

# House of Words.

*A place where you can share your love of reading with the world.*

---

## Project Final Report

Prepared by: Farah Al Wardani  
2018-2019

---



**Lebanese University-Faculty of Sciences IV**  
**I3308 - Project**  
**Supervisor: Dr Mohammad Dandache**

# Table of Contents

<b>Introduction</b>	<b>4</b>
<b>Abstract</b>	<b>6</b>
<b>Technologies used:</b>	<b>6</b>
MySQL	7
2. Apache	8
3. PHP	9
4. JavaScript	9
5. HTML	10
6. CSS	10
7. Ajax and JavaScript's jQuery Library	11
<b>House of Word's Features</b>	<b>12</b>
Welcome Page	12
Registering	12
Logging in	13
Home Page	14
The Side Navigation	14
The Updates Division	14
The Like and Comment Options	15
The Search Division	15
The Reading Challenge Widget	16
Challenge by year Page	16
Set Challenge Page	16
Single Challenge Page	17
Profile Page	17
Edit Profile Page	17
Friend's Profile Page	18
Friends Page	18
Friends' Requests Page	18
Author Page	18

Book Page	19
Book Division	19
<b>Methodology and Technical Aspects</b>	<b>21</b>
The Database	21
Tables Created to Store the needed Information	21
Connection to the database in php:	23
Preventing Sql Injections	23
Welcome Page	24
Registering	24
Logging in	25
1- Regarding the "Remember me" checkbox:	25
2- Regarding the existence of the user and the verification of the username/password combination:	26
In case the user is blocked:	27
Requirements for the mail function to work:	28
Home Page	29
The Side Navigation	29
The Updates Division	29
Loading More Updates	31
The Like and Comment Options	32
The Search Division	34
The Reading Challenge Widget	34
Challenge by year Page	35
Profile Page	35
Book Page	36
Book Division	36
<b>Perspective and Acknowledgement</b>	<b>38</b>
<b>Index</b>	<b>39</b>

# Introduction

A social networking service (also social networking site, or SNS or social media) is an online platform which people use to build social networks or social relationships with other people who share similar personal or career interests, activities, backgrounds or real-life connections.

The social network is distributed across various computer networks. The social networks are inherently computer networks, linking people, organization, and knowledge. Social networking services vary in format and the number of features. They can incorporate a range of new information and communication tools, operating on desktops, laptops, and on mobile devices such as tablet computers and smartphones. They may feature digital photo/video/sharing and "web logging" diary entries online (blogging). Defined as "websites that facilitate the building of a network of contacts in order to exchange various types of content online," social networking sites provide a space for interaction to continue beyond in person interactions. These computer mediated interactions link members of various networks and may help to both maintain and develop new social ties.

The **main types** of social networking services contain category places (such as age or occupation or religion), means to connect with friends (usually with self-description pages), and a recommendation system linked to trust. One can categorize social-network services into four types:

1. Socializing social network services used primarily for socializing with existing friends (e.g., Facebook)
2. Multimedia sharing social networks (e.g., Youtube).
3. Networking social network services used primarily for non-social interpersonal communication (e.g., LinkedIn, a career- and employment-oriented site)
4. **Social navigation** social network services used primarily for helping users to **find specific information or resources** (e.g., Books social network services)

This Project is an example of the fourth type. It is used mainly for finding information about books and interacting with other readers.

# Abstract

House of Words is a website in which a wide range of readers can join and share their reading experience with their friends and the “House of Words” community. A user can view information about books, authors, and other users. He is free to rate a book and review it, add other users as friends or remove them, follow other users, and eventually interact with their updates. It is developed using php as the server-side scripting language.

## Technologies used:

1. For creating and interacting with the project’s database tables, MySQL DBMS has been used as the Database Server.
2. Apache was used as the Web Server.
3. PHP has been used as the server side’s scripting language.
4. JavaScript has been used at the front-end.
5. HTML.
6. CSS has been used at the front end (for styling the html elements).
7. Ajax: JavaScript’s jQuery library has been used for handling some ajax requests. (With Ajax, web applications can send and retrieve data from a server asynchronously without interfering with the display and behavior of the existing page.)

# 1. *MySQL*

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

## **2. Apache**

A web server is like a restaurant host. When you arrive in a restaurant, the host greets you, checks your booking information and takes you to your table. Similar to the restaurant host, the web server checks for the web page you have requested and fetches it for your viewing pleasure. However, A web server is not just your host but also your server. Once it has found the web page you requested, it also serves you the web page. A web server like Apache, is also the Maitre D' of the restaurant. It handles your communications with the website (the kitchen), handles your requests, makes sure that other staff (modules) are ready to serve you. It is also the bus boy, as it cleans the tables (memory, cache, modules) and clears them for new customers.

So basically a web server is the software that receives your request to access a web page. It runs a few security checks on your HTTP request and takes you to the web page. Depending on the page you have requested, the page may ask the server to run a few extra modules while generating the document to serve you. It then serves you the document you requested.

Apache is the most widely used web server software. Developed and maintained by the Apache Software Foundation, Apache is an open source software available for free. It runs on 67% of all web-servers in the world. It is fast, reliable, and secure. It can be highly customized to meet the needs of many different environments by using extensions and modules.



### **3. PHP**

A scripting language is a language that interprets scripts at runtime. Scripts are usually embedded into other software environments. The purpose of the scripts is usually to enhance the performance or perform routine tasks for an application.

Server side scripts are interpreted on the server while client side scripts are interpreted by the client application.

PHP is a server side scripting language that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Preprocessor, that earlier stood for Personal Home Pages.

PHP scripts can only be interpreted on a server that has PHP installed. The client computers accessing the PHP scripts require a web browser only.

A PHP file contains PHP tags and ends with the extension ".php".

### **4. JavaScript**

JavaScript is a programming language for the web. It is supported by most web browsers including Chrome, Firefox, Safari, Internet Explorer, Edge, Opera, etc. Most mobile browsers for smart phones support JavaScript too.

It is primarily used to enhance web pages to provide for a more user friendly experience. These include dynamically updating web pages, user interface enhancements such as menus and dialog boxes, animations, 2D and 3D graphics, interactive maps, video players, and more. This mode of JavaScript usage in the web browser is also referred to as client-side javascript.

## **5. HTML**

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create.

The definition of HTML is HyperText Markup Language.

- HyperText is the method by which you move around on the web — by clicking on special text called hyperlinks which bring you to the next page. The fact that it is hyper just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking links — there is no set order to do things in.
- Markup is what HTML tags do to the text inside them. They mark it as a certain type of text (italicised text, for example).
- HTML is a Language, as it has code-words and syntax like any other language.

## **6. CSS**

CSS stands for Cascading Style Sheets with an emphasis placed on “Style.” While HTML is used to structure a web document (defining things like headlines and paragraphs, and allowing you to embed images, video, and other media), CSS comes through and specifies your document’s style—page layouts, colors, and fonts are all determined with CSS. Think of HTML as the foundation (every house has one), and CSS as the aesthetic choices (there’s a big difference between a Victorian mansion and a mid-century modern home).

## ***7. Ajax and JavaScript's jQuery Library***

**AJAX** stands for Asynchronous JavaScript and XML. It is a web development technique for creating interactive web applications. AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page without reloading the whole page.

A great example would be the Google suggestion list you see in the Google search box. When you start typing something in the Google search box, the list in that drop-down box changes. The content on the page dynamically changes without refreshing the page.

**jQuery** is a lightweight, "write less, do more" JavaScript library.

The purpose of jQuery is to make it much easier to use JavaScript on your website.

jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

# House of Word's Features

## *Welcome Page*

The first page of the website includes several options; the ability to sign in if a user has already signed up with an account in addition to the option of remembering the username and password, the ability of changing the password of a registered account, and the ability of signing up.

### Registering

- A link in the home page leads to the registering page.
- A form which has several fields for the user to input information is in the page.
- Input fields are: first name, last name, username(to login with), password(viewed as dots), country, date of birth, gender. All the fields are required.
- Email and Username are unique, a user can't register with an already registered email or username. If one of them already exists, a message is shown to the user.
- The password is encrypted (for privacy).
- The user's default profile picture is set according to the gender.
- When the button sign up is clicked, the user is registered if all the fields are written and headed to his home page.

## Logging in

- Username and password are required to sign in.
- Password appears as dots.
- If the checkbox "Remember me" is checked, the username and password are saved for signing in next time.
- If the username and password were previously saved and the remember me box is unchecked, they won't be saved anymore.
- After the "Sign in" button is pressed, the username and password are verified.
- The user is allowed to enter a wrong password up to 5 times. If more than 5 times, the user will be blocked.
- If the username or password is incorrect, a message "Wrong Combination" will be displayed to the user.
- In case the user is blocked, he will be redirected to a page to request unblocking. If he presses the "unblock" button, he will be asked to enter the code sent to his registered email.
- If the code entered is correct, the user will be unblocked. If not, a message will be displayed.
- A user can also click the link "forget password" if he wants to change his password. He will be asked to enter his email and in the same way, a code will be sent, and if it is verified, he will be able to change his password.

## ***Home Page***

Once a user has signed in, he is redirected to the home page.

This page consists of a side navigation, the reading challenge widget, a center division in which the user's friends' updates are displayed, and a search division where a user can search for users, books, authors, and genres.

### The Side Navigation

It contains:

- A link for Home page.
- A link for Profile page.
- A link for Friends page.
- A link for Friend Requests page.
- A "Sign Out" button that signs out the user.

### The Updates Division

If the user has friends, and if these friends have certain updates, these updates are displayed in addition to the updates of the followed users. An update indicates if a user has read, is reading, or wants to read a certain book.

For each update, the fields displayed are:

- The friend's profile image along with his name as a link to this user's page.
- The concerned book's image and name and a part of its description along with the link that leads to this book's page.
- A phrase indicating the changed state of the book according to the user: if a user wants to read the book, currently reading it, read it, rated it only, or rated and reviewed it, along with the rating and review if they exist.
- A see more option which displays the hidden part of a long review.
- A see more option after book's description links to the book's page.

- A like link so that the logged in user can like or unlike this update.
- A text input where a user can add a comment.
- A comments division where the comments on the update will be displayed.
- A link that displays the number of likes and which upon clicking opens a modal which displays the users who liked the update.
- The time difference displayed either in "seconds ago", "minutes ago", "hours ago", or the date if the time difference is more than one day.
- The updates are retrieved sequentially. Once the scrolling has reached the button, new (10) updates will be retrieved and displayed.

### The Like and Comment Options

- A like link for each update allows the user to like a certain update. If the user likes an update, the link text will turn into "unlike" and the user will be able to remove his like.
- A link with text indicating the number of likes and when it is clicked a modal is displayed which contains the names of the users who liked the update.
- A text field for adding a comment.
- A comments division in which the comments on a certain update are displayed.

