บทที่ 6
การพัฒนาระบบช่วยจัดการเอกสาร สำหรับระบบแนะนำหอสมุดกลางบนเว็บ

ระบบช่วยจัดการเอกสาร สำหรับระบบแนะนำหอสมุดกลางบนเว็บ เป็นแอปพลิเคชั่นสำหรับช่วยผู้ดูแลระบบให้สามารถ เพิ่มข้อมูลใหม่ (สร้างเอกสารข้อมูลใหม่), แก้ไขหรืออัพเดทข้อมูลในเอกสารเก่า และลบเอกสารเก่าที่ไม่ต้องการแสดงในระบบ

โดย "เอกสาร" ในที่นี้คือ เอกสารข้อมูลของระบบที่เป็น XML ซึ่งระบบช่วยจัดการเอกสารจะสามารถ add, update, delete เอกสาร XML ที่ใช้รูปแบบโครงสร้างข้อมูลแบบที่ 3[ฃ] ได้ นั่นคือ ระบบช่วยจัดการเอกสารนี้ สามารถช่วยผู้ดูแลระบบ จัดการเอกสารในหมวดของบริการของสำนักหอสมุดกลาง, การสืบค้นข้อมูลด้วยคอมพิวเตอร์, การแบ่งส่วนราชการ, แนะนำห้องสมุดคณะ, ห้องสมุดในอนาคต ซึ่งเป็นหมวดหมู่ที่ข้อมูลมีความอ่อนไหวและมีโอกาสเปลี่ยนแปลงได้มาก

ผู้มีสิทธิใช้งานระบบช่วยจัดการเอกสาร จะต้องเป็นผู้ดูแลระบบ ที่ทราบ username และ password ในการเข้าสู่ระบบเท่านั้นที่มีสิทธิ์ในการใช้งาน โดยเมื่อ Login สำเร็จ ระบบจะสร้าง session สำหรับตรวจสอบสถานะการ login ตลอดการทำงานของระบบ ผู้ใช้งานระบบสามารถเลือกการจัดการเอกสารได้ 3 แบบ จากตัวเลือก

1. **เพิ่มข้อมูลใหม่** เมื่อมีข้อมูลในหัวข้อใหม่ สามารถเลือกเพิ่มข้อมูลใหม่ และเลือกหมวดหมู่ของข้อมูล เพื่อกรอกข้อมูลที่เป็นหัวข้อใหม่ ลงในแบบฟอร์มที่ระบบเตรียมไว้ให้ จากนั้นระบบจะทำการสร้างเอกสาร XML ด้วยข้อมูลจากแบบฟอร์ม ให้อยู่ภายใต้หมวดหมู่ที่เลือกและเพิ่ม link ไปยังหัวข้อใหม่ลงในเอกสารที่เก็บรายการเมนูย่อยสำหรับหมวดหมู่นั้น ๆ โดยอัตโนมัติ

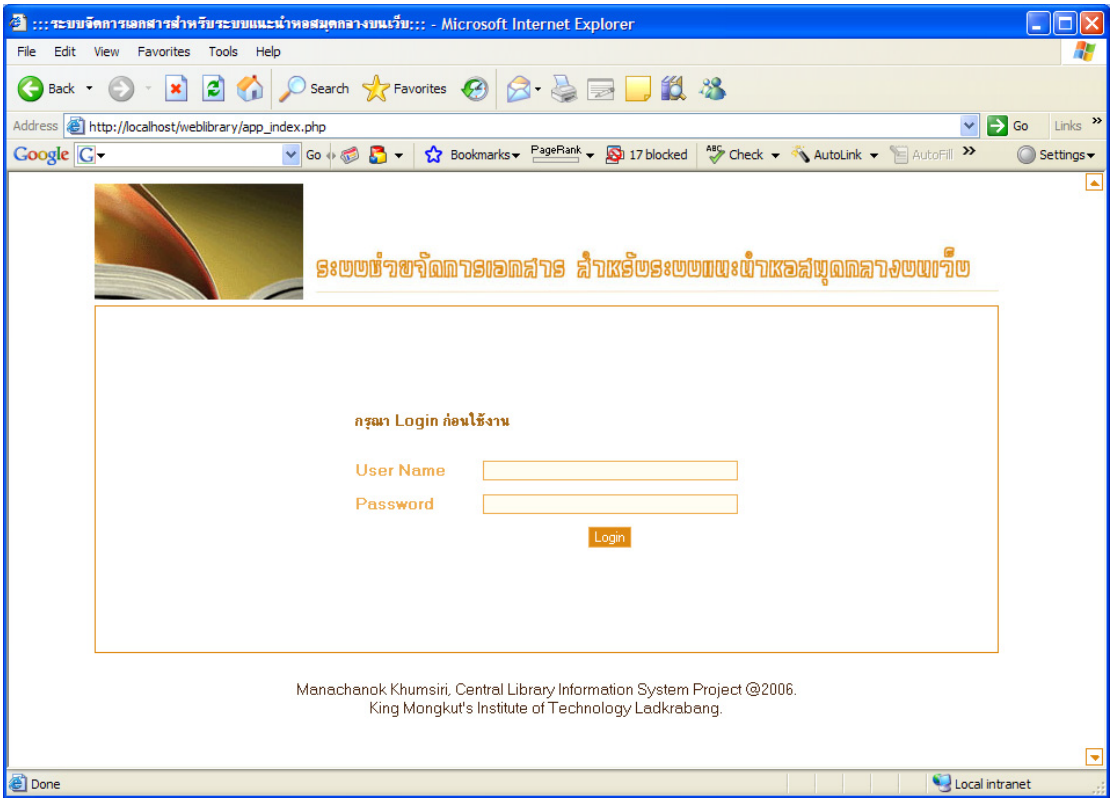
2. **แก้ไขข้อมูล** เมื่อข้อมูลการให้บริการต่าง ๆ ของหอสมุดกลางเปลี่ยนแปลงไป ต้องมีการอัพเดทข้อมูลที่ทันสมัยให้แก่ผู้ใช้บริการได้ทราบอยู่เสมอ โดยผู้ดูแลระบบสามารถแก้ไขหรืออัพเดทข้อมูลภายในเอกสาร XML ในหัวข้อการบริการที่มีอยู่แล้ว โดยเลือกการแก้ไขข้อมูล เลือกหมวดหมู่และหัวข้อที่ต้องการแก้ไข ระบบจะนำข้อมูลจากเอกสาร XML มาแสดงในแบบฟอร์มให้ผู้ดูแลระบบเลือกแก้ไขในส่วนต่าง ๆ ให้ตรงกับข้อมูลที่มีการอัพเดท โดยสามารถแก้ไขรูปภาพประกอบได้ด้วย เมื่อแก้ไขเสร็จระบบจะจัดเก็บข้อมูลลงในเอกสาร XML ให้

[ฃ] ดูบทที่ 5 หัวข้อที่ 5.3.3 หน้า 72

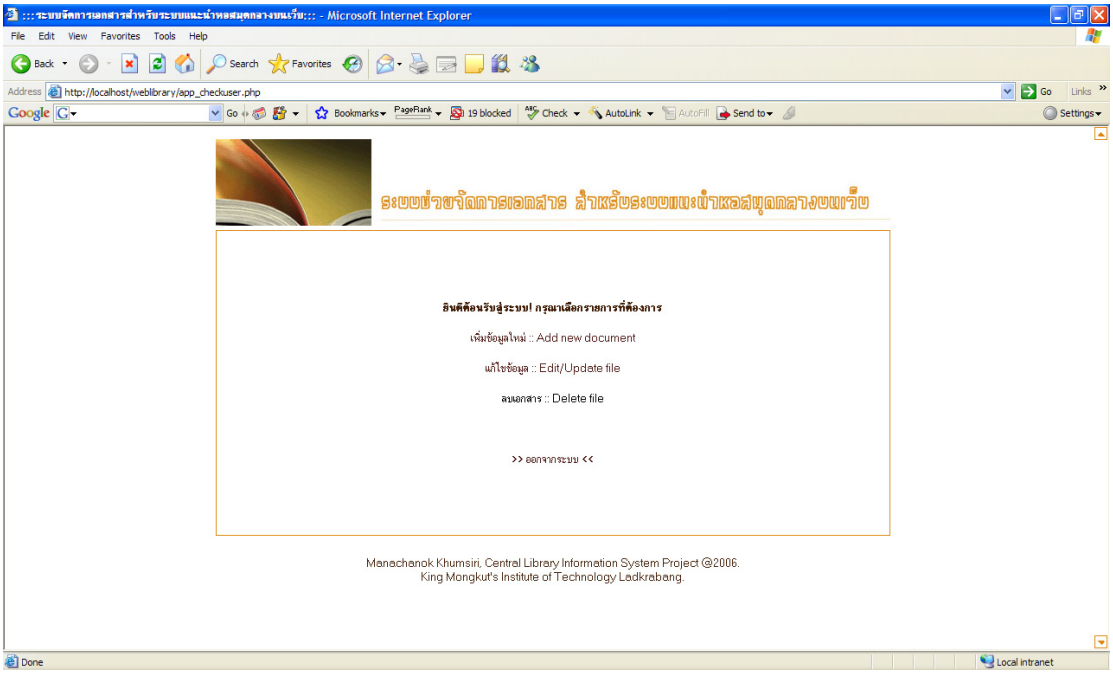
3. **ลบเอกสาร** เมื่อบริการหัวข้อใด ไม่มีให้บริการในหอสมุดกลางแล้ว สามารถลบออกจากระบบได้โดยเลือกลบเอกสาร เลือกหมวดหมู่และหัวข้อที่ต้องการลบ ระบบจะทำการลบ link ออกจากเมนูย่อยและลบเอกสารที่เก็บข้อมูลนั้นออกไปจากระบบ

สำหรับการทำงานของระบบช่วยจัดการเอกสาร สามารถเขียนเป็น Flowchart ได้ ดังรูปที่ 6.1

รูปที่ 6.1 การทำงานของระบบช่วยจัดการเอกสาร



รูปที่ 6.2 แสดง “app_index.php” ซึ่งเป็น homepage ของระบบช่วยจัดการเอกสาร



รูปที่ 6.3 หาก user name และ password ถูกต้อง จะสามารถเลือกรายการที่ต้องการได้

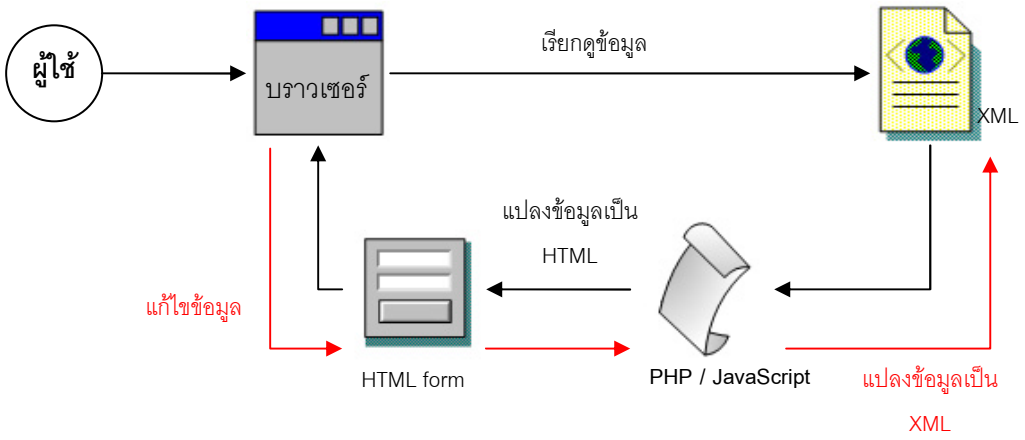
6.1 การออกแบบระบบช่วยจัดการเอกสาร

การออกแบบระบบช่วยจัดการเอกสาร สำหรับระบบแนะนำหอสมุดกลางบนเว็บ มีจุดประสงค์เพื่อช่วยให้ผู้ดูแลระบบสามารถแก้ไข อัพเดทข้อมูลต่าง ๆ ที่อยู่ในระบบแนะนำหอสมุดกลางบนเว็บ โดยไม่ต้องศึกษา XML รวมไปถึงความหมายของอิลิเมนต์ที่กำหนดขึ้นเพื่ออธิบายข้อมูลสำหรับเอกสารนั้น ๆ มาก่อน ทำให้การแก้ไข อัพเดทข้อมูลซึ่งมีโอกาสเปลี่ยนแปลงได้อยู่เสมอ เป็นเรื่องง่าย และสะดวก โดยจัดทำเป็น web application เนื่องจากการทำเป็น web application จะทำให้ใช้งานได้ง่าย สามารถ config ระบบและติดตั้งบน server ให้ผู้ดูแลระบบได้สะดวก นอกจากนี้การแปลงข้อมูล XML ยังต้องการใช้ parser ซึ่งมีพร้อมแล้วใน web server และบราวเซอร์ อีกด้วย สำหรับภาษาที่ใช้ในการพัฒนาระบบคือภาษา PHP และ JavaScript

หลักการของระบบจะทำการห่อหุ้ม XML code เอาไว้ และแสดงเฉพาะข้อมูลและความหมายผ่านแบบฟอร์มที่เข้าใจได้ง่าย โดยใช้ภาษา PHP และ JavaScript มาช่วยในการอ่านเอกสาร XML แล้วนำไปแสดงผลข้อมูลติดต่อกับผู้ใช้งานผ่านแบบฟอร์ม (ผ่าน DOM โดยเขียน code ด้วยภาษา PHP หรือ JavaScript)

เมื่อผู้ใช้งานแก้ไขข้อมูลเรียบร้อยและทำการบันทึกข้อมูล ข้อมูลจากแบบฟอร์มนั้นก็จะถูกแปลงกลับเป็นเอกสาร XML และบันทึกลงในระบบ (โดยใช้ PHP แปลงกลับเช่นกัน)

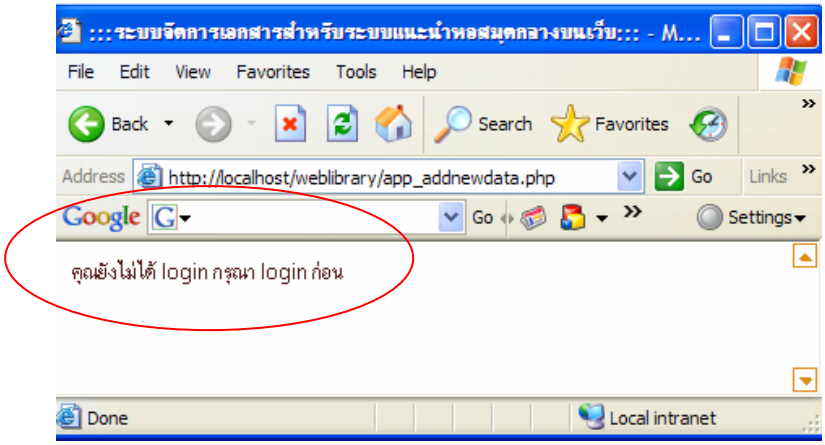
ขั้นตอนในการอ่านและเขียนเอกสาร XML ด้วย PHP และ JavaScript แสดงดังรูปที่ 6.4



รูปที่ 6.4 ขั้นตอนการแปลงเอกสาร XML ด้วย PHP และ JavaScript เพื่อแสดงผล

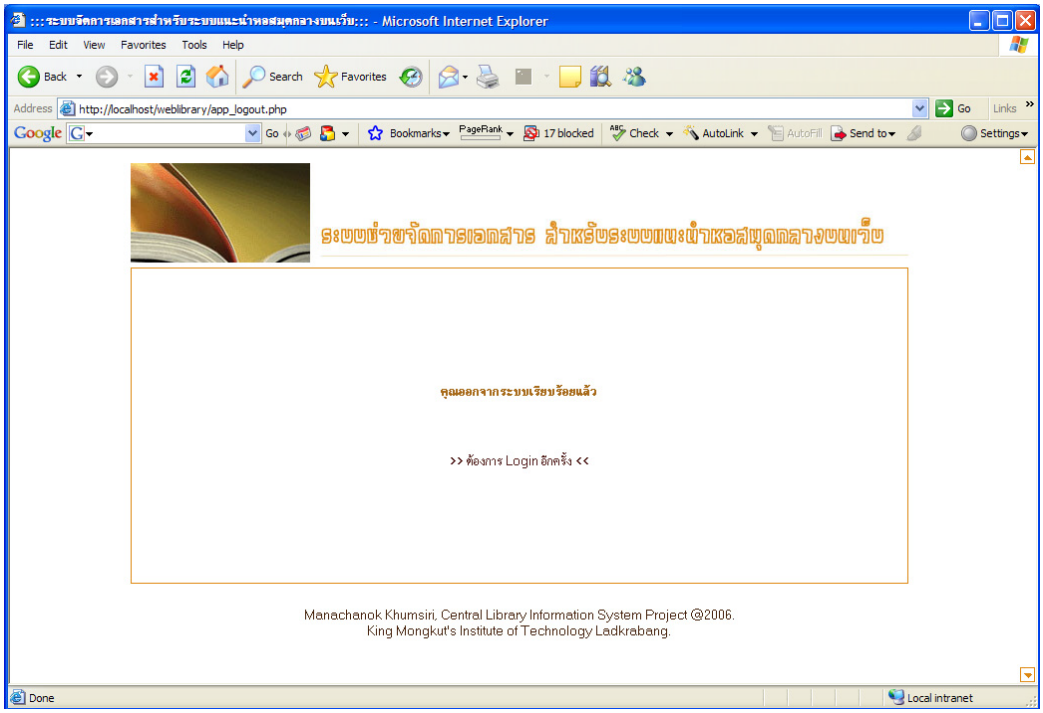
สำหรับไฟล์ที่เกี่ยวข้องกับระบบจัดการเอกสารจะถูกเก็บไว้ในไดเร็กทอรีเดียวกันกับเอกสาร XML โดยไฟล์ที่เกี่ยวข้องกับระบบจัดการเอกสารจะชื่อนำหน้าว่า “app_” โดยผู้ใช้อื่น

ที่ไม่ใช่ผู้ดูแลระบบที่ทราบ username และ password จะไม่สามารถเข้าถึงข้อมูลในระบบส่วนนี้ได้ เนื่องจากถูกจำกัดการใช้งานโดย session จึงทำให้สามารถเข้าสู่ระบบได้ผ่านหน้า Login เท่านั้น



