KXT209

Assignment 2 Specification

Contents

ASSIGNMENT 2 SPECIFICATION 2

INTRODUCTION 2

DUE DATE AND SUBMISSION METHOD 2

MARKING 2

SYSTEM DESCRIPTION 3

MARKING INFORMATION 19

HELP AND HINTS 21

PLAGIARISM AND CHEATING 21

SUBMISSION DETAILS 22

CHECKLIST 22

ASSIGNMENT 2 SPECIFICATION

INTRODUCTION

The aim of this assignment is to complete the implementation of a "Computing and Information Systems Library" website.
In the website, users will be able to interact with the database at three levels:

- "Public" – for users who have not signed up or logged in to the CIS Library website.
- "General Member" – for users who have logged in to the CIS Library website as a general member.
- "Administrator" - for users who have logged in to the CIS Library website as an administrator member.

For more detailed information about the components of the system, you should read **'SYSTEM DESCRIPTION'** carefully from page 2 to 18.
A demo video for assignment is available on MyLO. If you are unsure about the assignment specification, please watch the demonstration video.

DUE DATE AND SUBMISSION METHOD

Due date: Friday 24 May 3PM

NOTE: Submission will **not** be through MyLO, it will be made using the School of Computing & Information Systems submission system (see **SUBMISSION DETAILS** on page 22).

Late Submissions

- Late assignments will only be accepted in exceptional circumstances and provided that the proper procedures have been followed (see the School Office or the CIS web site for details). Assignments that are submitted late without lecturer's approval will be subject to mark penalties if they are accepted at all (see School office or web site for details on this as well).
- Forms to request extensions of time to submit assignments are available from the School of Computing office. Requests must be accompanied by suitable documentation and should be submitted before the assignment due date.

MARKING

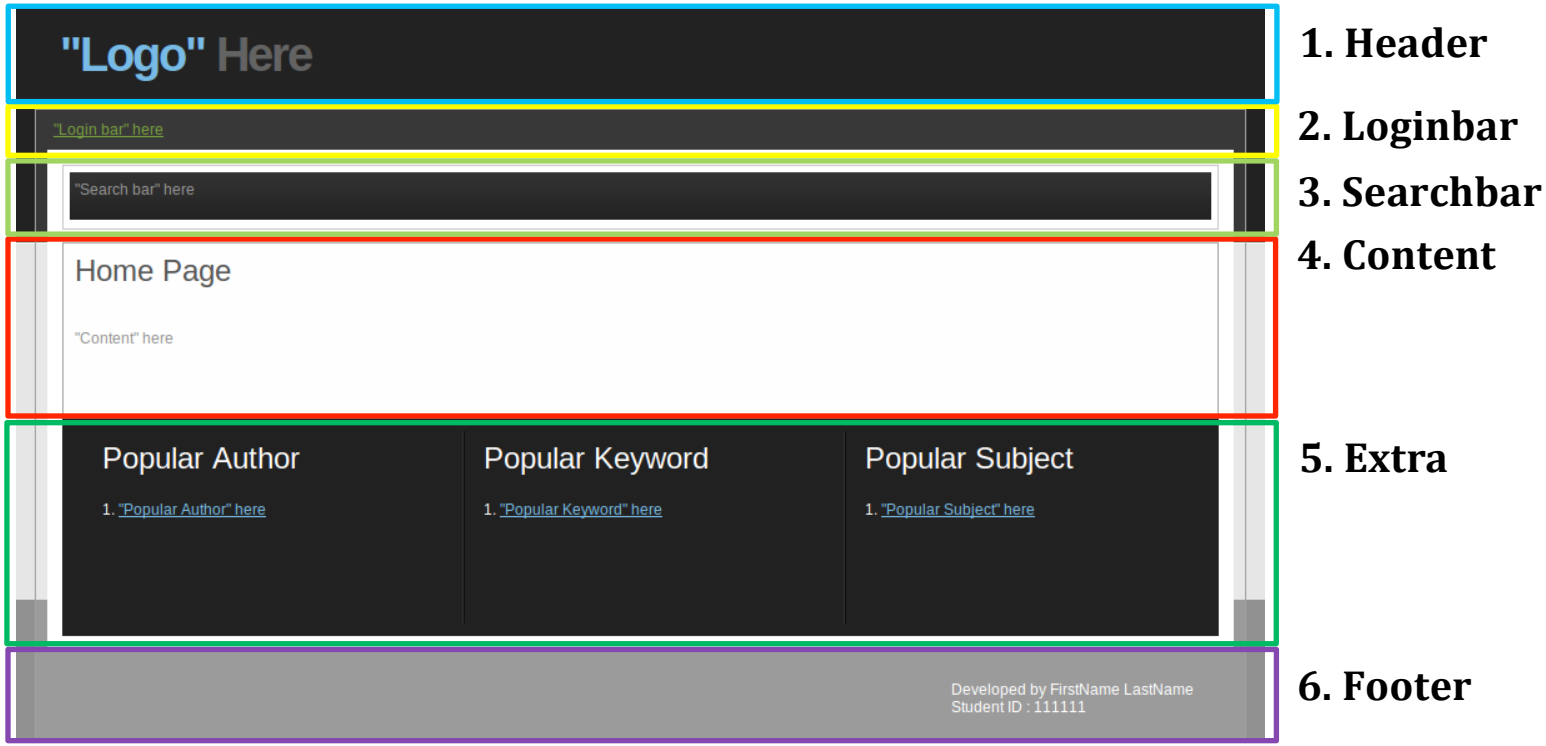
This assignment carries 20% of the marks for this unit. Remember that you must obtain at least 45% of the available marks for the in-semester assessment component to be able to obtain a passing grade in this unit. Details of the marking scheme are given in the **'MARKING INFORMATION'** on page 19).

