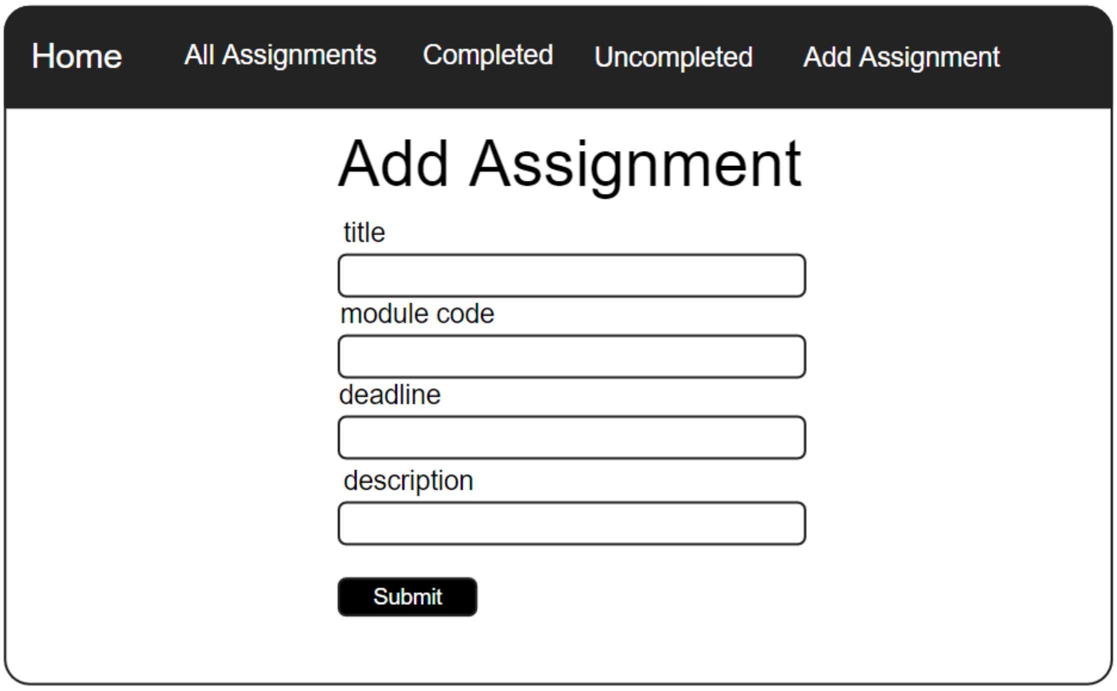
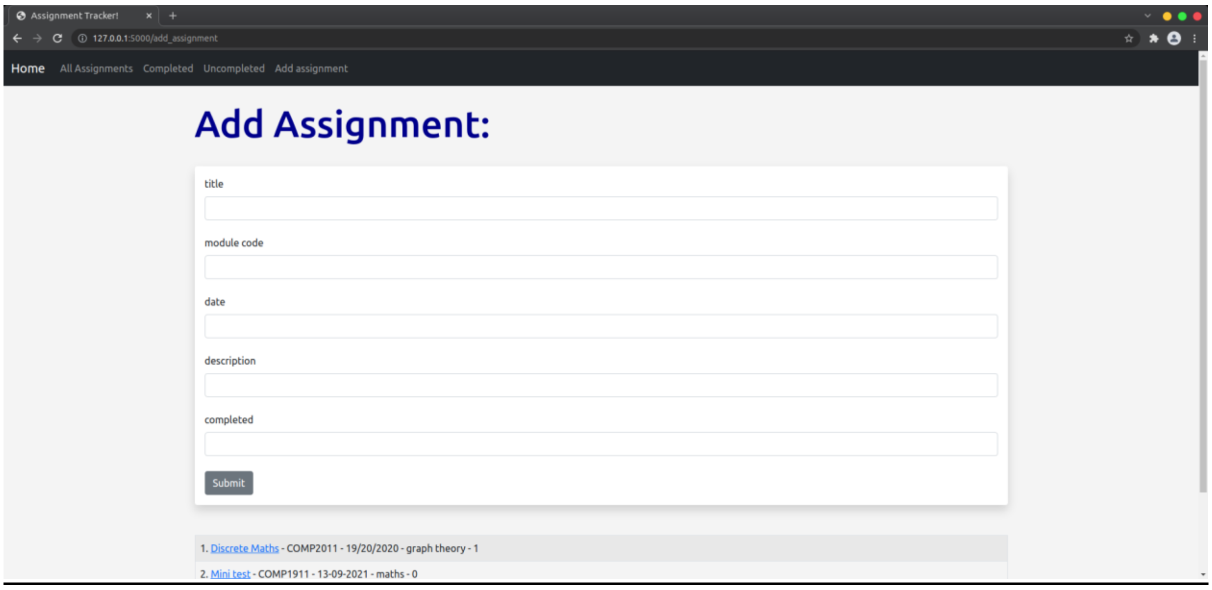
