

Assignment 2, Part B: Saving Headlines

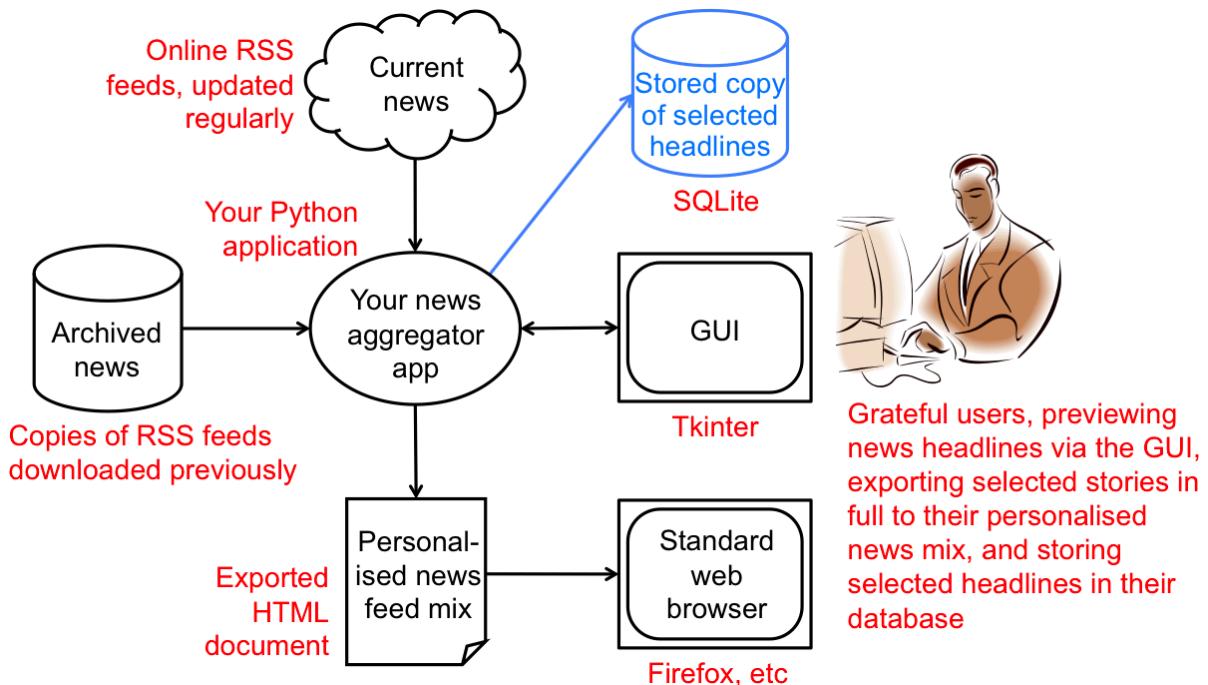
(4%, due 11:59pm Sunday, October 20th, end of Week 12)

Overview

This is the second part of a two-part assignment. This part is worth 4% of your final grade for IFB104. Part A which preceded it was worth 21%. This part is intended as a last-minute extension to the assignment, thereby testing the maintainability of your solution to Part A and your ability to work under time pressure. If you have a neat, clear solution to Part A you will find completing Part B easy. For the whole assignment you will submit only one solution, containing your combined response to both Parts A and B, and you will receive one grade for the whole 25% assignment.

Goal

In pre-computer times avid news readers would clip stories of interest from newspapers and paste them into scrapbooks as a record of recent events. Today we can do the same thing, but more easily and flexibly, by taking advantage of database technology. In Part A of this assignment you built a significant application which allows its users to select and preview headlines from multiple news feeds and export the corresponding stories in full as a web document. Here you will extend your application so that it also allows the previewed headlines to be stored in a database, either for later viewing via a database browser or processing by some other software application. This extension is shown in blue below.



To complete this part of the assignment you must extend the Python code you have written for Part A so that it gives the user the ability to store their currently-selected headlines from all news feeds in the database whenever they want. This involves both extending the GUI and linking your application to an SQLite database, `news_log.db`.