SYSTEM DESCRIPTION

File Structure

There is no restriction for the creating new files.

Basic Layout



'Header', **'Login'**, **'Search'**, **'Extra'** and **'Footer'** section should be same in every page.
'Content' should be appropriate for each page.

Detailed specification for each section

There is a demonstration video clip on MyLO. If you are unsure what to do, please see the demo video.

1. **Header**: Leave this section as you have implemented in the assignment 1.
2. **Login bar**: Leave this section as you have implemented in the assignment 1.
3. **Search bar**: Leave this section as you have implemented in the assignment 1.
4. **Content**: It should contain the relevant information for each page. Each page should have an appropriate **'Content Title'** (i.e. Home, Search Result, Book details, etc.). The appearance of the pages should not be exactly same as demo, but the appearance should be reasonable.

4.1. **'Home'** page: Leave this page as you have implemented in the assignment 1.

4.2. **'Search result'** page

The page should retrieve a list of books (display 'Title' and 'Author' of the book) that match a search term in the CIS Library database. For the assignment 2, every search term that contains space should be exploded by space. For example, if the user searches 'web services' with 'Title' option, we need to separate the word 'web service' into two words: 'web' and 'services' (not like assignment 1).

[Tips] When you do the query for retrieving the result, you may use the following query.

- Title, Subject and Keyword search:
  "`SELECT … FROM … WHERE … `Title` LIKE '%web%' OR `Title` LIKE '%services%';`"
- Author search: Author name is separate into three parts in the database: `` `FName` ``, `` `MName` `` and `` `LName` ``. Each word should be used to search in the all three parts of the author name. Assumed that we want to search 'Byeong Ho Kang':
  "`SELECT … FROM … WHERE … `FName` LIKE '%Byeong%' OR `MName` LIKE '%Byeong %' OR `LName` LIKE '%Byeong %' OR `FName` LIKE '%Ho%' OR ` MName ` LIKE '%Ho%' OR ` LName ` LIKE '%Ho%' OR `FName` LIKE '%Kang%' OR ` MName ` LIKE '%Kang%' OR ` LName ` LIKE '%Kang%';`"

Leave the following functions as you have implemented in the assignment 1.

- **Redirecting to the 'Book details' page**
- **Redirecting to the 'Search result' page**
- **Popularity counting**

4.3. **'Book details'** page

The following image indicates how the content section of **'Book details'** page should be.

1. **Redirecting to the 'Search result' page -** Leave this functions as you have implemented in the assignment 1.
2. **Redirecting to the URL of book.** - Leave this functions as you have implemented in the assignment 1
3. **Adding the book to the 'My Library' – only for general members and administrators.**
   The page should not display the link "ADD THIS PAPER to My Library" for the public users. If the user (general members or administrators) clicks the link "ADD THIS PAPER to My Library", the page should display whether the book is successfully added in the "My Library" with a link "Go back to Book Details" that redirects back to the book details page. Then, the added book should be stored in table 'KXT209_User_Library' and the user can check the added book in **4.6. 'My Library'** page. * "PAPER" means "book" *

If the book is already added in the "My Library", the page should display the message stating, "You have already added this book" with a link "Go back to Book Details" that redirects back to the book details page.

4.4. **'Sign up'** page: Leave this page as you have implemented in the assignment 1.

4.5. **'My Page':**

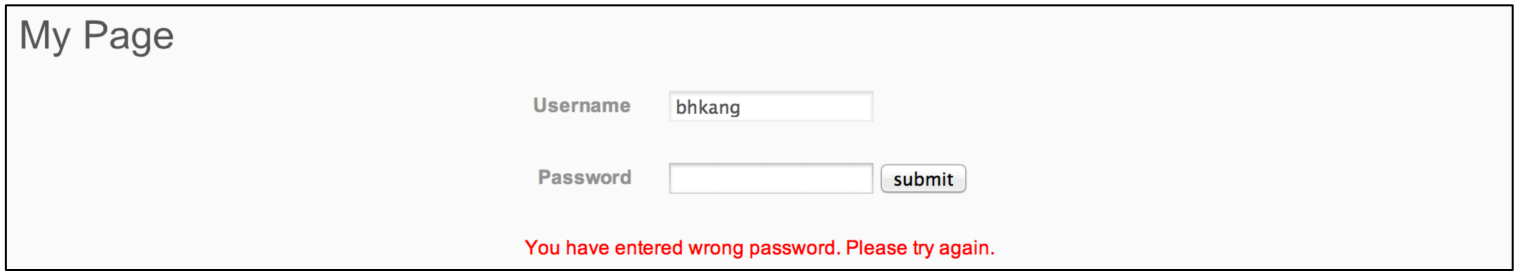
If the public user tries to access the page, the page should display the message "You do not have the access to this page." and a link redirects to Home page.

4.5.1. Authentication form

The page should have a form with username field and password field. The value of the username field should be the current logged-in username, and the username text-field should be un-clickable.

NOTE: This authentication form should be displayed every time the user enters this page.

If the user submits invalid password, the page should display the message "You have entered wrong password. Please try again."



If the user submits the valid password, the page should display the 4.5.2. User details form.

