**淡江大學**

**資料庫期末專題報告**

**LOLM 手遊攻略網**

**組員：**

**406411057 資工四C 陳泉豪**

**406410844 資工四C 馮聖淳**

**406410786 資工四C 林清峰**

**406410513 資工四C 王泓壬**

**民國110年05月08日**

**目錄**

1. 前言
2. 文獻探討
3. 資料庫規劃
   1. 實體關係圖
   2. 關聯模式
   3. 欄位定義
4. 專題資料庫系統功能
5. 結論與心得感想
6. 參考文獻及程式碼
7. **前言**

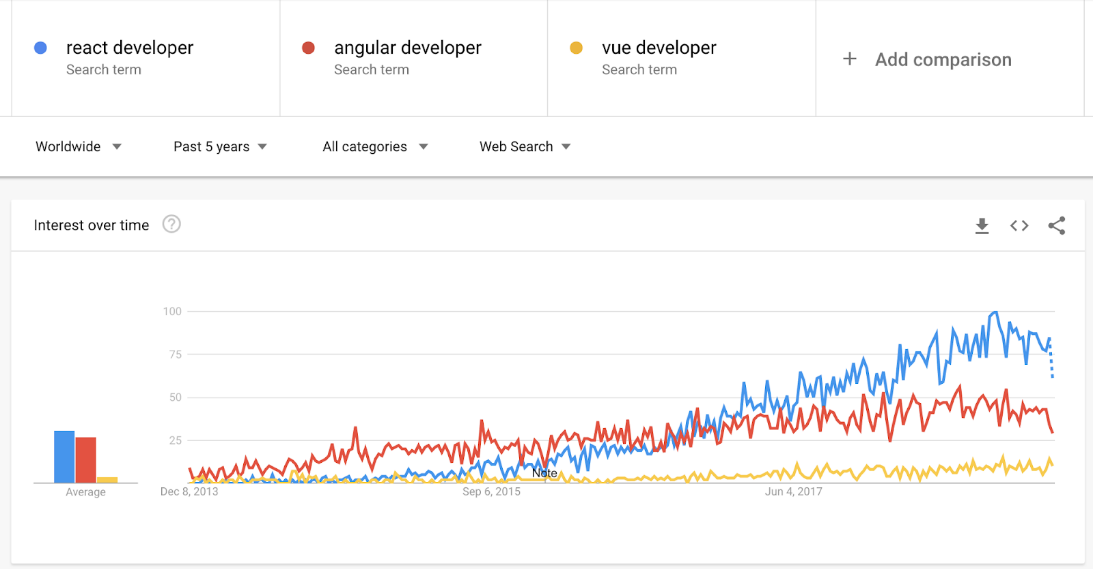
在人手一機的時代，不再是只能透過坐在電腦桌前享受線上遊戲的快感，消除一整天的壓力，手遊的崛起漸漸取代傳統線上遊戲，在通勤的路上或是下課休息時間，都能享受手遊的樂趣。而我們這次的專題則是以我們平常就有遊玩的一款手遊，加以分析歸類以及建資料庫，最後決定以做成一個手遊攻略網，以提供玩家們攻略技巧以及遊玩上的幫助為目標。

1. **文獻探討**

前端使用React / 資料庫使用MySQL / 後端使用Node.js

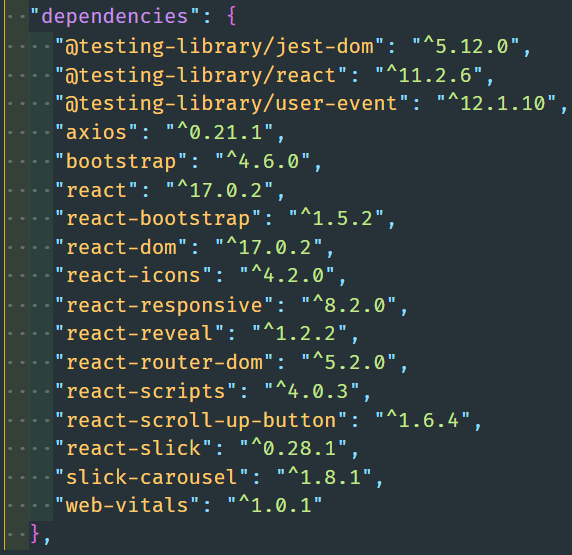
我們這次的報告是選用REACT當作我們的前端開發，React 是 Facebook 在 2013 年 3 月開源的 open source，是現在最熱門主流的前端 JavaScript 函式庫 (library) 之一，專門用來寫 UI (User Interfaces) 用的。

由網路上引用的下圖可知，react的使用次數上升的速度非常快。

****

React負責解決其它javascript框架所面對的一大常見難題，即對大規模數據集的處理。能夠使用虛擬DOM並在發生變更時利用補丁安裝機制只對DOM中的dirty部分進行重新渲染，React得以實現遠超其它框架的速度表現。

我們的前端引用了以下dependencies

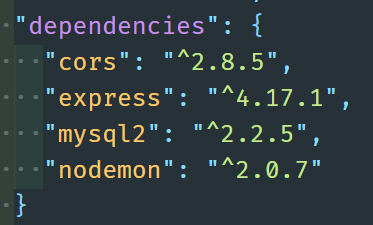


* 運用axios向後端請求資料，並將其資料抓取至前端使用
* 運用react-bootstrap美化網頁排版
* 運用react-icons顯示美化更多Icons
* 運用react-router-dom建立路由
* 運用react-responsive來監聽用戶是否由Computer轉成Mobile裝置讀取網頁
* 運用react-reveal來製作相關的動畫特效顯示
* 引用react-scroll-up-button來顯示置頂按鈕功能
* 運用react-slick && slick-carousel幫助顯示英雄細節的特效瀏覽

