**SRN: 19036903**

**UNIVERSITY OF HERTFORDSHIRE**

**School of Computer Science**

**Modular BSc Honours in Computer Science**

**6COM1056 - Software Engineering Project**

**Final Report**

**June 2025**

**TITLE OF PROJECT**

**Author's First name and surname**

**Mary**

**Supervised by: Supervisor's Name**

**Yury**

Contents

[Abstract 3](#_Toc202213990)

[CHAPTER 1 3](#_Toc202213991)

[Introduction 3](#_Toc202213992)

[Project Aim 4](#_Toc202213993)

[Motivations 4](#_Toc202213994)

[Risk Assessment 5](#_Toc202213995)

[Chapter 2- Research 6](#_Toc202213996)

[Chapter 3- Design and Development 9](#_Toc202213997)

[Chapter 4: Implementation 18](#_Toc202213998)

[Chapter 5: Testing 27](#_Toc202213999)

[Chapter 6: Evaluation 29](#_Toc202214000)

Acknowledgements

Psalms 3:5

"Trust in the Lord with all your heart and lean not on your own understanding."  
I would like to thank my friends, my family and everyone who listened to me rant, cry and breakdown over this dissertation. It was an incredibly difficult body of work for me to do, and it created such a learning experience for me. Despite how many times I doubted my capabilities. I would lastly like to thank God, who was such a crutch for me to rely on and remind myself that I can do it.

# Abstract

This project is the implementation of a website that allows students who are experiencing challenges with their mental health. To access a website through their student email. In which they’re able to write about the emotions their experiencing

This project seeks to provide solutions for students to express their thoughts and feelings by providing a website that allows journal entries for students to make. They’re able to submit to the student's well-being, and it provides another alternative for students to start addressing their mental health without having to have a physical meeting or phone call. As it may be too overwhelming for the student to do in person.

Loneliness and isolation are common struggles that many university students go through. As for many students, it's their first time being away from their family and friends whilst trying to acclimate to a new environment altogether. Creating an adjustment period in which some students struggle more than others.

# CHAPTER 1

## Introduction

mental health is an ever-growing topic in today’s fast-paced world, especially among young people and students, typically university students. A website that can help in addressing this issue can help provide users with a safe, supportive and simple space to reflect on their mental health and potentially reach out for help.

The website intends to offer basic functionality such as account registration, secure login using passwords, and a personal area where users can create and manage their diary entries. The goal is to encourage self-reflection, promote awareness, and give users a private place to process their thoughts. The design focuses on accessibility, ease of use and a calm, welcoming interface.

## Project Aim

This project seeks to provide solutions for students to express their thoughts and feelings by providing a website that allows students to make entries. They’re able to submit to the student's well-being, and it provides another alternative for students to start addressing their mental health without having to have a physical meeting or phone call. As it may be too overwhelming for the student to do in person.

## Motivations

My main motivation for this project is to help address mental health challenges like stress, anxiety and depression. Many students struggle with this but hesitate to seek traditional support due to fear, overwhelming emotions, and logistical factors such as wait times for counselling sessions. This project provides a private, accessible and non-intimidating digital platform for students to express their thoughts and emotions through journaling. As someone who has struggled with mental health heavily whilst in university. Seeking help can become very daunting due to the fear of how one would be perceived. A study from the UK Parliament (UK Parliament Briefing Paper on Student Mental Health, 2021) reported a 1% to 5.7% increase in the mental health condition of home students in university. Another study showed that students had a hard time adjusting to a university setting/environment. The data showed a significant increase in mental health illnesses, such as depression, anxiety, among students during their university years. With the sharpest increase happening in their first year (Bewick et al, 2010).

Further motivation came from previous struggles with reaching out for help. As I was overwhelmed with depressive thoughts, I was unable to ask for and receive help due to the isolation I had put myself in. A low-pressure website provides an alternative for students to process their emotions at a pace ready for them. Studies have shown that expressive writing, such as journaling, has helped reduce distress, improve emotional regulation (Pennebaker & Smyth). The integration of journaling with an optional connection to the student well-being services. The website acts as a bridge between self-help (Journaling) and professional support (Student Well-Being). This may help strengthen students to actively work towards improving their mental health in a way that feels manageable and safe.

Another reason is the desire to push myself by focusing on web development. I’ve enjoyed modules focusing on web development, such as the accessibility and usability module. I want to increase my knowledge in this area as it is something that interests me and something I would like to consider for my future. Focusing on this area will allow me to explore many functionalities and implement different programming languages.

## Risk Assessment

A risk assessment is carried out to show the potential risks that could affect this project. by making a risk assessment, you can then mitigate and minimise any potential risk. The severity level is ranked from 1 – 10, with 1 being the lowest severity and 10 being the highest severity.

## Tabel 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Description** | **Severity** | **Mitigation** |
| 1. Illness | Physical illnesses may impact my ability to carry out my work. Such as low energy levels, which negatively influence my productivity | 4 | There would be days when I wouldn’t be able to carry out the work. So, I would have to work around it |
| 1. Managing workload | Because I am doing this project alongside my other modules. This may act as a challenge | 7 | Because I am doing this project alongside my other modules. This may act as a challenge, especially for myself, as I’m prone to overwhelming thoughts that can affect my ability to manage my workload. To mitigate and lessen this, I can practise grounding techniques and reach out for help from the student support team |
| 1. Inadequate programming skills | As this project consists of web development with languages like JavaScript, HTML, | 6 | Web development is not my strongest skill, so I may face some difficulty in navigating this. However, there is a multitude of resources where I can receive help from |
| 1. Losing my work | My laptop may malfunction or shut down, causing a loss of work | 6 | I have considered version control by frequently saving work on a portable hard drive to prevent losing work. |

## Project Scope

The main scope of the project is to develop a functioning and easy-to-use website. Where students struggling with their mental health can make entries about it. The website aims to be easy to use and functional. Security isn’t within the scope of the project, but will be considered on user security, like user passwords and how the data is managed. For example, the website should only be accessible to university students.

# Chapter 2- Research

### 2.0 Background Research

*Mental health in Higher education (University )*

Mental health issues among university students have become a growing concern, with studies indicating an increase in stress, anxiety, and depression. Studies have shown that over 60% of students have been experiencing an overwhelming amount of anxiety. Over 40% of American university students have reported severe depression that has impacted their ability to engage with their academics (Eisenberg et al., 2019)

Despite the availability of mental health services, many students struggle to seek help due to stigma, privacy concerns, or logistical barriers. For example, many university students avoid seeking mental health support due to the fear of how others will think of them. Eisenberg et al found that 62% of students avoided seeking mental health support due to the fear of how they’d be perceived. A study from the UK Parliament (UK Parliament, 20/21) reported a 1% to 5.7% increase in the mental health condition of home students in university. Which is only heightened by being in competitive environments such as a university.

In the UK, to combat this, digital platforms have been discussed as an alternative; however, concerns about security and accessibility are brought up. University students are more likely to be victims of cyberattacks.

*Security and privacy*

Security and privacy are necessary when considering modern web development. With an increase in personal data being shared online, especially when sensitive information like mental health is shared. Protecting user information is imperative for developers.

This is particularly important for university students, university students typically use public networks, found in lecture halls, libraries, etc and are less likely to have good password practices, such as using different passwords for each site they use, and overall weak passwords. Broadhurst et al. (2020) found that first-year and international students were more prone to cyber fraud techniques like deceptive emails.  
  
Cyber-attacks are common among university students due to the sheer amount of private and sensitive information collected from students. Such as their academics, personal contact data, including family and friends, medical records and more.

*Journaling*

Journaling provides great benefits for mental health as it provides a safe and adaptable outlet for emotional expression and a space for more self-reflection. Journaling is the structured or unstructured written practice where participants record their emotions, experiences and reflections. The purpose of this is for self-expression, emotional processing and personal growth (Pennebaker & Seagal, 1999). Journaling allows for constant recording of personal thoughts and feelings and can take many different forms.

