

國立嘉義大學資訊工程學系
計算機專題報告

Department of Computer Science and
Information Engineering
National Chiayi University
Computer Project Report

利用 Bootstrap 建置友善的 UI/UX 框架系統

指導教授： 李龍盛 老師

年度： 一百零七學年度

組別： 347-108-21

學生： 1042992 黃祺淵

1043029 賴彥儒

中華民國 一百零八 年 一 月

利用 Bootstrap 建置友善的 UI/UX 框架系統

指導教授：李龍盛 老師 學生：黃祺淵、賴彥儒
國立嘉義大學資訊工程學系

摘要

在這個人手一機的時代，以行動裝置為導向的網站介面日益增多，但是我們還是能夠看到有些網站並不支援 RWD，造成大多數行動裝置使用者的體驗不如預期，而開發者並不是不願意將原本的網站升級成響應式網站，而是修改起來過於麻煩，若是太舊的網站要升級基本上等於要重新做，需要耗費大量的時間、人力與金錢。於是我們想製作一個方便開發者修改他們網站的 UI/UX 系統，此系統會與 Bootstrap 框架結合，讓升級的成本大幅降低。

Bootstrap 是用於網站與網路應用程式開發的前端框架，提供字體排印、表單、按鈕、導航及各種元件。因為代碼庫相同，所以減少了維護上的成本，但也因此造成了開發者無法更加自由的去做更改，因為若是使用不同的代碼庫，會容易造成不一致性的問題。因此我們開發一個 Bootstrap 的修改系統，讓開發者能夠方便的將欲處理程式碼映射到源程式碼，進而使得開發者有更自由的空間，不必受限於源代碼的框架，同時也解決了不一致性的問題、以及降低維護上的成本。

關鍵字：Bootstrap、UI/UX、Source Map

目錄

摘要	I
目錄	II
圖目錄	IV
第一章、緒論	1
1.1 製作動機	1
1.2 製作目的	2
第二章、專題架構	3
2.1 系統架構	3
2.2 系統流程	5
2.3 相關技術與工具	8
2.3.1 PHP	8
2.3.2 Bootstrap	8

2.3.3 Less	8
2.3.4 CSS.....	9
2.3.4 Source Map	10
2.3.5 JavaScript.....	11
2.3.6 Node.js	11
2.3.7 Ajax	12
第三章、專題成果.....	13
3.1 系統特色.....	13
3.2 專題資料 Hierachy	16
3.3 修改介面.....	17
第四章、結果與討論.....	20
第五章、結論.....	21
參考文獻.....	22

圖目錄

圖 2.1 系統架構.....	5
圖 2.2 PHP 端系統流程.....	6
圖 2.3 Node.js 端系統流程	7
圖 3.1 網站首頁	14
圖 3.2 網站首頁(手機端).....	14
圖 3.3 前端頁面 Hierachy	16
圖 3.4 後端資料 Hierachy	16
圖 3.5 修改頁面	17
圖 3.6 修改頁面(手機端).....	18
圖 3.7 取得 CSS 之 URL.....	18

第一章、緒論

1.1 製作動機

在大三上時幫忙學校寫網頁，第一次接觸到了 Bootstrap[5]。一開始，學校的首頁不支援 RWD，而學校要求我們將學校的首頁升級成支援 RWD 的介面，並且要符合無障礙網頁的規範。一開始覺得窒礙難行，後來查了很多資料以及詢問指導教授的建議，我們決定使用 Bootstrap，來作為我們開發的框架。一開始我們覺得方便好用，因為框架都幫你設定好了，而且它就是一個行動裝置導向的框架，但當遇到需要某部分需要修改時，需要花很多時間去找出問題點，因為 Bootstrap 的 CSS 多採複用且混合的寫法，而且一般的網頁引用的都是經壓縮過後的 CSS 檔和 JS 檔，所以要找出問題點並修改是非常麻煩的一件事，再加上雖然在 Bootstrap 2.0 版之後有提供讓開發者選取需要的套件功能，但其實客製化的程度還是不夠。根據資料顯示，有 75% 的前端工程師曾經想修改原生的 Bootstrap 程式碼，這是為甚麼呢？因為一個 Bootstrap 的框架實在太龐大，而一個正常的網站只會運用到大約 5% 的內容，這就表示有 95% 是被浪費掉的，而這 95% 不僅無用，還會造成網頁在載入時的負擔、拖垮整個網頁載入的速度，身為一個好的網頁設計師，必須將可能造成使用者不好的體驗排除，因此才有那麼多的前端工程師曾經想要修改原生的 Bootstrap 程式碼[6]。