The database file accompanies these instructions. It contains a single table, `selected_stories`, which has three fields, `headline`, `news_feed` and `publication_date`. For each news story currently chosen by the user your program must insert these three values as a new row in the table. Your program should assume the database already exists and the table is empty when it is started. Whenever the user chooses to store the currently-selected headlines, any previous data in the table is overwritten; only the currently-selected news stories are stored, not previous selections.

Illustrative example

Supplied with these instructions is an empty SQLite database called `news_log.db`. If you open it in the *DB Browser for SQLite* or an equivalent tool you will see that it contains a single table, `selected_stories`, which has three text fields, as shown below.

The screenshot shows the 'Database Structure' tab of the DB Browser for SQLite. A table named 'selected_stories' is selected. The schema is displayed as:

```
CREATE TABLE "selected_stories" (
    `headline` TEXT,
    `news_feed` TEXT,
    `publication_date` TEXT
)
```

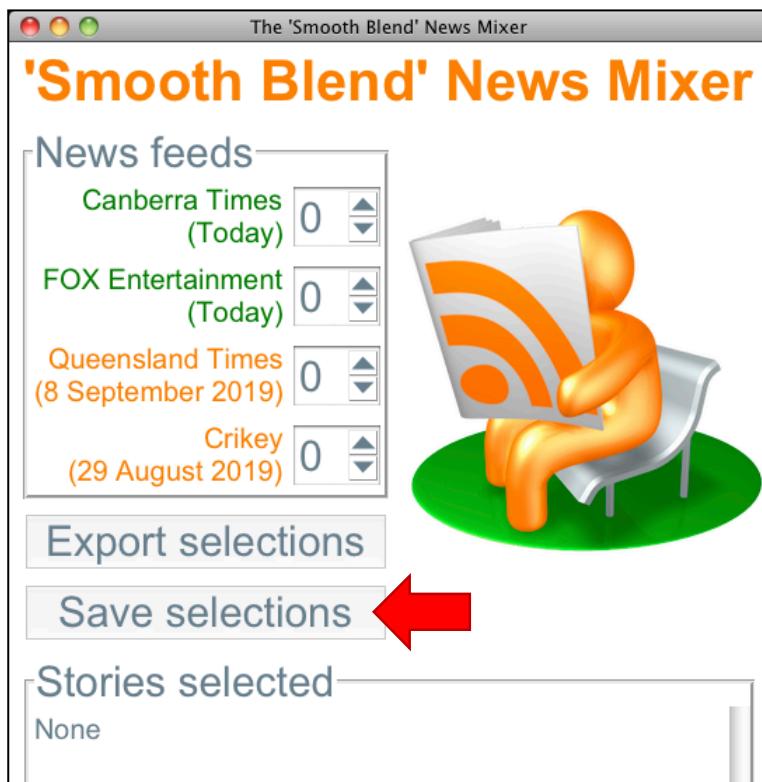
The table has 3 columns: headline, news_feed, and publication_date, all of type TEXT.

As supplied, the table in this database is initially empty, as shown below.

The screenshot shows the 'Browse Data' tab of the DB Browser for SQLite. The 'selected_stories' table is selected. The data grid is empty, showing only column headers: headline, news_feed, and publication_date. Navigation buttons (|<, <, >, >|) and a 'Go to:' input field (set to 1) are visible at the bottom.

In the instructions for Part A we used a demonstration program which we called “*The ‘Smooth Blend’ News Mixer*”. It allowed its user to preview news stories from multiple sources in a graphical user interface and export the chosen mixture of stories as an HTML document. Now we extend its capabilities so that it additionally allows the user to store the chosen headlines in the database. To illustrate this, consider the following example.

As explained in the Part A instructions, your Python program must allow the user to preview (at least) four news feeds, two “archived” ones that never change, and two “live” ones from the Internet. To allow the user to store specific stories in the database, we have extended our GUI with an additional widget, in this case a push button labelled “Save selections”, as shown by the red arrow below.



