



TECHNICAL PROJECT REPORT

Zoe Fabre-Anderson



STUDENT NUMBER: D22125460
Course Code: TU858 Year 3

Contents

Introduction.....	3
Programs & Languages:	3
Directions for Running the Web App:	3
Preliminary Design	4
Wireframes:	4
Login.....	4
SignUp.....	4
Home Page	5
New Journal Entry.....	5
View Journal Contents.....	6
Final Design:.....	6
Home.php.....	7
View_entry.php.....	8
Add_entry.php	9
Other PHP Pages:	9
Back-end:.....	11
Front-end:	12
Data:.....	12
User Table	13
Journal Entry Table	13
Use-Case	14
Scenarios:	15
1. User Registration	15
2. Add New Journal Entry	15
3. Delete Journal Entry	15
Appendix A	16
Login.php & Processing_input.php	16
Signup_input.php & Signup.php	17
Appendix B	17
Add_entry.php	17
Add_style.css	21
Connection_db.php.....	24
Delete_entry.php.....	24
Home.php.....	26

Login_style.css	30
Login.php.....	34
Logout.php	36
Processing_input.php.....	36
Signup_input.php	38
Signup.php	40
View_entry.php.....	42
View_image.php	44
View_style.css	45

Introduction

For this project I have decided to create a journaling web app. I have called it Secret Journal. A user creates an account which then allows them to make digital journal entries. The user can look back on past entries much like a real physical journal.

I chose this to be my web app because my travels abroad on Erasmus. During my time abroad, I made many new friends and memories. I participated in many activities and trips. A part of me wanted a way to document my daily activities so that I would have a permanent record of time that I could look back on. This is what fueled my desire to create Secret Journal. It is “secret” in that you must have a username and password in order to access a specific user’s journal entries. Had I had a web app such as this one while abroad I would have used it everyday to fill with all my thoughts and memories.

Programs & Languages:

- **Languages:** I coded primarily in HTML and PHP, with some small amounts of SQL (for queries) and CSS (for styling). I use JavaScript to provide confirmation dialog boxes for the user.
- **Servers:** I used the XAMPP Control Panel in order to run an Apache server to host the site, and MySQL to run the database server. User data and the journal entries data are stored in the MySQL database. PhpMyAdmin is a free open source admin tool that allows me to host my MySQL database.
- **Code Environment:** VSCode.

Directions for Running the Web App:

A readme has been included in the project folder, however I will also state it here. In order to run my web app, you must have the XAMPP control panel. In the XAMPP program files you will need to put the program files (or folder) into the htdocs folder. Start Apache and then MySQL. Next you will have to create a database called “journaldb” in phpMyAdmin. Create two tables: journalentries, and users. The sql files have been provided in the “database-files” folder, so you only need to import. I

recommended creating the database yourself first (not importing the file), and then importing the tables. Then click on the following url. You may need to alter this url, depending on the file path of the project files: <http://localhost/credit-project/login.php>

Preliminary Design

Wireframes:

Login

Secret Journal

Login?

Username

Password

Don't Have an Account?

Sign Up Here?

The user is greeted by this page. The logo of the site is the words “Secret Journal” on top of a key. This logo is to reinforce to the user that this is a private journal for their eyes only. The user may login with a username and password if they already have an account. If not, they can click on the signup button to create one.

SignUp

Create An Account

Username

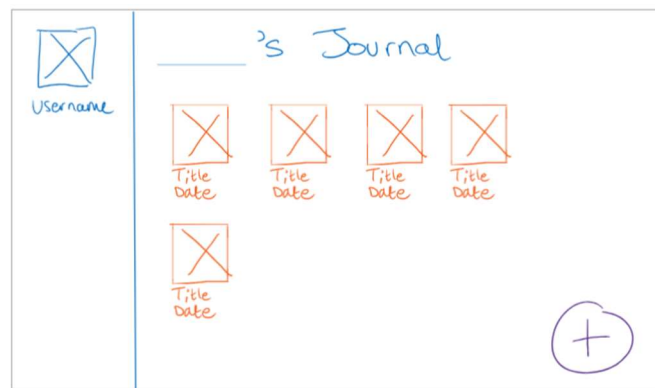
Password

Retype Password

create

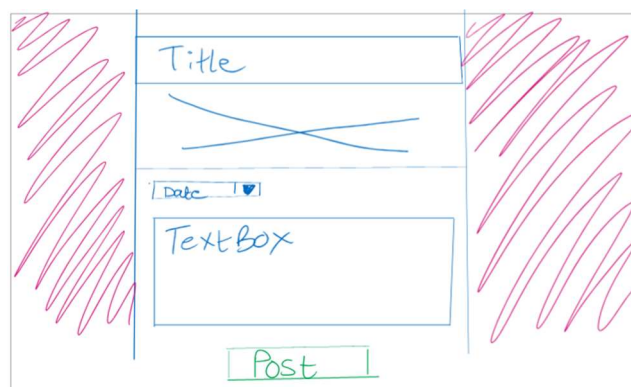
The user enters in a username and password. The username must be unique. The user then enters a password, and retypes it for password confirmation. Clicking create, enter's this new user into the user's table and then brings the user back to the home page to now login as described before.

Home Page



When logged in, the user can view their already created journal entries. A button at the bottom when clicked brings them to a page where a new journal entry can be made. The entries are ordered by date, from most new to old. On the left is the user profile. It displays a profile picture, and their username. On top of the page it says [User]'s Journal. If they are a new user this page will be most blank. The journal entries are formatted as cards, with the uploaded image being on the cover. If you click on a entry card it brings you to a new page where you can view that journal entry in full.