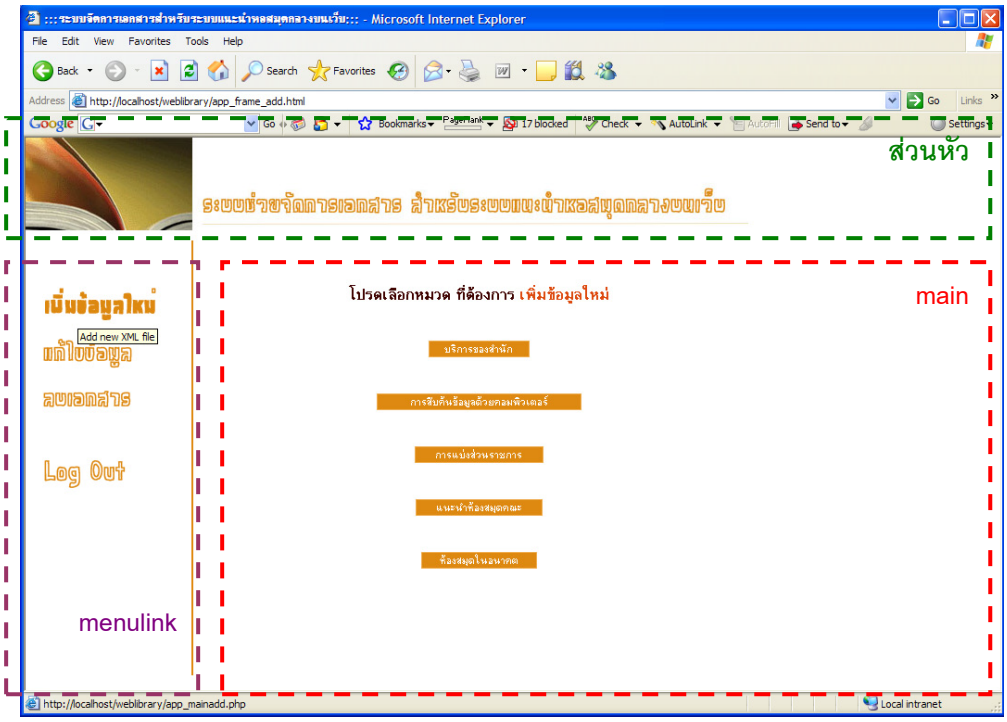
รูปที่ 6.5 ไม่สามารถใช้งานแอปพลิเคชั่นนี้ได้ถ้าไม่ได้ Login

ในส่วนของอินเทอร์เฟสของระบบ ประกอบด้วยอินเทอร์เฟสสองรูปแบบ ได้แก่ อินเทอร์เฟสสำหรับหน้า login ซึ่งใช้กับหน้า login เข้าสู่ระบบ (app_index.php), หน้าตรวจสอบสิทธิของผู้ใช้ (app_checkuser.php) และหน้าที่แสดงผล logout สำเร็จ (app_logout.php) ตัวอย่างอินเทอร์เฟสในส่วนนี้แสดงดังรูปที่ 6.2, 6.3 และ 6.6



รูปที่ 6.6 แสดง “app_logout.php”

อินเทอร์เฟสสำหรับการจัดการข้อมูล ใช้ frame ในการแสดงผล โดยเมนูทางซ้ายมือจะเป็น link สำหรับเพิ่มข้อมูล, แก้ไขข้อมูล, ลบเอกสาร และ Log out ทำให้ผู้ใช้งานระบบสามารถเรียกใช้งานแอปพลิเคชั่นนั้นได้ง่ายขึ้น ส่วน frame ฝั่งขวามือใช้สำหรับแสดงผล page ต่าง ๆ ที่กำลังใช้งาน



รูปที่ 6.7 อินเทอร์เฟสสำหรับส่วนจัดการข้อมูล ซึ่งใช้เฟรมในการแสดงผล

6.2 การเพิ่มข้อมูลใหม่

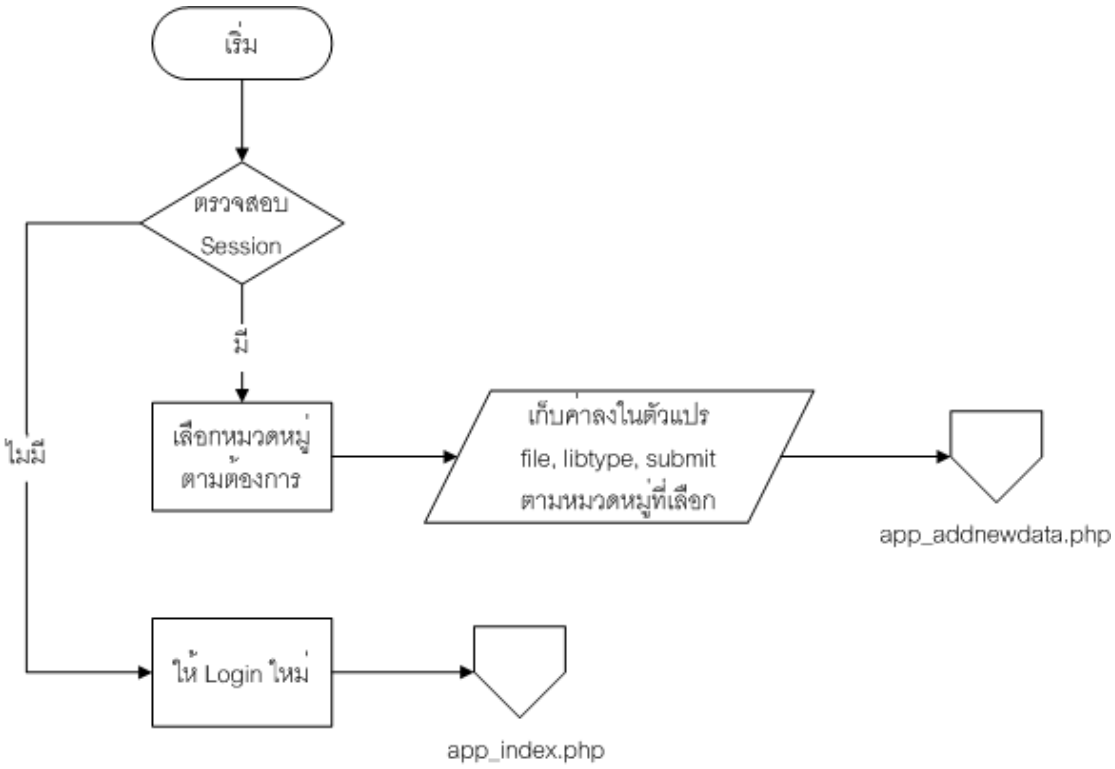
หากต้องการเพิ่มข่าวสารเกี่ยวกับบริการใหม่ ๆ ของสำนักหอสมุดกลาง ซึ่งเป็นหัวข้อที่ไม่เคยมีมาก่อนในระบบแนะนำหอสมุดกลางบนเว็บ สามารถเลือก “เพิ่มข้อมูลใหม่” เพื่อสร้างเอกสาร XML สำหรับเก็บข้อมูลนั้น โดยมีขั้นตอนการเพิ่มข้อมูลใหม่ดังนี้

รูปที่ 6.8 Flowchart ของการเพิ่มข้อมูลใหม่

การทำงานของโปรแกรมในส่วน การเพิ่มข้อมูลใหม่

จากรูปที่ 6.8 จะเห็นว่า สคริปต์หลักที่เกี่ยวข้องกับทำงานในส่วน การเพิ่มข้อมูลใหม่ มีทั้งหมด 3 สคริปต์ประกอบด้วย

app_mainadd.php ทำหน้าที่แสดงรายการหมวดหมู่ของข้อมูล เพื่อให้ผู้ใช้เลือกหมวดหมู่ให้กับข้อมูลใหม่ก่อน



รูปที่ 6.9 Flowchart **สำหรับ** app_mainadd.php

การทำงานของ app_mainadd.php เริ่มจากการสร้าง session และตรวจสอบว่ามีการสร้าง session หรือไม่ ถ้ามีจึงแสดงรายการหมวดหมู่ เป็นปุ่ม button ออกทางจอภาพ (แต่ถ้าการตรวจสอบ session พบว่าไม่มีการตัวแปร session จะเตือนให้ log in ใหม่)

เมื่อผู้ใช้กดเลือกหมวดหมู่สำหรับเอกสารข้อมูลใหม่แล้ว จึงส่งต่อค่าจากแบบฟอร์มที่เลือกไปยังสคริปต์หน้าต่อไป (app_addnewdata.php) โดยความหมายของตัวแปรจากฟอร์มที่ส่งออกไปได้แก่

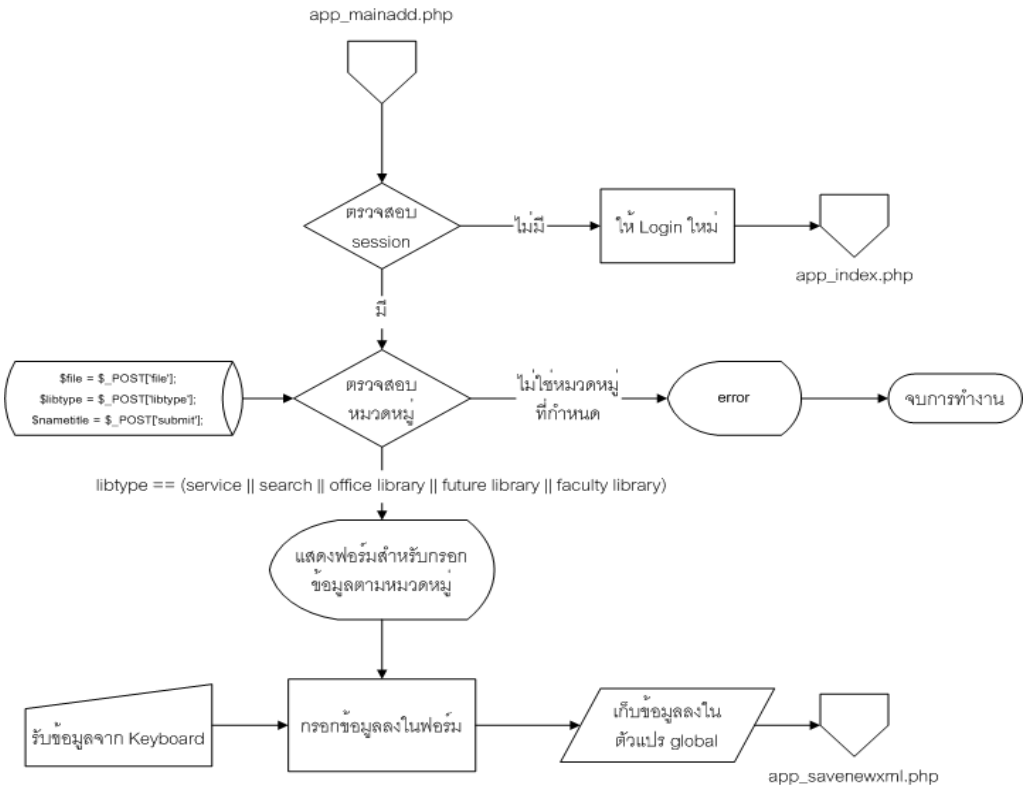
“file” เก็บ ชื่อไฟล์ XML ที่เก็บข้อมูลเมนูย่อยสำหรับหมวดหมู่นั้น (ในรูป xxx_link.xml เมื่อ xxx เป็นรหัสของหมวดหมู่)

“libtype” เก็บ รหัสชื่อหมวดหมู่ เป็นภาษาอังกฤษ เป็นตัวแปรสำคัญสำหรับบ่งบอกให้สคริปต์ต่อ ๆ ไปทราบว่า ข้อมูลที่กำลังดำเนินการอยู่ในหมวดหมู่ใด

“submit” เก็บ ชื่อหมวดหมู่เป็นภาษาไทย

รูปที่ 6.10 **ผลลัพธ์ของ** app_mainadd.php **บนบราวเซอร์** IE 6

app_addnewdata.php เป็นแบบฟอร์มสำหรับกรอกข้อมูลเพื่อนำไปสร้างเอกสาร XML โดยสคริปต์นี้จะทำหน้าที่เลือกฟอร์มที่เหมาะสมกับข้อมูลในหมวดหมู่ต่าง ๆ เพื่อให้ผู้ใช้กรอกข้อมูลและเก็บข้อมูลได้เหมาะสมกับโครงสร้าง XML ของหมวดหมู่นั้น และทำการกำหนดชื่อไฟล์ให้เอกสารและรูปภาพที่ทำการ upload



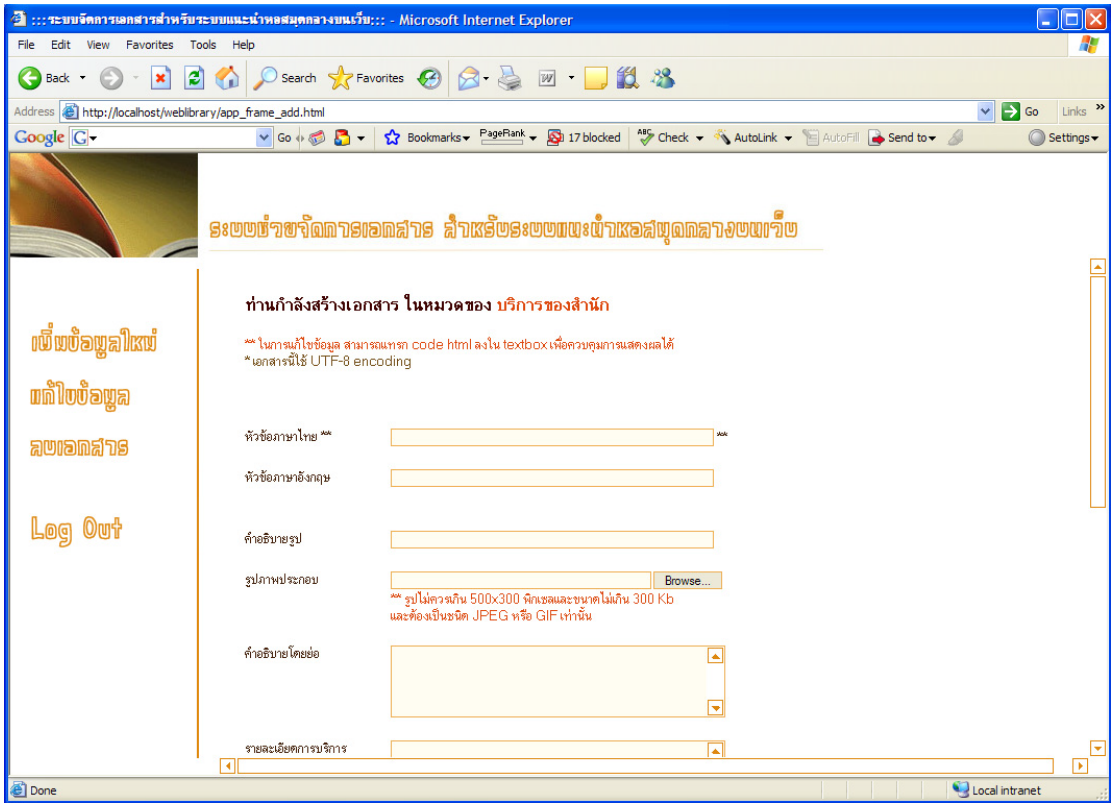
รูปที่ 6.11 Flowchart **สำหรับ** app_addnewdata.php

การทำงานของ app_addnewdata.php เริ่มจากการสร้าง session , include ส่วนหัวของเอกสาร HTML (include("app_header.php")) และสร้างตัวแปรมารับค่าที่ส่งมาจากฟอร์มของ app_mainadd.php โดยตัวแปรที่มีความสำคัญคือตัวแปร libtype ซึ่งเก็บรหัสหมวดหมู่ที่ส่งมาจากแบบฟอร์มก่อนหน้านี้ แล้วทำการตรวจสอบ session ถ้ามีตัวแปร session ถูกต้อง สคริปต์จะสร้างแบบฟอร์ม ที่มีชื่อว่า “form” และเก็บค่าของชื่อหมวดหมู่ที่เป็นภาษาไทย และรหัสหมวดหมู่ภาษาอังกฤษ (ซึ่งรับค่ามาจาก app_mainadd.php อีกที) ลงใน input type='hidden' ที่มีชื่อว่า filename และ mylibtype ตามลำดับ

จากนั้นทำการตรวจสอบรหัสหมวดหมู่จากตัวแปร libtype ว่าจะเป็นหมวดหมู่ใด เพื่อเลือกรูปแบบของฟอร์มที่เหมาะสมกับหมวดหมู่ โดยตัวแปร libtype จะมีผลต่อ input text ของแบบฟอร์ม ซึ่ง input เหล่านี้ แต่ละ input จะถูกนำไปแปลงเป็นเอกสาร XML ต่อไป

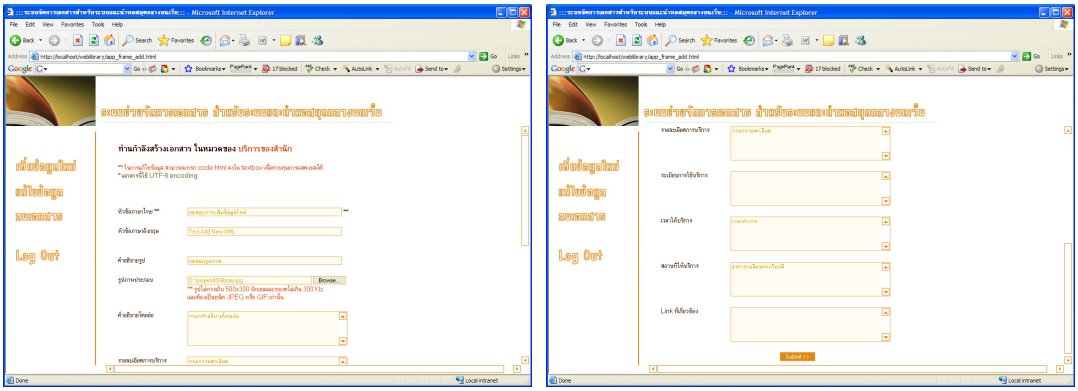
ในการเพิ่มข้อมูลใหม่ ควรเลือกหมวดหมู่ให้ตรงกับข้อมูลที่ต้องการเพิ่ม และในการกรอกแบบฟอร์มข้อมูลใหม่มีข้อแนะนำดังนี้

- ต้องกรอกข้อมูลในช่อง “หัวข้อภาษาไทย” เสมอ (หรือ ช่อง “หัวข้อ/ประเด็นยุทธศาสตร์” สำหรับหมวดหมู่ห้องสมุดในอนาคต) เพราะเป็น key สำคัญที่จะบอกว่าข้อมูลใหม่นี้เกี่ยวกับเรื่องราวใด (อาจเป็นชื่อของบริการ เป็นต้น) และไม่ควรเป็นหัวข้อที่มีอยู่แล้วในระบบ (ถ้าข้อมูลนั้นเกี่ยวกับหัวข้อที่มีอยู่แล้ว ควรใช้การ “แก้ไขข้อมูล”)
- สามารถแนบรูปภาพได้ โดยรูปภาพที่ upload จะต้องเป็นชนิด JPEG หรือ GIF เท่านั้น และต้องมีขนาดไม่เกิน 300 kb (ไม่เช่นนั้นจะไม่อนุญาติให้ upload รูป) และไม่ควรมีความกว้างxสูงเกินกว่า 500x300 พิกเซล (เนื่องจากจะทำให้การแสดงผลไม่สวยงาม) โดยรูปจะถูก upload ไว้ในไดเร็กทอรี่ “images/” บน server
- สามารถพิมพ์ HTML code ลงในช่อง คำอธิบายโดยย่อ และช่องที่อยู่ต่ำลงไปได้ ซึ่งการเพิ่ม tag HTML จะมีผลต่อการแสดงผลของข้อมูลบนหน้าเว็บด้วย (คำแนะนำ หากต้องการให้ข้อมูลแสดงผลด้วยการขึ้นบรรทัดใหม่ ควรเพิ่ม หรือ
 ลงในช่องที่ทำการกรอกข้อมูลด้วย)
- นอกจากหัวข้อภาษาไทยแล้ว ไม่จำเป็นต้องกรอกข้อมูลให้ครบทุกช่อง สามารถกรอกข้อมูลเฉพาะส่วนที่ต้องการแจ้งก็ได้ เช่นอาจจะกรอกเฉพาะส่วนของ สถานที่ให้บริการ แต่ไม่กรอกเวลาให้บริการ ก็ได้ (แต่ผู้เข้าชมเว็บ ก็จะไม่ทราบข้อมูลเวลา เนื่องจากไม่มีข้อมูล)