### The Search Division

- A user could search for other users, books, genres, or for authors. The search is done according to the letters entered in an input text and according to the option selected in the choice list(users, books, genres, or authors).
- For example, if a user searches for a book, the image and name (link to the book's page) will be displayed.
- The search criteria is according to the entered search key if it is contained in a book's name (or author,genre..), description, and other information.

### The Reading Challenge Widget

A user can set a reading challenge for the current year where he sets a certain number of books that he is willing to read and the progress is measured according to the number of books he reads.

- This widget contains information about the user's reading challenge for the current year.
- If the challenge was created, the progress number of books, percentage, 3 books with images are displayed.
- If the challenge is completed, a message "Done!" is displayed.
- If the challenge is not set, a button takes the user to a page to set his challenge.
- A button "see more" which takes the user to a page where he can discover his and his friends' reading challenges by year.
- The user is allowed to edit his reading challenge.

### ***Challenge by year Page***

This page contains a dropdown list to select a year. Once a year is selected, the user's reading challenge is loaded into the page with the link "see more" to see all the books in this challenge, and edit challenge option.

Underneath, the user's friend's challenges are displayed for the current page with the option to see all the books of the selected challenge.

### ***Set Challenge Page***

This page contains a textfield and a button "set challenge" that allow the user to update or set his reading challenge for the current year. A user is allowed to set and edit challenge only for the current year.



## ***Single Challenge Page***

This page contains the progress and books of the reading challenge of the selected user and selected year.

## ***Profile Page***

This page contains the same side navigation as in the home page, two divisions: one for displaying the books of a certain shelf, and one for displaying the user's updates. It also contains a division which displays the user's profile image, name, about, gender, country, birthday, and links to display shelves.

A shelf is either the "Currently Reading" book shelf, "Read", or "Want to Read". The image and name (link to book page) of the book are displayed.

This page also contains the option for editing a profile which takes the user to the "Edit Profile" page.

## ***Edit Profile Page***

The information that could be changed are:

- First Name
- Last Name
- Username
- Gender
- Country
- Interests (to be stored in the database for future uses such as in a recommendation system)
- About
- Profile picture
- Viewing profile privacy (anyone or just friends)
- Following option

Once the button "Update Changes" is pressed, the user's profile will be updated.

## ***Friend's Profile Page***

This page is similar to the profile page. The difference is that the updates are retrieved from the database are the friend's updates and not the logged in user.

The added options to this page are:

- Adding as a friend if the user is not in the friends list.
- Cancelling the friend request if it has been already sent.
- Removing the user from friends if he is already a friend.
- Confirming the user as a friend if he had previously sent a request to the signed in user, the other option is to ignore and delete the request.
- Follow the user if not a friend and if allowed to be followed.
- Unfollow the user if already followed.

A profile could be either private or public according to what the user marked it as. If a profile is private, a non-friend cannot view the profile, if public, every user can view it.

## ***Friends Page***

This page contains a list of the user's friends, each as a link to his page.

## ***Friends' Requests Page***

This page contains a list of the signed in user's received friend requests with the option to either confirm or ignore. The process is similar to handling the requests in the friend's profile page.

## ***Author Page***

Displays information about authors along with their books.

## ***Book Page***

- This page is for displaying information about a book with the ability for a user to add it or change its state, rate and review it, along with displaying the signed in user's friends' reviews, and community reviews.
- If a user isn't signed in, he is only allowed to view the book information and community reviews. If a user is signed in, he is allowed to like, view likes, and add comments to his friends' reviews but he is only allowed to like and view likes of the community reviews (users other than friends).
- The side navigation is the same as that in the home page. On the right side, the image and name of the author of the book ( who has the role author and not translator..) and a link to the author's page are displayed. (The author's page is only for displaying information about the author from the database).
- At the center there is the book division where the information about the book and the reviews are displayed.

### Book Division

- The book's title and image are displayed along with the description with the read more option stated previously which was used for the reviews in the home page.
- Under the book's title, the contributors of the book are displayed each with his role.
- Moreover, the average rating of each book is displayed along with a small progress bar with the average rating as a value and maximum 5 (the maximum rating is 5), also the number of ratings is displayed.
- Under the book's image, a "want to read" submit button is displayed). This button is only enabled if the signed in user did not previously add the book to his shelves. If the user has previously added the book, the book's state will be displayed in the button(read..) and the button will be disabled.
- Beside the button, a dropdown list ( styled using css, refer to the code ) which contains the options of changing the state. The options that appear are only the options different than the current state. For example, if the book hasn't been added yet, all the options will be displayed, but if the book was added as read, only the "want to read" and "reading now options appear".

- Under the button, the signed in user's rating appears if it has been set (as number of stars-rating)
- Before the friends' updates are displayed. The signed in user review appears: if he has not previously added the book it is indicated, or he marked it as read and didn't add a review, etc. (all the possible cases).
- The friends' updates are displayed similarly to how they were displayed in the home page. The main difference is that only the last update for each friend is displayed.
- Also a button "view comments" which hides or displays the comments' division is added. The signed in user is allowed to like and comment on his friends' reviews.
- After friends' updates, the community updates are displayed (other than friends). The user is only allowed to like and not comment on non friends' reviews.

# Methodology and Technical Aspects

## *The Database*

### Tables Created to Store the needed Information

- 1) **Book:** Stores information about books with the number of ratings and the average rating of each book (according to the users of our website).
- 2) **Author:** Stores information about authors.
- 3) **Book-author:** A book may have more than one author with different roles.
- 4) **Role:** Stores authors roles possibilities-a book might have more than one author with different roles (author, translator..)
- 5) **Genre:** Stores names and description of different book genres (Fiction, Literature, Science...).
- 6) **Book-in-Genre:** A book might belong to more than one genre.
- 7) **Author-Genre:** An author may write in more than one genre (we used this table for displaying information about the author: in which genre he writes).
- 8) **User:** Stores information about a user which registers to the website. The username and email are unique. The attribute `incorrect_times` represents the number of attempts to login with a wrong combination of username and password. The tests are done with php. The user is blocked in certain cases. It also contains fields for view privacy and follow privacy.
- 9) **Added-book:**
  - A user adds a certain book with a certain state and possibly with a rating and a review.
  - Triggers are added on this table to modify a book's average rating and number of ratings in the table book.
  - A trigger after insert: if the book does not previously exist so and a rating is added the average rating and number of ratings will be modified.
  - A trigger after delete: if a user deletes a previously rated book from his shelves, the average rating and the number of ratings will be modified.

- A trigger after update which increments the number of ratings if a rating is added and decrements it if it is removed and only modifies the average rating if a rating previously existed and was just changed.

- Moreover, triggers are added on this table to insert new rows into the updates table. Each insertion and update to this table adds a new row in the updates table with the same information.

10) **Updates:** This is the table in which the rows are inserted by triggers on the added\_book table; after insert, delete, or update. It is used for historization purposes; whenever a user changes a state of a book(read or currently reading or want to read). The update is stored in this table.

11) **Likes:** A user can like an update at a certain time.

12) **Comments:** A user can comment on an update at a certain time.

13) **Friend-relation:** A user can have more than one friend.

14) **Friend-request:** A user can send a friend request to another user at a certain time.

We use this table in order for a user to view his received request so he can either confirm or ignore the request.

15) **Country:** Stores information about countries.

16) **Follow-records:** A user can follow other users.

17) **Interests:** Stores user interests to be used later in recommendation systems.

18) **User\_interest:** Links interests to users

19) **Reading\_challenge:** A challenge is unique for a certain user and a certain book.

Refer to the file houseofwords.sql for full details about the steps of the modification of the database. We used the phpMyAdmin web application in order to perform the modifications on the database

## **Connection to the database in php:**

The connection to the database is simply done in a separate file "db\_connection.php" using the php `mysqli_connect()` function which takes as parameters the server's hostname which is in our case "localhost" since the server is currently local, the username, and password. "root" is used as a username. Then the `mysqli_select_db()` function which takes the connection and the concerned database name as a parameter is used.

This file is included in each page of the website that needs to perform modifications in the database.

## **Preventing Sql Injections**

SQL injection is a code injection technique that might destroy your database.

SQL injection is one of the most common web hacking techniques.

SQL injection is the placement of malicious code in SQL statements, via web page input.

We use `mysqli` connection in order to be able to bind parameters in case of user input in order to prevent **sql injections**. We use it in case of login, register, edit profile, etc.

## ***Welcome Page***

### Registering

The first page contains a link (New? Sign up here) which leads to the page "register.php".

At the beginning, a session is started using `session_start()` and a test is done on the session variable 'username' in order to prevent a logged in user from directly accessing this page.

```
if(isset($_SESSION['username']))  
    header("location:page1.php");
```

A form which has several fields for the user to input information and of method post is contained in this page. The method POST is used instead of GET for the fields not to be shown in the url; more secure.

The textfields: first name, last name, email, username, and password(type attribute equals to "password" and not text in order for the characters not to be shown). All of these fields are marked as required by using the html attribute `required="true"`, which means that the form cannot be submitted unless these fields are filled.

The form also contains 2 choice lists, country for which the options are retrieved from the database(value is the country code and the text is the country name), and gender.

The retrieval of the values of the country list is simply done using `mysql_query` function which takes a query string as a parameter, and then each row of the output is fetched in a loop using `mysql_fetch_assoc()` function.

After submitting the form using the "Sign up" button which is an html input of type "submit" and name "register", a testing is done in the page(the action of the form is the register page itself).



It has been mentioned before that the email and username of a user are unique, so no 2 users can sign up with the same username or the same email.

The entered username and email are compared to the ones existing in the database and if they don't exist in the database, the user information are inserted into the database and thus the user is registered and then logged in and redirected to his home page. If one of them already exists, a message is shown to the user.

Before the insertion, the password is encrypted using the `password_hash()` function which takes the password and the algorithm (we used default) as parameters, also the gender is tested to set the default profile picture of the user. Refer to the code in the index (1).

## Logging in

Once the form is submitted, several tests are done:

### **1- Regarding the "Remember me" checkbox:**

This checkbox is used for the username and password to be still written in the fields when a user wants to sign in the next time. This process is done using cookies.

According to Wikipedia, "An HTTP cookie is a small piece of data sent from a website and stored on the user's computer by the user's web browser while the user is browsing. Cookies were designed to be a reliable mechanism for websites to remember stateful information or to record the user's browsing activity."

So, if `$_POST['rememberme']` is set (the checkbox is checked), 2 cookies are created, one for the username and one for the password. These 2 cookies are set for a month from the time they are set in.

