INFO2222 Usability Report

2021 Semester 1

THE UNIVERSITY OF SYDNEY

Team:

INFO2222_CC08_M12_B2 Excelsiors

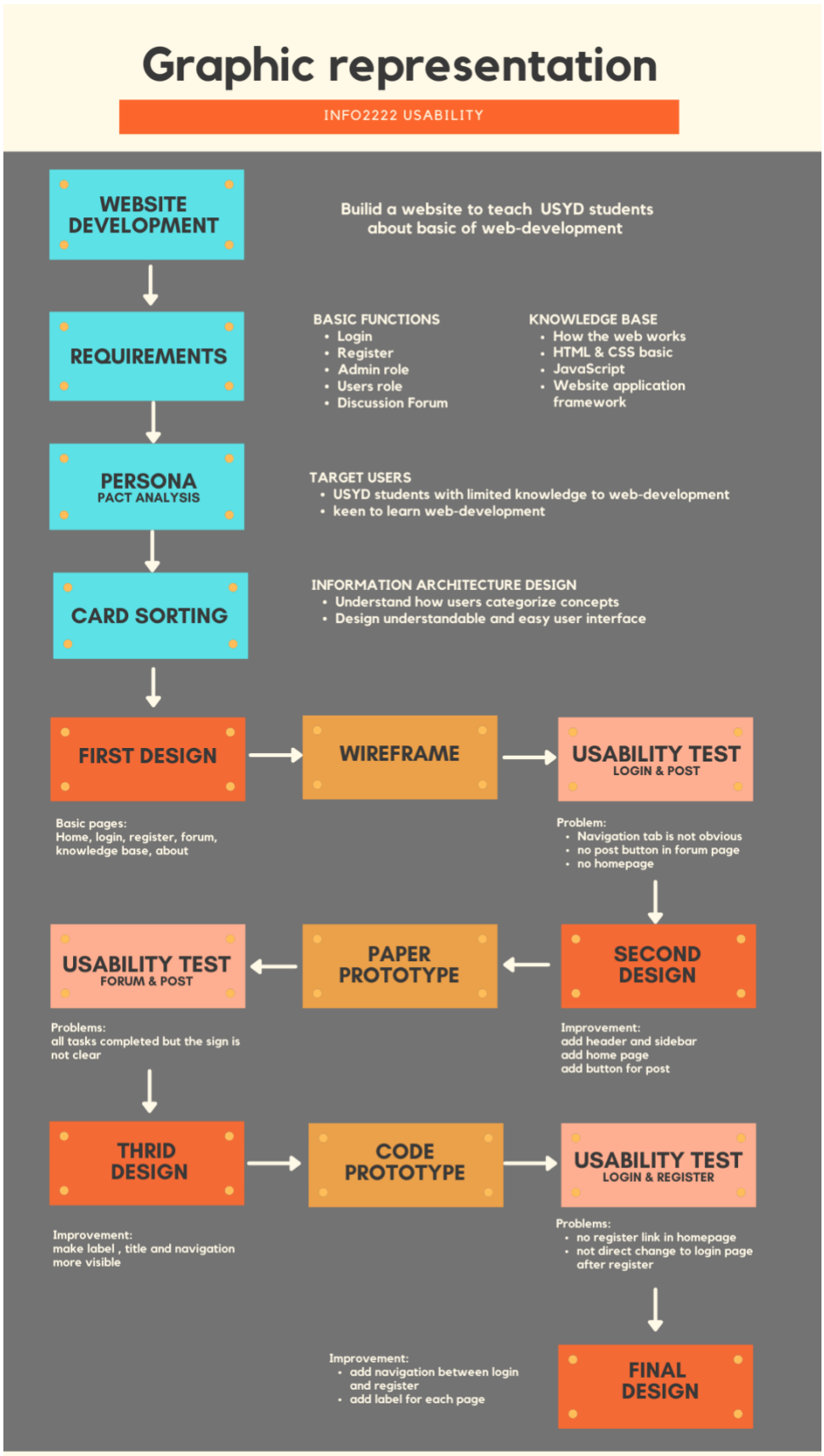
Members:

Name	SID
Qihan	500304981
Alan	500170629
Jiahao	500214288
Alison/Yisheng	500520857
Vincent	500202148

Table of content

Design Process

Click on the area in Graphic representation to go to the relevant section of the report.