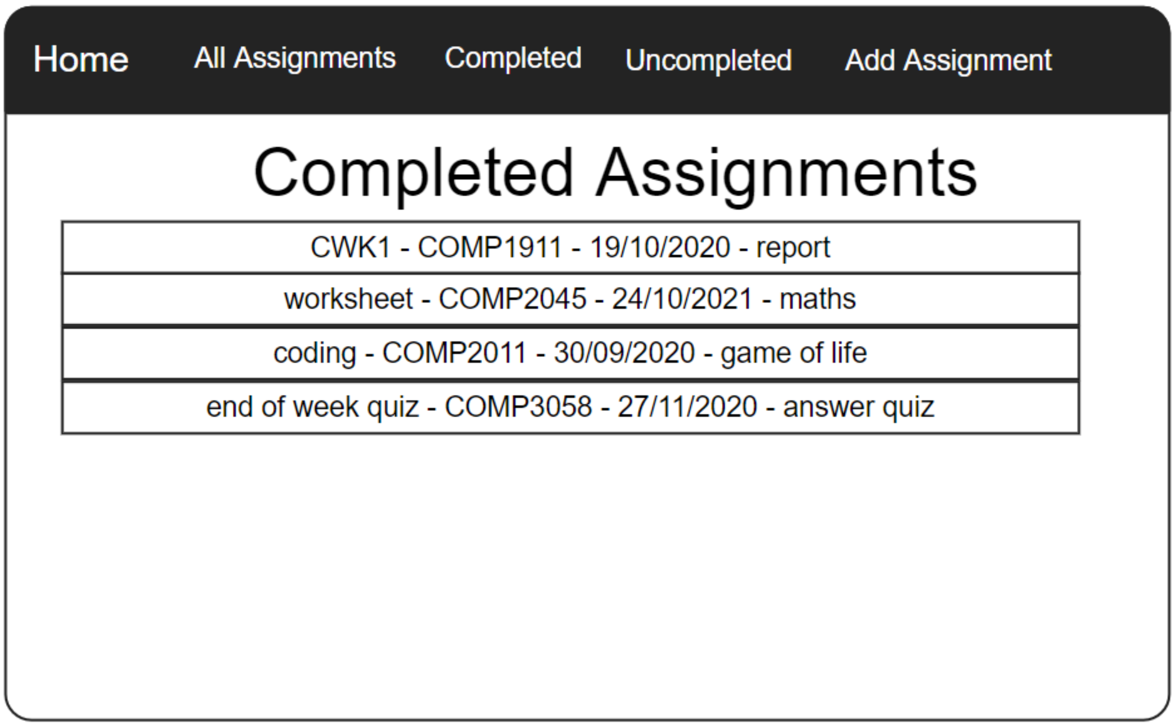
Wireframe – Add Assignment Page