On the other hand, if the checkbox is not set and the cookies are previously set, the 2 existing cookies are destroyed by setting 2 cookies by the same name but for only one second. Refer to the code in the index (2).

## **2- Regarding the existence of the user and the verification of the username/password combination:**

A select query is performed from the user table where the username equals to the submitted username. If the user exists in the database, the entered password is tested. Here a new condition appears. It has been mentioned before that an attribute "incorrect\_times" which represents the number of failed attempts to login exists in the user table of the database. The user could only sign in if the incorrect\_times attribute is less than 5.

So, if the incorrect times is less than 5, the password is tested using the password\_verify() function since we have previously encrypted the password using the password\_hash() function. If the password is verified, the SESSION array value of key 'combination' is set to true (to be used later), the value of key 'userid' is set to the user id found (to be used later in the browsing process), the incorrect\_times attribute is reset to 0(updated in the database), and finally the user is redirected to his home page.

If the username doesn't exist or the password is wrong, \$\_SESSION['combination'] is set to false. The incorrect\_times attribute is incremented by 1 if the password is wrong.

If the incorrect\_times is more than or equals 5, the user is considered blocked and the email is saved in the \$\_SESSION array and then the user is redirected to the "contactus.php" page. Refer to the code in the index (3).

If \$\_SESSION['combination'] is set to false, a message is displayed to the user: "Wrong username/password combination".

The username and password fields have initial values if the cookies are set or if the form is submitted(wrong combination→ the user is not redirected out of the page). Refer to the code in the index (4).

### **In case the user is blocked:**

In case the number of incorrect times is greater than 5, the user is redirected to the page "contactus.php". This page contains a submit button "unblock" which submits the form to a page "verifycontactus.php".

In this page, if the button "unblock" in the previous page was clicked, a random code is generated and saved in the `$_SESSION` array. This code is sent to the blocked email previously saved in `$_SESSION` and then there's a text input for the code and an "unblock" button which submits the form to the page "mail.php".

In the page "mail.php", the submitted code is compared to the code previously saved in the `$_SESSION`, if they are equal, the user is unblocked by resetting the `incorrect_times` attribute to 0.

If the code is verified, the user is redirected to a page "loading.php" where a message informing him that his account was unblocked is displayed and a link to sign in again. If the code is not verified, a message is displayed to the user on the page "verifycontactus.php", and he is able to reenter the code. Refer to the code in the index(5).

In the same way, a user can click on the "forget password" link in the login page which redirects him to the page "forgetpassword.php" where he can enter his email and submit (to the page mail.php). An email with a code will be sent to the email he entered and then he could enter the new password with the received code. If the code is verified, the password will be encrypted and updated in the database. Otherwise, a message will be displayed that the code is incorrect. The user will be able to enter the code again.

### **Requirements for the mail function to work:**

For the mail functions to be available, PHP requires an installed and working email system. The program to be used is defined by the configuration settings in the php.ini file.

The mail functions are part of the PHP core. There is no installation needed to use these functions.

### **The configurations done:**

In the file "sendmail.ini":

- 1- We set the smtp\_server to smtp.gmail.com. (we applied our mail function to gmail accounts).
- 2- We set the smtp\_port to 587 (we can choose a different number).
- 3- We set the auth\_username to the email we are sending from. In our case we are sending from the email websitesender7@gmail.com
- 4- We set the auth\_password to the password of the email.
- 5- We set force\_sender to the same email.

In the file "php.ini":

We set after [mail function]:

- 1- smtp = smtp.gmail.com
- 2- sendmail\_path = "C:\sendmail\sendmail.exe-t".

We finally restart the apache server and we lower the security of the gmail sender account.

## ***Home Page***

Once a user has signed in, he is redirected to the home page.

This page consists of a side navigation, the reading challenge widget, a center division in which the user's friends' updates are displayed, and a search division where a user can search for users, books, authors, and genres.

### The Side Navigation

It contains a submit button "sign out" which logs out the user by setting `$_SESSION['signedout']` to true, then the user is redirected to the "login.php" where this variable is tested. If this variable is set, the session is destroyed and a message "You are signed out" is displayed.

The session was destroyed using `$_SESSION=array()` and not `session_destroy()`; we initialized the `$_SESSION` array because sometimes the server failed to destroy the session because it had not already been created (it did not have enough time).

### The Updates Division

First, the php function `update_friends()` that has been defined in the file "functions.php" and which updates the `$_SESSION['friends']`: array of friends ids, is called. Refer to the code in the index(7).

A string `ids` which is a concatenation of the user's friends' ids separated by a comma is retrieved by the function `get_friends_ids()`. Refer to the code in index (7). This string is used in the query that gets the friends updates.

```
$get_friends_updates_query = "select *,(NOW()-timestamp) as timediff from
updates,user,book where updates.book_id=book.book_id and
```

updates.user\_id=user.user\_id and updates.user\_id in (".\$ids.") order by updates.timestamp desc";

The information about the update, the user which have done the update, the book which the update was made on and the time difference between the current time and the update's time are retrieved.

The results of this query are fetched row by row so each update can be displayed. So:

First, we set 3 counters(for elements' id naming), one for the stars display elements(rating), one for the hidden review elements(if the review is long, we don't fully display it, and one for the like and comment divisions and links).

1- The profile image is an html image with source the image's name in the database in the folder user\_profile\_pics. The link of the user page is "friend\_profile.php" with the user's id passed in the url to be retrieved from the \$\_GET array.

2- The book's image is of path book\_pics/ the name of the image in the database. The books link is "book.php" with the book's id passed in the url to be retrieved from the \$\_GET array.

3- The attribute 'changed\_state' is tested:

If it equals to 0 (read), the rating is tested: if the book is rated(rating not null), the phrase is "has rated a book" along with an html span of id "stars" concatenated with the first counter's value where the number of stars(rating) is displayed using the javascript function view\_stars() which takes as parameters the rating and the counter's value. The counter is incremented. Refer to the code in index (8) to view the javascript function. If the book was not rated, the phrase is "has read a book". The review then is tested, if it is not null, the phrase is "has reviewed a book" and then the review should be displayed. If the review's length is greater than 100, the review is separated into 2 parts, visible and hidden, and a read more link is displayed (if there is a hidden part) which calls the javascript function view() which takes the hidden review's span's id and the read more link's id(according to the counter) which displays or hides the hidden part and changes the text of the link

into (less) or (read more). Then the corresponding counter is incremented. Refer to the index(8) to view the javascript function.

If the changed state is 1, the phrase is "currently reading a book", and if it is 2, the phrase is "wants to read a book".

8- After calculating the time difference in the query, the time difference display is done according to several cases:

If the time difference is less than 60 seconds, the display is the number of seconds + "seconds ago".

Otherwise if the time difference is less than 60 minutes, the display is the number of minutes + "minutes ago".

Otherwise if the time difference is less than 24 hours, the display is the number of hours + "hrs" + the number of minutes + "min ago".

Otherwise, only the date is displayed (the substring of the user table date-time timestamp attribute from 0 up to 10 characters ex: "in 2019-02-19").

Refer to the code in the index (9).

### Loading More Updates

The updates division has been modified to view updates sequentially from the database for the purpose of making the request lighter.

At each scroll end, a new division is generated and an ajax request is made to load new 10 updates into the new division. \$\_SESSION is used to store the counter values and a hidden html element is used to reload at each time the updates division counter value also by jquery's load.

The function is called if the interval between scrolls is more than 3 seconds in order to properly synchronize the requests.

## The Like and Comment Options

The jQuery library of javascript has been used for the like to be inserted in the database's likes table and for the comment to be inserted and to be displayed. Specifically we used the jQuery's load() method:

According to w3schools, The jQuery load() method is a simple, but powerful AJAX method.

The load() method loads data from a server and puts the returned data into the selected element.

The syntax:

```
$(selector).load(URL,data,callback);
```

The required URL parameter specifies the URL you wish to load.

The optional data parameter specifies a set of querystring key/value pairs to send along with the request.

The selector is the id of the html element we want to load data in.

The optional callback parameter is the name of a function to be executed after the load() method is completed.

This Library can be simply downloaded from <https://jquery.com/download/> as a javascript file and included using the line "<script src='jquery-3.3.1.min.js'></script>".



So in the code:

Each element(like link, nb of likes link, comments text input..) has been given an id according to the its type and to the value of the third counter. The number of likes is retrieved by a MySQL query and the existence of the like by the signed in user is tested (also from the database) to modify the text of the like link(like or unlike). The click on the like link calls the javascript function `clicked_like()` which takes as parameters: logged in user's id, update id, the like link's id, the number of likes link's id. This function uses the `load()` method where the url is "processlikes.php" which takes the user's id and the update's id as parameters and the id of the span we want to load the content into is the number of likes link's id since the number of likes will change, and where the callback function alternates the like link's text. Refer to the code in index (9).

In "processlikes.php", the existence of the like is tested, if it exists it will be removed, if it doesn't it will be inserted into the database. Then the number of likes is retrieved and printed to be displayed in the link given in the parameter of the javascript `clicked_like()` function.

Refer to the code in index (10).

The click on the nb of likes link calls the `display_likes()` function which takes the update id as parameter. This function displays the modal and retrieves the liking users asynchronously from the database also using the `load()` method. Refer to the code in index (10).

Inserting a comment and displaying the comment is done using jQuery's `load` method as well. The comment is inserted when the "Enter" key is pressed and the callback function of the `load()` method empties the text field. The comments are displayed using the php "display\_comments" function. Refer to index(8), (9) and (10) for details on these 2 functionalities. Function `display_comments()` is in index (7).

The updates are displayed if the user has friends and for each update the third counter(for id naming) is incremented. A message indicating the lack of friends is displayed if the user does not have any friends. If a user is not signed in, he is redirected to the login page.

### The Search Division

A user could search for other users, books, or for authors. The search is done according to the letters entered in an input text and according to the option selected in the choice list(users, books, or authors). oninput event calls the display\_search() function. This function passes the value of the key and the value of the choice list as parameters in the url to be retrieved in \$\_GET and the results are loaded in the search result division by using the load() method. This url is "search.php" in addition to the key and the value of the choice list.

In "search.php", the search choice is tested and the search is done in the corresponding table. For example, if the user is searching for a book, all the books are selected from the book table and for each row the book's name and description are concatenated and the occurrence of the key in this string is tested using POSIX regular expressions by using the function eregi(). If the key exists, the image and the name of the book are displayed in the division which is loaded into the search result division of the home page. Refer to indexes (8), (11), and (12) for detailed code and steps.

### The Reading Challenge Widget

A user can set a reading challenge for the current year where he sets a certain number of books that he is willing to read and the progress is measured according to the number of books he reads.