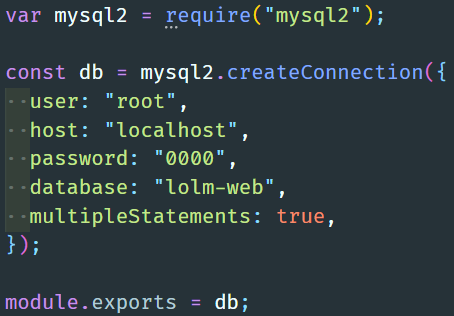
我們後端運用**node.js**

用Express當作我們後端的框架

後端引用以下dependencies：

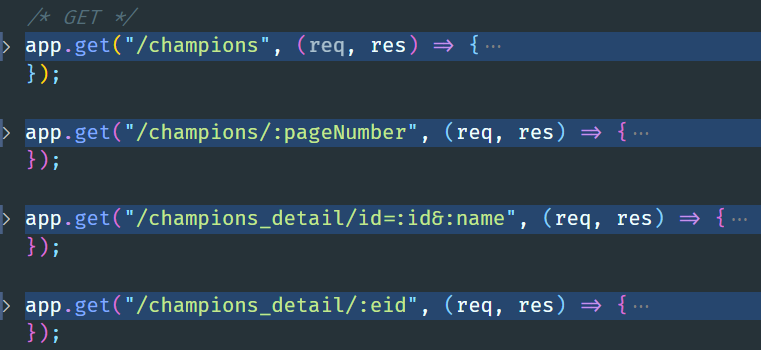


* 運用cors(跨來源資源共用)建立跨來源HTTP請求
* 運用mysql2訪問後端資料庫



* 運用nodemon自動重啟後端，幫助後端更方便開發

建立API



**/champions/**

* 抓取全部英雄(champions)資料。

**/champions/:pageNumber**

* 抓取該筆數(pageNumber)英雄資料。

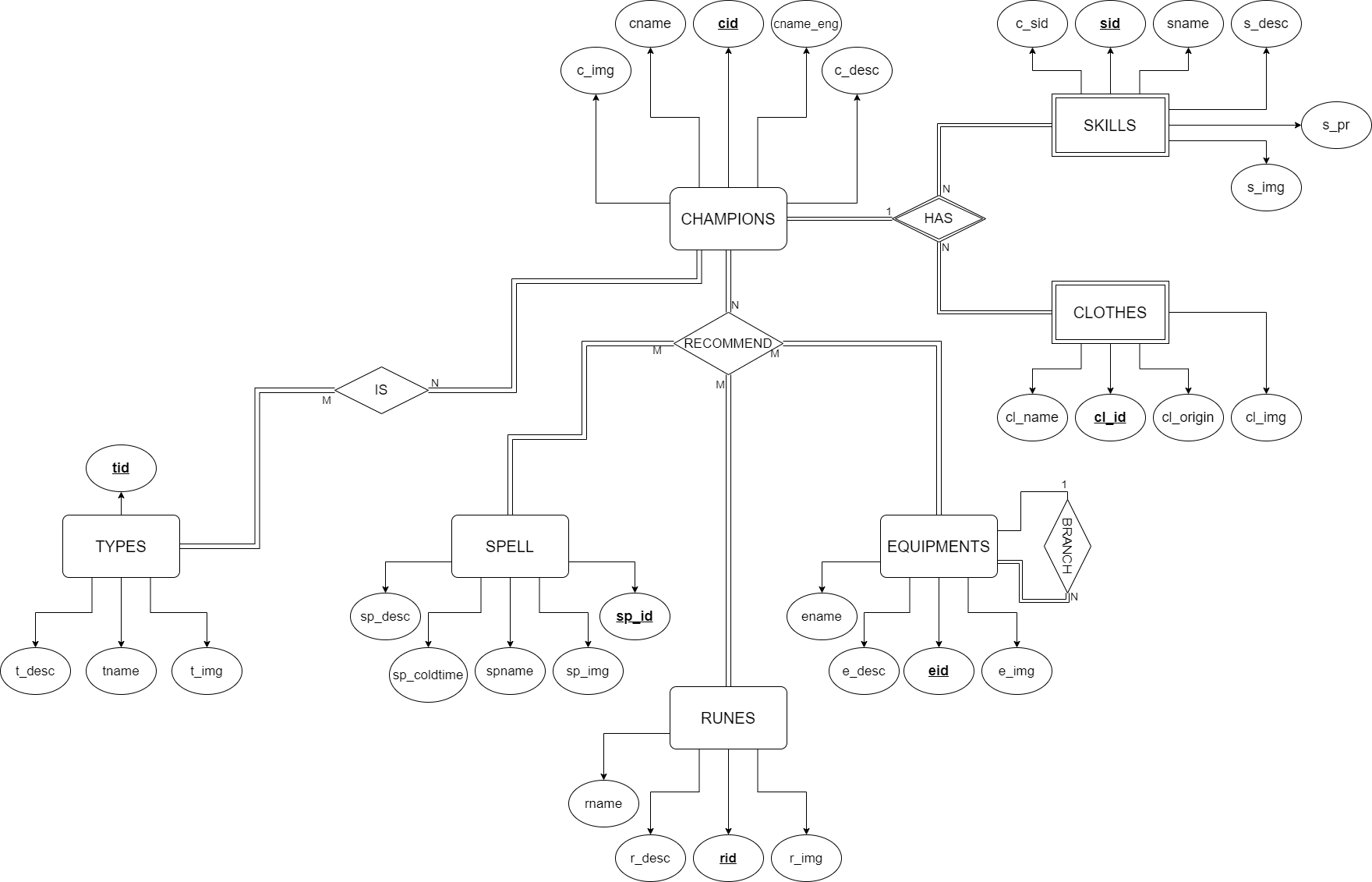
**/champions\_detail/id=:id&:name**

* 利用cid抓取該英雄的其他詳細資料，例如：Type、Skill、Rune、Clothes、Spell。

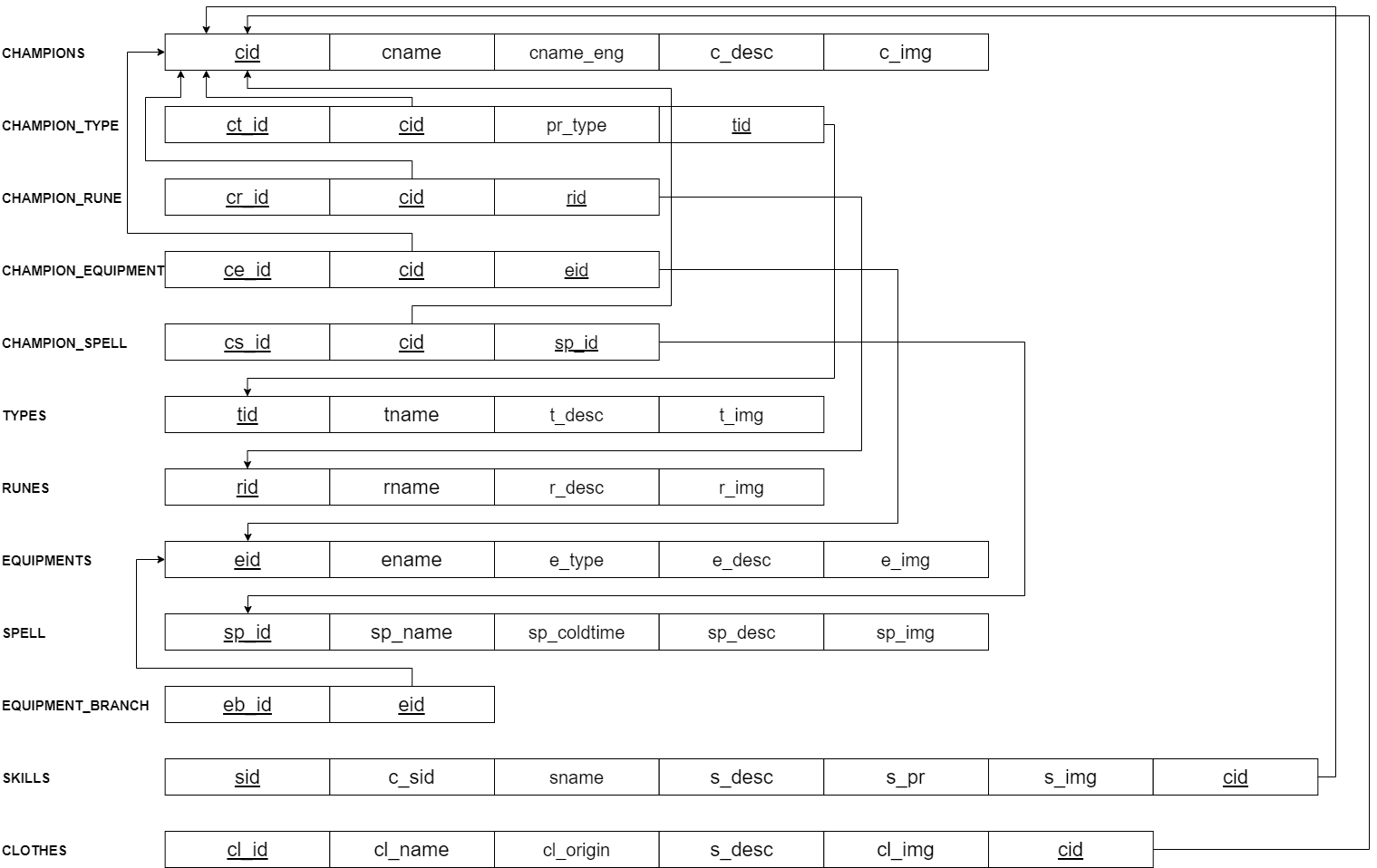
**/champions\_detail/:eid**

* 抓取該裝備的合成裝備詳細資料。

1. **資料庫規劃**
   1. 實體關係圖



* 1. 關聯模式



* 1. 欄位定義

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table** | **欄位名稱** | **欄位\_中文名稱** | **PK** | **資料型態** | **資料長度** | **Nullable** |
| CHAMPIONS | cid | 英雄ID | Y | INT |  | N |
| CHAMPIONS | cname | 英雄名稱 |  | VARCHAR | 45 | N |
| CHAMPIONS | cname\_eng | 英雄名稱(英文名) |  | VARCHAR | 45 | N |
| CHAMPIONS | c\_desc | 英雄\_故事描述 |  | VARCHAR | 1024 | Y |
| CHAMPIONS | c\_img | 英雄照片連結 |  | VARCHAR | 256 | N |
| CHAMPION\_TYPE | ct\_id | 英雄 角色定位ID | Y | INT |  | N |
| CHAMPION\_TYPE | cid | 英雄ID |  | INT |  | N |
| CHAMPION\_TYPE | tid | 角色定位ID |  | INT |  | N |
| CHAMPION\_TYPE | pr\_type | 角色定位優先度 |  | INT |  | Y |
| CHAMPION\_RUNE | cr\_id | 英雄 推薦符文ID | Y | INT |  | N |
| CHAMPION\_RUNE | cid | 英雄ID |  | INT |  | N |
| CHAMPION\_RUNE | rid | 符文ID |  | INT |  | N |
| CHAMPION\_EQUIPMENT | ce\_id | 英雄 推薦裝備ID | Y | INT |  | N |
| CHAMPION\_EQUIPMENT | cid | 英雄ID |  | INT |  | N |
| CHAMPION\_EQUIPMENT | eid | 裝備ID |  | INT |  | N |
| CHAMPION\_SPELL | cs\_id | 英雄 推薦召喚技能ID | Y | INT |  | N |
| CHAMPION\_SPELL | cid | 英雄ID |  | INT |  | N |
