會員功能開發懶人包

一、基本設置

1. 引入 useAuth

```
import { useAuth } from '@/app/_hooks/useAuth';

// 使用 hook
const { member, token, isLoggedIn } = useAuth();
```

2. 檢查登入狀態

```
if (!isLoggedIn) {
  // 可以選擇：
  // 1. 跳轉到登入頁面
  router.push('/member/login');
  // 2. 顯示登入提示
  return <div>請先登入</div>;
  // 3. 禁用某些功能
  return <button disabled>請先登入</button>;
}
```

二、會員中心功能 (museum-frontend/app/member/center/features/)

1. 購物車功能

```
// museum-frontend/app/member/center/features/cart/CartTab.js
import { useAuth } from '@/app/_hooks/useAuth';
import { useState, useEffect } from 'react';

export default function CartTab() {
  const { member, token, isLoggedIn } = useAuth();
  const [cartItems, setCartItems] = useState([]);
  const [loading, setLoading] = useState(true);

  // 獲取購物車
  const getCart = async () => {
    try {
      const response = await fetch(
        `${process.env.NEXT_PUBLIC_API_URL}/api/members/cart`,
        {
          headers: {
            'Authorization': `Bearer ${token}`
```

```
          }
        }
      );
      const data = await response.json();
      if (data.success) {
        setCartItems(data.data);
      }
    } catch (error) {
      console.error('獲取購物車失敗:', error);
    } finally {
      setLoading(false);
    }
  };

  useEffect(() => {
    if (isLoggedIn) {
      getCart();
    }
  }, [isLoggedIn]);

  if (!isLoggedIn) return <div>請先登入</div>;
  if (loading) return <div>載入中...</div>;

  return (
    <div>
      <h2>購物車</h2>
      {cartItems.map(item => (
        <div key={item.id}>
          <img src={item.image} alt={item.name} />
          <h3>{item.name}</h3>
          <p>數量: {item.quantity}</p>
          <p>價格: ${item.price}</p>
        </div>
      ))}
    </div>
  );
}
```

2. 優惠券功能

```
// museum-frontend/app/member/center/features/coupons/CouponsTab.js
import { useAuth } from '@/app/_hooks/useAuth';
import { useState, useEffect } from 'react';

export default function CouponsTab() {
  const { member, token, isLoggedIn } = useAuth();
  const [coupons, setCoupons] = useState([]);
  const [loading, setLoading] = useState(true);

  // 獲取優惠券
```

```
  const getCoupons = async () => {
    try {
      const response = await fetch(
        `${process.env.NEXT_PUBLIC_API_URL}/api/members/coupons`,
        {
          headers: {
            'Authorization': `Bearer ${token}`
          }
        }
      );
      const data = await response.json();
      if (data.success) {
        setCoupons(data.data);
      }
    } catch (error) {
      console.error('獲取優惠券失敗:', error);
    } finally {
      setLoading(false);
    }
  };

  useEffect(() => {
    if (isLoggedIn) {
      getCoupons();
    }
  }, [isLoggedIn]);

  if (!isLoggedIn) return <div>請先登入</div>;
  if (loading) return <div>載入中...</div>;

  return (
    <div>
      <h2>我的優惠券</h2>
      {coupons.map(coupon => (
        <div key={coupon.id}>
          <h3>{coupon.name}</h3>
          <p>折扣: {coupon.discount}%</p>
          <p>有效期至: {coupon.expiryDate}</p>
        </div>
      ))}
    </div>
  );
}
```

三、電商頁面功能 (museum-frontend/app/products/)

1. 商品列表頁面

```
// museum-frontend/app/products/page.js
import { useAuth } from '@/app/_hooks/useAuth';
```

```
export default function ShopPage() {
  const { member, token, isLoggedIn } = useAuth();
  const [products, setProducts] = useState([]);

  // 收藏商品
  const handleFavorite = async (productId) => {
    if (!isLoggedIn) {
      router.push('/member/login');
      return;
    }

    try {
      const response = await fetch(

`${process.env.NEXT_PUBLIC_API_URL}/api/members/favorites/products`,
        {
          method: 'POST',
          headers: {
            'Authorization': `Bearer ${token}`,
            'Content-Type': 'application/json'
          },
          body: JSON.stringify({ productId })
        }
      );
      const data = await response.json();
      if (data.success) {
        showToast('success', '收藏成功');
      }
    } catch (error) {
      console.error('收藏失敗:', error);
    }
  };

  return (
    <div>
      {products.map(product => (
        <div key={product.id}>
          <img src={product.image} alt={product.name} />
          <h3>{product.name}</h3>
          <p>${product.price}</p>
          <button onClick={() => handleFavorite(product.id)}>
            {isLoggedIn ? '收藏' : '登入後收藏'}
          </button>
          <button onClick={() => handleAddToCart(product.id)}>
            {isLoggedIn ? '加入購物車' : '登入後購買'}
          </button>
        </div>
      ))}
    </div>
  );
}
```