- This widget contains information about the user's reading challenge for the current year.
- If the challenge was created, the progress number of books, percentage, 3 books with images are displayed.
- If the challenge is completed, a message "Done!" is displayed.
- If the challenge is not set, a button takes the user to a page to set his challenge.
- A button "see more" which takes the user to a page where he can discover his and his friends' reading challenges by year.
- The user is allowed to edit his reading challenge.

## ***Challenge by year Page***

The widgets are loaded using jquery's load function and this function is triggered when the year selection is changed.

## ***Profile Page***

At the loading of the page, the want to read shelf is displayed. Once a link is clicked(read, currently reading, or want to read), the corresponding shelf is displayed in the shelf division using the function display\_shelf() which takes the corresponding state (0,1, or 2) and the user's id as parameters.

This function uses the jQuery's load() method to asynchronously display books in the profile's shelf division.

```
function display_shelf(state, userId){
    var url = "displaying_shelf.php?state=" + state + "&user_id=" + userId + " #shelf";
    $(document).ready(function(){
        $("#shelf_division").load(url, function(){});
    });
    return false;
}
```

In "displaying\_shelf.php", the \$\_GET values are tested and the books are retrieved from the database according to the state and user and the image and name are displayed. Refer to the index (13).

## ***Book Page***

The book to be displayed in this page is identified by the book id passed in the url that leads to a certain book.

### Book Division

At the beginning, the information about the book is retrieved from the database according to the book's id in \$\_GET. Then the book's state according to the signed in user is also retrieved from the database's added\_book table.

Under the "Want to read", the signed in user's rating appears if it has been set (the image of stars is viewed by the view\_stars() function, refer to index(8)).

### **The query for retrieving the friends' last updates:**

```
$get_friends_reviews_query = "select * from user a,updates b where  
a.user_id=b.user_id and b.user_id in (".$ids.") and b.book_id=".$book_id." and  
b.timestamp in (select max(timestamp) from updates where user_id=b.user_id and  
book_id=b.book_id group by user_id,book_id)";
```

Ids is the string returned by get\_friends-ids() function . Refer to index(7).

After friends' updates, the community updates are displayed(other than friends).  
The user is only allowed to like and not comment on non friends' reviews.

### **The query for retrieving the community's last updates:**

```
$get_community_reviews_query = "select * from user a,updates b where  
a.user_id=b.user_id and b.book_id=".$book_id." and b.timestamp in (select  
max(timestamp) from updates where user_id=b.user_id and book_id=b.book_id  
group by user_id,book_id)";
```

### **-Changing the book's state when clicking "want to read button" or one of the dropdown list links:**

The `$_GET['state']` is tested if set. Then we check if the user previously added the book by retrieving the information from the database. A flag in `$_SESSION` is set to true if a row with this book id and this user id exists in the `added_book` table, if not the flag is set to false. If the state in `$_GET` is not 0 (not read state) and the flag is set to false, a new row is inserted with the necessary information, if the flag is set to true, the old information is updated.

If the new state is 0 (read state), the user is redirected to the page "rate.php" where he can set a rating (using a list and the number of stars will be displayed on change event of the value of the list), write a review and submit the changes and the changes will similarly be either inserted or updated.

Refer to the code of "book.php" in index(15) and the code of "rate.php" in index (16).

# Perspective and Acknowledgement

House of Words is made for a person to share his reading experience with his friends and the house of words community and to be able to benefit from other users' experiences and to interact with their opinions.

This project is just a beginning for there still are many options to be added in the future. For example, there's the option of creating new shelves and adding books to them, the option for a user to be a Librarian and add more books to the database, a recommendation system, a messaging system, and a lot more.

Special thanks to our dear supervisors Dr. Mohammad Dandache, Dr. Oussama Zein, Dr. Bilal Ghazal, Dr. Youssef Atat, Dr. Charbel Rahal, Dr. Hussein Joumaa for their continuous support, precious time, and insightful ideas.

# Index

```
(1) if(isset($_POST['register'])){
    $user_exists = false;
    $email_exists = false;
    $search_user_query = "select user_name,email from user where
user_name='".$_POST['username']."' or email='".$_POST['email']."'";
    $user_output = mysql_query($search_user_query);
    while($line = mysql_fetch_assoc($user_output)){
        if($line['user_name'] == $_POST['username']){
            $user_exists = true;
            break;
        }
        else if($line['email'] == $_POST['email']){
            $email_exists = true;
            break;
        }
    }

    if(!$user_exists && !$email_exists) {
        if($_POST['gender'] == "M") $image = 'male.png';
        else if($_POST['gender'] == "F") $image = 'female.jpg';
        $password_hash = password_hash($_POST['password'],
PASSWORD_DEFAULT);
        $register_query = "INSERT INTO `user` (`user_id`,`user_name`,
`password`,`email`,`first_name`,`last_name`,`gender`,`country_code`,
`about_me`,`image`,`incorrect_times`) VALUES
(NULL,'".$_POST['username']."','".$password_hash."','".$_POST['email']."','".$_POST['fname']."','".$_POST['lname']."','".$_POST['gender']."','".$_POST['country']."' ,NU
LL,'".$image."',0);";

        if(mysql_query($register_query)){
            $_SESSION['username'] = $_POST['username'];
            $_SESSION['userid'] = mysql_insert_id();
            header("location:page1.php");
```

```

    }
    else echo "error has occurred ";
}
// if(mysql_erro() == 1062){ //error for duplicate username
else if($user_exists)
    echo "<h4 style='color:white'>This Username already exists.
Choose another one.</h4>";
else if($email_exists)
    echo "<h4 style='color:white'>This email is already used. Choose
another one</h4>";
}

```

```

(2) if(isset($_POST['login']) && isset($_POST['rememberme'])) {
    $time = time() + 3600 * 24 * 30;
    setcookie('username', $_POST['username'], $time);
    setcookie('password', $_POST['password'], $time);
}
else if(isset($_POST['login']) && isset($_COOKIE['username'])) {
    setcookie('username', $_POST['username'], 1);
    setcookie('password', $_POST['password'], 1);
}

```

```

(3) if(isset($_POST['login'])) {
    $login_query = "select * from user where
user_name='".$_POST['username']."'";
    $output = mysql_query($login_query);
    if(mysql_num_rows($output) == 1){
        //$_SESSION['user_found'] = true;
        $line = mysql_fetch_assoc($output);
        $_SESSION['image'] = $line['image'];
        if($line['incorrect_times'] < 5){
            if(password_verify($_POST['password'],
$line['password'])) {
                $_SESSION['combination'] = true;

```



```

        $_SESSION['username'] = $_POST['username'];
        $_SESSION['userid'] = $line['user_id'];
        $correctpassword_query = "update user set
incorrect_times=0 where user_name='".$_POST['username']."'";
        mysql_query($correctpassword_query);
        header("location:page1.php");
    }
    else {
        $_SESSION['combination'] = false;
        $wrongpassword_query = "update user set
incorrect_times=incorrect_times+1 where user_name='".$_POST['username']."'";
        mysql_query($wrongpassword_query);
    }
}
else {
    $_SESSION['blocked_email'] = $line['email'];
    //unset($_SESSION['password_correct']); //?????
    header("location:contactus.php");
}
}
else $_SESSION['combination'] = false;
}

```

```

(4) <input type="text" name="username" style="background-color:#e7d8cb"
placeholder="Enter Username" required='true' <?php
if(isset($_POST['username']))){ echo "value='".$_POST['username']."' ";}
if(isset($_COOKIE['username'])) echo "value='".$_COOKIE['username']."' "; ?>>

```

```

<input type="Password" name="password" style="background-color:#e7d8cb"
placeholder="Enter Password" required='true' <?php if(isset($_POST['password']))){
echo "value='".$_POST['password']."'";} if(isset($_COOKIE['password'])) echo
"value='".$_COOKIE['password']."' "; ?>>

```

(5) In "verifycontactus.php"

```
session_start();
if(isset($_SESSION['code_verified']) && !$_SESSION['code_verified']) echo
"<h3 style='color:white'>code is not verified</h3>";
if(isset($_POST['unblock']))){
    $code= floor(rand());
    $_SESSION['code']= $code;
    if(mail($_SESSION['blocked_email'],'Forget your password ?','kindly
use the code '.$code.' to change your account password.' , 'From:
websitesender7@gmail.com')){
        echo "<h3 style='color:white'>A code was sent to your email
".$_SESSION['blocked_email']."\n please enter the correct code to proceed.</h3>";
    }
    else echo 'email not sending ,check ur internet connection' ;
}
else if(!isset($_SESSION['code_verified'])) header("location:login.php");
```

In "mail.php"

```
if(isset($_POST['unblock']) && isset($_POST['code']) && $_POST['code'] ==
$_SESSION['code']){
    $unblock_user_query="update user set incorrect_times = 0 where
email = '".$_SESSION['blocked_email']."'";
    mysql_query($unblock_user_query);
    $_SESSION['code_verified'] = true;
    header("location:loading.php");
}
```

```

else if(isset($_POST['unblock']) && isset($_POST['code']) && $_POST['code'] !=
$_SESSION['code'] ){
    $_SESSION['code_verified']=false;
    header("location:verifycontactus.php");
}

```

In "loading.php"

```

<?php
if(isset($_SESSION['code_verified'])){
    echo "<h3 style='color:white'>You're account is not blocked anymore</h3>";
}
session_destroy();
?>
<p><a href='login.php'>Sign in again</a> </p>

```

(6)

(7) "functions.php"

```

<?php

function update_friends(){
    $friends = array();
    $friends_query = "select friend_user_id from friend_relation where
user_id='".$_SESSION['userid']."'";
    $output = mysql_query($friends_query);
    while ($line=mysql_fetch_assoc($output)) {
        $friends[] = $line['friend_user_id'];
    }
    if(isset($friends)) $_SESSION['friends'] = $friends;
}

```

```

}

function get_friends_ids(){
    $ids="";
    if(isset($_SESSION['friends'])){
        $friends = $_SESSION['friends'];
        if(sizeof($friends)>0){
            for($i=0;$i<sizeof($friends)-1;$i++) {
                $ids .= "$friends[$i]",";
            }
            $ids .= "$friends[$i]";
        }
    }
    return $ids;
}

```

```

function display_comments($update_id,$db){
    $select_comments_query = "select * from comments,user where
user.user_id=comments.user_id and comments.update_id=$update_id";
    $result = mysql_query($select_comments_query);
    echo "<table border='0' style='word-wrap: break-word; table-layout:fixed'
width='500px'>";
    while($line = mysql_fetch_assoc($result)){
        $image = $line['image'];
        $first_name = $line['first_name'];
        $last_name = $line['last_name'];
        $user_id = $line['user_id'];
        $comment_txt = str_replace('\\', ", ", $line['comment_text']);

        //echo "<span id='numlikes'>";
        echo "<tr><td width='50px' height='auto' valign='top'><img
src='user_profile_pics/$image' width='50px' height='50px'></td>";
        echo "<td style='font-weight:bold; color:white;' width='400px'>";
        $href = "friend_profile.php?userid=$user_id";
    }
}

