

บทที่ 4

เทคโนโลยีที่ใช้ในการศึกษา

ระบบการจัดการฐานข้อมูลนักวิจัยอุทยานวิทยาศาสตร์มหาวิทยาลัยแม่โจ้ (The Researchers Database Management System for Science Park Maejo University) ได้เลือกเทคโนโลยีในการศึกษาและพัฒนาระบบ ดังนี้ Codeigniter framework

4.1 Codeigniter framework

4.1.1 Codeigniter คืออะไร

Codeigniter คือ framework ที่ถูกพัฒนาขึ้นด้วย ภาษา PHP สำหรับ Codeigniter Framework เป็นเครื่องมือสำหรับช่วยพัฒนาเว็บไซต์และเว็บแอปพลิเคชัน โดยมีโครงสร้างการพัฒนาโปรแกรมอย่างเป็นระบบ และรวมคำสั่งต่างๆ ที่จำเป็นต่อการพัฒนาเว็บไซต์ด้วย PHP ไว้ภายใน Codeigniter พัฒนาโดย Rick Ellis ซึ่งมีลิขสิทธิ์เป็น Open Source สามารถดาวน์โหลดและใช้ได้ฟรี ถูกพัฒนาขึ้นตั้งแต่ปี 2006 และยังมีการพัฒนามาเรื่อยๆ จนถึงปัจจุบัน

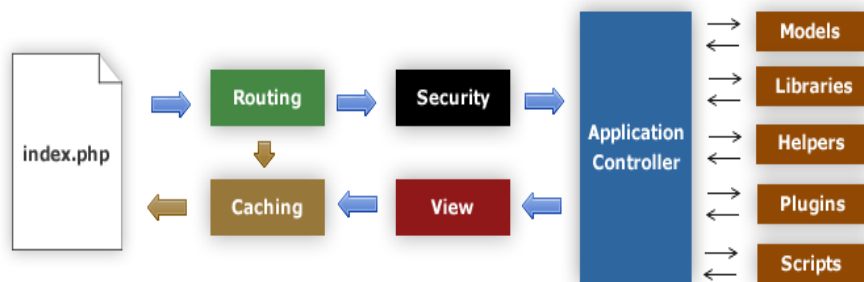
4.1.2 โครงสร้างของ Codeigniter Structure

Codeigniter รับรองการเขียนแบบ Model-View-Controller (MVC) เป็นการแยกส่วนประมวลผลออกจากส่วนแสดงผล ทำให้โครงสร้างมีความปลอดภัยมากยิ่งขึ้น และง่ายต่อการพัฒนาโปรแกรม โค้ดที่ได้มีความเป็นระเบียบและง่ายต่อการนำไปแก้ไข Codeigniter ยังออกแบบให้สามารถติดต่อกับฐานข้อมูลชนิดต่างๆ ได้ง่ายอีกด้วย เช่น mysql, postgres, sqlserver หรือเจ้าพ่อยักษ์ใหญ่อย่าง Oracle เป็นต้น หากเรียกใช้ฐานข้อมูลถูกต้องตามหลักของ database library แล้ว ผู้ใช้สามารถเปลี่ยนชนิดของฐานข้อมูลได้ เพียงแค่แก้ค่าคอนฟิกเพียงไม่กี่ที่เท่านั้น ส่วนระบบ url ของระบบ codeigniter สามารถสร้างได้สะอาดและเป็นมิตรกับกลไกการค้นหา (Search - Engine) ทำให้สามารถนำไปพัฒนาได้ทั้ง เว็บไซต์ และเว็บแอปพลิเคชัน ไฟล์ภายใน Codeigniter ประกอบด้วย โครงสร้างหลักๆ คือ application, system, index.php ระบบภายในสามารถแยกโปรเจกต์ออกเป็นหลายๆ application ได้

