東海大學資訊管理研究所

碩士學位論文

結合 PHP 與 CSS 框架之網站建置最佳化

Optimize web development on PHP and CSS framework

指導教授：姜自強 博士

研 究 生：白翰霖 撰

中華民國 102 年 6 月 5 日

東海大學資訊管理學系碩士學位

考試委員審定書

資訊管理學系研究所 白翰霖 君所提之論文

結合 PHP 與 CSS 框架之網站建置最佳化

經本考試委員會審查，符合碩士資格標準。

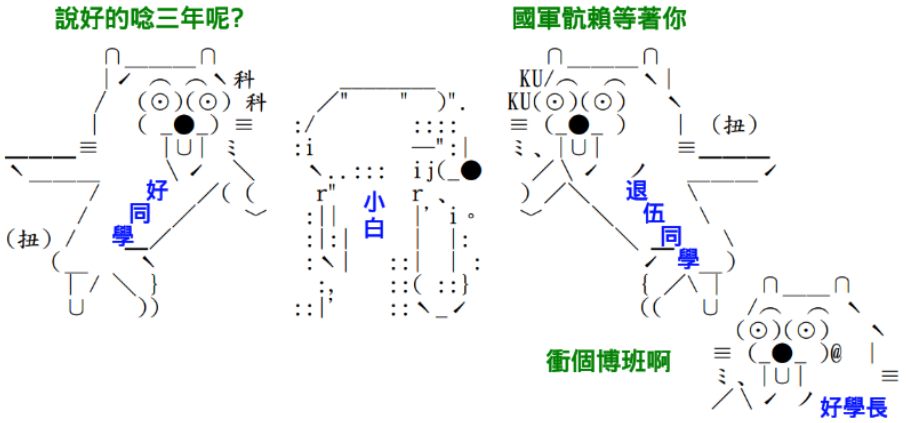
學位考試委員會 召集人：（簽章）

委 員：

中 華 民 國 102 年 06 月 05 日

致謝

短暫的碩士班生涯當中，卻是我學習生涯當中最精華不過的一個階段了，怎麼說呢？這得從我大學四年級時講起。那時的我，自以為學了點技術懂了點皮毛便不可一世，也因此吃了不少苦頭而導致延畢，直到隔年畢業，我也還是個不知道未來要幹什麼的小鬼，說實在話，要一個從正統台灣教育體系成長上來的學生在大學畢業後馬上有個他媽的人生目標還真的給他很困難，因此我選擇進了碩士班繼續逃避，是的，逃避，至少當時的我是這樣想的。上了碩士班之後，沒了大量的必修學分，有了更多的時間可以運用，結識了技術同好，參加了許多大大小小的技術分享研討會，開闊了眼界的同時，更讓我變得謙虛，懂得虛心受教與自我學習，這是我在大學時期所欠缺的。直到現在口試結束畢業了，我覺得似乎已經不需要找尋人生目標了，因為我發現接下來我想要做的事情很多，一一去完成這些事就是我的人生目標吧。以下是個人意見，我覺得其實沒必要真正去定位說要完成一件什麼樣的大夢想或偉大的事，因為在達成那些事情之前，還有好多事情等著完成呢，如果光顧著挖掘或找尋那所謂的目標或夢想，而錯過了身旁一閃即逝的美好時光，再來後悔豈不可惜，藉此機會與大家共勉。以下為感謝文，首先感謝口試委員們給予我論文的建議與肯定，再來感謝我的碩士班指導老師姜自強教授給了我充足的時間與空間讓我發揮與成長，也要感謝前兩屆的學長姐 Blue、小草與莎百在我懵懂進入碩士班時的帶領，同時也感謝第三屆資管穴兄弟們的陪伴與成長，最後，感恩在我身邊的每一個人。



論文名稱：結合 PHP 與 CSS 框架之網站建置最佳化
校所名稱：東海大學資訊管理學系研究所
畢業時間：2013年06月
研究生：　白翰霖　　指導教授：姜自強
論文摘要：

由於網際網路的興起，帶動了許多中小型企業乃至民間組織團體紛紛建立起屬於自己的網站，但是網站的效能及安全性卻是參差不齊，起因於每個網站開發人員的設計架構與理念的不同。另外，手持行動裝置的盛行也在網站開發人員之間掀起了一股熱潮，網站的開發是否能適應於行動裝置瀏覽變成了開發人員的必修課題。因此本研究旨在推行一種結合基於 MVC 架構[2]（Model-View-Controller Pattern 又稱模型-檢視-控制器架構）的 CodeIgniter[5] PHP 框架以及 Twitter Bootstrap[6] CSS 框架的開發模式，並提出網站效能、網站安全性以及跨平台瀏覽三項指標做為評估依據。先與其他業界常用的框架做效能上的比較，然後根據 OWASP（The Open Web Application Security Project）組織在 2013 年所提出十大安全性議題[17]中的三項：SQL Injection、Cross-Site Scripting（XSS）、Cross-Site Request Forgery（CSRF）去模擬攻擊並提出防範方法，最後利用自適性設計使網站可跨平台瀏覽，證明此開發模式是具有其優勢且可以藉此改善台灣現有的企業網站。

關鍵詞：模型-檢視-控制器架構、CodeIgniter PHP 框架、Twitter Bootstrap CSS 框架、SQL Injection、Cross-Site Scripting、Cross-Site Request Forgery

Title of Thesis：Optimize web development on PHP and CSS framework

Name of Institute: Tunghai University, Institute of Information Management

Graduation Time：（06／2013）

Student Name：Han-Lin Pai　　Advisor Name：Tzu-Chiang Chiang

Abstract：

Many enterprise and private organization starts to build their own website because of internet growing. But each of their websites still has some problem like performance and security. This situation caused by different design style of different developers. Besides, the mobile supporting of website becomes the new lesson of website developer. So this research aims to design a development model combine CodeIgniter PHP framework based on MVC pattern (Model-View-Controller Pattern) and Twitter Bootstrap CSS framework, and proposes three metrics that are website loading speed, web security and cross-platform. First, compare another framework with performance, and using three of top ten security issues from OWASP group to simulate attack activities. In the end, we conclude the comparison data and defend method to prove this development model can improve the old websites.

Keywords：Model-View-Controller Pattern, CodeIgniter PHP framework, Twitter Bootstrap CSS framework, SQL Injection, Cross-Site Scripting, Cross-Site Request Forgery

目次

表次

圖次

第一章 前言

由於網際網路的興起，帶動了許多中小型企業乃至民間組織團體紛紛建立起屬於自己的網站，但是各家網站的效能卻是參差不齊，由於開發人員的設計概念不同，所建置起的網站架構也就不同，其所造成的不便之處，如：後續維護、工作交接以及美術設計人員與程式設計人員的分工等問題都需要額外耗費人力成本。除此之外，資訊安全一直是消費者相當在意的問題，並且由於個資法規的推行，也間接促使了人們更重視自己的個人資料，因此如果企業網站毫無資訊安全可言，就等同於將客戶的資料陷於外流的險境，如此的一間企業或公司行號沒有客戶會願意信任。此外，行動裝置的進步以及其龐大的銷售量也在網路界興起一陣不小的旋風，且隨著人們花在行動裝置上的時間逐漸拉長，網站開發者漸漸開始正視行動瀏覽的問題，如何讓使用者能更快速的透過行動裝置的瀏覽取得其所需的資訊成為了網站開發者的必修課題。因此本研究旨在推行一種結合基於 MVC 架構（模型-檢視-控制器架構）的 CodeIgniter PHP 框架以及 Twitter Bootstrap CSS 框架的開發模式，並提出網站效能、網站安全性及跨平台瀏覽三項指標做為評估依據。先與業界常用的框架做效能上的比較，然後根據 OWASP（The Open Web Application Security Project）組織在 2013 年所提出十大安全性議題中的三項：SQL Injection、Cross-Site Scripting、Cross-Site Request Forgery 去模擬攻擊並提出防範方法，最後利用自適性設計使網站可跨平台瀏覽，證明此開發模式是具有其優勢且可以藉此改善台灣現有的企業網站。本研究預計達成之研究目的為：

1. 證明 CodeIgniter 在效能上較業界知名的常用框架具有優勢。
2. 證明 CodeIgniter 在資訊安全方面有其因應的對策。
3. 證明 Twitter Bootstrap 能有效地將網頁適應行動裝置瀏覽。
4. 利用產學合作方式以本研究所提出之開發模式實作。

第二章 文獻探討

第一節 PHP 框架：CodeIgniter

CodeIgniter 是一套開發 PHP 應用程式的框架，提供簡易的介面與清晰的邏輯結構來使用豐富的函式庫，其目的在於加速開發速度，只需寫少量的程式碼，便可達成原先需大量程式碼所編撰的功能，讓程式設計師可將注意力集中在專案開發。CodeIgniter 最早是由 Rick Ellis 開發的（EllisLab 公司的 CEO），它針對效能性開發了許多物件類別函式庫及輔助函數（helpers），以及參考 Expression Engine 基底開發子系統，目前由 Expression Engine 開發團隊開發並維護。

為什麼選擇 CodeIgniter？在伺服器需求上，它支援 PHP5 以上的版本，資料庫的部分則支援MySQL（4.1+）、MySQLi、MSSQL、Postgres、Oracle、SQLite 及 ODBC 等資料庫引擎，為用戶提供了多樣性的選擇，不必受限於支援的太少而屈就。

為什麼選擇 CodeIgniter？它採用 Apache/BSD-style open source license 授權，讓用戶可隨心所欲的使用，但請務必詳讀下述使用規範。在滿足下列條件的前提之下，EllisLab 允許使用者對此軟體及其文件做使用、複製、修改以及重製行為：

1. 軟體重製時需含有本許可協議的副本。
2. 重製後的所有軟體原始碼當中需保留完整的著作權聲明。
3. 以二進位碼呈現的重製軟體必須在程式本體或其他說明資料中包含授權條款的著作權聲明和授權條款的所有內容。
4. 所有被修改後的檔案必須註明被修改的部分以及修改者的姓名。
5. 衍生自本軟體的產品必須在文件中承認他們使用了來自於 CodeIgniter 所提供的功能元件。
6. 衍生自本軟體的產品在沒有 EllisLab 事先書面許可之下，不能擅自將產品命名為 CodeIgniter 或者含有任何 CodeIgniter 字樣。

為什麼選擇 CodeIgniter？它極致的輕量化，核心系統只需少量的程式即可運行，比較起其他需要耗費大量資源的 Framework 相對較優，如果需要額外的函式庫，亦可以透過動態載入的方式使用，所以其基本的系統是相當的精簡且快速。

為什麼選擇 CodeIgniter？效能也是其引以為傲的地方，相較於業界常見的 Zend Framework 以及 CakePHP 速度都要快上許多，另外在記憶體的使用也相當的精簡，這兩項是 CodeIgniter 相當為人所稱道的地方。

為什麼選擇 CodeIgniter？它使用 MVC 架構（模型-檢視-控制器），將邏輯結構與視覺呈現分離，這樣的方式對於專案開發相當的有利，程式設計師可專心設計程式邏輯其較功能面的部份，而美術設計師僅需要處理含有最少量程式的樣板檔案即可。

為什麼選擇 CodeIgniter？它所產生的 URLs 相當的淺顯易懂且對於搜尋引擎來說是相當友善的，相較於傳統的字串查詢（query string）方法，她所採用的是分段式的方法。

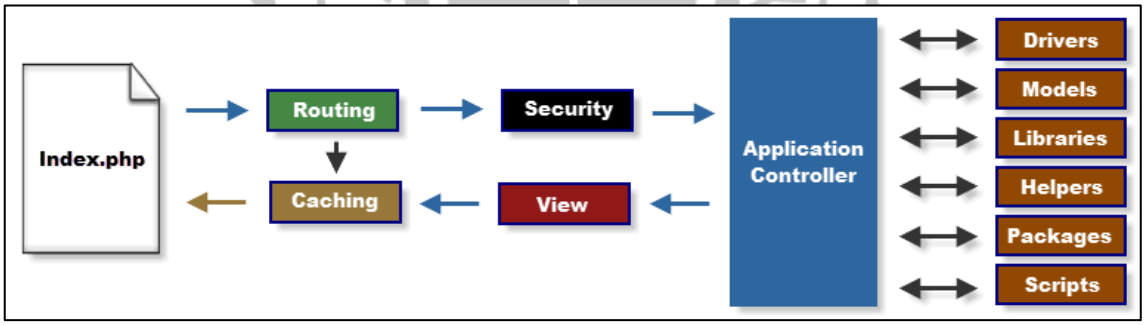


圖 2-1 CodeIgniter 的程式流程圖

第二節 CSS 框架：Twitter Bootstrap

程式設計師在開發網站時，最苦惱的問題莫過於版面太陽春，無論是按鈕、表格或是表單，所呈現的皆是預設的元件，除非開發者本身熟悉 CSS 語法的運用，也同時須具有一定的美感才有可能讓網站的版面不至於太單調死板，當然可以額外請美術人員提供相關協助，但是這樣一來所耗費的人力物力皆提高不少。因此與其花費心思在這些煩惱上，倒不如使用現有的 CSS 框架來輔助。

Twitter Bootstrap 簡單清晰又不失大方的版面格局，預先訂制好的常用元件，

加上採用 Apache/BSD-style open source license 授權可自由調整其語法以達客製化，同時亦具有相當良好的擴充性。使用 Twitter Bootstrap 非但可以讓程式設計師在向客戶呈現版面時不失體面，亦可以藉由其他網站（如. bootswatchr、stylebootstrap）所提供的客製功能來訂製客戶想要的配色與格調。

首先，Twitter Bootstrap 將網站以網格（Grid）方式配置，將整個網頁版面分成 12 個格子（12-column grids），並同時加入自適性設計（Responsive Design），所謂的自適性設計就是網頁會根據顯示設備的解析度自行定義其樣式（Style），防止傳統設計因瀏覽設備的大小不同而產生破圖甚至是異位的結果，而導致使用者無法清楚辨識其內容，同時為完善自適性設計，Twitter Bootstrap 更藉由最小寬度（min-width）與最大寬度（max-width）來針對多項行動裝置制定樣式表，還可以讓使用者以自定義的方式控制在特定類型裝置中顯示或隱藏，如：平板裝置則顯示（.visible-tablet）或手持裝置則隱藏（.hidden-phone）等。因此，Twitter Bootstrap 在跨平台網站設計的這個部份可說是相當合適的。

圖 2-2 網格式分割網頁

Label	Layout width	Column width	Gutter width
Large display	1200px and up	70px	30px
Default	980px and up	60px	20px
Portrait tablets	768px and above	42px	20px
Phones to tablets	767px and below	Fluid columns, no fixed widths	
Phones	480px and below	Fluid columns, no fixed widths	

圖 2-3 根據寬度來定義不同類型的裝置

```
/* Large desktop */
@media (min-width: 1200px) { ... }

/* Portrait tablet to landscape and desktop */
@media (min-width: 768px) and (max-width: 979px) { ... }

/* Landscape phone to portrait tablet */
@media (max-width: 767px) { ... }

/* Landscape phones and down */
@media (max-width: 480px) { ... }
```