4.5.2. User details form

Every field should be pre-filled (retrieves data from the database by using the current username).

My Page

* Username	bhkang
* First Name	Byeong
Middle Name	Ho
* Last Name	Kang
* Date of Birth	1920-01-03
* Email	Byeong.Kang@utas.edu.au
Phone	
Website	
* Affliation	University of Tasmania
Last Updated	2013-02-26 23:28:53

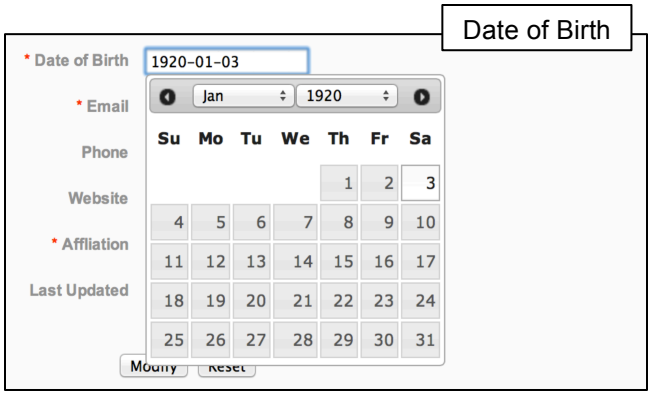
Modify Reset

The validation of the form should be done by using AJAX (JavaScript or jQuery – use plugin, but make sure you should specify it in the readme file). The validation criteria should be as below: (same as the one in the signup page that we have implemented in the assignment 1)

• Username – This field should be pre-filled and un-clickable (disabled).

• First Name, Last Name – This is a required field.

• Middle Name – This is an optional field.



• Date of Birth – This is a required field. Users should be given one input box to enter this field – If the user clicks on the input box; the date picker should show up with the proper date (Please see Demo). 'Year' and 'Month' can be selected from the dropdown list – use the plugin (does not have to be the same one in the demo) and specify it in the readme file.

• Email – This is a required field. To be valid, it must enter @ sign. (If you are using plugin, use the default email function of the plugin)

• Phone, Website – These are optional fields.

• Affiliation – This is a required field.

• Last Updated – This field should be disabled to edit. The data should be retrieved from the 'Updated_date' in the 'KXT209_Users' table.

If the entered data is not valid, the error message should appear next to the related field and the submission should not be made (Please see Demo).

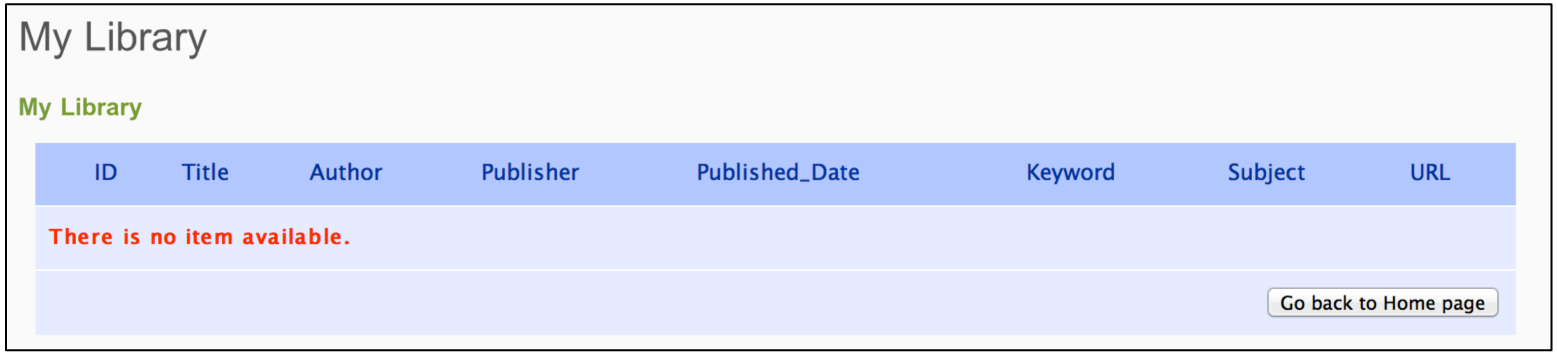
If every submitted data is valid, the submitted data should be used to update the data in the database.

If the user clicks the 'Reset' button, the value should be reset as the default value (when the form is first loaded).