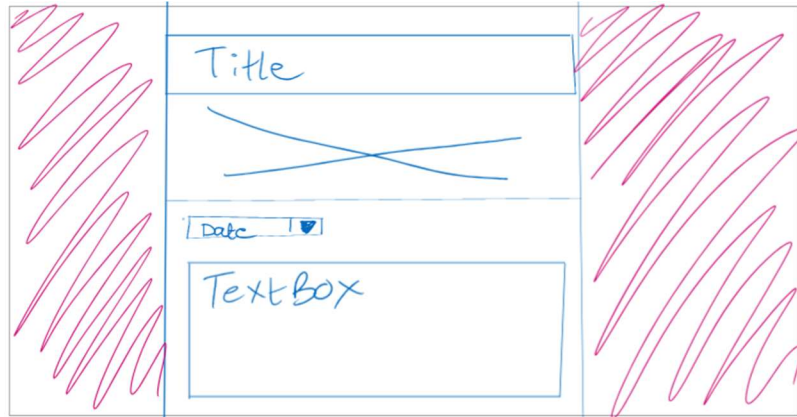
New Journal Entry



The user titles the entry. They can then upload a picture from their gallery, if they want. They can then either type in a day (in numbers) or click the arrow to drop down the

calendar where they can click a specific day. The user clicks “Post” in order to create a new entry in the journalentries table, and therefore adding it to their home page.

View Journal Contents



A hand-drawn wireframe of a journal entry view. The layout is contained within a rectangular frame. On the left and right sides of the frame, there are vertical columns filled with pink, wavy scribbles. The central area is divided into four horizontal sections by thin blue lines. The top section is a rectangular box labeled "Title" in blue handwriting. The second section contains a blue 'X' mark. The third section is a smaller rectangular box labeled "Date" in blue, followed by a small blue heart icon. The bottom section is a larger rectangular box labeled "Text Box" in blue.

When the user clicks on an already created journal entry, it opens it up separately in a new page. They cannot alter the data. A back button, brings the user back to the home page.

Final Design:

As I do not want to repeat myself I will only put select pages where the design has been significantly changed. With screenshots of the other pages in the [Appendix A](#) section.

I have streamlined the over all design. The logo on the login.php and signup.php was created by me. On every page I put a disclaimer in the footer to state that all assets are free use/copyright free. It can't be see in my screenshots, but I also made the mouse have a custom design of a pen. I picked a a custom font that is the style of handwriting but still well legible. These aesthetic choices were made in order to fit the theme of a personal journal.

Home.php



As you can see this quite different to my initial wireframe. I was not able to implement a user uploading a profile picture and having it displayed. For the design, I put notebook lines in the background to give the feeling of having opened a journal from logging into an account. I added a page curl image to each card, as I wanted to recreate the physical experience of flicking back the pages in a journal. Journal entries are in the order of oldest to newest, so you have to scroll down to your latest entry.

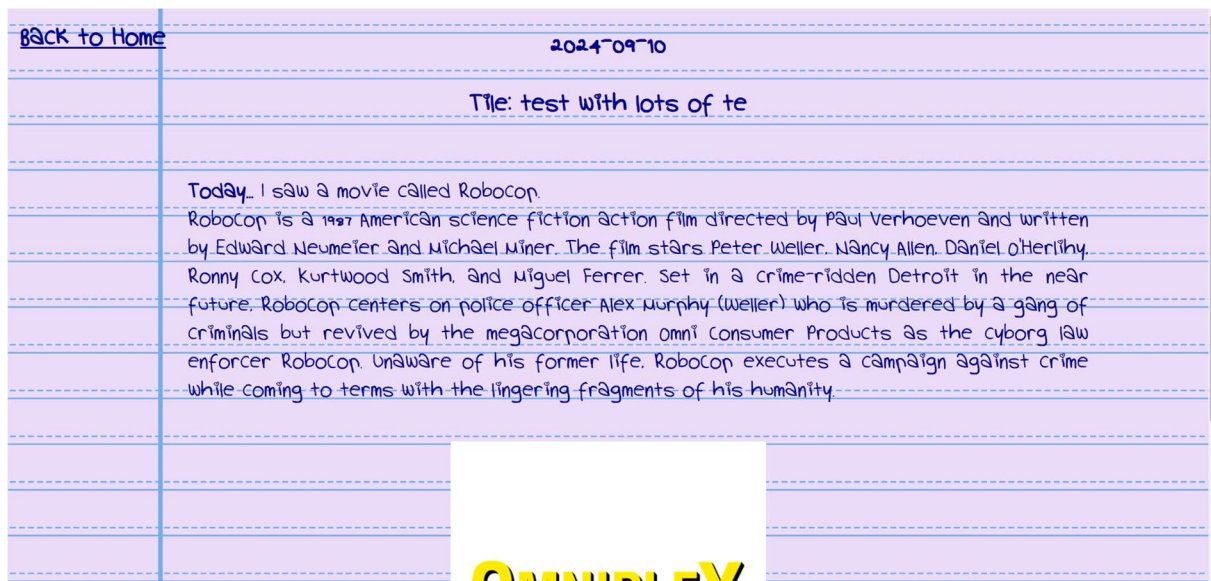
This on my test account which I used for checking and testing as I coded the web app. Near the top of the page is the button to logout and to create a new entry. I want to move these button further apart, however I was unable to. It is not possible to enter someone else Journal without having logged in successfully through the login page. You also cannot by pass the login page by changing the url. These were just some of the security measure I took to improve the overall integrity of the web app. If a did not upload an image, or it didn't upload to the database properly, no image is displayed as seen in some of the tests I conducted. If the user upload a .gif file, then it will play within the card.

On each card is a delete button, that when clicked, deletes the entry from both the home page and the database. When the delete button is clicked, the user is given a chance to cancel the action as a message appears at the top asking if they wish to continue. This prevents an entry from being deleted accidentally. When the "Add a new

entry” button is clicked, it will bring the user to the add_entry.php page. The user is able to log out from the home page, bringing the back to the login page.

To retrieve this data, the userID is fetched from the database through the user session. As the users and journalentries tables are linked, its easy to query for the specific entries of the sessioned user. View_image.php allows for the displaying of the images by essentially reconstructing the file along with its file extension.

View_entry.php



When the user clicks on a journal entry card, they are brought to the view_entry.php page. Here the date is at the tope, then the title, under that is the main entry text, and lastly, is the image uploaded. For this entry I tested to see what it was like when there is a lot of text. I hit the character limit in the Title, but not the in the Text. I carried over the background image. The method to call all the journal data is the same as for the home page. View_image.php allows for the image to be viewable.

I would have liked to have it so the user could edit parts of the journal, say the date or the text, in the occasion data was entered in incorrectly. However, I did not have enough time to do so.

