North South University

Department of Electrical & Computer Engineering

Course Name: **CSE-311 Lab** Section: **7** **Group-14**

Project Title: **Web Novel Hosting Site**

Shadman Sakib - 2014310042

Nadia Binte Rahman Peeya - 1731358642

Score

Remarks:

Introduction of the Project:

The project we worked on is like any typical blog website. More specifically it is a website where users can read web novels and discuss about them upon logging in to the system.

The login system accommodates two types of logins, a 'user' login and an 'admin' login. Each login has a unique landing page from the others. Admin logins have access to more functions and views in the website than regular user logins. For example, admins can read, write, and delete comments (of any user) in any chapter page. But users can only read and write comments in any chapter page.

The comment system is no doubt the highlight of our project as it encompasses many complex backend functions. Each chapter has its own set of comments created by separate tables in the database. The list of comments that can be viewed in a chapter is limited (currently limited to 50 at max when 'show more' button is clicked) to prevent long load times.

Admins have an access to whole separate page called admin-panel. This page has most of the complex database queries. They can view all list of users signed up to the website. Password of accounts is however not visible to them. They can view joined data, result of complex sub queries and edit some data values and much more.

While it may seem like any typical website, we have managed to code it with use of only html, Css, php and mysql. JavaScript was used only twice in the project for very minimum search functions. The extensive use of php and mysql for almost everything is what makes this website unique to others. Not that, this is a good thing in general, but it is great feat considering the project was made for the course aiming to teach php and mysql.

Background research/Motivations:

As mentioned in the introduction part, this website is made with only html, Css, php and mysql. This case is not true for almost any website in the market. That is because the likes of JavaScript or Ajax etc are all really good for not only making coding simpler, but also makes the website more interactive than you possibly could with only php as the back bone. The front-end part of the website is also not very common, as we had decided to implement it with just raw Css.

The reason we chose to do this, despite the drawbacks was that it was for one more challenging making it fun to do, and it allowed us to raise our skills in php, sql, and css significantly, which was the purpose of the project to begin with. Though needless to say, I wouldn't do this again, unless it was for learning purpose once again.

Why this project? How this will help people or solve a problem? Are you trying to solve a problem from your experience?

Let's try to answer this one by one. We decided on this project because we really wanted to work with comment section when we first brainstormed ideas. Almost everything in this project seemed to be possible to execute with pretty simple html, which would allow us to focus on how comment section works. Thus, we went with it.

This project will not help the general public/user. However it might help other young developers learning back-end of websites. The problems that could a rise in making the comment section or unique landing page for different users were all broken down very simplistically making itself explanatory. Most of the code is also commented with brief detail about its function, and the Css files are also easy to read and reusable (do note, I do not own the images used).

Finally, onto the last question my answer is NO. I am not trying to solve a problem for the general public with this project. I was simply motivated to work on this project because it would help me solidify my basics of php, mysql, css and html.

Resources/technologies

Project utilized html, css, php and mysql, and 2 instances of JavaScript.

Every code in the project is a result of trial and error. But here are some mentions of resources that helped me;

- W3schools was my go to website when I was stuck with any html or css problems. https://www.w3schools.com/css/

https://www.w3schools.com/html/

- The two JavaScript codes implemented for interactive search were ripped from https://www.w3schools.com/howto/howto_js_filter_table.asp - in admin panel
  https://www.w3schools.com/howto/howto_js_search_menu.asp - in chapter list
  these 2 links has also been mentioned in the comments of the code utilizing them.
- Several youtube videos also gave an idea for how to approach the problems, but none in specific.