```

```

    echo "<a href='\".$href.\"'>$first_name&nbsp;&nbsp;$last_name</a>";
    echo "<br><span style='color:black; font-weight:italic;
width:500px;'>$comment_txt</span></td></tr>";
}
echo "</table>";
}

?>

```

(8) "functions\_jscript.js":

```

function insertcomment(event, element, userId, updateId){
    if(event.keyCode==13){ //enter key
        var id = element.id.substr(13);
        id = "#comments_div" + id;
        //the space in the cmnt made a problem
        var url = "processlikes.php?insertcomment=1&user_id=" + userId +
"&comment_txt=" + encodeURIComponent(element.value) + "&update_id=" + updateId + "
#comments_div";
        $(document).ready(function(){
            $(id).load(url,function(){
                element.value="";
            });
        })
    }
}

```

```

function display_search(){
    var key = document.getElementById('search_form').search_key.value; //space
is causing a problem
    // key = key.replace(/\\s/g, "");
    //alert(key);
    var filter = document.getElementById('search_form').filter.value;
    var url = "search.php?key=" + encodeURIComponent(key) + "&filter="+filter+ "
#search_results_div";

```

```

//alert(url);
$(document).ready(function(){
    $("#search_result").load(url,function(){

    });
})
}

```

```

function view_stars(myRating,eltId){
    var theImage = "";
    for(var i=0; i<myRating; i++)
        theImage += "<img src='star.png' width='10px' height='10px; '>";
    var id="stars"+eltId;
    document.getElementById(id).innerHTML = theImage;
}

```

```

function view(readlink,desc) {
    if(document.getElementById(readlink).innerHTML == "..read more") {
        document.getElementById(desc).hidden = false;
        document.getElementById(readlink).innerHTML = " less";
    }
    else {
        document.getElementById(desc).hidden = true;
        document.getElementById(readlink).innerHTML = "..read more";
    }
    return false;
}

```

```

function clicked_like(user_id,update_id,like_link_id,nb_of_likes_id){
    $(document).ready(function(){
        var url = "processlikes.php?update_id=" + update_id + "&user_id=" + user_id + "
#" + "numlikes";
        $("#" + nb_of_likes_id).load(url,function(){
            if(document.getElementById(like_link_id).innerHTML == "Like")
                document.getElementById(like_link_id).innerHTML = "Unlike";

```

```

        else
            document.getElementById(like_link_id).innerHTML = "Like";
    });
});
return false;
}

function display_likes(updateId){
    $(document).ready(function(){
        var url = "processlikes.php?update_id=" + updateId + " #" + "likes_display";
        $("#likers_modal").load(url,function(){
            // Get the modal
            var modal = document.getElementById('myModal');
            // Get the <span> element that closes the modal
            var span = document.getElementsByClassName("close")[0];
            modal.style.display = "block";
            // When the user clicks on <span> (x), close the modal
            span.onclick = function() {
                modal.style.display = "none";
            }
            // When the user clicks anywhere outside of the modal, close it
            window.onclick = function(event) {
                if (event.target == modal) {
                    modal.style.display = "none";
                }
            }
        });
    });
    return false;
}

function hide_view(eltId){
    var x = document.getElementById(eltId);
    x.hidden = !x.hidden;
}

```

```

function display_shelf(state, userId){
    var url = "displaying_shelf.php?state=" + state + "&user_id=" + userId + " #shelf";
    //alert(url);
    $(document).ready(function(){
        $("#shelf_division").load(url, function(){});
    });
    return false;
}

```

(9) in "page1.php"

```

$friends_updates_output = mysql_query($get_friends_updates_query);
    $i=0; //variable for the stars image elements
    $j=0; //variable for the review(hidden) elements
    $k=0; //variable for like and comment(id for each update)
    while($line=mysql_fetch_assoc($friends_updates_output)){
        $like_link =
"processlikes.php?user_id=".$user_id."&update_id=".$line['update_id'];
        $update_id = $line['update_id'];

        echo "<table border='0' width='550px'><tr><td width='55px'
valign='top'><img src='user_profile_pics/".$line['image']."' style='object-fit: cover;
width:50px; height:50px;'></td>";
        $profile_link = "friend_profile.php?userid=".$line['user_id'];
        $book_link = "book.php?book_id=".$line['book_id'];
        if($line['changed_state']==0){ //read state
            echo "<td valign='center'><a href='".$profile_link.'"
style='text-decoration: none; color: white; font-weight: bold'>".$line['first_name'].
".$line['last_name'].</a>";
            if($line['rating']!=null) {
                echo " has rated a book. <span id='stars".$i."'></span>"; ?>

```



```

        <script type="text/javascript">view_stars(<?php
if(isset($line['rating'])) echo $line['rating']; else echo "0"; echo ",$i"; ?>);</script>
        <?php
            $i++;
        }
    else {
        echo " has read a book.";
    }

    //review -->read more.
    if($line['review']!=null) {
        echo "<br>";
        $review = $line['review'];
        if(strlen($review)>100){
            $review_view = substr($review, 0 ,100);
            $review_hidden = substr($review, 100);
        }
        else $review_view = $review;
        echo "<span>".$review_view;
        if(isset($review_hidden))
            echo "</span><span id='review".$j.'"
hidden='true'>".$review_hidden."</span>".<a href='asd' id='readlink".$j.'"
onclick='return view(\"readlink".$j.\"\", \"review".$j.\"\")' style='color: #FFFF00
'>..read more</a>";
            $j++;
            unset($review_hidden); //for next hidden reviewtext
        }

    }
    else{ //if state not read
        echo "<td valign='center'><a href='".$profile_link.'"
style='text-decoration: none; color: white; font-weight: bold'>".$line['first_name'].
".$line['last_name']."</a>";
        if($line['changed_state']==1){
            echo " is reading a book.";
        }
    }
}

```

```

}
else if($line['changed_state']==2){
    echo " wants to read a book.";
}
}
$timediff = $line['timediff'];
if($timediff<60)
    $timediff = floor($timediff)."sec ago";
else if($timediff/60<60 )
    $timediff = floor($timediff/60)."min ago";
else if($timediff/3600<24){
    $timediffmin = floor(($timediff%3600)/60);
    $timediff = floor($timediff/3600)."hr, ".$timediffmin."min ago";
}
else $timediff = "in ".substr($line['timestamp'], 0,10);

echo "</td><td align='center' valign='center' width='140px'
style='font-family: courier; font-size: 13px'>".$timediff."</td></tr>";
echo "<tr><td width='55px'></td>";
echo "<td width='55px'><table border='0'><tr><td rowspan='2'
valign='top'><img src='book_pics/'."$line['book_image'].'" style='object-fit: cover;
width:75px; height:auto;'></td>";
echo "<td valign='top'><a
href=$book_link>".$line['book_name']."</a></td></tr>";
//description-->read more

$description = $line['description'];

$description_view = substr($description, 0 ,100);
echo "<tr><td style='padding: 3%; ' valign='top'
><span>".$description_view;
echo "<a href='\".$book_link.\"'>..read more.</a>";
echo "</td></tr></table></td></tr>";
echo "<tr><td></td>";
$like_link_id = 'like_link'.$k;

```

```

$likes_display_id = 'likes_display'.$k; ///?
$nb_of_likes_id = 'nb_of_likes'.$k;
$insertcomment_id = 'insertcomment'.$k;
$comments_div_id = 'comments_div'.$k;
$num_likes_query = "select *,coalesce(count(like_id),0) as num_likes from
likes where update_id=$update_id group by update_id";
$likes_output = mysql_query($num_likes_query);
$likes_line = mysql_fetch_assoc($likes_output);
if(mysql_num_rows($likes_output)==0) $likes_line="0 likes";
else $likes_line = $likes_line['num_likes']." likes";
$user_like_exists_query = "select * from likes where user_id=$user_id
and update_id=$update_id";
$user_like_exists_output = mysql_query($user_like_exists_query);
if(mysql_num_rows($user_like_exists_output)==1){
    $like_text = "Unlike";
}
else $like_text = "Like";
echo "<td><a id='$like_link_id' href='\".$like_link.\"' onclick='return
clicked_like($user_id,$update_id,\"$like_link_id\", \"$nb_of_likes_id\")'>$like_text</a>&nbsp;&nbsp; ";
echo "<a onclick='return display_likes($update_id)' href='\"'><span
id='$nb_of_likes_id'>$likes_line</span></a></td></tr>";
echo "<tr><td></td><td><textarea rows='2' cols='30' placeholder='Write
a comment..' class='insertcomment' id='$insertcomment_id'
onkeypress='insertcomment(event, this, $user_id,
$update_id)'></textarea></td></tr></table>";
echo "<div class='comments_div' id='$comments_div_id'>";
display_comments($update_id,$db);
echo "</div>";
$k++;

```

(10) "processlikes.php":

```

if(isset($_GET['user_id']) && isset($_GET['update_id']) &&
isset($_GET['insertcomment']))){
    $user_id = $_GET['user_id'];
    $update_id = $_GET['update_id'];
    $comment_txt = $_GET['comment_txt'];
    $comment_txt = mysql_real_escape_string($comment_txt); //for char like '
    //echo $comment_txt;
    $insert_comment_query = "insert into comments
values(null,$user_id,$update_id, ".$comment_txt.", CURRENT_TIMESTAMP)";
    //echo $insert_comment_query;

    $result = mysql_query($insert_comment_query);

    echo "<span id='comments_div'>";
    display_comments($update_id,$db);
}
else if(isset($_GET['user_id']) && isset($_GET['update_id'])) { //to insert like in
the db and to load nblike
    $user_id = $_GET['user_id'];
    $update_id = $_GET['update_id'];
    $user_like_exists_query = "select * from likes where user_id=$user_id and
update_id=$update_id";
    $user_like_exists_output = mysql_query($user_like_exists_query);
    if(mysql_num_rows($user_like_exists_output)==0) {
        $modify_like_query = "insert into likes
values(null,$user_id,$update_id,CURRENT_TIMESTAMP)";
    }
    else $modify_like_query = "delete from likes where user_id=$user_id and
update_id=$update_id";
    if(mysql_query($modify_like_query)){
        $num_likes_query = "select *,count(like_id) as num_likes from
likes where update_id=$update_id group by update_id";
        $likes_output = mysql_query($num_likes_query);
        $line = mysql_fetch_assoc($likes_output);
        if(mysql_num_rows($likes_output)==0) $line = 0;
    }
}