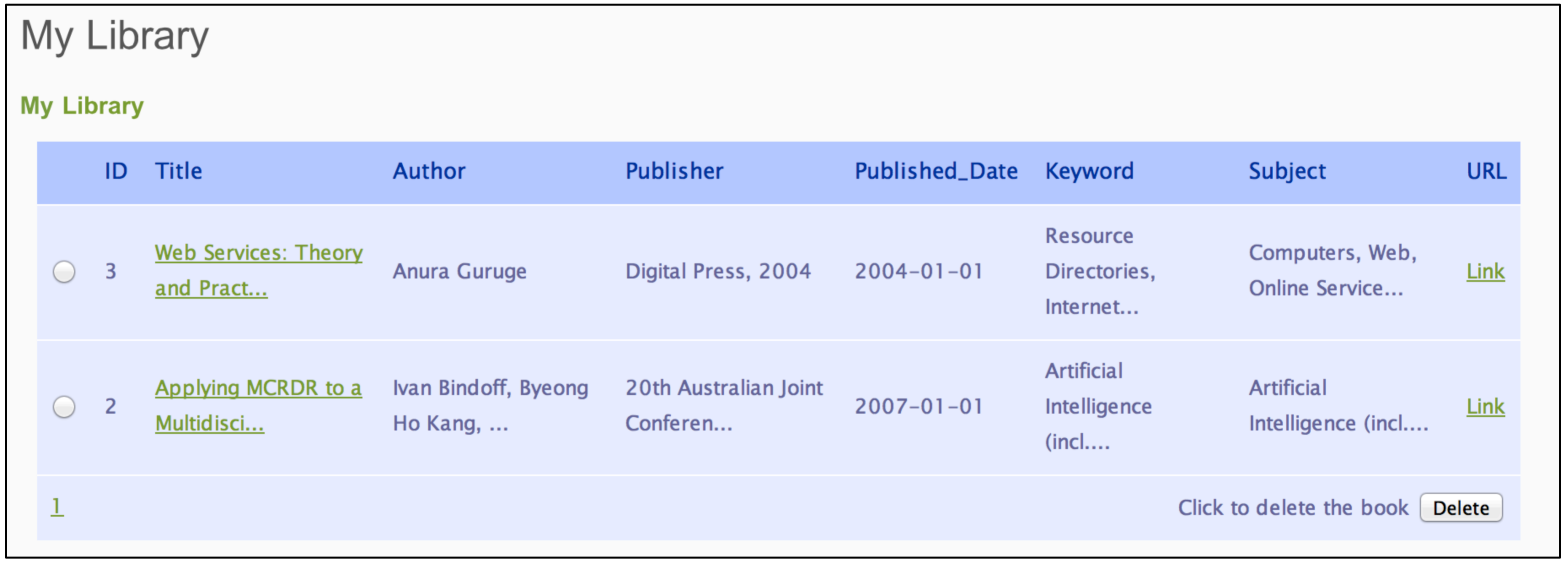
4.6. **'My Library':**

If the public user tries to access the page, the page should display the message "You do not have the access to this page." and a link redirects to Home page.

The page should display the list of books that added by the user (the list should be retrieved from the table 'KXT209_User_Libraries') If there is no item added by the current user in the table 'KXT209_User_Libraries', the page should display the message the following message, "There is no item available" and a button "Go back to Home page" that redirects to the Home page (see the following image).

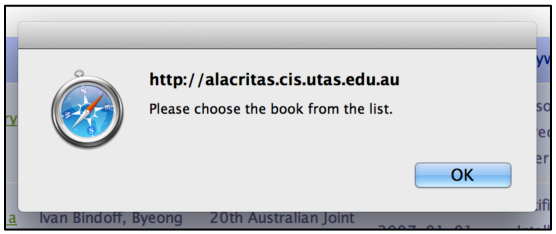


If there is an item/items added by the current user in the table 'KXT209_User_Libraries', the page should display as below:

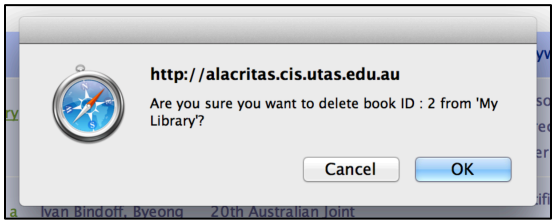


The following functions should be implemented in the My Library Page:

1. The 'Title' of the book (retrieved from the table 'KXT209_Books') should be displayed with <a> tag. If the user clicks the 'Title' (<a> tag) of the book, the page should be redirected to the 4.3. **'Book details'** page that is appropriate to the book.
2. If the data is longer than 30 characters (use `strlen()` function to check the length of the string), the data should be displayed the first 30 characters with "…" at the end of the string.
3. The 'URL' of the book (retrieved from the table 'KXT209_Books') should be displayed as "Link" with <a> tag. If the user clicks the 'Link' (<a> tag), the page should be redirected to the appropriate URL (retrieved from the table 'KXT209_Books').
4. The radio button should be selected, if any part of the row is clicked (this function is related to the tutorial 7-1 work).
5. When the user clicks the 'Delete' button, the system should check whether the radio button is selected or not. If the radio button is not selected, the page should alert the message "Please choose the book from the list."



If the user selects the radio button and clicks the 'Delete' button, the page should ask whether the user wants so delete the book or not (Displaying the confirming message). The confirming message should states "Are you sure you want to delete book ID : *value (ID of the selected book)* from 'My Library'?"



If the user clicks the 'Cancel' button, the pop up window should be closed, but the page should not be refreshed.
If the user clicks the 'OK' button, the system should delete the selected book from the table 'KXT209_User_Libraries' and display the updated page.

6. The page should support the pagination. If the user added more than 5 books, the page should displays the five books per page.
7. If the user hovers each row of the table, the hovered row should be highlighted (i.e. changing color of the row). - Please see the Demo

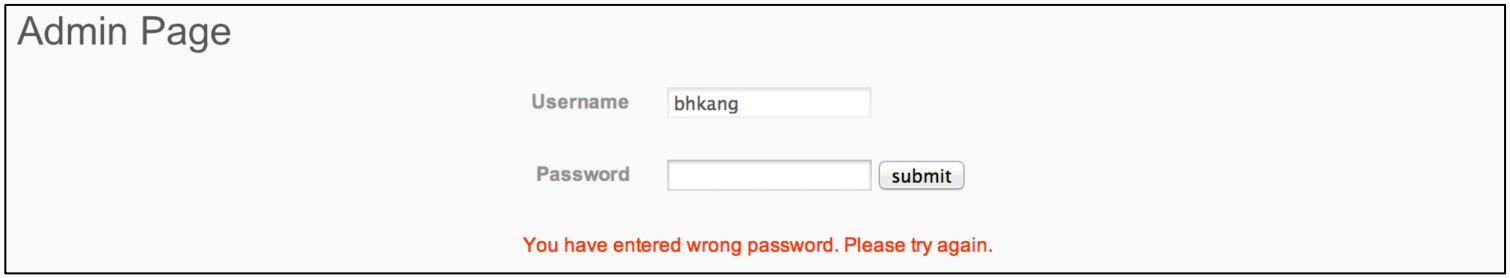
4.7. **'Admin Page':**

Only administrator users should have access to the page. If the public user or general member user tries to access the page, the page should display the message "You do not have the access to this page."