When pressed, this button saves the headlines for the user’s current story selections in the database. Assume that at first the user wants to review some of the archived news stories. In the screenshot below the user has selected the top two stories from the archived copy of the *Queensland Times*, and the top story from *Crikey*. Headlines, sources and publication dates for the three stories selected can be seen in the GUI at the bottom. Neither of these news sources have been updated since the Part A instructions were written, which is why it’s important to let the reader know the original publication dates.



Now assume the user is interested in these particular news events and decides to save them in the database. In this particular GUI design this is done by pressing the “Save selections” button. If we view the database’s contents at this point we see that the previewed information has now been stored in the `selected_stories` table, as shown below.

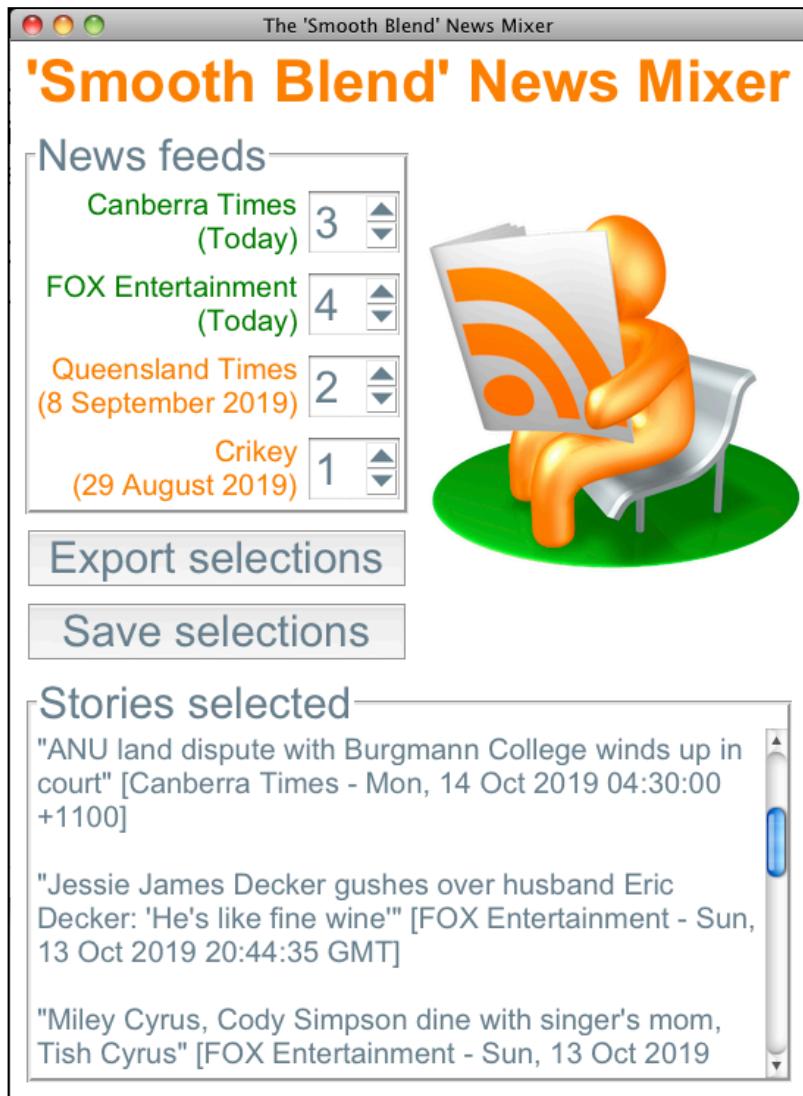
The screenshot shows the DB Browser for SQLite application interface. The top menu bar includes "New Database", "Open Database", "Write Changes", and "Revert Changes". The toolbar below the menu includes "Database Structure", "Browse Data" (which is selected), "Edit Pragmas", and "Execute SQL".

The main area displays the "selected_stories" table with the following data:

	headline	news_feed	publication_date
1	UPDATE: Fires burning across the r...	Queensland Times	Sun, 08 Sep 2019 00:00:00 +1000
2	Respected sporting role model insp...	Queensland Times	Sun, 08 Sep 2019 00:00:00 +1000
3	Boris Johnson to suspend UK parlia...	Crikey	Wed, 28 Aug 2019 21:04:20 +0000

At the bottom, there are navigation buttons for the first page, previous page, next page, last page, and a "Go to:" input field with the value "1".