Add_entry.php

Back to Home

Title:

Date:

Dear Journal Today...

Image (optional): No file chosen

All cursor, logo, font assets used in this website are copyright free. Website created by Zoe Fabre-Anderson 2024

The user must fill in a Title, Date and Text. Only the image upload is optional. If required fields are not filled, a message will appear telling the user. The user can type in the date manually or click the little calendar icon to select a date. When the user clicks the “choose file” button, the file explorer of their laptop/pc pops open. The maximum file size is 40MiB which is roughly equivalent to 41MB. When the user is finished, they then click the “Post” button to upload the entry to the database. The user is then redirected to the home page where they will see that their new journal entry.

I would have liked to add more things for the user to fill in, perhaps a star rating where the user could rate the day, among other things. However, I did not have the time, nor the knowledge to implement these.

Other PHP Pages:

- **Connection_db.php:** This file connects the database to the website. If there is a problem while connecting, it kills the connection. This file is called in every other php file.
- **Delete_entry.php:** This is where the delete statement is stored, ready to be called. Using \$_POST the delete statement is sent to the database. The delete

button on the home page, is stored inside a form that calls the delete_entry.php. In this way only a specific entry is deleted.

- **Login.php:** Contains the html layout of the page. It calls processing_input.php to verify the user attempting to log in. Once logged in, the user's username is stored in a session. A user cannot skip this page to jump to other pages. I have ensured they will be redirected to the login page.
- **Logout.php:** This is a very short script. It destroys the session and redirects the user to the login page.
- **Processing_input.php:** Similar to delete_entry.php this is the php script that creates a premade sql statement to be posted to the database, only this is for verifying the user. It checks if the input fields are blank, if the username exists, and if the password is correct. If either field (username or password) is inputted wrong, then a warning message is displayed. It will not specify which field is wrong for better security. If the information is correct, a session is created and the user is redirected to the home page.
- **Signup_input.php:** This file contains the html tags for the overall layout of the page. The input fields are incased in a form. The user must enter a unique username, a password, and then re-type the password. There is error handling in the case there are fields left empty, or inputted incorrectly. When "create account" is clicked a new entry is made in the users table with the username and password they inputted.
- **Signup.php:** Pretty much the same as processing_input.php, except instead of comparing data, this file actively adds an entry into the users table. It checks that the password and re-typed password match. When the create account button is clicked it posts the insert sql statement to the database, adding in a new user. If successful, the user is redirected to the login page, where they input their account details.
- **View_image.php:** This script is called in the view_entry.php and home.php in order to display the image of a specific journal entry. The image that is stored as binary data is then converted into its correct file type by seeing which type is

stored in the image_extension column in the table. The image is then sent to the browser. If there is no image, then an appropriate message is sent.

Back-end:

- **Session Management & Authentication:** I used the function session_start() to manage user sessions. It checks if the user is logged in if(isset(\$_SESSION['User'])), and if not they are redirected to a login page. This is the foundation to my security protections. On pages where the user's journal data is required, this can be pulled easily by using the sessioned username to get to that user's journal data.
- **Database Interaction:** I used prepared sql statements \$stat->bind_param() to interact with the database. This prevents SQL injection. I also query the database for journal entries and user details, inserting a new entry. The connection_db.php script creates the connection to the database, and is called in every other php file.
- **Retrieving User Data:** In pages like home.php where the journal data is required, the UserID is fetched through the sessioned Username. SELECT UserID users WHERE Username =? Ensures that the correct user is identified. As the users and journalentries tables have a relationship of 1 to Many, I then create a new query statement to select that user's journal data.
- **Data Deletion:** In home.php, a user can delete a journal entry by clicking the delete button. This calls the delete_entry.php script to process the action when the form is submitted (the button is incased in a form).
- **Handling User Input:** Users input text and can upload images which are then stored as records in the database. The backend processing for the user input and files can be seen in files such as processing_input.php, add_entry.php, signup.php.**
- **Image Processing:** The view_image.php retrieves the image, and the image extension from the database. Once fetched, the script checks if an image exists. A switch determines the MIME type based on the image extension. This is

necessary for the browser to correctly interpret and display the image. When the user uploads a picture, a similar process happens in reverse.

Front-end:


- **Styling:** Across my webapp I used a variety of css scripts for the styling. While I there is some structural layout and design in the html, I tried to keep most of the styling in the css. In the css sheets is where I call the background images, custom font, and custom mouse cursor assets.
- **Structure:** For each page there is a HTML structure of a header, divs and footer tags. In the header I have the page title and navigation links. The divs contain the main content. In the footer I put a notice that all assets are copyright free, and my author info.
- **Dynamic Content:** I used PHP to inject dynamic content into the HTML, e.g. home.php I call the username of the sessioned user: `<?php echo htmlspecialchars($username); ?>`, and also the journal entries with a while loop to fetch the journal entries. I also use `htmlspecialchars()` to ensure the content is safely escaped to prevent cross-site scripting attacks.
- **Javascript:** I used JavaScript to create confirmation dialogs with the user. In the case of home.php, when the user clicks delete on a journal entry, a pop up message appears asking if the user is certain they want to carry out this action: `onsubmit="return confirm('Are you sure you want to delete this entry?');"`. JS also allowed me to make the journal entry card to be made clickable: `onclick="window.location.href='view_entry.php?id={JournalID}'"`.
- **Images:** the `img` tag dynamically loads images from `view_image.php` using the `JournalID` to reference the correct image for each journal entry: `src="view_image.php?id=<?php echo htmlspecialchars($entry['JournalID']); ?>"`

Data:

I created the database called `journaldb`, using the `phpMyAdmin` interface. This was easier than writing it from scratch in SQL. There are two tables. One to store user data, and the other to store the journal entries. They have a relationship of 1 to Many such



that one user has many journal entries. These tables are stored on phpMyAdmin, and are accessed via the MySQL server provided by XAMPP.

User Table

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	UserID 	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	Username 	varchar(16)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	3	Password	varchar(20)	utf8mb4_general_ci		No	None		

UserID is the primary key. Usernames must be unique. When a new user is created, the UserID is set to auto-increment so that there is no input needed. All fields are required. I have protections in place to prevent sql injection, and so that data entries that don't meet the requirements of the table are not inputted. Character limits are put on the Username and Password.