```

```

        else $line = $line['num_likes'];
        echo "<span id='numlikes'>$line likes.</span>";
    }

}

else if(isset($_GET['update_id'])){           //to display likes in the modal
    $update_id = $_GET['update_id'];
    $likes_query = "select * from likes,user where likes.user_id=user.user_id
and update_id=$update_id";
    $likes_output = mysql_query($likes_query);
    echo "<span id='likes_display'>";
    while($line=mysql_fetch_assoc($likes_output)){
        $user_id = $line['user_id'];
        $fname = $line['first_name'];
        $lname = $line['last_name'];
        $href = "friend_profile.php?userid=$user_id";
        echo "<a href=$href>$fname $lname</a><hr>";
    }
    echo "</span>";
}

```

(11) "search.php"

```

<!DOCTYPE html>
<html>

<div id="search_results_div">
<?php
include "db_connection.php";

//if(isset($_GET['search'])){
    $filter = $_GET['filter'];
    if($filter==0){
        $all_books_query = "select * from book";

```

```

$books_output = mysql_query($all_books_query);
while ($line = mysql_fetch_assoc($books_output)) {
    $result_string = "";
    $result_string .= $line['book_name'].$line['description'];
    $key = $_GET['key'];
    $model = "([[:alnum:]] | [[:space:]])*$key([[:alnum:]] |
[[:space:]])*";
    if(@eregi($model, $result_string)){
        $book_name = $line['book_name'];
        $href = "book.php?book_id=".$line['book_id'];
        echo "<table border='0' width='550px'><tr><td
width='55px' valign='top'><img src='book_pics/'".$line['book_image']."'
style='object-fit: cover; width:60px; height:auto;'></td>";
        echo "<td><a href='".$href."' style='color:white;
text-decoration:none;'>".$book_name."</a></td></table>";
    }
}
}
else if($filter==1){
    $all_users_query = "select * from user";
    $users_output = mysql_query($all_users_query);
    while($line = mysql_fetch_assoc($users_output)){
        $result_string = "";
        $result_string .=
$line['user_id'].$line['user_name'].$line['first_name'].$line['last_name'].$line['em
ail'].$line['about_me'];
        $key = $_GET['key'];

        $model = "([[:alnum:]])*$key([[:alnum:]])*";
        if(@eregi($model, $result_string)){
            $name = $line['first_name']." ".$line['last_name'];
            $href = "friend_profile.php?userid=".$line['user_id'];
            echo "<table border='0' width='550px'><tr><td
width='55px' valign='top'><img src='user_profile_pics/'".$line['image']."'
style='object-fit: cover; width:50px; height:50px;'></td>";

```

```

        echo "<td><a href='".$href.'" style='color:white;
text-decoration:none'>".$name."</a></td></table>";
    }
}
}
else if($filter==2){
    $all_authors_query = "select * from author";
    $authors_output = mysql_query($all_authors_query);
    while($line=mysql_fetch_assoc($authors_output)){
        $result_string = "";
        $result_string .=
$line['fname'].$line['lname'].$line['about'].$line['website'];
        $key = $_GET['key'];
        $model = "([[:alnum:]]*$key([[:alnum:]]*)";
        if(@eregi($model, $result_string)){
            $name = $line['fname']." ".$line['lname'];
            $href = "author.php?author_id=".$line['author_id'];
            echo "<table border='0' width='550px'><tr><td
width='55px' valign='top'><img src='author_pics/".$line['image']."'
style='object-fit: cover; width:50px; height:50px;'></td>";
            echo "<td><a href='".$href.'" style='color:white;
text-decoration:none'>".$name."</a></td></table>";
        }
    }
}
//} continue for author and genre.
?>
</div>

</html>

```

(12) in "page1.php":

```

<div class="search_div">
  <form name='search_form' method="post" action="" id='search_form'>
    <h3 style='color: white;'>Discover</h3>
    <input type="text" name="search_key" onchange="display_search()"
onfocus="display_search()" oninput="display_search()" autocomplete="off">
<!--onchange happens only after losing focus -->
    <select name="filter" onchange="display_search()">
      <option value='0'>Book</option>
      <option value="1">User</option>
      <option value="2">Author</option>
      <option value="3">Genre</option>
    </select>
    <br/>
    <!--<input type="button" name="search" value="Search"
onclick="display_search()">
    <br/> -->
    <div id="search_result"></div>
  </form>
</div>

```

(13) "displaying\_shelf.php"

```
<?php
```

```
include "db_connection.php";
```

```
if(isset($_GET['state']) && isset($_GET['user_id'])){
```

```
    $state = $_GET['state'];
```

```
    $user_id = $_GET['user_id'];
```

```
    $shelf_query = "select * from added_book,book where
added_book.book_id=book.book_id and added_book.user_id = $user_id and
added_book.state=$state";
```

```
    $shelf_output = mysql_query($shelf_query);
```

```
    echo "<div id='shelf'>";
```

```
    echo "<h3 style='color:maroon;'>";
```



```

        if($state==0)
            echo "Books marked as Read:";
        else if($state==1)
            echo "Books marked as Currently Reading:";
        else if($state==2)
            echo "Books marked as Want to Read:";
        echo "</h3>";
        while ($line=mysql_fetch_assoc($shelf_output)) {
            $book_name = $line['book_name'];
            $href = "book.php?book_id=".$line['book_id'];
            echo "<table border='0' width='550px'><tr><td width='55px'
valign='top'><img src='book_pics/'".$line['book_image']."' style='object-fit: cover;
width:60px; height:auto;'></td>";
            echo "<td><a href='".$href."' style='color:white;
text-decoration:none;'>".$book_name."</a></td></table>";
        }
        echo "</div>";
    }

?>

```

(14) in "friend\_profile.php"

```

$requested_query = "select to_user from friend_request where
from_user='".$$_SESSION['userid']."' and
to_user='".$$_SESSION['friend_user_id']."'";
$output = mysql_query($requested_query);
$accept_query = "select to_user from friend_request where
to_user='".$$_SESSION['userid']."' and
from_user='".$$_SESSION['friend_user_id']."'";
$output1 = mysql_query($accept_query);
if(mysql_num_rows($output) == 1) echo "name='cancel' value='Cancel'>";
else if(mysql_num_rows($output1) == 1) {
    echo "name='confirm' value='Confirm'<br> ";
    echo "<input type='submit' name='ignore' value='Ignore'> ";
}

```

```

    }
    else if(isset($_SESSION['friends']) &&
in_array($_SESSION['friend_user_id'], $_SESSION['friends'])) echo
"name='remove' value='Remove Friend'>";
    else echo "name='add' value='Add as a Friend'>";

if(isset($_POST['add'])){
    $add_friend_query = "insert into friend_request
values(null,'" . $_SESSION['userid'] . "','" . $_SESSION['friend_user_id'] . "','" . CURRENT_TI
MESTAMP)";
    mysql_query($add_friend_query);
    header("Refresh:0");
}

else if(isset($_POST['cancel'])){
    $cancel_friend_query = "delete from friend_request where
from_user='" . $_SESSION['userid'] . "' and
to_user='" . $_SESSION['friend_user_id'] . "'";
    mysql_query($cancel_friend_query);
    header("Refresh:0");
}

else if(isset($_POST['confirm'])){
    $accept_friend_query = "insert into friend_relation
values(null,'" . $_SESSION['userid'] . "','" . $_SESSION['friend_user_id'] . "')";
    mysql_query($accept_friend_query);
    $accept_friend_query = "insert into friend_relation
values(null,'" . $_SESSION['friend_user_id'] . "','" . $_SESSION['userid'] . "')";
    mysql_query($accept_friend_query);
    $delete_request_query = "delete from friend_request where
from_user='" . $_SESSION['friend_user_id'] . "' and
to_user='" . $_SESSION['userid'] . "'";
    mysql_query($delete_request_query);

```

```

        header("Refresh:0");
    }

    else if(isset($_POST['ignore'])) {
        $ignore_friend_query = "delete from friend_request where
from_user='".$_SESSION['friend_user_id']."' and
to_user='".$_SESSION['userid']."'";
        mysql_query($ignore_friend_query);
        header("Refresh:0");
    }

    else if(isset($_POST['remove'])) {
        $delete_friend_query = "delete from friend_relation where user_id in
('".$_SESSION['friend_user_id']."','".$_SESSION['userid']."') and friend_user_id in
('".$_SESSION['friend_user_id']."','".$_SESSION['userid']."')";
        mysql_query($delete_friend_query);
        header("Refresh:0");
    }

```

(15) "book.php"

```

<html>
<head>
<title>House Of Words. Book</title>
<script src="jquery-3.3.1.min.js"></script>
<script>
document.addEventListener("keypress",function(event){
    if(event.keyCode==13)
        event.preventDefault();
});
</script>
<script type="text/javascript" src="functions_jscript.js"></script>

```

```

<link rel="stylesheet" type="text/css" href="book_stylesheet.css">
<?php session_start();
    include 'db_connection.php';
    include 'functions.php';
    if(isset($_GET['book_id'])){
        $book_id = $_GET['book_id'];
        $book_query = "select * from book where book_id=".$book_id;
        $output = mysql_query($book_query);
        if($line = mysql_fetch_assoc($output)) {}
        else header("location:page1.php");
    }
    else header("location:page1.php");

    if(isset($_SESSION['userid'])){
        update_friends();
        $ids = get_friends_ids();
        $user_id = $_SESSION['userid'];
        //$_SESSION['userid'] = $user_id; ??
        $book_state_query = "select * from added_book where
book_id=".$book_id." and user_id=".$user_id;
        $book_state_output = mysql_query($book_state_query);
        $line2 = mysql_fetch_assoc($book_state_output);
    }
    //updating the book's state
    if(isset($_GET['state'])) {
        $state = $_GET['state'];
        if(isset($line2['state'])){
            $_SESSION['row_exists'] = true;
        }
        else
            $_SESSION['row_exists'] = false;
        if($state != 0) { //want to read or readingnow(update-or-ins)
            if(!$_SESSION['row_exists']){
                $add_bookstate_query = "insert into added_book
values(null, ".$user_id.", ".$book_id.", ".$state.", null, null, CURRENT_TIMESTAMP)";