Of course, seeing current news is much more interesting. Therefore, we now assume that our user selects the top three stories currently reported by the *Canberra Times* and the top four stories currently in *FOX News' Entertainment* section. This produced the following result in the GUI when we ran our demonstration solution on the morning of October 14th, 2019. (Not all the selected stories are can be seen in the screenshot below, but they are accessible by scrolling the text box.)



The user has now created a mix containing ten news stories in total from four different sources (but this could be as high as 40 stories if the maximum number was chosen from each source). Satisfied with this new selection the user again presses the “Save selections” button. This causes the previous contents of table `selected_stories` to be overwritten with the new selections. Viewing the database’s contents at this point shows all the selected headlines, as can be seen below.

	headline	news_feed	publication_date
1	Echoes of the past, seeds in the bank: ins...	Canberra Times	Mon, 14 Oct 2019 04:30:00 +1100
2	Jim's still going strong, 41 fun runs and c...	Canberra Times	Mon, 14 Oct 2019 04:30:00 +1100
3	ANU land dispute with Burgmann College...	Canberra Times	Mon, 14 Oct 2019 04:30:00 +1100
4	Jessie James Decker gushes over husband...	FOX Entertainment	Sun, 13 Oct 2019 20:44:35 GMT
5	Miley Cyrus, Cody Simpson dine with sin...	FOX Entertainment	Sun, 13 Oct 2019 18:49:23 GMT
6	Thomas Rhett, Lauren Akins celebrate 7 y...	FOX Entertainment	Sun, 13 Oct 2019 18:00:52 GMT
7	Julianne Hough mourns beloved dogs wh...	FOX Entertainment	Sun, 13 Oct 2019 16:57:47 GMT
8	UPDATE: Fires burning across the region	Queensland Times	Sun, 08 Sep 2019 00:00:00 +1000
9	Respected sporting role model inspires a ...	Queensland Times	Sun, 08 Sep 2019 00:00:00 +1000
10	Boris Johnson to suspend UK parliament	Crikey	Wed, 28 Aug 2019 21:04:20 +0000