圖 2-4 根據寬度來定義不同類型的裝置（CSS 語法）

在完成版面配置之後，接下來可藉由選用 Twitter Bootstrap 制定好的元件來佈置網站，對於網站基礎元件的設計 Twitter Bootstrap 也不馬虎，無論是表格（Table）、表單（Form）、圖示（Icon）或字型（Typography），都經過重新調整並賦予新的形象，相較於預設定義的元件，可謂是有過之而無不及，另外當然也可以依據使用者的需求予以調整。另外，Twitter Bootstrap 更提供了當今許多大型網站常用的特色元件來輔助使用者開發，如動態下拉式選單（Dropdowns）、導航條（Navbar）及分頁（Pagination）等等。只要簡單的在元件上加上 style class 就會自動套用 Twitter Bootstrap 訂製好的特色元件，非常的方便又迅速。

圖 2-5 傳統 HTML 語法所產生的表單

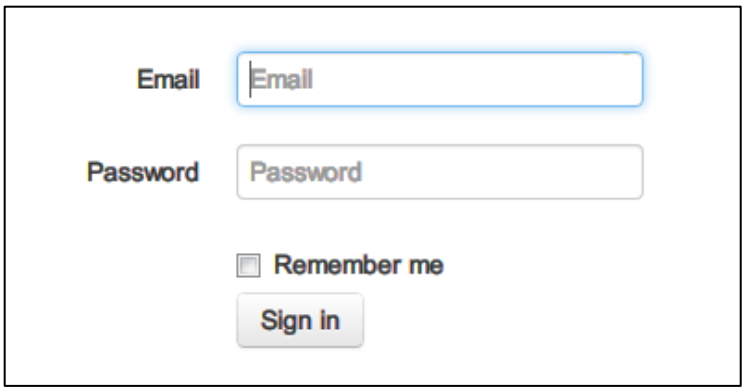


圖 2-6 利用 Twitter Bootstrap 設計的表單

第三節 Google Chrome 開發人員工具：Network Panel

開發人員工具又稱「DevTools」[7]，開發人員工具內建於 Google Chrome 瀏覽器當中，其目的是讓 WEB 開發人員可以藉由瀏覽器來深入的存取網頁應用程式相關的元件，並提供除錯以及監控等功能。而 Network Panel 在開發人員工具中扮演著紀錄網站或網頁應用程式所有相關於網路運作資訊的角色，包含了 detail timing data、HTTP request, response and headers、cookies 及 WebSocket data 等等資訊。

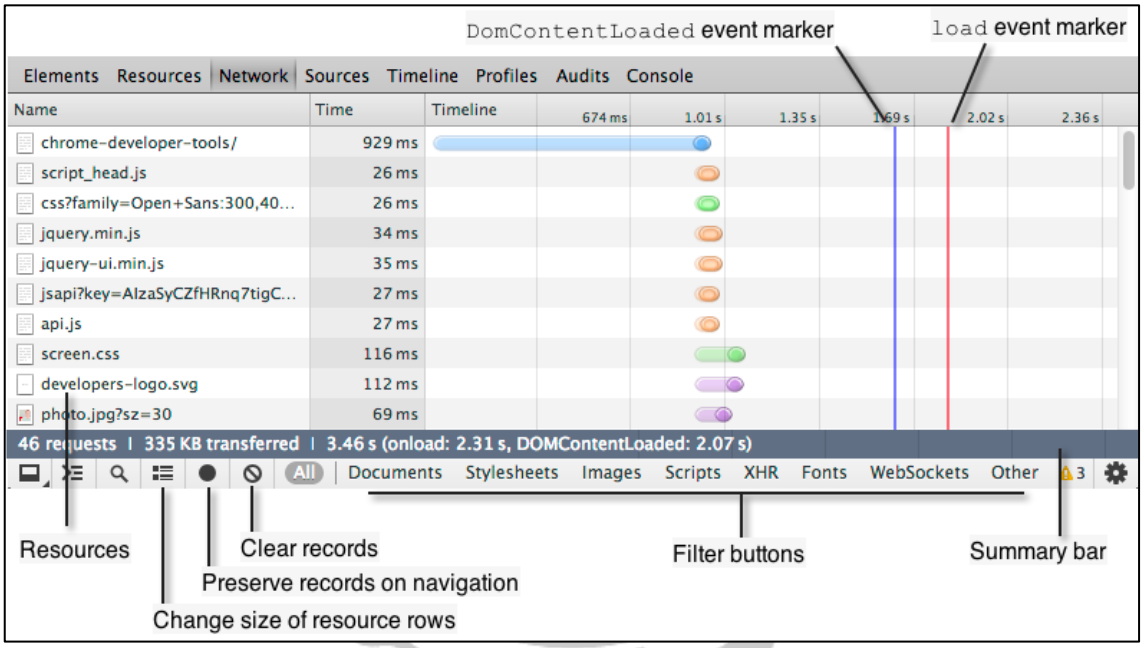


圖 2-7 Network Panel 總覽

Network Panel 只會在 DevTools 開啟的狀態之下記錄所有的網路活動資訊，所以如果先開網頁再開 Network Panel 就會有沒紀錄的情況發生，這時只需要重新整理網頁就會開始記錄。而每一個請求或資源都會以一行的形式被記錄在 Network Table 當中，下方的表格列出了所有 Network Table 當中所有的欄位。

表 2-1 Network Table 欄位說明

欄位	描述
Name and Path	資源的名稱及 URL Path
Method	請求資源使用的 HTTP Method（GET, POST）

Status and Text	資源目前的 Status Code 跟 Text Message
Domain	資源來自於哪個 domain
Type	資源的檔案型態 MIME type
Initiator	產生請求的物件（Parser, Redirect, Script or Other）
Cookies	請求所傳遞的 cookie 總數
Set-Cookies	有多少 cookie 在本次請求當中被設定
Size and Content	Size 是 response header 加上 response body 的總大小 Content 是資源經過解碼後的大小
Time and Latency	Time 是從請求的開始到結束總時間 Latency 是請求開始到開始回應之間所經過的時間
Timeline	此欄位會將所有請求繪製成時間軸

壹、持續追蹤網路活動

在預設的情況下，當你切換或重新整理網頁的時候 Network Panel 就會清空原有的紀錄，為了持續性的紀錄網路活動，可藉由按下在 Network Panel 底部的持續記錄鈕  來啟動此功能，新紀錄的資訊就會加入到 Network Table 而不會清空原有資訊，如果不需要此功能便可藉由按下同樣的按鈕（現在它是  ）來停用此功能。

貳、排序及過濾

在預設的情況下，Network Table 當中的資料都是根據開始時間來做排序的，當然你可以有更多的選擇，藉由點選不同的標頭欄位來進行不同類型的排序，另外重複點擊同樣的標頭欄位可以做升冪排序跟降冪排序的切換。

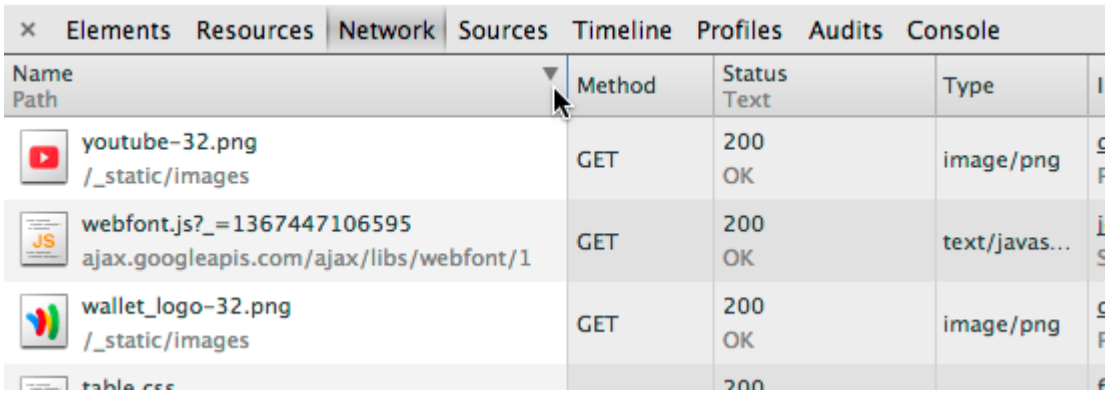


圖 2-8 點選標頭欄位進行排序

Timeline 標頭欄位比較特別，排序的部分它具有下列選項供使用者選擇：

1. Timeline—根據每個請求的開始時間做排序（此為預設值）
2. Start time—根據每個請求的開始時間做排序(排序結果跟 Timeline 相同）
3. Response Time—根據每個請求的回應時間
4. End Time—根據每個請求的結束時間
5. Duration—根據每個請求的總花費時間
6. Latency—根據請求開始到開始回應之間所經過的時間做排序

如果想過濾掉不相關的檔案類型，可藉由點選 Network Panel 底部的標籤群：All、Documents、Stylesheets、Images、Scripts、XHR、Fonts、WebSockets 和 Other，來進行特定檔案類型的過濾，下方截圖所代表的是只顯示 CSS 資源，如果要看所有檔案類型，點「All」標籤按鈕即可。

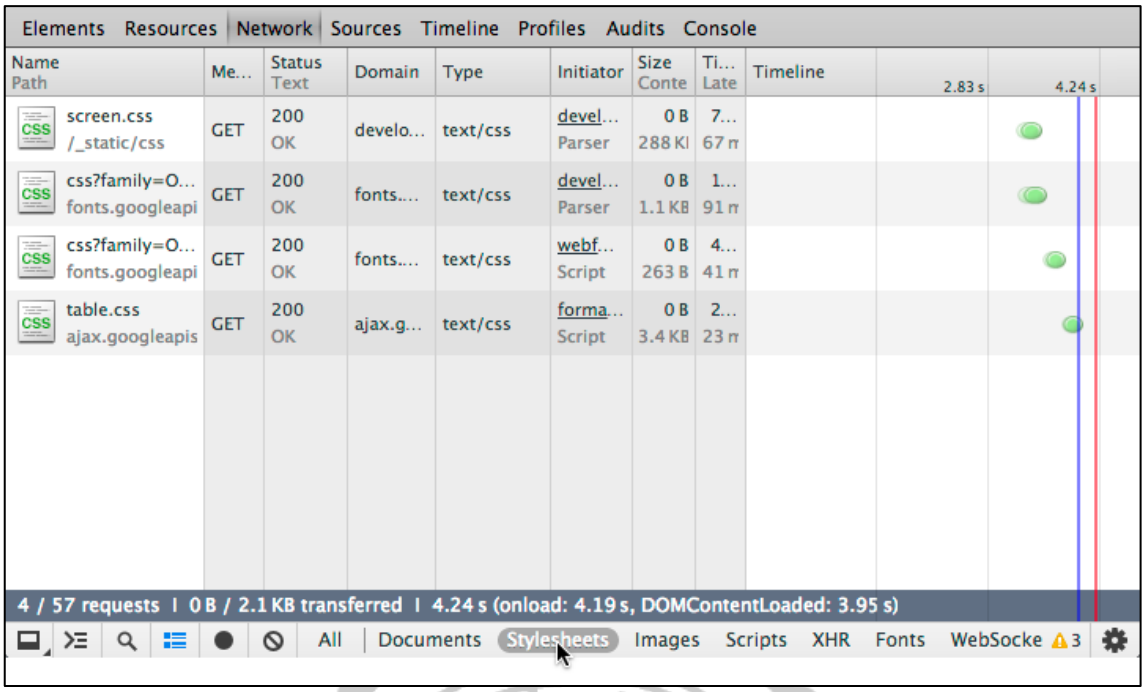


圖 2-9 利用 Network Panel 所提供的過濾標籤僅顯示 CSS 資源

參、增減表格欄位

Network Table 中的欄位是可以自由增減的，只要在標頭欄位點右鍵並勾選列表中所需增減的欄位即可。

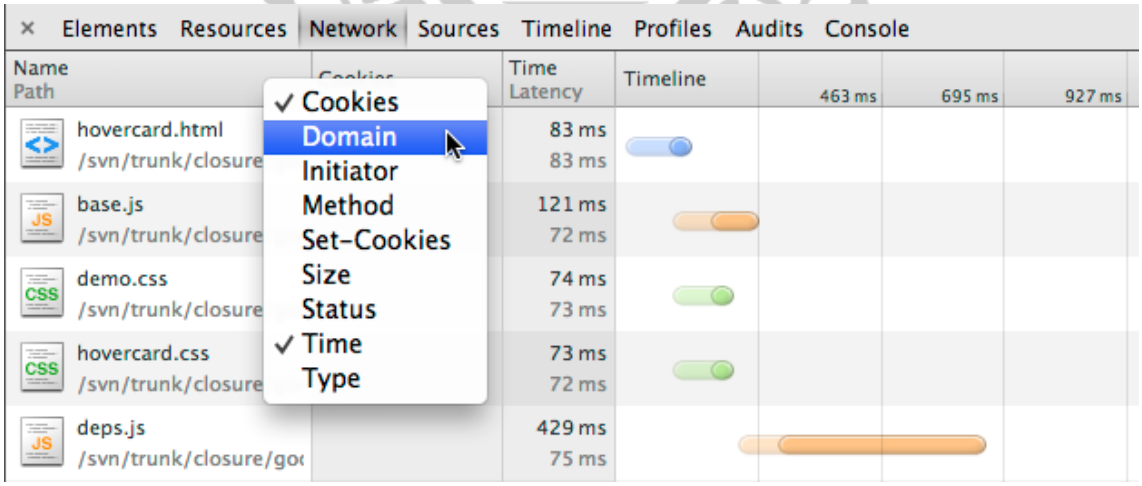


圖 2-10 可自由變更 Network Table 中預設的欄位

肆、切換單行/多行模式

在預設的情況下所看到 Network Table 是處於多行模式，而多行模式中的每一行高度相對於單行模式會較大，所以如果今天需要觀看大量的資料則需要切換為

單行模式，可藉由點擊位於 Network Panel 底部的切換鈕  ，切換為單行模式後圖示會變為  ，而欄位當中的 Path、Text、Content 及 Latency 四個副項目就會被隱藏只顯示主項目以節省空間。

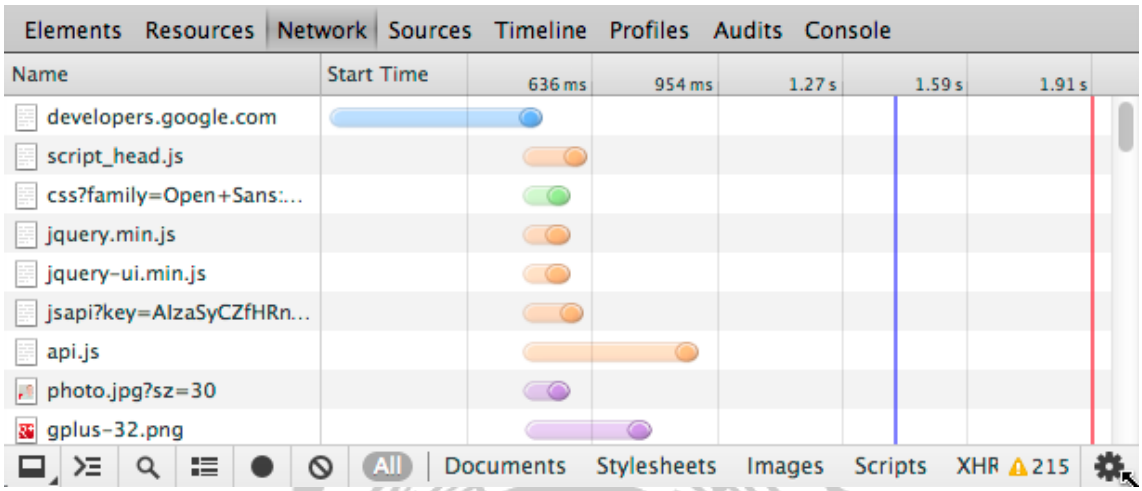


圖 2-11 切換為單行模式方便瀏覽 Timeline 欄位

伍、時間軸

Network Panel 會將所有 HTTP request 根據其請求開始至回應結束的時間以時間軸的模式繪製成圖，每一個 request 會以一個 bar 呈現並根據其檔案型態予以不同的顏色，且在每個 bar 當中，顏色較淺且長度較長的部分代表該 request 的 latency 值，顏色較深且長度較短的部分代表接收到 response data 所花的時間。