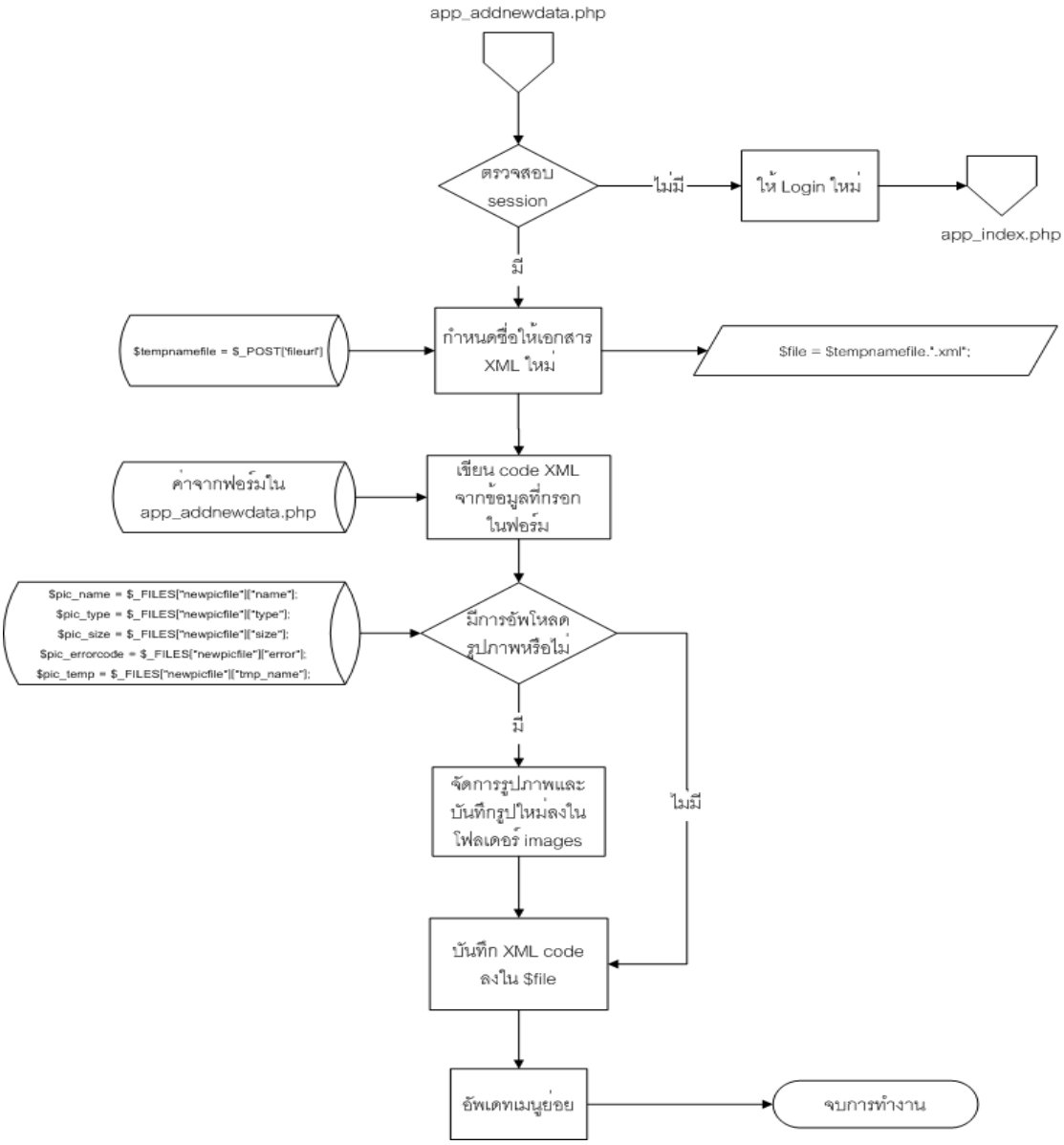
รูปที่ 6.12 แบบฟอร์มสำหรับกรอกข้อมูลใหม่

เมื่อกรอกข้อมูลเสร็จสิ้น ให้กดปุ่ม “Submit >>” ด้านล่าง เพื่อให้ระบบทำการบันทึกข้อมูล โดยเรียกใช้สคริปต์ app_savenewxml.php



รูปที่ 6.13 เมื่อกรอกข้อมูลเรียบร้อยให้กดปุ่ม “Submit >>”

app_savenewxml.php เป็นสคริปที่ทำหน้าที่แปลงข้อมูลที่รับมาจากฟอร์มจาก app_addnewdata.php เป็นเอกสาร XML และเพิ่ม link ข้อมูลใหม่ลงในเมนูย่อยภายใต้หมวดหมู่ที่เราเพิ่มข้อมูลลงไป



รูปที่ 6.14 Flowchart **สำหรับ** app_savenewxml.php

การทำงานของสคริปต์ app_savenewxml.php เริ่มจากการสร้างและตรวจสอบ session และสร้างตัวแปรมารับค่าต่าง ๆ ที่ส่งมาจากฟอร์ม และตัวแปรที่เก็บอิลิเมนต์ข้อมูลส่วนต่าง ๆ ของ XML สำหรับข้อมูลในหมวดต่าง ๆ เช่น headxml เก็บ string ที่เป็น Tag เริ่มต้นเอกสาร XML, rootxml คือ string ของ Tag libraryinfo ที่กำหนดให้มี type เท่ากับ libraryinfo_type , engname เท่ากับ ENGTITLE ฯลฯ เป็นต้น จะเห็นได้ว่า value ต่าง ๆ ที่รับมาจากฟอร์มของสคริปต์ก่อนหน้าจะถูกนำมาใส่ในส่วนของข้อมูลของ XML

ฟังก์ชัน setInfoData(value) เป็นฟังก์ชันที่ตรวจสอบรหัสหมวดหมู่ที่รับมา ว่าเป็นเอกสารหมวดหมู่ใด ซึ่งจะทำให้เลือกชุดข้อมูลของ <info> ได้เหมาะสมกับหมวดหมู่ของเอกสาร

ฟังก์ชัน managePicture() เป็นฟังก์ชันสำคัญในการจัดการรูปภาพ ซึ่งสามารถตรวจสอบชนิดของรูปภาพและขนาดไฟล์ได้ หากผู้ใช้ได้เพิ่มไฟล์รูปซึ่งมีคุณสมบัติถูกต้องตามข้อตกลงที่ตั้งไว้ (ต้องเป็น JPEG หรือ GIF และมีขนาดไม่เกิน 300 kb) สคริปต์จะทำการตั้งชื่อไฟล์รูปที่อัพโหลดให้ใหม่ซึ่งจะเป็นชื่อเดียวกับไฟล์เอกสาร และบันทึกลงในไดเร็กทอรี่ "/images/" ของ server และเรียกฟังก์ชัน setPictureData โดยส่งค่า แหล่งของรูปที่กำหนดให้ไปด้วย แต่ถ้าไฟล์รูปภาพที่อัพโหลดไม่ผ่านตามเงื่อนไข ระบบจะไม่บันทึกไฟล์รูปลงใน server และส่งแหล่งของรูปเป็น "" ไปให้ฟังก์ชัน setPictureData

ฟังก์ชัน setPictureData(picATL, picSRC, picW, picH) จะรับค่า คำอธิบายรูปภาพ, แหล่งของรูป, ความกว้าง, ความยาว ตามลำดับ เพื่อนำมาสร้าง <picture> เพื่อสร้าง <picture> สำหรับเขียนข้อมูลเกี่ยวกับรูปภาพลงในไฟล์ XML

ฟังก์ชัน toDataString() เป็นฟังก์ชันสำหรับต่อ string ซึ่งเป็น code XML ที่สมบูรณ์เพื่อนำไปเขียนลงในไฟล์ใหม่ที่เราจะสร้างขึ้น

โดยการสร้างเอกสาร XML ขึ้นใหม่จากข้อมูลที่มีอยู่นั้น สามารถทำได้ด้วยคำสั่ง fopen(file, "w") เพื่อเปิดไฟล์แล้วเขียนแบบธรรมดา แต่ที่ทำให้เป็น XML เพราะเราเตรียมข้อมูลที่จะเขียนให้อยู่ในรูปแบบ Tag ของ XML ไว้แล้ว และเซฟไฟล์เป็น .xml นั่นเอง เป็นอันสิ้นสุดขั้นตอนการสร้างเอกสาร XML ใหม่ด้วยข้อมูลที่รับมาจากฟอร์ม

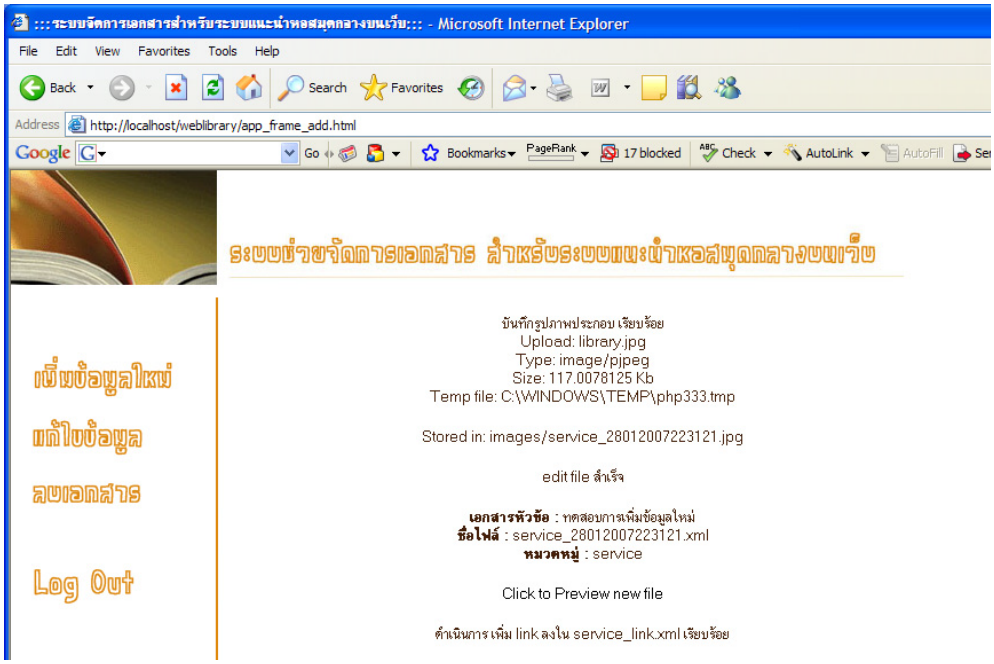
เมื่อมีเอกสารข้อมูลเพิ่มขึ้นมาใหม่ จะต้องมีการสร้างเมนูย่อยให้ระบบสามารถ link ไปยังไฟล์ข้อมูลใหม่นี้ได้ โดยสคริปต์ app_savenewxml.php นี้จะทำการเพิ่ม link ในเมนูย่อยให้ด้วย โดยใช้คำสั่งตั้งแต่บรรทัดที่มี comment ว่า "////////////////////========== open xml link file (xxx_link.xml) for add <subject> element ==========////////////////" ลงไป

ฟังก์ชัน setFileLink(value) เป็นฟังก์ชันสำหรับระบุชื่อไฟล์ XML ที่เก็บข้อมูลเมนูย่อยของแต่ละหมวดหมู่ โดยเมื่อทำการเรียกฟังก์ชันและส่งค่ารหัสของหมวดหมู่แล้วฟังก์ชันจะคืนค่าไฟล์ XML ที่เก็บเมนูย่อยของหมวดหมู่นั้นมาให้ เพื่อนำไปเปิด และเพิ่มเมนูย่อยต่อไป

ในการเพิ่มเมนูย่อยลงในเอกสาร XML นั้น PHP จะเรียกฟังก์ชันที่เกี่ยวกับ xml_parser [ฑ] ซึ่งเป็นฟังก์ชันที่มีอยู่แล้วใน library ของ PHP ซึ่งช่วยให้สามารถเปิดไฟล์ XML ขึ้นมาอ่านแล้วเขียนฟังก์ชันมาเพื่อกำหนดได้ว่าเมื่ออ่านเจอ element อะไรแล้วให้จัดการอย่างไร หรืออ่านเจอ PCDATA แล้วให้จัดการอย่างไร โดยในที่นี้ เราต้องการเพิ่ม node subject ที่เก็บหัวข้อใหม่และ link ลงไปท้ายเอกสาร (สามารถดูโครงสร้าง XML สำหรับเมนูย่อยในบทที่ 5 ตัวอย่างที่ 5.1 หน้า 69) จึงกำหนดให้สคริปต์ทำการอ่านไฟล์ XML ไปเรื่อย ๆ และบันทึก string ที่อ่านได้ลงในตัวแปร

[ฑ] ดูบทที่ 4

tempData จนกว่าจะอ่านเจอ </menulink> (ฟังก์ชัน endElement) จึงให้แทรกโหนด subject ของข้อมูลเมนูย่อยใหม่ที่เก็บอยู่ในตัวแปร lastelement ลงไปก่อน ปิดท้ายด้วย end tag </libraryinfo> จากนั้นทำการบันทึก String ทั้งหมดลงในไฟล์เมนูย่อยเดิม

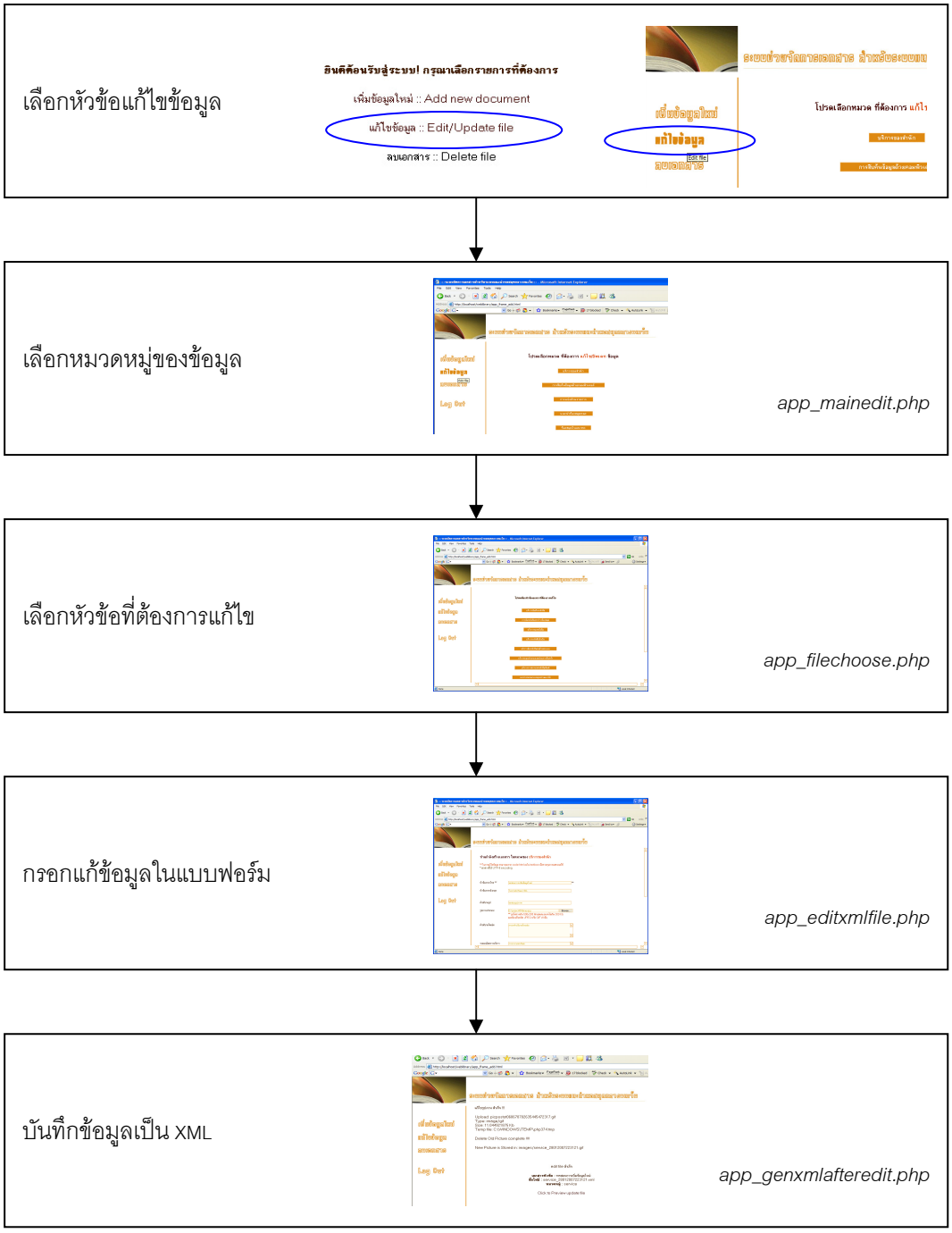


รูปที่ 6.15 แจ้งผลการบันทึกเอกสาร XML บันทึกรูปภาพ และเพิ่มเมนูย่อยลงในระบบ

การทำงานทั้งหมดทั้งการสร้างไฟล์ XML ใหม่ การอัพโหลดรูปภาพ และการอัพเดทเมนูย่อย สคริปต์จะจัดการให้ทั้งหมด โดยผู้ใช้สามารถดูผลลัพธ์ของ page ที่สร้างขึ้นใหม่ได้จาก “Click to Preview new file” หากต้องการแก้ไขข้อมูลภายใต้หัวข้อนั้นอีกครั้ง ให้เลือกปุ่ม “แก้ไขข้อมูล” ทางด้านซ้ายมือและดำเนินการตามขั้นตอนการแก้ไขข้อมูล

