParam("username", $username, PDO::PARAM_STR);
    $query->execute();
    $result = $query->fetch(PDO::FETCH_ASSOC);
    if (!$result) {
        echo '<p class="error">Username password combination is wrong!</p>';
    } else {
        if (password_verify($password, $result['password'])) {
            $_SESSION['user_id'] = $result['id'];
            echo '<p class="success">Congratulations, you are logged in!</p>';
            header("location: dino");
            exit();
        } else {
            echo '<p class="error">Username password combination is wrong!</p>';
        }
    }
```

```php
session_start();
include('config.php');
if (isset($_POST['register'])) {
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $password_hash = password_hash($password, PASSWORD_BCRYPT);
    $query = $connection->prepare("SELECT * FROM users WHERE email=:email");
    $query->bindParam("email", $email, PDO::PARAM_STR);
    $query->execute();
    if ($query->rowCount() > 0) {
        echo '<p class="error">The email address is already registered!</p>';
    }
    if ($query->rowCount() == 0) {
        $query = $connection->prepare("INSERT INTO users(username,password,email) VALUES (:username,:password_hash,:email)");
        $query->bindParam("username", $username, PDO::PARAM_STR);
        $query->bindParam("password_hash", $password_hash, PDO::PARAM_STR);
        $query->bindParam("email", $email, PDO::PARAM_STR);
        $result = $query->execute();
        if ($result) {
            echo '<p class="success">Your registration was successful!</p>';
        } else {
            echo '<p class="error">Something went wrong!</p>';
        }
    }
```

```php
session_start();
include('config.php');
if (isset($_POST['login'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    $userscore = $_POST['userscore'];
    $query = $connection->prepare("SELECT * FROM users WHERE username=:username");
    $query->bindParam("username", $username, PDO::PARAM_STR);
    $query->execute();
    $result = $query->fetch(PDO::FETCH_ASSOC);
    if (!$result) {
        echo '<p class="error">Username password combination is wrong!</p>';
    } else {
        if (password_verify($password, $result['password'])) {
            $_SESSION['user_id'] = $result['id'];
            $query = $connection->prepare("UPDATE users SET score='" . $userscore . "' WHERE username='" . $_POST['username'] . "'");
            $result = $query->execute();
            if ($result) {
                echo '<p class="success">Your score update was successful!</p>';
                echo '<p class="success">Congratulations, your socre is in!</p>';
                //header("location: dino");
            } else {
                echo '<p class="error">Something went wrong!</p>';
            }


        } else {
            echo '<p class="error">Username password combination is wrong!</p>';
        }
```

And through the code in the web page (index.php) the save button is linked with and through the page (scoreReg.php) and updates the player's score through this code: $ sql = "UPDATE user SET score = '$scorre' WHERE full_name= '$username'" ,the player information is called with the code:$sql = "SELECT id,full_name, score FROM user ORDER BY score DESC".

GitHub link:

https://github.com/oymmusyri/CS438-project

References

1- https://thecodingtrain.com/CodingChallenges/147-chrome-dinosaur.html

2- https://youtu.be/l0HoJHc-63Q

3- https://stackoverflow.com/questions/17902483/show-values-from-a-mysql-database-table-inside-a-html-table-on-a-webpage#:~:text=First%2C%20connect%20to%20the%20database,if%20the%20URL%20was%20%2Findex

4- https://www.php.net/manual/en/pdo.query.php