Journal Entry Table

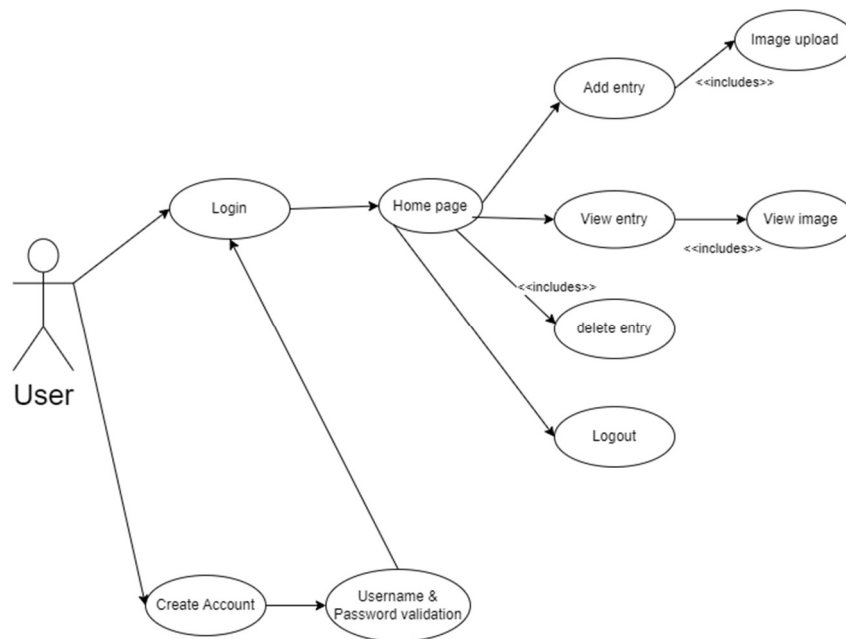
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	JournalID 	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	UserID 	int(11)			No	None		
<input type="checkbox"/>	3	Title	varchar(20)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	4	Entry_Date	date			No	None		
<input type="checkbox"/>	5	Entry_Text	varchar(1000)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	6	Entry_Image	longblob			Yes	NULL		
<input type="checkbox"/>	7	Image_Extension	varchar(10)	utf8mb4_general_ci		Yes	NULL		

JournalID is the primary key. UserID is the foreign key, it references a user in the User Table. Title and Date are non-unique. The uploaded Entry_Image is saved as a file in the data base. Title has a character limit of 20, as the is people write more in the Text section. Text is the main body of text in the journal entry. It will have a character limit of 1000. Image_Extension saves the file extension of an image (png, jpeg, gif), and allows for images to uploaded through my web app and to then be queried. Otherwise the only way an image would be viewable on my web app, is if you uploaded it manually through phpMyAdmin. Uploading an image is optional and not necessary for a user to create a successful journal entry. Like UserID in the users table, the JournalID is set to auto-increment. UserID is filled in by fetching the UserID from the database using the

sessioned Username. I have protections in place to prevent sql injection, and so that data entries that don't meet the requirements of the table are not inputted.

I wanted to change the date format to match the Irish format, however I was not familiar enough with the phpMyAdmin in order to do this.

Use-Case



This is the use-case diagram of my web app. Starting off the User can either login or create a new account i.e. sign up. If they choose to sign up, then their username and password is validated. After which, they are brought to the login page again. After logging in, the user is redirected to the home page. They have the choice to add a new entry, view a pre-existing entry, or delete an entry. Creating a new entry, the user fills the required fields and has the option to upload an image. If the user clicks to view the entry, then the image and along with the text they inputted will be displayed. If a user clicks delete, then the entry is deleted. Lastly the user can then Logout, returning them to the login page.

Scenarios:

1. User Registration

- The user wishes to create a new account. They have clicked sign up. Now on the sign up page, they must fill 3 fields: Username, Password, Re-type password.
- If username is left empty, an error message appears to notify the user. If either password field is left empty or they don't match, an error message appears.
- Once all fields are filled, both passwords match, the user clicks Create Account and is redirected to the login.

2. Add New Journal Entry

- The user is on the home page and clicks the add new entry button.
- The user is brought to the add entry page. They are required to fill in a title, date, and the main text of the entry. The user has the option upload an image.
- If the choose file button is clicked, then the user's file explorer pops open allowing them to select the image.
- Once the user has filled out the page to their satisfaction, they click the post button.
- The user is redirected to the home page, and will see their new journal entry is available.
- If an errors occurs while uploading, or fields are not filled appropriately an error message will display notifying the user.