6.3 การแก้ไขข้อมูล

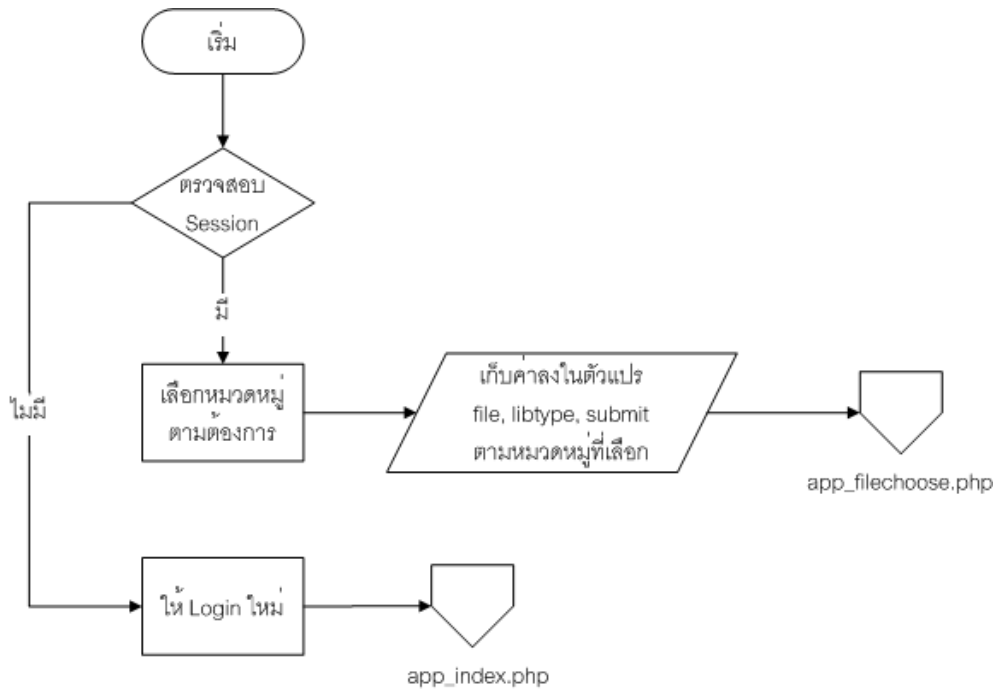
หัวข้อการแก้ไขข้อมูล ช่วยให้ผู้ดูแลระบบสามารถแก้ไข / อัพเดทข้อมูลในเอกสาร XML ที่มีอยู่ในระบบ โดยการแก้ไขข้อมูลผ่านแบบฟอร์ม โดยมีขั้นตอนการแก้ไขข้อมูลดังนี้



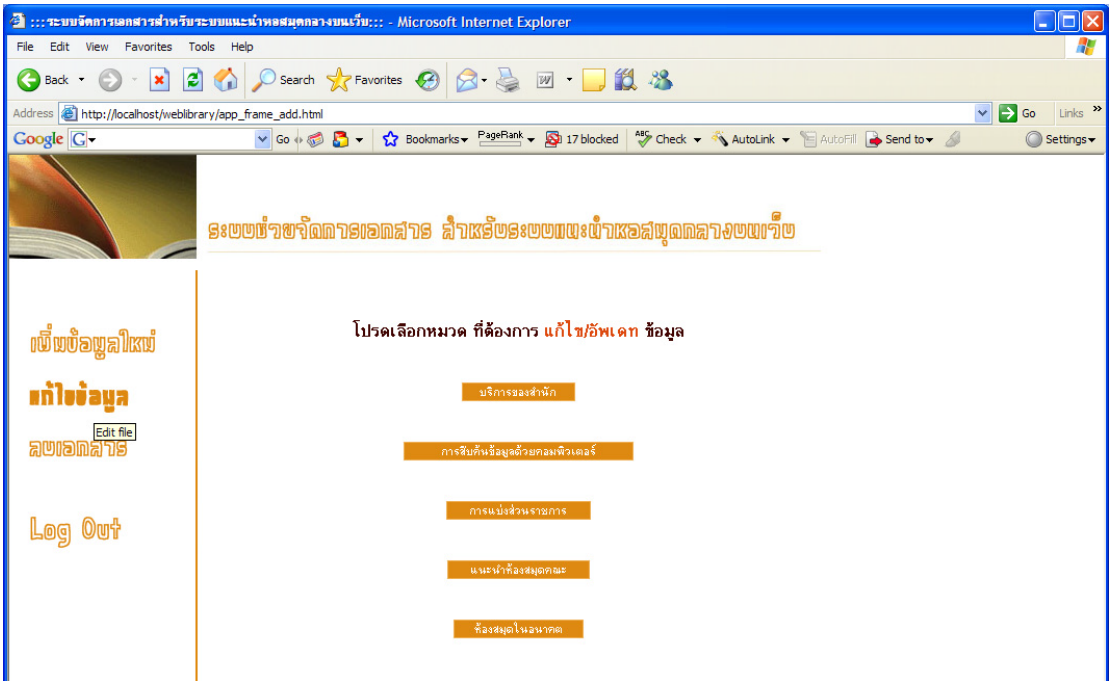
รูปที่ 6.16 Flowchart ของการแก้ไขข้อมูล

การทำงานของโปรแกรมในส่วน การแก้ไขข้อมูล ประกอบด้วยสคริปต์ในการจัดการ 4 ไฟล์ ได้แก่

app_mainedit.php ทำหน้าที่แสดงรายการหมวดหมู่ของข้อมูล เพื่อให้ผู้ใช้เลือกหมวดหมู่ของข้อมูลที่ต้องแก้ไข มีการทำงานคล้ายกับ app_mainadd.php แต่ฟอร์มของ app_mainedit.php จะส่งชื่อไฟล์ของเมนูย่อย (เป็น xxx_link.xml) ไปกับตัวแปร file ด้วย

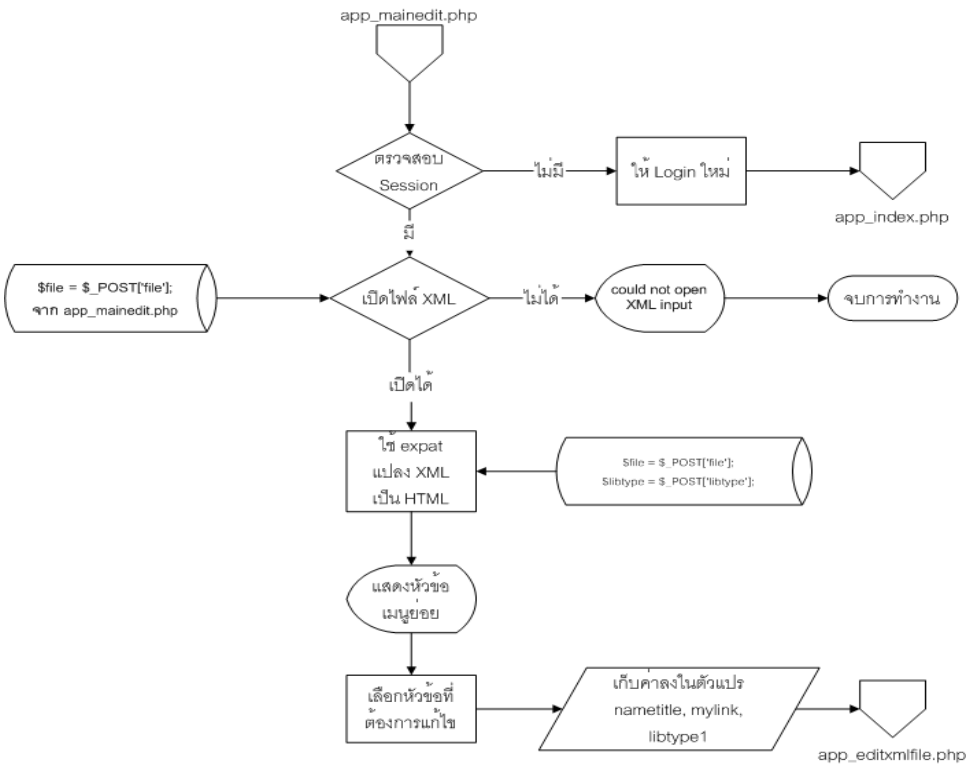


รูปที่ 6.17 Flowchart **สำหรับ** app_mainedit.php



รูปที่ 6.18 **ผลลัพธ์ของ** app_mainedit.php **บนบราวเซอร์** IE 6

app_filechoose.php แสดงรายการหัวข้อ จากหมวดหมู่ที่เลือก เพื่อให้ผู้ใช้เลือกหัวข้อที่ต้องการแก้ไขข้อมูล โดยสคริปต์จะนำข้อมูลจากเอกสาร XML ที่เก็บเมนูย่อยของหมวดหมู่ที่เลือกมาแสดงเป็นปุ่ม button รายการชื่อหัวข้อที่มีอยู่

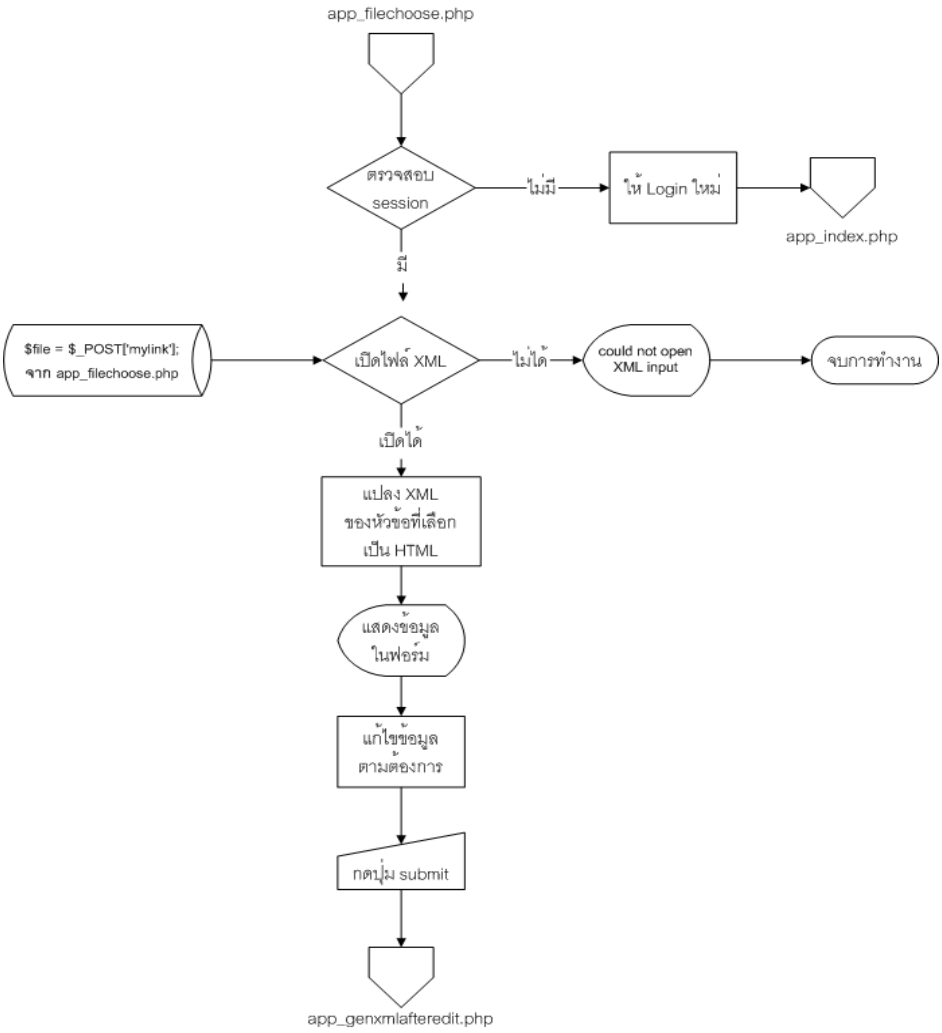


รูปที่ 6.19 Flowchart **สำหรับ** app_filechoose.php

การทำงานของ app_filechoose.php เริ่มจากการสร้าง session และตรวจสอบ session ถ้ามี session ก็จะทำการอ่านไฟล์ XML ที่เก็บข้อมูลเมนูย่อยของหมวดหมู่ที่เลือก (โดยรับชื่อไฟล์มาจาก file ที่ app_mainedit.php ส่งมาให้) จากนั้นใช้ xml parser ของ php ในการอ่านข้อมูลจากไฟล์ XML และแปลงอิลิเมนต์ที่อ่านได้เป็น HTML code โดยแต่ละ <subject> node จาก xxx_link.xml จะถูกแปลงเป็น HTML form 1 ฟอร์มที่มี input ประเภท hidden ชื่อ mylink สำหรับส่งชื่อไฟล์ข้อมูล XML ที่เลือก และชื่อ libtype1 ส่งต่อชื่อไฟล์เมนูย่อยที่รับมาจาก app_mainedit.php

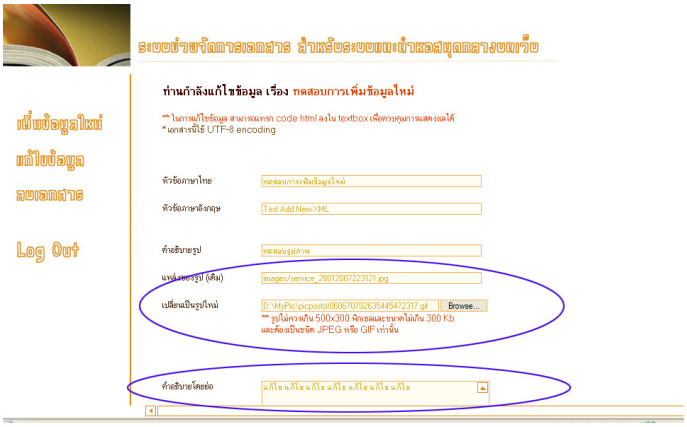
รูปที่ 6.20 **ผลลัพธ์ของ** app_filechoose.php **บนบราวเซอร์**

app_editxmlfile.php เป็นสคริปต์ที่ทำการอ่านข้อมูลจากไฟล์ xml ของหัวข้อที่เลือก แล้วนำมาแปลงให้อยู่ในรูปแบบฟอร์มของ HTML โดยผู้ใช้สามารถแก้ไขข้อมูลผ่านแบบฟอร์มได้เลย



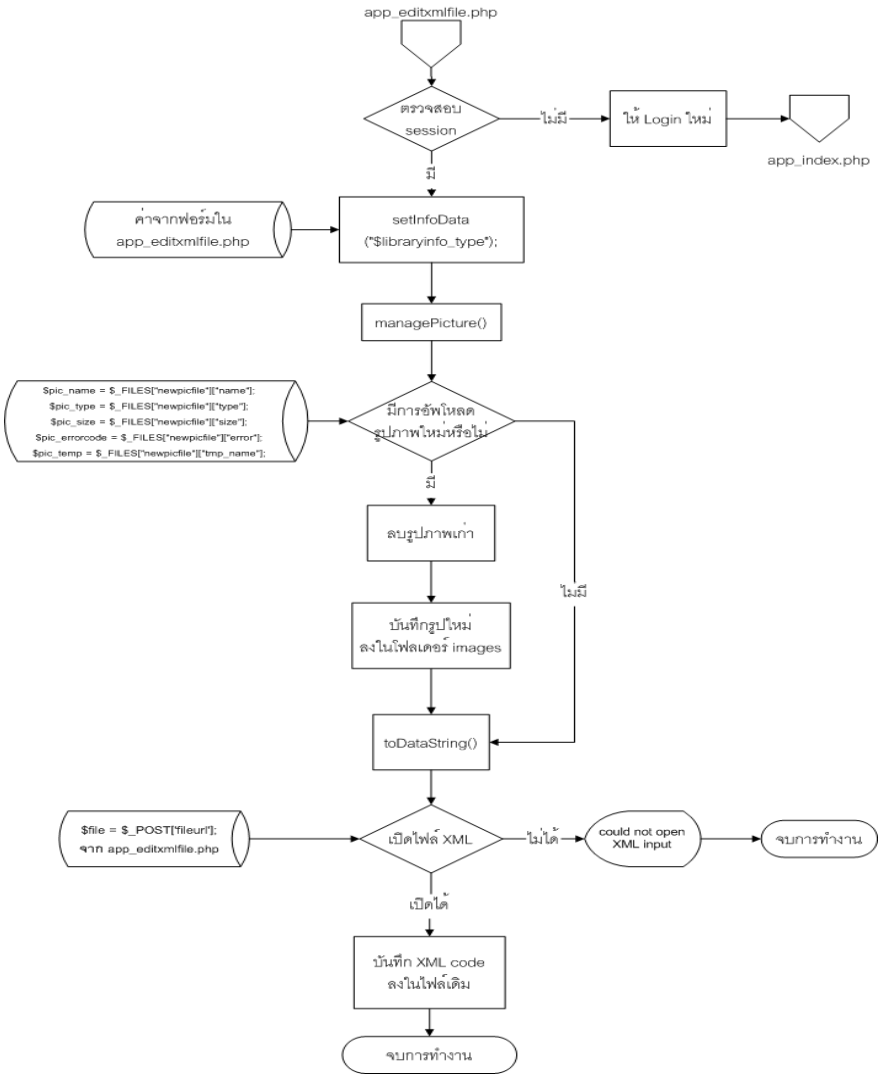
รูปที่ 6.21 Flowchart **สำหรับ** app_editxmlfile.php

การทำงานของ app_editxmlfile.php เริ่มจากการสร้าง session และตรวจสอบ session ถ้ามี session ก็จะทำการอ่านไฟล์ XML ที่เก็บข้อมูลเมนูย่อยของหมวดหมู่ที่เลือก (โดยรับชื่อไฟล์มาจาก file ที่ app_mainedit.php ส่งมาให้) จากนั้นใช้ xml parser ของ php ในการอ่านข้อมูลจากไฟล์ XML และแปลงอิลิเมนต์ที่อ่านได้เป็น HTML code โดยให้แปลงเป็นฟอร์มที่มีข้อมูลจาก PCDATA ให้อยู่ใน input ซึ่งผู้ใช้สามารถแก้ไขได้ทุกช่อง ยกเว้น หัวข้อเอกสาร (หัวข้อภาษาไทยหรือประเด็นยุทธศาสตร์) รวมทั้งสามารถเปลี่ยนรูปประกอบใหม่ได้ (ชนิดและขนาดของรูปต้องตรงตามเงื่อนไข) เมื่อแก้ไขเสร็จจึงกดปุ่ม “Submit >>” เพื่อส่งข้อมูลที่แก้ไขแล้วให้ app_genxmlafteredit.php ทำการเขียนข้อมูลทับไฟล์เก่า