In 2021, a study from Koziol et al reported that 94.74% of participants found that journaling had been beneficial to their mental health, many had reported that they felt as though their anxiety had reduced and had feelings of relief and calm. From this study, Koziol et al found that when journaling is used as a way to document day-to-day life, it helps in processing emotions and navigating life challenges in a healthy and positive way. Further research found that structured journaling, called ‘resource diaries’ helped improve emotional regulation and the intensity of depressive episodes. The research was further conducted by Suhr et al reported that ‘resource diaries’ showed a decrease in depression. These diaries allowed for better emotional stability, where the participants were able to reframe their negative thoughts to something much more positive and affirming.

# Chapter 3- Design and Development

## Design and Development

This part will discuss the design and development of this project

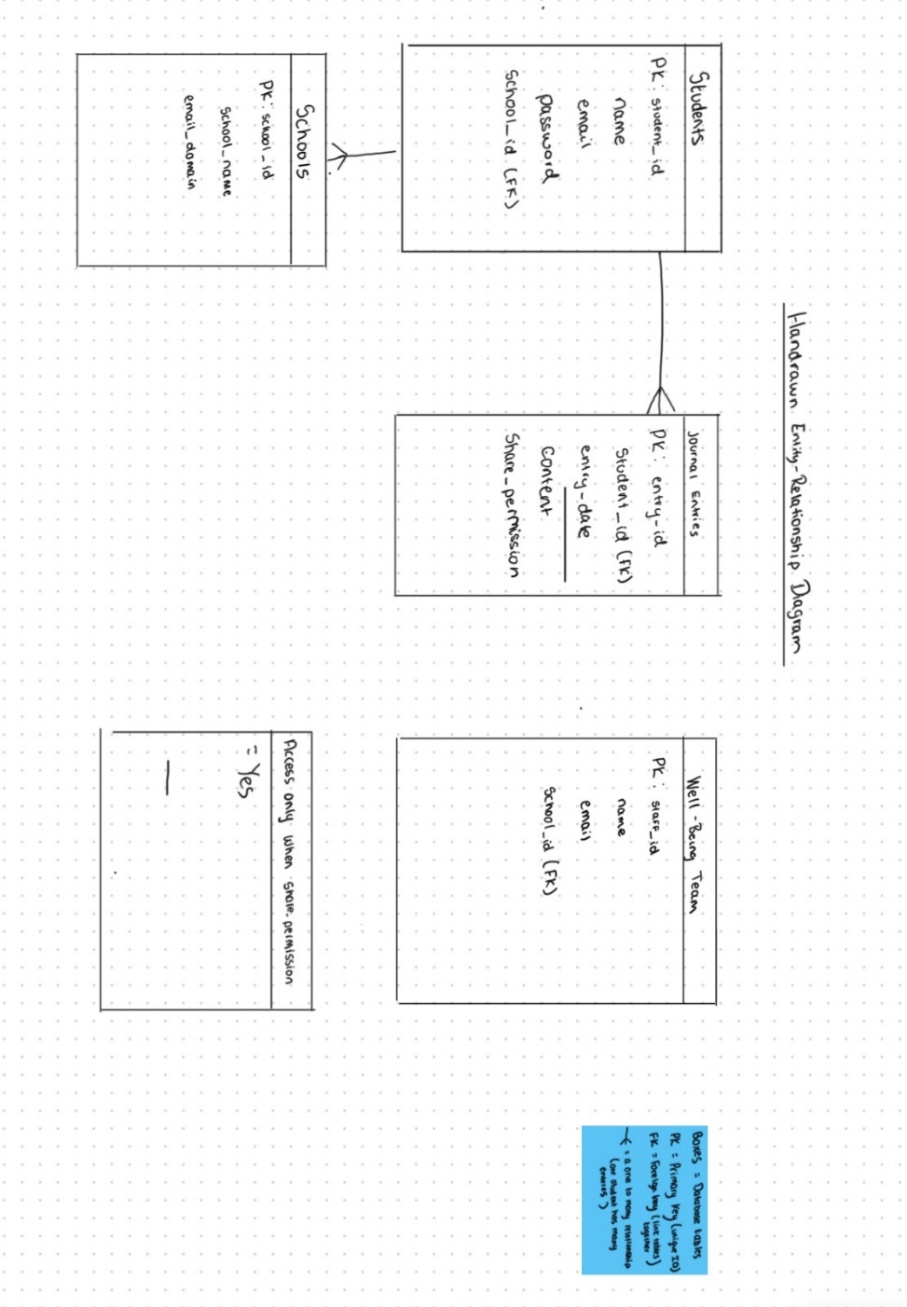
## 3.1 Entity-Relationship Diagram

For this project, a university-based mental health journaling platform, careful considerations must be made to ensure functionality within the database. An ERD (Entity-Relationship Diagram) will be used; the ERD acts as a base. As the ERD maps out how the data will be stored, accessed and managed

The ERD for this website is designed to allow students to ensure and maintain their privacy while writing their journal entries, whilst also having the choice to share their entries with the university's student well-being. Below are the attributes of the diagram and a hand-drawn version of the diagram.

1. **Students**
   * student\_id (primary key)
   * name
   * email (school email only)
   * password
   * school\_id (which university they're from)
2. **Journal Entries**
   * entry\_id (primary key)
   * student\_id (who wrote it)
   * entry\_date
   * content
   * share\_permission (YES/NO) ← This controls if the well-being team can see it
3. **Well-being Team**
   * staff\_id (primary key)
   * name
   * email
   * school\_id (which university they work for)

image 1:

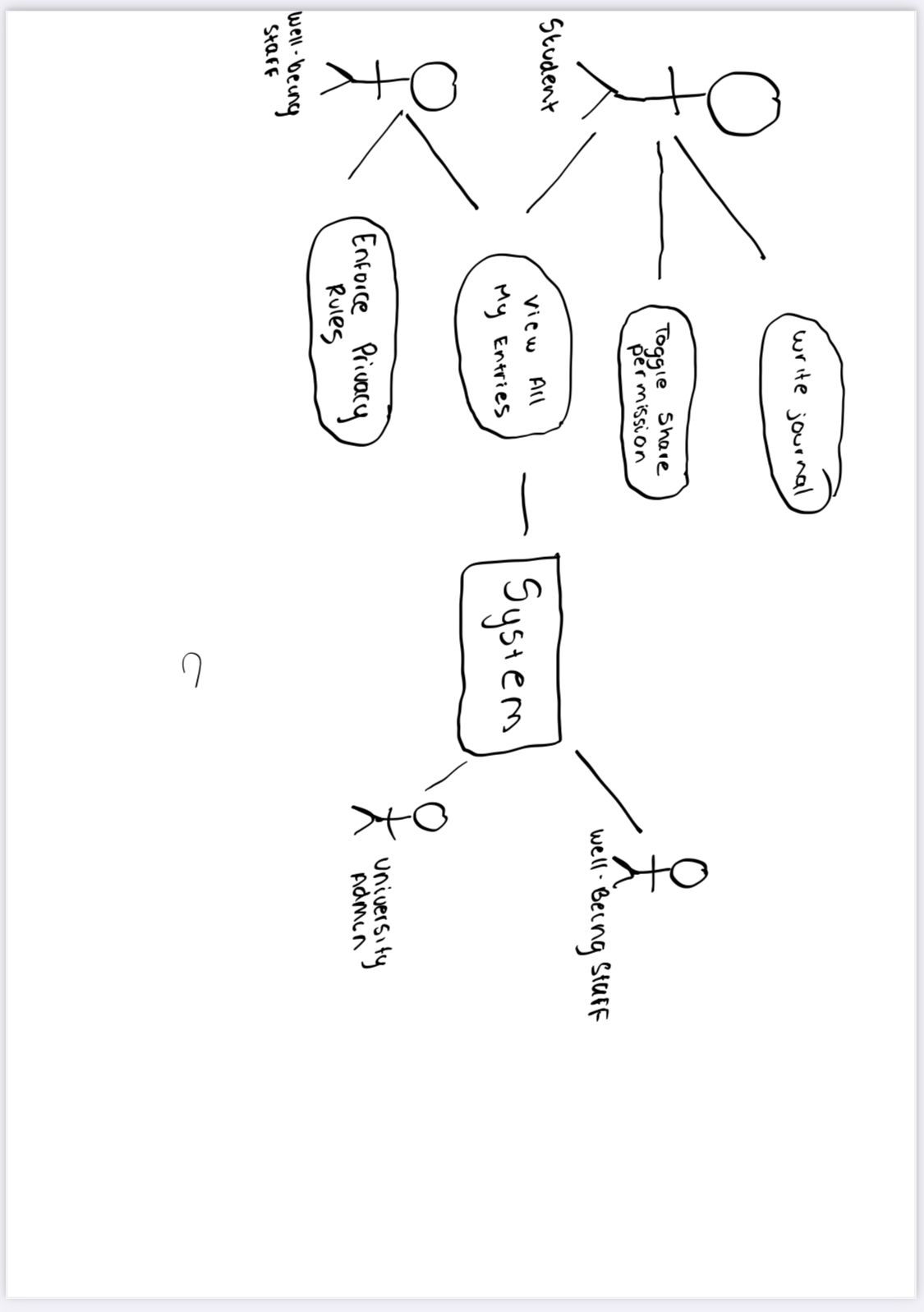


## 3.2 Use Case Diagram

A use case diagram is the graphic representation of the interaction between the elements of a system and a user. Developed by Ivar Jacobson in 1986, a use case diagram has also been considered for the project, as it is a methodology used in system analysis to identify, clarify and organise system requirements. It allows the focus to be on how the user interacts with the system. The use case focuses on human actions as it prioritises users' needs. For example, students being able to write their private journal entries and the choice of control sharing. The well-being staff being able to access students' entries with consent from the students. This must be done without having complex interfaces.

A use case diagram consists of these core components:

* Actors: typically represented by stick figures who represent the individuals within the system according to their roles
* Use cases: the specific role played by the actors within the system. They are typically represented by an oval shape
* Relationship: are the lines and arrows that connect the actor to the use case. It represents the relationship and interaction between each other
* System boundary: is the rectangle that encloses the use cases, it defines the system and the world around it



## Image 2 displays the use case diagram

## 3.4 Web Accessibility

Web accessibility is an important feature to consider and implement as it ensures that the website is accessible, allowing users to navigate and interact freely and easily. Aiding in the overall satisfaction of the website. Online, many guidelines help make sure that a website is accessible. The main and most popular one is called the WCAG 2.0. Also known as the Web Content Accessibility Guidelines. There are four main principles that the WCAG adhere to, which are:

* **P**erceivable – means that the user must be able to perceive all the information being presented. It asks the question of whether the information presented, and the interface can be available in multiple forms. It means that the information in the content must appeal to at least one of the user’s senses. As the information needs to be presented in a way that each user can use, regardless of which senses they may or may not have.
* **O**perable- relates to the user interface components and how navigation works with other input methods. It means that the user interface and navigation aspect of the website needs to be accessible to everyone. As the interface needs to be inclusive and consider those not able to use a standard keyboard and mouse.
* **U**nderstandable- means that the website should be formatted in a way that the  
  information is presented clearly to all users. it means that the information, the navigation, and the tools used on the website must be easily understood.
* **R**obust- means that the website must be able to be supported on multiple platforms and different devices, meaning that despite the changing of technology and devices, the user may use it. The website must remain intact and be easily accessed

The POUR principles act as an easy-to-use, practical, and ethical guide for building websites and other digital platforms. By making sure the website is perceivable, operable, understandable and robust. Using the POUR principles makes sure that the website is accessible for all types of users.

## 3.5 Nielsen’s Heuristics

Nielsen’s Heuristics are a broad set of applicable design principles developed by Jakob Nielsen. They act as a practical checklist for evaluating the user interface without testing. Nielsen’s heuristic consists of 10 principles, which will be considered for this website. Nielsen's heuristics will help in evaluating the usability of this website, both in the design and implementation stages. Using Nielsen’s heuristics helps reduce the likelihood of problems arising with this website. As it can address user control, flexibility and efficient use.

The 10 heuristics are:

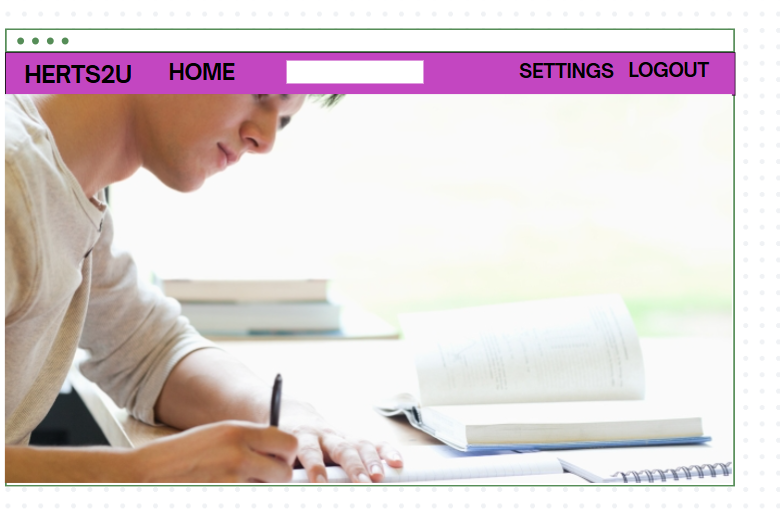
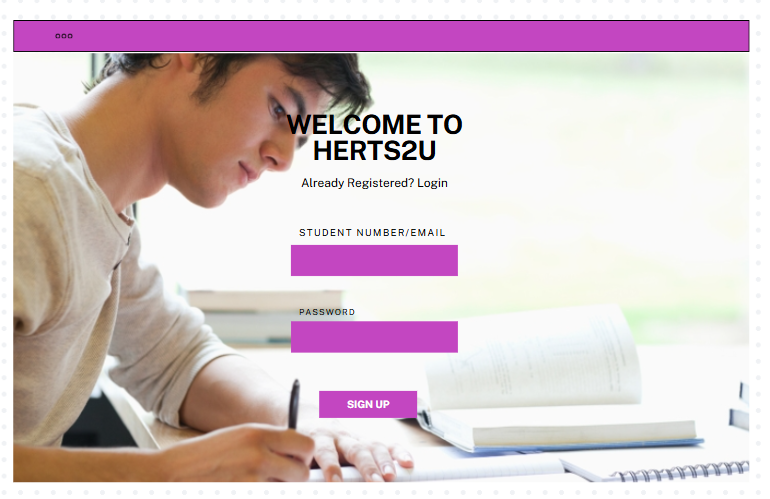
1. ‘Visibility of system status’: explaining that users should be informed about what is happening. Feedback is given in a reasonable time frame. The idea behind this is that users are aware of the current system, they know what to expect, creating trust in the product.
2. ‘Match between system and the real world’: is essentially saying that the design should speak the language of the users. meaning the site should have words, icons and concepts that feel familiar to the users.
3. ‘User control & freedom’: meaning that users should be able to go back, amend mistakes without being stuck. Allowing users to retain control of the system and avoid getting stuck.
4. ‘Consistency and Standards’: users expect that design, wording and functionality should be the same throughout the website to avoid confusion.
5. ‘Error and prevention’: meaning that the website should be designed in a way to avoid errors happening in the first place. It calls for getting rid of mistakes that are prone to happen.
6. ‘Recognition rather than recall’: means that users aren’t supposed to remember everything when using the website. the information needed to interact with the website, such as labels.
7. ‘Flexibility and efficiency of use’: meaning that users of different experience levels should be able to navigate the website with ease. So, the website should be able to accommodate all types of users.
8. ‘Aesthetic and minimalist design’: The design of the website should be simplistic, with the essential content and functionality. This ensures that the visual aspect of the website supports users’ primary goals.
9. ‘Help users recognise, diagnose, and recover from errors’: meaning that error messages should be addressed plainly, guiding users to fix them.
10. ‘Help and documentation’: meaning that content that should help should be easy to find and understand, especially when users need help and guidance.