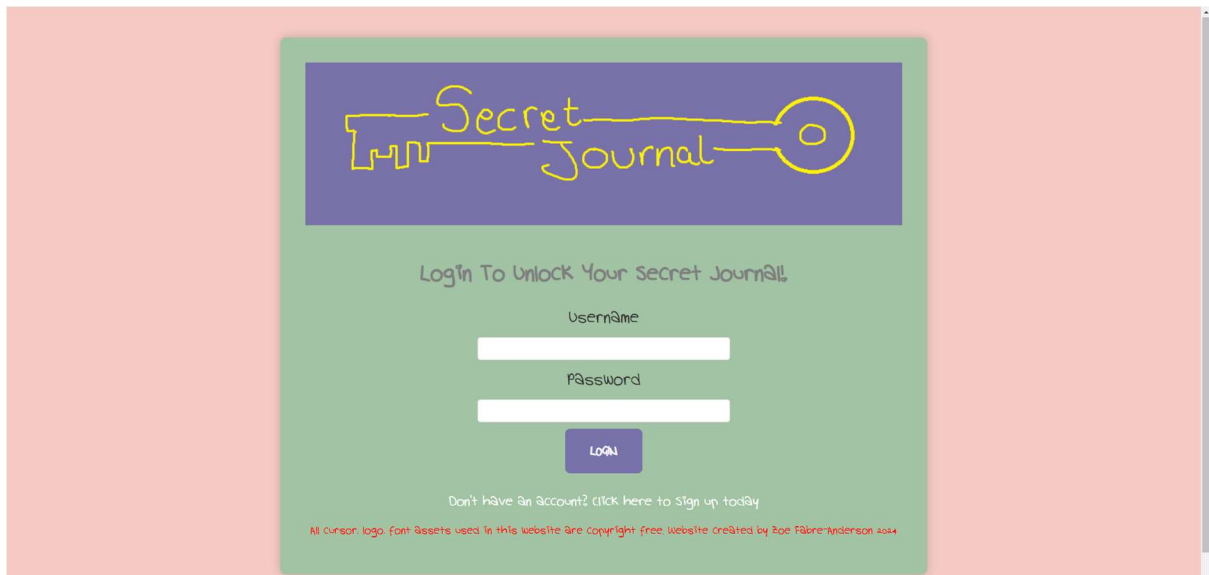
3. Delete Journal Entry

- There is no way to alter a journal entry after it has been added. If there is a mistake, then the only course of action is to delete the entry and recreate again.
- The delete button is also there in the case the user doesn't want that entry to exist anymore, for one reason or another.

- The user clicks the red delete button on the entry.
- A pop-up message asks to the user if they are certain they want to proceed. Clicking “ok” will carry out the delete action. Clicking “cancel” will stop it.
- The user clicks “ok”, the home page reloads updating to show one less entry.

Appendix A

Login.php & Processing_input.php



Signup_input.php & Signup.php

Secret Journal

Sign Up

Username

Password

Re-Type Password

CREATE ACCOUNT

Already have an account? Login here!

All cursor logo font assets used in this website are copyright free. Website created by Zoe Fabre-Anderson 2024

Appendix B

Add_entry.php

```
<?php

require_once("connection_db.php");

session_start();

if(isset($_SESSION['User'])){

}

else{

    header('location:login.php');

}

$username = $_SESSION['User'];

#fetch userID from the database using sessioned username

$query = "SELECT UserID FROM users WHERE Username =?";

$stmt = $con->prepare($query);
```

```
$stat->bind_param("s", $username);

$stat->execute();

$result = $stat->get_result();


if($result->num_rows ==1){

    $row = $result->fetch_assoc();

    $userID = $row['UserID'];

}

else{

    echo "User not found";

    exit;

}


if($_SERVER['REQUEST_METHOD'] == 'POST'){

    if(isset($_POST['title'], $_POST['entry_date'], $_POST['entry_text'])){

        $title = $_POST['title'];

        $entry_date = $_POST['entry_date'];

        $entry_text = $_POST['entry_text'];

        $image = $_FILES['image'];


        #validate data

        if(empty($title) || empty($entry_date) || empty($entry_text)) {

            echo "All fields are required.";

        }

        elseif(strlen($entry_text) > 1000){

            echo "Cannot exceed 1000 characters!";

        }

        else{

            $imageData = null;
```

```

$imageExtension = null;
if($image['error'] == UPLOAD_ERR_OK){
    $imageData = file_get_contents($image['tmp_name']);
    #to get file extension
    $imageExtension = pathinfo($image['name'], PATHINFO_EXTENSION);
}

#insert new entry into the database

$query = "INSERT INTO journaldb.journalentries (UserID, Title, Entry_Date, Entry_Text, Entry_Im
mage, Image_Extension) VALUES (?, ?, ?, ?, ?, ?)";

$stat = $con->prepare($query);

$stat->
>bind_param("isssss", $userID, $title, $entry_date, $entry_text, $imageData, $imageExtension);

$stat->execute();

#redirect to home.php if successful insertion
header('Location: home.php');
exit;
}
}
else{
    echo "Missing form data";
}
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">

```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Add Journal Entry</title>
```

```
<link rel="stylesheet" href="add_style.css">
```

```
</head>
```

```
<body>
```

```
<form action="add_entry.php" method="post" enctype="multipart/form-data">
```

```
<p>
```

```
<label for="title"><strong>Title:</strong></label>
```

```
<input type="text" id="title" name="title" required>
```

```
</p>
```

```
<p>
```

```
<label for="entry_date"><strong>Date:</strong></label>
```

```
<input type="date" id="entry_date" name="entry_date" required>
```

```
</p>
```

```
<p>
```

```
<label for="entry_text">Dear Journal, Today...</label>
```

```
<textarea name="entry_text" id="entry_text" cols="50" rows="5" maxlength="1000" required>
</textarea>
```

```
</p>
```

```
<p>
```

```
<label for="image"><strong>Image (Optional):</strong></label>
```

```
<input type="file" id="image" name="image" accept="image/*">
```

```
</p><br>
```

```
<button type="submit" class="custom-post-button">Post</button>
```

```
</form>
```

```
<a href="home.php">Back to Home</a>
```

```
<footer>
```

```
<p>
```

All cursor, logo, font assets used in this website are copyright free.

Website created by Zoe Fabre-Anderson 2024

```
</p>
```

```
</footer>
```

```
</body>
```

```
</html>
```

Add_style.css

```
@font-face{
```

```
font-family: 'WritingFont';
```

```
src: url(Handwriting-poxD.ttf) format('truetype');
```

```
font-weight: normal;
```

```
font-style: normal;
```

```
}
```

```
body{
```

```
margin: 0;
```

```
padding: 0;
```

```
font-family: 'WritingFont', sans-serif;
```

```
background-color: rgba(120, 113, 170, 1.0);
```

```
background-image: url('paper-texture.jpg');
```

```
background-size: cover;
```

```
background-position: center;
background-repeat: no-repeat;
background-blend-mode: overlay;
text-align: center;
cursor: url('fountain pen handwriting.cur'), auto;
```

```
}
```

```
header{
    color: darkblue;
```

```
}
```

```
h1{
    font-size: 40px;
```

```
}
```

```
h2{
    font-size: 35px;
```

```
}
```

```
p{
    font-size: 30px;
    line-height: 1.5;
    color: darkblue;
    margin-left: 15%;
    margin-right: 10%;
    text-align: justify;
```

```
}
```

```
footer p{
    color: red;
    font-size: 20px;
```

```
font-family: Arial, Helvetica, sans-serif;
text-align: center;
}
```

```
a{
position: absolute;
top: 20px;
left: 20px;
font-size: 35px;
font-weight: bold;
color: darkblue;
}
```

```
.custom-post-button{
display: inline-block;
padding: 20px 40px;
background-color: #00e35c;
color: white;
border: none;
border-radius: 99px;
cursor: url('fountain pen handwriting.cur'), pointer;
font-family: 'WritingFont', sans-serif;
font-size: 28px;
font-weight: bold;
text-align: center;
}
```

```
.custom-post-button:hover{
background-color: #00a135;
}
```


Connection_db.php

```
<?php

# Connecting Database to website

$con = mysqli_connect("localhost", "root", "", "journaldb");

if ($con-> connect_error)

{

    die("Connection failed:" . $con-> connection_error);

}

?>
```

Delete_entry.php

```
<?php

#this file handles the deletion of a specific journal entry

#based on the JournalID and UserID.

