

動態網頁程式設計 Group 16 Final Project Report

Digital Art Walk

1. 工作分配

組員	工作內容	占比
111550024 蔡芳慈	展間、查詢作品	33%
111550108 吳佳諭	收藏頁、登入註冊	33%
111550168 陳奕	展覽廳、查詢作品	33%

2. 目標

傳統博物館網站僅提供靜態的展示，缺乏沉浸式與互動式的體驗，我們希望透過這樣一個互動式的網站，提升大眾對觀展的興趣，也對展品有所了解，增加觀賞意願，不只達到寓教於樂的效果，也鼓勵藝術工作者開展，提升整個社會的文藝氛圍。

3. 介紹

網站主要分成展區和個人收藏兩個部分。

a. 展區

一進入網站會先到展覽館大廳，透過鍵盤wasd移動卡比進入展間或是其他頁面。

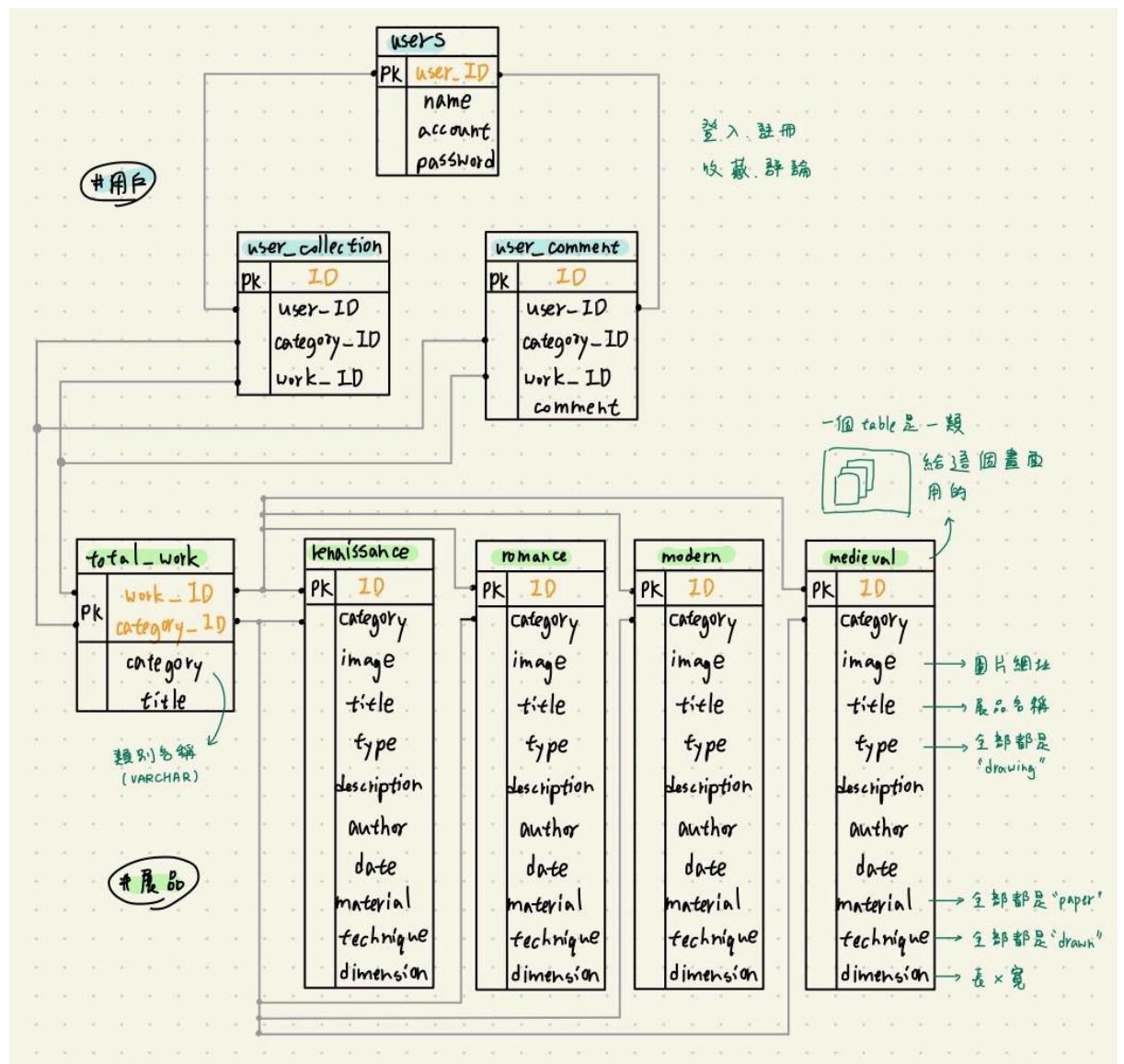
展間中可以透過鍵盤上下在展間中往前、往後觀看展品，每個展品旁有他的介紹，如果用戶有登入的話也可以收藏該展品或是對展品留下評論。

b. 個人收藏

用戶登入之後可以收藏喜歡的展品，然後到收藏頁面查看過去收藏過的展品，或是取消收藏品。點擊每個作品都可以顯示該作品詳細資料，包括名稱、畫家、創作年代、介紹等，也可以看到其他使用者的留言、或是自己給展品留言。