| CHAMPION\_SPELL | sp\_id | 召換師技能ID |  | INT |  | N |
| TYPES | tid | 角色定位ID | Y | INT |  | N |
| TYPES | tname | 定位名稱 |  | VARCHAR | 45 | N |
| TYPES | t\_desc | 定位描述 |  | VARCHAR | 256 | Y |
| TYPES | t\_img | 定位照片連結 |  | VARCHAR | 256 | N |
| SKILLS | sid | 所有技能總ID | Y | INT |  | N |
| SKILLS | cid | 英雄ID |  | INT |  | N |
| SKILLS | c\_sid | 英雄\_技能ID |  | INT |  | N |
| SKILLS | sname | 技能名稱 |  | VARCHAR | 45 | N |
| SKILLS | s\_desc | 技能描述 |  | VARCHAR | 1024 | Y |
| SKILLS | s\_pr | 技能點選優先度 |  | INT |  | Y |
| SKILLS | s\_img | 技能照片連結 |  | VARCHAR | 256 | N |
| RUNES | rid | 符文ID | Y | INT |  | N |
| RUNES | rname | 符文名稱 |  | VARCHAR | 45 | N |
| RUNES | r\_type | 符文種類 |  | INT |  | N |
| RUNES | r\_desc | 符文描述 |  | VARCHAR | 1024 | Y |
| RUNES | r\_img | 符文照片連結 |  | VARCHAR | 256 | N |
| CLOTHES | cl\_id | 所有造型總ID | Y | INT |  | N |
| CLOTHES | cid | 英雄\_造型ID |  | INT |  | N |
| CLOTHES | cl\_name | 造型名稱 |  | VARCHAR | 45 | N |
| CLOTHES | cl\_origin | 原造型 |  | INT |  | Y |
| CLOTHES | cl\_img | 造型照片連結 |  | VARCHAR | 256 | N |
| EQUIPMENTS | eid | 裝備ID | Y | INT |  | N |
| EQUIPMENTS | ename | 裝備名稱 |  | VARCHAR | 45 | N |
| EQUIPMENTS | e\_type | 裝備種類ID |  | INT |  | N |
| EQUIPMENTS | e\_desc | 裝備描述 |  | VARCHAR | 1024 | Y |
| EQUIPMENTS | e\_img | 裝備照片連結 |  | VARCHAR | 256 | N |
| EQUIPMENTS\_TYPE | et\_id | 裝備種類ID | Y | INT |  | N |
| EQUIPMENTS\_TYPE | et\_name | 裝備種類名稱 |  | VARCHAR | 45 | N |
| EQUIPMENTS\_BRANCH | eb\_id | 裝備組合ID | Y | INT |  | N |
| EQUIPMENTS\_BRANCH | bid | 裝備目標ID |  | INT |  | N |
| EQUIPMENTS\_BRANCH | eid | 裝備ID |  | INT |  | N |
| SPELL | sp\_id | 召換師技能ID | Y | INT |  | N |