```

```

        mysql_query($add_bookstate_query);
    }
    else {
        $update_bookstate_query = "update added_book set
state=".$state.",rating=null,review=null where user_id=".$user_id." and
book_id=".$book_id;
        $output = mysql_query($update_bookstate_query);
        echo $update_bookstate_query;
    }
    $location = "book.php?book_id=".$book_id;
    header("location:$location");
}
else {
    header("location:rate.php?book_id=$book_id");
}
}

//the action of want to read button
$action = "book.php?book_id=".$book_id."&state=2";
echo "</head><body>
<form name='bookform' action=".$action." method='POST'
ENCTYPE='multipart/form-data'>
<div class='sidenav'>
<a href='page1.php'>Home</a>;
if(isset($_SESSION['username'])) {
    echo "<a href='profile.php'>Profile</a>
        <a href='friends.php'>Friends</a>
        <a href='requests.php'>Friend Requests</a>
        <input type='submit' name='logout' class='button' value='Sign Out'
onclick='document.bookform.action=\"book.php?book_id=$book_id\">";
    }
    else {
        echo "<a href='register.php' >Join Us</a>";
    }

echo "</div>

```

```

<div class='book' id='bookdiv'>";

echo "<table border='0'>";
echo "<tr><td rowspan='2' valign='top'><img class='book_img'
style='margin-right: 20px;' src='book_pics/" . $line['book_image'] . "' alt='avatar'></br>";
if(isset($_SESSION['username'])) {
    echo "<button class='dropbtn' name='statebtn' ";
    if(mysql_num_rows($book_state_output)==0){
        echo " onclick='document.bookform.submit()'>";
        echo "Want to Read</button>";
    }
    else {
        echo "onmouseover='this.style.backgroundColor= \"rgba(249,194,95,.1)\"";
this.style.color=\"white\"' onclick='return false'>";
        if($line2['state']==0)
            echo "Read";
        else if($line2['state']==1)
            echo "Reading Now";
        else if($line2['state']==2)
            echo "Want to Read";
    }
    echo "</button>";
    <div class='dropdown'>
        <input type='image' src='dropdown_down.png' class='btnimg'
onmouseover='""this.src=\"dropdown_up.png\"'
onmouseout='""this.src=\"dropdown_down.png\"' onclick='return false'
"."></button>
        <div class='dropdown-content'>";
        $href = "book.php?book_id=$book_id&state=";
        if(!isset($line2['state']) || $line2['state']!=1) echo "<a href='\" . $href . \"'1'
value='1'>Reading Now</a>";
        if(!isset($line2['state']) || $line2['state']!=0) echo "<a href='\" . $href . \"'0'
value='0'>Read</a>";

```

```

        if(isset($line2['state']) && $line2['state']!=2) echo "<a href='".$href.'"2'
value='2'>Want to Read</a>";
        //checkbox for adding a new shelf
        echo "<input type='checkbox' name='asd'>
        </div>
    </div>";
    if(isset($line2['state']) && $line2['state']==0){
        echo "<span style='margin-left: 10%; margin-top: 10%; font-size: 16px;
opacity: 0.6;'>My Rating</span></br>";
        echo "<span id='stars1' style='margin-left: 17%;'></span>";
    }
} ?>
<script type="text/javascript">view_stars(<?php if(isset($line2['rating'])) echo
$line2['rating']; else echo "0"; echo ",1" ?>);</script>

<?php
    echo "</td>";
    echo "<td valign='top' align='left' style='padding-left: 3%' colspan='2'><p
style='color: maroon; font-size: 30px; font-weight: bold;
margin-bottom: 1%'>".$line['book_name']. "</p>";

    $authors_query="select * from author,book_author,role where
book_author.book_id=".$book_id." and author.author_id=book_author.author_id
and book_author.role_id=role.role_id";
    $authors_output = mysql_query($authors_query);
    echo "<span style='color: rgba(255,255,255,0.7);'>by ";
    while($line1=mysql_fetch_assoc($authors_output)){
        echo $line1['fname']." ".$line1['lname']. "(".$line1['role_name'].") ";
    }
    echo "</span>";
    $rating = $line['avg_rating'];
    echo "</br>";
    echo "<progress value='".$rating.'" max='5' style='height: 5px; width:
85px'></progress>&nbsp;&nbsp;&nbsp;<span style='color:rgba(116, 164,
242,0.8);'>".$rating."&nbsp;&nbsp;&nbsp;".$line['num_ratings']. " ratings</span>";

```

```

echo "</td></tr>";

$description = $line['description'];
if(strlen($description)>300){
    $description_view = substr($description, 0 ,300);
    $description_hidden = substr($description, 300);
}
else $description_view = $description;
echo "<tr><td style='padding: 3%; ' valign='top' ><span>".$description_view;
if(isset($description_hidden))
    echo "</span><span id='desc'
hidden='true'>".$description_hidden."</span>".<a href='asd' id='readlink'
onclick='return view(\"readlink\", \"desc\")' style='color: #FFFF00 '>..read
more</a>";
    echo "</td></tr></table>";
    echo "<hr style='margin-right: 5%; opacity: 0.6'>";
    $i=2;
    $j=0; //variable for the review(hidden) elements
    $k=0; //variable for like and comment(id for each update)

    if(isset($_SESSION['userid'])){ //no friends reviews if not logged in
        echo "<h3 style='color: maroon'>My Review</h3><hr style='opacity: 0.3;
margin-right: 5%'>";
        if(isset($line2['state']) && $line2['state']==0 && $line2['review']!=null)
            echo "<span>".$line2['review']."</span>";
        else {
            if(isset($line2['state']) && $line2['state']==0)
                echo "You've marked it as 'Read'";
            else if(isset($line2['state']) && $line2['state']==1)
                echo "You've marked it as 'Reading now'";
            else if(isset($line2['state']) && $line2['state']==2)
                echo "You've marked it as 'Want to read'";
            echo "<br><span>You have not written any review yet.</span><pre
style='font-family: courier; font-size: 13px'>".$line2['timestamp']."</pre>";
        }
    }

```



```

    echo "<h3 style='color: maroon'>Friends Reviews</h3><hr style='opacity:
0.3; margin-right: 5%'>";
    if($ids=="") echo "You have not added any friends yet.";
    else {
        $get_friends_reviews_query = "select * from user a,updates b where
a.user_id=b.user_id and b.user_id in (".$ids.") and b.book_id=".$book_id." and
b.timestamp in (select max(timestamp) from updates where user_id=b.user_id and
book_id=b.book_id group by user_id,book_id)";
        $friends_reviews_output = mysql_query($get_friends_reviews_query);
        if(mysql_num_rows($friends_reviews_output)==0)
            echo "None of your friends have added any reviews of
".$line['book_name']. " yet.";
        else {
            while($line3=mysql_fetch_assoc($friends_reviews_output)){
                echo "<table border='0' width='550px'><tr><td rowspan='4' valign='top'
width='55px'><img src='user_profile_pics/'.".$line3['image']. "' style='object-fit:
cover; width:50px; height:50px;'></td>";
                $profile_link = "friend_profile.php?userid=".$line3['user_id'];
                if($line3['changed_state']==0){
                    echo "<td><a href=".".$profile_link." " style='text-decoration: none; color:
white; font-weight: bold'>". $line3['first_name']. " ". $line3['last_name']. "</a>";
                    if($line3['rating']!=null) {
                        echo " has rated it <span id='stars".$i."'></span>; ?>
                        <script type='text/javascript'>view_stars(<?php
if(isset($line3['rating'])) echo $line3['rating']; else echo "0"; echo ",".$i ?>);</script>
                        <?php
                        $i++;
                    }
                    else {echo " has read it.";}
                    echo "</td><td align='right' width='180px' style='font-family: courier;
font-size: 13px'>". $line3['timestamp']. "</td></tr>";
                    if($line3['review']!=null) echo "<tr><td>". $line3['review']. "</td></tr>";
                }
            }
        }
    }
}
else{

```

```

        echo "<tr><td><a href='\".$profile_link.\"' style='text-decoration: none;
color: white; font-weight: bold'>\".$line3['first_name'].\" \".$line3['last_name'].\"</a>
";
        if($line3['changed_state']==1)
            echo "is reading it now.</td>";
        else if($line3['changed_state']==2)
            echo "marked it as 'Want to read'.</td>";
        echo "</td><td align='right' width='180px' style='font-family: courier;
font-size: 13px'>\".$line3['timestamp'].\"</td></tr>";
    }
    $like_link =
"processlikes.php?user_id=".$user_id."&update_id=".$line3['update_id'];
    $update_id = $line3['update_id'];
    $like_link_id = 'like_link'.$k;
    $likes_display_id = 'likes_display'.$k;
    $nb_of_likes_id = 'nb_of_likes'.$k;
    $insertcomment_id = 'insertcomment'.$k;
    $comments_div_id = 'comments_div'.$k;
    $num_likes_query = "select *,coalesce(count(like_id),0) as num_likes
from likes where update_id=$update_id group by update_id";
    $likes_output = mysql_query($num_likes_query);
    $likes_line = mysql_fetch_assoc($likes_output);
    if(mysql_num_rows($likes_output)==0) $likes_line="0 likes";
    else $likes_line = $likes_line['num_likes']." likes";
    $user_like_exists_query = "select * from likes where user_id=$user_id
and update_id=$update_id";
    $user_like_exists_output = mysql_query($user_like_exists_query);
    if(mysql_num_rows($user_like_exists_output)==1){
        $like_text = "Unlike";
    }
    else $like_text = "Like";
    echo "<tr><td><a id='$like_link_id' href='\".$like_link.\"' onclick='return
clicked_like($user_id,$update_id,\"$like_link_id\", \"$nb_of_likes_id\")>$like_te
xt</a>&nbsp;&nbsp; ";

```

```

        echo "<a onclick='return display_likes($update_id)' href='><span
id='$nb_of_likes_id' >$likes_line</span></a></td></tr>";
        echo "<tr><td><textarea rows='2' cols='30' placeholder='Write a
comment..' class='insertcomment' id='$insertcomment_id'
onkeypress='insertcomment(event, this, $user_id,
$update_id)'></textarea></td></tr>";
        echo "<tr>";
        if($line3['review']!=null) echo "<td></td>";
        echo "</tr></table>";
        echo"<div style='margin-left:57px;'><button type='button'
onclick='\".\"hide_view(\"$comments_div_id\")\".\">View
comments</button></div><div class='comments_div' id='$comments_div_id'
hidden='true'>";
        display_comments($update_id,$db);
        echo "</div></div>";
        $k++;

    }//for while all lines

}
}
echo "<h3 style='color: maroon'>Community Reviews</h3><hr
style='opacity: 0.3; margin-right: 5%'>";
$get_community_reviews_query = "select * from user a,updates b where
a.user_id=b.user_id and b.user_id not in (\".$ids.\") and b.book_id=\".$book_id.\" and
b.timestamp in (select max(timestamp) from updates where user_id=b.user_id and
book_id=b.book_id group by user_id,book_id)";
}
else {          //if not signed in
    $get_community_reviews_query = "select * from user a,updates b where
a.user_id=b.user_id and b.book_id=\".$book_id.\" and b.timestamp in (select
max(timestamp) from updates where user_id=b.user_id and book_id=b.book_id
group by user_id,book_id)";
}
//in all cases , if signed in or not, friends exist or not