4. 程式

a. 資料庫



我們的資料庫主要分成兩部分，一部分是跟用戶相關的，包括用戶的基本資料(users)、收藏(user_collection)、以及評論(user_comment)，另一部分則是和展品相關的，包括全部展品(total_work)以及分類的展品，一類展品就是一個展間。用戶之間用user_ID相連，用戶和作品則是以 work_ID (作品ID) 和 category_ID (類別ID) 當作查詢條件。

b. 登入 / 註冊

```
<form method="post" action="utils/login.php">
    <div class="form-block">
        <label for="email" class="form-label">Email<span class="red-star">*</span></label>
        <input type="email" id="email" name="email" class="form-input" placeholder="example123@gmail.com">
    </div>
    <div class="form-block">
        <label for="password" class="form-label">Password<span class="red-star">*</span></label>
        <input type="password" id="password" name="password" class="form-input" placeholder="*****">
    </div>
    <p class="error-msg" id="wrong-pw">Incorrect password. Please try again.</p>
    <p class="error-msg" id="no-account">The account does not exist. <a href="register.html" style="color: black; font-weight: bold;">Create</a> an account!</p>
    <p class="error-msg" id="no-input">Please enter your email and password.</p>
    <p class="error-msg" id="account-exist">The account already exists. Please login.</p>
    <input type="submit" class="form-btn" value="LOGIN">
    <p class="form-bottom">Don't have an account yet? <a href="register.html" style="color: black; font-weight: bold;">Register</a> now.</p>
</form>
```

login.html (register.html), HTML的部分用form將資料傳給後端處理。

```
$mail = $_POST['email'];
$password = $_POST['password'];
$hash_pw = hash('sha256', (string)$password);

// check valid input
if(empty($mail) || empty($password)){
    // no input
    header("Location: ../login.html?state=no-input");
    return;
}

$sql = "SELECT * FROM `users` WHERE `account` = '$mail' AND `password` = '$hash_pw'";
$res = $DB -> query($sql);
if($res->num_rows > 0){
    // success login

    // get user id
    $sql = "SELECT user_ID FROM `users` WHERE account = '$mail'";
    $id_ret = $DB -> query($sql);
    $user_id = $id_ret -> fetch_assoc();
    $user_id = (int)$user_id['user_ID'];

    // use js to redirect instead of php
    echo "<script>localStorage.setItem('user_email', '$mail'); localStorage.setItem('user_id', $user_id); window.location.href = '../index.html';</script>";
    // header("Location: ../index.html");
    return;
}
```

login.php, 後端接收到資料會先檢查是否合法 (是否有輸入東西), 若不合法則返回HTML, 並在網址後面附上error message, 讓HTML和JS可以將錯誤提示顯示在登入頁上。密碼會先用sha256加密後才存到資料庫, 增加一定的安全性。登入成功會把user_ID、user_account存在localStorage, 方便之後取用, 登出時再把資料刪除。

登入頁右邊的跑馬燈則是用CSS搭配資料庫(login.css, login.js, right_content.php)完成的, 另外動畫時間也會隨著跑馬燈長度決定, 若較多用戶評論, 跑馬燈內容較多, 則延長動畫時間, 讓整體速度不會過快。

c. 展覽大廳

```
<canvas id="gameCanvas" style="left:0%;top:0%;"></canvas>

<div id="lobby">
    <div id="Renaissance" class="painting">
        
        <div id="exb_name">renaissance</div>
    </div>
    <div id="Romance" class="painting">
        
        <div id="exb_name">romance</div>
    </div>
    <div id="Modern" class="painting">
        
        <div id="exb_name">modern</div>
    </div>
    <div id="npc_dog" class="dailog">
        
        <div id="dailog">Burf!</div>
    </div>
    <div id="collection" class="painting">
        
        <div id="exb_name">collection</div>
    </div>
</div>
<form id="paintingForm" action="exb.php" method="POST" style="display: none;">>...
</form>
<div id="searchBox" style="display: none; position: absolute; z-index: 1000;">>...
</div>
<form id="searchingForm" action="search.php" method="POST" style="display: none;">>...
</form>
</body>
```

homepage.html 負責基本頁面框架，包括 header 區塊顯示用戶問候語和登出按鈕、Canvas 畫布作為角色互動區域，以及 lobby 區域展示可互動的畫作與 NPC。paintingForm與searchingForm則是負責POST user_id及其他必要資訊給後端接收。

```
function gameLoop() {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    drawBackground();
    //drawWalkablePolygon(); // 繪製多邊形

    updatePlayer();      // 更新和繪製玩家
    updatePlayerGifPosition();

    checkProximityToPaintings();
    // 重複呼叫gameLoop
    requestAnimationFrame(gameLoop);
}
```