1.2 製作目的

根據資料統計，目前世界上大約有 18.7 億的網頁使用 Bootstrap 的框架，而 Bootstrap 的開發取向就是以行動裝置導向進行開發。而截至 2017 年底，行動上網用戶已經佔了所有網際網路流量的 52%，從這項數據來看我們可以發現，瀏覽網頁方式的主流已經從一般電腦轉至手機、平板等行動裝置[4]，也就代表著網頁前端工程師必須將重心放在行動裝置用戶的體驗上面[7]。但是 Bootstrap 實在太過龐大，一般開發者無法方便、輕鬆地去修改 Bootstrap 原生的程式碼，因此我們打造出一個專門提供開發者方便進行修改的系統，此系統以 Bootstrap 為底，讓使用者可以做修改、增添、刪除程式碼等等相關編輯動作，並將使用者輸入的程式碼映射至原程式碼中。最後提供一組 URL，讓使用者可以直接 import 至他的網頁程式碼中，進而取代原本的 Bootstrap CSS 檔。

第二章、 專題架構

2.1 系統架構

本節將介紹本專題的系統架構，其架構如圖 2.1 所示。

我們將系統架構主要分為兩大功能區的分類，第一部分為修改，第二部分為轉換。第一部分為修改部分，主要由 PHP 撰寫，此部分負責輔助使用者處理程式碼，讓使用者可直接在網站[1] 上達到新增、修改、刪除程式碼的效果，可細分功能如下：

- (1)會員專案與模板管理：會員能根據自己的專案、以及其專案之下的模板來選擇程式碼
- (2)編輯器視覺化：在網站中實作出一個視覺化的編輯器效果，讓程式碼會根據關鍵字變化以及出現輸入建議並顯示行數方便使用者進行修改
- (3)Ajax 技術做及時修改：利用 Ajax 技術將使用者的程式碼及時呼叫並修改，改進使用者體驗。

第二部份為轉換部分：

後端部分由 Node.js 架構而成，我們的主要目標是會員系統以及對於 LESS[1] 轉換 CSS。

首先從 MySQL 中抓取會員的會員名稱、專案名稱、以及專案底下各個模板名稱，藉由這些資料在我們伺服器端創立專屬於每個會員的目錄，由於我們主要針對的使用者為前端工程師，通常前端工程師也都會接許多不同的工作，所以目錄底下會有著各個不同的專案，每個專案底下也會有不同模板的目錄，在每個目錄底下我們先將幫助我們轉換的檔案

放入，未來我們轉換出的 CSS 也放在此目錄底下[3]。

建立好會員目錄後，接著由 MySQL 抓取會員所有修改的 LESS 程式碼，目前我們提供了五種不同區塊的 LESS 提供給使用者修改，所以抓取時會同時將此模板底下的五個修改或未修改的 LESS 都抓去到後端，將同一個專案中同一個模板的 LESS 檔全部 Decode 後放在此模板目錄之下，並且將全部轉換並且合併成單一一個 CSS，我們將此 CSS 名稱取為 custom-Bootstrap.css。

這裡我們轉換是透過 LESS 官方網站所提供的方法，此方法只適用於 Node.js 的環境架構下才能使用，接著透過指令碼將原本的 CSS 壓縮成一個更小的 CSS 檔，並且產出此 CSS 的 URL，再透過 Node.js 回傳給資料庫，以便於使用者使用。