```

```

$community_reviews_output =
mysql_query($get_community_reviews_query);
if(mysql_num_rows($community_reviews_output)==0){
    echo "It seems to be a little quiet right here..";
}
// else if there is community reviews
else {
    while($line4=mysql_fetch_assoc($community_reviews_output)){
        echo "<table border='0' width='550px'><tr><td rowspan='4' valign='top'
width='55px'><img src='user_profile_pics/'.".$line4['image'].'" style='object-fit:
cover; width:50px; height:50px;'></td>";
        $profile_link = "friend_profile.php?userid=".$line4['user_id'];
        if($line4['changed_state']==0){
            echo "<td><a href='". $profile_link.'" style='text-decoration: none; color:
white; font-weight: bold'>". $line4['first_name']. " ". $line4['last_name']. "</a>";
            if($line4['rating']!=null) {
                echo " has rated it <span id='stars".$i.'"></span>"; ?>
                <script type="text/javascript">view_stars(<?php
if(isset($line4['rating'])) echo $line4['rating']; else echo "0"; echo ",$i"; ?>)</script>
                <?php
                $i++;
            }
            else {echo " has read it.";}
            echo "</td><td align='right' width='180px' style='font-family: courier;
font-size: 13px'>". $line4['timestamp']. "</td></tr>";
            if($line4['review']!=null) echo "<tr><td>". $line4['review']. "</td></tr>";
        }
    }
    else{
        echo "<td><a href='". $profile_link.'" style='text-decoration: none; color:
white; font-weight: bold'>". $line4['first_name']. " ". $line4['last_name']. "</a> ";
        if($line4['changed_state']==1)
            echo "is reading it now.</td>";
        else if($line4['changed_state']==2)
            echo "marked it as 'Want to read'.</td>";
    }
}

```

```

        echo "</td><td align='right' width='180px' style='font-family: courier;
font-size: 13px'>".$line4['timestamp']."</td></tr>";
    }

```

```

$update_id = $line4['update_id'];
$nb_of_likes_id = 'nb_of_likes'.$k;
$num_likes_query = "select *,coalesce(count(like_id),0) as num_likes
from likes where update_id=$update_id group by update_id";

```

```

$likes_output = mysql_query($num_likes_query);
$likes_line = mysql_fetch_assoc($likes_output);
if(mysql_num_rows($likes_output)==0) $likes_line="0 likes";
else $likes_line = $likes_line['num_likes']." likes";

```

```

if(isset($_SESSION['userid'])) {
    $like_link =
"processlikes.php?user_id=".$_user_id."&update_id=".$_line4['update_id'];
    $like_link_id = 'like_link'.$k;
    $likes_display_id = 'likes_display'.$k;
    $user_like_exists_query = "select * from likes where user_id=$_user_id
and update_id=$update_id";

```

```

$user_like_exists_output = mysql_query($user_like_exists_query);
if(mysql_num_rows($user_like_exists_output)==1){
    $like_text = "Unlike";
}
else $like_text = "Like";
echo "<tr><td><a id='$like_link_id' href='".$_like_link.'" onclick='return
clicked_like($_user_id,$_update_id,\"$like_link_id\", \"$nb_of_likes_id\")'>$like_text</a>&nbsp;&nbsp; ";

```

```

}
else echo "<tr><td>";

```

```

        echo "<a onclick='return display_likes($update_id)' href='><span
id='$nb_of_likes_id' >$likes_line</span></a></td></tr></table>";
        $comments_div_id = 'comments_div'.$k;
        echo "<div style='margin-left:57px;'><button type='button'
onclick='\".\"hide_view(\"$comments_div_id\")\".\">View
comments</button></br><div class='comments_div' id='$comments_div_id'
hidden='true'>";
        display_comments($update_id,$db);
        echo "</div></div></br>";
        $k++;
    } //for while

}
echo "</div>";
echo "</div>";

// if(!isset($_SESSION['username'])) header("location:login.php");
$author_information_query="select * from author,book_author,country
where book_author.book_id=".$book_id." and book_author.role_id=1 and
author.author_id=book_author.author_id and
author.country_code=country.code";
$author_information_output = mysql_query($author_information_query);
if(mysql_num_rows($author_information_output)==1){
    $line = mysql_fetch_assoc($author_information_output);
    echo "<div class='column'>";
    echo "<div class='card'>";
    echo "<img src='author_pics/'."$line['image'].\".\" alt='author_pics/avatar.jpg'
style='width:100%;'>";
    echo "<div class='container'>";
    echo "<h2 style='color:white;'>\"."$line['fname'].\" \"."$line['lname'].\"</h2>";
    //print_r($line);
    echo "<h3 style='color: rgba(255,255,255,0.6);'>\"."$line['Name'].\"</h3>";
    $author_id = $line['author_id'];
    $href = "author.php?author_id=$author_id";

```

```

        echo "<a href='\". $href.\"' id='authorlink' style='color: rgba(255,255,255,0.8);
text-decoration: none; background-color: maroon; :hover:yellow'>Know the
Author</a>";
        echo "</div></div></div>";
    }

```

```

?>
<!-- The Modal -->
<div id="myModal" class="modal">
    <!-- Modal content -->
    <div class="modal-content">
        <span class="close">&times;</span>
        <p id='likers_modal'>Some text in the Modal..</p>
    </div>
</div>
</form>
</body>
</html>

```

(16) "rate.php"

```

<html>
<head>
<title>House Of Words. Register</title>
<link rel="stylesheet" type="text/css" href="rate_stylesheet.css">
<script>
    function view_stars(myRating){
        var theImage = "";
        for(var i=0; i<myRating.value; i++)
            theImage += "<img src='star.png' width='20px' height='20px'>";
        document.getElementById('stars').innerHTML = theImage;
    }
</script>

```

```

<?php session_start();
include 'db_connection.php';
if(isset($_GET['book_id'])){
    $book_id = $_GET['book_id'];
    $book_query = "select * from book where book_id=".$_book_id;
    $output = mysql_query($book_query);
    if($line = mysql_fetch_assoc($output)) {}
    else header("location:page1.php");
}
else header("location:page1.php");

if(isset($_POST['save'])){
    if($_POST['rating']==0)
        $rating = null;
    else
        $rating = $_POST['rating'];
    if($_POST['review']== "")
        $review = null;
    else
        $review = $_POST['review'];
    if(!$ _SESSION['row_exists']) {
        $review_query = "insert into added_book
values(null,".$ _SESSION['userid'].",$book_id,0,$rating,".$review.",CURRENT_TIM
ESTAMP)";
    }

    else {
        $review_query = "update added_book set
state=0.",rating=".$rating.",review=".$review.",timestamp=CURRENT_TIMESTA
MP where user_id=".$ _SESSION['userid']. " and book_id=".$book_id;
    }

    if(mysql_query($review_query)){
        header("location:book.php?book_id=$book_id");
    }
}

```



```
}  
}
```

```
$action = "rate.php?book_id=".$book_id;  
echo "</head><body>  
  <form name='ratingsform' action=$action method='POST'  
ENCTYPE='multipart/form-data'>  
  <div class='sidenav'>  
    <a href='page1.php'>Home</a>;  
    if(isset($_SESSION['username'])) {  
      echo "<a href='profile.php'>Profile</a>  
        <a href='friends.php'>Friends</a>  
        <a href='requests.php'>Friend Requests</a>  
        <input type='submit' name='logout' class='button' value='Sign Out'>";  
    }  
    else {  
      echo "<a href='register.php' >Join Us</a>";  
    }  
  }  
}
```

```
echo "</div>
```

```
<div class='book' id='bookdiv'>;
```

```
  echo "<table border='0'>";  
  echo "<tr><td rowspan='4' valign='top'><img class='book_img'  
style='margin-right: 20px;' src='book_pics/'."$line['book_image'].""." "  
alt='avatar'></td></br>";
```

```
  $authors_query="select * from author,book_author,role where  
book_author.book_id=".$book_id." and author.author_id=book_author.author_id  
and book_author.role_id=role.role_id";  
  $authors_output = mysql_query($authors_query);  
  while($line1=mysql_fetch_assoc($authors_output)){  
    if($line1['role_id'] == 1){          //author  
      $author = $line1['fname']." ".$line1['lname'];
```

```

        break;
    }
}

echo "<td valign='top' align='left' style='padding-left: 3%' colspan='2'><p
style='color: maroon; font-size: 30px; margin-bottom:1%'>Write Your Review of
\".$line['book_name'].\" by \".$author.\".</p></br>";

    echo "Your Rating &nbsp;&nbsp;<select name='rating'
onchange='view_stars(this)'>
    <option value='0'></option>
    <option value='1'>1</option>
    <option value='2'>2</option>
    <option value='3'>3</option>
    <option value='4'>4</option>
    <option value='5'>5</option>
</select>&nbsp;&nbsp;&nbsp;
<span id='stars'></span></br></br>";

    echo "Your Review</br></br><textarea maxlength='2000'
name='review'></textarea></br></br>";

    echo "<input type='submit' value='Save' name='save' class='savebtn'>";

echo "</td></tr></table>";

echo "</div>";

if(isset($_POST['logout'])) {
    //unset($_SESSION['username']);
    $_SESSION['signed_out'] = true;
    header("location:login.php");
}
// if(!isset($_SESSION['username'])) header("location:login.php");

```

```

$author_information_query="select * from author,book_author,country
where book_author.book_id=".$book_id." and book_author.role_id=1 and
author.author_id=book_author.author_id and
author.country_code=country.code";
$author_information_output = mysql_query($author_information_query);
if(mysql_num_rows($author_information_output)==1){
    $line = mysql_fetch_assoc($author_information_output);
    echo "<div class='column'>";
    echo "<div class='card'>";
    echo "<img src='author_pics/'."$line['image'].\"\".\" alt='author_pics/avatar.jpg'
style='width:100%;'>";
    echo "<div class='container'>";
    echo "<h2 style='color:white;'>".$line['fname'].\" \".$line['lname'].\"</h2>";
    //print_r($line);
    echo "<h3 style='color: rgba(255,255,255,0.6);'>".$line['Name'].\"</h3>";
    $author_id = $line['author_id'];
    $href = "author.php?author_id=$author_id";
    echo "<a href='\".$href.\"' id='authorlink' style='color: rgba(255,255,255,0.8);
text-decoration: none; background-color: maroon; :hover:yellow'>Know the
Author</a>";
    echo "</div></div></div>";
}

```

```

?>
</form>
</body>
</html>

```