homepage_kirby.js 負責遊戲功能實現，在遊戲迴圈中更新玩家角色的移動和動畫，通過監聽鍵盤事件(a, w, s, d)改變角色在畫布上的位置。

```

// 可行走區域 (地面範圍)
const walkablePolygon = [
  { x: canvas.width * 0.05, y: canvas.height * 0.95 },
  { x: canvas.width * 0.3, y: canvas.height * 0.65 },
  { x: canvas.width * 0.6, y: canvas.height * 0.65 },
  { x: canvas.width * 0.88, y: canvas.height * 0.95 }
];
function isPointInPolygon(point, polygon) {
  let x = point.x, y = point.y;
  let inside = false;

  for (let i = 0, j = polygon.length - 1; i < polygon.length; j = i++) {
    let xi = polygon[i].x, yi = polygon[i].y;
    let xj = polygon[j].x, yj = polygon[j].y;

    let intersect = ((yi > y) !== (yj > y)) &&
      ((x < (xj - xi) * (y - yi) / (yj - yi) + xi);
    if (intersect) inside = !inside;
  }
  return inside;
}

```

updatePlayer()功能實現設置多邊形範圍檢測，確保玩家只能在指定區域內行走，模擬真實展覽空間。

```

let activePainting = null;
function checkProximityToPaintings() {
  const playerCenter = { x: player.x + player.width / 2, y: player.y + player.height / 2 };
  const paintings = document.querySelectorAll('.painting');
  const npRect = door.getBoundingClientRect();

  paintings.forEach(painting => {
    const rect = painting.getBoundingClientRect();
    const paintingBottom = { x: rect.left + rect.width / 2, y: rect.bottom };

    const distance = Math.sqrt(
      Math.pow(playerCenter.x - paintingBottom.x, 2) +
      Math.pow(playerCenter.y - paintingBottom.y, 2)
    );

    if (distance < 200) {
      activePainting = painting;
      //console.log('Player is near a painting.', painting);
      painting.classList.add('active');
      painting.style.transform = 'scale(1.2)';
      painting.style.opacity = '1';

    } else {
      if (painting.classList.contains('active')) {
        //console.log('Player is not near a painting.');
        activePainting = null;
        painting.classList.remove('active');
        painting.style.transform = 'scale(1)';
        painting.style.opacity = '0.8';
      }
    }
  });
}

// 按下空白鍵導向展廳頁面
document.addEventListener('keydown', (event) => {
  if (event.code === 'Space' && activePainting) {
    const paintingName = activePainting.textContent.trim();

    if (paintingName === "collection") {
      console.log('Redirecting to collection page...');

      window.location.href = 'collection.html';
    } else {
      // 將畫作名稱填入隱藏表單
      const userID = localStorage.getItem('user_id');
      document.getElementById('paintingName').value = paintingName;
      document.getElementById('user_ID').value = userID;
      console.log('User ID:', userID);
      console.log('Painting name:', paintingName);

      // 提交表單
      document.getElementById('paintingForm').submit();
    }
  }
});

```

checkProximityToPainting() 與按下空白鍵進入展廳實現方法是當玩家靠近畫作時，程式會檢測玩家與物件的距離並觸發互動效果，如畫作縮放。當用戶按下空白鍵後，程式會將畫作的相關數據填入隱藏表單，並將表單提交至後端的 exb.php，隨後重定向用戶進入展廳頁面。

d. 展間

分為以下檔案：

1. exb.php：由資料庫匯入展區作品資訊、留言、使用者個人收藏，根據資料自動生成網頁

```
$user_ID = 1;
$table = 'renaissance';
if ($table == 'renaissance') {
    $category_ID = 1;
} else if ($table == 'romance') {
    $category_ID = 2;
} else if ($table == 'morden') {
    $category_ID = 3;
} else {
    $category_ID = 4;
}

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $table = $_POST['table'];
    $user_ID = $_POST['user_ID'];
}

$query = "SELECT * FROM " . $table;
$stmt = $conn->query($query);
$artworks = $stmt->fetchAll(PDO::FETCH_ASSOC);
echo "<script>sessionStorage.setItem('artworkCount', " . count($artworks) . ");</script>";
setcookie(name: 'user_ID', value: $user_ID, expires_or_options: time() + (86400 * 30), path: "/");
setcookie(name: 'category_ID', value: $category_ID, expires_or_options: time() + (86400 * 30), path: "/");
```

接收表單存取來自大廳的使用者資訊以及展廳類別，並存入Cookie以供react.php更新資料庫時使用。

```
<?php foreach ($artworks as $artwork): ?>
    <div class="info" id="info<?php echo $artwork['ID'] ?>">
        <h1><?php echo $artwork['title'] ?></h1>
        <h3 class="author"><?php echo $artwork['author'] ?></h3>
        <h4 class="date"><?php echo $artwork['date'] ?></h4>
        <hr>
        <p><?php echo $artwork['description'] ?></p>
    </div>
<?php endforeach; ?>
```