Project Design

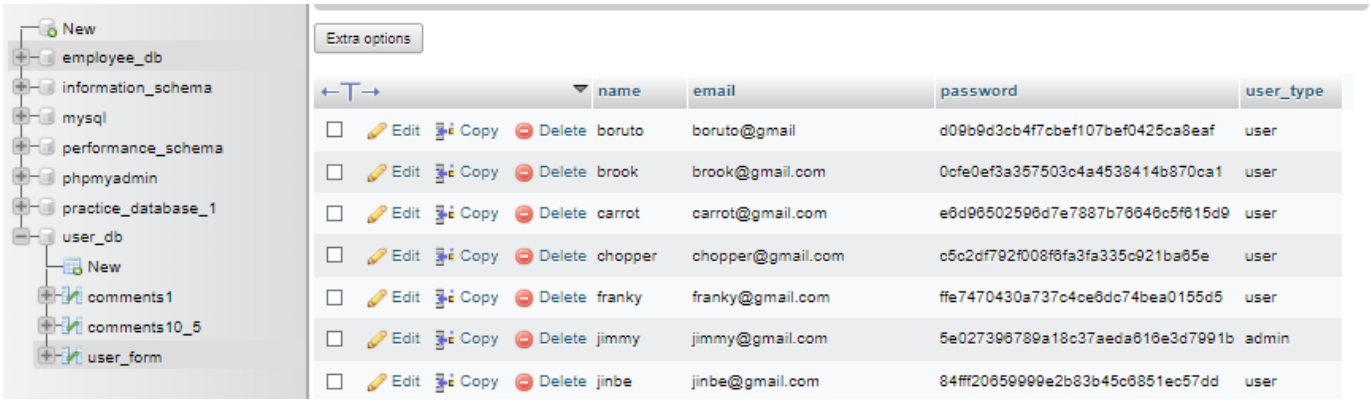
User Types: 'admin' and 'user'.

As mentioned prior, admin has more views than user. They have access to a whole page called admin panel that is inaccessible by user types. Admin can see delete button on comments which is also inaccessible to users.

Each login is met with different landing page by use of $_SESSION. Landing page is different because the page displays name of the user logged in and the sidenav also looks different for user and admin login. Some pages use fixed height sidenav while some don't depending on the requirement of page height. The Same situation is true for footer. These are some front-end aspects so I won't go into detail for them here.

Moving on to the database design;

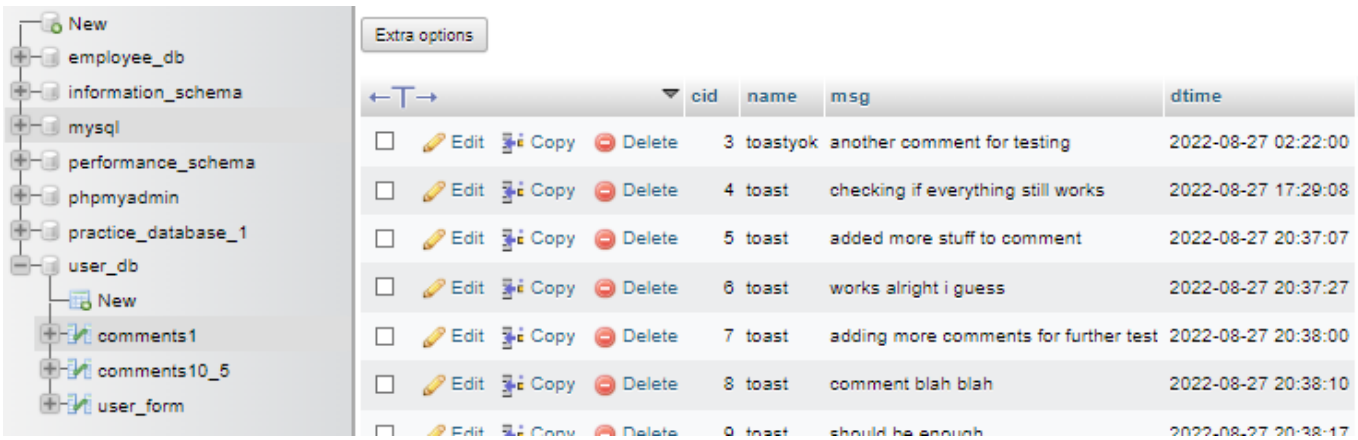
user_form table saves information from user for login. Name and Email are candidate keys. Email is set as the Primary Key. ***No Null value is allowed for any table.***



	name	email	password	user_type
Edit Copy Delete	boruto	boruto@gmail	d09b9d3cb4f7cbef107bef0425ca8eaf	user
Edit Copy Delete	brook	brook@gmail.com	0cfe0ef3a357503c4a4538414b870ca1	user
Edit Copy Delete	carrot	carrot@gmail.com	e6d96502596d7e7887b76646c5f615d9	user
Edit Copy Delete	chopper	chopper@gmail.com	c5c2df792f008f6fa3fa335c921ba65e	user
Edit Copy Delete	franky	franky@gmail.com	ffe7470430a737c4ce6dc74bea0155d5	user
Edit Copy Delete	jimmy	jimmy@gmail.com	5e027396789a18c37aeda616e3d7991b	admin
Edit Copy Delete	jinbe	jinbe@gmail.com	84fff20659999e2b83b45c6851ec57dd	user

Comment1 (basically comment table for ch1, and comments 10_5 for ch10_5. Same table just different values saved). Separate comment table for separate chapters to ensure users are not confused/spoiled by old/new chapter comments when viewing comments, as database pulls the latest comments.