4.1.3 โครงสร้างไฟล์ภายในโฟลเดอร์ application ที่ต้องใช้งาน

1. cache : เป็นส่วนเก็บข้อมูลที่ประมวลผลแล้ว เพื่อการเรียกใช้ครั้งต่อไป
2. config : ใช้ตั้งค่าระบบต่างๆ ภายใน application
3. controllers : เป็นส่วนรวมโค้ดการประมวลผล
4. core : -
5. helpers : เป็นการเขียนฟังก์ชันในระบบ
6. hooks : ใช้สำหรับสร้างไฟล์เพื่อเข้าไป เพิ่ม , แก้ไข Class ต่างๆ ในระบบ
7. language : เป็นที่เก็บไฟล์ภาษาต่างๆ สำหรับผู้ที่ต้องการทำเว็บหลายภาษา

8. **libraries** : เป็นที่เก็บไฟล์ Class ที่คุณเขียนขึ้นเอง หากต้องการเรียกใช้ Class อื่นๆ ต้องเรียก `get_instance` ก่อน
9. **logs** : เป็นที่เก็บไฟล์บันทึกประวัติการทำงานภายใน application
10. **models** : คล้ายๆ libraries แต่ models จะสามารถเรียกใช้ libraries ที่คุณโหลดเข้ามาในระบบได้เลย
11. **third_party** : -
12. **views** : ส่วนนี้เป็นการแสดงผล ส่วนใหญ่จะเขียนโค้ด html ที่นี่ แต่นามสกุลไฟล์ต้องเป็น `.php` เท่านั้น



รูปที่ 4.1 เทคโนโลยีที่ใช้ในการพัฒนาโปรแกรม

4.1.4 Model-View-Controller MVC คืออะไร

โครงสร้างตามหลักการระบบจะแยกส่วนประมวลผล ออกจากส่วนแสดงผล แต่ในทางปฏิบัติ จะสามารถเขียนโค้ด PHP ลงในส่วนแสดงผลได้เช่นกัน

1. **Models** : อธิบายโครงสร้างข้อมูล แบบฉบับของคลาส แบบจำลองจะมีฟังก์ชันที่ช่วยให้คุณรับ , ใส่ , ปรับปรุงข้อมูลของคุณในฐานข้อมูล
2. **View** : คือข้อมูลที่ถูกแสดงให้กับผู้ใช้ View ส่วนใหญ่จะเป็นเว็บเพจ แต่ใน CodeIgniter , view สามารถเป็นเพียงบางส่วนของหน้าได้ เช่น ส่วนหัว ส่วนท้าย มันสามารถจะเป็นหน้า RSS หรือเป็นอย่างอื่นได้ที่เป็น "เพจ"
3. **Controller** : ถูกใช้เป็น สื่อกลาง ระหว่าง Model , View และแหล่งที่มาอื่นๆที่ต้องการในการ ทำงาน HTTP request และการสร้างเว็บเพจ

4.1.5 ขั้นตอนการทำงานของ CodeIgniter

1. **index.php** : เป็นตัวควบคุมส่วนหน้า, สร้างทรัพยากรพื้นฐานที่ต้องการในการรัน CodeIgniter
2. **Routing** : ตัว Router ทำการตรวจสอบ HTTP request กำหนดว่าควรทำอะไร
3. **Caching** : ถ้ามีไฟล์แคชอยู่ ตัวมันจะถูกส่งกลับทันทีไปยังบราวเซอร์ โดยไม่ผ่านการทำงานปกติของระบบ

4. Security : ก่อนที่จะโหลดตัวควบคุมของแอปพลิเคชัน (Application Controller) HTTP request และผู้ใช้ใด ๆ ที่ส่งข้อมูลมาจะถูกกรองข้อมูลเพื่อความปลอดภัย
5. Application Controller : โหลดแบบจำลอง(Model) , ไลบรารีหลัก(Libraries) , plugins, ผู้ช่วย (Helpers) และทรัพยากรอื่นๆที่จำเป็นในการทำงานที่ถูกร้องขอมา
6. View : ปฏิบัติงานและถูกส่งกลับไปยังเบราว์เซอร์เพื่อโชว์หน้าจอ ถ้าระบบแคชถูกใช้งาน หน้าจอจะถูกแคชก่อนแล้วจึงค่อยส่งสิ่งที่ร้องขอมาเป็นลำดับถัดไป