| SPELL | sp\_name | 召換師技能名稱 |  | VARCHAR | 45 | N |
| SPELL | sp\_coldtime | 召換師技能冷卻時間 |  | VARCHAR | 45 | Y |
| SPELL | sp\_desc | 召換師技能描述 |  | VARCHAR | 1024 | Y |
| SPELL | sp\_img | 召換師技能照片連結 |  | VARCHAR | 256 | N |

1. 專題資料庫系統功能
2. **系統主要功能**
3. 在Champion頁面中，可以查看全部英雄。當用戶一開始進來時會先向後端請求前8個英雄的資料，並監聽最下方的區塊，是否顯示來判斷是否繼續撈資料。當滾動到該區塊顯示時，才會再向後端索取下8個英雄的資料，並將原本監聽的區塊取消監聽，並聽重新渲染後的頁面產生的區塊，繼續上述的步驟，直到後端沒有撈到data為止。
4. 當點選該Champion的卡片後，將會把剛剛抓到的data利用該英雄的id傳入到該指定Champion的Details頁面，並同時透過該id向後端請求更多該英雄的其他資訊，並傳回前端使用。
5. 在ChampionDetail的頁面中，所有的detail component都是使用當頁面渲染時從後端抓來的資料加以呈現美化。
6. **其他Component更多細節功能** 🡺

* **<NavBar />**：主選單/導覽列
* **<ScrollToTop />**：當頁面刷新會將頁面跳轉至最上方
* **<ScrollUpButton />**：當頁面到達滾動指定高度後，將顯示跳轉至最上方的按鈕
* **<MainContent />**：父類別，其他Component皆會是其子類別
  + - **<Home />**：首頁
    - **<About />**：關於我們
    - **<Footer />**：頁尾
* **<PageNotFound />**：當網址並非route中設置的，將會顯示該畫面

**最主要的兩個Component：**

* + **<ChampionsContainer />**：所有英雄
  + **<ChampionsDetailContainer />：**英雄的詳細資料

**/route/route.js 路由設置**：

利用‘react-router-dom’中的BrowserRouter到routes.js抓取route路徑，控制每個路由，實現到達正確的Component(其中exact（bool）：為true時，則要求路徑與location.pathname必須完全一致才會顯示)。

const routes = [

  {

    path: '/',

    component: Home,

    exact: true,

    breadcrumbName: 'Home',

  },

  {

    path: '/about',

    component: About,

    breadcrumbName: 'About',

  },

  {

    path: '/champions',

    component: ChampionsContainer,

    breadcrumbName: 'ChampionList',

  },

  {

    path: '/champions\_detail/id=:id&:name',

    component: ChampionsDetailContainer,

    breadcrumbName: 'ChampionsDetail',

  },

  {

    path: '\*',

    component: PageNotFound,

  },

];