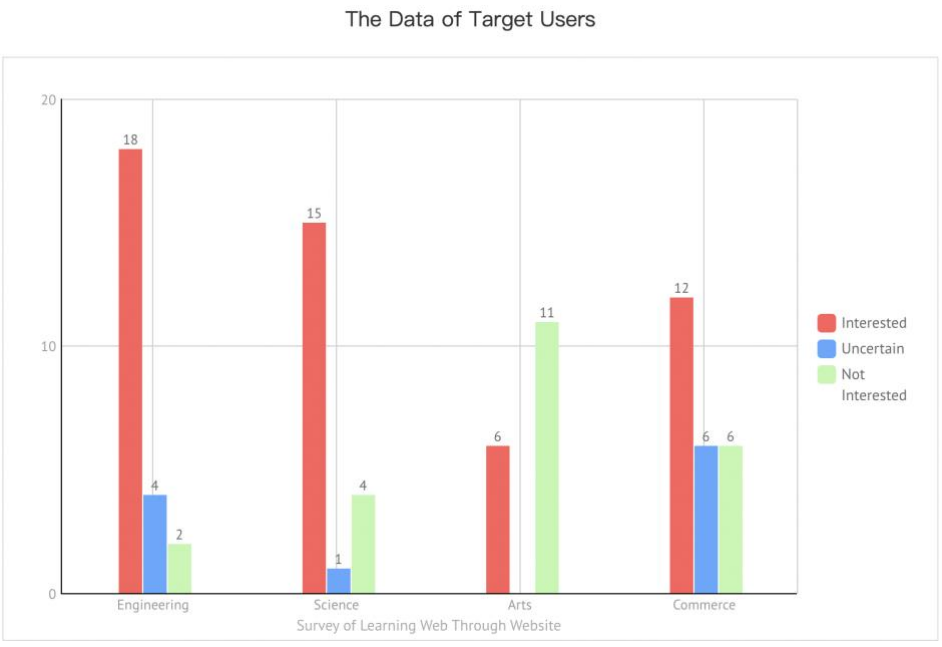
Brief description

During the project, our group took the human-centered design approach to ensure the usability of the website. We early paid great attention to the tasks and users. We conduct the PACT analysis to focus on the target persona for our website and understand their needs. Conducting card sorting helps us to understand how users conceptualize so that we use labels that are easy to understand. Empirical measurements were considered and made. Moreover, we used iterative design (design, test, measure, and redesign) as the cycles shown in the graph above. Many prototypes and artifacts were produced at various stages. Plenty of problems were found and fixed during processes and many improvements were made. Finally, we got a website that meets all the requirements of the project. By this process, our team achieves usability so that the target user (USYD students) can achieve their goal of gaining basic knowledge for website development with ease, efficiency, and satisfaction.

Persona

Survey

A survey was done on a group of USYD students to investigate their interest in learning website development. After investigation, a result is concluded that most users have engineering and science backgrounds and have a high interest in learning how to build a website for various purposes.



We only collect a small-scale sample so that the result is not sufficient to give rise to a definite conclusion. However, our team believes the users of the website would focus on people with high interest and some need for website development. From this, we create the following persona.

Persona

Photo:	 Photo from Uni Reviews - University of Sydney Reviews
Fictional name:	Jack Lee
Job title/ major responsibilities:	As a student, he learns knowledge as well as gets a better understanding of the society and industries.
Demographics:	● 18-24 years old ● Married/Single ● Has graduated from the senior high school ● Doing a Bachelor or Master in any degree
Goals and tasks:	● He is passionate to learn about web development skills so that he can use them to communicate with the wider society. He is also goal-oriented within an ardent desire for self-success and social responsibilities. The desired further career of him would be a software developer and web developer. ● Spends his work time: ○ studying for his university degree ○ paying close attention to the social issues ○ learning the web-develop skill by himself
Environment:	He comes to campus several days a week or studies remotely. He keeps the computer and phone with him most of the time. He has basic computer skills and knowledge about the Internet.He meets with other students/tutors or communicates with them from Monday to Friday. He has some free time after finishing his study task from Monday to Friday and is more available on weekends.
Quote:	“It’s a good day to learn something new!”
Frustration:	Want to get high grade points, but like to play games, sometimes staying up late to play games leads to low learning efficiency the next day.

PACT Analysis

PACT	Description
People	**Users:** USYD students with limited skills on HTML/CSS, web development and Computer Science in general but have interests in Web. **Admin:** professors in the area of web development, passionate tutors who would like to offer help and maintain the forum environment. **Software developers:** develop and design a comfortable and easy environment and maintain the usability of the website**.**
Activity	The learning activities for users would happen once to twice per week for two hours and at any time when they have access to the internet and computer. The activities have great flexibility and independence. They would be required to do activities like register, log in with username/email or password. If a mistake like forgetting a password happens, they would need to reset via email. The admin can mutes users, delete posts, and maintain the website.
Context	**Physical** Anywhere with a computer: home, Internet bar, in the office, etc **Social** Group up with other people interested in learning web development on the **Internet** interact with professors of web development **Organization** User base is usyd students
Technology	**Backend database** to store users' basic username, user id, password, admin and muted. Forum backend includes posts, comments and time.

Analysis and Conclusion

The target user is keen to learn about web development but has various backgrounds in computer science. Thus, the content needs to be oriented, so it is suitable for various levels of learners. Also, they are helpful to others, so a forum page can let some more users answer questions from others. Both the suitable difficulties levels and promoting learning communities help to achieve their goal of learning website building with satisfaction.

Creating the persona helps us to make our website right for the target users and understand the user's needs in a greater depth. This takes into real consideration the user so that we choose the prioritized features during our implementation such as the knowledge pages with various difficulties. This ensures that different users feel they get suitable resources for their needs and achieve learning goals with ease and satisfaction. Hence, so usability is ensured.

Card sorting

Our team conducted a closed card sorting online [https://study.kardsort.com/website-open-card-sorting] to investigate the how the user sort the contents into following categories:

- Forum
- How the web works
- How to use this website
- HTML and CSS
- JavaScript
- Web application framework