分析資料庫取得的展品資料，自動生成多個div區塊。此實作方式可保證在資料更新時，網站會自動更新，不須再另作修改，提升網站彈性。

```

<div id="allcomments">
    <?php foreach ($artworks as $artwork): ?>
        <div id="allcomments<?php echo $artwork['ID'] ?>" class="allcomments">
            <?php
                $query = "SELECT * FROM user_comment WHERE work_ID = " . $artwork['ID'] . " AND category_ID = " . $category_ID;
                $stmt = $conn->query($query);
                $comments = $stmt->fetchAll(PDO::FETCH_ASSOC);
                if (empty($comments)) {
                    echo "<p>No comment yet.</p>";
                }
            ?>
            <?php foreach ($comments as $comment): ?>
                <?php
                    $query = "SELECT * FROM users WHERE user_ID = " . $comment['user_ID'];
                    $stmt = $conn->query($query);
                    $user = $stmt->fetch(PDO::FETCH_ASSOC);
                ?>
                <div class="comment">
                    <p class="account"><?php echo $user[0]['name'] ?></p>
                    <p><?php echo $comment['comment'] ?></p>
                </div>
            <?php endforeach; ?>
        </div>
    <?php endforeach; ?>

```

自資料庫取得該展區各展品的留言，並根據留言者ID查詢使用者名稱，生成留言區內容。

2. exb.js: 處理按鍵事件，動態調整排版

```

document.getElementById('info1').style.display = 'block';
document.getElementById('allcomments1').style.display = 'block';
document.getElementById('artworks').style.left = 25 + '%';
document.getElementById('artworks').style.top = 42.5 + '%';
for (let i = 0; i < num_perartwork; i++) {
    artworks[i].style.opacity = 1;
    positionx[i] = 50 + i * 10;
    positiony[i] = 65 - i * 10;
    artworks[i].style.left = positionx[i] + '%';
    artworks[i].style.top = positiony[i] + '%';
    artworks[i].style.zIndex = num_perartwork - i;
    artworks[i].style.width = 50 - 8 * i + 'vh';
    artworks[i].style.height = 50 - 8 * i + 'vh';
}

```

規律計算圖片位置、大小，並利用此規則在按鍵事件發生時更新所有展品位置。當下方向鍵被按下，所有展品位置、大小將變成其上一個展品的位置、大小，並將最前的展品隱藏，反之亦然。此流動性的畫作展示提供自然的視覺體驗，彷彿使用者正身入其境。

```

let data = { 'workID': userposition, 'operation': 'collect' };
fetch('react.php', {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json',
    },
    body: JSON.stringify(data),
})

```

在使用者按下收藏、發送留言時，發出表單更新資料庫，上圖是使用者新增收藏時的邏輯，表單涵蓋操作名稱及藏品順序，傳至react.php之後再結合cookie內容產生sql。

3. exb.css: 排版
4. react.php: 將使用者的輸入資料更新至資料庫

```

if($_SERVER['REQUEST_METHOD'] == 'POST') {
    $data= json_decode(json: file_get_contents(filename: 'php://input'), associative: true);
    $workID = isset($data['workID']) ? $data['workID'] : null;
    if($data['operation']=='comment'){
        $comment = $data['comment'];
        $query= "INSERT INTO user_comment (user_ID, category_ID, work_ID, comment) VALUES ($user_ID, $category_ID, $workID, '$comment')";
    }
    else if($data['operation']=='collect') {
        $query= "INSERT INTO user_collection (user_ID, category_ID, work_ID) VALUES ($user_ID, $category_ID, $workID)";
    }
    else{
        $query= "DELETE FROM user_collection WHERE user_ID = $user_ID AND category_ID = $category_ID AND work_ID = $workID";
    }
    $stmt = $conn->prepare(query: $query);
    if($stmt->execute()){
    }
    else{
        echo 'fail';
    }
}

```

取得來自exb.js的表單資料以及cookie內容，先判斷使用者操作類型，根據操作生成相關的sql，處理收藏、取消收藏、發布留言等操作。

5. walk.js: 處理卡比位置相關的按鍵事件、移動邏輯

```

if (event.code === "KeyA") {
    if (nowX < 90) {
        if (nowDirection != 2) gif.src = 'character/walk_l.gif';
        nowX += 1;
        gifdiv.style.right = nowX + '%';
        nowDirection = 2;
    }
}
else if (event.code === "KeyD") {
    if (nowX > 5) {
        if (nowDirection != 1) gif.src = 'character/walk_r.gif';
        nowX -= 1;
        gifdiv.style.right = nowX + '%';
        nowDirection = 1;
    }
}

else if (event.code === "Space") {
    if (nowX < 10 && nowX > 4) {
        nowDirection = 0;
        let interval = setInterval(function () {
            nowY += 10;
            gifdiv.style.bottom = nowY + 'px';
        }, 20);
        let interval2;
        this.setTimeout(function () {
            clearInterval(interval);
            interval2 = setInterval(function () {
                nowY -= 10;
                gifdiv.style.bottom = nowY + 'px';
            }, 20);
        }, 300);
        this.setTimeout(function () {
            clearInterval(interval2);
            window.location.href = 'homepage.html';
        }, 600);
    }
}