後端隨時監控資料庫的檔案有無被修改，若有被修改則會馬上做出反應及時的更新我們 CSS 檔。因此轉換可以分為主要三個部分：

(1)建立會員目錄：根據使用者創建的專案以及模板做出會員的目錄架構

(2)LESS 轉換為 CSS：實作將 LESS 程式碼轉換為 CSS 的程式碼

(3)產出 URL：產出一個相對應的 URL 供使用者直接 import 到自己的網頁中。

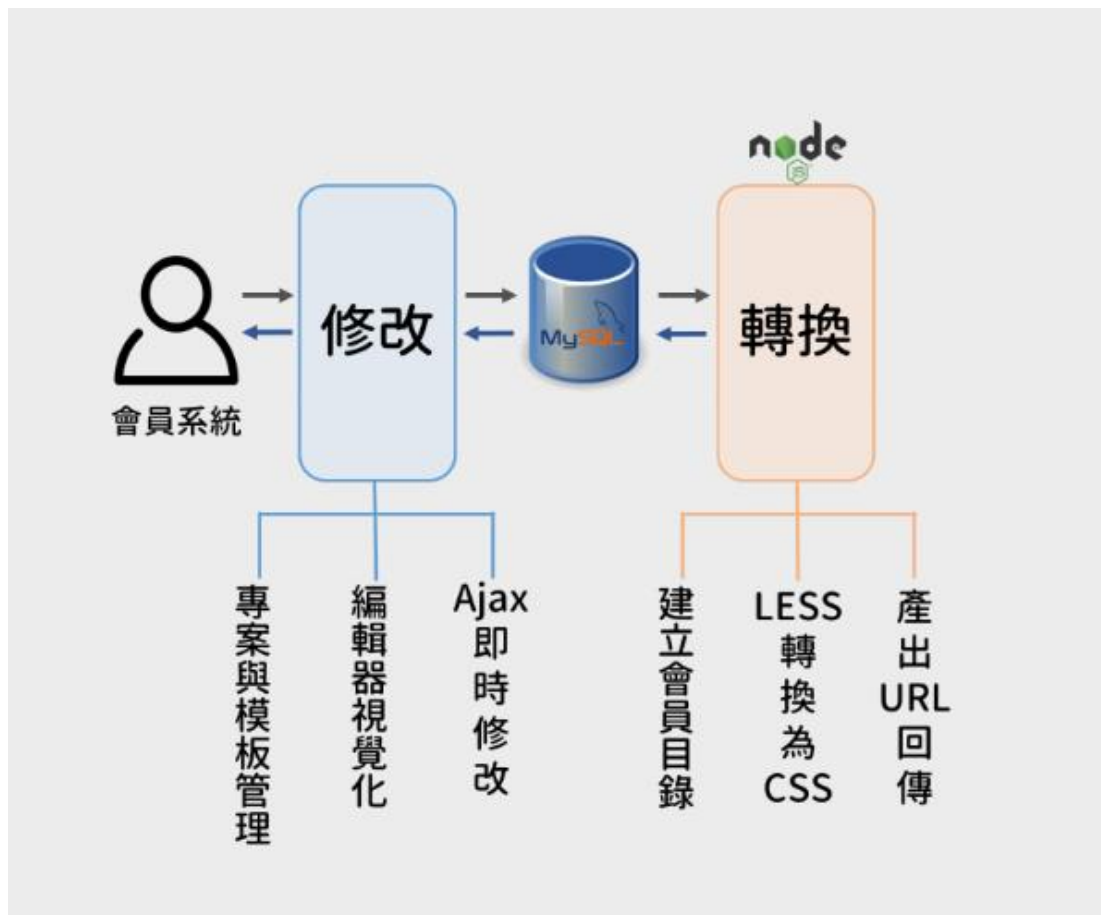


圖 2.1 系統架構

2.2 系統流程

如圖 2.2，一開始使用必須建立一組帳密用以登入我們的會員系統，在使用者登入後，使用者可以選擇新增一專案或者選擇目前現有的專案，選擇專案完畢後會進入到選擇模板部分，在這裡所有的模板都是屬於剛剛選擇的專案底下的，使用者同樣可以選擇新增一新模板或是選擇現有模板，選擇所想要的模板後就可以進入到我們的修改頁面，在這邊我們有將 Bootstrap 各元件做分類，讓使用者可以直接對目標進行修改。在後端部分會根據使用者所選擇的專案、模板、元件等，透過 Ajax 技術將程式碼展現，若是新生成的模板，後端將會導入新的 Bootstrap 程式碼進入到模板供使用者使用，另外為了將使用者的程式碼存入