1. **各Component中運用到的功能**

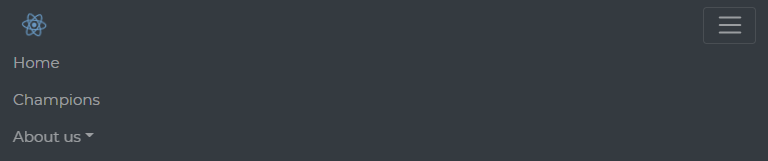
* **NavBar.js(導覽列)**

畫面：

網頁顯示



Responsive Web

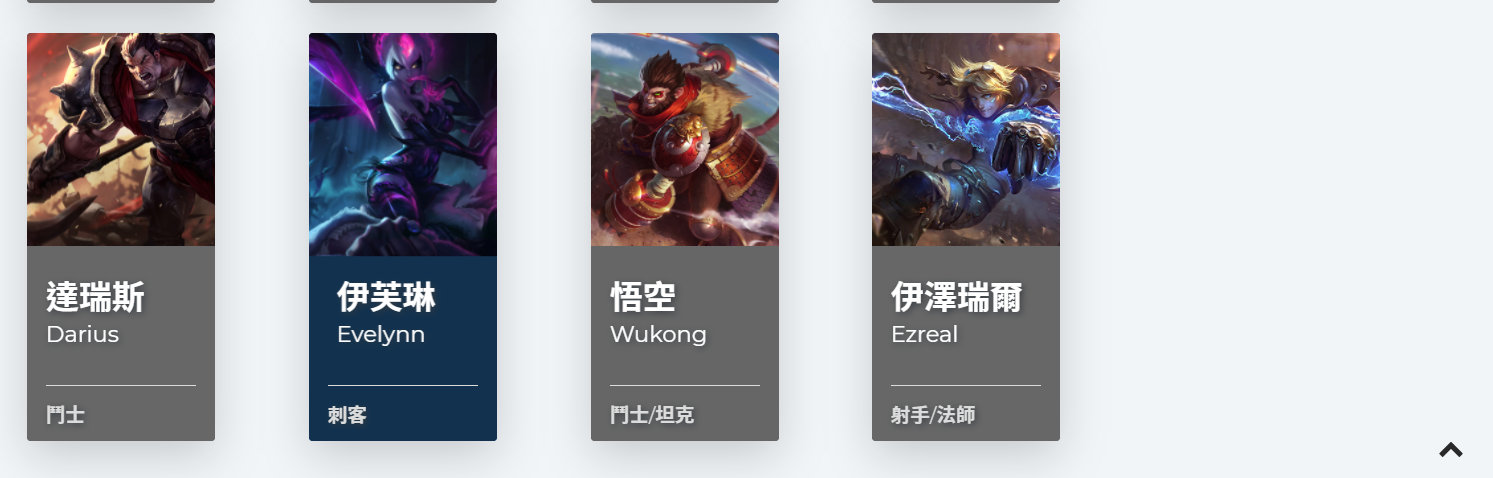


說明🡺

當點選導覽列中的選項後，將利用useHistory 的 push function => 將route指向該指定的頁面，並跳轉至該畫面(如圖：跳轉至首頁)。

* **ScrollUpButton.js(置頂按鈕)**

畫面：(右下角)



Configure：

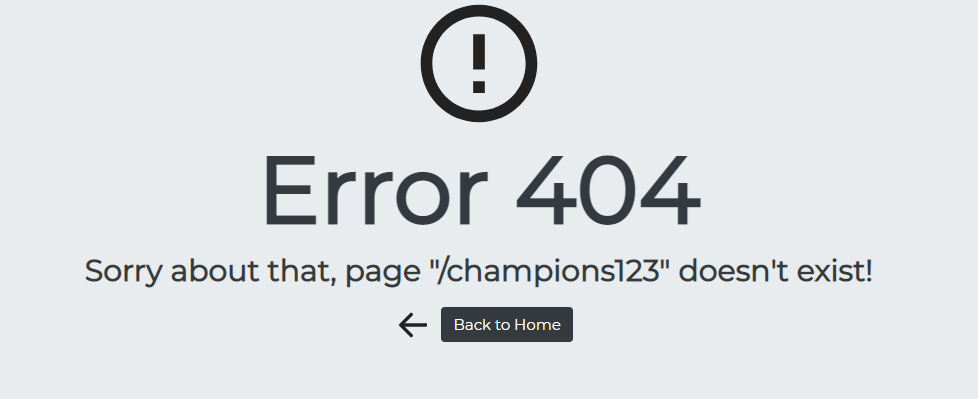
*ShowAtPosition*🡺當頁面到達這個位置，就會顯示回到最上方的按鍵

*EasingType*🡺按鈕顯示動畫

*AnimationDuration*🡺動畫時間

* **PageNotFound.js(無畫面)**

畫面：

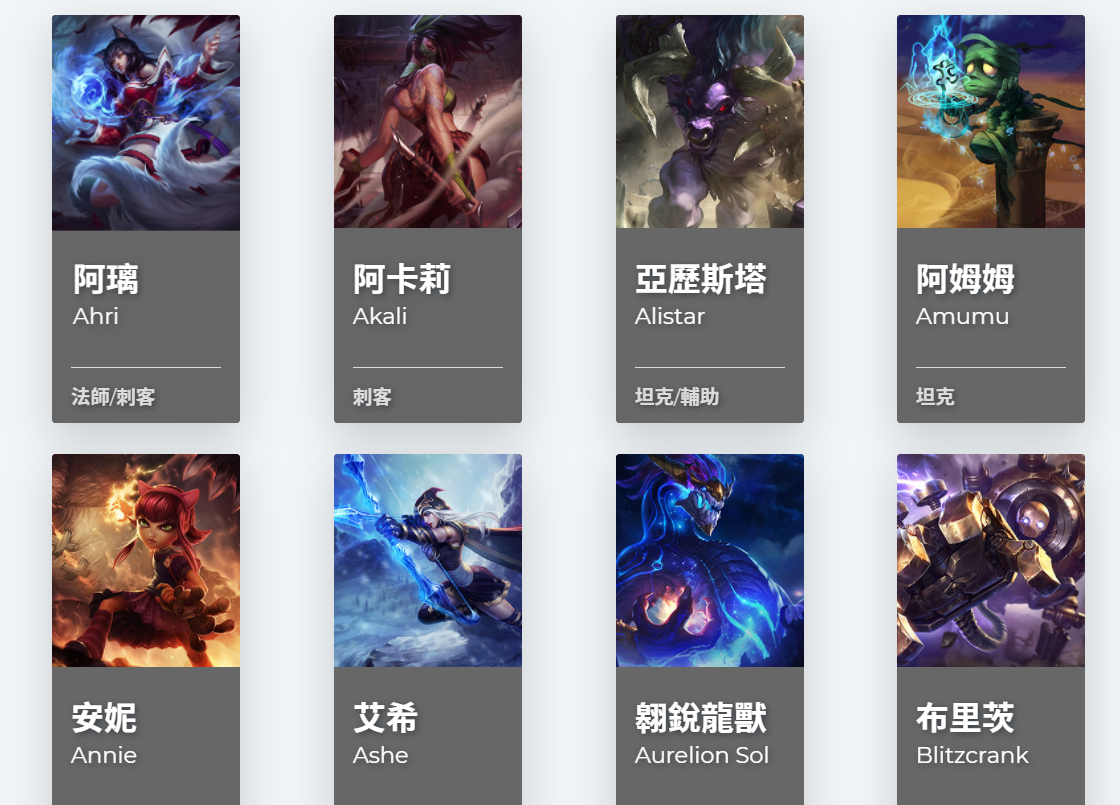


說明🡺

運用到useLocation Hook🡺會回傳一個 Object 的物件, 裡面包含 pathname, search, hash, key 與state等資訊，其中我們運用到pathname將其錯誤的網址顯示出來。