#The user is then redirected to the home page again

#Links to connection_db.php, home.php


require_once('connection_db.php');

session_start();

if(isset($_SESSION['User'])){

}

else{

    #cant do this action unless logged in(redirect to login page)

    header('location:login.php');

}


if($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['journal_id'])){

    $journalID = $_POST['journal_id'];
```

```

#fetch userID of the sessioned user

$username = $_SESSION['User'];

$query = "SELECT UserID FROM users WHERE Username =?";

$stat = $con->prepare($query);

$stat->bind_param("s", $username);

$stat->execute();

$result = $stat->get_result();


if($result->num_rows ==1){

    $row = $result->fetch_assoc();

    $userID = $row['UserID'];


    #Delete journal entry only if it is that specific user's

    #Extra protection

    $deleteQuery = "DELETE FROM journaldb.journalentries WHERE journalID = ? AND UserID = ?";

    $delstat = $con->prepare($deleteQuery);

    $delstat->bind_param("ii", $journalID, $userID);

    $delstat->execute();


    #after deleting, redirect to home page

    header('Location: home.php');

    exit();

}

else{

    echo "User not found";

    exit();

}

}

else{

```

```
    echo "Invalid request";  
    exit();  
}  
?>
```

Home.php

```
<?php  
  
#This is the home page of th user. It is unique to each user as the  
#journal entries of the specific logged in user are displayed  
#Links to viewentry.php  
#echo 'welcome to home';  
require_once('connection_db.php');  
session_start();  
  
#can only access this if logged in  
if(isset($_SESSION['User'])){  
  
}  
else{  
  
    #cant access this page unless logged in(redirect to login page)  
    header('location:login.php');  
  
    #add popup message explaing to user***  
}  
  
#get username from session  
if(isset($_SESSION['User']))  
{  
    // obtaining username if user logged in or user registered  
    if(isset($_SESSION['Username']))  
    {  
        $username = $_SESSION['Username'];  
    }  
    elseif(isset($_SESSION['User']))
```

```

{
    $username = $_SESSION['User'];
}

}

#fetch userID from the database using session username
$query = "SELECT UserID FROM users WHERE Username = ?";
$stat = $con->prepare($query);
$stat->bind_param("s", $username);
$stat->execute();
$result = $stat->get_result();

if($result->num_rows == 1){
    #get userID from the result
    $row = $result->fetch_assoc();
    $userID = $row['UserID'];
}
else{
    echo "User not found";
    exit;
}

#fetching journal entries for the user
$jquery = "SELECT JournalID, Title, Entry_Date, Entry_Text, Entry_Image FROM journaldb.journal
entries
    WHERE UserID = ?";

$stat = $con->prepare($jquery);
$stat->bind_param("i", $userID);

```

```
$stat->execute();
```

```
$jentries = $stat->get_result();
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title><?php echo htmlspecialchars($username); ?>'s Journal</title>
```

```
<link rel="stylesheet" href="home_style.css">
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<h1><?php echo htmlspecialchars($username); ?>'s Journal</h1><br>
```

```
<div>
```

```
<a href="logout.php?logout" class="logout-link">Logout</a>
```

```
<a href="add_entry.php" class="floating-button">Add a new entry</a>
```

```
</div>
```

```
</header><br>
```

```
<div class="card-container">
```

```
<?php
```

```
if ($jentries->num_rows > 0){
```

```

#loop through each journal entry and display it as a card

while($entry = $jentries->fetch_assoc()){

    ?>

    <div class="card" onclick="window.location.href='view_entry.php?id=?php echo htmlspecialchars($entry['JournalID']); ?>'">

        <div class="card-content">

            <p class="card-title">

                <?php echo htmlspecialchars($entry['Title']); ?>

            </p>

            <p class="card-date">

                <?php echo htmlspecialchars($entry['Entry_Date']); ?>

            </p>

        </div>

        <form method="post" action="delete_entry.php" onsubmit="return confirm('Are you sure you
want to delete this entry?');">

            <input type="hidden" name="journal_id" value="<?php echo htmlspecialchars($entry['Jo
urnalID']); ?>">

            <button type="submit" class="delete-button">Delete</button>

        </form>

    </div>

    <?php

}

}

else{

    echo "<p>No journal entries found</p>";

}

```

?>

</div>

<footer>

<p>

All cursor, logo, font assets used in this website are copyright free.

Website created by Zoe Fabre-Anderson 2024

</p>

</footer>

</body>

</html>

Login_style.css

@font-face{

font-family: 'WritingFont';

src: url(Handwriting-poxD.ttf) format('truetype');

font-weight: normal;

font-style: normal;

color: grey;

}

body{

margin: 0;

padding: 0;

font-family: 'WritingFont', sans-serif;

background-color: #F5CAC3;

color: grey;

}

```
h1{  
  font-family: 'WritingFont', serif;  
}
```

```
/*styling of the outer container to center the content*/
```

```
#content {  
  width: 50%;  
  margin: 50px auto;  
  padding: 40px;  
  background-color: #A2C3A4;  
  border-radius: 10px;  
  box-shadow: 0 0 15px rgba(0, 0, 0, 0.2);  
  text-align: center;  
}
```

```
/*Logo image styling*/
```

```
#logo {  
  width: 100%;  
  height: auto;  
  margin-bottom: 20px;  
  /*max-width: 1000px;*/  
}
```

```
/*Styling for the thick left and right margin
```

```
body::before, body::after{  
  content: "";  
  display: block;  
  position: fixed;  
  top: 0;  
  bottom:0;
```