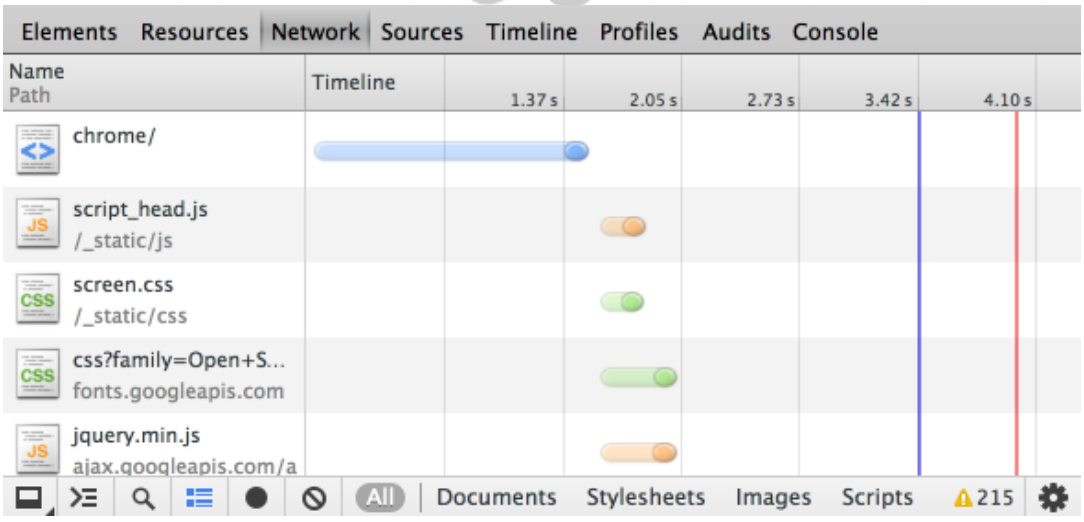


圖 2-12 時間軸的呈現

當你將滑鼠移至 Network Table 的某一行當中（不是移到 bar 之上），該 request 的 latency 值與接收到 response data 所花的時間分別會顯示在 bar 的顏色較淺處與顏色較深處。

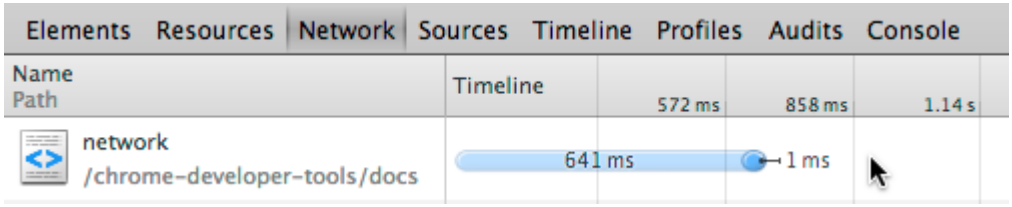


圖 2-13 latency 與 receipt time 的顯現

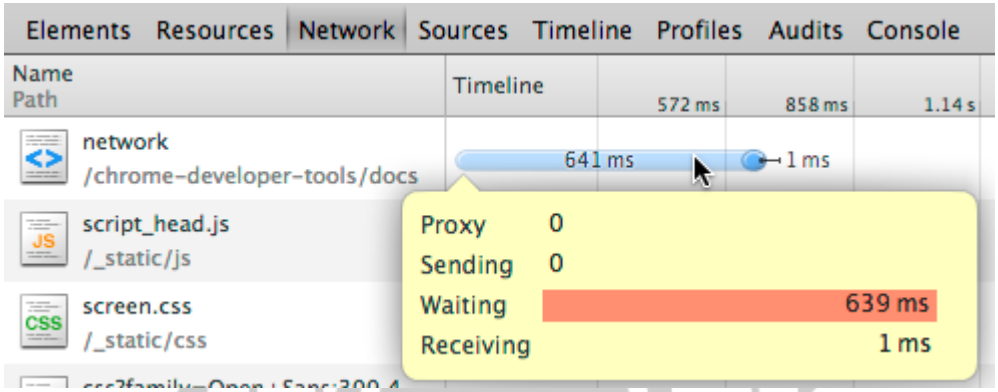


圖 2-14 將滑鼠移至 bar 上時，完整的時間紀錄資料會以 popup 的方式呈現

當 DOMContentLoaded()與 load()事件觸發時，Timeline 會以藍色以及紅色的垂直線分別代表兩事件，其線所在的位置就代表著觸發的時間。DOMContentLoaded()觸發的時間點在於 document 已經被完整的讀取並產生成網頁了，而 load()觸發的時間點在於網頁當中的所有元素（如.圖片或影像等）都完整的讀取完畢時。

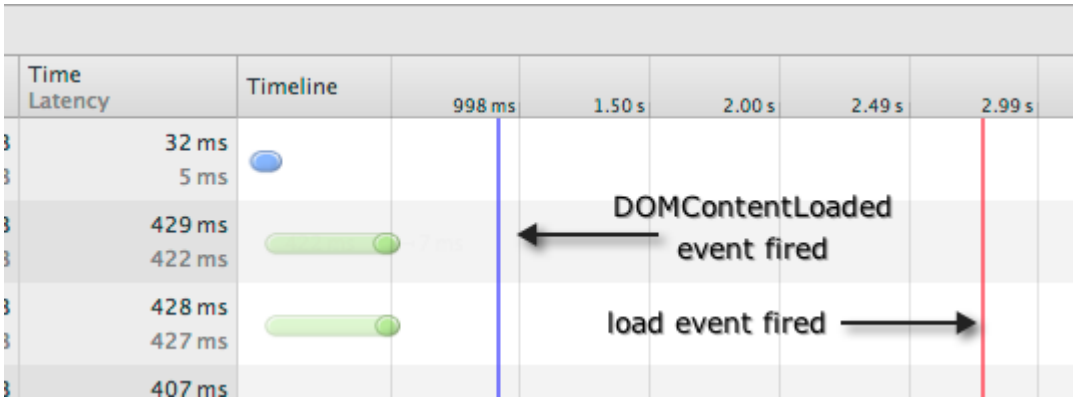


圖 2-15 藍線代表 DOMContentloaded()事件觸發 紅線代表 load()事件觸發

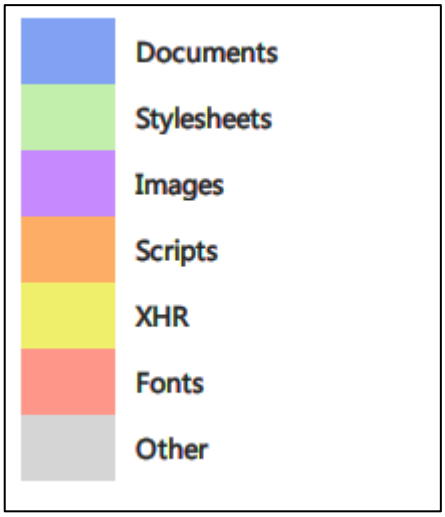


圖 2-16 Timeline 中 bar 的顏色所代表的檔案型態

陸、儲存及複製網路資訊

對著 Network Table 的任一行請求資訊點擊右鍵即會跳出選單，以下將會列出選單中所提供的選項及其功能描述：

1. Open link in new tab—以此請求為來源網址開啟新視窗也可連點開啟。
2. Copy link address—複製來源網址。
3. Copy request headers—複製 HTTP request headers。
4. Copy as curl—複製 HTTP response headers。
5. Replay XHR—如果此請求為 XMLHTTPRequest，則會重送此請求。
6. Copy all as HAR—將此請求的 network record 儲存成 HAR（HTTP Archive file）格式。
7. Save as HAR with content—將此請求的 network record 連同網頁中的內容以及圖片均以 base64 編碼後儲存成 HAR 格式。

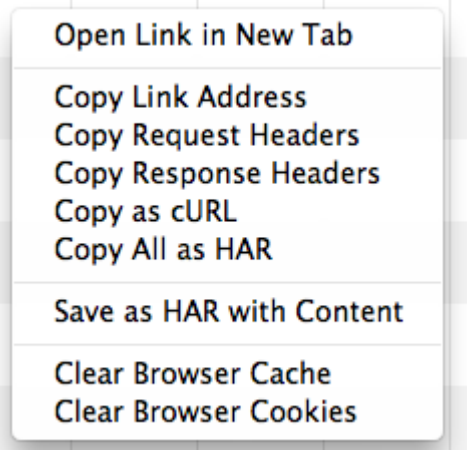


圖 2-17 對任一行請求資訊點擊右鍵後出現的選單

柒、資源/請求的細部資料

當點擊 Network Table 中的任一資源或請求的名稱將會出現由下列細節項目所構成標籤頁：

1. HTTP request and response data
2. Resource preview
3. HTTP response
4. Cookie names and values
5. Resource network timing

一、HTTP request and response data

Headers 標籤頁將會列出請求的 URL、HTTP Method 及 Response status code，此外還有 HTTP response and request headers 及所有 query string parameters。預設所瀏覽的 HTTP headers 都是經過解析及格式化過後的，如果需要切換只要點擊「view source」及「view parsed」連結即可，此二連結位於每個區塊的標題旁。另外瀏覽 query string parameters 的時候，也可以利用「View decoded」及「View URL encoded」來做編碼前後的切換。