資料庫以及避免 XSS 以及 SQL Injection 攻擊，在傳遞資料前會先 ENCODE 程式碼，等到要呈現或使用時再將程式碼 DECODE。

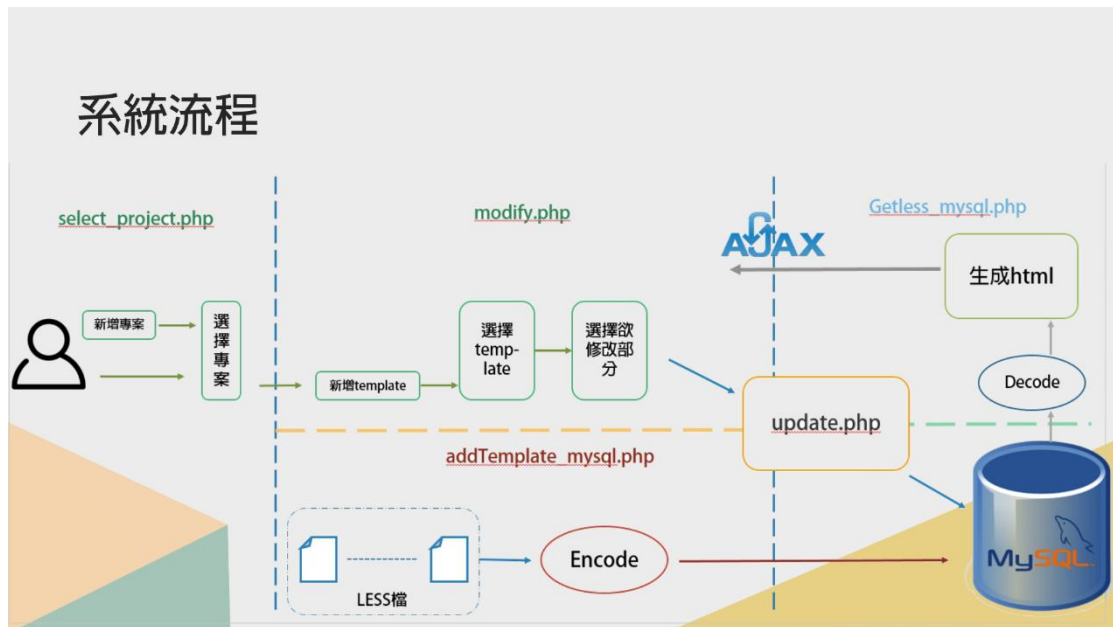


圖 2.2 php 端系統流程

另外在 Node.js 端，如圖 2.3，主要的部份為將 LESS 程式碼轉為 CSS 程式碼，會根據使用者的專案、模板生成資料階層，最後將 CSS 程式碼壓縮後生成一組 URL 回傳給使用者。

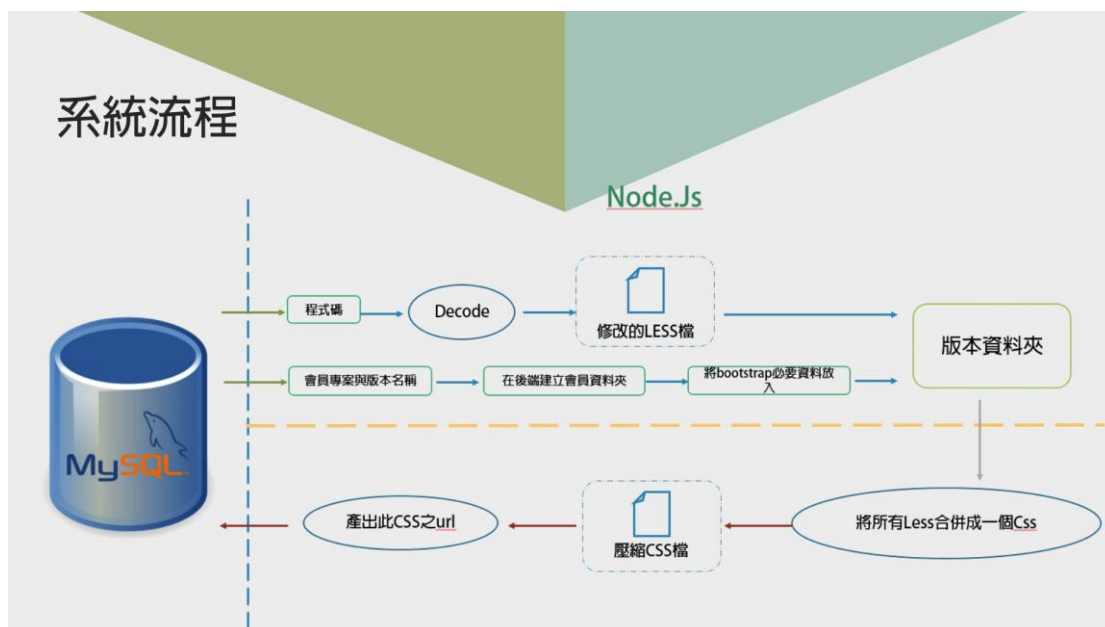


圖 2.3 Node.js 端系統流程

2.3 相關技術與工具

2.3.1 PHP

PHP 是一種開源的通用電腦手稿語言，適用於網路開發並可嵌入 HTML 中使用，主要目標是允許網路開發人員快速編寫動態頁面。PHP 大多在伺服器端執行，透過執行 PHP 的程式碼來產生網頁提供瀏覽器讀取。PHP 可以在許多的不同種的伺服器、作業系統、平台上執行，也可以和許多資料庫系統結合。

2.3.2 Bootstrap

Bootstrap 是一個前端開發的網頁模組，是一個響應式(RWD)、且以行動裝置為優先導向的網頁模組。主要由 HTML、CSS、JAVASCRIPT 等語言組成，因簡潔靈活，且大部分必要功能都有套件，因此能加速開發。但對於需要靈活應用的開發者來說，Bootstrap 的客製化程度還是不夠，因此我們希望打造一個完全可客製，且方便使用者增添或修改 Bootstrap 的系統。

2.3.3 Less

LESS 是一個 CSS 的預處理器，讓 CSS 具有動態性。因為 LESS 能內嵌入瀏覽器中，動態的轉為 CSS。LESS 擁有變數、混和、層次結構、運算式及函數呼叫等概念。擁