```

感應鍵盤事件，動態調整卡比在畫面上的位置以及方向。卡比position設定fixed，根據按鍵A或D增減x軸上的位置。並在抵達門口時，以js動畫定時更新y座標，呈現卡比原地跳，並返回大廳。

e. 查詢

```
//check proximity to npc;
const npcCenter = {
  x: npcRect.left + npcRect.width / 2,
  y: npcRect.top + npcRect.height / 2,
};

const distance = Math.sqrt(
  Math.pow(playerCenter.x - npcCenter.x, 2) +
  Math.pow(playerCenter.y - npcCenter.y, 2)
);

if (distance < canvas.height * 0.1) {
  console.log('Player is near the NPC.');
  searchBox.style.display = 'block';
} else {
  searchBox.style.display = 'none';
}

function submitSearch() {
  const keyword = document.getElementById('searchInput').value;
  if (keyword.trim() === "") {
    alert("Please enter a keyword to search.");
    return;
  }
  document.getElementById('keyword').value = keyword;
  const userID = localStorage.getItem('user_id');
  document.getElementById('user_ID2').value = userID;
  document.getElementById('searchingForm').submit();
}

function cancelSearch() {
  console.log('Search cancelled.');
  //clear input
  searchInput.value = '';
  searchBox.style.display = 'none';
}
```

homepage_kirby.js 程式檢測玩家角色與 NPC 的距離，顯示搜尋框供用戶輸入關鍵字。用戶在搜尋框輸入關鍵字後，按下「搜尋」按鈕觸發 submitSearch 函數，將關鍵字和用戶 ID 填入隱藏表單，並以 POST 方式提交至後端 PHP 腳本。



```
$host = 'localhost';
$dbname = 'final_proj';
$username = 'root';
$password = '';

try {
  $conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
  $conn->exec("set names utf8");
} catch (PDOException $e) {
  echo "連接失敗: " . $e->getMessage();
}

$user_ID = 1;
$search_keyword = '';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  $search_keyword = $_POST['keyword'];
  $user_ID = $_POST['user_ID'];
}

// 搜尋四個 table
$query = "
SELECT 'renaissance' AS table_name, ID, title, description, image, author, date
FROM renaissance WHERE title LIKE :keyword OR description LIKE :keyword
UNION ALL
SELECT 'romance' AS table_name, ID, title, description, image, author, date
FROM romance WHERE title LIKE :keyword OR description LIKE :keyword
UNION ALL
SELECT 'modern' AS table_name, ID, title, description, image, author, date
FROM modern WHERE title LIKE :keyword OR description LIKE :keyword
UNION ALL
SELECT 'medieval' AS table_name, ID, title, description, image, author, date
FROM medieval WHERE title LIKE :keyword OR description LIKE :keyword
";

$stmt = $conn->prepare($query);
$stmt->bindValue(':keyword', value: '%' . $search_keyword . '%', type: PDO::PARAM_STR);
$stmt->execute();
$artworks = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

後端根據關鍵字查詢多個資料表（如 renaissance、romance 等），使用 UNION ALL 將結果合併，並將匹配關鍵字的藝術品資料返回至前端展示。如果無搜尋結果，頁面會顯示提示訊息「No results found」。其餘展廳展示與d.展間相同

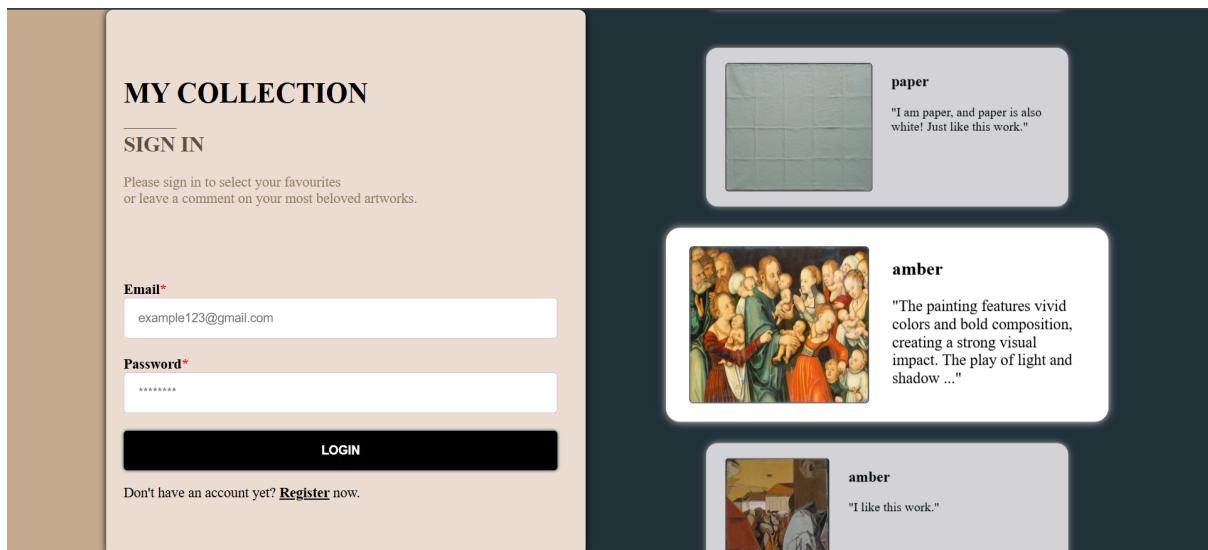
f. 收藏