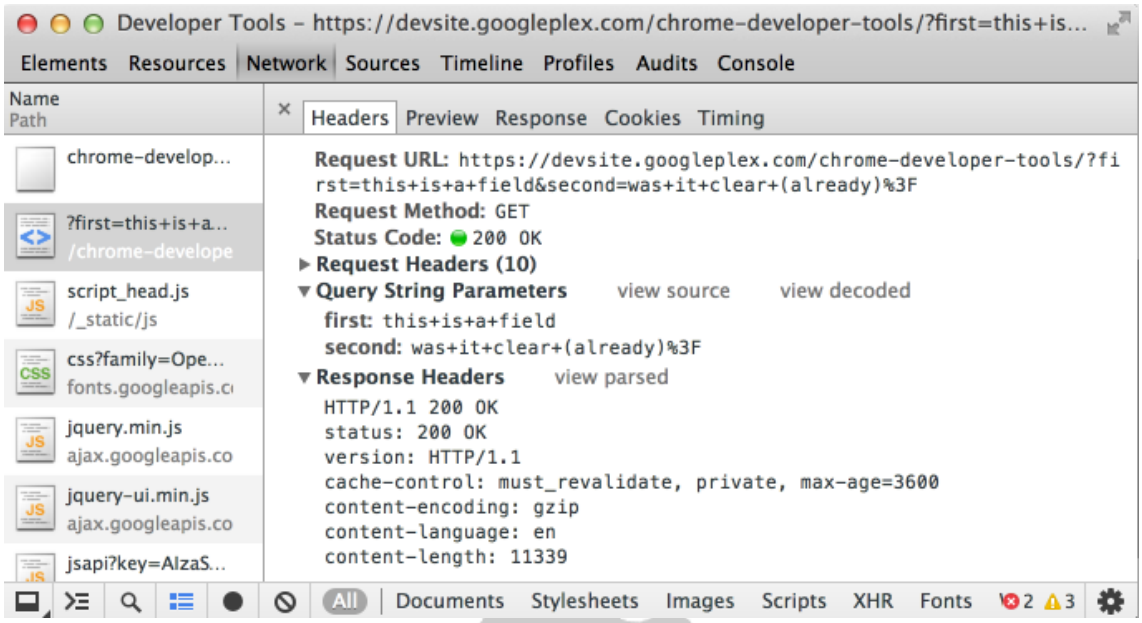


圖 2-18 Headers 標籤頁

二、Resource preview

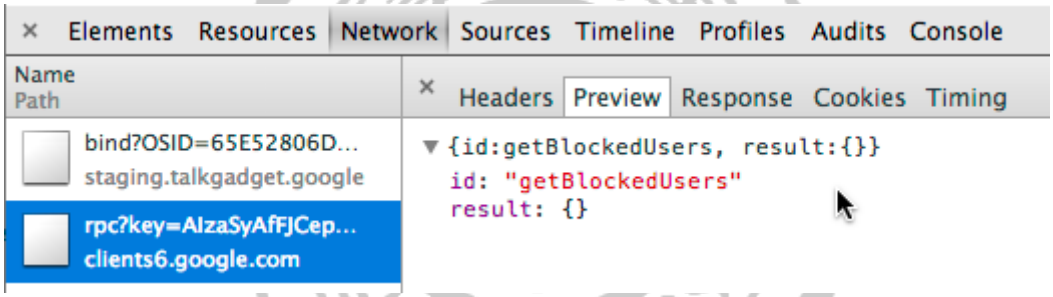


圖 2-19 Preview 標籤頁對經過格式化的 JSON 資源進行預覽

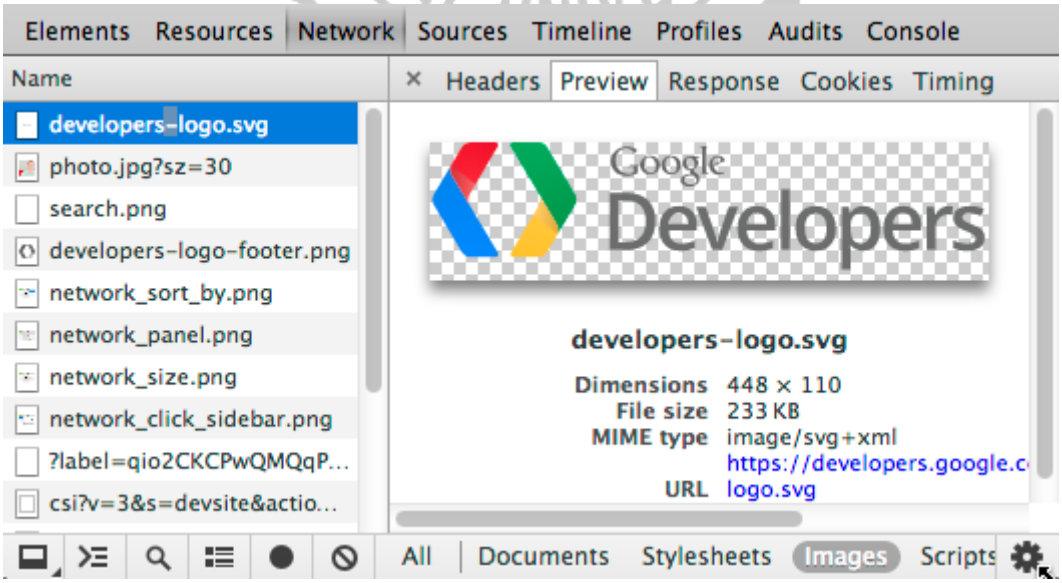


圖 2-20 Preview 標籤頁對圖片資源進行預覽

三、HTTP Response

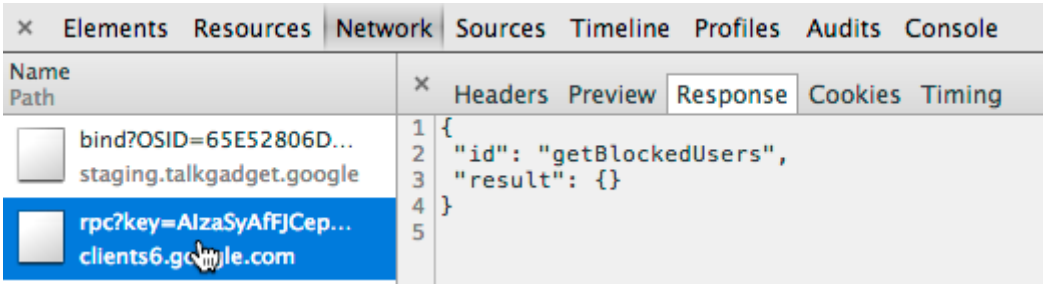


圖 2-21 Response 標籤頁直接印出未經格式化的 JSON

四、Cookie names and values

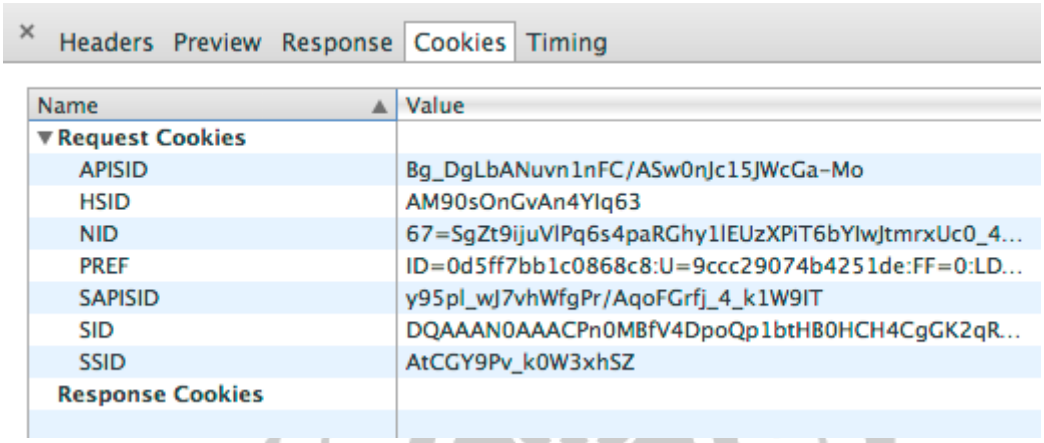


圖 2-22 Cookie 標籤頁詳列出請求過程當中所送出的 cookie 內容

表 2-2 Cookie 標籤頁各個欄位及其相關描述

Property	Description
Name	Cookie 的名稱
Value	Cookie 的值
Domain	Cookie 的網域名稱
Path	Cookie 的 URL 位址
Expires/Max-age	Cookie 的有效期限/最長時限
Size	Cookie 的大小（以 byte 計）
HTTP	這表示此 cookie 只能被瀏覽器設置，不能被 Javascript 存取。
Secure	如果出現此屬性表示此 cookie 只能在安全的連線當中被傳遞。

五、Resource network timing

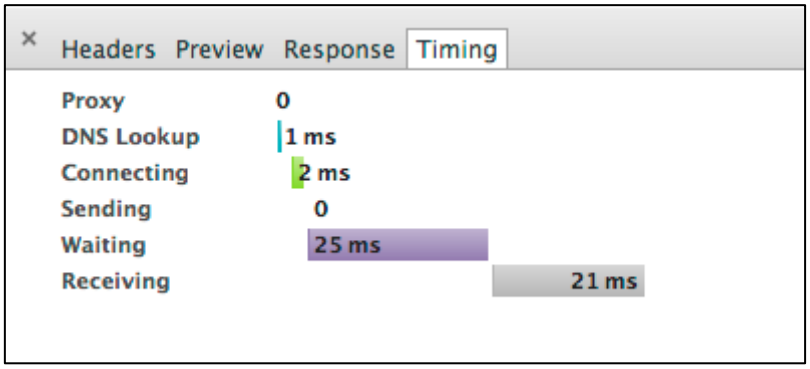


圖 2-23 Timing 標籤頁會把讀取此資源的各個請求階段所花的時間繪製成圖

表 2-3 Timing 標籤頁各個欄位及其相關描述

Property	Description
Proxy	與 proxy server 連線所花費的時間
DNS Lookup	DNS Lookup 過程所花費的時間
Blocking	重複使用已經建立的連結所花費的時間
Connecting	花在建立連結的時間。(包括 TCP Handshakes/retries、DNS Lookup、proxy connection 或 secure-socket layer)
Sending	送出請求所花費的時間
Waiting	初始化回覆所花費的時間
Receiving	接收回覆資料所花費的時間

第四節 網路安全

壹、SQL Injection

網站應用程式執行來自外部包括資料庫在內的惡意指令，SQL Injection 與 Command Injection 等攻擊包括在內。 因為駭客必須猜測管理者所撰寫的方式，因此又稱「駭客的填空遊戲」。

舉例來說，原本管理者設計的登入頁面資料庫語法如下：

```
SELECT *
FROM   users
WHERE  username = '".$user."'
AND password = '".$pass."'
```

圖 2-24 登入頁面的資料庫語法

如果說$user 以及$pass 變數沒有做保護，黑客只要輸入「' or "='」字串，就會變成如下圖 2-25 所示，這個 SQL 語法就會規避驗證手續，直接讓黑客登入甚至會顯示機密資料。

```
SELECT *
FROM   users
WHERE  username = '' OR '' = ''
AND password = '' OR '' = ''
```

圖 2-25 遭受黑客以不法字元攻擊的資料庫語法

簡述黑客攻擊流程：

1. 找出未保護變數，作為注入點
2. 猜測完整 Command 並嘗試插入
3. 推測欄位數、Table 名稱、SQL 版本等資訊
4. 完整插入完成攻擊程序

防護建議：

- 使用 Prepared Statements，例如：PHP PDO bindParam()
- 使用 Stored Procedures
- 嚴密的檢查所有輸入值
- 使用過濾字串函數過濾非法的字元，例如：mysql_real_escape_string、addslashes
- 控管錯誤訊息只有管理者可以閱讀
- 控管資料庫及網站使用者帳號權限為何

貳、Cross-Site Scripting （XSS）

網站應用程式直接將來自使用者的執行請求送回瀏覽器執行，使得攻擊者可擷取使用者的 Cookie 或 Session 資料而能假冒直接登入為合法使用者。

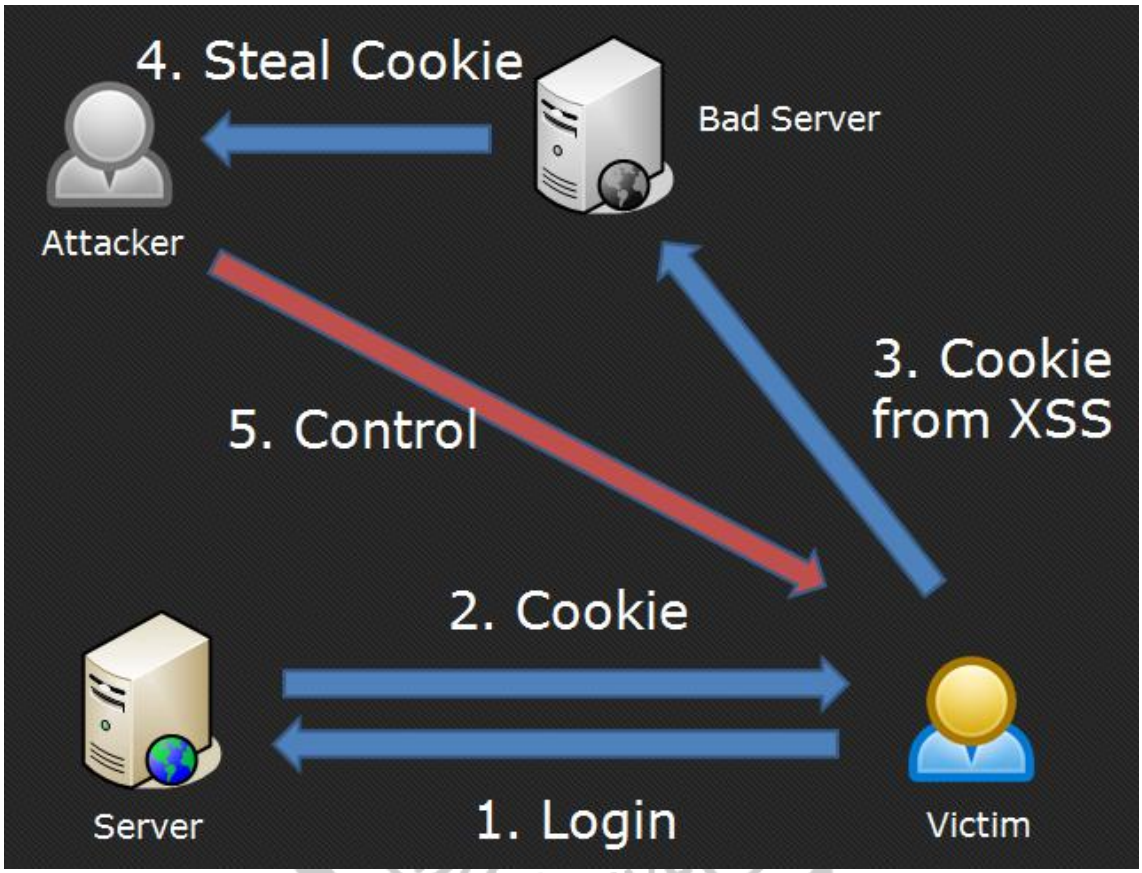


圖 2-26 XSS 攻擊流程圖

簡述黑客攻擊流程：

1. 受害者登入一個網站
2. 從 Server 端取得 Cookie
3. 但是 Server 端上有著 XSS 攻擊，使受害者將 Cookie 回傳至 Bad Server
4. 攻擊者從自己架設的 Bad Server 上取得受害者 Cookie
5. 攻擊者取得控制使用受害者的身分

防護建議：

- 檢查頁面輸入數值
- 輸出頁面做 Encoding 檢查

- 使用白名單機制過濾，而不單只是黑名單
- PHP 使用 htmlentities 過濾字串
- OWASP Cross Site Scripting Prevention Cheat Sheet
- 各種 XSS 攻擊的 Pattern 參考

參、Cross-Site Request Forgery（CSRF）

已登入網站應用程式的合法使用者執行到惡意的 HTTP 指令，但網站卻當成合法需求處理，使得惡意指令被正常執行。

舉例來說，攻擊者在網站內放置了  ，此圖片的 HTML 如圖 2-27，當受害者讀取到該圖片之後，就會去 crack.me 主機執行如冒名購買等惡意行為。

```
1 <img src='http://crack.me/buystuff.php?productId=1'>
```

圖 2-27 攻擊者所設置的 CSRF 陷阱

防護建議：

- 確保網站內沒有任何可供 XSS 攻擊的弱點
- 在 Input 欄位加上亂數產生的驗證編碼
- 在能使用高權限的頁面，重新驗證使用者
- 禁止使用 GET 參數傳遞防止快速散佈
- 使用 Captcha 等技術驗證是否為人為操作

或者參考 OWASP 所提供的 CSRF Solution

- OWASP CSRFTester Project
- OWASP CSRFGuard Project
- OWASP CSRF Prevention Cheat Sheet

第三章 開發模式之優勢較量模型

本研究旨在推行一種結合了 CodeIgniter PHP 框架及 Bootstrap CSS 框架的開發模式，用於建置兼顧網站效能、資訊安全及使用者介面的網站。網站效能部分，以 CodeIgniter 所建置出的網站為實驗對象，使用 Google 開發人員工具、Apache Benchmark[13]與 XHProf[9]來做網站效能的計算。在資訊安全方面，則根據 OWASP（The Open Web Application Security Project）組織在 2013 年所提出的 10 大安全性議題中取出下列三項安全性議題 SQL Injection、Cross-Site Scripting、Cross-Site Request Forgery 去模擬攻擊並提出防範方法。最後則是利用 Bootstrap 框架其過人的 CSS Style 及擁有 Responsive Design 的特性，將網站使用者介面打造得兼具美觀且易於操作，同時又有利於多樣行動裝置瀏覽。因此本研究提出下列三項指標做為評估依據：