```
width: 20%;  
background-color: #F5CAC3;  
z-index: -1;  
}
```

```
body::before{  
  left:0;  
}
```

```
body::after{  
  right:0;  
}
```

```
*/
```

```
/*styling form elements*/
```

```
form{  
  text-align: center;  
}
```

```
form label{  
  font-size: 1.5em;  
  color: #333;  
  
}
```

```
form input[type="text"], form input[type="password"]{  
  width: 40%;  
  padding: 10px;  
  margin: 10px 0;  
  border: 1px solid #ccc;  
  border-radius: 5px;  
  cursor: url('fountain pen handwriting.cur'), pointer;
```

```
}
```

```
form button{  
    background-color: #7871AA;  
    color: white;  
    padding: 20px 40px;  
    font-size: 1.1em;  
    border: none;  
    border-radius: 8px;  
    text-transform: uppercase; /* Optional: For button text in uppercase */  
    font-weight: bold; /* Optional: For bold text */  
    cursor: url('fountain pen handwriting.cur'), pointer;  
    font-family: 'WritingFont', serif;  
}
```

```
form button:hover{  
    background-color: 357ABD;  
}
```

```
a{  
    color: white;  
    text-decoration: none;  
    cursor: url('fountain pen handwriting.cur'), pointer;  
    font-size: 1.2em;  
}  
a:hover{  
    text-decoration: underline;  
}
```

```
footer {  
    font-size: 1em;
```

```
color: red;

font-family: 'WritingFont', sans-serif;
}

body{ /*custom cursor*/

    cursor: url('fountain pen handwriting.cur'), auto;

}
```

Login.php

```
<?php

#User can log in if they already have an account


#Connecting to database
$con = mysqli_connect("localhost", "root", "", "journaldb");

if($con-> connect_error){

    die("Connection failed:" . $con-> connection_error);
}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login</title>

    <link rel="stylesheet" href="login_style.css">


</head>

<body>
```

```
<div id="content">
```

```
<!--logo image-->
```

```

```

```
<h1>Login To Unlock Your Secret Journal!</h1>
```

```
<top></top>
```

```
<form method="post" action="processing_input.php">
```

```
<label>Username</label><br>
```

```
<input type="text" name="Username"><br>
```

```
<label>Password</label><br>
```

```
<input type="password" name="Password"><br>
```

```
<button id="button" name="Login">Login</button><br><br>
```

```
<?php
```

```
if (isset($_GET['Empty'])) {
```

```
    echo "<p style='color:red'>" . $_GET['Empty'] . "</p>";
```

```
} elseif (isset($_GET['Invalid'])) {
```

```
    echo "<p style='color:red'>" . $_GET['Invalid'] . "</p>";
```

```
}
```

```
?>
```

```
<a href="signup_input.php">Don't have an account? Click here to Sign up today</a><br>
```

```
</form>
```

```
<bottom></bottom>
```

```
<footer>
```

```
<p>
```

All cursor, logo, font assets used in this website are copyright free.

Website created by Zoe Fabre-Anderson 2024

```
</p>
```

```
</footer>
```

```
</div>
```

```
</body>
```

```
</html>
```

Logout.php

```
<?php
```

```
#logs the user out
```

```
#Links to login.php
```

```
session_start();
```

```
if(isset($_GET['logout'])){
```

```
    session_destroy();
```

```
    header('location:login.php');
```

```
}
```

```
?>
```

Processing_input.php

```
<?php
```

```
#This file processes the login information that the user inputs
```

```
#links to connection_db.php and home.php
```

```
require_once('connection_db.php');
```

```
session_start();
```

```
if(isset($_POST['Login'])){

    if(empty($_POST['Username']) || empty($_POST['Password'])){

        header("location:login.php?Empty=Username and or Password is empty");

    }

    else{

        $query = "SELECT * FROM journaldb.users WHERE Username='".$_POST['Username']."' AND
Password='".$_POST['Password']."'";

        $result = mysqli_query($con, $query);

        #if credentials are correct, make a session
        if(mysqli_fetch_assoc($result)){

            $_SESSION['User'] = $_POST['Username'];

            header('location:home.php');

        }

        else{

            #if credentials are incorrect

            header("location:login.php?Invalid= Please enter correct username and password");

        }

    }

}

else{

    echo 'Error: processing_input.php not working';

}

?>
```

Signup_input.php

```
<?php
```

```
# This file collects the user's sign-up information and sends it
```

```
# to the signup.php page for error checking and user creation.
```

```
# Linked with: signup.php, connection_db.php, login.php
```

```
session_start();
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Sign Up</title>
```

```
  <link rel="stylesheet" href="login_style.css">
```

```
</head>
```

```
<body>
```

```
  <div id="content">
```

```
    
```

```
    <h1>Sign Up</h1>
```

```
    <form method="post" action="signup.php">
```

```
      <!-- Error handling for empty fields -->
```

```
      <?php if(isset($_GET['Empty'])): ?>
```

```
        <p style="color:red"><?php echo $_GET['Empty']; ?></p>
```

```
      <?php endif; ?>
```

```
<!-- Error handling for password mismatch -->
```

```
<?php if(isset($_GET['Invalid'])): ?>
```

```
    <p style="color:red"><?php echo $_GET['Invalid']; ?></p>
```

```
<?php endif; ?>
```

```
<!-- Error handling for existing username -->
```

```
<?php if(isset($_GET['User'])): ?>
```

```
    <p style="color:red"><?php echo $_GET['User']; ?></p>
```

```
<?php endif; ?>
```

```
<!-- Form Fields -->
```

```
<label>Username</label><br>
```

```
<input type="text" name="Username" value="<?php echo isset($_SESSION['Username']) ? $_SESSION['Username'] : ''; ?>"><br>
```

```
<label>Password</label><br>
```

```
<input type="password" name="Password"><br>
```

```
<label>Re-Type Password</label><br>
```

```
<input type="password" name="Re_Password"><br>
```

```
<button id="button" type="submit" name="Register">Create Account</button><br><br>
```

```
<a href="login.php">Already have an account? Login here!</a><br>
```

```
</form>
```

```
<footer>
```

```
<p>
```

All cursor, logo, font assets used in this website are copyright free.

Website created by Zoe Fabre-Anderson 2024