有變數的概念就可以允許開發者單獨定一系列通用的樣式，然後在需要的時候去調用。所以在做全局樣式調整的時候，我們可能只需要修改幾行程式碼就可以了。混合可以將一個定義好的 class A 輕鬆的引進另一個 classB 中，從而簡單實現 class B 繼承 class A 中所有的屬性。還可以帶參數地調用，就像使用函數一樣。而層次結構就是方便開發者在撰寫過程中，能供清楚的看出繼承的關係，在實際使用時能大程度的減少程式碼，讓程式碼看起來更清晰。運算 LESS 有提供加法、減法、乘法以及除法的簡單操作，我們可以做屬性值以及顏色的運算，這樣就可以實現屬性質之間的複雜關係。

2.3.4 CSS

CSS 是一種為結構化文件(如 HTML、XML)增添樣式，如字型、間距、顏色等的語言。最主要的目的是將檔案的內容與顯示分隔開來。這樣有以下的好處：

- 檔案的可讀性增強
- 檔案的結構更靈活
- 作者與讀者可以自己決定檔案如何顯示
- 檔案結構簡化

而 Bootstrap 就是將 CSS 都幫開發者寫好了，雖然稱不上說是寫死，但是一般的開發者是不會去更動他的 CSS 檔的，為什麼呢？因為通常引用至網站的幾乎都是 Bootstrap.min.css，也就是經過壓縮過後的 CSS 檔，此種 CSS 檔不是用來給撰寫者來閱

讀的，因為它的可讀性極低，也是我們說的”一行程式碼”。此種 CSS 檔為了加快網頁載入，於是將所有換行，以及非必要的空白通通移除。所以通常開發者不會直接去修改，若是真的要修改的話則會去改 bootstrap.css 檔，然後網頁之後就直接引用這個修改後的 bootstrap.css，可是這樣會造成網頁載入速度變慢，因為 bootstrap.css 裡的換行跟空白降低了讀取的速度，所以有人會把它在自己的主機端架設 Node.js，安裝 Gulp 來做壓縮整合的動作，將之轉成 bootstrap.min.css 後再來做使用，但是這樣對開發者來說太麻煩了，所以我們是希望將壓縮的動作由 Server 端來做執行，開發者只需要專心在開發上就好。

2.3.4 Source Map

Source Map 是除錯時用的，它是用來映射對照"正式執行的程式碼檔案內容"，與"開發時撰寫的程式碼檔案內容"。它不只用於 Javascript，也可以用於 CSS 框架如 SASS/LESS 的除錯。目前這種檔案的規格是由 Google Chrome 開始制定的，大約是在 2013 年開始。Chrome 也是最早能支援 Source Map 的瀏覽器，目前還有另外的瀏覽器也可以支援 - Firefox、Opera、Safari、IE11 等等。

Source Map 會被制定出來，主要要因為 Javascript 程式碼，雖然直接從瀏覽器的原始碼檢視工具可以看得到，但在應用程式正式發行時，為了增加檔案傳輸的效率，進行最小化(Minify)或醜化(Uglify)之類的壓縮，造成無法直接進行原始碼除錯的工作。以下是最常見的情況

2.3.5 JavaScript

JavaScript 是一種高階的程式語言，是一個動態、物件導向的直譯語言，它被世界上的絕大多數網站所使用，也被世界主流瀏覽器支援。雖然 JavaScript 在傳統意義上被視為一種解釋語言，但在最近它已經可以被即時編譯執行，隨著最新的 HTML5 和 CSS3 語言標準的推行它還可以用於遊戲、桌面、行動應用程式等開發，也可以在伺服器端網路環境執行，如 Node.js。如果說 HTML 定義了網頁的內容、CSS 描述了網頁的布局的話，那麼 JavaScript 就是網頁的行為。

2.3.6 Node.js

Node.js 是一個能夠在伺服器端運行 JavaScript 的開放原始碼、跨平台 JavaScript 執行環境。Node.js 由 Node.js 基金會持有和維護，並與 Linux 基金會有合作關係。Node.js 採用 Google 開發的 V8 執行程式碼，使用事件驅動、非阻塞和非同步輸入輸出模型等技術來提高效能，可優化應用程式的傳輸量和規模。這些技術通常用於資料密集的即時應用程式。

Node.js 大部分基本模組都用 JavaScript 語言編寫。在 Node.js 出現之前，JavaScript 通常作為用戶端程式設計語言使用，以 JavaScript 寫出的程式常在用戶的瀏覽器上執行。Node.js 的出現使 JavaScript 也能用於伺服器端編程。Node.js 含有一系列內置模組，使得程式可以脫離 Apache HTTP Server 或 IIS，作為獨立伺服器執行。

2.3.7 Ajax

AJAX 即「Asynchronous JavaScript and XML」（非同步的 JavaScript 與 XML 技術），指的是一套綜合了多項技術的瀏覽器端網頁開發技術。Ajax 的概念由傑西·詹姆士·賈瑞特所提出。