รูปที่ 6.22 หน้า app_editxmlfile.php สำหรับแก้ไขข้อมูลและรูปภาพ

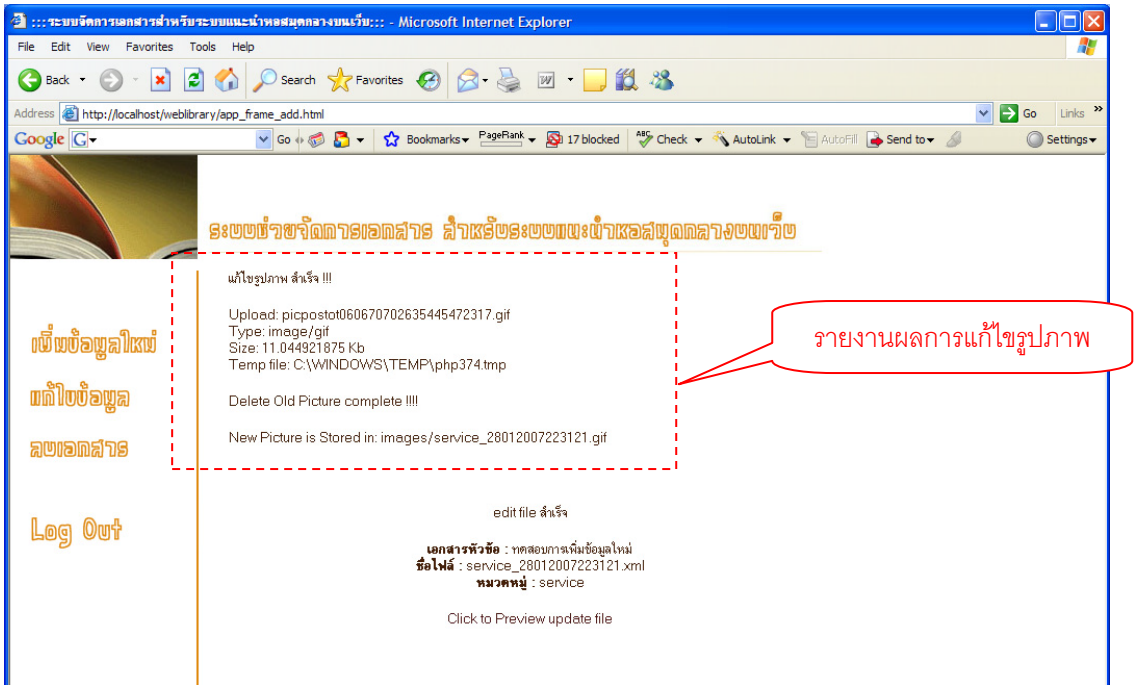
app_genxmlafteredit.php เป็นสคริปต์ที่ทำหน้าที่บันทึกข้อมูลที่แก้ไขจากฟอร์ม ลงในไฟล์ XML เดิม



รูปที่ 6.23 Folwchart สำหรับ app_genxmlafteredit.php

app_genxmlafteredit.php มีการทำงานคล้ายกับ app_savenewxml.php คือนำข้อมูลจากฟอร์มมาเขียนให้อยู่ในรูป XML แล้วนำมาเขียนทับไฟล์เดิม แต่มีความแตกต่างที่ ไม่มีการอัพเดทเมนูย่อย เนื่องจากมี link ในเมนูย่อยอยู่แล้ว (เป็นไฟล์เดิม) และสิ่งที่แตกต่างอีกประการคือมีการเพิ่มฟังก์ชัน deleteOldPic สำหรับลบไฟล์รูปเก่าหากตรวจสอบพบว่ามีการอัพโหลดไฟล์รูปใหม่ และยังคงกำหนดให้ไฟล์รูปภาพใหม่มีชื่อเหมือนกับชื่อไฟล์เอกสาร XML เหมือนเดิมด้วย

โดยเมื่ออัพเดทข้อมูลเรียบร้อยจะมีการรายงานผลการแก้ไข ดังแสดงในรูปที่ 6.24 โดยหากมีการแก้ไขรูปภาพจะแสดงผลการแก้ไขไฟล์รูปภาพด้วย และสามารถดูตัวอย่างผลลัพธ์ของเอกสาร XML ที่แก้ไขแล้วได้จาก “Click to Preview update file”



รูปที่ 6.24 แจ้งผลการอัพเดทเอกสาร XML และบันทึกรูปภาพ

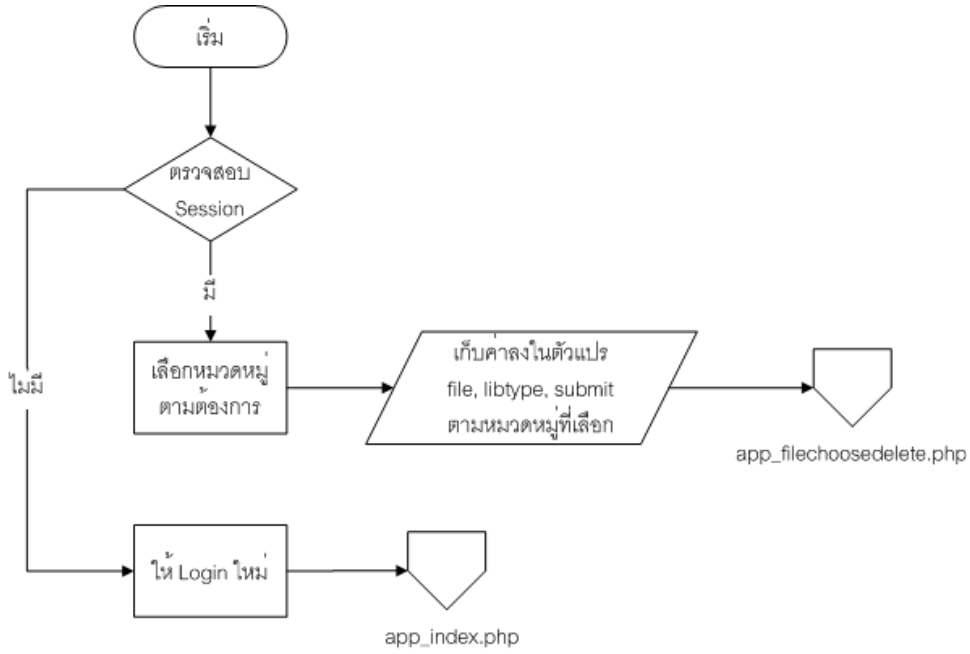
6.4 การลบเอกสาร

เป็นส่วนที่ช่วย ลบเอกสาร XML ในหัวข้อที่ไม่ต้องการ หรือหัวข้อที่ทางสำนักหอสมุดได้ยกเลิกการบริการในส่วนนั้นแล้ว ออกไปจากระบบ นอกจากนั้นยังจัดการ อัพเดทเมนูย่อย โดยลบ link ไปยังเอกสารที่ถูกลบ ออกไปจากเมนูย่อย และลบรูปภาพประกอบของไฟล์นั้น (ถ้ามี) ออกจากระบบด้วย

รูปที่ 6.25 Flow Chart ของการแก้ไขข้อมูล

การทำงานของโปรแกรมในส่วน การแก้ไขข้อมูล ประกอบด้วยสคริปต์ในการจัดการ 4 ไฟล์ ได้แก่

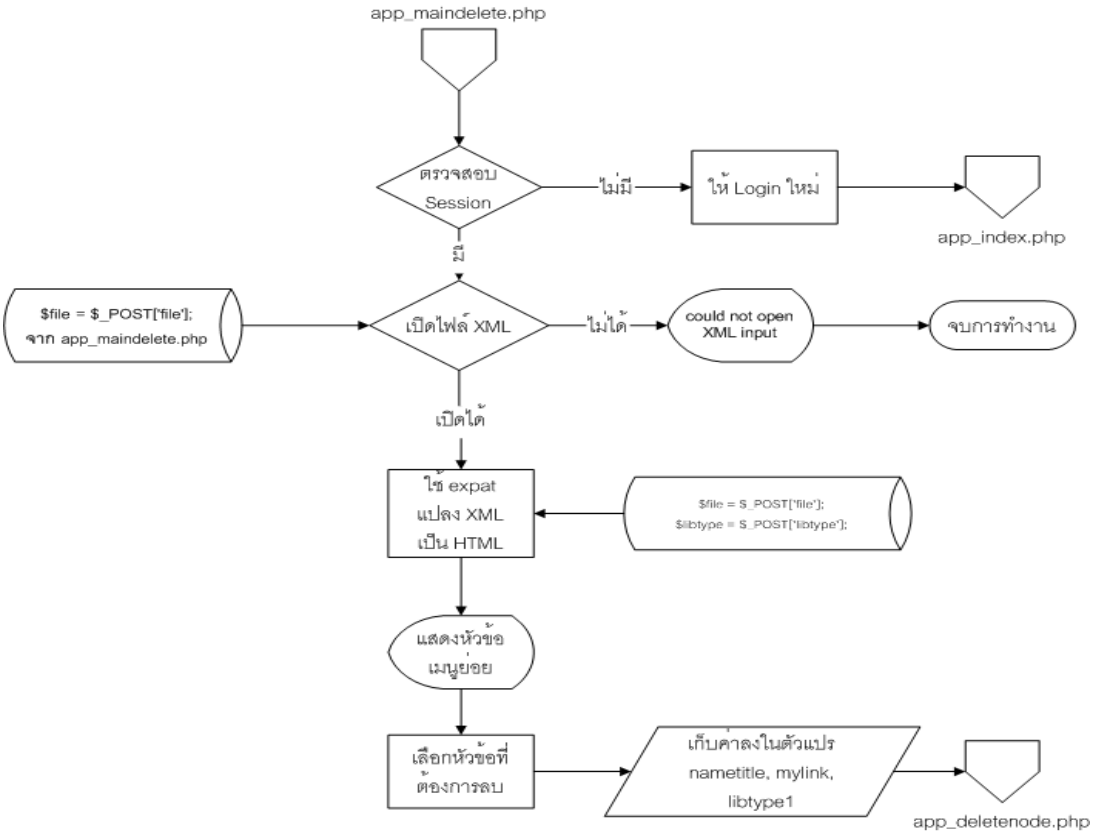
app_maindelete.php ทำหน้าที่แสดงรายการหมวดหมู่ของข้อมูล เพื่อให้ผู้ใช้เลือกหมวดหมู่ของข้อมูลที่ต้องการลบ มีการทำงานคล้ายกับ app_mainadd.php แต่ฟอร์มของ app_mainedit.php จะส่งชื่อไฟล์ของเมนูย่อย (เป็น xxx_link.xml) ไปกับตัวแปร file แบบเดียวกับ app_mainadd.php



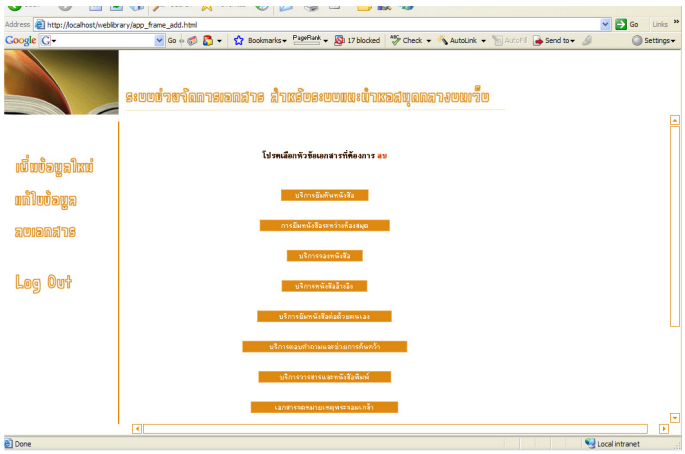
รูปที่ 6.26 Flowchart **สำหรับ** app_maindelete.php

รูปที่ 6.27 **ผลลัพธ์ของ** app_maindelete.php **บนบราวเซอร์** IE 6

app_filechoosedelete.php เป็นสคริปต์ที่ใช้ XML parser ของ PHP อ่านเอกสาร XML จากไฟล์เมนูย่อยของหมวดหมู่ที่เลือก และนำเมนูแสดงเป็นปุ่ม button ของแบบฟอร์ม ซึ่งมีการทำงานคล้ายกับ app_filechoose.php คือเมื่อกดเลือกหัวข้อจะเป็นการส่งชื่อไฟล์ XML ที่ต้องการลบไปยัง app_deletenode.php

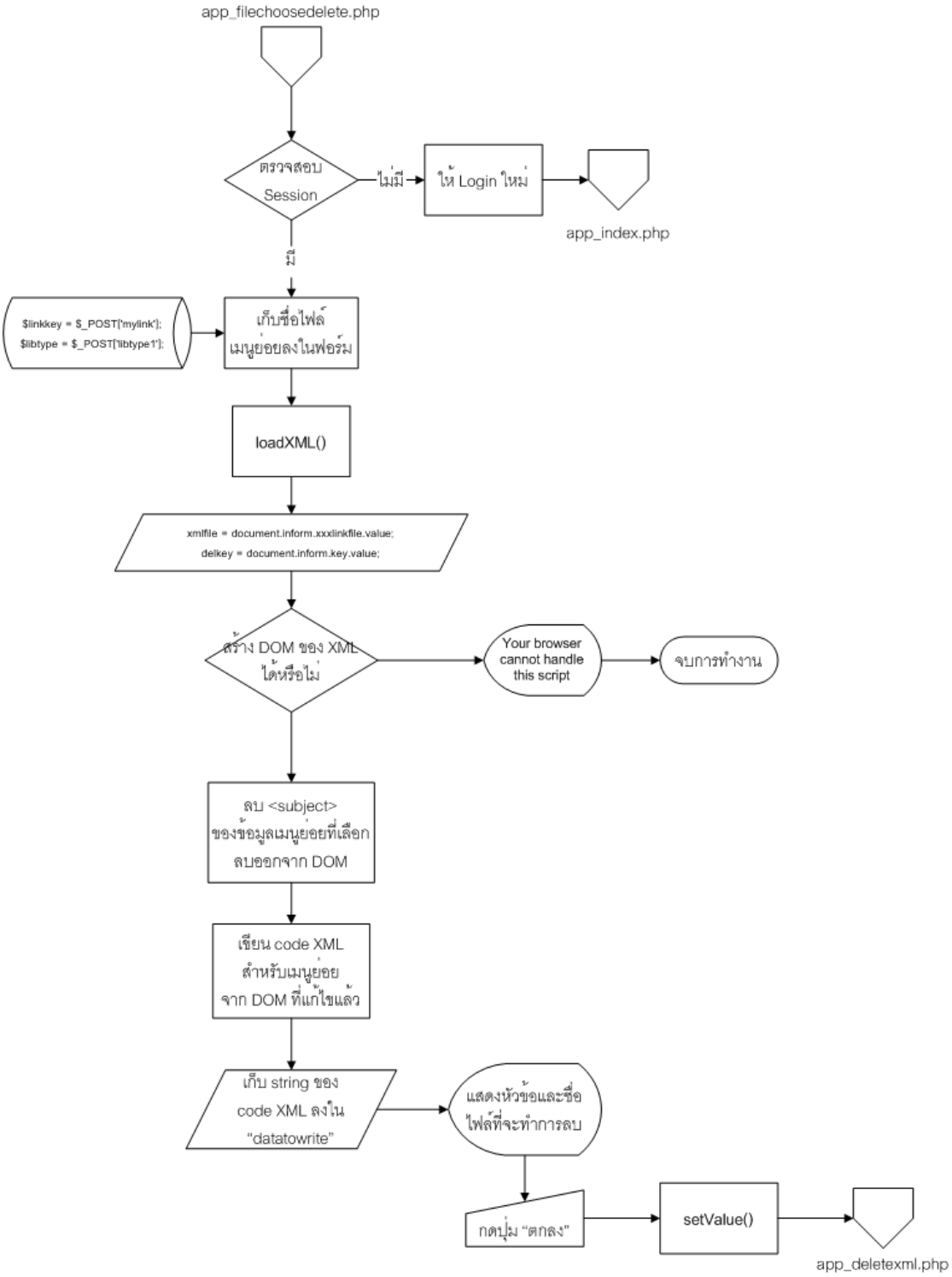


รูปที่ 6.28 Flowchart **สำหรับ** app_filechoosedelete.php



รูปที่ 6.29 **ผลลัพธ์ของ** app_filechoosedelete.php

app_deletenode.php ทำหน้าที่ลบ node <subject> ที่มีข้อมูลในโหนดลูกตรงกับหัวข้อที่ต้องการลบไฟล์ โดยเราไม่ทราบว่าโหนดที่เราต้องการลบ อยู่ตำแหน่งไหนในเอกสาร จึงใช้ JavaScript เพื่อสร้าง DOM มาช่วยในการเข้าถึงโหนดต่าง ๆ ในเอกสาร



รูปที่ 6.30 Flowchart **สำหรับ** app_deletenode.php

เนื่องจากมีการรับค่า file name ของเอกสารที่ต้องการจะลบและเอกสารที่เก็บเมนูย่อย ซึ่งส่งมาจาก app_filechoosedelete.php (ตัวแปร mylink และ libtype1) โดย PHP ซึ่งต้องส่งค่าเหล่านี้ให้ JavaScript เพื่อนำไปประมวลผลต่อ จึงใช้วิธีการส่งค่าผ่าน form ของ HTML แล้วเขียนฟังก์ชัน passKeyValue() ด้วย JavaScript มารับค่าจากฟอร์มอีกทีหนึ่ง

เมื่อ JavaScript มีชื่อไฟล์ที่ต้องการลบ (ตัวแปร delfile) และชื่อไฟล์ที่เก็บเมนูย่อย (ตัวแปร xmlfile) แล้ว จึงโหลด xml ที่ชื่อ xmlfile ลงใน DOM โดยใช้ฟังก์ชัน loadXML() ซึ่งวิธีการของ XML parser ใน IE และ Mozilla ไม่เหมือนกัน ในฟังก์ชันจึงมีการตรวจสอบประเภทของบราวเซอร์เพื่อใช้วิธีโหลดได้อย่างถูกต้อง

ในการ loadXML() จะทำให้ parser โหลด XML มาเก็บไว้ใน memory บนเครื่องของเราและทำให้สามารถจัดการข้อมูลได้ด้วย DOM ดังนั้น คำสั่งที่ใช้บนสคริปต์สำหรับเข้าถึงโหนดต่าง ๆ ของ DOM Tree จะใช้คำสั่งของ DOM ในการเข้าถึง

เมื่อโหลด XML มาแล้ว จะต้องหาโหนด <link> ที่มี PCDATA เท่ากับค่าในตัวแปร delfile ซึ่งสามารถหาได้จากฟังก์ชัน checkID(ชื่อโหนด, ค่า) ซึ่งจะทำการค้นหา “ลำดับ” ของโหนดนั้น ๆ ที่มีค่าเท่ากับค่าที่ส่งไปให้ และ return ลำดับที่ ที่เป็นตัวเลขมาให้

ในการลบเมนูย่อย ซึ่งเก็บข้อมูลเป็น <subject><name>หัวข้อ</name><link>ชื่อไฟล์</link></subject> เราจะต้องลบทั้ง <subject> และโหนดลูกทุกโหนดภายใต้ <subject> นั้น จึงนำตัวเลขลำดับที่ได้จากฟังก์ชัน checkID มาใช้ระบุตำแหน่งของ <subject> ที่ต้องการลบได้

จากนั้นหา root element (เก็บใน parent) แล้วใช้คำสั่ง parent.removeChild(x) เมื่อ x = xmlDoc.getElementsByTagName("subject")[id] (ชี้ไปยัง <subject> ที่ต้องการลบ) และ id คือลำดับที่ของโหนดที่ต้องการลบ

จากนั้นจึงใช้ creatDataXML(parent) จะทำให้ได้ DOM ใหม่ที่ลบโหนดที่ไม่ต้องการทิ้งไป แต่ DOM ดังกล่าวเป็นแค่ข้อมูลที่อยู่ใน memory บนเครื่องของผู้ใช้เท่านั้น