Cid is the primary key and auto increments. This is very helpful when deleting comments. Name and msg together form a unique key to prevent spam comments. Dtime just shows date and time of posted comment.

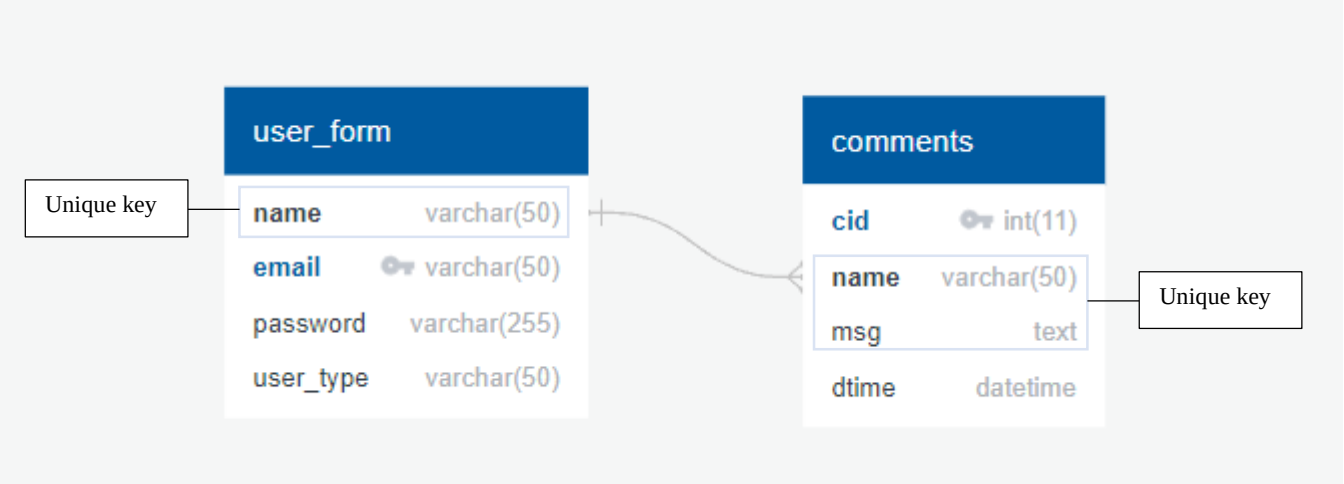


	cid	name	msg	dtime
Edit Copy Delete	3	toastyok	another comment for testing	2022-08-27 02:22:00
Edit Copy Delete	4	toast	checking if everything still works	2022-08-27 17:29:08
Edit Copy Delete	5	toast	added more stuff to comment	2022-08-27 20:37:07
Edit Copy Delete	6	toast	works alright i guess	2022-08-27 20:37:27
Edit Copy Delete	7	toast	adding more comments for further test	2022-08-27 20:38:00
Edit Copy Delete	8	toast	comment blah blah	2022-08-27 20:38:10
Edit Copy Delete	9	toast	should be enough	2022-08-27 20:38:17

Admin panel calls on some functions that utilizes (joins) both tables to give output.

More on user and admin features on the snapshot section.

ER-Diagram:



Work flow:

User:

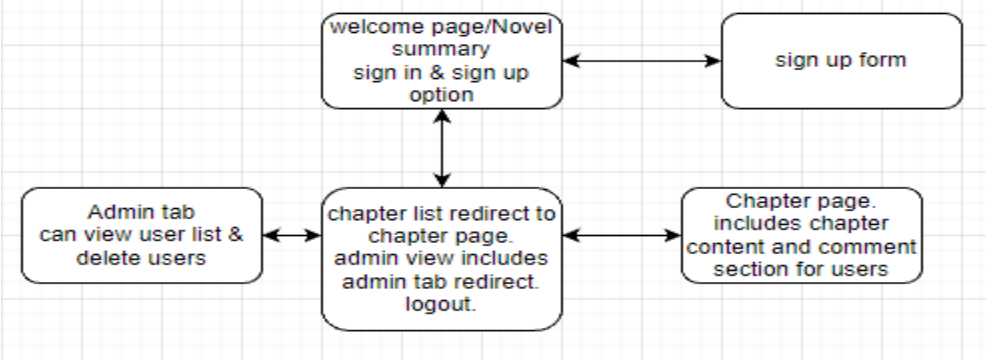
- Home page – user can see short summary of the website. User is informed to login to view content. User has access to buttons; login, register, and home (logo), that redirects them to respective locations.
- Login page – user sees login form, has to enter all details correctly and click login to be connected to the landing page, else stays on login form. User also can click register now button to be redirected to register page.
- Register page – user sees form, has to enter all info appropriately or error dialog pops and form asks for resubmission. Name of user must be unique according to database, dialog with username already exists pops for already existing name in database. If register is successful, user is redirected to Login page.
- Landing page – displays appropriate unique welcome text, button and button names of sidebar. Sidebar buttons; home (logo-returns to home page but can come to landing page with back button of browser as long as logged in), logout(ends session). Can see scrollable list of chapters with search option. Clicking on chapter name redirects to the chapter page.
- Chapter page - displays appropriate ch name, Sidebar buttons; home (logo-returns to home page but can come to landing page with back button of browser as long as logged in), logout(ends session), chapter list button (redirects to landing page). Can see chapter content in scrollable box. Comment form at end of chapter, user can enter text comment (ensure no spam). User can see 3 latest comments if any exist. On click of show more button, can view a total of latest 50 comments.

Admin:

- Everything that user can do.
- Landing page will have another button in sidebar that links to admin panel
- Comment section in chapter page will display delete comment button
- Admin panel: view list of all registered entities. Password is not displayed. Form that takes name to remove entity from database (kick), another form that takes name to promote user to admin. Another form that takes table name and username to display all count of comments grouped by username. Finally another form that takes table and username to get their last activity log.

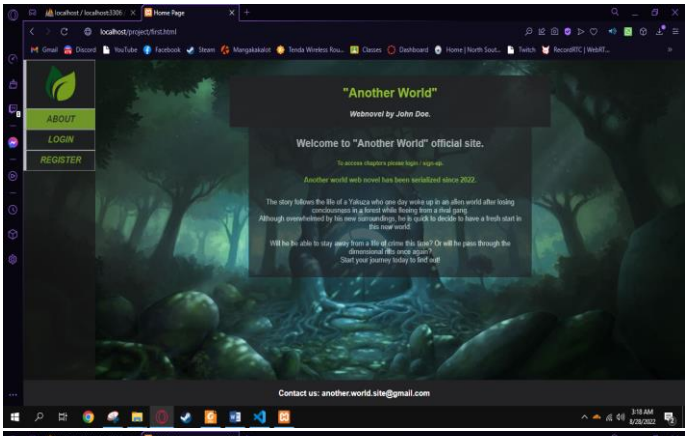
Sitemap of the Project.

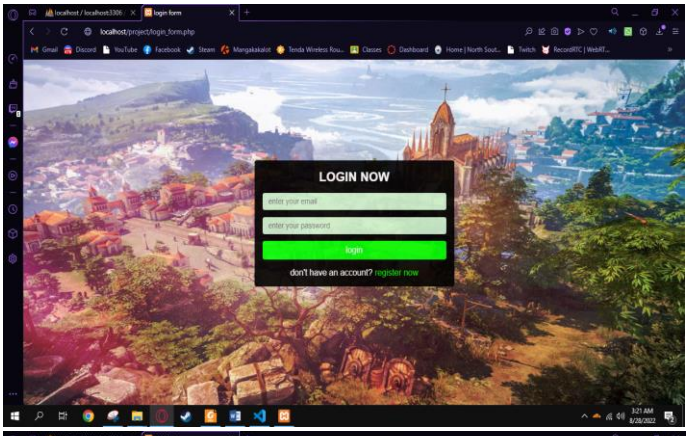
A rough idea of the sitemap. Browser back and forward buttons are utilized to move around in some cases.