傳統的 Web 應用允許用戶端填寫表單，當送出表單時就向網頁伺服器傳送一個請求。伺服器接收並處理傳來的表單，然後送回一個新的網頁，但這個做法浪費了許多頻寬，因為在前後兩個頁面中的大部分 HTML 碼往往是相同的。由於每次應用的溝通都需要向伺服器傳送請求，應用的回應時間依賴於伺服器的回應時間。這導致了使用者介面的回應比本機應用慢得多。

與此不同，AJAX 應用可以僅向伺服器傳送並取回必須的資料，並在用戶端採用 JavaScript 處理來自伺服器的回應。因為在伺服器和瀏覽器之間交換的資料大量減少，伺服器回應更快了。同時，很多的處理工作可以在發出請求的用戶端機器上完成，因此 Web 伺服器的負荷也減少了。

第三章、 專題成果

3.1 系統特色

本專題的服務對象為網頁前端工程師，透過我們的網站讓前端工程師可以方便的去修改 Bootstrap 的程式碼。而我們的特色有以下幾點：第一，我們是採用 LESS 語言去修改 Bootstrap 的 CSS 語言，透過 LESS 語言，我們可以做到以往 CSS 做不到的事，例如變數、function、迴圈、數學上的運算。於是對於一個專業的網頁前端工程師，撰寫 LESS 絕對會比 CSS 方便很多。第二，我們可以行動端編輯程式碼，在這個人手一機的時代，編程不再需要編輯器或電腦，只要有一台手機，連上我們的網站，人人都能編程。第三，雲端儲存，我們將使用者的程式碼都儲存在雲端，隨時方便編輯或提供使用者匯入自己的網頁使用。第四，直覺操作，我們網站有著清晰、直覺化的介面並且支援 RWD，就算是初來乍到的使用者也能夠透過我們設計的流程，輕鬆使用我們的網站。



圖 3.1 網站首頁



圖 3.2 網站首頁(手機端)

網站介面支援 RWD，即使使用者透過手機也能夠方便操控，並且能夠在手機上輕鬆的編寫程式，不用再需要一台電腦、一個編輯器以及你的程式碼檔案，因為我們將編輯器直接嵌入在我們的網頁中，而程式碼檔案則是儲存在我們的資料庫，對於一個出門在外，或著是靈機一動，突然想編寫程式的使用者來說，只要有一支手機，就能達到他的需求。