* **ChampionContainer(英雄頁面)**

畫面：



說明🡺

運用useState hook 來設定變數，第一個參數為變數名稱，後方則是改變該參數的function，useState中則設定該參數的性質。

1. championList🡺用來存放後端抓來英雄的資料
2. loading🡺畫面是否滾到該div出現
3. pageNumber🡺控制現在已經抓幾批英雄
4. hasMore🡺資料庫是否還有更多英雄

當該畫面渲染時，則開始做以下function，其中用useEffect hook當畫面一開始將標籤頁的標頭改成指定格式‘Champions List’。之後運用IntersectionObserver來監聽指定的div是否出現在畫面上。

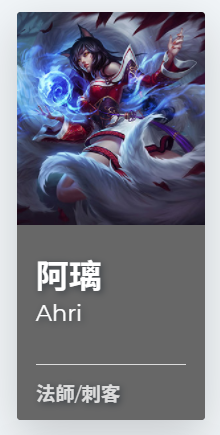
observer.observe(lastElement.current);

observer.unobserve(lastElement.current);

其中運用useRef設定current屬性，將其綁指定的div上。當該div出現時loading為True，並執行loadMore()，將pageNumber+1的英雄，再呼叫**Function fetchChampionList(pageNumber)使用**Axios向後端請求資料下一筆的英雄資料。