```
</p>
</footer>
</div>
```

```
</body>
</html>
```

Signup.php

```
<?php

# This page processes the user input upon clicking the signup button
# It checks if fields are empty and if the passwords match.
# Linked to: connection_db.php, home.php

require_once('connection_db.php');
session_start();

if(isset($_POST['Register'])){

    $send_username = $_POST['Username'];
    $send_password = $_POST['Password'];
    $re_password = $_POST['Re_Password'];

    # Store the username in session to persist form data
    $_SESSION['Username'] = $send_username;

    # Check if any fields are empty
    if (empty($send_username) || empty($send_password) || empty($re_password)) {
        header('location:signup_input.php?Empty=Please fill all the fields');
    }
    else {

        # Check if passwords match
        if ($send_password === $re_password) {
```

```

# Check if the user already exists

$query = mysqli_query($con, "SELECT * FROM journaldb.users WHERE Username='$send_username'");

if (mysqli_num_rows($query) > 0) {
    header('location:signup_input.php?User=That username is already taken!');
}
else {
    # Prepare and execute the SQL statement to insert the new user

    $sql = "INSERT INTO journaldb.users (Username, Password) VALUES ('$send_username', '$send_password')";

    # Check if the query was successful
    if ($con->query($sql) === TRUE) {
        $_SESSION['User'] = $send_username;

        header('location:home.php'); # Redirect to home page after successful registration
    }
    else {
        echo "Error: " . $sql . "<br>" . $con->error;
    }
}
}
else {
    # If passwords don't match

    header('location:signup_input.php?Invalid=Re-typed password does not match');
}
}
}
else {
    echo 'Error: Not Working';
}
?>

```

View_entry.php

```
<?php
```

```
#This file displays the information of the specific journal entry
```

```
require_once('connection_db.php');
```

```
session_start();
```

```
$journalID = intval($_GET['id']);
```

```
#fetch journal entries
```

```
$query = "SELECT Title, Entry_Date, Entry_Text, Entry_Image FROM journaldb.journalentries WHERE JournalID = ?";
```

```
$stat = $con->prepare($query);
```

```
$stat->bind_param("i", $journalID);
```

```
$stat->execute();
```

```
$result = $stat->get_result();
```

```
if($result->num_rows == 1){
```

```
    $entry = $result->fetch_assoc();
```

```
}
```

```
else{
```

```
    echo "Entry not found";
```

```
    exit;
```

```
}
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Journal Entry Details</title>

<link rel="stylesheet" href="view_style.css">

</head>

<body>

<header>

  <a href="home.php">Back to Home</a>

  <h1><strong></strong> <?php echo htmlspecialchars($entry['Entry_Date']); ?></h1>

  <h2><strong>Tile:</strong> <?php echo htmlspecialchars($entry['Title']); ?></h2>

</header><br>

<p><strong>Today...</strong> <?php echo nl2br(htmlspecialchars($entry['Entry_Text'])); ?></p>
<br>

<?php if (!empty($entry['Entry_Image'])): ?>

  <?php else: ?>

    <p>No image available</p>

  <?php endif; ?>

<footer>

<p>

  All cursor, logo, font assets used in this website are copyright free.

  Website created by Zoe Fabre-Anderson 2024

</p>
```

```
</footer>
```

```
</body>
```

```
</html>
```

View_image.php

```
<?php
```

```
#to display the BLOB image from the database for each journal entry
```

```
require_once('connection_db.php');
```

```
if (isset($_GET['id'])) {
```

```
    $journalID = $_GET['id'];
```

```
    #fetch image data
```

```
    $query = "SELECT Entry_Image, Image_Extension FROM journaldb.journalentries WHERE JournalID = ?";
```

```
    $stat = $con->prepare($query);
```

```
    $stat->bind_param("i", $journalID);
```

```
    $stat->execute();
```

```
    $result = $stat->get_result();
```

```
    if ($result->num_rows == 1) {
```

```
        $row = $result->fetch_assoc();
```

```
        $imageData = $row['Entry_Image'];
```

```
        $imageExtension = $row['Image_Extension'];
```

```
        if ($imageData) {
```

```
            #determine the mime type based on the file content
```

```
            $mimeType = "";
```

```
            switch(strtolower($imageExtension)){
```

```

        case 'jpg':
        case 'jpeg':
            $mimeType = 'image/jpeg';
            break;
        case 'png':
            $mimeType = 'image/png';
            break;
        case 'gif':
            $mimeType = 'image/gif';
            break;
        default:
            #default to binary stream if extension is unknown
            $mimeType = 'application/octet-stream';
    }

    header("Content-Type: $mimeType");
    echo $imageData;
} else {
    echo "No image available";
}
} else {
    echo "Image not found";
}
} else {
    echo "No ID specified";
}
?>

```

View_style.css

```

@font-face{
    font-family: 'WritingFont';
    src: url(Handwriting-poxD.ttf) format('truetype');
}

```

```
font-weight: normal;  
font-style: normal;  
}
```

```
body{  
  margin: 0;  
  padding: 0;  
  font-family: 'WritingFont', sans-serif;  
  background-color: rgba(120, 113, 170, 1.0);  
  background-image: url('paper-texture.jpg');  
  background-size: cover;  
  background-position: center;  
  background-repeat: no-repeat;  
  background-blend-mode: overlay;  
  text-align: center;  
  cursor: url('fountain pen handwriting.cur'), auto;  
  
}
```

```
header{  
  color: darkblue;  
  
}
```

```
h1{  
  font-size: 40px;  
}  
h2{  
  font-size: 35px;  
}
```

```
p{  
  font-size: 30px;  
  line-height: 1.5;  
  color: darkblue;  
  margin-left: 15%;  
  margin-right: 10%;  
  text-align: justify;  
}
```

```
footer p{  
  color: red;  
  font-size: 20px;  
  font-family: Arial, Helvetica, sans-serif;  
  text-align: center;  
}
```

```
a{  
  position: absolute;  
  top: 20px;  
  left: 20px;  
  font-size: 35px;  
  font-weight: bold;  
  color: darkblue;  
}
```