## 3.6 Wireframe

A wireframe is the visual representation of how the webpage might look. It’s also commonly referred to as the blueprint or the schema of a project. A wireframe exists to outline basic structure and components. A wireframe was used to plan the interface and what it should look like

## 3.7 High-fidelity user interface (UI) mock-up

After making wireframes, a high-fidelity user interface was made to get a more accurate visual representation of what this website is to look like. A high-fidelity mock-up is a detailed, more polished representation of what the website should look like. It typically consists of a proper layout with real colours, the type of font used, etc. Below are some mock-ups that were designed using Canva.



## 3.8 Visual design justification

When designing this mock-up, the colour purple was selected to help create a feeling of calmness, creativity and emotional balance. These are all known to help support mental health. Hues like blue, green, and purple and blends amongst these are considered positive (Wilms & Oberfeld, 2018), suggesting a positive emotional response. In addition, Jonauskaite and Mohr (2025) highlight the strong connections between colour and emotion, reinforcing the importance of careful colour selection in designs aimed at mental health support. Using purple helps establish a welcoming, reassuring environment that aligns with the aim of the website.

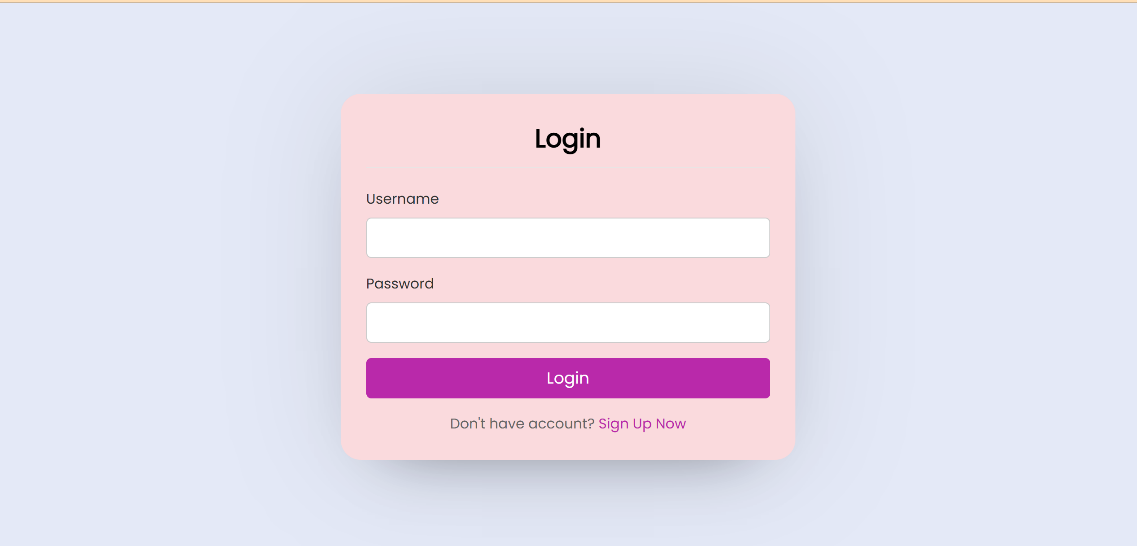
In addition to using purple, a harmonious colour palette has been used to balance and accommodate the purple used. Colour harmony derived from theory refers to the visual and aesthetic rearranging of colours. Colour harmony is defined as an application of colour theory principles that focuses on creating an aesthetically pleasing colour combination. From the Interaction Design Foundation, colour harmony is defined as the “theory of combining colours that is harmonious to the eyes. It creates balance and engagement by using colours that work well together according to the rules of colour theory”

The website design uses a harmonious colour palette of white and muted accents that is highlighted by the colour purple. Colour harmony is needed in visual design as it helps create a sense of balance and order, altering and determining the user experience. The colour harmony of a website can shift the way a user may feel when interacting with the website. With the context of the website being mental health-related, it is important to make sure that the website is comfortable for the users. Colour harmony can be used to help avoid visual fatigue and overstimulation, which could trigger anxiety-inducing feelings. A complementary colour palette that works together can help create a calming and cohesive look. This can invoke a sense of calm and trust, which is important for users, especially university students. In this design, the primary colour is purple, and the more muted secondary colours help ensure that the website remains gentle and appropriate for a platform focused on mental health support.

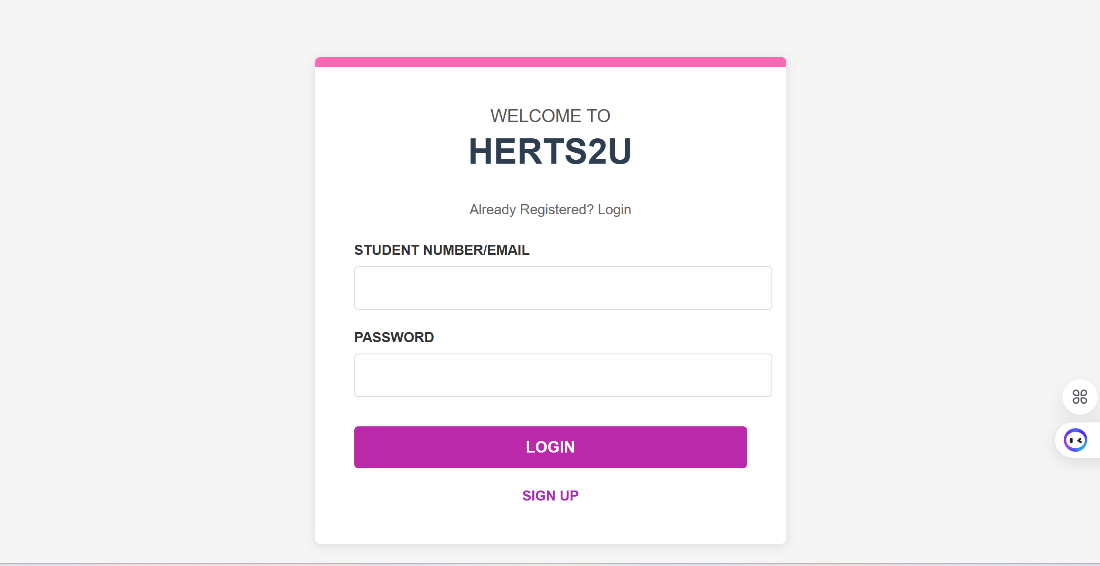
## 3.9 Design Prototypes

Because this is a website, simple design prototypes were made to get an idea of the layout of the website. As it helped visualise the user interface before the development, allowing the focus to shift onto usability and the structure. Prototypes are important in web development as they provide an idea of what to expect once the website is finalised. It helps identify any potential design issues, making the development process more efficient, reducing the need for corrections and major changes later on. Whilst the design prototypes were simplistic in design, they helped in designing the final login page

## Image 5



## Image 6



## 3.10 Information architecture

Without careful attention to information architecture, valuable features may be lost, leading to a poor user experience. Information architecture is the discipline of making information findable and understandable. According to the UX Design Institute, information architecture is about creating a straight-to-the-point navigation structure that helps users find what they’re looking for without getting confused or frustrated (Jayne Hannah, 2023). Coined by Richard Saul Wurman in 1975, a trained architect and graphic designer. This idea was introduced to make complex information presented to people clearer and understandable. Like how an architect breaks down complex building systems into blueprints or schemas. Wurman is interested in how information is gathered and broken down to be understood. As the technological world increased, the rapid rise of the internet was ensured. By the 1990s Morville and Rosenfeld expanded on the concept of information architecture by applying digital environments. Blueprints are made for a digital product (website) layout and navigation, ensuring that the system is simple and understandable. Information architecture will be used to improve usability. A well-structured information architecture will reduce confusion for users by organising the information in a well-structured way. It allows users to locate what they need with ease. For example, a website that emphasises a simple navigation is likely to retain the user.