1. Discrete Maths - COMP2011 - 19/20/2020 - graph theory - 1
2. Mini test - COMP1911 - 13-09-2021 - maths - 0

Wireframe – Completed Assignment Page



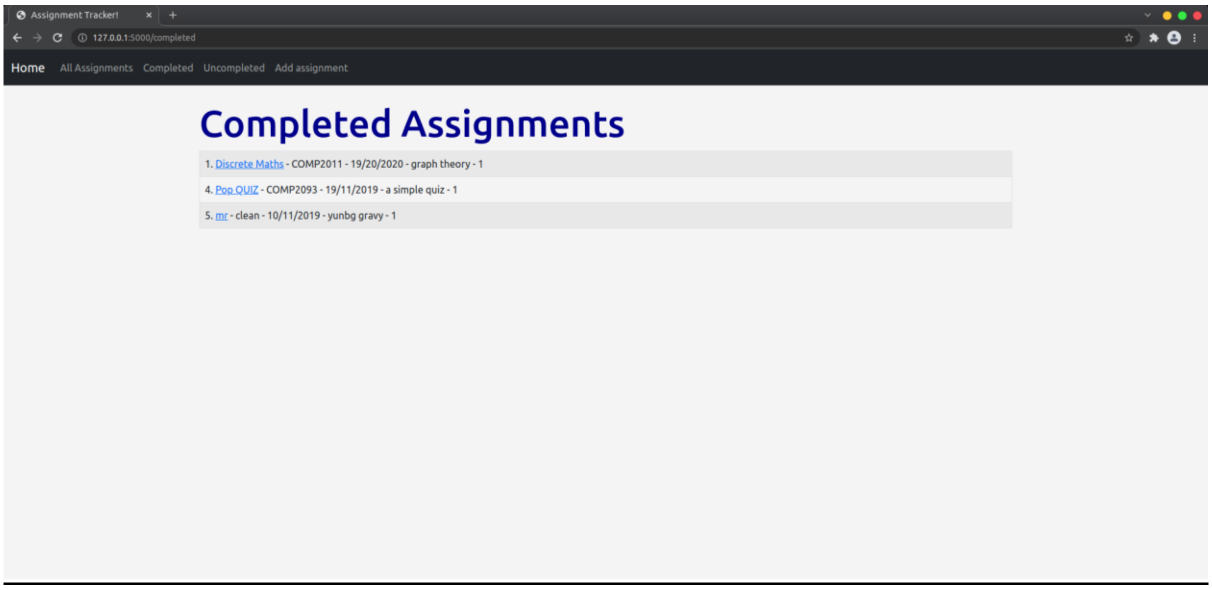
Completed Assignments

CWK1 - COMP1911 - 19/10/2020 - report

worksheet - COMP2045 - 24/10/2021 - maths

coding - COMP2011 - 30/09/2020 - game of life

end of week quiz - COMP3058 - 27/11/2020 - answer quiz