1. 網站效能
2. 網站安全性
3. 跨平台瀏覽

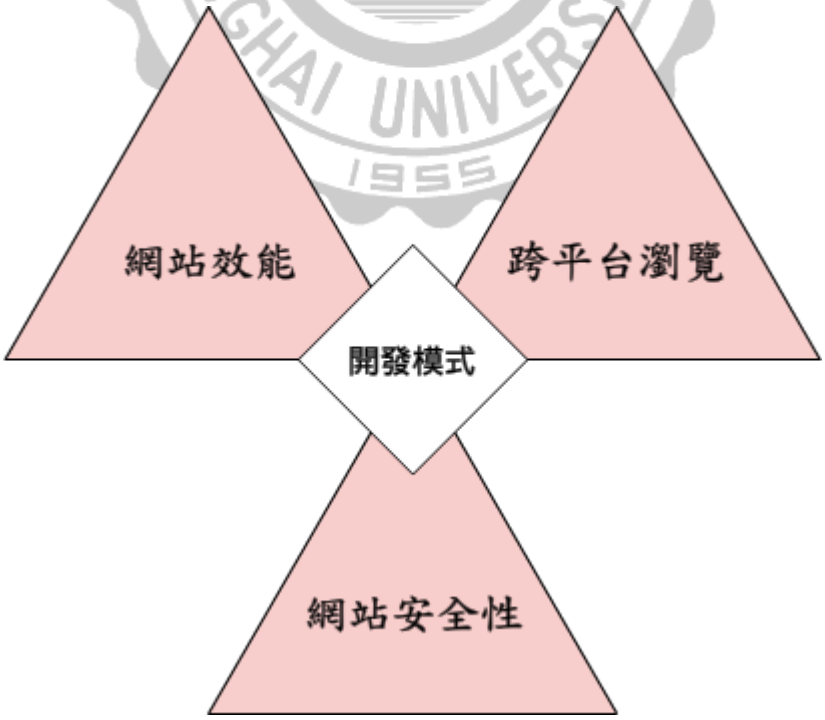


圖 3-1 開發模式之優勢較量模型圖

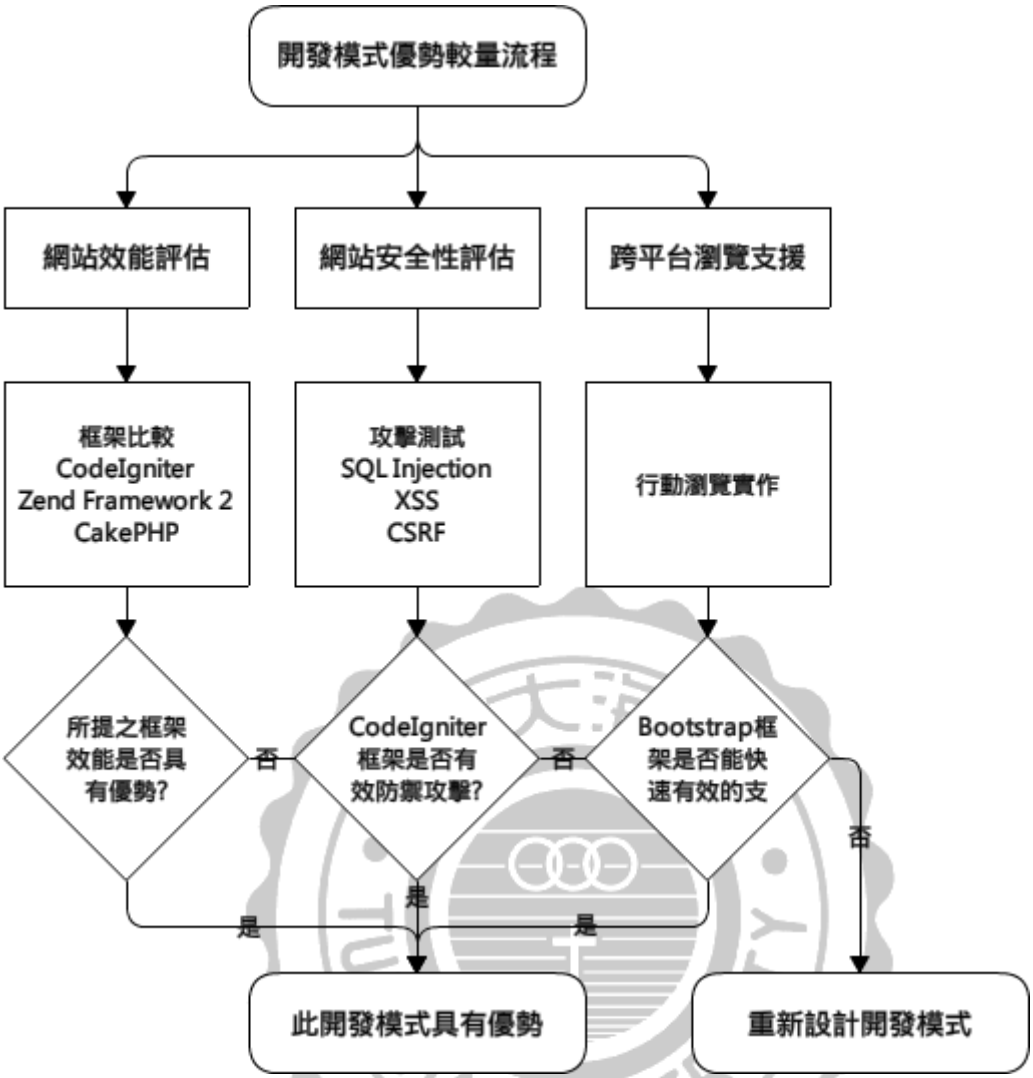


圖 3-2 開發模式之優勢較量流程圖

第一節 網站效能

網站效能一向都是使用者非常在意的一件事情，一個網站做的再怎麼好，如果需要花費相當長的一段時間讀取，那想必使用者會相當的不耐煩，因而放棄瀏覽此網站，所以網站效能的測量是必須的。

本研究將採用 Benchmarking 的方式進行，所謂的 Benchmarking 就是使用量測方法來計算效能的意思，在此本研究將會以被業界廣為使用的 Zend Framework 和 CakePHP 為對照組，CodeIgniter 則為實驗組，透過四項檢測方式來做評斷分別為：

1. total, onload, DOMContentLoaded
2. microtime(), memory_get_usage()

3. Apache Benchmark

4. XHProf

壹、total, onload, DOMContentLoaded

第一項檢測方式主要使用 Google Chrome 所提供的開發人員工具作為量測工具，其中的 Network Panel 具有測量並統計 total, onload, DOMContentLoaded 三項時間參數的功能，且網頁當中的所有元素，包含 CSS style、Script、Flash、Image 甚至是 subframe 當中的元素等等動態都會記錄在此處，其中根據出現的時間不同會產生 Timeline 提供使用者判別是哪個檔案導致網頁讀取延遲。

total 顧名思義就是網頁的總花費時間，從第一個檔案直到最後一個讀完之間的時間加總，當然過程中如果有動態的資料或圖片存取過程中的閒置和讀取時間也會列入計算。而 onload 則是在完成了 CSS style、Script、Image 及 subframe 等下載動作後結束並觸發 onload( )事件。DOMContentLoaded 則是在 DOM(Document Object Model）被完整的讀取跟解析之後就會結束時間計算，不會等待其他檔案。

貳、microtime(), memory_get_usage()

microtime(), memory_get_usage()為 php 的語法，microtime()執行後會回傳以毫秒為單位的 Unix Timestamp，而 Unix Timestamp 所代表的意義是從格林威治標準時間 1970-01-01 00:00:00 到現在總共所經過的秒數，藉由擺放在框架的起始與結尾的 function 處來計算出時間差，藉此得出各框架的反應時間，另外 memory_get_usage()指的就是當前的記憶體消耗量，也是藉由擺放在起始與結尾之處來體現差異。

參、Apache Benchmark

Apache Benchmark 顧名思義就是利用 Apache Server 來做量測工具，我們將會使用到一個叫做 ab 的指令，指令構成為：“ab -c 100 -n 30000”，參數 c 的意思是每秒鐘會對網站提出 100 個 request，參數 n 的意思則是將會建立 30000 個連線數來測試，藉此得出各種不同的框架每秒接受到這些 request 的時候能 reply 多少回應數。

肆、XHProf

XHProf 是一個函數層級的量測工具，意思就是可以將每個 function 所耗費的時間明確標示出來並且將會統計在一次的 load page 當中該框架執行了多少 function，本研究將會根據其提供的數據做為實驗結果的評斷依據。

第二節 網站安全性

網站安全性一向都是網站經營者最需要重視的問題之一，再小的漏洞都有可能造成無法挽救的後果，因此在撰寫網站的時候首當必須考慮到資訊安全，本次研究將藉由 OWASP（The Open Web Application Security Project）組織在 2013 年所提出的十大安全性議題中挑選下列三項去評估 CodeIgniter 是否對這些攻擊有相對應的防禦措施：

1. SQL Injection
2. Cross-Site Scripting
3. Cross-Site Request Forgery

在十大安全性議題當中挑選此三項攻擊手段的主要原因是其他項目如 Broken Authentication and Session Management、Security Misconfiguration 皆屬於偏網站管理員個人的安全意識及專業技術的部份，比較不容易利用實驗呈現，而 SQL Injection、Cross-site Scripting 及 Cross-Site Request Forgery 皆屬於第三方發動的攻擊，可藉由模擬攻擊行為去測試框架其安全性防禦機制是否有作用，因而做此選擇。

壹、SQL Injection

SQL Injection 翻譯為 SQL 指令植入式攻擊，其所針對的目標是未將使用者輸入的資訊做好過濾，就直接將資料置入 SQL 語法中做查詢，如此一來黑客只要利用某些對資料庫系統含有特定意義的符號或語法就可以直接讓其下達指令，進而促成了攻擊的產生，甚至最終導致企業財產的損失。

貳、Cross-Site Scripting

Cross-Site Scripting 為了與 Cascading Style Sheets 的縮寫 CSS 區分，將 Cross 改為發音相似的 X 替代，故縮寫為 XSS，翻譯為跨網站腳本攻擊。XSS 是一種攻擊網站應用程式漏洞的行為，起因於程式設計師撰寫了不夠嚴謹的程式，運行過程中沒有將使用者輸入的資訊做充分過濾，便有黑客藉此插入惡意程式碼於網站當中，當不知情的使用者訪問到該段惡意程式碼，就會遭受到攻擊。

參、Cross-Site Request Forgery

Cross-Site Request Forgery 縮寫為 CSRF，也可縮寫為 XSRF，翻譯為跨網站請求偽造。CSRF 是一種利用網站對合法使用者的信任，以合法使用者的身分發出偽造的請求，造成使用者在不知情的情況下執行一些行為，甚至是黑客指定的惡意行為，如：讓受害者發布夾帶惡意程式的內容擴散攻擊、受害者若是網管人員則讓其修改網站設定或變更密碼藉此入侵網站等等。

第三節 跨平台瀏覽

行動裝置如手機與平板電腦日漸普及，近年來更有「人手一機」的說法，因此行動裝置所帶來的市場更是不容小覷，想當然網站這個領域也深受其影響，多數中大型網站都支援行動版網頁，但仍有為數不少企業的網站不支援行動裝置瀏覽，就連國內知名的網路購物網站露天拍賣及 PCHome 購物都尚且不支援行動版網頁瀏覽，相較之下內地的淘寶與阿里巴巴都建立了簡明易操作的行動版網頁，更遑論那些火紅的社群網站了。現今這個時代，行動瀏覽購物儼然成為了一種趨勢更是一大商機，若企業仍故守舊不知變通，對其發展實在是一大打擊。因此，本研究會藉由評估 Bootstrap 對行動版網頁的支援及其相容性來證明該框架是有助於開發者的。

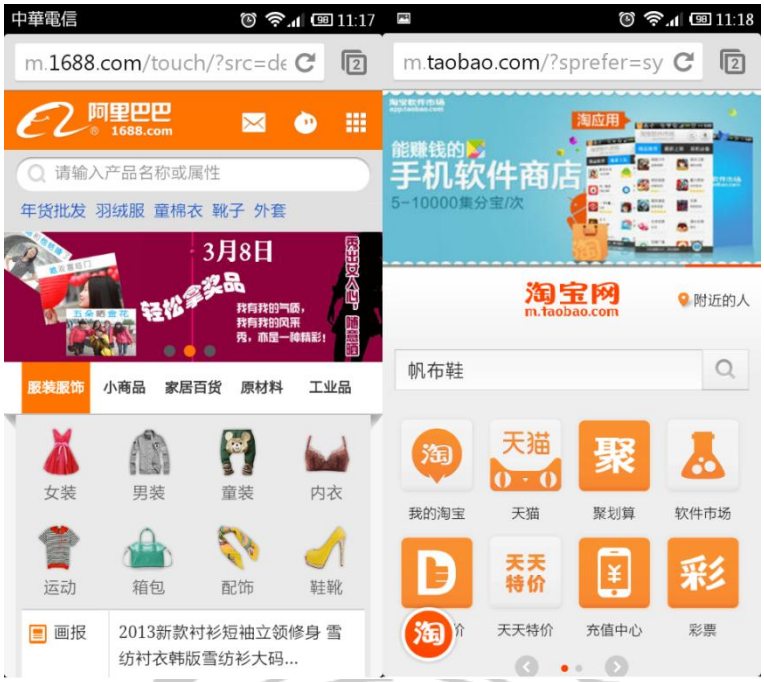


圖 3-3 左為阿里巴巴右為淘寶的行動版網頁

第四章 開發模式之優勢評估

表 4-1 系統環境與框架版本

系統環境	
OS	Windows 7 x64
CPU	AMD Athlon II X2 250 3.00GHz
RAM	8G
Browser	Google Chrome 25.0.1364
框架版本	
對照組	CakePHP 2.3.1、Zend Framework 2
實驗組	CodeIgniter 2.1.3

第一節 網站效能

本研究提出了結合 CodeIgniter 與 Bootstrap 框架的開發模式，並以網站效能、網站安全性與跨平台瀏覽三項指標做為評估依據，首先針對網站效能提出了下列四項檢測方式：

1. total, onload, DOMContentLoaded
2. microtime(), memory_get_usage()
3. Apache Benchmark
4. XHProf

壹、total, onload, DOMContentLoaded

第一項檢測使用 Google 開發人員工具中的 Network Panel 做網站效能的檢測，分別統計出下列三項數據：

1. total
2. onload
3. DOMContentLoaded

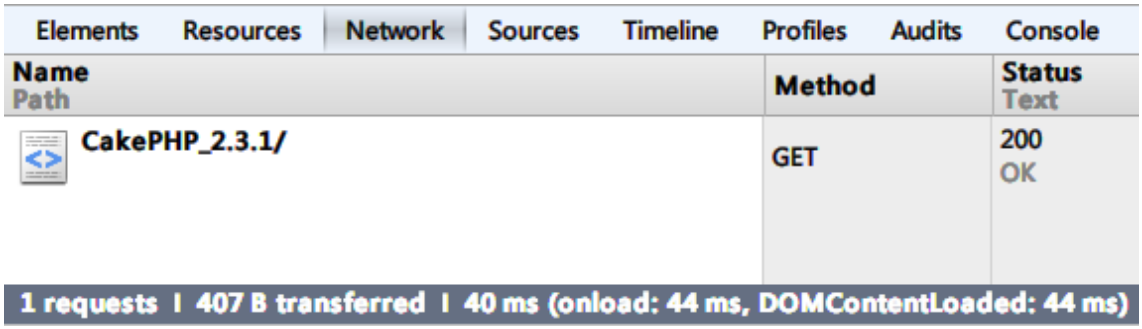


圖 4-1 CakePHP 檢測結果

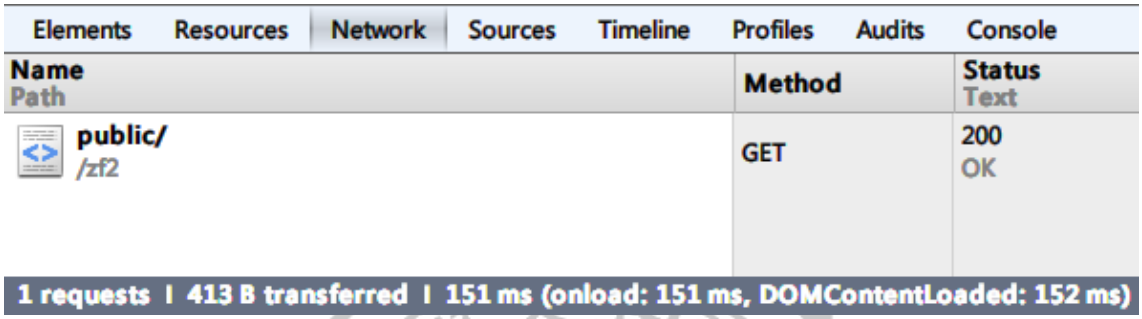


圖 4-2 Zend Framework 檢測結果

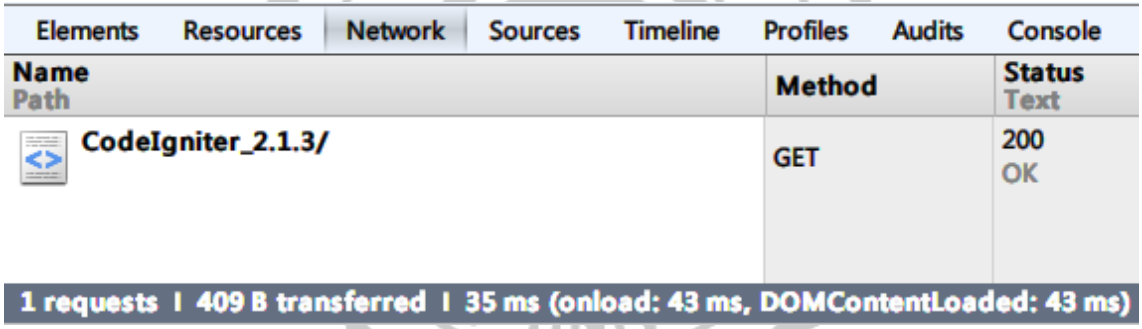


圖 4-3 CodeIgniter 檢測結果

表 4-2 網頁耗時統計表格（單位：毫秒）

	CakePHP	Zend Framework	CodeIgniter
total	40	151	35
onload	44	151	43
DOMContentLoaded	44	152	43