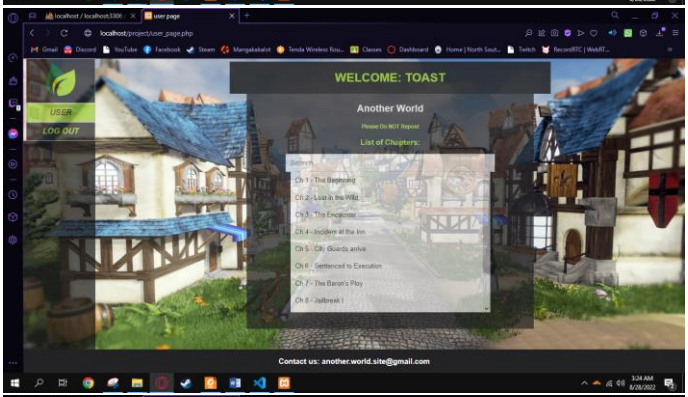


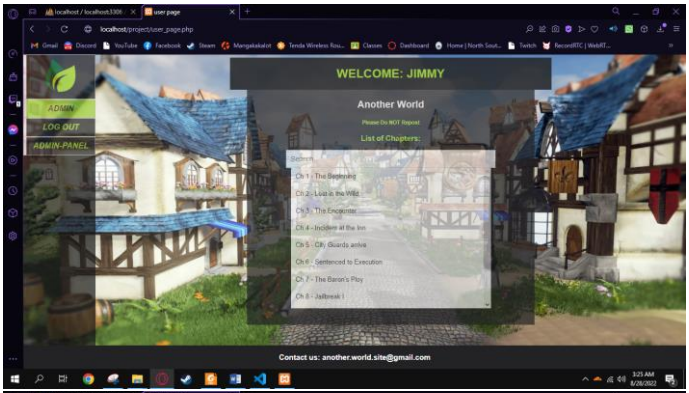
Snapshots:

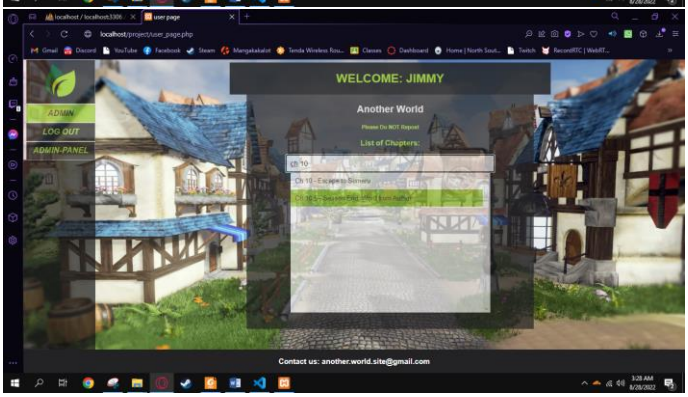
The leaf logo is clickable and will always redirect to home page.