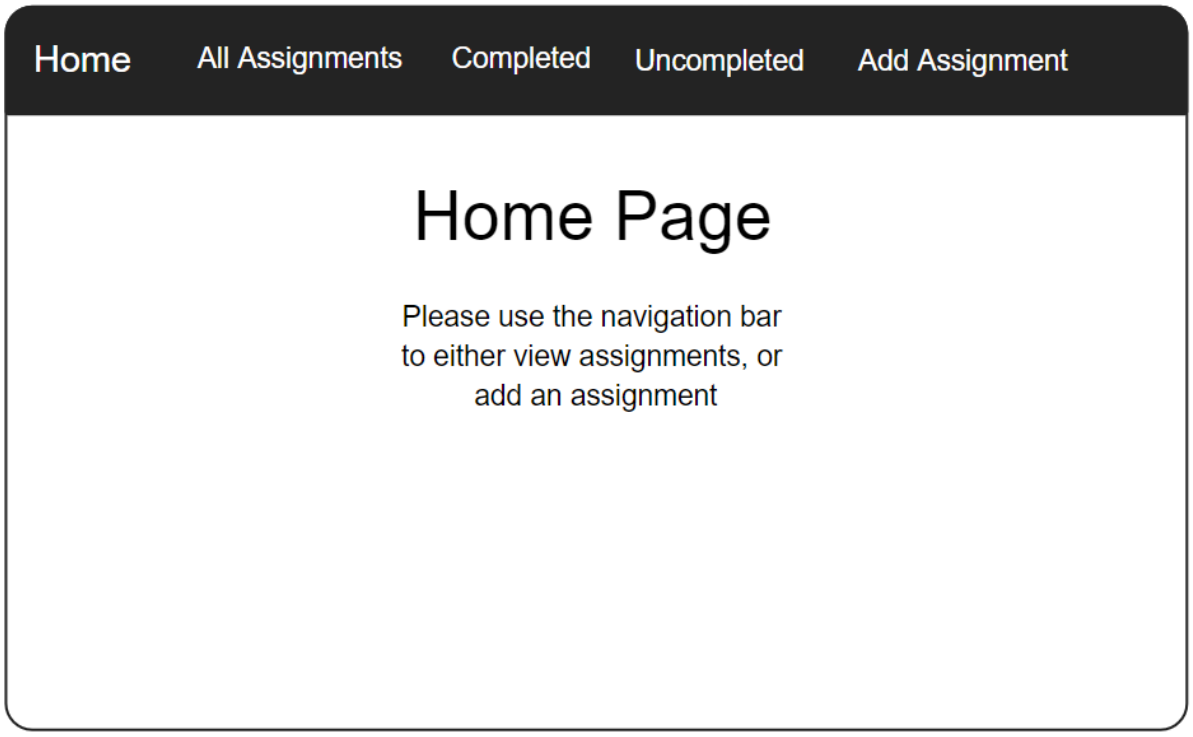
Completed Assignments

1. Discrete Maths - COMP2011 - 19/20/2020 - graph theory - 1

4. Pop QUIZ - COMP2093 - 19/11/2019 - a simple quiz - 1

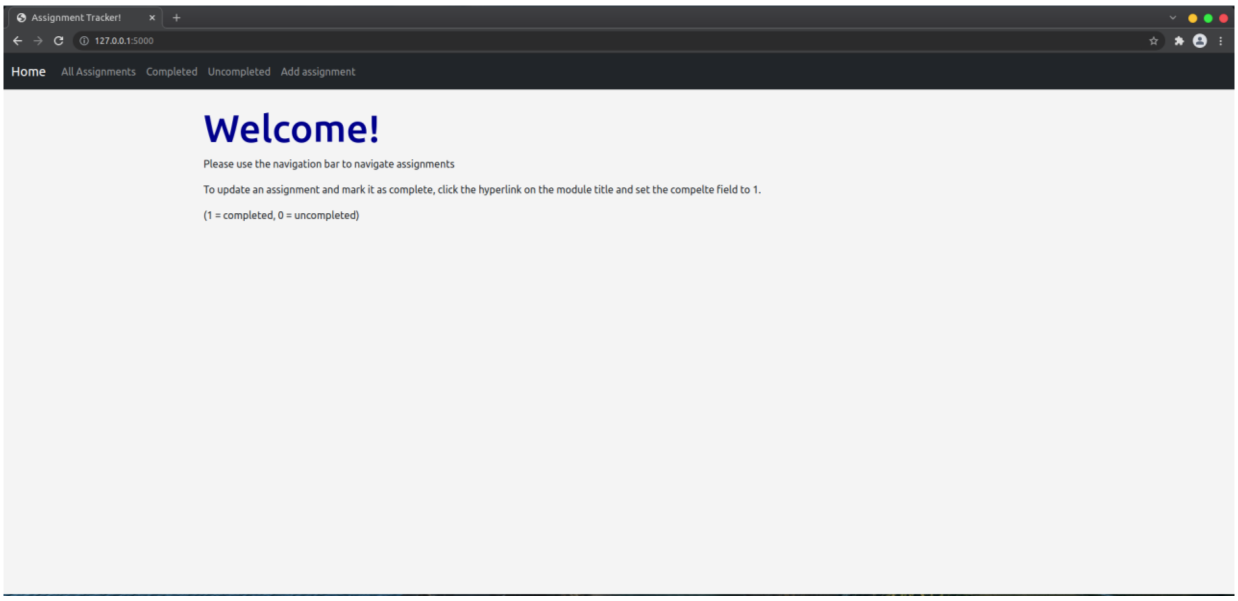
5. mr - clean - 10/11/2019 - yunbg gravy - 1

Wireframe – Home Page



Home Page

Please use the navigation bar to either view assignments, or add an assignment



Welcome!