Result of Card Sort

User count: 14; Categories count: 6; Cards count: 28

Card	Forum	How the web works	How to use this website	HTML and CSS	JavaScript	Web application frameworks
CSS Advanced (5)				14		
CSS Basics (9)				14		
CSS Intermediate (1)				14		
How can I make post and comment on other''s post? (20)	9		5			
How do I change font style and size in CSS? (6)	1			13		
How does the basic website work? (21)	1	11	1	1		
How to change the background of CSS? (8)	1			13		
How to create a forum post? (19)	8	1	5			
How to do arithmetic operation in JavaScript? (12)					14	
How to make a table in HTML? (4)			1	13		
How to register? (15)	1		13			
How to search for the help? (14)	3		11			
How to set up local environment? (27)	2	2	1			9
How to set up the height and width in CSS? (7)				14		
HTML Advanced (2)				14		
HTML Basics (3)	1			13		
HTML Intermediate (23)	1			12		1
JavaScript Advanced (10)	1				13	
JavaScript Basics (13)	1		1		12	
JavaScript Intermediate (11)				1	13	
What do I do if I forget my password? (17)	1		13			
What is an user interface? (28)		10				4
What is bottle in python? (25)	1	3			1	9
What is route and how does it works? (24)		6	1	1		6
What is the website frame and how it can be established? (26)	1	6	1	1		5
What is this website about? (16)	3	2	8			1
What is website server and how does it works? (22)		12	1		1	
Where can I get help? (18)	4		10			

From the results, we can tell that most users sort the CSS, HTML, and Javascript with the corresponding contexts. However, the label "forum" and "How to use the website" has crossing contents suggesting these two labels are misleading. The user

also can not distinguish the label “Web application framework and How the web works”. This might be due to the fact the users do have efficient knowledge of the website.

Analysis and Conclusion

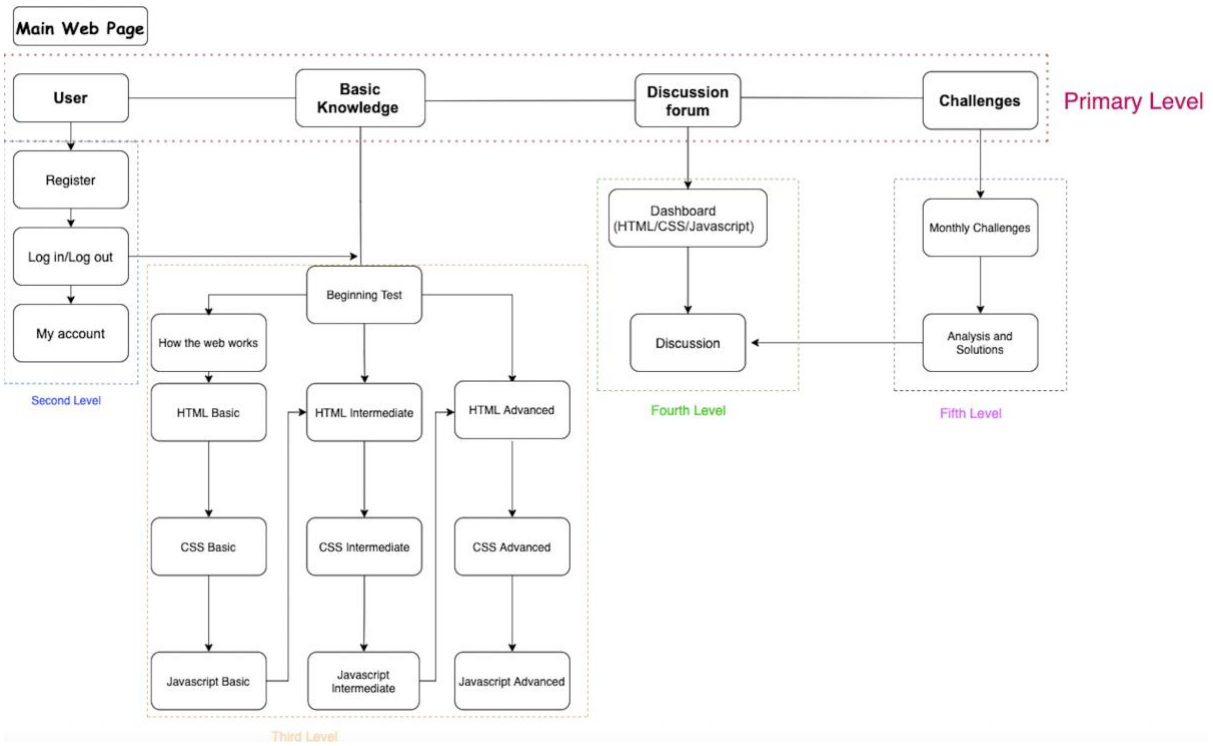
The purpose of this card sorting session is to help us to evaluate the information architecture design of our website. This technique helps us to understand how people expect, understand, and conceptualize content. This helps to classify the content being put on the homepage, label categories, and build the structure of the website. In this way, the overall design and the navigation will be more comfortable for people to use so that usability and accessibility can be achieved.

The implication of card sorting toward our project is that we understand users can conceptualize the knowledge content easily, suggesting the categories meet users’ expectations. However, we found out that users generally mix these two labels “forum” and “How to use the website” so we decide to combine them into the forum where FAQ can include the latter. Moreover, the labels “how the website works” and “web application” are blended into the knowledge page and can be searched in the forum.