## 3.11 Programming/Scripting Language

Because this is a website, HTML, CSS and JavaScript will be used in this project. according to Mal S et al (2024), HTML (Hyper Text Markup Language) is the “blueprint of the internet”. HTML is responsible for defining and organising different elements found on the page. These elements include headings, images, paragraphs, etc. HTML forms the foundational structure of every website. HTML acts as the backbone of every language. It tells the browser assistive technologies like screen readers what each element and content provided is. HTML focuses on labelling the elements. For example, <p> in HTML means a paragraph, whilst <h> means a heading. Without HTML, there would be no clarification between the two.

First developed in 1994 by Håkon Wium Lie, CSS, also known as Cascading Style Sheets, is a styling language that controls the visual presentation of a website/web page, such as the colour of the webpage and the typography. Unlike HTML, CSS is focused on the aesthetic and visual part of the web page. CSS allows developers to decide and control how the elements look while keeping the HTML structure. The separation of CSS and HTML allows for flexibility in web design. For example, CSS heavily affects web accessibility. CSS is responsible for affecting features like adjustable text or high contrast colour schemes. This is beneficial to the visually impaired. The separation of CSS and HTML allows for ease of maintenance, as the developer can update one CSS file as opposed to changing the style on every page. Increasing development time and reducing costs due to a minimisation of errors.

First introduced in 1995 by Brendan Eich, JavaScript acts as a high-performing language. Allowing for interaction and dynamic functionality within web pages unlike HTML, which handles the actual content and CSS, which handles the styling of the content. JavaScript controls how a website responds to user actions such as clicking buttons and hovering over some elements. Improving the overall user experience as the website is faster and more dynamic. As a result of this, the use of JavaScript has become integral in web design and development as it has a large influence over how a website looks and feels.

# Chapter 4: Implementation

## Website Implementation:

This section discusses both the front and backend development of this website

## Frontend Development:

At the beginning of the development process, I initially intended to focus on the backend development. However, I found it quite complex at first due to the learning curve, so I started off working on the frontend of this website. allowing me to build the visual aspects of the website first, before gaining confidence to tackle the backend.

The frontend development of the project made use of HTML, CSS, JavaScript and PHP was later used to support the user interaction. This will be discussed further in the backend development section.

* Each page of the website used HTML to establish a basic structure and a layout of the website. HTML was used to define key elements like headings, paragraphs, navigation bars and more. HTML provided the base on which the website was built
* CSS was used to style the HTML and provide a better user experience visually. CSS defines the visual presentation of the website. from its colours, fonts used, overall layout, its spacing and any animations used. To make the website cohesive, a consistent design was kept and maintained throughout all the web pages, sharing classes and variables, improving readability and usability.
* JavaScript added interactivity and dynamic features. For example, it allowed toggle switches, interactive forms and dropdown selections. These features allowed users to interact in real time. Creating a more immersive and engaging experience.

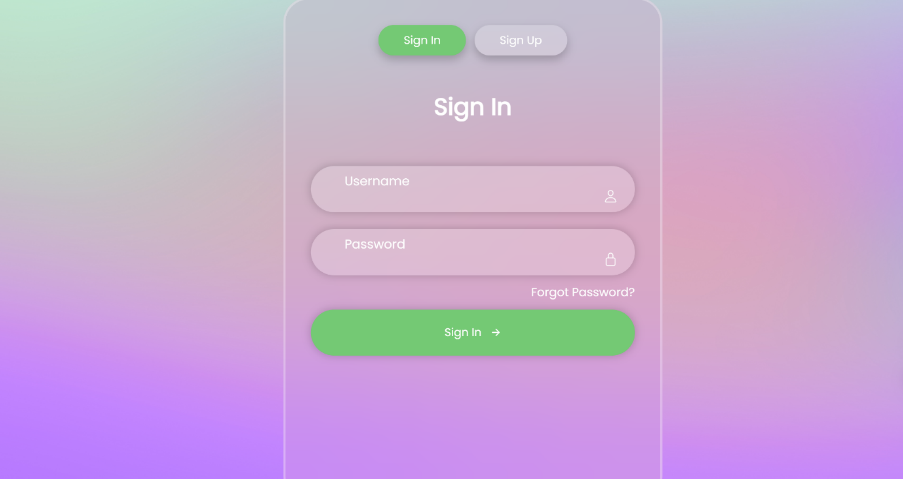
During the frontend development, a few mock-ups and prototypes were made. Allowing for testing and seeing how they look to see if any changes need to be made. This approach made sure that the user interface aligned with accessibility and usability. Once the frontend phase was finalised, it made the backend part a lot easier to incorporate, as the visual aspect was kept intact

**Sign In/Sign Up:**

Image 7 down below shows the final interface of the first webpage created. Displaying how users access the website reflects a more refined application of the visual design principles highlighted in section 3.8. A harmonious array of colours like pale shades of pink, purple, and green, that are informed by psychology to promote calmness and emotional safety. This design aims to support the mental health-focused purpose of the website. As it helps in creating an emotionally sensitive user experience.

The final version of the login/registration page builds upon the earlier prototypes discussed in section 3.9. It maintains the overall theme from those prototypes by expanding on it in a more aesthetically pleasing format, improving the overall execution and user interaction. Rounded buttons, minimalist icons and smooth transitions enhance usability without compromising the calming aesthetic. It also references the WCAG usability guidelines as the colour scheme is soft shades. Rather than bright colours, as they can trigger photosensitive conditions like epilepsy. Resulting in a login/register interface that is fully functional and aligns with the aim of this project, which is essentially providing a welcoming digital space for mental health journaling.