Please use the navigation bar to navigate assignments

To update an assignment and mark it as complete, click the hyperlink on the module title and set the compelte field to 1.

(1 = completed, 0 = uncompleted)

- I chose black for the navigation bar, and white for the hyperlinks, as it is providing a contrast that will grab attention, and it will make it stand out more to be readable to the user
- I decided to have the navigation bar at the top, so that it is easier to access and look at, as it is in the line of eyesight
- I decided to make all text central and minimise the number of words used so that it is compact and clean
- I implemented bootstrap to make the website more responsive for all resolutions
- This will make the website more user friendly and easier to use, by resizing all elements to make it better for the end user.

Justification of Three Tier Architecture

This website uses HTML and HTTP requests to handle switching to/from other pages. To implement this, I have used templates so that I do not have to repeat code for similar features on all the webpages I have created. For templating, the Jinja2 templating engine has been used which allows me to use the **render_template()** function to load each .html page and pass variables. Also, Flask-WTF allows me to process data that is received from POST requests. POST requests are private meaning that you need a key to prevent CSRF.

For handling what happens on each page, Flask uses routes where you can declare which methods you will be using (POST or GET). This allows you to split each page logic separately, and create functions and variables that you will be using in each route. You can use functions in routes to check for different things (e.g., if a session has expired or a user has logged in)

For handling data access, I have used SQLAlchemy which allows me to utilise sqlite3. SQLAlchemy allows you to create databases so that you can store data that the user enters (e.g., information about the assignment that they have to complete). You can also update, delete and insert more records into a database by using **db.session.commit(),** which updates the database.

Description of request handling and HTTP features relating to web application

When a web application wants to request or send data in an encrypted/encoded form, the POST verb is used. This means that CSRF is not able to happen, which means that a user's cookie/session can't be used for malicious attacks. When a POST request is received, the body of the request is read and the request is decoded and can be used for processing. In this coursework, the Flask-WTF extension is used to handle POST requests

However, the GET method is used to get information from the web server using the URI. The GET method can easily be manipulated using CSRF as the information is saved on the web application URL.

Evaluation of web application (testing and debugging)

It is important to test and debug a web application so that you are sure that the website works as intended, and if they are any errors, they are handled so that the website doesn't crash or become unusable.

For example, in my website if the user doesn't enter any data into the title or module code fields, there will be a message flashed telling them that they have to enter data, as opposed to no message popping up and the user being confused as to how to proceed with the website.

Also debugging is important as you can see line by line how the program flows to see where the problem is, so that you can fix it to get the intended functionality. This is very useful, because it can help with areas of your code that you are certain are working, but the error lies in this code.

Furthermore, setting debugging on whilst running a flask application will mean that you don’t have to close the application every time you make an update, which can save you time.

Security of current application

The current application is secure, as I have a key and use POST methods so that CSRF is not possible. By using SQLAlchemy and Flask-WTF, this means that data will not be stored all in one place, and also means that the data that is sent and received isn’t stored in the URL.

Evaluation

After creating an assignment to do list in flask, I found most difficult manipulating the database so that you can store and update data to mark an assessment as complete. However, I now understand how templating can be used to reduce code redundancy and the jinja 2 engine is a useful tool. Another strength for me would be implementing app routes and the logic that goes in each function to implement the assignment list. Another weakness of mine was that I wasn’t familiar with using flask, so finding the right types to display on each page **(IntegerField, StringField)**, and filtering a table in a database was challenging for me.

Improvements (Accessibility)

An improvement to make will be to host the website somewhere else. This will mean that you can update the web application as long as you have the admin credentials to edit, and other people can use the website.

Another improvement will be to implement a text to speech feature where a user can say the title, module code, deadline and description of their assignment rather than typing it in.

References

POST and GET methods: https://rapidapi.com/blog/api-glossary/get/