```
let mail = localStorage.getItem("user_mail");
// redirect to login if haven't login
if(mail == null){
    window.location.href = 'login.html';
}
// render HTML. get collection from DB
$.ajax({
    url: 'utils/collection.php',
    type: 'POST',
    data: {action: "fill content", mail: mail},
    // success, render HTML
    success: function(response){ ... },
    error: function(){
        console.log("collection ajax fail:");
    }
});
```

collection.js, 收藏作品和細節彈框都是JS先用 ajax 送到後端(collection.php), 從資料庫中獲得資料後再回到JS寫上HTML。另外因為只有登入才能訪問這個頁面，所以一開始會先檢查localStorage是否有存user_mail, 若沒有則表示他沒有登入，將倒回登入頁要求他先登入。

5. 結果

a. 登入 / 註冊



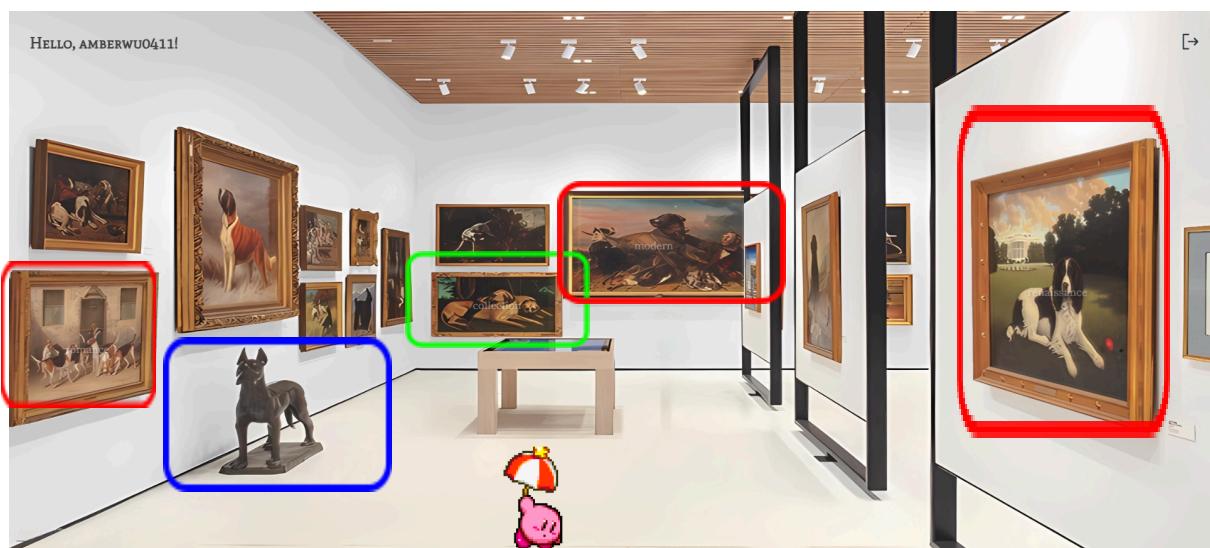
上圖左邊為登入/註冊，輸入信箱和密碼以登入或註冊，若輸入錯誤帳密則會出現錯誤訊息，要再輸入一次，例如沒有輸入、密碼錯誤、帳號不存在等，如下圖。

Please enter your email and password. Incorrect password. Please try again.

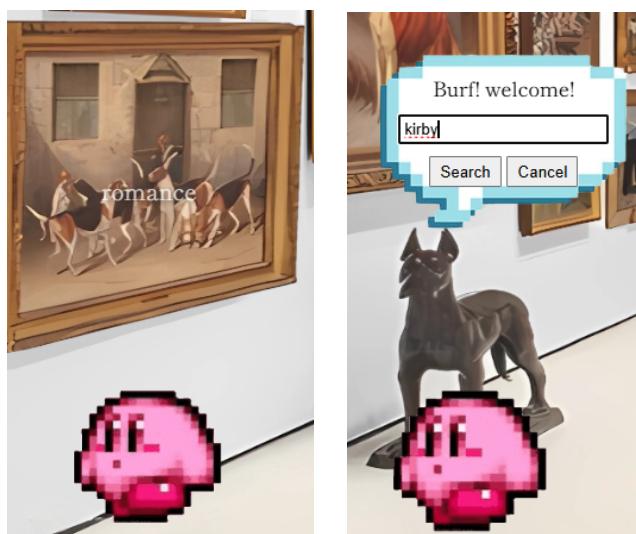
The account does not exist. [Create an account!](#)

右邊是用戶們的評論，是以跑馬燈的方式呈現，當評論滾動到螢幕中間時會放大，其他則將透明度減小降低存在感。

b. 展覽大廳



這是網站的第一頁，頁面載入時卡比會慢慢飛下來，到達地面後，參觀者就可以用鍵盤AWSD控制卡比在大廳中移動，前往不同展廳或頁面。



總共有三間展廳(紅框)、一個收藏頁(綠框，要登入才可以訪問)、和一個搜尋頁(藍框)，一幅畫就代表一個展間或頁面，當卡比靠近該畫作時畫作會變大，點擊空白件即可進入展間。

左圖為進入展間前，右圖為搜尋。

c. 展間



RENAISSANCE



The Adoration of the Magi

Albrecht Altdorfer

1530 - 1535

Altdorfer was born in Regensburg and is the most important representative of the so-called Danube School. This late work is characteristic of its style. The figures and the setting fuse to create a festive scene. Maria's mantle and head covering flow in voluptuous folds, and the Three Kings are opulently dressed. The architecture of