จนถึงขั้นตอนนี้เราจึงต้องเก็บข้อมูลใหม่ที่ DOM เขียนขึ้น ลงในตัวแปรเพื่อนำไปเขียนทับลงไฟล์ดั้งเดิมอีกครั้ง ซึ่งขั้นตอนในการเขียนไฟล์จะส่งต่อให้ app_deletexml.php

หมายเหตุ : ในขั้นตอนนี้ DOM ที่ลบโหนดแล้วยังไม่ได้ถูกบันทึกเป็นเอกสารเมนูย่อยที่แก้ไขแล้ว ดังนั้นหากผู้ใช้กดยกเลิกหรือเปลี่ยนไปทำกิจกรรมอื่น ๆ บนระบบ ข้อมูลในเมนูย่อยจะไม่เปลี่ยนแปลงและไฟล์จะยังไม่ถูกลบอีกด้วย

ในการส่งค่าข้อมูล XML ที่ใช้ DOM แก้ไขแล้วให้ app_deletexml.php ต้องส่งค่าผ่านฟอร์มของ HTML โดยใช้ฟังก์ชัน setValue() ในการกำหนดค่าให้ฟอร์มก่อนที่จะส่ง

app_deletexml.php ทำหน้าที่ลบเอกสาร XML ที่เป็นตัวข้อมูลออกจากระบบ รวมทั้งหากเอกสารนั้นมีไฟล์รูปภาพประกอบก็จะทำการลบไฟล์รูปภาพนั้นด้วย และยังทำหน้าที่ในการนำข้อมูลเมนูย่อยที่ได้รับการแก้ไขจาก app_deletenode.php แล้วมาเขียนทับข้อมูลเมนูย่อยเดิม (ทำการอัพเดทเอกสารที่เก็บข้อมูลเมนูย่อย)

รูปที่ 6.31 Flowchart **สำหรับ** app_deletexml.php

การทำงานของ app_deletexml.php แบ่งออกเป็น 3 ส่วน คือ

ส่วนที่จัดการเรื่อง remove <subject> และ child node จากเอกสารที่เก็บเมนูย่อย (xxx_link.xml) โดยนำข้อมูล XML ใหม่จากตัวแปร datafromjs ที่รับมาจากฟอร์มใน app_deletenode.php มาเขียนลงในไฟล์เมนูย่อยเดิม ด้วยคำสั่ง fopen, fwrite และ fclose นั่นเอง

ส่วนที่สอง จัดการเรื่องลบไฟล์ข้อมูลนั้น ออกจากระบบ ก็ใช้คำสั่ง unlink(“*ชื่อไฟล์ที่ต้องการลบ*”) โดยชื่อไฟล์ที่ต้องการลบก็รับค่ามาจาก app_deletenode.php

ส่วนที่สาม จะตรวจสอบว่ามีรูปภาพใดในโฟลเดอร์ images มีชื่อไฟล์เหมือนกับชื่อเอกสารที่ต้องการลบ ถ้ามีก็จะทำการลบรูปภาพนั้นผ่านการเรียกใช้ฟังก์ชัน delPic (รูปภาพประจำเอกสาร จะมีชื่อเดียวกับเอกสาร)

6.5 การทดสอบระบบช่วยจัดการเอกสารสำหรับระบบแนะนำหอสมุดกลาง

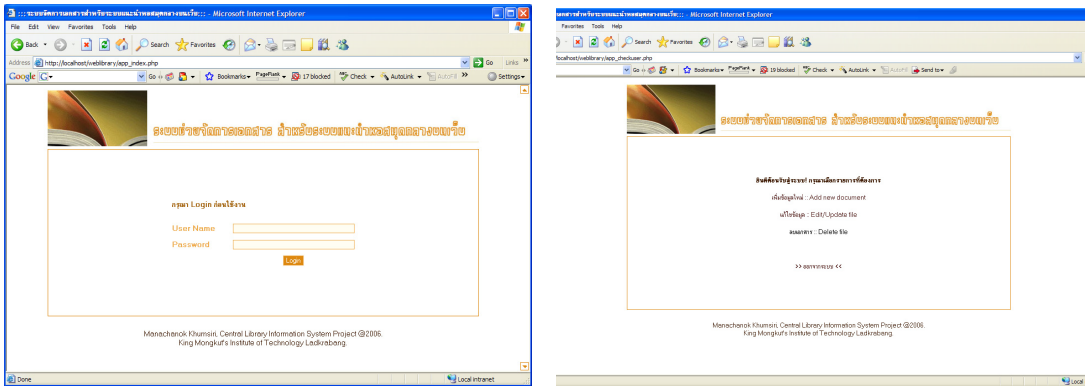
เพื่อเป็นตัวอย่างสำหรับการใช้งานจริงสำหรับผู้ดูแลระบบ จึงขอนำเสนอการทดสอบระบบช่วยจัดการเอกสารสำหรับระบบแนะนำหอสมุดกลางบนเว็บ ประกอบคำแนะนำในการใช้งาน โดยจะใช้ข้อมูลในตารางที่ 6.1 ในการเพิ่มข้อมูลใหม่ ในหมวดหมู่ การบริการของสำนัก

ตารางที่ 6.1 ข้อมูลที่ต้องการนำไปแสดงในระบบแนะนำหอสมุดกลางบนเว็บ

หมวดหมู่	บริการของสำนัก
หัวข้อ	ทดสอบการเพิ่มข้อมูลใหม่
หัวข้อภาษาอังกฤษ	Test Add New XML
คำอธิบายรูปภาพ	ทดสอบรูปภาพ
รูปภาพประกอบ	D:\project49\library.jpg
คำอธิบายโดยย่อ	กรอกคำอธิบายโดยย่อ
รายละเอียดการบริการ	กรอกรายละเอียดการบริการ
ระเบียบการใช้บริการ	
เวลาให้บริการ	เวลาทำการ
สถานที่ให้บริการ	อาคารเฉลิมพระเกียรติ
Link ที่เกี่ยวข้อง	

ในการเริ่มต้นระบบช่วยจัดการเอกสารสำหรับระบบแนะนำหอสมุดกลางบนเว็บ จะต้องเริ่ม login เพื่อเข้าระบบจากหน้าหลักของระบบก่อน โดยเข้าถึงได้จากไฟล์ “app_index.php”

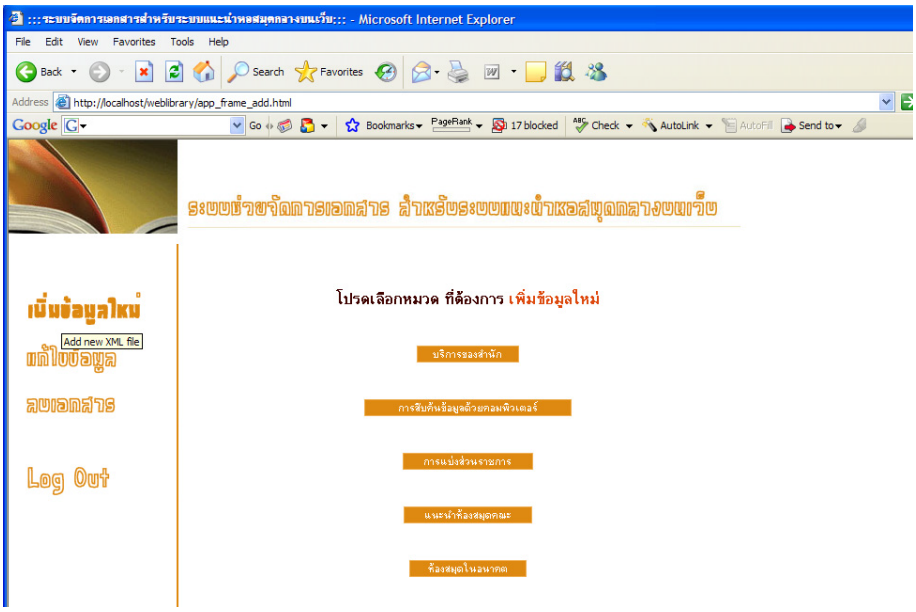
เมื่อ login ได้แล้ว ผู้ใช้สามารถเลือกที่จะเพิ่มข้อมูลใหม่ แก้ไขข้อมูล หรือลบเอกสารได้



รูปที่ 6.32 แสดงหน้าหลักสำหรับ Login และหน้าแรกเมื่อ Login สำเร็จ

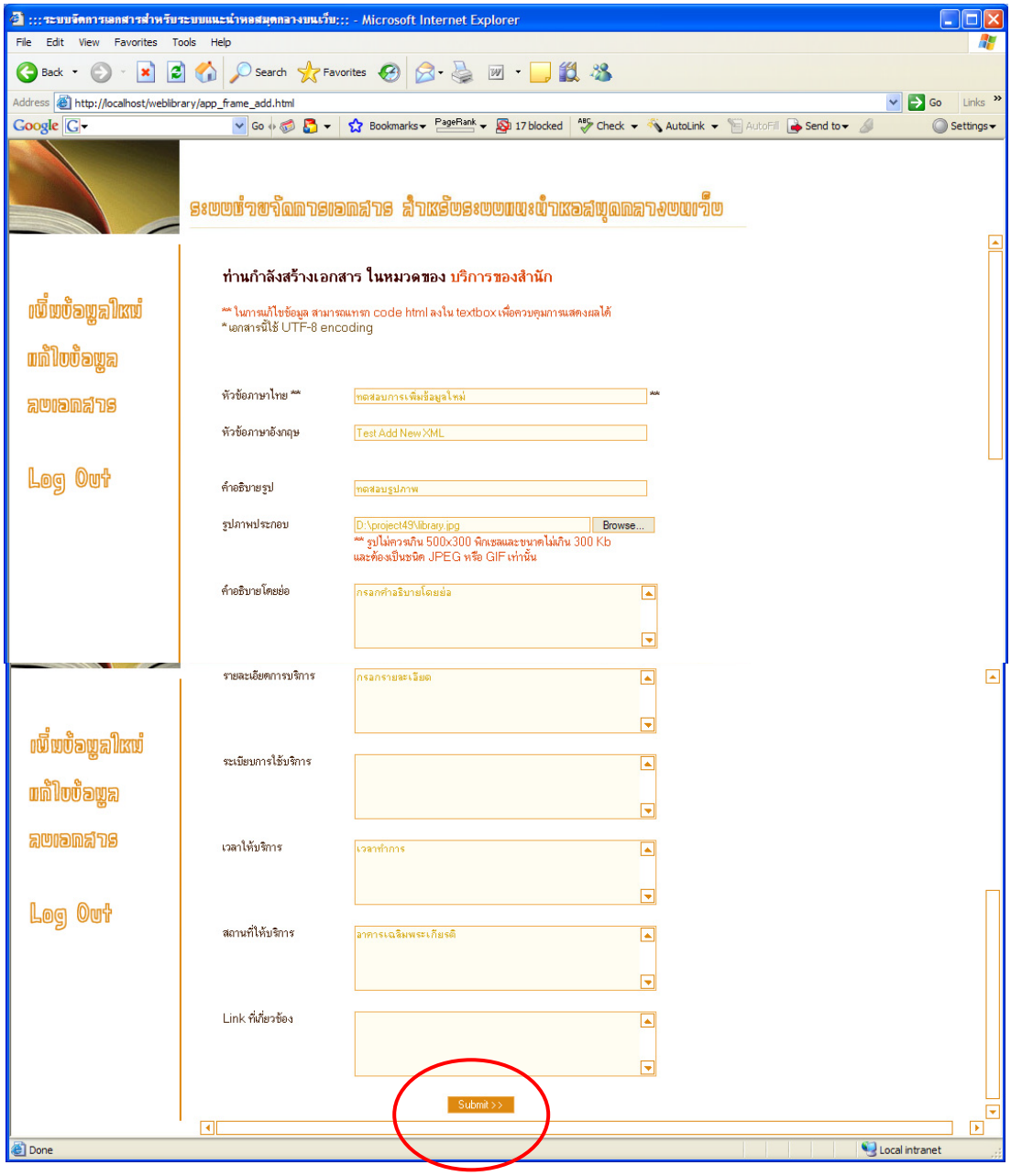
6.5.1 การเพิ่มข้อมูลใหม่

1. เลือกหมวดหมู่ที่ตรงกับข้อมูลใหม่



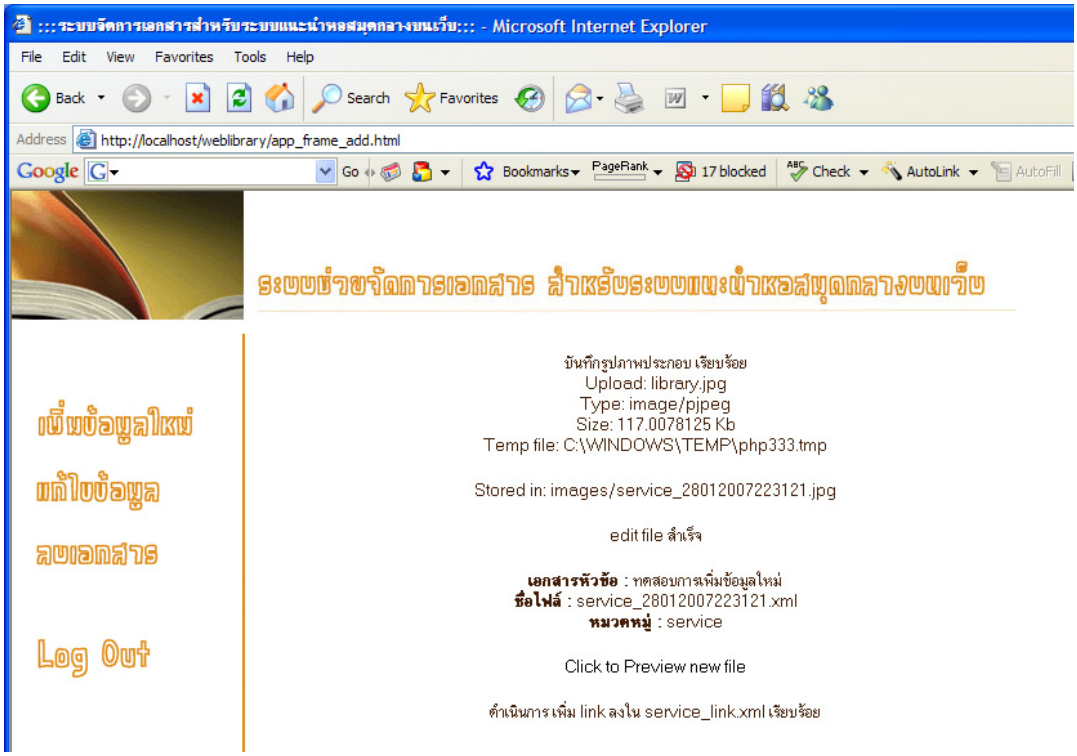
รูปที่ 6.33 แสดงหน้าแรกสำหรับการเพิ่มข้อมูลใหม่

2. กรอกข้อมูลลงในแบบฟอร์ม พร้อมทั้งเลือกรูปที่ต้องการ upload เมื่อกรอกเสร็จกดปุ่ม “Submit >>” ด้านล่าง

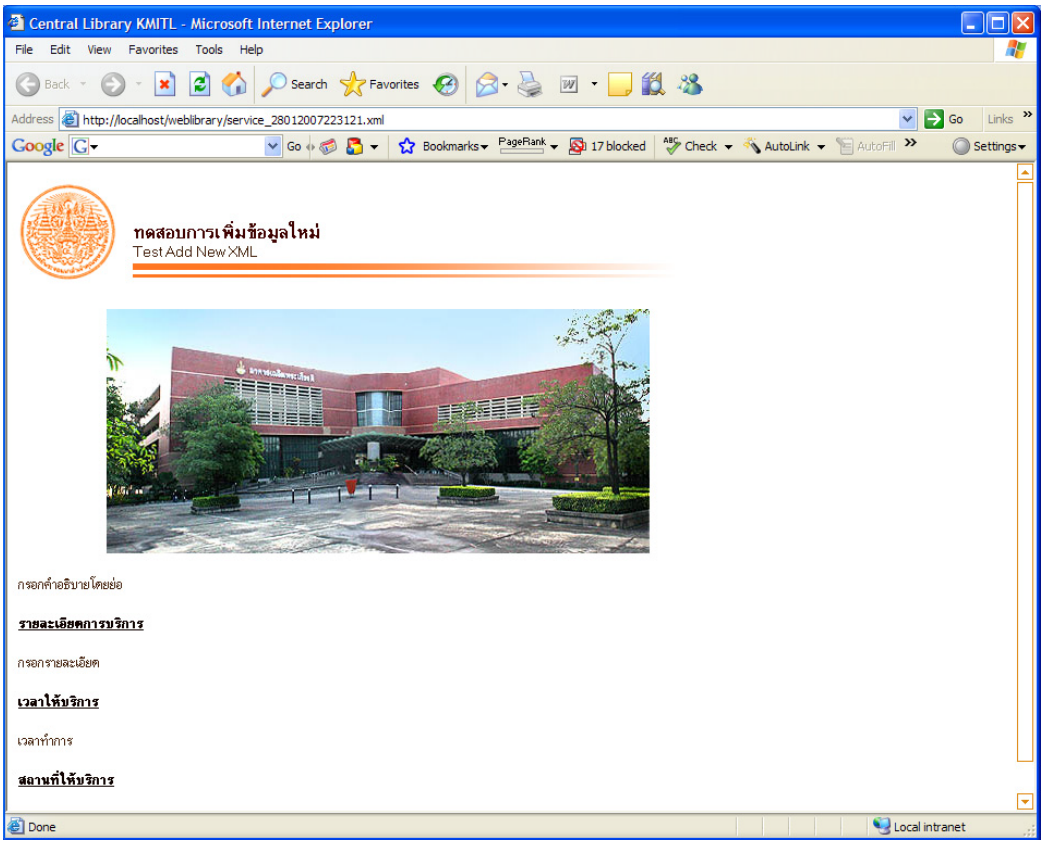


รูปที่ 6.34 แสดงแบบฟอร์มสำหรับกรอกข้อมูลใหม่

3. ระบบจะตั้งชื่อให้กับเอกสาร ไฟล์รูปภาพ และแปลงข้อมูลที่รับมาจากฟอร์มเป็น XML แล้วจัดเก็บลงในระบบ โดยระบบจะแสดงผลการ การจัดเก็บไฟล์เอกสาร ซึ่งตามตัวอย่างนี้เอกสารมีชื่อว่า “service_28012007223121.xml” และจัดเก็บไฟล์รูปภาพลงใน “images/service_28012007223121.jpg” อีกทั้งทำการอัพเดทเมนูย่อยในส่วนของบริการของสำนักให้ เป็นอันเสร็จสิ้นการเพิ่มข้อมูลใหม่ เราสามารถดูผลลัพธ์จากการเพิ่มข้อมูลใหม่ด้วยการ click ที่ “Click to Preview new File”