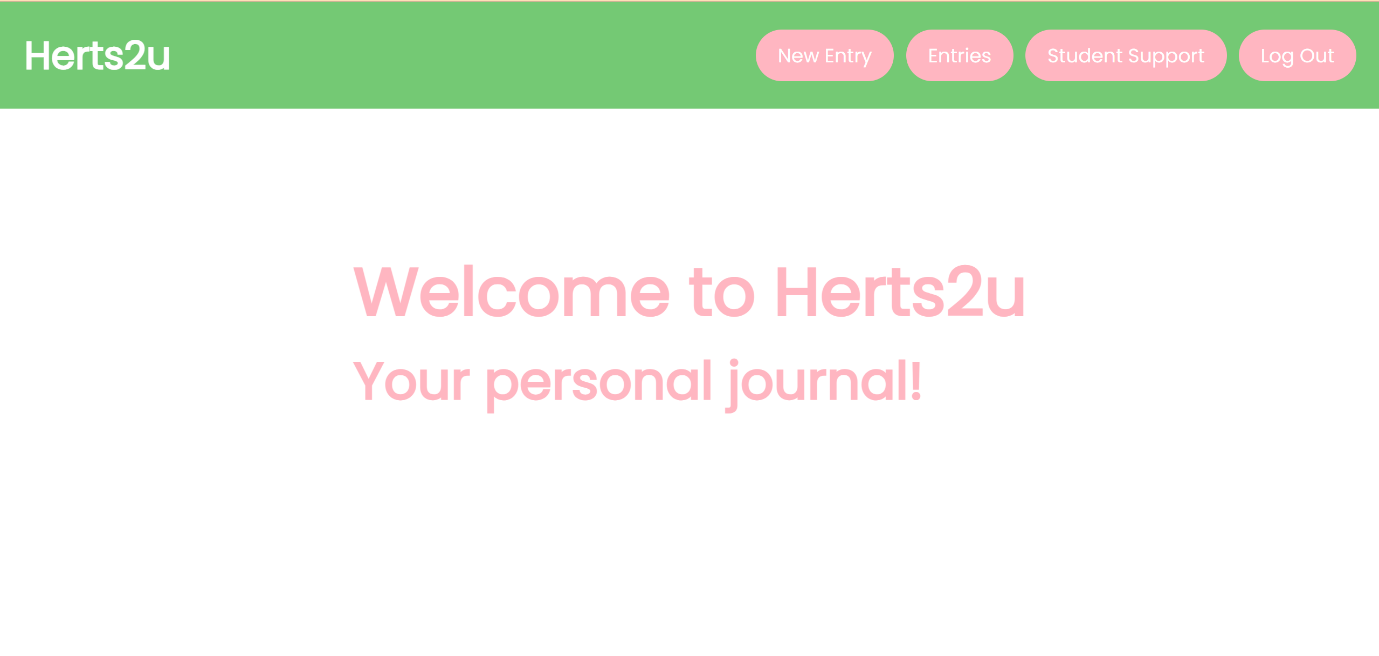
## Image 7



Following the design of the final sign-in/sign-up page, it was important to establish and maintain a consistent visual theme throughout the website. instead of adhering to a monochromatic palette, such as various shades of blue, a more diverse yet harmonious set of colours was chosen, including soft pinks, greens, and purples. These colours were selected for their calming and supportive qualities, which are especially relevant in the context of a mental health journaling platform.

For the welcome section (Image 8), a clean white background was introduced to help maintain user focus. This was an important choice for users who are preparing to journal, as minimising distractions supports emotional clarity and concentration. Maintaining consistency with the sign-in/sign-up interface, green accents are used most notably in the navigation header, while pink buttons highlight the pathways to other pages. These elements reinforce visual continuity and contribute to a soothing, user-centred experience across the site.

## image 8



## Image 9

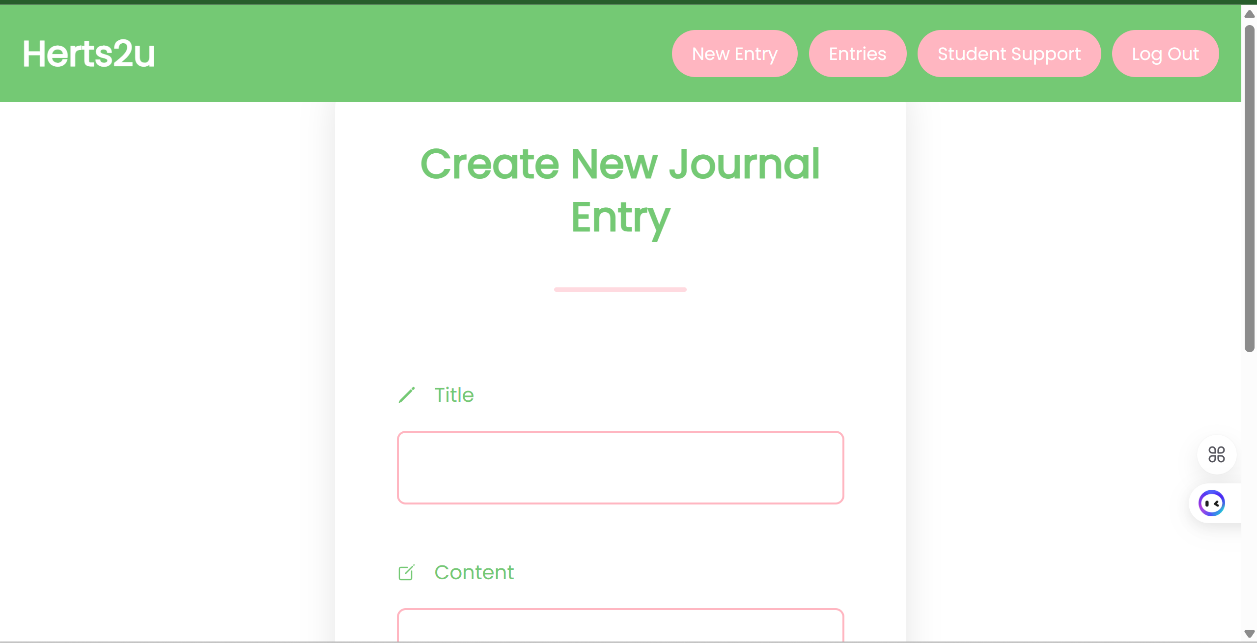


Image 9 showcases the interface of the new entry page of a logged-in user, which allows students to create and submit new journal entries. The design showcases a simple and clean interface using soft colours like pale pink and green, creating a calming and soothing experience, which fits the theme as it's for mental health, as it appears to be friendly and non-distracting. Aiding in the creation of a non-threatening environment for users to experience anxiety and overwhelming stress, as Jonauskite and Mohr highlighted the importance of colour selection in designs targeted towards mental health.

* HTML to structure the elements such as input fields (Title and Content), headers, and navigation.
* CSS was used to style the form with rounded edges, pastel colours, and a clear font style, making it easy to read and interact with.
* Responsive design practices ensure the layout adjusts appropriately to various screen sizes for mobile and tablet devices.
* A navigation bar at the top provides quick access to the main sections (New Entry, Entries, Student Support, Log Out), maintaining usability and consistency across pages.
* Placeholder text and labels guide the user in filling out the form fields, enhancing usability.

It also addresses some accessibility issues, allowing users with different abilities can interact comfortably with the website. The large input fields are designed for visually impaired users. The use of high-contrast text and a soft background colour increases readability, and the font styling maintains clarity throughout.

## Image 10

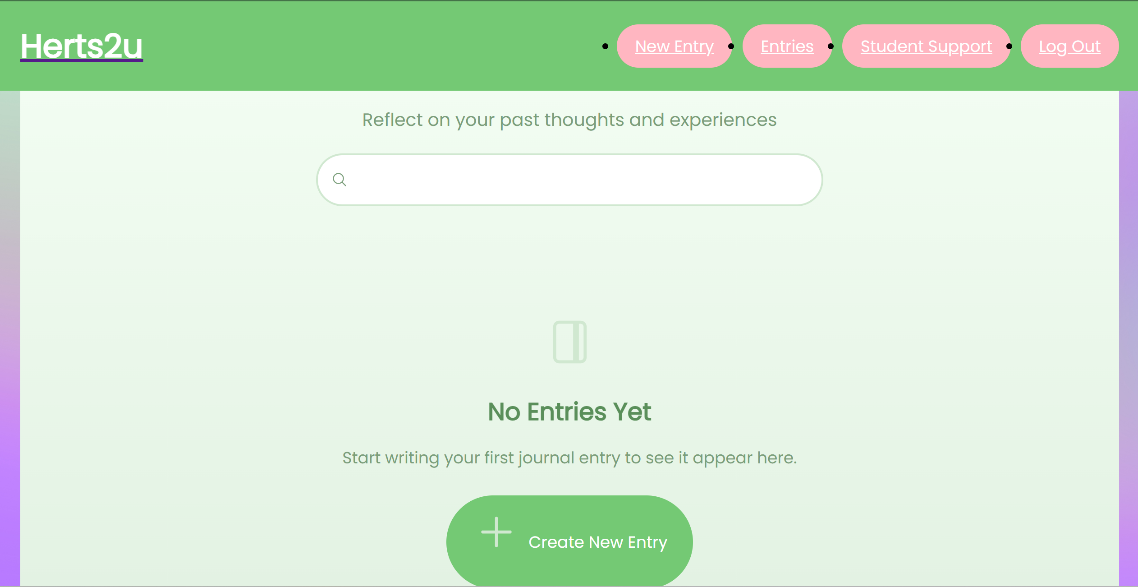
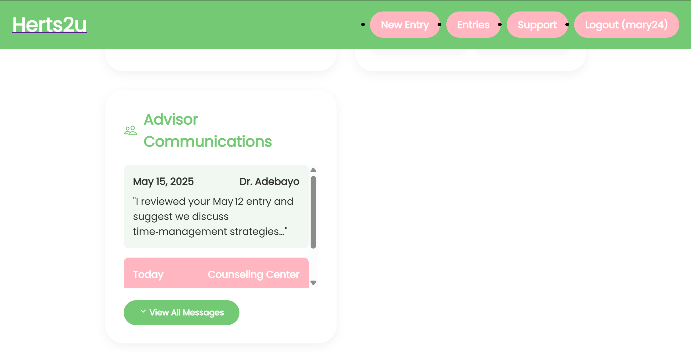