2. 商品詳情頁面

```
// museum-frontend/app/products/[id]/page.js
import { useAuth } from '@/app/_hooks/useAuth';

export default function ProductDetailPage({ params }) {
  const { member, token, isLoggedIn } = useAuth();
  const [product, setProduct] = useState(null);
  const [quantity, setQuantity] = useState(1);

  // 加入購物車
  const handleAddToCart = async () => {
    if (!isLoggedIn) {
      router.push('/member/login');
      return;
    }

    try {
      const response = await fetch(
        `${process.env.NEXT_PUBLIC_API_URL}/api/members/cart`,
        {
          method: 'POST',
          headers: {
            'Authorization': `Bearer ${token}`,
            'Content-Type': 'application/json'
          },
          body: JSON.stringify({
            productId: params.id,
            quantity
          })
        }
      );
      const data = await response.json();
      if (data.success) {
        showToast('success', '已加入購物車');
      }
    } catch (error) {
      console.error('加入購物車失敗:', error);
    }
  };

  return (
    <div>
      {product && (
        <>
          <img src={product.image} alt={product.name} />
          <h1>{product.name}</h1>
          <p>${product.price}</p>
          <div>
            <button onClick={() => setQuantity(q => Math.max(1, q -
```

```
1))}>-</button>
            <span>{quantity}</span>
            <button onClick={() => setQuantity(q => q + 1)}>+</button>
          </div>
          <button onClick={handleAddToCart}>
            {isLoggedIn ? '加入購物車' : '登入後購買'}
          </button>
        </>
      )}
    </div>
  );
}
```

四、後端部分 (museum-backend)

1. 路由設置

```
// museum-backend/src/routes/
import express from 'express';
import { authenticateToken } from '../middleware/auth.js';
import pool from '../config/database.js';

const router = express.Router();

// 購物車路由
router.get('/cart', authenticateToken, async (req, res) => {
  try {
    const memberId = req.user.id;
    const [cartItems] = await pool.query(
      `SELECT c.*, p.name, p.price, p.image
       FROM cart c
       JOIN products p ON c.product_id = p.id
       WHERE c.member_id = ?`,
      [memberId]
    );

    res.json({
      success: true,
      data: cartItems
    });
  } catch (error) {
    res.status(500).json({
      success: false,
      message: error.message
    });
  }
});

// 優惠券路由
router.get('/coupons', authenticateToken, async (req, res) => {
```

```
  try {
    const memberId = req.user.id;
    const [coupons] = await pool.query(
      `SELECT c.*
       FROM member_coupons mc
       JOIN coupons c ON mc.coupon_id = c.id
       WHERE mc.member_id = ? AND c.expiry_date > NOW()`,
      [memberId]
    );

    res.json({
      success: true,
      data: coupons
    });
  } catch (error) {
    res.status(500).json({
      success: false,
      message: error.message
    });
  }
});

export default router;
```

2. 服務層

```
// museum-backend/src/services/
import pool from '../config/database.js';

// 購物車服務
export const getCartItems = async (memberId) => {
  const [items] = await pool.query(
    `SELECT c.*, p.name, p.price, p.image
     FROM cart c
     JOIN products p ON c.product_id = p.id
     WHERE c.member_id = ?`,
    [memberId]
  );
  return items;
};

// 優惠券服務
export const getMemberCoupons = async (memberId) => {
  const [coupons] = await pool.query(
    `SELECT c.*
     FROM member_coupons mc
     JOIN coupons c ON mc.coupon_id = c.id
     WHERE mc.member_id = ? AND c.expiry_date > NOW()`,
    [memberId]
  );
```

```
  return coupons;
};
```

五、使用說明

1. 前端開發：
    - 會員中心功能：在 `museum-frontend/app/member/center/features/` 下開發
    - 電商頁面功能：在 `museum-frontend/app/shop/` 下開發
    - 使用 `useAuth` hook 獲取會員狀態
    - 實現對應的 UI 組件
    - 調用後端 API
2. 後端開發：
    - 在 `museum-backend/src/routes/` 下添加新的路由
    - 在 `museum-backend/src/services/` 下添加新的服務
    - 在 `museum-backend/src/config/` 下添加新的資料表
3. 資料庫：
    - 執行 SQL 腳本創建必要的資料表
    - 確保外鍵關係正確
4. 測試：
    - 前端：檢查 UI 顯示和交互
    - 後端：測試 API 端點
    - 資料庫：驗證數據存儲

六、注意事項

1. 會員狀態檢查：
    - 所有需要會員狀態的 API 都要使用 `authenticateToken` 中間件
    - 前端要檢查 `isLoggedIn` 狀態
    - 未登入時可以：
        - 跳轉到登入頁面
        - 顯示登入提示
        - 禁用某些功能
2. API 請求：
    - 所有 API 請求都要帶上 token
    - 使用 `member.id` 作為會員識別
    - 錯誤處理要統一格式
3. 資料預填：
    - 使用 `member` 中的資料預填表單

- 例如：地址、電話等

4. 路由保護：

    - 購物車頁面
    - 結帳頁面
    - 訂單頁面
      這些頁面都需要登入才能訪問