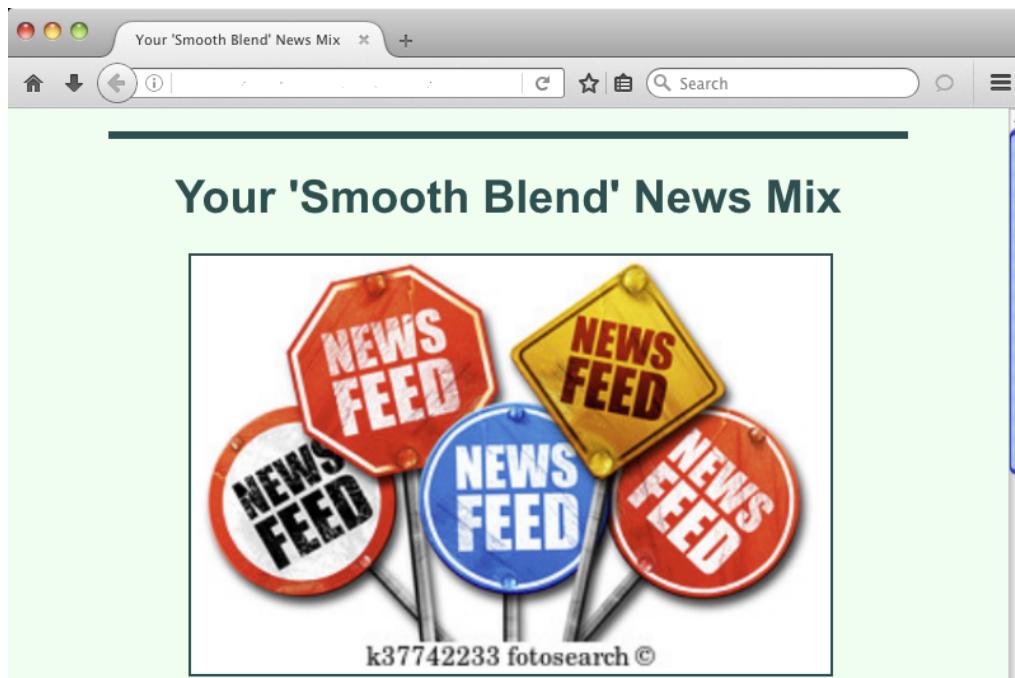
|< < 1 - 10 of 10 > >| Go to: 1

UTF-8

Recall that the order of columns and rows in a relational database table is irrelevant. In the screenshot above the items appear in the order they were inserted by our Python program, but in some *DB Browser for SQLite* implementations the order in which rows are displayed may be different.

This demonstration was done on the morning of October 14th. However, due to time zone differences, the stories from *FOX Entertainment* are timestamped as October 13th. We simply reproduce the publication dates exactly as they appear in the original web pages' source codes.

Finally, to complete the example, we assume our user also exports the selected stories in full as an HTML document, as shown below when viewed in a web browser. Notice that both old "archived" stories and "live" stories are shown. As usual, all the images displayed are links to online files, not files stored on the local computer, so that the exported HTML document is portable across any computing platform.



Echoes of the past, seeds in the bank: inside the Yarralumla Nursery



It's not quite the Svalbard Global Seed Vault - but inside a cool and slightly musty room at the Yarralumla Nursery, the seeds of the future rest dormant.

Canberra Times - Mon, 14 Oct 2019 04:30:00 +1100

Jim's still going strong, 41 fun runs and counting





Jim White has competed in 41 of the 43 Canberra Times Fun Runs. Not even a suspected aneurysm can stop his 42nd.

Canberra Times - Mon, 14 Oct 2019 04:30:00 +1100

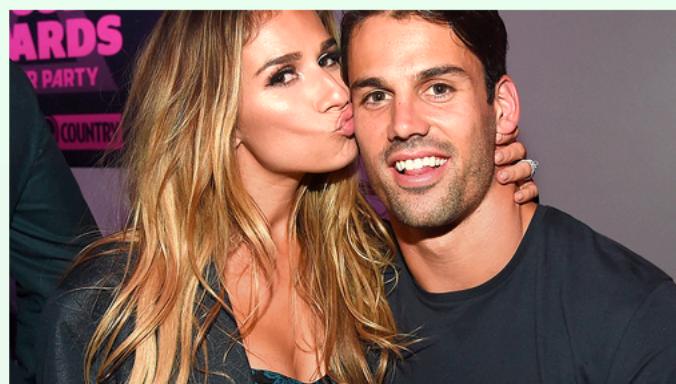
ANU land dispute with Burgmann College winds up in court



Burgmann College is suing the ANU to stop it building a road through its leased land.

Canberra Times - Mon, 14 Oct 2019 04:30:00 +1100

Jessie James Decker gushes over husband Eric Decker: 'He's like fine wine'



Jessie James Decker can't keep her hands to herself.

FOX Entertainment - Sun, 13 Oct 2019 20:44:35 GMT



Jessie James Decker can't keep her hands to herself.

FOX Entertainment - Sun, 13 Oct 2019 20:44:35 GMT

Miley Cyrus, Cody Simpson dine with singer's mom, Tish Cyrus



Miley Cyrus and Cody Simpson were spotted on another date -- this time, with the singer's mom, Tish Cyrus, in tow.

FOX Entertainment - Sun, 13 Oct 2019 18:49:23 GMT

Thomas Rhett, Lauren Akins celebrate 7 years of marriage: I find 'new ways to love you every single day'



Thomas Rhett shared a sweet tribute to his wife, Lauren Akins, on their seventh wedding anniversary.

FOX Entertainment - Sun, 13 Oct 2019 18:00:52 GMT

Julianne Hough mourns beloved dogs who died on the same day: 'Pure love is real'

Julianne Hough is mourning the loss of her two dogs who died on the same day.