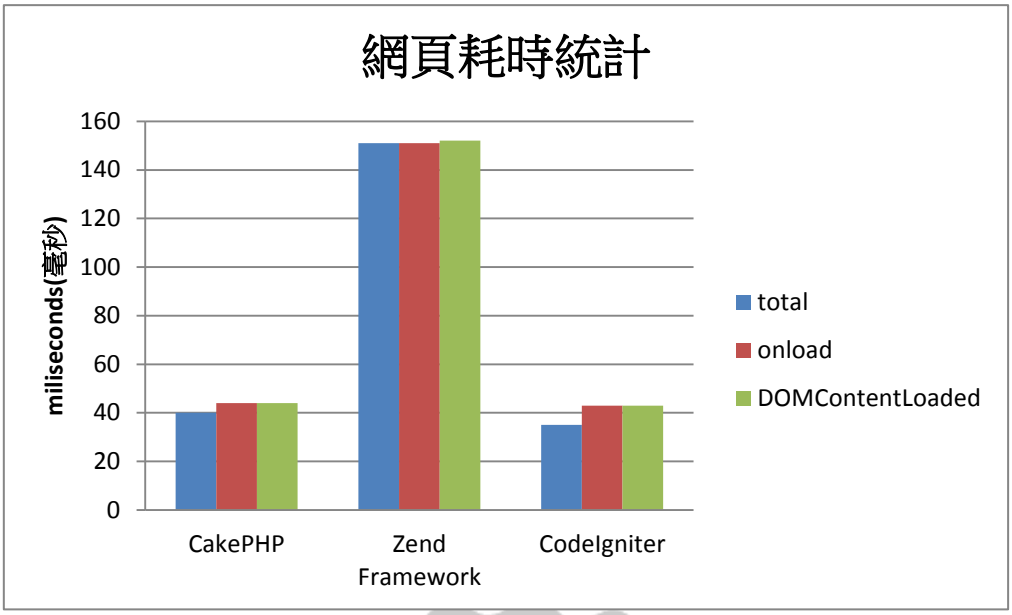


圖 4-4 網頁耗時統計圖表

從 total 來看，CodeIgniter 僅比 CakePHP 快了約 5 毫秒左右差異不大，但與 Zend Framework 相比差了將近五倍之多，另外在 onload 和 DOMContentLoaded 的部分 CodeIgniter 依舊與 CakePHP 不相上下，但是基本上都大幅領先 Zend Framework，因此在第一項檢測當中實驗組得出的結果不全是勝過對照組的，會根據框架的不同而產生持平的狀態。

貳、microtime(), memory_get_usage()

第二項檢測將利用 Chrome 瀏覽器分別執行三種框架的 Hello World 網頁取得其網頁產生時間與記憶體消耗總量。

圖 4-5 CakePHP 的執行畫面

圖 4-6 Zend Framework 的執行畫面

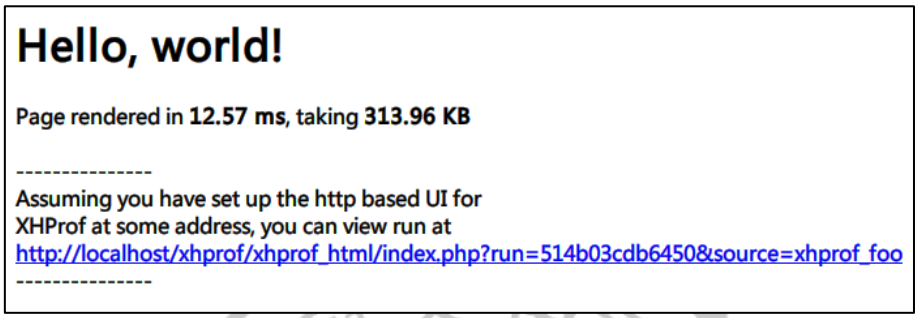


圖 4-7 CodeIgniter 的執行畫面

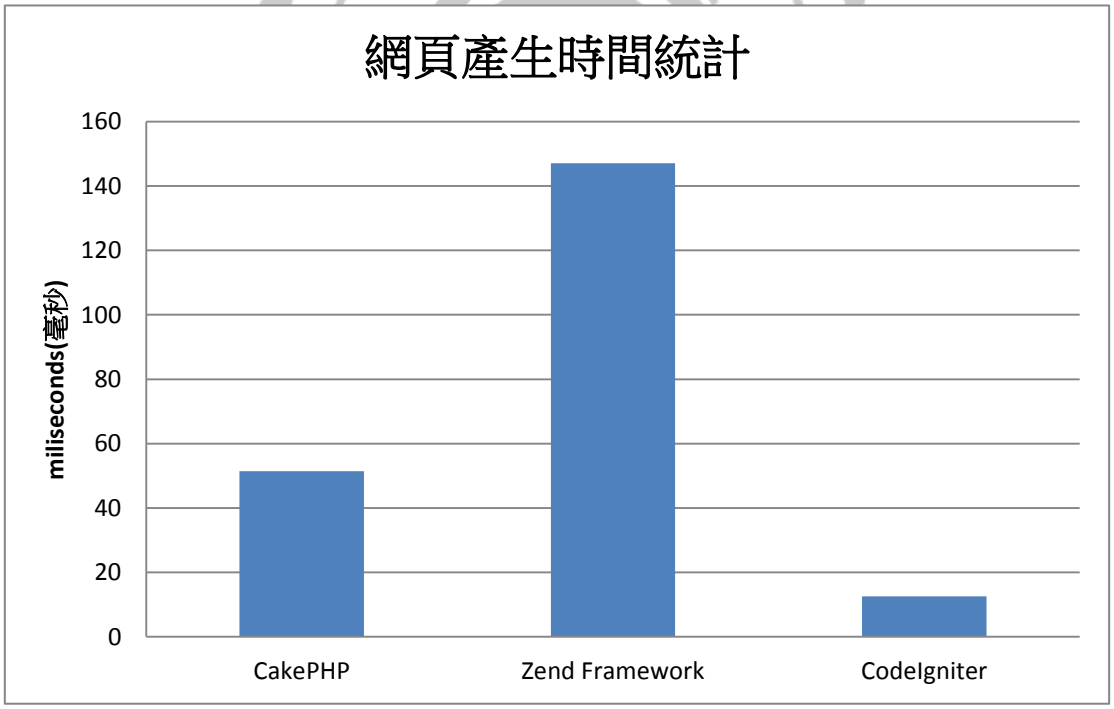


圖 4-8 網頁產生時間統計圖表

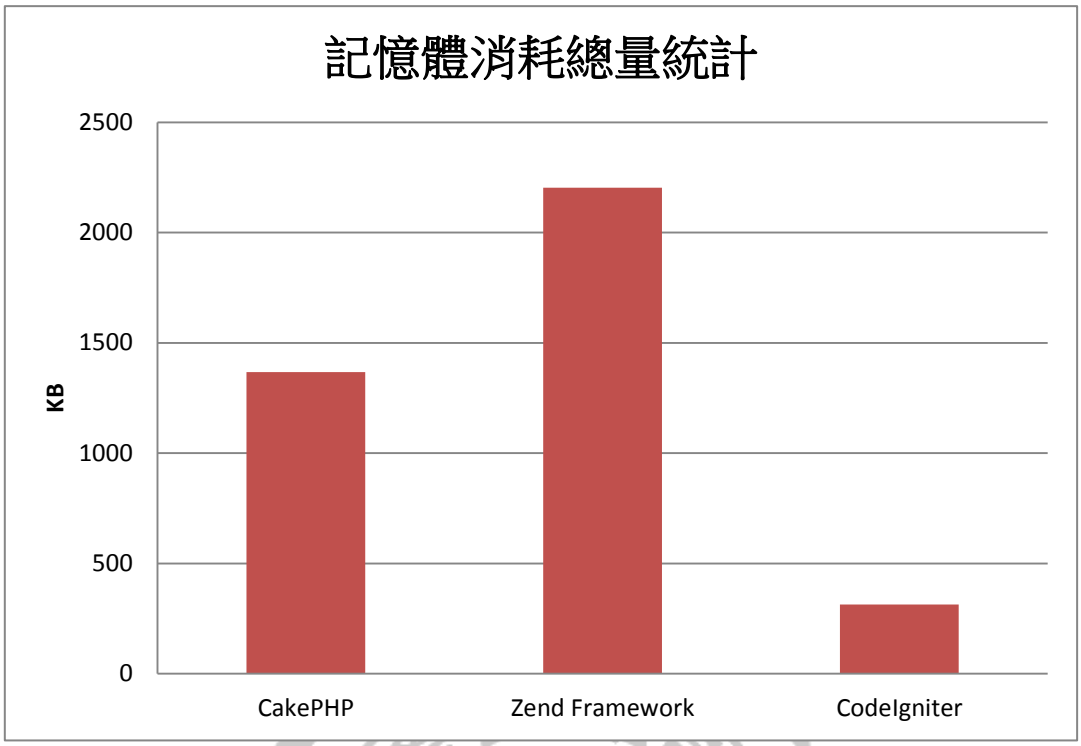


圖 4-9 記憶體消耗總量統計圖表

CodeIgniter 快了 CakePHP 將近四倍的時間，比起 Zend Framework 更是快了十倍之多，至於記憶體消耗總量，CodeIgniter 的記憶體消耗也是相當少的 313KB，而 CakePHP 跟 Zend Framework 基本上都使用了超過 1MB 的記憶體，因此上述數據可證明實驗組在第二項檢測上的結果是遠勝於對照組。

參、Apache Benchmark

第三項檢測使用的工具是 Apache Benchmark，此工具大多已內建在 Apache Server 當中，其用途主要是測試提供服務的目標主機能在每秒鐘處理多少個 request，以下是本次實驗所使用的指令：“ab –c 100 –n 30000”意思是將會對目標建立 30000 個連線數並且每秒鐘發送 100 個 request。