* **Champion.js(英雄卡)**

畫面：

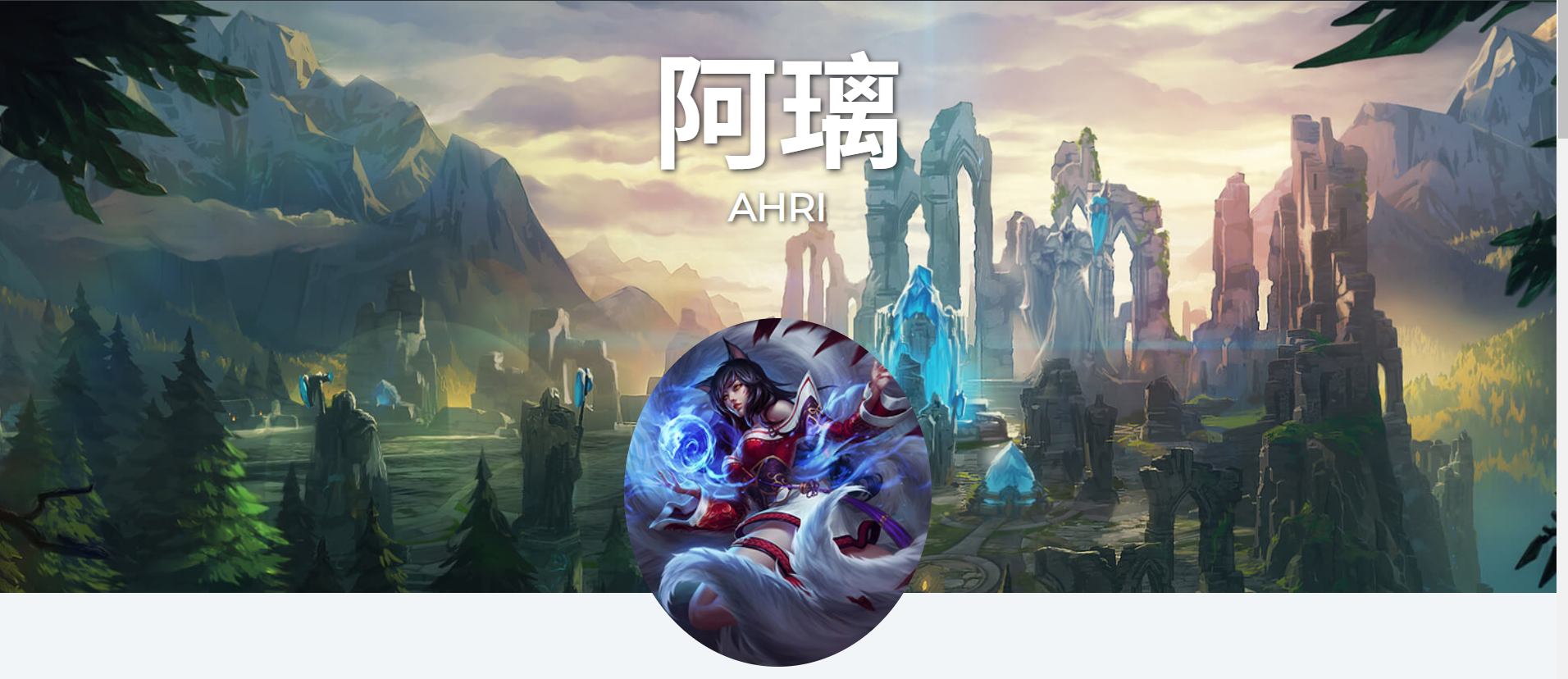


說明🡺

點擊該英雄卡，就會把該英雄的各個資料，利用useHistoiry push到該路由中。

* **ChampionsDetailContainer.js(英雄細節)**

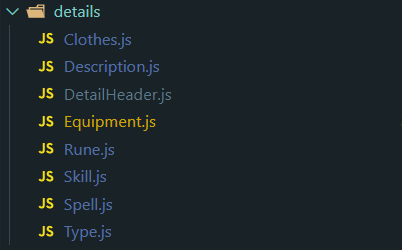
畫面：



說明：

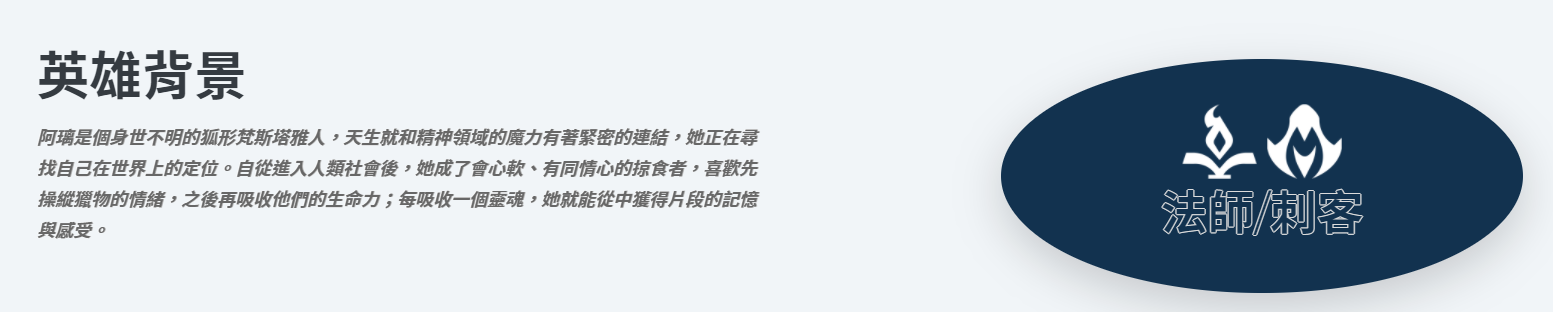
當畫面渲染時，執行fetchChampionDetail()，並將後端抓到的資料分別放入champDetails中，並將後端抓取到的值(如：types、skills、runes、spells、equipment、clothes)傳到ChampionDetails裡去渲染畫面。