FOX Entertainment - Sun, 13 Oct 2019 16:57:47 GMT

UPDATE: Fires burning across the region

Firefighters are working to contain two fires in Ipswich and warning residents to call Triple Zero immediately if their property is under threat.

Queensland Times - Sun, 08 Sep 2019 00:00:00 +1000

Respected sporting role model inspires a future leader

Respected sporting role model inspires a future leader



Both are Ipswich-bred Hancock Brothers A-Grade players who have excelled this season on and off the hockey field.

Queensland Times - Sun, 08 Sep 2019 00:00:00 +1000

Boris Johnson to suspend UK parliament



Good morning, early birds. Boris Johnson will suspend parliament in a bid to halt Brexit resistance, and NSW Labor has suspended general secretary Kaila Murnain amid fallout from the party's donations scandal. It's the news you need to know, with **Chris Woods**.

Crikey - Wed, 28 Aug 2019 21:04:20 +0000

Sources

- Canberra Times: <http://www.canberratimes.com.au/rss.xml>
 - FOX Entertainment: <http://feeds.foxnews.com/foxnews/entertainment>
 - Queensland Times: <https://www.qt.com.au/feeds/rss/homepage>
 - Crikey: <https://www.crikey.com.au/feed>
-

Your task is to extend your solution to Part A of the assignment with a database storage capability equivalent to that above. However, you do not need to follow the GUI design above. Creativity is encouraged and any solution that is easy to use and provides the user with equivalent functionality is acceptable.

Development hints

Notice that the information to be stored in the database is exactly the same as the information previewed in the GUI. Therefore, you should be able to reuse the same solution for extracting elements from web documents for both the previewing and database storage features of your application.

The SQLite code needed to complete this task is quite simple. Similar examples can be found in the relevant lecture demonstrations and workshop exercise solutions.

Requirements and marking guide

To complete this task you are required to further extend the provided `news_aggregator.py` template file with your solution to Part B, on top of your solution to Part A, to provide a data storage capability equivalent to that illustrated above.

The extension for Part B must satisfy the following criteria. Marks available are as shown.

- **Widget(s) for choosing to store selected headlines (2%).** Your GUI must provide some simple, intuitive feature to allow the user to control whether or not the currently selected headlines are stored in the database. A number of solutions are possible, e.g., a pushbutton widget which the user presses in order to save the headlines currently selected, or a “save mode” checkbox or radiobuttons which can be toggled so that the currently-selected headlines are stored automatically whenever a fresh news mix is selected. Either way, the user must have full and clear control over whether or not the headlines are saved in the database.
- **Storing headlines in the database (2%).** Whenever the user chooses to do so, all of the currently-selected headlines must be written into the `news_log.db` database’s `selected_stories` table. Each record must consist of the headline, the identity of the source news feed, and the publication date. Existing data in the table, if any, should be overwritten, so that only the headlines currently selected are stored. Your Python code should assume that the necessary SQLite database already exists in the same folder as your Python program.

You must complete this part of the assignment using only basic Python features and the `sqlite3` module. It should be possible to complete this task merely by adding code to your existing solution, with little or no change to the code you have already completed in Part A.

If you are unable to complete the whole task, just submit whatever parts you can get working. You will receive *partial marks* for incomplete solutions.

Supporting material

Accompanying these instructions we have provided a copy of the necessary SQLite database, `news_log.db`. It contains a single table, `selected_stories`, which has three fields, `headline`, `news_feed` and `publication_date`. For the purposes of this assignment the definition of the database’s `selected_stories` table is very simple, merely three fields of type `Text`, and no primary key for the table is specified.

As supplied the database table is empty. Your solution should assume that this database and its table *already exists in the same folder as your Python program*. Your Python program does not need to create the database or the table. Before you begin you should confirm that you can open this database with the *DB Browser for SQLite* or a similar database tool.