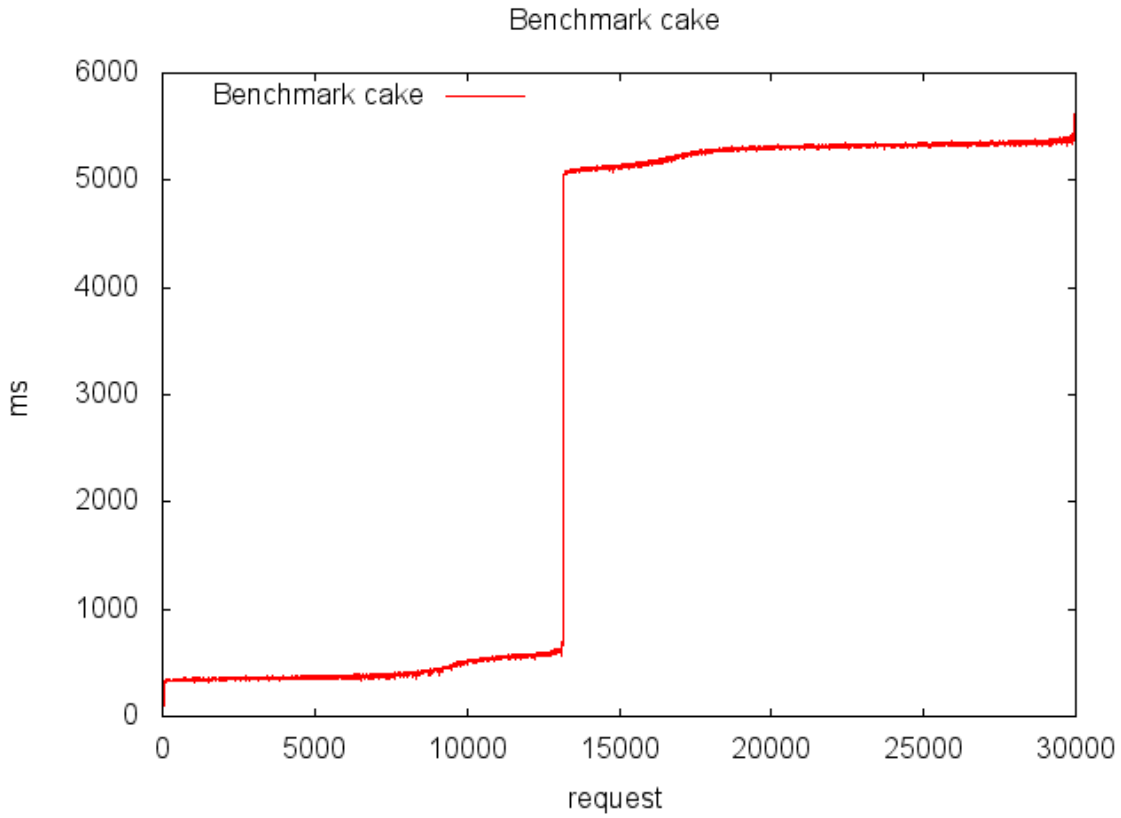


圖 4-10 Apache Benchmarking （CakePHP）

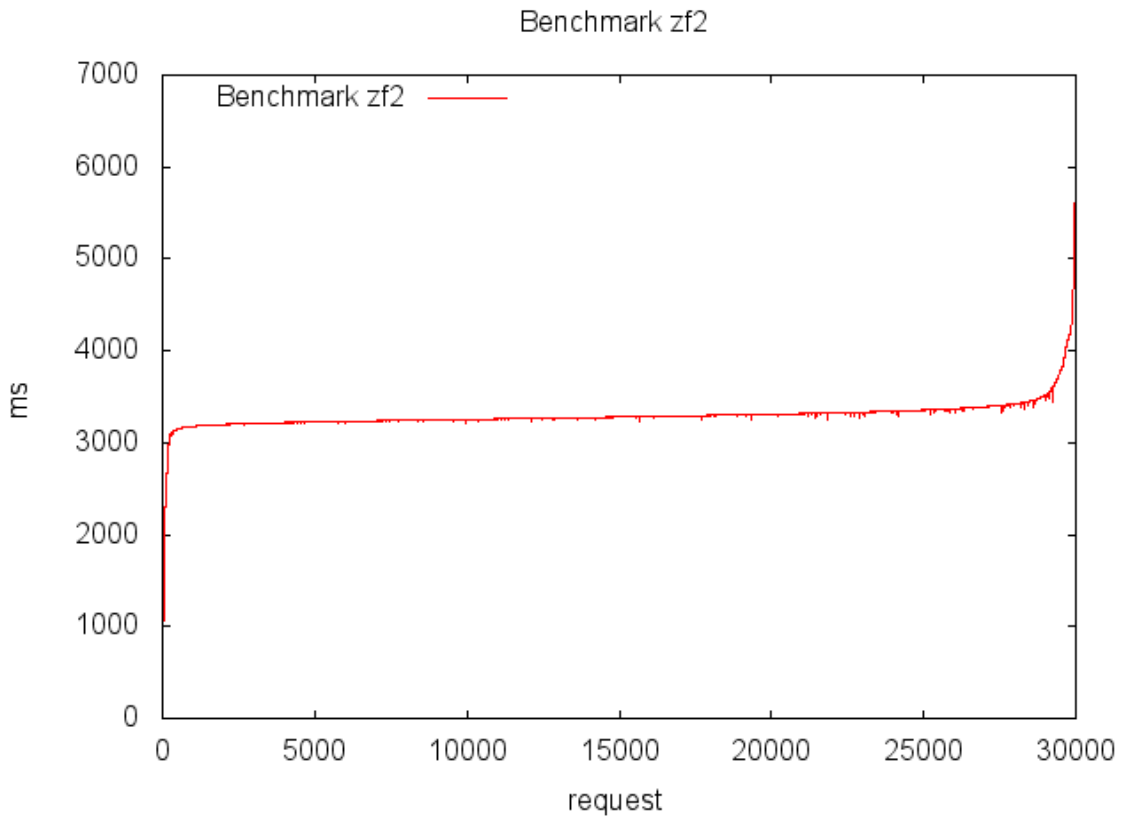


圖 4-11 Apache Benchmarking（Zend Framework 2）

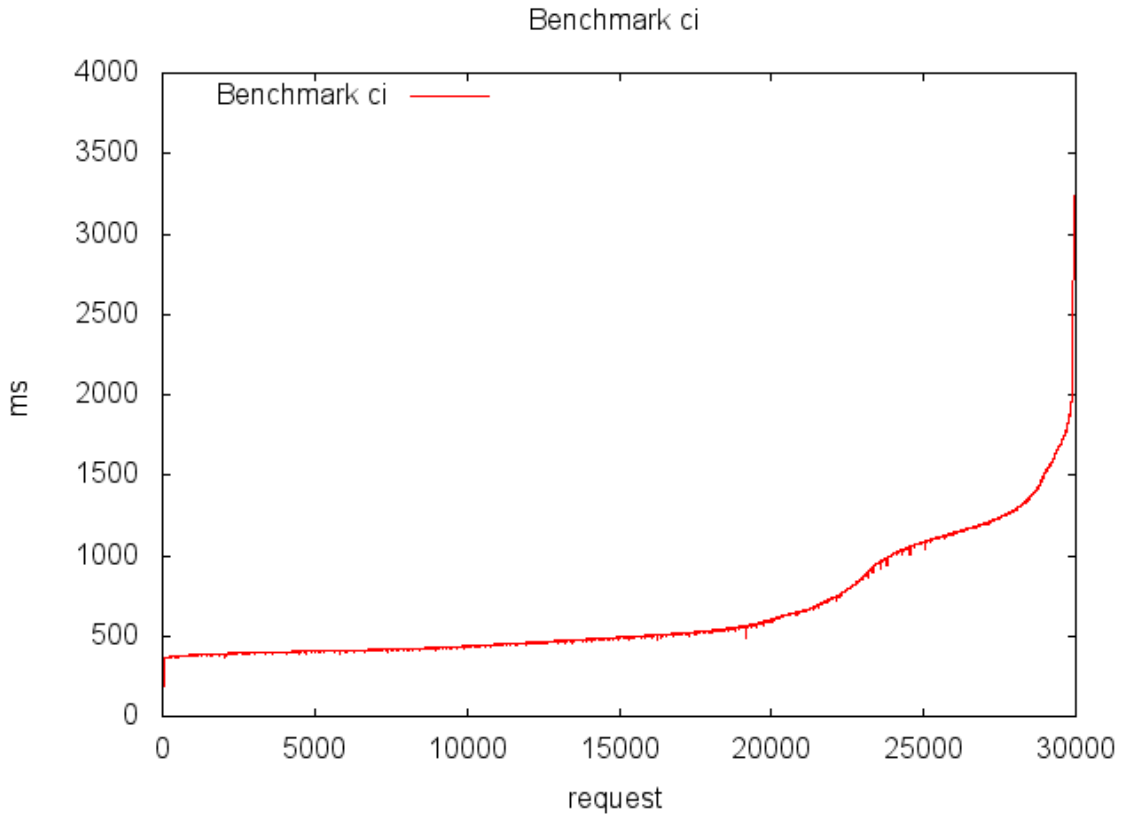


圖 4-12 Apache Benchmarking（CodeIgniter）

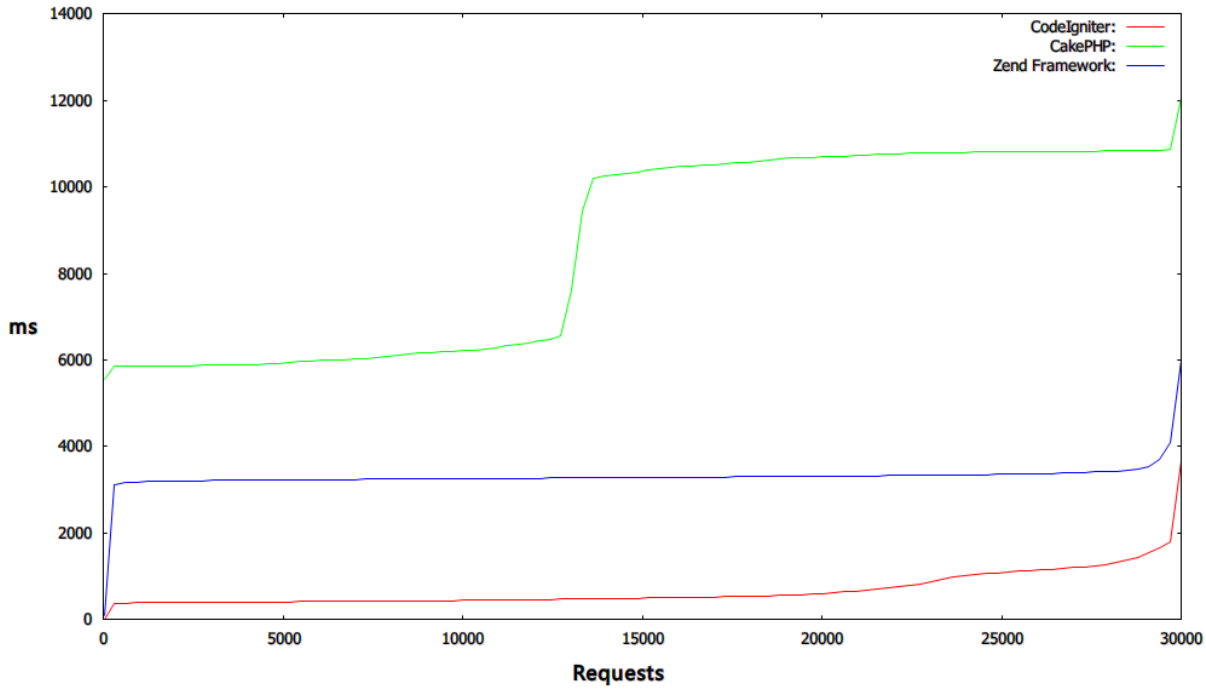


圖 4-13 使用 GNUPlot 所繪製出的統計折線圖

藉由折線圖可以看出 CakePHP 在處理超過 15000 requests 的時候回應時間突然大幅攀升，由此可判斷 CakePHP 在面臨到超過一定數量的 requests 時網站會有

效能降低的可能，而 Zend Framework 的折線圖相對地較為平順，代表其在處理大量 requests 時，網站效能不會因此降低。至於 CodeIgniter 的回應時間基本上都在 3000 毫秒以下，比起對照組的結果是相對優異的。

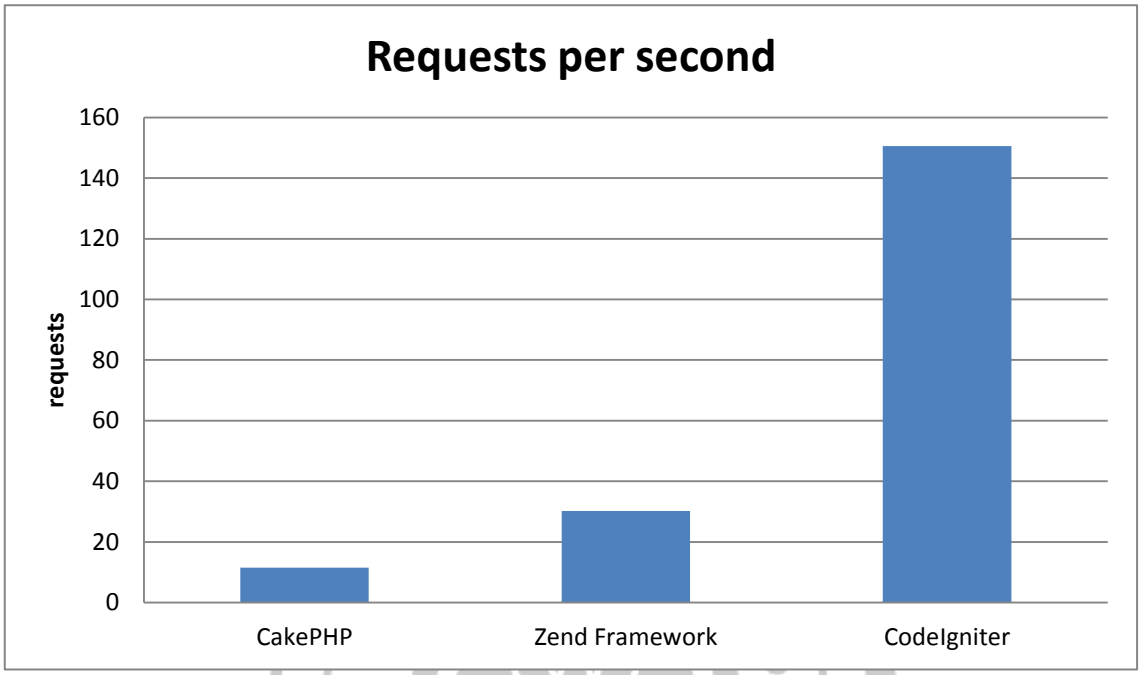


圖 4-14 每秒回應數量（值愈大愈好）

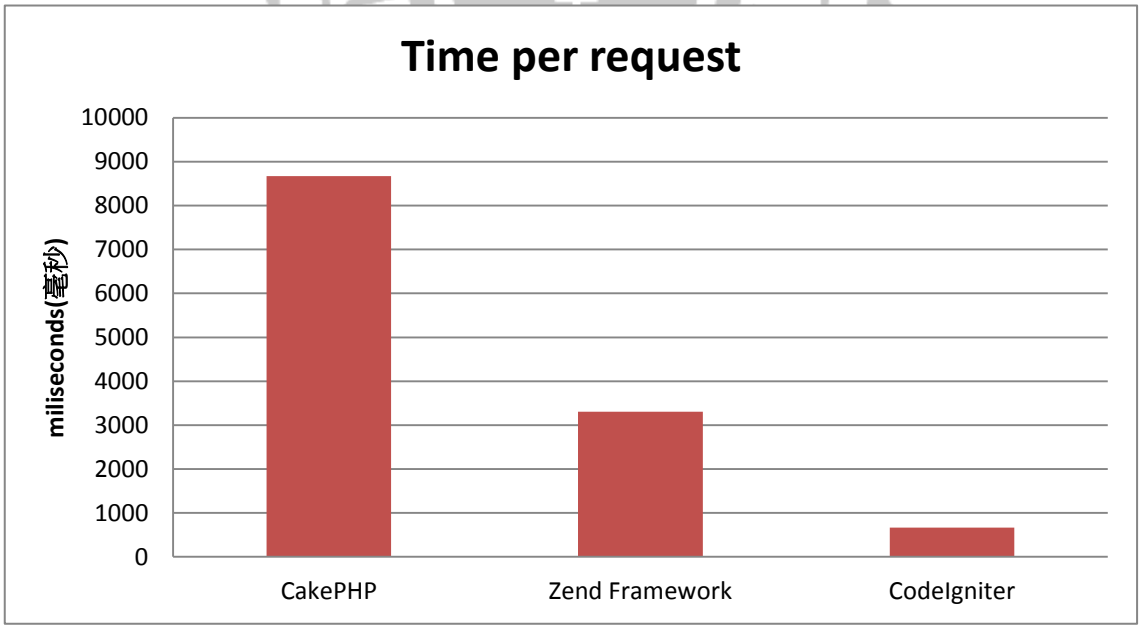


圖 4-15 每個回應所耗費時間（值愈小愈好）

表 4-3 實測數據統計表

	CakePHP	Zend Framework	CodeIgniter
Requests/Second	11.53	30.23	150.55
Microseconds/Requests	8670.439	3307.949	664.238

根據實測 Console 給出的數值（如上表所示），實驗組得出的結果優於對照組約五到十倍有餘，由此可得知實驗組在處理 request 的回應時間與每單位時間可處理的 requests 數量是優於對照組的。

肆、XHProf

第四項檢測將利用 XHProf 工具針對對照組與實驗組三種框架做量測，其量測方式為從框架執行的起始點開始監測，至網頁產生後結算所有 function 的呼叫流程以及總花費時間，主要會根據下列三項數據做為本次檢測結果的判斷依據：

1. CPU 耗費於程式單元呼叫之時間
2. 最大記憶體使用量
3. 函數呼叫總數

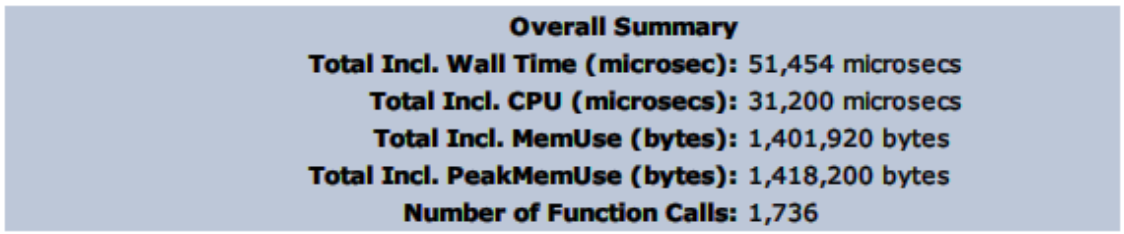


圖 4-16 CakePHP 於 XHProf 的統計結果

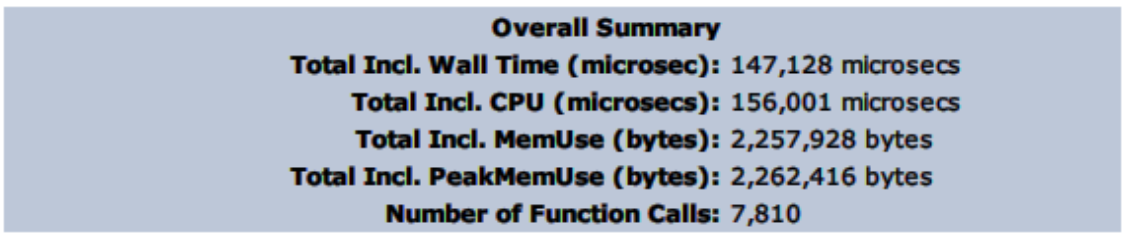


圖 4-17 Zend Framework 於 XHProf 的統計結果

Overall Summary	
Total Incl. Wall Time (microsec):	12,623 microsecs
Total Incl. CPU (microsecs):	15,600 microsecs
Total Incl. MemUse (bytes):	323,672 bytes
Total Incl. PeakMemUse (bytes):	357,512 bytes
Number of Function Calls:	473