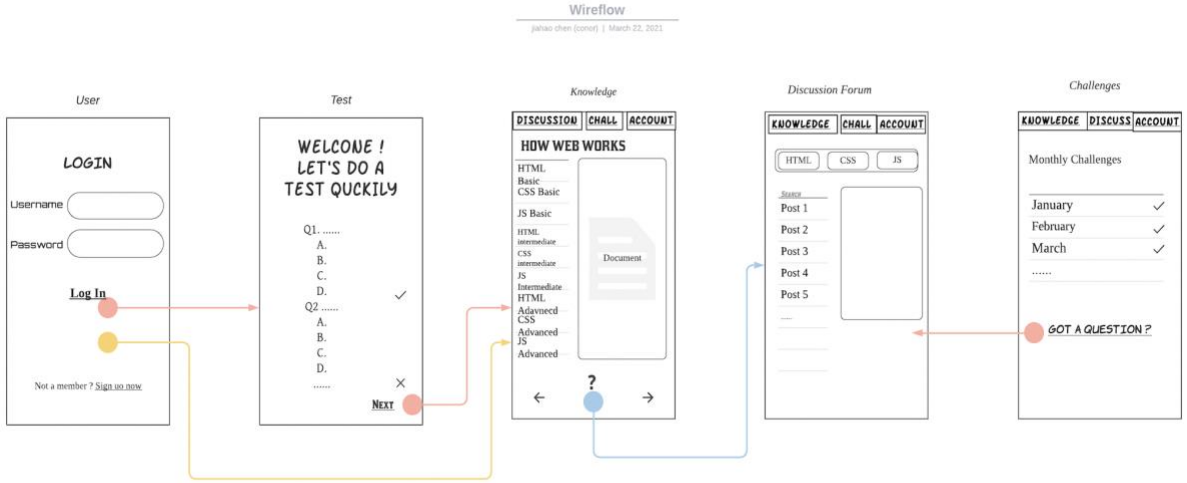
Interface Design and Guideline

Interface Design - From Prototype to Final Product

Prototype



The sitemap including all functions of website



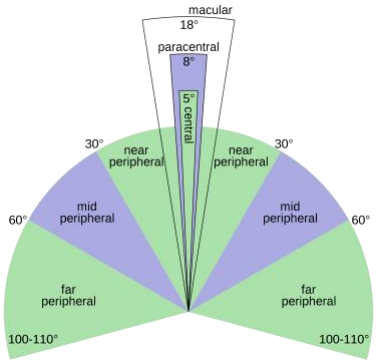
Wireflow diagram of initial site design

The prototype consists of a sitemap and wire flow diagram. We used the wire flow diagram to conduct the first usability test and collected feedback for improvements. The complete description of the first usability test is in User Evaluations.

First Iteration

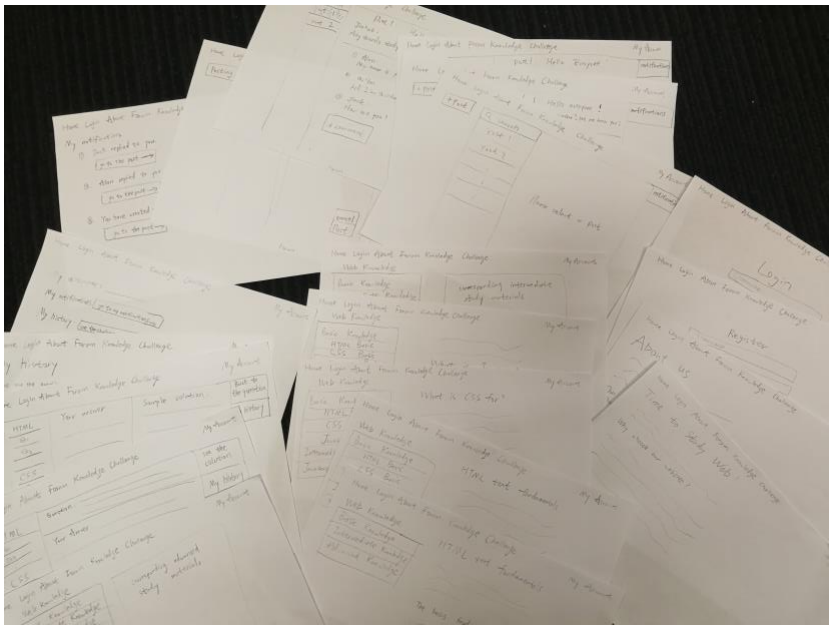
The first iteration was about adding new functions above the template, thus it does not have design implications.

On the other hand, the second usability test conducted using this version of the site collects meaningful feedback on design. We recognized buttons in the body section - which are within the user's fovea vision range - are better than the header in terms of navigating through the site.