Image 10 showcases the entries section, where all saved entries would be, and it continues on the cohesive visual theme with the sign-up/sign-in page with the purple/green background. With the overall design structure of the welcome screen and the rest of these web pages. With a green colour on top of the purple/green background. The green colour centres focus on the entries as it highlights the user’s attention to the journal entries. The combination of these two helps distinguish the entries from other sections within this website. Whilst maintaining consistency within the whole website. Still aligning with accessibility issues, as the colour contrast is enough for readability without any harsh contrast. The navigation bar also corresponds with Nielsen's Heuristics, such as ‘Consistency and Standards’ and ‘Aesthetic and minimalist design’ are met.

## Images 11& 12



In images 11 and 12, the styling and design are continued as it mimics the welcome page, where the navigation bar is green with its pink buttons. The design is minimal but readable, creating a calming experience as the student support page is probably the most nerve-wrecking part of the website, as students may feel a bit anxious in seeking help and support, so making sure that this webpage has a calming atmosphere is highly important.

## 4.2 Security, Privacy and Protection

As this website deals with personal information, it is a diary for students struggling with their mental health, raising ethical concerns. Security, privacy and protection must be considered and implemented. Whilst this project doesn’t address the full scope of security, privacy and protection. Herts2u does collect personal data on a much smaller scale, primarily being usernames and passwords, which are stored in a secure database. No real user data was collected during development, and access control was implemented to allow registered users to interact with the platform

## 4.3 Backend development:

Once the frontend layout and structure were complete, I started working on the backend of this website. I felt uncertain about handling the backend aspect of this website, so I almost didn’t do it. But after tackling the frontend, I felt more comfortable doing so, as the frontend part provided the structure needed to begin integrating the backend functionality.

To develop and test the backend of this website, I used XAMPP as a local server environment. XAMPP provided the necessary tools, like Apache for hosting the website locally and MySQL for managing the related database. Allowing me to interact and run the website locally before bringing it to a live server.

The backend of this website was primarily for the sign-in/sign-up, where PHP and MySQL were used. PHP was chosen because of its seamless and easy integration with HTML and server-side scripting. PHP was used for key features such as user registration, authenticating user login and interaction with the database. PHP also allowed for free hosting like Infinity Free.

For storing user data such as usernames and passwords, a database called herts2u was created. Passwords were protected using PHP’s password\_hash () function, which ensures they are not stored in plain text. This is important in preventing unauthorised access to the website, aligning with good security practice. The backend development also consisted of basic form validation and the registration process.

## 4.4 Database Implementation

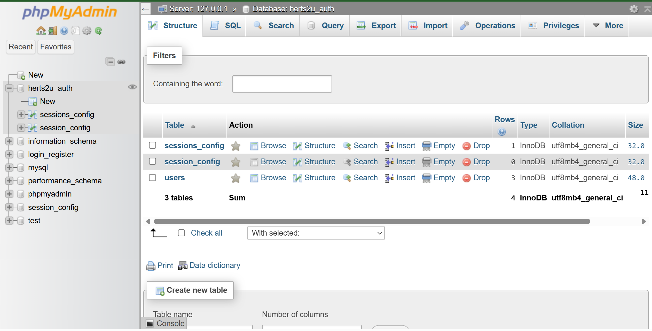
To help with the sign-in/sign-up web page, I created a MySQL database using phpMyAdmin through XAMPP, shown in image 13. The database was named herts2u and was used to store user data in a secure and safe way. Whilst the focus was to store user data such as their username, password and their email when it came to registering their account. The database did help with how data is handled, making the login process possible. the password was protected using a password\_hash (), ensuring that the passwords remain encrypted rather than being saved as plaintext, aiding in another security feature.

The database had a simple structure but played a big role in making the login process function correctly. It allowed for smooth interaction between the front-end aspect of this website and the stored data. This made it possible to validate users, check if credentials are correct and manage account access. Though minimal, this database does show effective organisational skills and the securing of user information. The database can be seen in image 14.

## Image 13

Before being able to make a database in phpMyAdmin, XAMPP had to be installed, as XAMPP provided access to phpMyAdmin. From image 13, you can see that Apache and MySQL had to be running to access phpMyAdmin. Once running, you need to locate the ‘localhost’. This can be done by either clicking admin or going on the web browser and typing localhost directly.

## Image 14



Showcases that the sessions\_config table shows that the login was registered through

# Chapter 5: Testing

5.1 Before testing this website, I considered many testing methodologies, one being Test-Driven Development (TDD). TDD is a methodology where tests are written before the code. TDD is typically used in backend development as it ensures that code behaves the way it is expected to. Which is important for data handling. However, I opted out of using this method as I wasn’t confident in developing an effective test due to my lack of experience with backend development. I instead decided to use Manual user testing and Black box testing.

Manual user testing means that the user goes through the website as if they were a real-world user. There is no ‘script’ to follow or conditions that must be met other than the website working with no errors. Manual testing checks whether the website works, does what it intends to and if it’s easy to use and navigate. Manual user testing was used to carry out the user experience and provide basic functionality of the website. down below are the areas that were tested in Table 2

## Table 2

|  |  |  |
| --- | --- | --- |
| Test Area | Description | Conditions Met (Yes/No) |
| Sign-up Process | Different emails, usernames and passwords are formed to check if they create an account successfully | Yes |
| Login Functionality | Logging in with both correct and incorrect credentials to ensure access is only given to registered users. | Yes |
| Navigation Bar | Clicking each link to verify it routes correctly and the page layout remains consistent. | Yes |

Black-box testing focuses on checking whether the system behaves as expected from a user's experience without needing to understand the internal code or reasoning behind it. The goal of black-box testing is to see if the outputs meet the intended results. This method was used for my website as it allows me to see if the features of this website work the way they’re intended to. Which can be seen in Table 3

## Table 3

|  |  |  |
| --- | --- | --- |
| Test Area | Description | Conditions met Yes/No |
| Sign-Up Form | Tried both correct and incorrect inputs (like short passwords or wrong email formats) to see if the form showed the right messages. | Yes |
| Login Authentication | Tried logging in with correct and wrong details (like wrong passwords or fake users) to check if access and error messages worked properly. | Yes |

# Chapter 6

## 6.1 Evaluation:

The aim of this project was a website that allows students to register and engage with a website that allows them to journal about their mental health. This website has a low entry barrier for students struggling with mental health to seek support. Whilst I think the visual design of this website was successful, as it overall provides a calm, clean interface that remains partially functional from a backend technical perspective.

The login system was implemented using both front-end and backend programming. Simple authentication was implemented, with hashed passwords using encryption, session management and a very simple error handling. A combination of these helped build a more secure and private platform, as it aligns with Nielsen’s heuristic, such as visibility of system status by providing feedback during login attempts.