3.2 專題資料 Hierachy

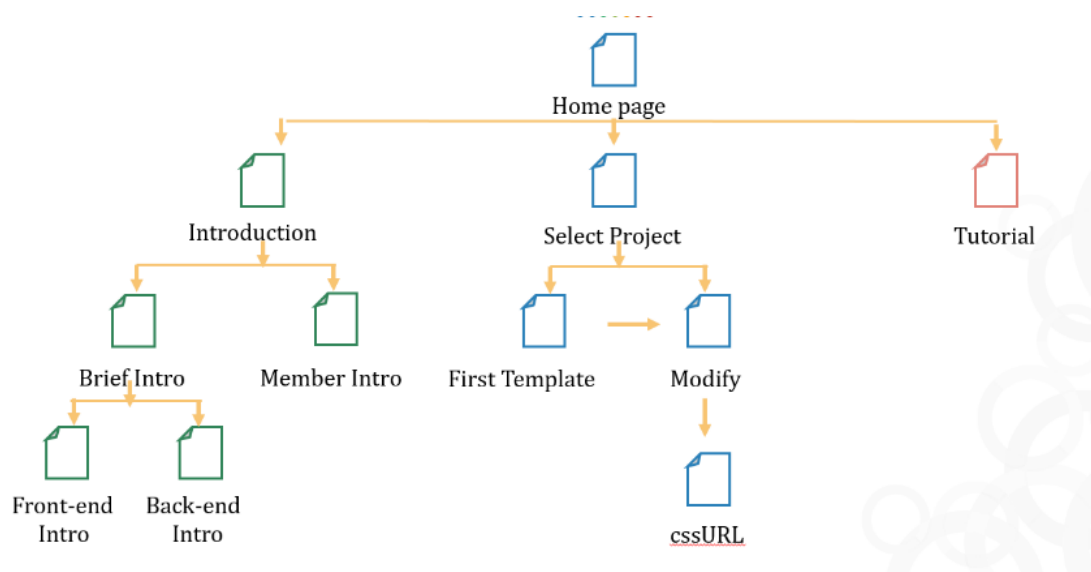


圖 3.3 前端頁面 Hierachy

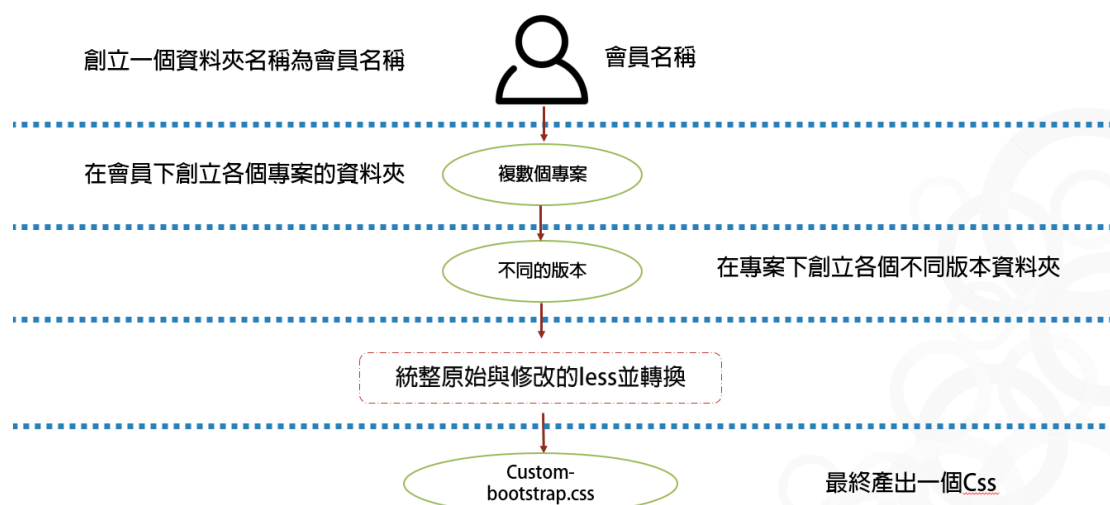


圖 3.4 後端資料 Hierachy

3.3 修改介面

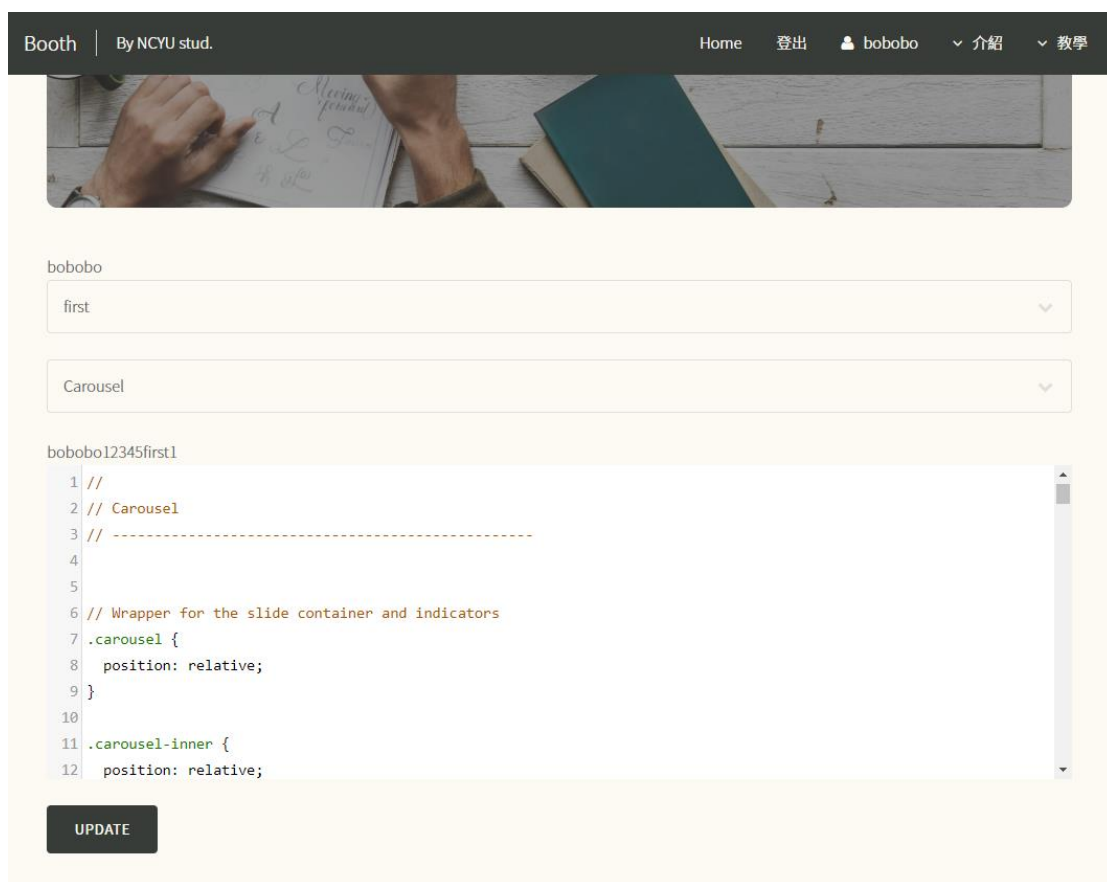


圖 3.5 修改頁面

修改介面一開始我們會先選擇此專案下的模板，選完模板後即可選擇想要修改的元件。例如若使用者想修改此專案中的第一模板的 Table 元件，那就只要先選取第一模板，接著再選擇元件 Table，這樣瀏覽器中的編輯器將會只出現專屬於第一模板且元件 Table 的程式碼，使用者無需再去從原本是幾千，幾萬行中，且已經被壓縮過後的 CSS 程式碼中苦苦尋找，我們直接呈現出來的程式碼即是使用者想修改的部分。