"Peripheral Vision" from Wikipedia

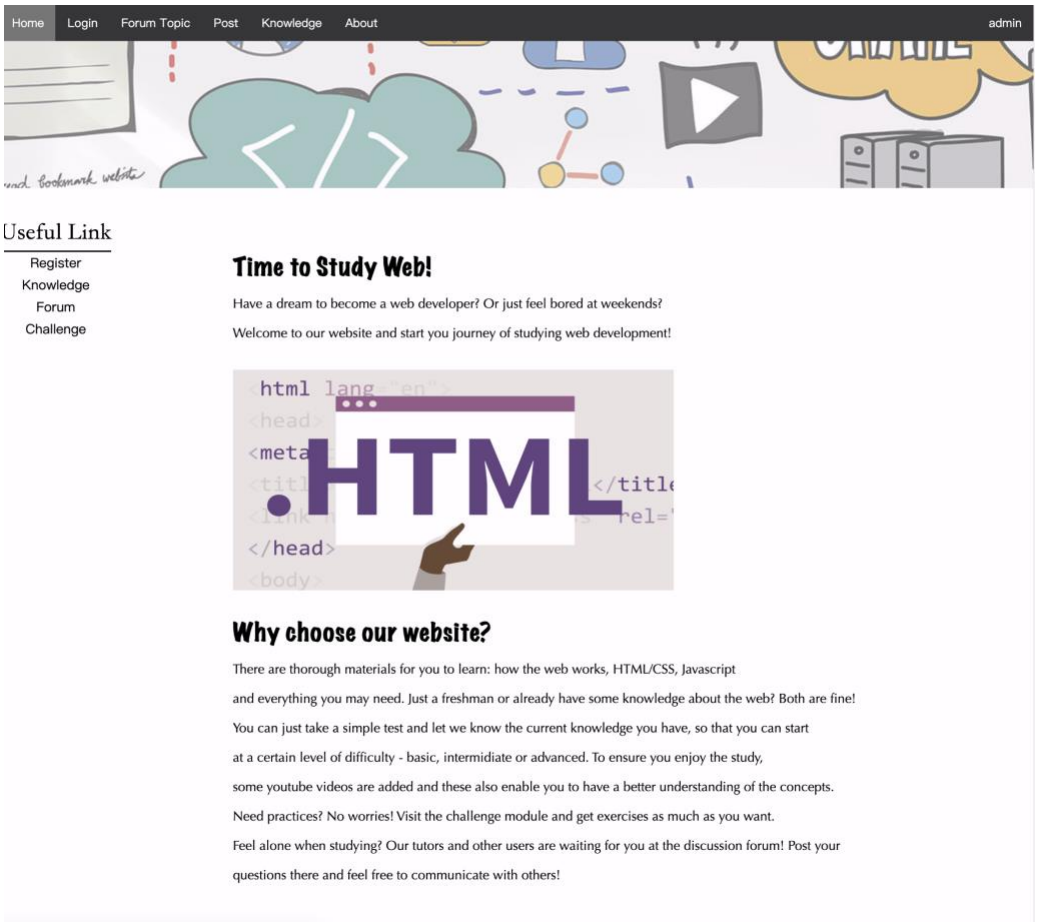
Second Iteration



Paper prototype completed on par with alpha version of website

The second iteration was developed following design guidelines. The index page was re-designed to add all functions of the website in body. Error notification pages are created to show that an error happens.

Third Iteration



The index page of final product

During the implementation stage, we found that using separate error notification pages was a waste of resources, thus we moved all errors and notifications into one dynamically generated page and added a return button to the page before action. The introduction of the “return” button also updates our design guidelines.

Design Guidelines

We start the design following the three notable design principles: visibility, affordances, and feedback.

Our team designed to bold and give greater font size to the main functions and so that the user can easily identify which page they are currently at by shaded header and also which function of the website they have access to. The clickable tabs will have different fonts and colors. They will be located at the sidebar and the top which is easy to notice, thus achieving the affordances.

When header is clicked

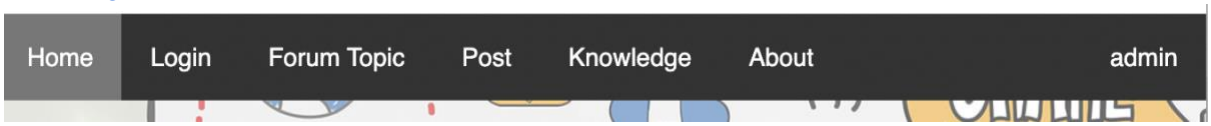
When the user moves the mouse over a clickable link, the side tab would become grey and the header will be blue to make access clear to give the user instant feedback.

When sidebar is clicked

However, due to our skill limitation, the more advanced feature like audio is discussed but not implemented.

Then we follow the Nielsen 10 Usability Heuristics to ensure that our user interface is easy and comfortable to use during the learning process.
For visibility of system status, we use a dynamic header to indicate the login status such as login or not, and the users. We originally have a different font and size for the header but it is changed to meet the “Consistency and standard”.

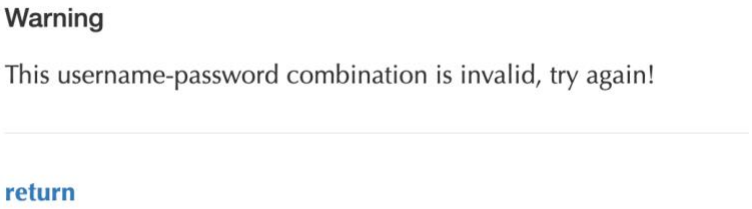
before login



after login

We “help the users to recognize, diagnose and recover from errors” through the warning. Any error or notification must have a “return” button for easy recovery. Specifically, errors can be classified as slips and mistakes. In the first sense, we need to build suitable mental models, offer suggestions and use forgiving formats to prevent these 2 kinds of errors. However, errors do happen, so we make sure the system can return to the former valid pages, thus making the system robust. For

example, “invalid username and password” and they can return to the login or the register page easily.



warning for invalid login

Our navigation is improved so that users can “recognize” what they should do by the label rather than “recall”. For example, buttons to the user's next target page (e.g. login after register and forum page after post) need to be inside the body section, so the user can easily navigate.

At the last stage, we also consider the WCAG guideline to improve accessibility so that more users can access our website.

Perceivable and Operable are key elements in the WCAG guidelines which means the users should be easy enough to see the information and use keyboard navigation. So a button here makes it obvious and helps to find content.

Send, post and other buttons to post information should be located near the form to type in the information. This follows the understandable guideline since it logically makes sense and makes the layout predictable. From a user’s perspective, the willingness to discuss and share is retained if they can access the post buttons quickly.

Reflection of Design Process

During the development process, we found that not only usability tests can find limitations on website design, but the development process can also help find design flaws and areas of improvement.

User Evaluations

The evaluations and usability tests below make sure that our design can meet the requirements of the target users and improve accessibility and usability.

User Feedback of Initial Design

The role of the admin needs to be defined. The content was broken down into HTML, CSS, and JavaScript, but this categorization was not good enough for a tutorial website. The categories of the forum and other knowledge contents are not separated and the user finds it confusing.