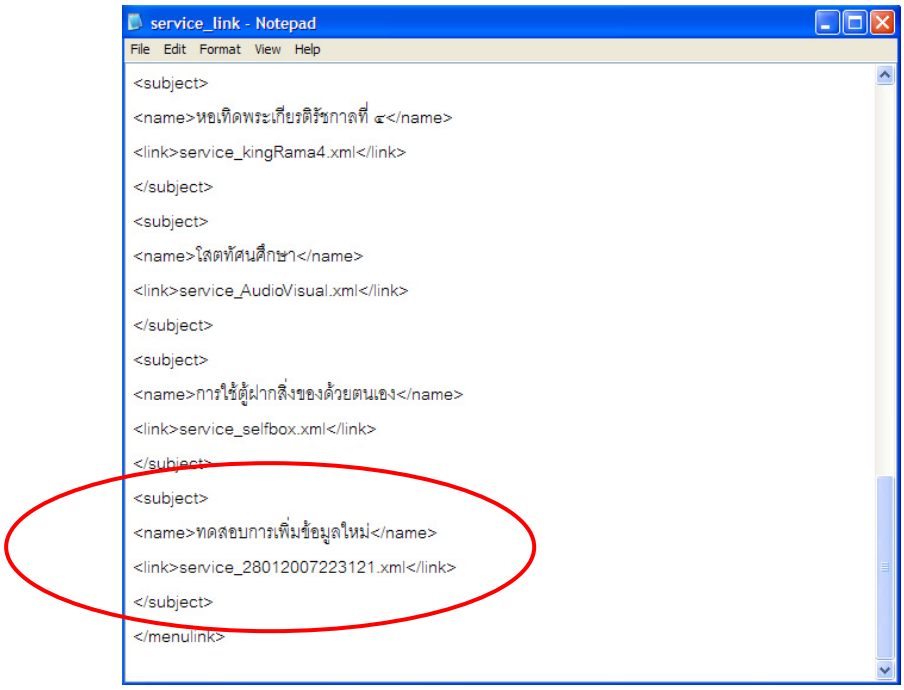


รูปที่ 6.35 หน้าแสดงผลการบันทึกข้อมูลใหม่

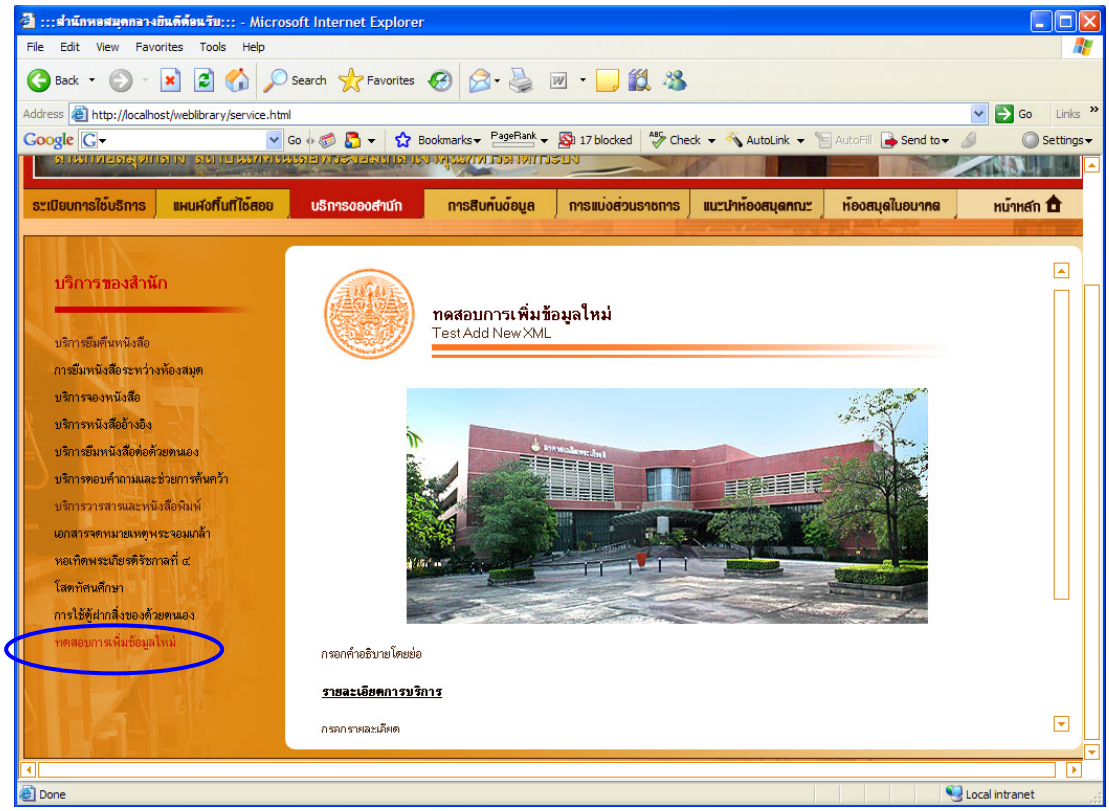


รูปที่ 6.36 ตัวอย่างผลลัพธ์จากเอกสารข้อมูลที่เพิ่มลงในระบบ

เมื่อเปิดตรวจสอบข้อมูลใน “service_link.xml” พบว่ามีการเพิ่ม link ของข้อมูลใหม่



รูปที่ 6.37 source code ของ “service_link.xml” ที่ได้รับการอัพเดทหลังการเพิ่มข้อมูลใหม่



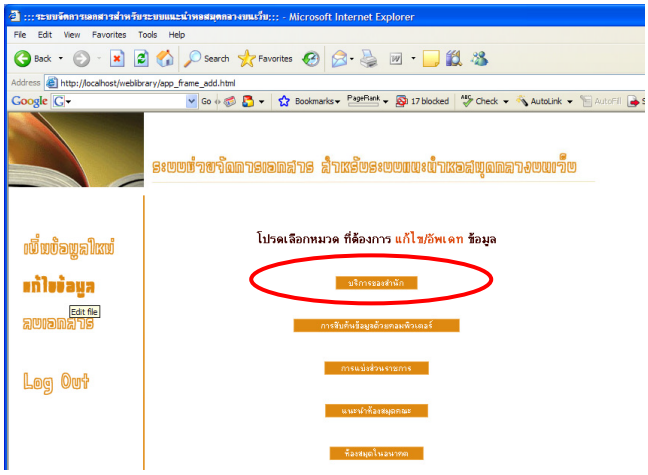
รูปที่ 6.38 ผลลัพธ์บนระบบแนะนำหอสมุดกลางบนเว็บสำหรับข้อมูลใหม่

6.5.2 **ทดสอบการแก้ไขข้อมูล** โดยแก้ไขข้อมูลจากเอกสารเดิม เป็นข้อมูลใหม่ตามตารางที่ 6.2 (การแก้ไขข้อมูลไม่สามารถแก้ไขในส่วนของ “หัวข้อ**” ได้)

ตารางที่ 6.2 แสดงข้อมูลที่ต้องการ แก้ไขด้วยตัวอักษรสีแดง

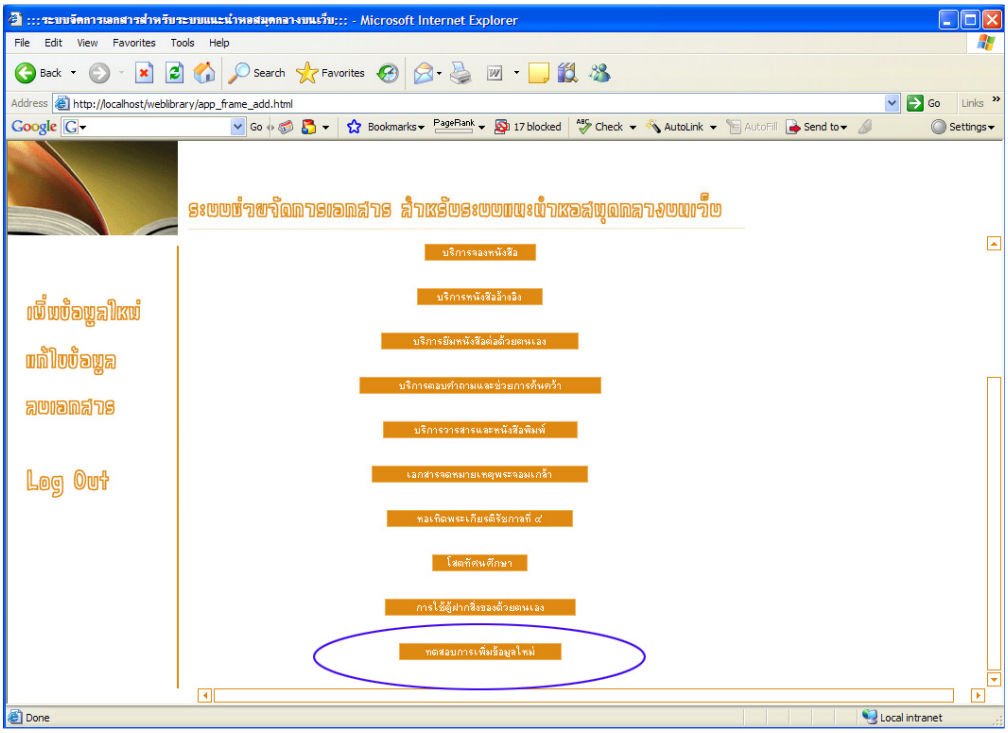
หมวดหมู่	บริการของสำนัก
หัวข้อ**	ทดสอบการเพิ่มข้อมูลใหม่
หัวข้อภาษาอังกฤษ	Test Add New XML
คำอธิบายรูปภาพ	ทดสอบรูปภาพ
รูปภาพประกอบ	D:\MyPic\picpostot06067070263544547231 7
คำอธิบายโดยย่อ	แก้ไข แก้ไข แก้ไข แก้ไข แก้ไข แก้ไข แก้ไข
รายละเอียดการบริการ	กรอกรายละเอียดการบริการ
ระเบียบการใช้บริการ	
เวลาให้บริการ	เวลาทำการ
สถานที่ให้บริการ	อาคารเฉลิมพระเกียรติ
Link ที่เกี่ยวข้อง	

1. เลือกการแก้ไขข้อมูล และเลือกหมวดหมู่ บริการของสำนัก (เลือกหมวดของข้อมูลที่ต้องการแก้ไข)



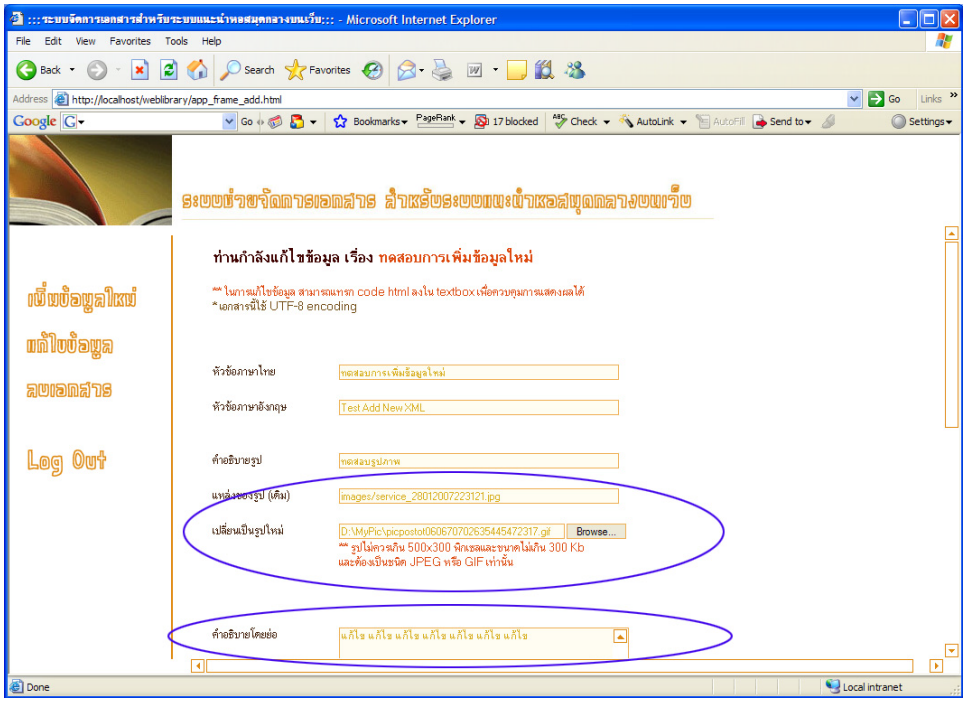
รูปที่ 6.39 **หน้าหลักของการแก้ไขข้อมูล ให้เลือกหมวดหมู่ที่ต้องการแก้ไข**

2. เลือกหัวข้อที่ต้องการแก้ไข ในที่นี้คือหัวข้อ “ทดสอบการเพิ่มข้อมูลใหม่”



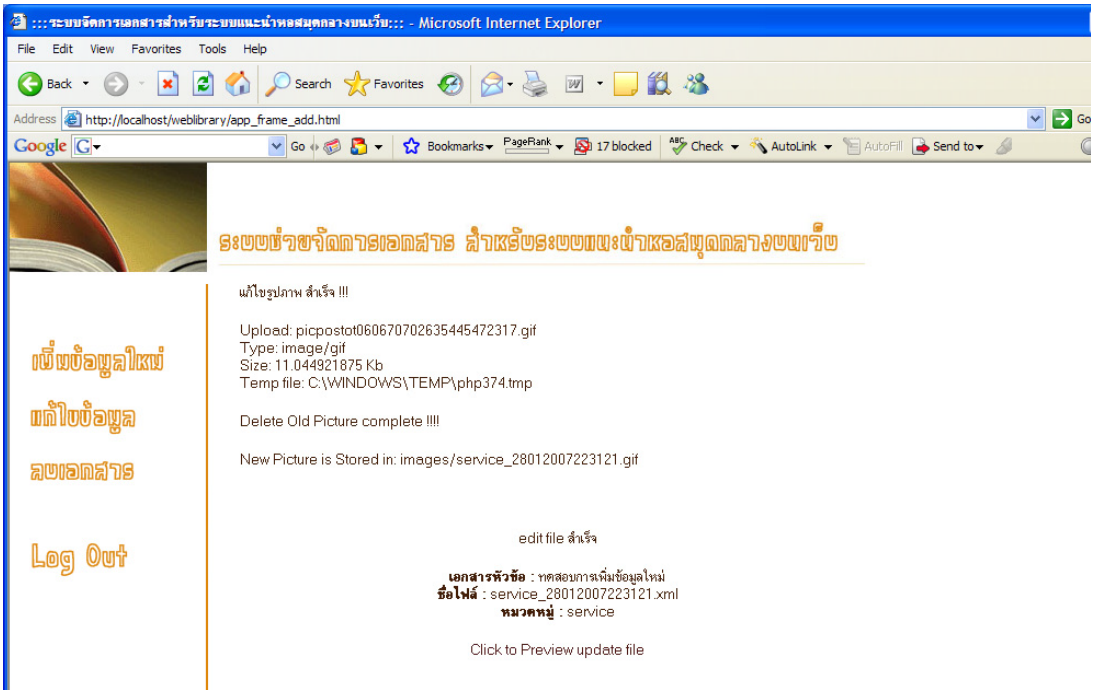
รูปที่ 6.40 แสดงการเลือกหัวข้อทีต้องการแก้ไข

3. แก้ไขข้อมูลตามต้องการลงในแบบฟอร์ม เมื่อเสร็จแล้วกดปุ่ม “Submit >>”

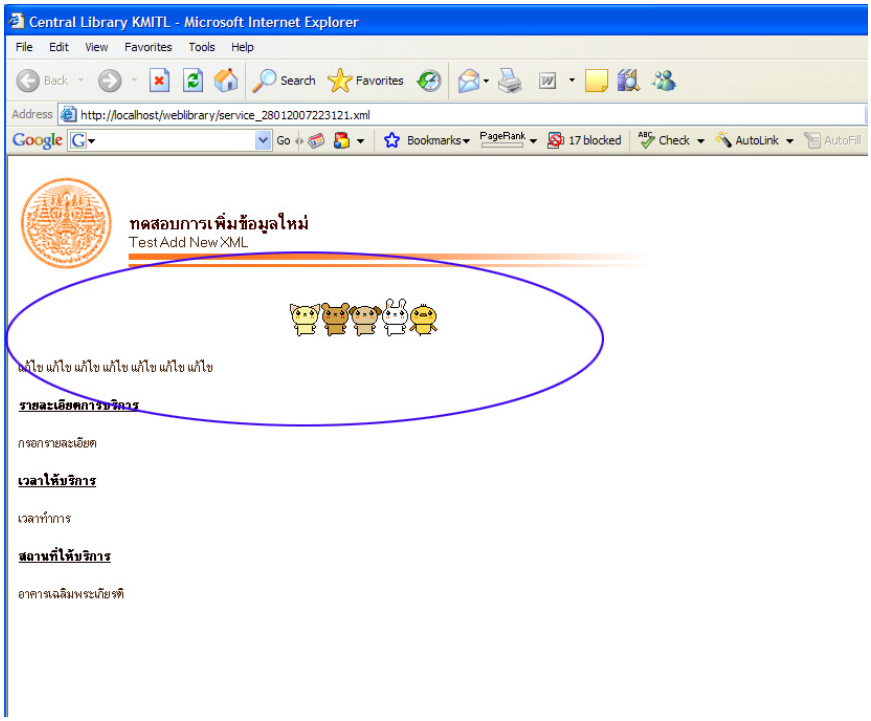


รูปที่ 6.41 แสดงตัวอย่างการแก้ไขข้อมูล

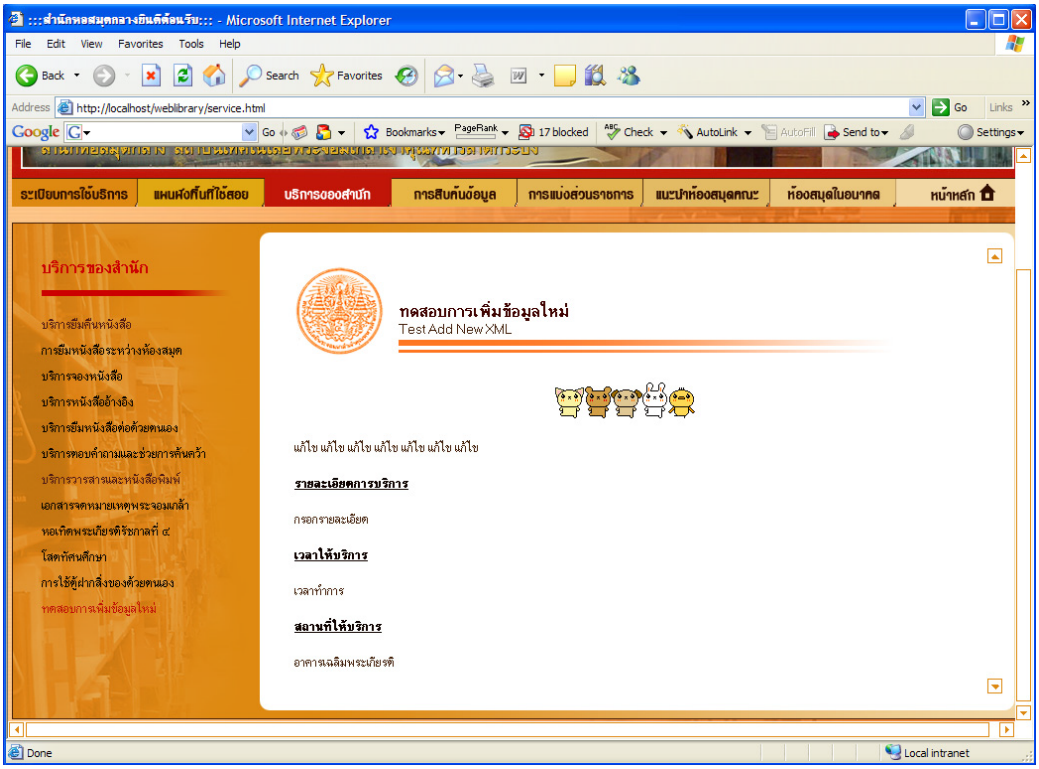
4. ระบบจะแก้ไขเอกสาร “service_28012007223121.xml” ให้มีข้อมูลใหม่ตามที่ได้แก้ไขในแบบฟอร์ม รวมทั้งการลบรูปเก่าและอัพโหลดรูปใหม่ โดยเราสามารถดูผลลัพธ์จากการแก้ไขข้อมูลด้วยการคลิ๊กที่ “Click to Preview update File”