圖 3.6 修改頁面(手機端)

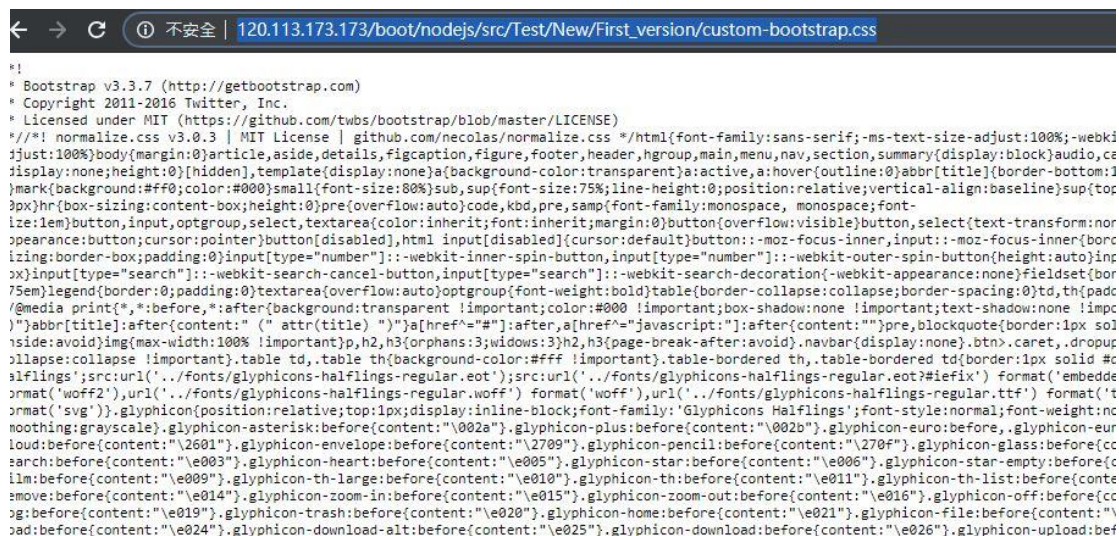


圖 3.7 取得 CSS 之 URL

我們會即時將使用者修改的 LESS 程式碼轉換成 CSS 程式碼，並且作壓縮後成為一行程式碼後，產出一組 URL，這樣使用者就能將這組 URL 引入自己的 HTML 文件中使用，並且在第一次動作後，接下來的修改都無須再去做引入的動作，因為我們給的 URL 是固定的，並且是階層式的 URL，使用者能夠清楚明瞭自己當下拿到的是哪一個專案中的模板的 CSS。

第四章、 結果與討論

未來相信會有越來越多人使用 Bootstrap 框架系統，目前我們還尚未能提供將全部的 LESS 檔都在我們的網頁上修改的功能，而在將 LESS 轉換為 CSS 並壓縮這部分速度也還有待加強。之後會盡可能的提供這些服務並加強後端系統，使得未來前端網頁工程師將來能夠透過我們的網站，擁有友善的體驗，並且對 Bootstrap 修改上也更加得方便。

第五章、 結論

我們開發了一個 Bootstrap 的修改系統，讓開發者能夠方便的將欲處理程式碼映射到源程式碼，進而使得開發者有更自由的空間，不必受限於源代碼的框架，同時也解決了不一致性的問題、以及降低維護上的成本。希望透過我們製作的專題讓網頁前端工程師們知道，修改 Bootstrap 的程式碼不再是非常困難的一件事。

參考文獻

- [1] LESS GITHUB

<https://github.com/less/less.js>

- [2] Gulp 安裝

<http://www.oxxostudio.tw/articles/201503/gulp-install-webserver.html>

- [3] 用 LESS 寫 CSS

<http://www.oxxostudio.tw/articles/201601/css-less-01.html>

- [4] RWD-維基百科

<https://www.ibest.tw/page01.php>

- [5] Bootstrap 官網:

<https://getbootstrap.com>

- [6] UI Bootstrap

<https://angular-ui.github.io/bootstrap>

- [7] UI UX 設計

<https://transbiz.com.tw/ui-ux-design-difference>

[8] w3schools.com(網頁程式教學網站)

https://www.w3schools.com/html/html_scripts.asp