Analysis

The role of the admin is to moderate content on the forum page to remove spam and unrelated posts and mute users.

As the persona shows, the user might have little knowledge about terms in web development. Thus, each section was broken down further into basic, intermediate, and advanced to suit the programming skill of the users. As the result of the card sorting shows, the labels can not provide correct information so that we decide to merge the “web framework application” and “how the website works” into knowledge, and the FAQ is included in the forum and can be searched easily.

User Feedback from Usability Tests

First Usability Test

The first usability test was conducted using the wireframe diagram. The user completed the task of finding the basic JavaScript page and read it but failed to recognize that the question mark sign is used to go to the forum page. The first task was completed within a minute.

Analysis

New header layout

Since the user finds it difficult to find the path to the forum, we move the forum page link to the header section of the website so it is easy to navigate on any page. This increases usability since users can be redirected to any page. Also, we included a link to the forum on the index page. The size of the buttons is adjusted to make them more noticeable and text description is added to ensure users understand what they mean.

Second Usability Test

The user finds the task of login using the provided account and posts a new thread on the forum page. But the user finds it difficult to relate a question mark sign on the index page with hyperlink to forum, which results in the user taking 5 seconds to find the appropriate link in the index page (the user found it on the header at the end).

Analysis

The layout of the index page needs to be updated so that every function of the system is shown clearly. Then we also added the “post” tab to the header and a link with text description under the forum page so that the users can post more conveniently from different positions.

The “Useful Link” section was added to the index page

Third Usability Test

The third usability test was conducted using an alpha version. The user completed the task of locating the forum and posting a comment in 1 minute. On the other hand, the user does not find a register button in the login page, and a login button in the register page so he took extra thirty second to register an account and login.

Analysis

It is confusing for a new user when there is no register sign in the index page and inconvenient after success registration the login page is not returned.
We add a register button on the login page and self link to the login page after successful registration to make the register and login process more easy and efficient to achieve usability.

Implementation

First Iteration

The first prototype was focused on implementing the backend functionality and lacks refinement on the user interface thus usability.
We developed the login and register feature using SQL database.

```
# Check login credentials
```

```
def check_credentials(self, username, password):
    query = "select * from Users where username = ? and password = ?"
    self.cur.execute(query, (username, password))
    # If our query returns
    response = self.cur.fetchone()
    if response:
        return response
    else:
        return None
```

This type of SQL query function was introduced in sql.py for add, delete, and search users.

```
# Check the login credentials
def login_check(username, password):
    """
        login_check
        Checks usernames and passwords

        :: username :: The username
        :: password :: The password

        Returns Cookie or None
    """
    sql_db = sql.SQLDatabase()
    res = sql_db.check_credentials(username, password)
    sql_db.close()
    return res
```

The check function in model.py analyzes results from SQL query and returns user information.

```
# Attempt the login
@post('/login')
def post_login():
    """
        post_login

        Handles login attempts
        Expects a form containing 'username' and 'password' fields
    """

    # Handle the form processing
    username = request.forms.get('username')
    password = request.forms.get('password')

    result = model.login_check(username, password)
    if result is not None:
        response.set_cookie("login", result[1], secret="1B32E674-443E-4602-89EA-
643ACF6FD637")
```

```
        return model.header(request, model.custom_error, 'This username-password
combination is valid, welcome! ', '/')
    else:
        return model.header(request, model.custom_error, 'This username-password
combination is invalid, try again! ',
                            '/login')
```

This controller function receives user information from the request, analyzes the result from the query and sets the user's cookie in an encrypted form to save the login status.

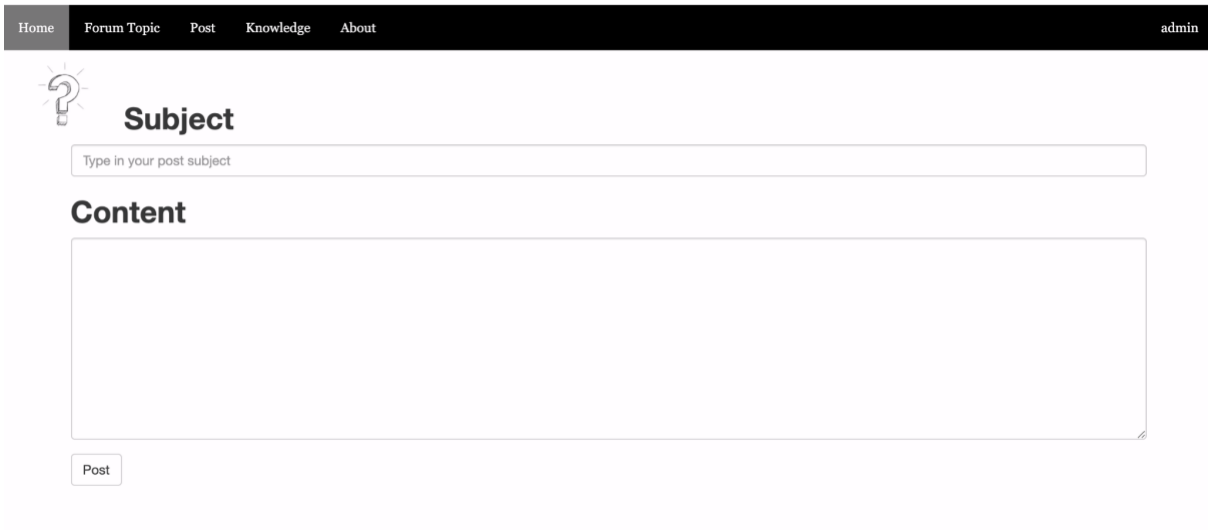
The register function was developed using similar logic, where the controller analyzes user requests, and the SQL insertion was completed using a function in sql.py.