  <ChampionDetails *details*={champDetails} *info*={championInfo} />



* **Description & Type(英雄、定位)**

畫面：



說明🡺

運用到判斷式，若後端資料庫中的第二類型態為空值得畫，則只顯示一個定位。

如圖：

當指標靠近時，則會顯示該定位的詳細資料內容。

畫面：

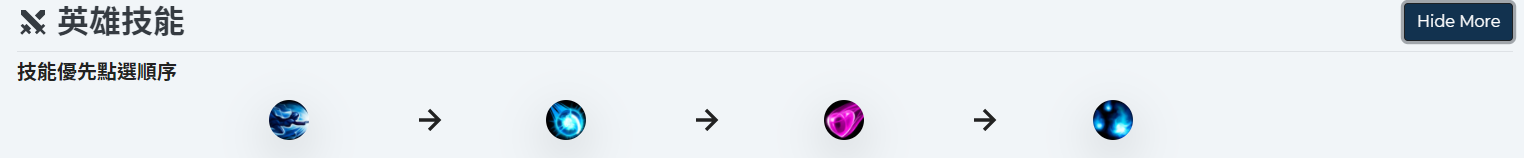
* **Skill(技能)**

畫面：



說明🡺

其中當c\_sid=4或5的時候將顯示『大絕』跟『被動』，否則如c\_sid標示。

點選右上角的按鈕會顯示推薦技能的點選順序，畫面如：

* **Rune & Spell(符文，召喚師技能)**

畫面：



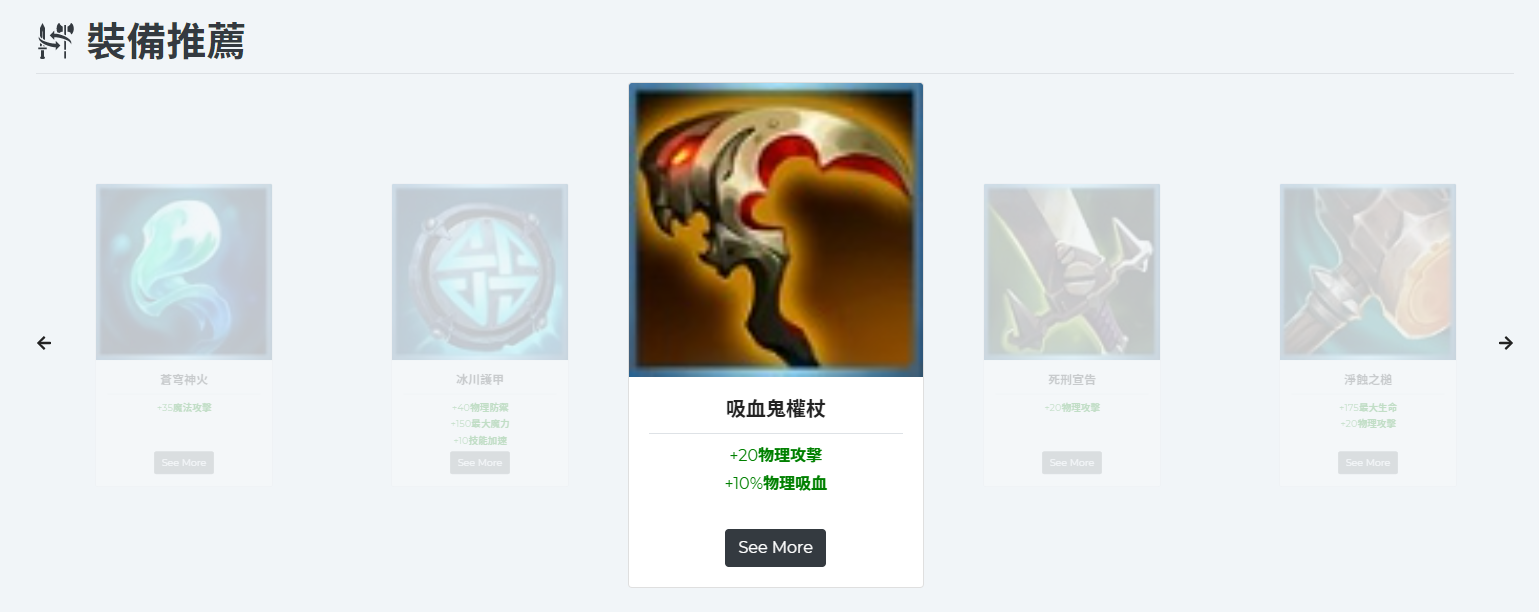


說明🡺

該兩個Component利用react-bootstrap中的overlay popover，製造出當指標靠近Image時顯示更多的詳細資料。

* **Equipment(裝備)**

畫面：



說明🡺

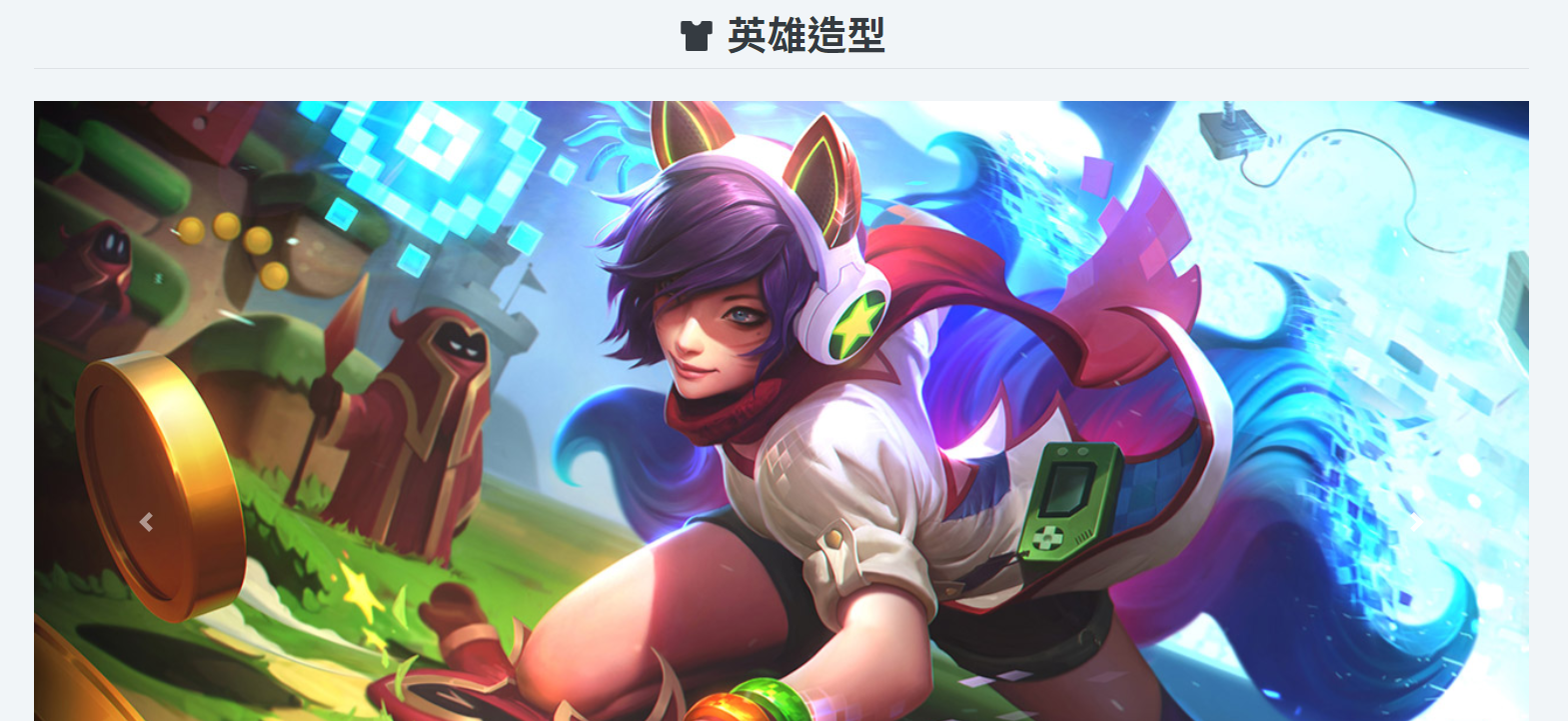
當點Button後，顯示該裝備更多細節及合成細節，再點擊一次則收回。

畫面如下：



* **Clothes(造型)**

畫面：



說明🡺

利用react-bootstrap的Carousel讓造型的照片自動輪放，也可手動控制顯示順序。

1. **結論與心得感想**

最後我們做出來的網頁算是將前期討論所訂定出來的大致方向以及目標呈現出來了。 總結來說我們認為我們做的網頁並沒有完全達到攻略網的目的。比較類似遊戲官網再介紹這款遊戲,而不是如何精通這款遊戲。在一些細節上的部分像是資料庫的數據內容不夠充足以及不完整，在網頁設計上我們認為能夠再做多一點的變化以及網頁的美觀設計。資料庫的部分我們之後會陸續補上去。美觀設計部份我們的想法是多用點遊戲的相關圖片增加整體視覺上的豐富程度。我們認為如果要達到攻略網的目的，必須更深入了解這款遊戲的遊玩風格以及參考國外攻略網的內容，花更多時間在這款遊戲，最好牌位能夠打到鑽石以上。

1. **參考文獻及程式碼**

<ScrollToTopButton>🡺 引用至https://www.cakeresume.com/portfolios/f22bc8

React介紹🡺 https://www.fooish.com/reactjs/

Dependencies：

<https://www.npmjs.com/package/react-scroll-up-button>

<https://www.npmjs.com/package/react-responsive>

<https://www.npmjs.com/package/react-bootstrap>

<https://www.npmjs.com/package/axios>

<https://www.npmjs.com/package/react-reveal>

<https://www.npmjs.com/package/react-icons>

<https://www.npmjs.com/package/react-slick>

<https://www.npmjs.com/package/cors>

<https://www.npmjs.com/package/nodemon>

Github程式碼專案🡺

<https://github.com/qqharry21/LOLM_GameStrategy_Project>