the ruins, shown in great detail, is intricate and overgrown with climbing plants. Even the sky is decorated. Extravagant perspectives like the Star of Bethlehem, depicted from below, and the wooden beams above the dark-skinned king heighten the decorative effect.

 Write a comment...

Comments

FCT

hahah



在展間中可以透過鍵盤上下帶著卡比觀賞作品，進入頁面時卡比會從右下角的門進入展間，然後停在畫作前面，往前時畫作會放大然後漸層消失，模仿向前走的效果。如果想返回大廳除了按左上角的按鈕，也可以用鍵盤A/D控制卡比回到門口，從出口回到大廳。點擊畫作下方的放大鏡會跳出作品彈框，可以放大更清楚的查看作品細節。



RENAISSANCE



The Adoration of the Magi

Albrecht Altdorfer

1530 - 1535

Altdorfer was born in Regensburg and is the most important representative of the so-called Danube School. This late work is characteristic of its style. The figures and the setting fuse to create a festive scene. Maria's mantle and head covering flow in voluptuous folds, and the Three Kings are opulently dressed. The architecture of the ruins, shown in great detail, is intricate and overgrown with climbing plants. Even the sky is decorated. Extravagant perspectives like the Star of Bethlehem, depicted from below, and the wooden beams above the dark-skinned king heighten the decorative effect.

FCT

hahah

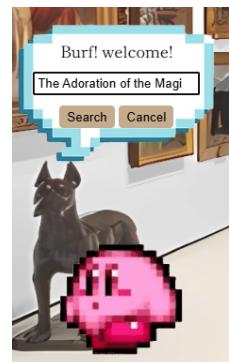


展間左邊是對每一個展品的介紹，留言區可以查看大家的評論，若用戶有登入也可以一起參與討論，在留言區留下自己獨特的見解，與大家交流分享。

此外，使用者可以登入來收藏自己喜歡的作品，點擊作品名稱右邊的星星將畫作加入自己的收藏，下圖上為還沒加入收藏的狀態，加入收藏後星星會亮起來，像下圖下一樣。畫面右上角的寶廂裝著使用者的收藏，點擊寶廂可以跳出彈框顯示收藏概要，如右圖，點擊彈窗即可前往收藏頁面查看更



d. 查詢



當卡比靠近NPC時，搜尋框會自動彈出，使用者可以在這裡搜尋想要找的作品名稱。

按下Search之後卡比會進到作品頁面，裡面是剛才查詢的結果，一樣可以用放大鏡查看圖片，或是評論收藏這個展品。

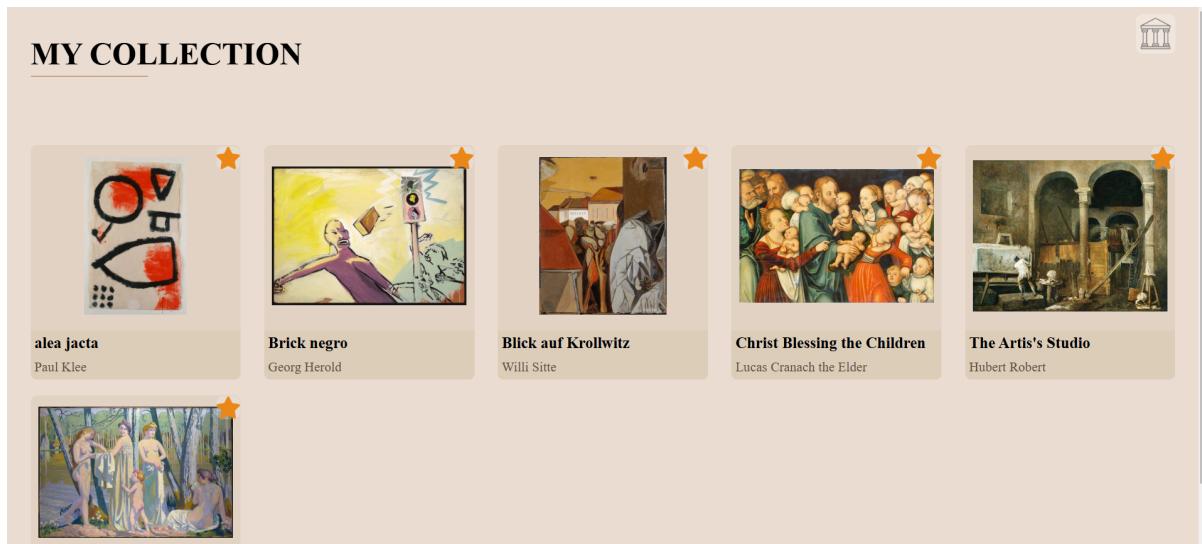
The Adoration of the Magi