4.7.1. Authentication form

The page should have a form with username field and password field. The username field should be pre-filled (the current username) and the field should be disabled to edit. NOTE: This authentication form should be displayed every time the user enters this page.

If the user submits invalid password, the page should display the message "You have entered wrong password. Please try again."



If the user submits the valid password, the page should display the 4.7.2. Management List.

4.7.2. Management List (Book Management / User Management)

Admin Page

Book Management

ID	Title	Author	Publisher	Published_Date	Keyword	Subject	URL
1	Knowledge Acquisition witho...	Paul Compton, Byeong Kang, ...	European Knowledge Acquisit...	1993-01-01	Artificial Intelligence	Artificial Intelligence	Link
2	Applying MCRDR to a Multidi...	Ivan Bindoff, Byeong Ho Kan...	20th Australian Joint Confe...	2007-01-01	Artificial Intelligence (in...	Artificial Intelligence (in...	Link
3	Web Services: Theory and Pr...	Anura Guruge	Digital Press, 2004	2004-01-01	Resource Directories, Inter...	Computers, Web, Online Serv...	Link
4	.NET Web Services Solutions	Kris Jamsa	John Wiley & Sons, 2006	2006-01-01	Windows Server & NT	Operating Systems	Link
5	Web Search: Public Searchin...	Amanda Spink, Bernard James...	Springer, 2004	2004-01-01	Storage & Retrieval, Comput...	System Administration, Comp...	Link

1 2 Click to add a book Add

User Management

	ID	User_Name	Name	Birth_Date	Email	Affiliation	Level
○	1	test	Test Test Test	1980-11-17	test.test@utas.edu.au	University of Tasmania	Admin
○	2	bhkang	Byeong Ho Kang	1920-01-03	Byeong.Kang@utas.edu.au	University of Tasmania	Admin
○	3	admin	admin admin admin	2013-01-01	admin@admin.com	University of Tasmania	Admin
○	4	general	general general general	1950-01-16	general@general.com	University of Tasmania	General

1 Click to edit the user Edit

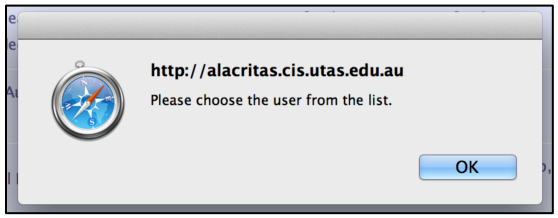
The following functions should be implemented in the Management List:

'Book Management' list

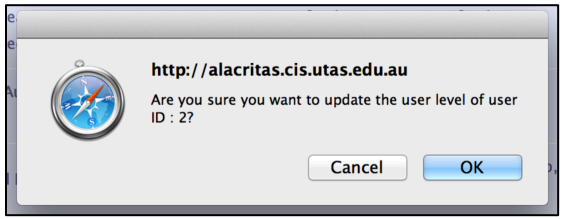
1. The 'Title' of the book (retrieved from the table 'KXT209_Books') should be displayed with <a> tag. If the user clicks the 'Title' (<a> tag) of the book, the page should be redirected to the 4.3. **'Book details'** page that is appropriate to the book.
2. If the data is longer than 30 characters (use strlen() function to check the length of the string), the data should be displayed the first 30 characters with "…" at the end of the string.
3. The 'URL' of the book (retrieved from the table 'KXT209_Books') should be displayed as "Link" with <a> tag. If the user clicks the 'Link' (<a> tag), the page should be redirected to the appropriate URL (retrieved from the table 'KXT209_Books').
4. If the user clicks the 'Add' button, the page should redirect to 4.7.3. Add book form.

'User Management' list

1. The list should display the following columns in the table 'KXT209_Users': ID, Username, FName, MName, LName, Birthdate, Email, Affli and Ulevel.
2. The name of the user should be displayed as a full name (combine FName, MName and LName).
3. User level of the user should be displayed as drop down list that has two options: 'General' and 'Admin'.
4. If the user clicks the 'Edit' button, the system should check whether the radio button is selected or not. If the radio button is not selected, the page should alert the message "Please choose the user from the list."