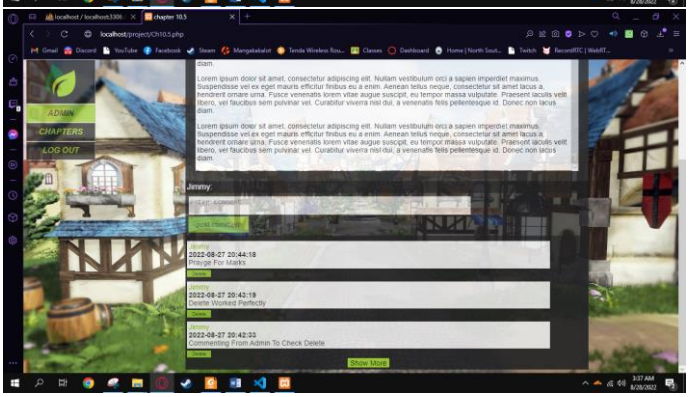


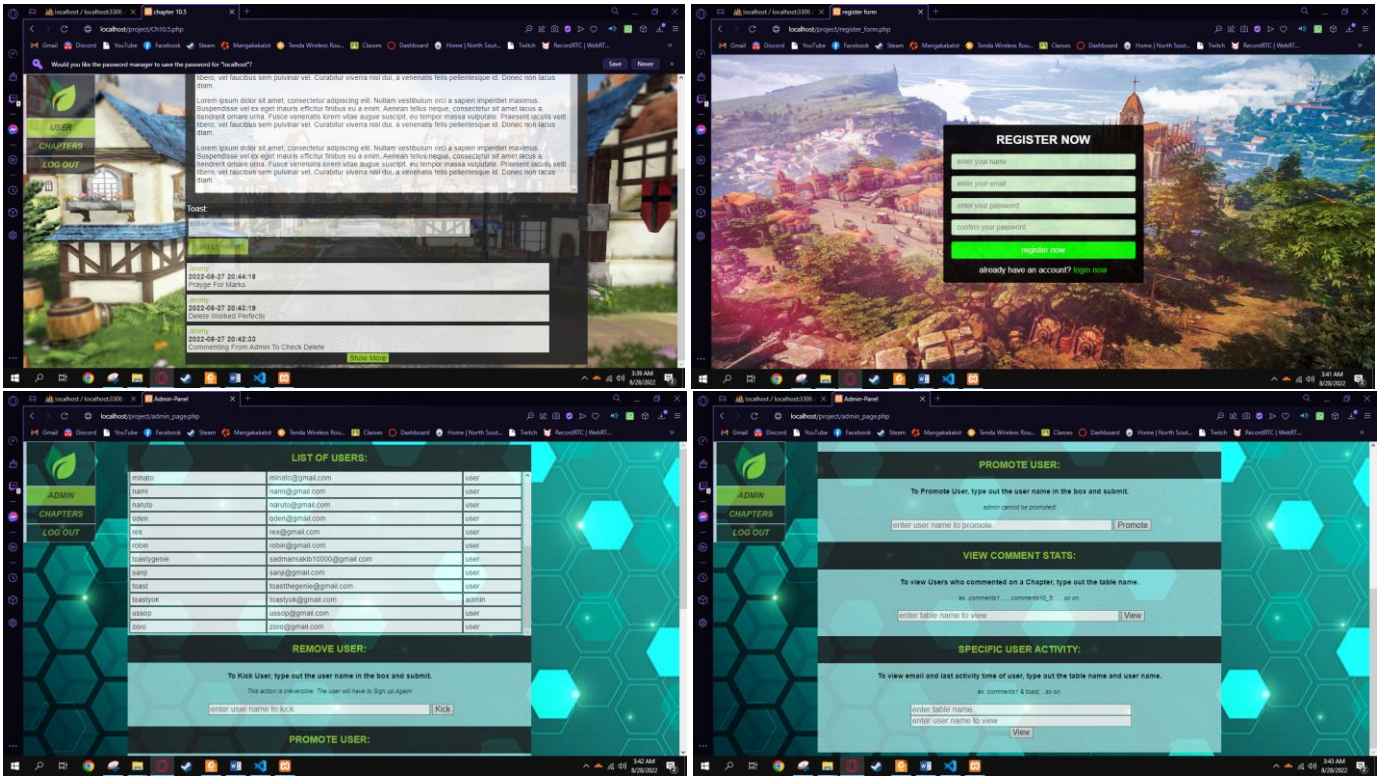












Conclusion & future work:

- Overall was a great project to work on to solidify the basics of php and mysql.
- Great motivation to further learn css.
- Not a good schema for commercial use.
- Need to improve on security.
- Need to make the website responsive, hidden side bar for mobile etc.
- Instead of dialog box popping up and refreshing page when failed to register, data could remain entered even after fail, and dialog box pops without reload. (use of Ajax.)
- The functions in admin panel don't really have much value by themselves, need to add more functions with them like send email automatically to a user after their entry has been modified, to add practical value. Right now the functions are there to meet minimum project requirements and learning purpose.
- Better work could be done on front-end if raw css wasn't used to build it.

Contributions:

Nadia Binte Rahman Peeya – 1731358642

- Worked on the login system.
- Login form backend
- Register form backend
- Config.php (database connection)

Shadman Sakib – 2014310042:

- Added clauses & conditions for error dialogs, set requirements for entry in database in the Login system.
- Setup/Modify connection of unique landing page for both user types from Login.
- Made the stylesheets, worked on the front end of the project. (Nadia did contribute in style.css)
- Worked on backend & Frontend in:
    1. Landing page (user_page.php)

2. Admin panel (admin_page.php)
3. Chapter Page (ch1.php , ch10_5.php)
4. Home page (first.html)

https://github.com/ShadmanSakib22/CSE311L_GroupProject/tree/main/Complete