Albrecht Altdorfer
1530 - 1535

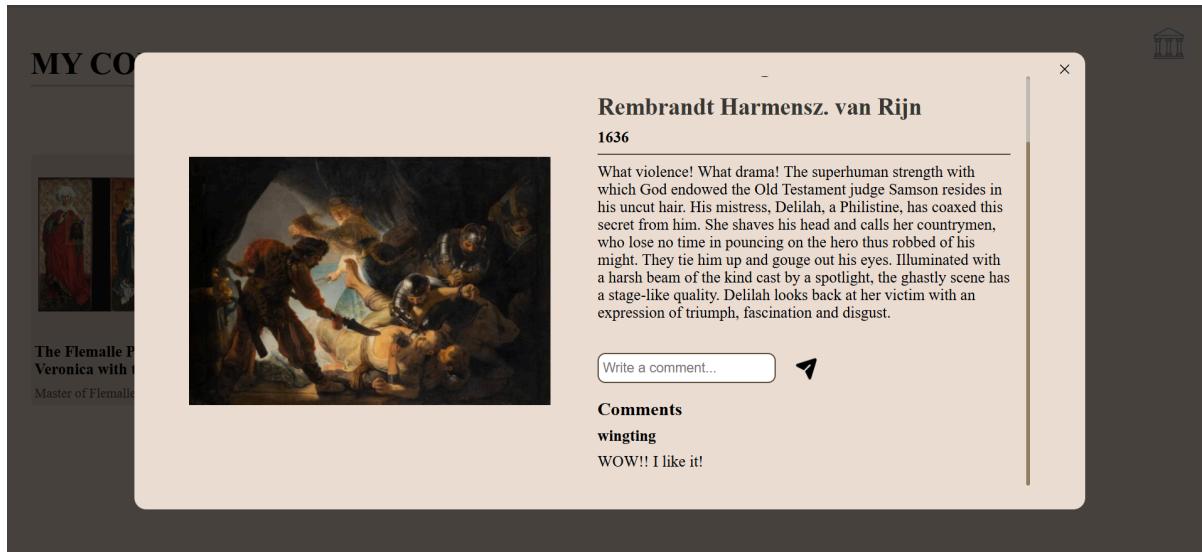
Altdorfer was born in Regensburg and is the most important representative of the so-called Danube School. This late work is characteristic of its style. The figures and the setting fuse to create a festive scene. Maria's mantle and head covering flow in voluptuous folds, and the Three Kings are opulently dressed. The architecture of the ruins, shown in great detail, is intricate and overgrown with climbing plants. Even the sky is decorated. Extravagant perspectives like the Star of Bethlehem, depicted from below, and the wooden beams above the dark-skinned king heighten the decorative effect.

Write a comment...

e. 收藏



用戶登入後可以在這個頁面查看曾經收藏過的畫作。每個畫作的右上角有一顆星，代表用戶有收藏那幅畫。如果不想收藏這幅畫了，可以點擊星星以取消收藏，則星星變為黑色，若後悔了可以在點擊加收回藏，頁面刷新後收藏從資料庫移除，沒辦法在這個頁面看到了。



點擊收藏的畫作可以展開詳細介紹，並且看到別人對這幅作品的留言，自己也可以在這裡加上自己的評論，右上角的星星一樣可以將畫作加入或取消收藏。

6. 未來展望

- 用戶生成內容：自定義展示品，用戶可上傳自己的圖片或藝術作品，在博物館中展示。
- 多人即時參觀：支持多個用戶同時進入博物館，並顯示彼此的角色位置。設計發言框促進即時交流。
- 更真實的參觀體驗：考量人與畫作的尺度，模擬人與畫作的比例關係呈現畫作、或是結合AR技術讓用戶使用手機或平板在現實空間中展示展品。
- 支援多語言：提供更友好觀賞體驗，促進不同語言和文化的交流。

7. 回饋

做完這次final project 對於前後端又更熟悉了，平常的課程當中真的沒什麼機會接觸到前後端，二年級第一次接觸就是資料庫的final project，要我們在毫無基礎的情況下寫出一個網頁，雖然跌跌撞撞地做完了，但根本不知道自己在做什麼，但這一次的project每個都有碰到前端和後端，可以寫自己喜歡、擅長的部份，也可以練習比較不熟悉的部份，寫完真的學到很多。

另外也真正第一次那麼深刻會到不同瀏覽器看到的網站都不一樣，即使大家都chrome看，看到的東西有時候也差很多，尤其我們又有要判斷位置決定是否前往特定頁面，位置稍微一改變整個網頁就壞了，每次在自己這裡測試都可以，到了別人電腦全部排版都變了，再也不敢用絕對位置了，把全部用px設大小的都改成用%，絕對位置都盡量改成相對的了。

展間注重視覺效果且有大量圖片，這次花了特別長時間對圖片進行定位，發現位置差幾百分比都會影響視覺，因此我現在看到版面整潔舒適的網頁會特別佩服，能把UI設計到美觀又直覺的背後肯定有大量經驗的堆砌，學習排版讓我對CSS的應用有更深一層的認識。