Evaluation

The backend is prioritized since we believe a website without errors when using is the first requirement. This ensures the users have the correct information and use the website without frustration and can achieve their goal properly.
Using browser cookies to store the status of login is good since it will be invalid in the next browser session. Login using username and password meets user expectations.
The login and register pages are then improved with aesthetic design and placeholder to help the user identify these functions with more satisfaction. This implementation conforms with the initials design.

Second Iteration (with feedback from second usability test)

In the second iteration, forum functions were included.

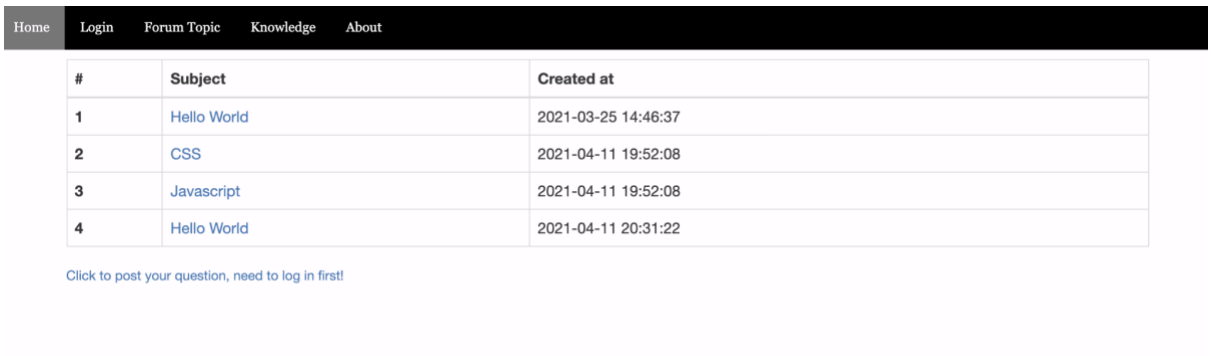


Screenshot of post page

```
@post('/post')
```

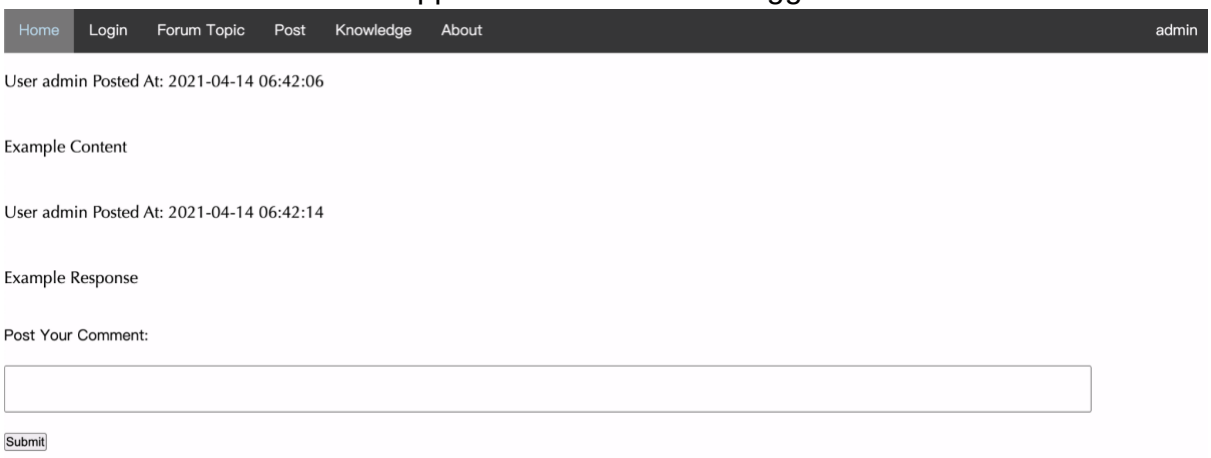
```
def post_request():
    cookie = request.get_cookie("login", secret="1B32E674-443E-4602-89EA-643ACF6FD637")
    result = model.verify_cookie(cookie)
    if result is None:
        return model.header(request, model.custom_error, 'You cannot post if not logged in!', '/login')
    user_id = result[0]
    thread_id = request.forms.get('thread_id')
    subject = request.forms.get('subject')
    content = request.forms.get('content')
    result = model.post_post(user_id, thread_id, subject, content)
    if result:
        return model.header(request, model.custom_error, 'Post successful', '/forumtopic')
    else:
        return model.header(request, model.custom_error, 'Post failed', '/post')
```

The post request function verifies the user's login status, then posts the content using model.post_post(), then returns an error(notification) page to show that the user is successful or unsuccessful on posting the page.



Screenshot of topic page as normal user

The remove function will disappear if the user is not logged in as admin.



Screenshot of thread page

```
def query_post_by_thread(self, thread_id):
   query = "select Posts.subject, Posts.body, Posts.created_at, Posts.topic_id,
Users.username from Posts " \
           "inner join Users on Posts.user_id = Users.Id where topic_id = ?
order by created_at"
   self.cur.execute(query, (thread_id,))
   result = self.cur.fetchall()
   return result
```

JOIN function is used to “combine rows from two or more (related) tables”. (w3schools)This case we use it to show username and response with a single SQL statement.

```
for column in result:
   body += '<div class="card"><div class="card-header"><p>User {} Posted At:
{}</p></div>\r\n'.format(column[4],

column[2])
   body += '<div class="card-body">'
   body += markdown.markdown(column[1], extensions=['fenced_code'])
   body += '</div>\r\n</div>'
```

String formatting was used to generate HTML for pages containing dynamic content.