As well as the SQLite database, `news_log.db`, we have provided a very simple “dump” script, `news_log.sql`, which re-creates the database’s table when executed. You should not need to use this script at all; it should be possible to open the `news_log.db` file from the *DB Browser for SQLite*’s “Open Database” menu option. (Do not drag-and-drop the database file into the *DB Browser*’s GUI. On some platforms this causes the *DB Browser* to ask for a password, even though the database is *not* password-protected.) In the highly unlikely event that you need to create a fresh copy of the database using the dump script you can do so easily using the *DB Browser*’s “Import database from SQL file” menu option. Ensure that you name the resulting database “`news_log.db`” and remember to write the changes to it after importing the script.

Portability

An important aspect of software development is to ensure that your solution will work correctly on all computing platforms (or at least as many as possible). For this reason you must complete the assignment using standard Python 3 functions and modules only. You may not import any additional modules or files into your program other than those already imported by the given template file. In particular, **you may not use any Python modules that need to be downloaded and installed separately, such as “Beautiful Soup” or “Pillow”. Only modules that are part of a standard Python 3 installation may be used.**

Security warning and plagiarism notice

This is an individual assessment item. All files submitted will be subjected to software plagiarism analysis using the MoSS system (<http://theory.stanford.edu/~aiken/moss/>). Serious violations of the university’s policies regarding plagiarism will be forwarded to the Science and Engineering Faculty’s Academic Misconduct Committee for formal prosecution.

As per QUT rules, you are not permitted to copy or *share* solutions to individual assessment items. In serious plagiarism cases SEF’s Academic Misconduct Committee prosecutes both the copier and the original author equally. It is your responsibility to keep your solution secure. In particular, **you must not make your solution visible online via cloud-based code development platforms such as GitHub**. Note that free accounts for such platforms are usually public. If you wish to use such a resource, do so only if you are certain you have a private repository that cannot be seen by anyone else. For instance, university students can apply for a free private repository in GitHub to keep their assignments secure (<https://education.github.com/pack>). However, we recommend that the best way to avoid being prosecuted for plagiarism is to keep your work well away from the Internet!

Deliverables

You should develop your solution by completing and submitting the provided Python template file `news_aggregator.py`. Submit this in a “zip” archive containing all the files needed to support your application as follows:

1. Your `news_aggregator.py` solution. Make sure you have completed the statement at the beginning of the Python file to confirm that this is your own individual

work by inserting your name and student number in the places indicated. **Submissions without a completed statement will be assumed not to be your own work.**

2. One or more *small* image files needed to support your GUI interface, but **no other image files**.
3. The previously-downloaded web documents used as your static “archive” of old news stories. Only HTML/XML source code files may be included. **No image or style files associated with the web documents may be included.** All images or styles needed to support your exported HTML document must be sourced from online when it is viewed in a web browser.
4. A copy of the `news_log.db` database as provided with these instructions. Ensure that the database’s `selected_stories` table is **empty when you submit your solution** so that the marker can clearly see your Part B solution adding data to it.

Once you have completed your solution and have zipped up these items submit them to Blackboard as a single file. **Submit your solution compressed as a “zip” archive. Do not use other compression formats such as “rar” or “7z”.**

Apart from working correctly your Python and HTML code must be well-presented and easy to understand, thanks to (sparse) commenting that explains the *purpose* of significant elements and *helpful* choices of variable, parameter and function names. **Professional presentation** of your code will be taken into account when marking this assignment.

If you are unable to solve the whole problem, submit whatever parts you can get working. You will receive **partial marks for incomplete solutions**.

How to submit your solution

A link is available on Blackboard under Assessment for uploading your solution before the deadline (11:59pm Sunday, October 20th, end of Week 12). Note that you can submit as many drafts of your solution as you like. You are strongly encouraged to *submit draft solutions* before the deadline as insurance against computer and network failures. If you are unsure whether or not you have successfully uploaded your file, upload it again!

Students who encounter problems uploading their files to Blackboard should contact the IT Helpdesk (ithelpdesk@qut.edu.au; 3138 4000) for assistance and advice. Teaching staff will not be available to answer email queries on the weekend the assignment is due, so ensure that you have successfully uploaded at least one solution by close-of-business on Friday, October 18th.