รูปที่ 6.42 รายงานผลการแก้ไขรูปข้อมูลและรูปภาพ



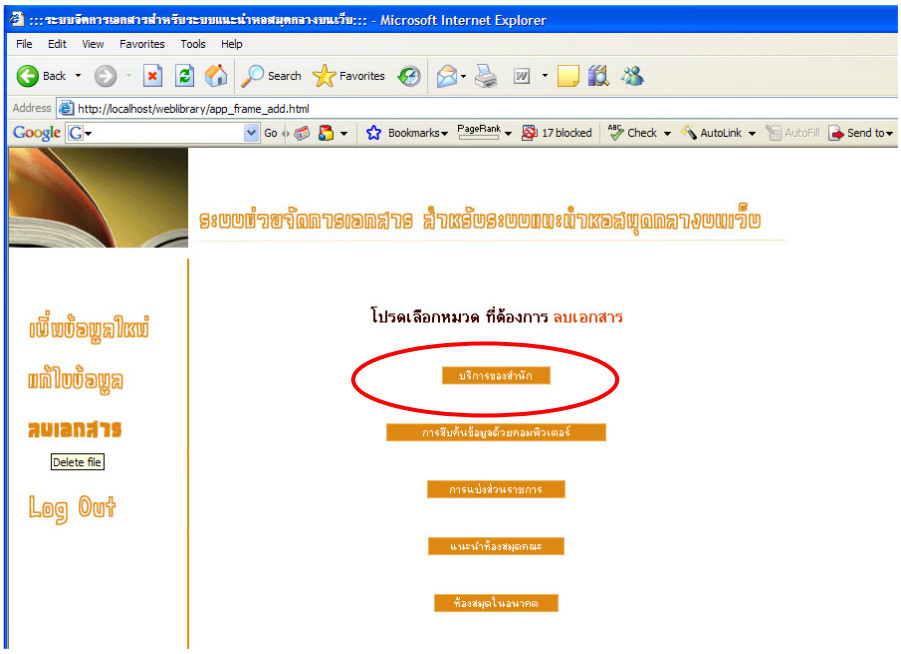
รูปที่ 6.43 ผลลัพธ์ของเอกสารที่แก้ไขแล้ว จากการ “Click to Preview new File”



รูปที่ 6.44 ผลลัพธ์บนระบบแนะนำหอสมุดกลางบนเว็บหลังแก้ไขข้อมูลใหม่

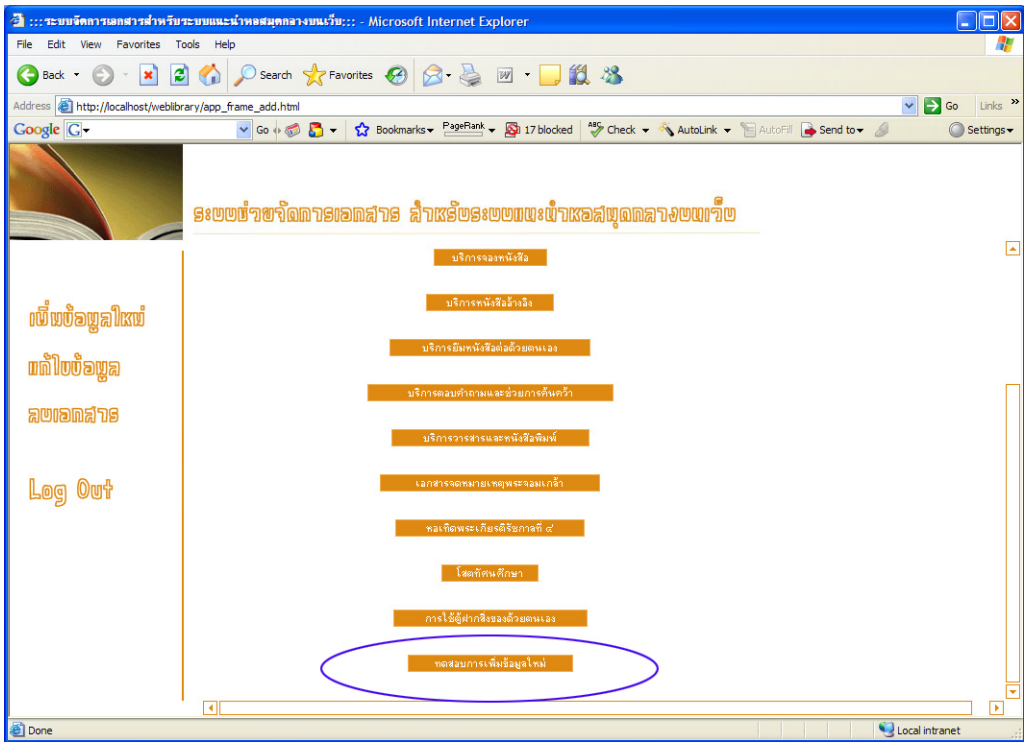
6.5.3 **การลบเอกสาร** เมื่อไม่ต้องการให้ข้อมูลแสดงบนระบบอีกต่อไป สามารถใช้การลบเอกสารเพื่อลบเอกสารข้อมูล รูปภาพประกอบและ link ในเมนูย่อยได้ โดยมีขั้นตอนดังนี้

1. เลือกการลบเอกสาร และเลือกหมวดหมู่ที่ต้องการลบ



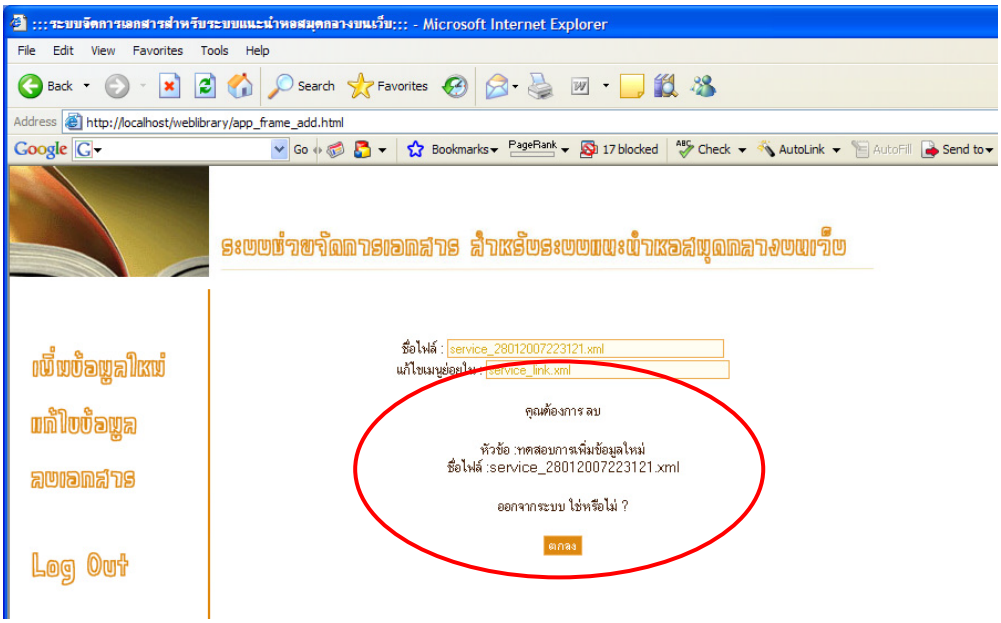
รูปที่ 6.45 แสดงการเลือกหมวดหมู่ในการลบเอกสาร

2. เลือกหัวข้อที่ต้องการลบไฟล์เอกสารออกจากระบบ



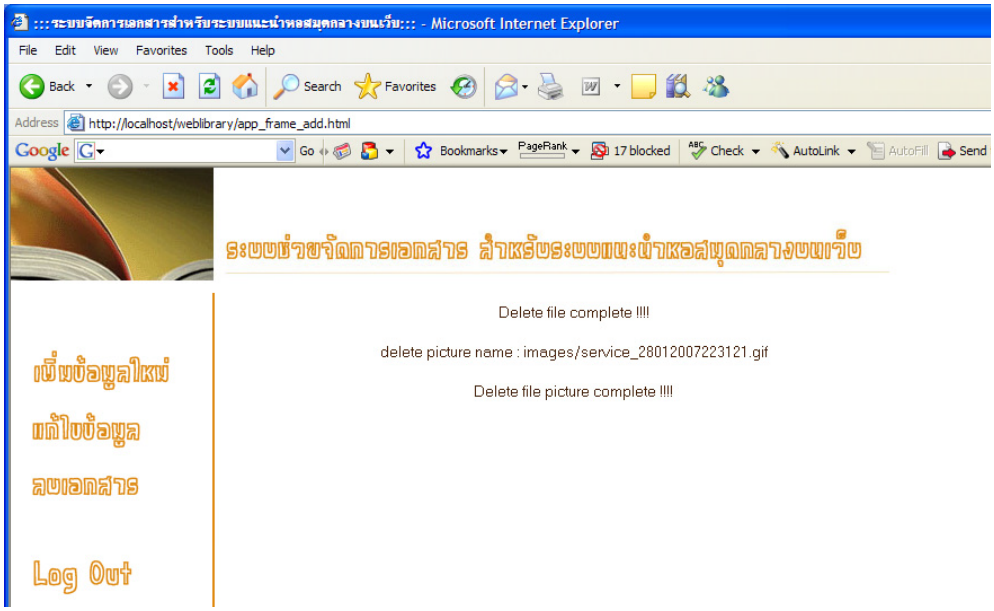
รูปที่ 6.46 เลือกหัวข้อที่ต้องการลบเอกสาร

3. ระบบจะถามว่าต้องการลบใช่หรือไม่ ถ้าต้องการกดตกลง ถ้าไม่ต้องการ สามารถเลือกเมนูลัดทางด้านซ้ายมือเพื่อจัดการเรื่องอื่น ๆ ได้ทันที



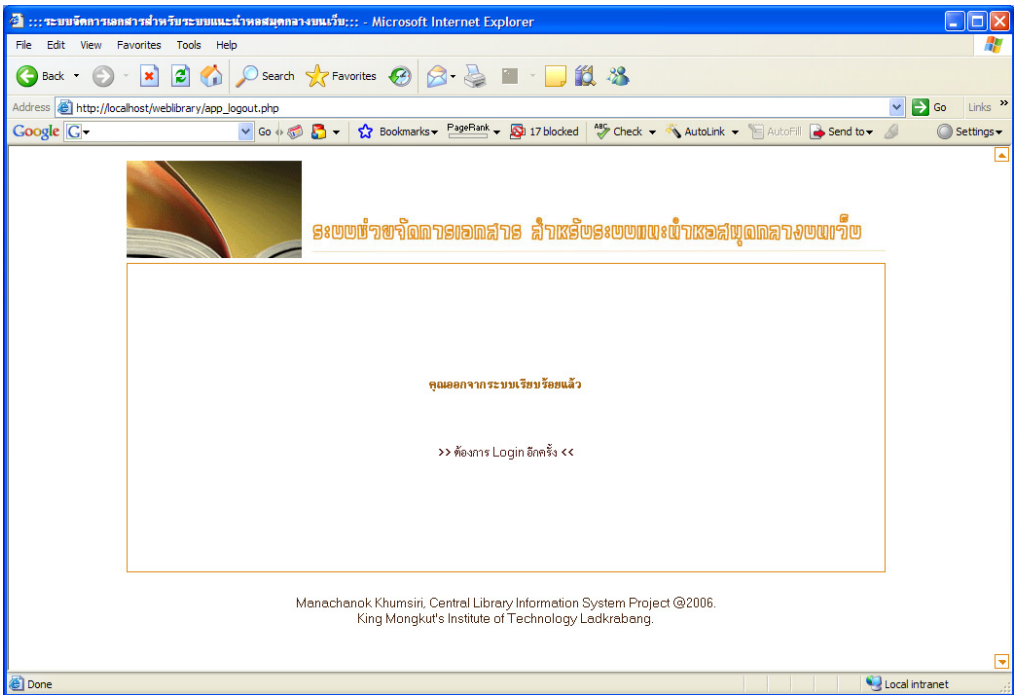
รูปที่ 6.47 ยืนยันข้อมูลในการลบเอกสาร

4. หากกดตกลง ระบบจะทำการ delete XML file ชื่อ ”service_28012007223121.xml” และลบรูปประกอบของเอกสาร นั่นคือ “service_28012007223121.gif” ออกจากระบบ และแก้ไขไฟล์ service_link.xml โดยลบเมนูย่อยในหัวข้อ “ทดสอบการเพิ่มข้อมูลใหม่” โดยอัตโนมัติ



รูปที่ 6.48 แสดงผลการลบเอกสารสำเร็จ

เมื่อเลิกใช้งานระบบช่วยจัดการเอกสาร สามารถ Log outได้จาก Link ทางด้านซ้ายมือ

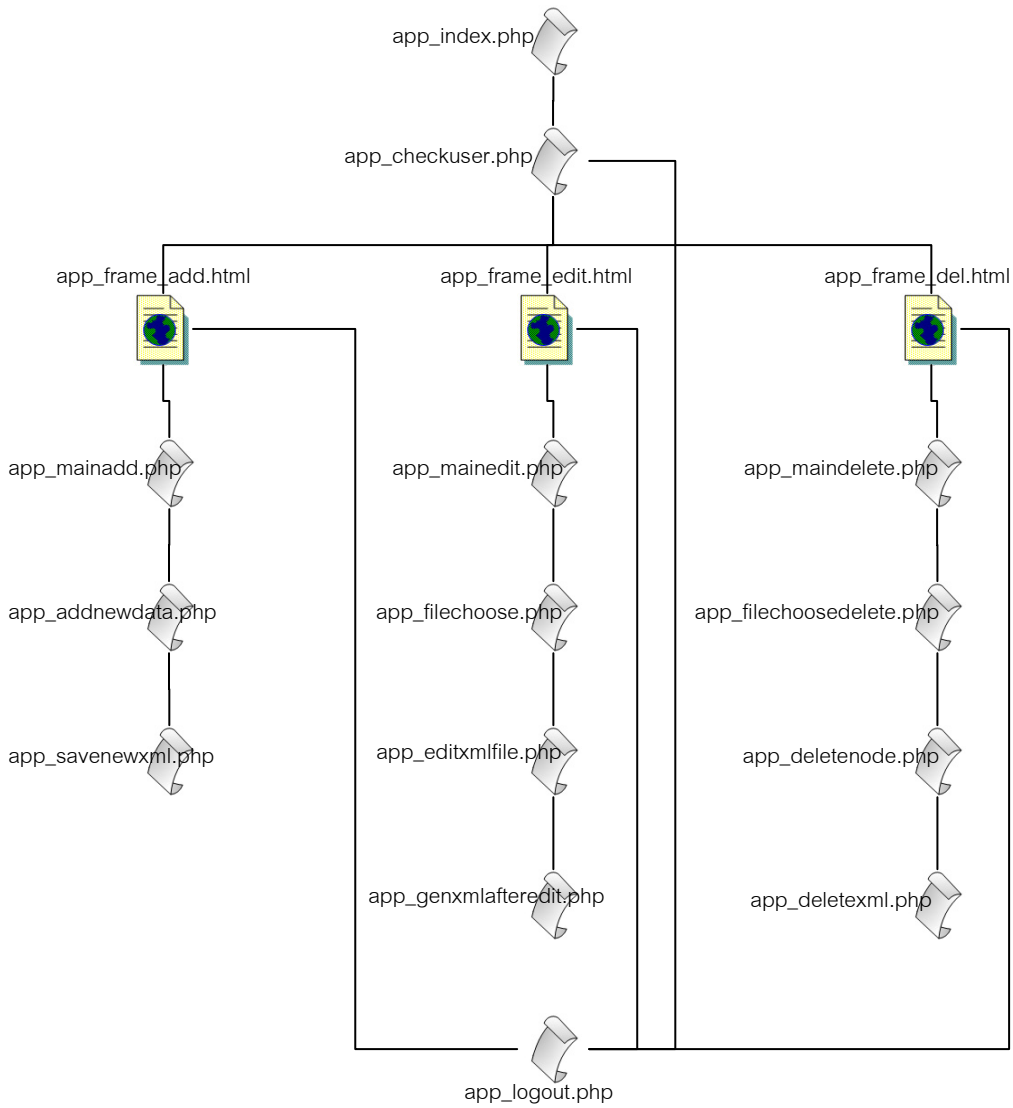


รูปที่ 6.49 ยืนยันผลการออกจากระบบ

6.6 สรุปการพัฒนาระบบช่วยจัดการเอกสารสำหรับระบบ

ระบบช่วยจัดการเอกสารสำหรับระบบ เป็น web application ที่ช่วยจัดการข้อมูลในเอกสาร XML ที่อยู่ในหมวดหมู่ บริการของสำนักหอสมุดกลาง, การสืบค้นข้อมูลด้วยคอมพิวเตอร์, การแบ่งส่วนราชการ (ยกเว้นหัวข้อแผนภูมิโครงสร้าง), แนะนำห้องสมุดคณะ, ห้องสมุดในอนาคต

โดยสามารถเพิ่มข้อมูลใหม่, แก้ไข/อัพเดทข้อมูล และลบเอกสารออกจากระบบได้ โดยระบบช่วยจัดการเอกสารสำหรับระบบ ไม่สามารถเข้าถึงได้จาก link ในส่วนใด ๆ ของระบบแนะนำหอสมุดกลางบนเว็บ โดยจะต้องเข้าถึงผ่าน (*http://localhost/*)app_index.php เท่านั้น และผู้ที่มีสิทธิในการใช้งานคือผู้ดูแลระบบที่ทราบรหัสผ่าน จึงสามารถเข้าไปจัดการเอกสารในระบบได้



รูปที่ 6.50 โครงสร้างความสัมพันธ์เอกสารในระบบช่วยจัดการเอกสาร