圖 4-18 CodeIgniter 於 XHProf 的統計結果

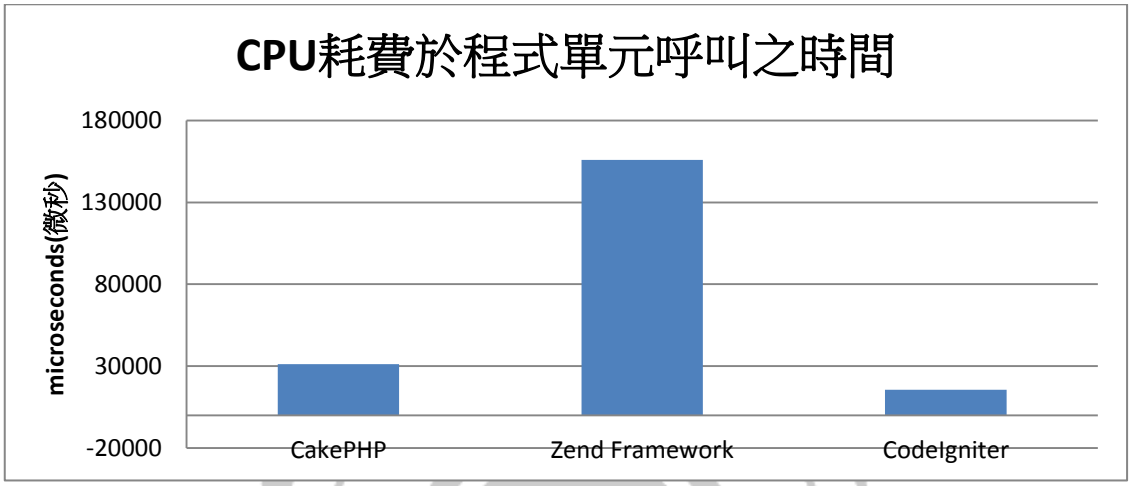


圖 4-19 CPU 耗費於程式單元呼叫之時間統計圖表

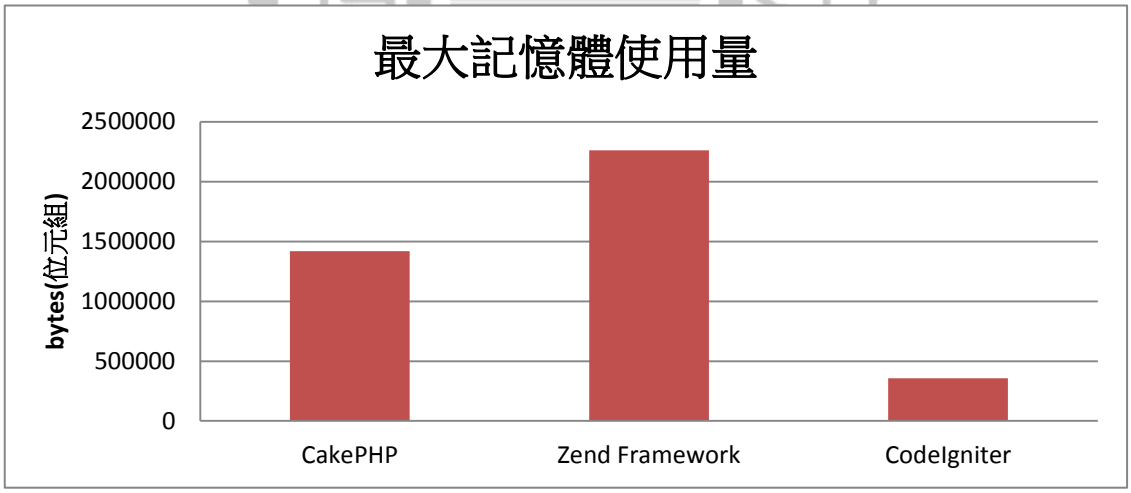


圖 4-20 最大記憶體使用量統計圖表

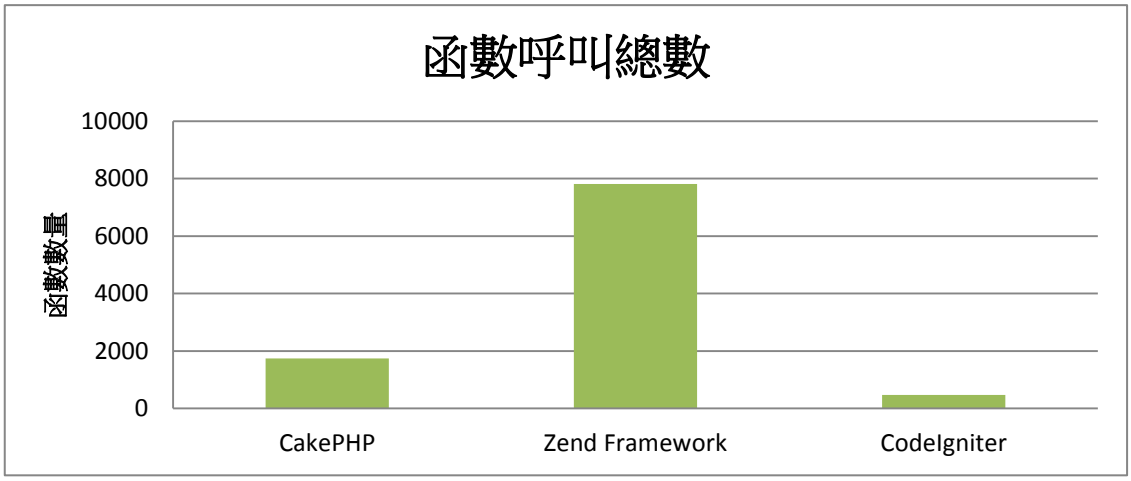


圖 4-21 函數呼叫總數統計圖表

由上述三張圖表可得知 CodeIgniter 無論是在 CPU 耗費於程式單元呼叫之時間、記憶體使用或者是函數呼叫數量所呈現出來的數據都遠比對照組框架優異上許多，因此得出實驗組於第四項檢測的表現比對照組高出兩倍至十倍不等的結論。

第二節 網站安全性

網站安全性三項檢測項目：SQL Injection、Cross-Site Scripting 及 Cross-Site Request Forgery。本研究以 CodeIgniter 框架建置了攻擊測試網站，僅供學術研究用途。

壹、SQL Injection 防禦

情境：黑客利用 SQL Injection 不需要密碼即可登入網站。

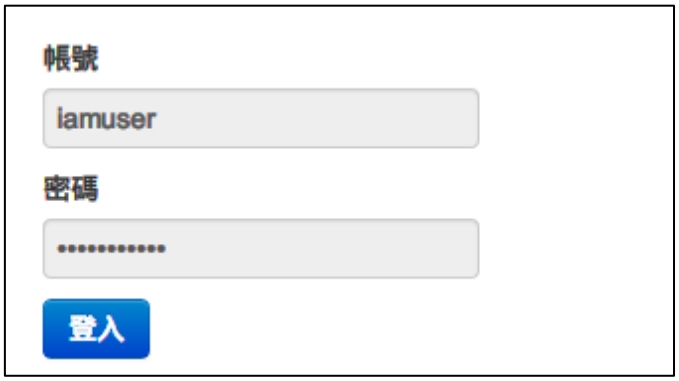


圖 4-22 黑客填入帳號及摻有不法字元的密碼

程式

```
$username = $this->db->escape($this->input->post('user'));
$password = $this->db->escape($this->input->post('pwd'));
$sql = "select count(*) from users where username = '".$usernam
e."' and password = '".$password."' limit 1";
```

圖 4-23 此為處理使用者登入的程式

程式

```
$sql = "select count(*) from users where username = 'ia
muser' and password = '\' or \'1\'=\'1' limit 1";
```

圖 4-24 使用了跳脫字元法（escape）將不法字元轉換成普通字串

程式

```
$this->db->query("select count(*) from users where user
name = 'iamuser' and password = '\' or \'1\'=\'1' limit
 1");
```

圖 4-25 將安全的 SQL 語法送出查詢

程式

```
$this->db->query("select count(*) from users where user
name = 'iamuser' and password = '\' or \'1\'=\'1' limit
 1");
```

輸出

```
0
```

圖 4-26 回傳值是 0 黑客登入失敗 成功防禦

貳、Cross-Site Scripting 防禦

情境：黑客利用 Cross-Site Scripting 在留言當中插入影響正常運作的惡意 Javascript 語法。

圖 4-27 黑客填入姓名 E-Mail 及含不法字元的留言

程式

```
$name = $this->db->escape($this->input->post('name', TRUE));
$email = $this->db->escape($this->input->post('email', TRUE));
$content = $this->db->escape($this->input->post('content', TRUE
));
$sql = "insert into messages(name, email, content) values('".$n
ame."', '".$email."', '".$content."')";
```

圖 4-28 只要將 POST 方法中的第二個參數設定為 TRUE 即會過濾 XSS

程式

```
$sql = "insert into messages(name, email, content) values('iamu
ser', 'iam@user.com', 'I say...[removed]alert('Hello World!');[
removed]')";
```

圖 4-29 經過 XSS 過濾的 SQL 語法

程式

```
$this->db->query("insert into messages(name, email, content) va
lues('iamuser', 'iam@user.com', 'I say...[removed]alert('Hello
World!');[removed]')");
```

圖 4-30 送至資料庫儲存

輸出

新留言

姓名：iamuser
E-Mail：iam@user.com
留言內容：I say...[removed]alert('Hello World!');[removed]

圖 4-31 沒有跳出警告（alert）視窗 成功防禦

參、Cross-Site Request Forgery 防禦

程式

```
application/config/config.php:296-299
$config['csrf_protection'] = TRUE;
$config['csrf_token_name'] = 'csrf_token';
$config['csrf_cookie_name'] = 'csrf_cookie';
$config['csrf_expire'] = 7200;
```

圖 4-32 開啟內建的 CSRF 防禦機制來抵禦攻擊

程式

```
$input = array(
 'name' => 'msgid',
 'value' => 1,
 'style' => 'display: none;'
);
$submit = array(
 'name' => 'remove',
 'value' => '刪除',
 'class' => 'btn btn-danger'
);
echo form_open('/message/remove').
  form_input($input).
  form_submit($submit).
  form_close();
```

圖 4-33 不要使用 GET 做重要的處理 利用 form_open 產生表單並使用 POST

程式

```
<form action='/message/remove' method='post'>
  <div style='display:none'>
    <input type='hidden' name='csrf_token' value='**********' />
  </div>
  <input type='text' name='msgid' value='1' style='display: none;' />
  <input type='submit' name='remove' value='刪除' class='btn btn-danger' />
</form>
```

圖 4-34 產生的表單會帶有防禦 CSRF 的 token 值

程式

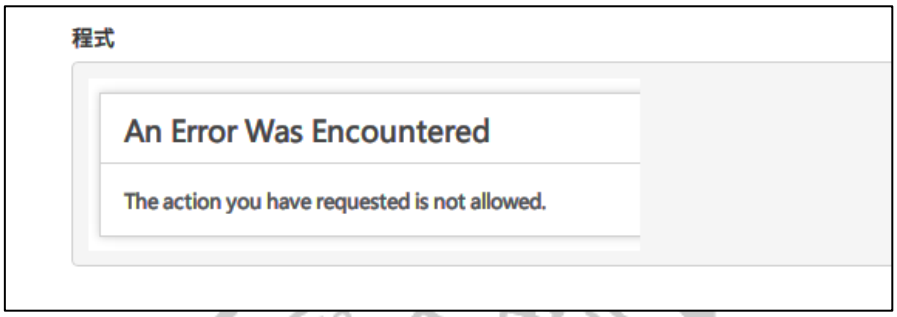


圖 4-35 黑客既無法利用 GET 亦無法利用偽造表單發送刪除請求 防禦成功

第三節 跨平台瀏覽

Bootstrap 框架在版面配置部分提供了 Responsive Design 的功能，只要將下方語法寫入網頁中的<head>標籤之間，網站的所有樣式就會依照行動裝置的大小來調整寬高。

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="assets/css/bootstrap-responsive.css" rel="stylesheet">
```

圖 4-36 啟用 Bootstrap 框架的自適性設計功能

Label	Layout width
Large display	1200px and up
Default	980px and up
Portrait tablets	768px and above
Phones to tablets	767px and below
Phones	480px and below

圖 4-37 支援的裝置類型分成五種

圖 4-38 產學合作網站於 24’液晶螢幕之瀏覽畫面

圖 4-39 產學合作網站於行動裝置之瀏覽畫面

第五章 結論

我們於研究當中發現基於 MVC 架構的開發框架，其架構清晰且分工明確的特性，減少了網站開發人員花費在溝通上的資源浪費，另外該框架無論是在網站執行效能的表現還是在資訊安全上有其因應的對策都體現出相較於舊式網站來得優異，而因此我們提出了下列三項指標做為驗證依據：

1. 網站效能
2. 網站安全性
3. 跨平台瀏覽

經由實驗所得出的數據可得知本研究提出結合 CodeIgniter 與 Twitter Bootstrap 的開發模式在先前所提到的網站效能、網站安全性及跨平台瀏覽此三項指標當中都取得不錯的結果，而且利用產學合作的方式將開發模式導入，證實本研究在實際運用上也有了些許經驗，並藉由此開發模式將企業網站做全面且實質性的提升。

本研究仍有許多地方可以改進，在後續研究的部分，如：「台灣當前知名企業的網站是使用何種開發模式為主」、「使用 MVC 架構的企業網站佔台灣市面的比例」或者「多數企業主對於優化企業網站的看法」等研究題目都是可以成為推廣此開發模式的強大助力，也期望未來能夠更加地努力，讓更多開發者能夠接納此開發模式。

參考文獻

[1] E. Athanasopoulos, V. Pappas, A. Krithinakis, S. Ligouras, E.P. Markatos, and T. Karagiannis（2010）, xJS: Practical XSS Prevention for Web Application Development. In Proceedings of the 1st USENIX WebApps Conference, Boston, US.

[2] 黃石欽（2002），《Model-View-Controller（MVC）架構之整合設計與實務應用》，私立中原大學應用數學研究所碩士論文。

[3] 蘇冠緯（2004），《以 MVC 架構為基礎之開發者入口網站設計與建置》，中原大學電子工程研究所。

[4] 黃世豪（2010），《一個基於 MVC 架構的社交網路服務應用程式開發框架之設計與實作－以 Facebook 應用程式為例》，國立交通大學資訊管理研究所碩士論文。

[5] EllisLab〈CodeIgniter〉，http://ellislab.com/codeigniter，2013/03/25

[6] Mark Otto〈Twitter Bootstrap〉，http://twitter.github.com/bootstrap/，2013/03/25

[7] Chrome DevTools〈Network Panel Overview〉，https://developers.google.com/chrome-developer-tools/docs/network#network_panel_overview，2013/04/23

[8] avnpc. com〈於 Windows 環境下搭建 Zend Framework 2〉，http://avnpc.com/pages/zend-framework-2-installation-for-windows，2013/03/25

[9] The PHP Group〈XHProf〉，http://pecl.php.net/package/xhprof，2013/03/25

[10] Coding Life〈於 Windows 環境下建置 XHProf〉，http://codinglife.sinaapp.com/?tag=xhprof-windows-%E5%AE%89%E8%A3%85，2013/03/25

[11] Ruilog〈PHP Framework MVC Benchmark〉，http://www.ruilog.com/blog/view/b6f0e42cf705.html，2013/03/25

[12] Eryx〈PHP Framework Benchmark〉，
https://github.com/eryx/php-framework-benchmark，2013/03/25

[13] The Apache Software Foundation〈Apache HTTP Server benchmarking tool〉，
http://httpd.apache.org/docs/2.2/programs/ab.html，2013/03/25

[14] GNUPlot〈GNUPlot〉，http://www.gnuplot.info/，2013/03/25

[15] KutuKupret〈Graphing ApacheBench Results Using GnuPlot〉，
http://www.kutukupret.com/2011/05/10/graphing-apachebench-results-using-gnuplot/，2013/03/25

[16] Stack Overflow〈How ApacheBench can follow redirection〉，
http://stackoverflow.com/questions/12209612/how-apachebench-can-follow-redirection，2013/04/23

[17] OWASP〈Top 10 2013〉，https://www.owasp.org/index.php/Top_10_2013-T10

[18] Open Foundry〈Web Security 網站安全基礎篇（二）〉，
http://www.openfoundry.org/en/tech-column/2354-web-security-，2013/04/23

[19] Kainy〈NavigationTiming〉，http://github.kainy.cn/NavigationTiming，2013/04/23

東海大學
資訊管理研究所
碩士論文

結合 PHP 與 CSS 框架之網站建置最佳化 〈101〉

研究生：白翰霖 撰