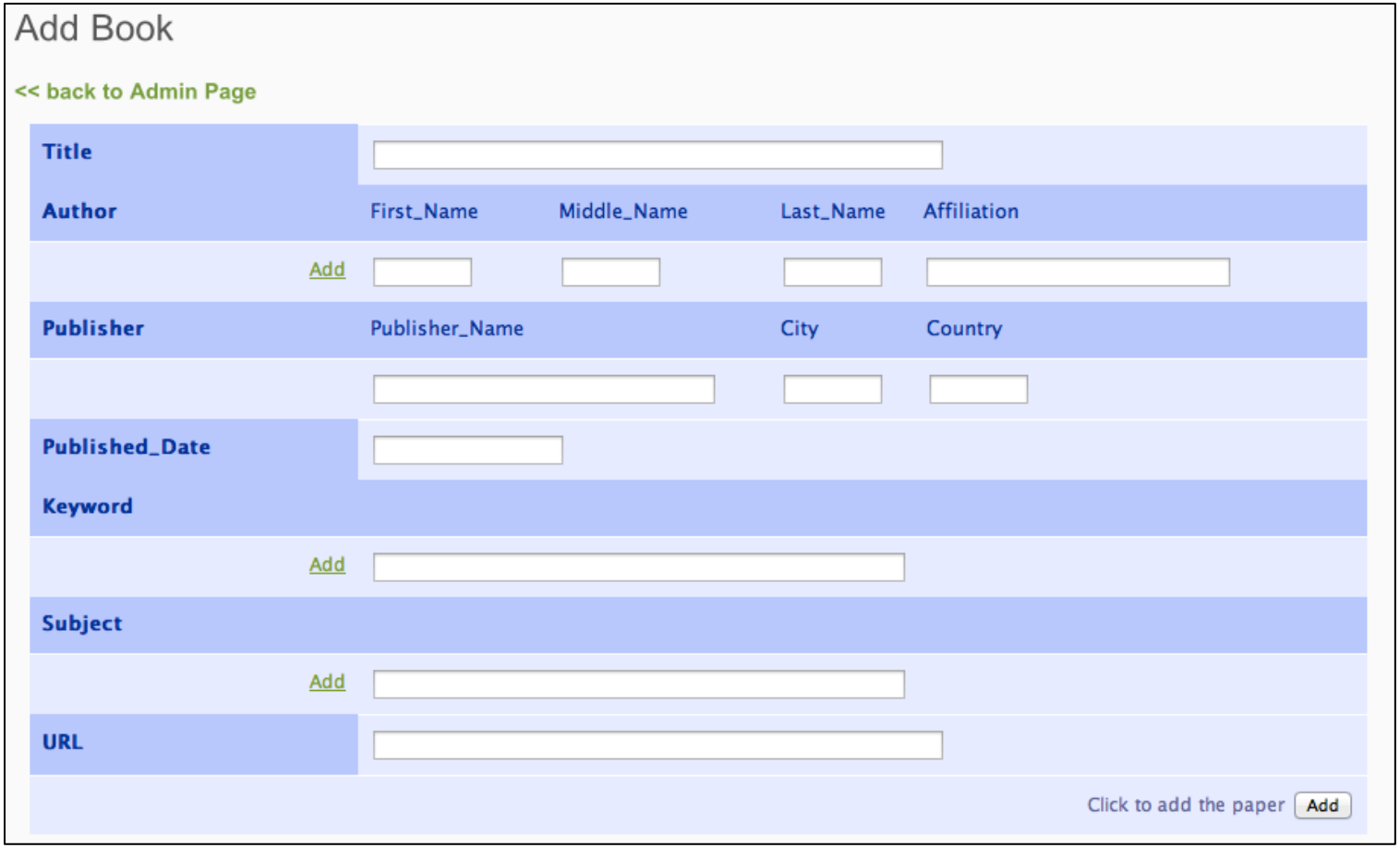
If the user selects the radio button and clicks the 'Edit' button, the page should confirm whether the user wants so change the user-level or not. The confirming message should states "Are you sure you want to update the user level of user ID : *value (ID of the selected user)*?"



If the user clicks the ‘Cancel’ button, the pop up window should be closed, but the page should not be refreshed.
If the user clicks the ‘OK’ button, the system should update the selected user with selected option of the user level (either ‘General’ or ‘Admin’) and display the updated ‘User Management List’

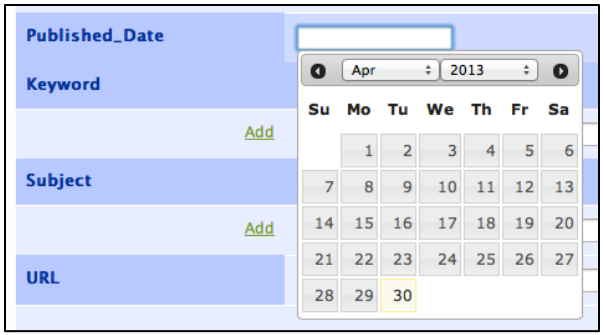
5. If the user hovers each row of the table, the hovered row should be highlighted (i.e. changing color of the row).

4.7.3. Add book form



The following functions should be implemented in this page:

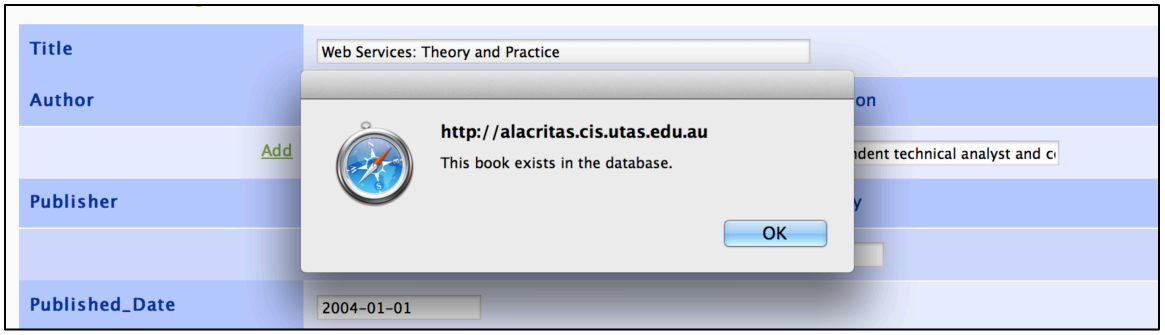
1. If the user clicks the link "back to Admin Page", the page should be redirected to the 4.7.2. Management List.
2. All fields (except a 'Middle Name' field) are required fields.
3. If the user hovers each row of the table, the hovered row should be highlighted (i.e. changing color of the row) - Please see the Demo.
4. If the user submits the form with the required field empty, the page should display the message "Required" (Please see the Demo)
5. If the user clicks the 'Add' link for Author, Keyword, or Subject field, the page should add one more row (can be added as many as user clicks the 'Add' link) However, the added field (by clicking 'Add' link) will not be a required field.
6. If the user clicks the input box of the Published_Date, the date picker should show up with the current date (Please see Demo). 'Year' and 'Month' can be selected from the dropdown list – You can use the plugin, but do not forget to put a reference in the readme file if you use the plugin or open source.



7. When the user clicks the "Add" button (the button in the bottom of the 'Add Book Form'), the system should check whether every required field is filled in.

   If every required field is filled in,

   1. First, the system should check whether the entered book is already in the database. If the data has same 'Title' with same 'Published_Date', the system should assume that the entered book data is already existed. If the existence is detected, the page should alert message "This book exists in the database" and redirect back to the 'Add book form' page.
      NOTE: If there is a book (in the database), which has the same 'Title' but the different 'Published_Date' with the entered book. We consider those are different books.



If the entered book data is not existed in the database, insert all of entered data into the database.

NOTE:
The system should check whether the entered Author, Keyword, Subject, or Publisher has previously added in each four related table: 'KXT209_Author', 'KXT209_Keyword', 'KXT209_Subject' or 'KXT209_Publisher'. If the entered data is already in the related table, the data should not be added in that table again. If the entered data is not in the related table, the system will add the data in all related tables. NOTE: When you put the data for Author, Keyword, or Subject, the system should **also** add the data indicating the relationship between the book and data (i.e. data from table 'KXT209_Books_author', 'KXT209_Books_keywords' and 'KXT209_Books_subject')

Here are the examples how the system is supposed to work:
For example, the user wants to add the book with the following details: 'Author' is 'Byeong Ho Kang', 'Keyword' is 'Mobile Web', 'Subject' is 'Ubiquitous Computing' and the Publisher is 'SERSC'. In this case, the system should work as follows:

Author:
As 'Byeong Ho Kang' already exists in the table 'KXT209_Author', the system should not add data of the author 'Byeong Ho Kang' in the table 'KXT209_Author'. However, the system should add the data indicating the relationship between the entered book (*Book_id=new value*) and author (*Author_id=20*) should be added in the table 'KXT209_Books_author'.

Keyword:
However, the entered Keyword 'Mobile Web' is not in the table 'KXT209_Keyword' yet so the system should add the data for the Keyword 'Mobile Web' in the table 'KXT209_Keyword'. The system should also add the data indicating the relationship between the entered book (*Book_id=new value*) and the new 'keyword id' should be added in the table 'KXT209_Books_keywords'

Subject:
(**'Subject' data should be checked the availability as above. If the entered data is existed, do not add in the table. If not, please add a new data in the table) The system should add 'Ubiquitous Computing' in the table 'KXT209_Subject' and also add the data indicating the relationship between the entered book (*Book_id=new value*) and the new 'subject id' should be added in the table 'KXT209_Books_subject'.

Publisher:
(**'Publisher' data should be checked check the availability as above. If the entered data is existed, do not add in the table. If not, please add the data in the table. However, the publisher_id (indicator of publisher id in the table 'KXT209_Publisher') is included in the KXT209_Books' table.) The system should add 'SERSC' in the table 'KXT209_Publisher' and the id of the entered publisher data should be added in the table 'KXT209_Books'.

5. **Extra:** Leave this section as you have implemented in the assignment 1.

6. **Footer:** Leave this section as you have implemented in the assignment 1.

7. **Others**
   7.1. **Read me** text file

The file readme.txt should contain the following:

1) Student details: full name, username, id number, and campus
2) A list of resources you used in preparing this assignment (i.e. plugins)
3) A list of known problems that you have identified when testing your implementation and have not been able to solve.

MARKING INFORMATION

This assignment will be marked out of 20. The performance standards required for full marks for each section are shown below. Where your performance standard is below the levels shown here your mark for that section will be reduced.

Your submitted work will be tested on Firefox for your marking. Please test your work before you submit your work.

