COMP 636: Web Technologies (Jinja + CSS) Assessment

Due: **5pm Wednesday 18th January 2023**

Worth **15%** of COMP636 grade

Submit via Akoraka | Learn

Instructions

This code is library management system for Waikirikiri Library. It is used to manage books, borrowers and loans. You are provided with a flask app with some template files. For this assessment modify only the provided template files, do not alter the python code provided.

30 marks available in total.

Add the following features to the system:

1. **Book list (2 marks):** Add bootstrap table formatting to the booklist.html template so that each row is striped and hoverable.
2. **Borrower list (6 marks):** Add Jinja code to borrowerlist.html so that it lists all borrowers in a table.
3. **Add Loan (6 marks):** Complete the addloan.html template so that a drop down box of Borrowers is shown along with a drop down box of available book copies. The form should be submitted using POST to the /loan/add route. The form should have a hidden input field set to the current date. Required control ids are shown in comments in the template.
4. **Loan History Report (10 marks):** Complete the currentloans.html template to display a list of current and past loans sorted by borrower family name then first name. Use a table. Each row should be coloured depending on the type of book borrowed (Paperback and Hardback, white; eBook, Green; Audio Book, Yellow). Indicate ‘Returned’ or ‘On Loan’ for each book. The borrower ID should not be shown and the borrower name should be shown only once, even if they have borrowed more than one book.
5. **Overall quality (6 marks):** Tidy output presentation, use of bootstrap, code structure and commenting across all questions.

File Download and Submission Instructions:

Download the following files from Akoraka | Learn on the COMP636 Assessment page:

Librarywebapp.zip

The file contains an app.py and connect.py python files, and a templates directory containing the template files for the app.

To run the app, install flask (pip install flask), unzip the assessment files into a folder, open the folder then open connect.py and update with your MySQL connection password.

Open app.py and in a terminal type: flask run

You can then click the link to http://127.0.0.1:5000/ to open the home page of the app.

You need to stop the app running (CTRL+C in terminal window) before modifying templates.

Remember to run the app again after making modifications to test them.

Submit (upload) only your template (.html) files as a zip file:

- zip your template directory
- Include your name and student ID in a comment at the start of every template file
- Submit your zip file via the submission link on the COMP636 Assessment page

Approximate Mark Allocation:

30 marks available:

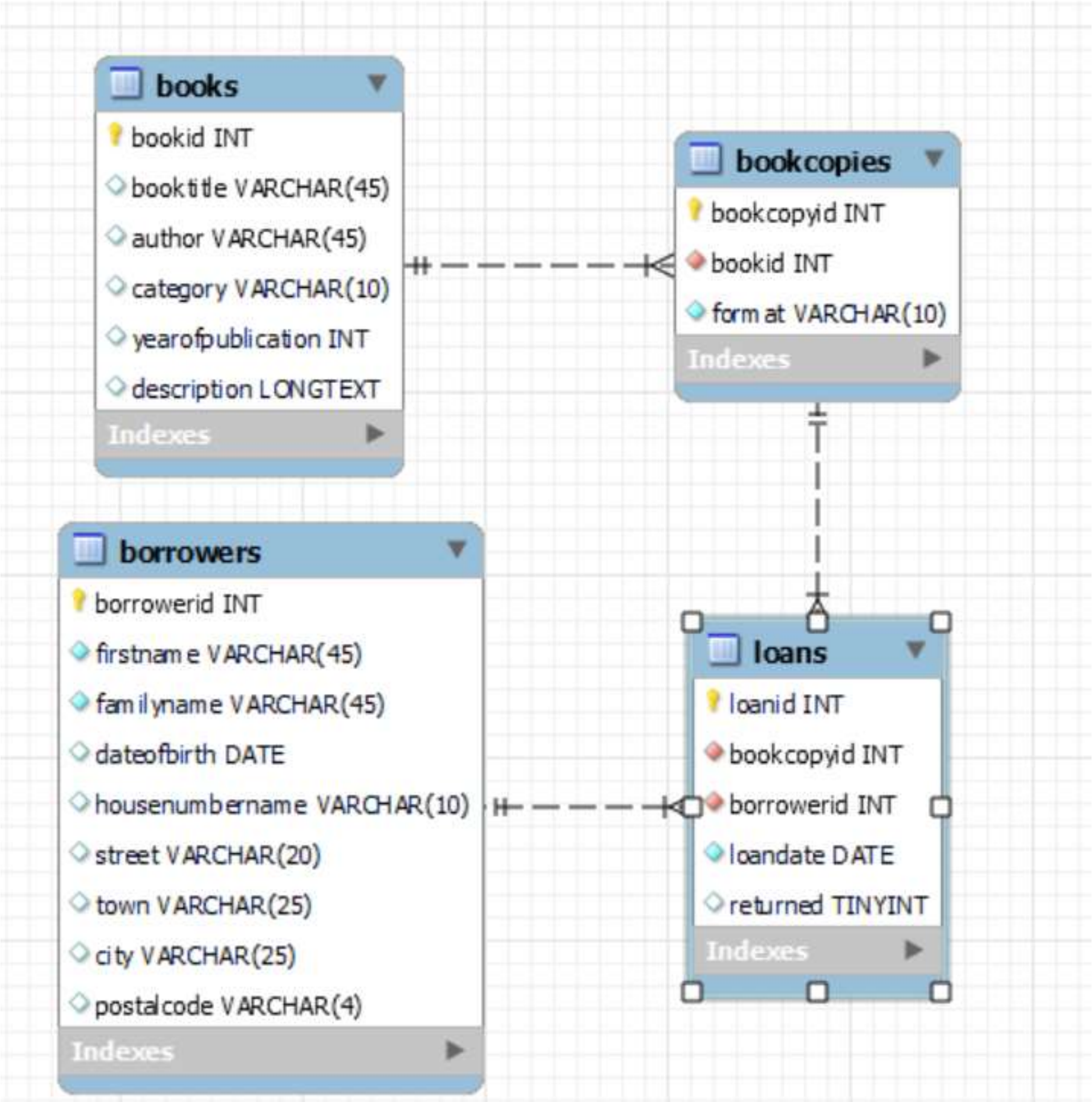
Item	**Marks available**
Book list	2
Borrower list	6
Add Loan	6
Current Loaned Books Report	10
Overall quality	6
TOTAL	**30**

Additional notes:

- **The quality of the user experience will be taken into account for each assessment item** as well as in the Overall Quality mark in the table above. Full marks for any item will require validation of data types (if required) and details in the interface that demonstrate some consideration of what would work well for the user (within the limitations of the terminal window output in VS Code).
- A diagram of the database tables, field names and their relationships, is provided at the end of this document.
- You will need to update `connect.py` with your own connection details to your database. These details are visible on the first screen of MySQL Workbench in the grey box.
- Add comments to your code. The existing comments give some hints about the behaviour expectations of each template page.

Database structure

Note: The lines shown only indicate the relationships between the tables. They do not indicate which fields are included in the table relationships. See the table below the diagram for the field details.



Foreign keys

Foreign keys enforce ‘referential integrity’ – which means that a value in the child table field must match an entry in the parent table field, e.g., a **bookid** cannot be entered into the **bookid** field of the **bookcopies** table if it is not in the **bookid** field of the book table.

Parent table and field(s)		Child table and field(s)
books.bookid	—<	bookcopies.bookid
bookcopies.bookcopyid	—<	loans.bookcopyid
borrowers.borrowerid	—<	loans.borrowerid

Loans that have been returned have the value 1 in the returned field.