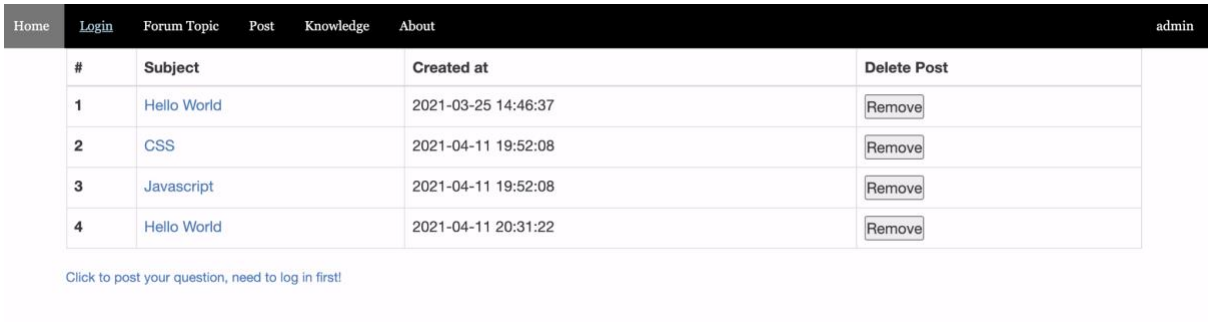
Evaluation

String formatting is not the optimal choice for dynamic pages such as the forum, a more optimal version should pass the value by JSON, then let JavaScript code in the front-end to analyze it and display the content on demand.

However, this implementation differs from our original design where we initially decided to put all the posts and click a new post to be the right. This is due to our limitation to website development to generate a page with two separated parts. The output of our website separated the forum and the post into two pages but still have the same functionality and usability.

Third Iteration

(with feedback from second usability test and design guideline)
In the third iteration, the admin functions were included.



Screenshot of topic page as admin

Screenshot of user page as admin, the label is "admin" indicating current logged in user

```
# admin-specific functions
def is_admin(request):
    cookie = request.get_cookie("login", secret="1B32E674-443E-4602-89EA-643ACF6FD637")
    result = verify_cookie(cookie)
    if result is not None and result[3] == 1:
        return True
    else:
        return False
```

The is_admin() function in model.py is developed to verify that if the user using admin-related function is logged in as an admin. Although admin function buttons will not appear for normal users, this function can find out program generated posts without valid cookie information.

```
def mute_user(self, user_id):
    query = "update Users set muted = case when muted = 0 then 1 else 0 end where Id = ?"
    self.cur.execute(query, (user_id,))
    try:
        self.commit()
    except mariadb.Error:
        self.rollback()
        return "Database error"
    return "Done"
```

"CASE WHEN" SQL operation goes through conditions and returns a value where the first condition is met. (w3schools) In our website, this statement is used for functions that can switch a status like a user's mute status between true and false.

Evaluation

The implementation meets our design with a dynamic header to indicate the status of the users so that they can perform different activities. The user page is located on the right side of the header, labelled using the username of the current user. However, during the usability test (demonstration) it was not clear for the participant what the button means. Thus, the button's layout needs some refinement. We discuss adding a new page on the header for admin only to mute/unmute an user and delete a post. This might be more visible and clear to user hence can improve the usability.

Conclusion

achievement

We achieved the goal of the project to develop a usable system, using user research and iterative design (design, evaluation, re-design) to make sure our website satisfies the requirements of target users. Procedures included PACT analysis, developing personas, card sorting, making several prototypes. The human-centred design approach was also considered during the process to guarantee accessibility and usability, so that any USYD student who visits the website can register, login, study the contents of the Knowledge base (how the web works, HTML, CSS, Javascript, etc) and communicate with others on the forum efficiently. Other core features including Admin role are also implemented to maintain the functionality of the website. In a word, the website meets all of the requirements of the project.

Limitation

we admit that there are some aspects we can improve. Due to the time limit, we were unable to do further adjustment to make the view of the website look better. The lack of asceticism and the plain label might lead to dissatisfaction with users' experience. For example, the indirection after the register can be displayed in the same format to maintain consistency. More usability tests can be done so that it can become more appropriate for the target users. Moreover, when writing pages, the CSS styles adapted by group members are diverse, making the codes a bit messy. The user interface is unstable since when the page changes the format sometimes changes which is lack of consistency. These problems will be fixed in the future when we conduct the security part.

References

Note

The project is a collaborative work of team excelsior. To find which member completed a specific work, please see the team meeting document at /meetings in our GitHub repository, the link to the repository is shown below.

1. SQL JOIN Statement. (n.d.). Retrieved from https://www.w3schools.com/sql/sql_join.asp
2. SQL CASE Statement. (n.d.). Retrieved from https://www.w3schools.com/sql/sql_case.asp
3. Peripheral vision. (2021, March 19). Retrieved from https://en.wikipedia.org/wiki/Peripheral_vision#/media/File:Peripheral_vision.svg
4. University of Sydney Reviews by Students. (2019, May 31). Retrieved from https://universityreviews.com.au/students/nsw/sydney/
5. INFO2222_CC08_M12B2_Excelsiors . Retrieved from https://github.sydney.edu.au/qcai8733/INFO2222_CC08_M12B2_Excelsiors