Criterion	Performance standards	Maximum mark for this criterion
Follows acceptable programming practice	• All references (URI values) within the site are **relative** • Display is not seriously broken • No other serious violations of the programming practices required for this unit • Application behaves correctly when the content of the data-store is changed • The database and tables are used as required (*database on school server is **case-sensitive**) • Assignment coversheet is included	N/A **This criterion is important.** If you do not match the required performance standard for this criterion your mark for this assignment will be reduced by between 1 and 2 grade levels. (i.e. DN → CR)
Correct submission of work as specified	• Files are correctly named • Read me file contains all required information	1.0 mark (5%)
Handling access level	• Public user or General member user cannot access to the non-permit pages.	1.0 mark (5%)
Search result page functions correctly	• The result is retrieved by separating the words (if the search term is more than one word) • Author search query works correctly (facilitating three parts correctly: first, middle, last name)	1.5 mark (7.5%)
Book details section functions correctly	• Display the link 'ADD THIS PAPER to My Library' ONLY for the general member and admin user • Display the messages correctly after the link is clicked	1.5 mark (7.5%)
My Library functions	• Display the table as specified	4.0 mark (20%)

correctly	• Pagination works correctly • Links work correctly (Title, and URL) • Hovering and Auto-select radio button work correctly • Delete button works correctly (Alert, Confirm message) • Data deletion works correctly	
My Page functions correctly	• Authentication form works correctly • The user details form displays as specified (including the data retrieval) • Date picker works correctly • Validation works correctly	3.0 mark (15%)
Admin Page functions correctly	• Authentication form works correctly • Display the table as specified • Pagination works correctly • Links work correctly (Add button, Title, and URL) • Hovering and Auto-select radio button work correctly • Edit button works correctly (Alert, Confirm message) • Displaying the current user access level and Editing user access level works correctly	4.0 mark (20%)
Admin Page: Add book form functions correctly	• Display the form as specified • Validation works correctly • Adding a row works correctly • Date picker works correctly • Adding book data works correctly	3.0 mark (15%)
Code is well written	• Code can be easily read and the logic follows • Uses built-in PHP functions where possible • Follows correct naming conventions for variables and functions • Object-oriented programming is used.	1.0 mark (5%)

HELP AND HINTS

- Read the specification carefully and make sure that you know what the PHP application needs to do (Demo video is also available).
- Look at the examples in self-studies, tutorials, and lectures, to see whether you have seen similar tasks before.
- Try out techniques that you think will do the job.
- DO NOT neglect your self-study and tutorial work in the unit to work on the assignment. Some of the activities may lead you to see what needs to be done in the assignment.
- You may seek help with this assignment in normal consultation times for this unit or via email to your lecturer or tutor. Here are some hints to make best use of help.
    - Make sure you know (or think you know) what your problem is.
    - Have details of the work you have done so far and the progress you have made on hand when you seek help.
    - The more specific you can be in your request for help the more immediately useful the help is likely to be.
- It is highly recommended to develop the websites in school machine. Student who uses his/her own server for assignment, check the programs with school machine before you submit the assignment. (Student must take responsibility for using his/her own server).

PLAGIARISM AND CHEATING

Practical assignments are used by the School of Computing and Information Systems for students to both reinforce and demonstrate their understanding of material which has been presented in class. They have a role both for assessment and for learning. It is a requirement that work you hand in for assessment is substantially your own.

Cheating

- Cheating occurs if you claim work as your own when it is substantially the work of someone else.
- Cheating is an offence under the Ordinance of Student Discipline within the University. Furthermore, the computing profession has ethical standards in which cheating have no place.
- Cheating involves two or more parties.
- If you allow written work, computer listings, or electronic version of your workbook to be borrowed or copied by another student you are an equal partner in the act of cheating.
- You should be careful to ensure that your work is not left in a situation where it may be stolen by others.
- Where there is a reasonable cause to believe that a case of cheating has occurred, this will be brought to the attention of the unit lecturer. If the lecturer considers that there is evidence of cheating, then no marks will be given to any of the students involved. The case will be referred to the Head of School for consideration of further action.

SUBMISSION DETAILS

- Make sure that the folder in which you have all the files for this assignment is enclosed in a folder that is labelled with your username.
- Download the word version of the School of Computing & Information Systems "assignment cover sheet" (follow the links from the home page), edit it to contain your details, and copy the edited version into the folder containing all the files for this assignment. In doing this you are deemed to have submitted a signed version of the cover sheet.
- Make a .zip archive of the folder. On a Mac machine, this is very easy – in the finder select the folder to "zip", right click and select the relevant option from the drop down list you will see (this will be labelled "compress …" or possibly "create archive"), the system will then create a .zip file. If your username is 'bhkang' and all of the files for the assignment submission were in a folder and labelled "`assign2_bhkang.zip`" then the archiving (zipping) process would produce a file called "`assign2_bhkang.zip`".
- Go to your home directory on alacritas (Hobart students) or lawson (Launceston students).
- You will find there a directory (folder) called kxt209submit (the "submit" folder) (If you do not have a submit folder, please contact the Computing Help desk.)
- Copy your .zip file into this directory. Open the submit folder to see that your file has been copied. If you want to resubmit, open the submit folder, remove the old version of your submitted work and copy in the new version. Please keep the submitted files.

CHECKLIST

When you can answer yes to all of these questions, you are ready to submit the assignment

Q: Are all the files contained in a folder (directory) that is labelled with your user name?

Q: Are all the files that are required present?

Q: Have you checked the function of all aspects of the application and documented any problems in the readme.txt file?

Q: If you have changed any file names or locations, have you rechecked the function?

Q: Have you have checked that all of the URIs are relative references? Your application will be moved to a different location for testing

– any links back to your alacritas / lawson account will be noticed and penalised.

Q: Have you filled in a copy of the School of Computing & Information Systems assignment cover sheet and added it to the folder that you will submit?

Q: Have you made a .zip file of the final version of your folder?

Q: Do you have a submission folder for this unit (kxt209) in your alacritas / lawson account?

Q: Have you checked whether your program is working properly in the school machine (Firefox in Mac)?