Despite this, the remaining part of the website is primarily front-end. Failing to achieve a fully functional website. For example, whilst the journal entries are presented, they remain as much of a design feature as it doesn’t fully work as there is no backend to ensure that the journal entries are saved. This is primarily due to the difficulty I had when implementing the database, which proved to be a challenge for me. Connecting PHP and MySQL. Constantly trying to connect to the database. Making sure that passwords were hashed was difficult, but overall, it proved to be a learning experience for me. As I watched multiple YouTube videos to help.

To improve this website, I would extend the backend code to the rest of this website. for example, features like saving the entries and them being stored in the entries section would be enabled through a further developed database. Improving the error handling and overall user experience by complying better with the WCAG guidelines would enhance usability, further supporting Nielsen's heuristics such as ‘Help and documentation’ and ‘Help users recognise, diagnose, and recover from errors.

## 6.2 Conclusion

To conclude, this project has been a learning experience with both front-end and backend web development. As I had very little history with both of them, the visual and structural design of the website was met, which created a calm and accessible platform for university students struggling with their mental health. The project also revealed a lot of issues and complications when implementing the backend. Especially with the PHP and MySQL integration. It took a lot of YouTube videos to understand it, and just taking breaks and then coming back with a refreshed mind. Despite the challenges faced, I am happy that I was able to build a working login system, as it served as a milestone for me. To improve this website overall, I would extend the backend development throughout the website, which is referenced again in my evaluation. However, in addition to that, I would create a CRUD system, which is the Create, Read, Update, and Delete. This would allow users to manage their entry logs a lot better. I would also implement a mood tracker, as it would provide analytical data. In the end, this project served as a great learning opportunity for me, as it helped me develop skills in areas I wouldn’t have considered and highlighted my areas for improvement

# References

Traversy Media. (2020) HTML Tutorial for Beginners: HTML Crash Course [Video]. YouTube. Available at: https://www.youtube.com/watch?v=qz0aGYrrlhU&t=1110s (Accessed: 23 Mar 2025).

The Net Ninja. (2017) PHP & MySQL Tutorial for Beginners #1 - Introduction [Video]. YouTube. Available at: https://www.youtube.com/watch?v=pWG7ajC\_OVo&list=PL4cUxeGkcC9gksOX3Kd9KPo-O68ncT05o (Accessed: 23 Mar 2025).

Bro Code. (2022) *PHP Login System with MySQL Database* [Video]. YouTube. Available at: <https://www.youtube.com/watch?v=r0lDDeVkaks&t=287s> (Accessed: 23 Mar 2025).

(Assumed Channel Name). (2021) How To Make Website Using HTML & CSS | Full Responsive Multi … [Video]. YouTube. Available at: https://www.youtube.com/watch?v=oYRda7UtuhA&t=427s (Accessed: 26 Mar 2025).

Coding2GO. (2021) *How to Create a Responsive Navigation Bar for Beginners* [Video]. YouTube. Available at: <https://www.youtube.com/watch?v=U8smiWQ8Seg> (Accessed: 26 Mar 2025).

Pfeffer Baum, B. and North, C.S., 2020. *Mental health and the COVID-19 pandemic*. The New England Journal of Medicine, 383(6), pp.510–512. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7361582/> [Accessed 16 Mar. 2025].

Meltzer, D.O., et al., 2024. *The impact of COVID-19 mitigation measures on mental health: A difference-in-differences analysis*. Scientific Reports, 14, Article 4892. Available at: <https://www.nature.com/articles/s41598-024-55879-9> [Accessed 16 Mar. 2025].

O'Reilly, M., Dogra, N., Whiteman, N. and Hughes, J., 2023. *The impact of COVID-19 on youth mental health: A qualitative study of perceived challenges and supports*. Psychiatry Research, 321, 115071. Available at: <https://www.sciencedirect.com/science/article/pii/S0165178123000355> [Accessed 16 Mar. 2025].

Suhr, M., Risch, A.K., Wilz, G. and Heidenreich, T., 2017. Maintaining mental health through positive writing: Effects of a resource diary on depression and emotion regulation. *Journal of Clinical Psychology*, 73(12), pp.1586–1598. Available at: <https://pubmed.ncbi.nlm.nih.gov/28186634/> [Accessed 17 Mar. 2025].

Nielsen Norman Group (2020). *10 Usability Heuristics for User Interface Design*. Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/> [Accessed: 25 May 2025].

Bibliography

UK Parliament. (2021). *Student mental health in higher education*. [online] Available at: https://www.parliament.uk/mental-health-students [Accessed 26 Jun. 2025].

Eisenberg, D., Hunt, J. and Speer, N. (2019). *Mental health in American colleges and universities: Variation across student subgroups and across campuses*. *Journal of Nervous and Mental Disease*, [online] 200(1), pp.60–67. Available at: https://journals.lww.com/jonmd/Fulltext/2013/01000/Mental\_Health\_in\_American\_Colleges\_and.9.aspx [Accessed 26 Jun. 2025].

Broadhurst, R., Grabosky, P., Alazab, M. and Chon, S. (2020). *Cybercrime risks for university students*. *Journal of Cyber Security Technology*, 4(2), pp.97–115. [Accessed 26 Jun. 2025].

Suhr, J.A., O’Connell, J. and Renaud, A. (2021). *Resource journaling as a resilience-building intervention: Effects on depression and emotional regulation*. *Journal of Mental Health Counselling*, 43(3), pp.220–236.

# Appendices: HTML/PHP

index php  
<?php

require\_once \_\_DIR\_\_ . '/session\_config.php';

*// handling error/success messages*

$error = isset($\_GET['error']) ? htmlspecialchars($\_GET['error']) : '';

$success = isset($\_GET['success']) ? htmlspecialchars($\_GET['success']) : '';

?>

<!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>Herts2u | Login & Register</title>

    <link rel="stylesheet" href="assets/css/style.css">

</head>

<body>

*<!-- error/success Mmssages -->*

    <?php if ($error): ?>

        <div style="color: red; text-align: center; padding: 10px; margin-bottom: 15px;">

            <?php echo $error; ?>

        </div>

    <?php endif; ?>

    <?php if ($success): ?>

        <div style="color: green; text-align: center; padding: 10px; margin-bottom: 15px;">

            <?php echo $success; ?>

        </div>

    <?php endif; ?>

    <div class="form-container">

        <div class="form-col">

            <div class="btn-box">

                <button class="btn btn-1" id="Login">Sign In</button>

                <button class="btn btn-2" id="Register">Sign Up</button>

            </div>

*<!--login form with php -->*

            <form class="form-box login-form" id="loginform" action="process\_login.php" method="POST">

                <div class="form-title">

                    <span>Sign In</span>

                </div>

                <div class="form-inputs">

                    <div class="input-box">

                        <input type="text" name="username" class="inputs input-field" placeholder="Username" required>

                        <ion-icon name="person-outline" class="icon"></ion-icon>

                    </div>

                    <div class="input-box">

                        <input type="password" name="password" oninput="changeIcon(this.value)" id="logPassword" class="inputs input-field" placeholder="Password" required>

                        <ion-icon name="lock-closed-outline" class="icon" id="log-pass-icon" onclick="myLogPassword()"></ion-icon>

                    </div>

                    <div class="forgot-pass">

                        <a href="#">Forgot Password?</a>

                    </div>

                    <div class="input-box">

                        <button type="submit" class="inputs submit-btn">

                            <span>Sign In</span>

                            <ion-icon name="arrow-forward-outline"></ion-icon>

                        </button>

                    </div>

                </div>

            </form>

*<!-- register form php now -->*

            <form class="form-box register-form" action="process\_reg.php" method="POST">

                <div class="form-title">

                    <span>Sign Up</span>

                </div>

                <div class="form-inputs">

                    <div class="input-box">

                        <input type="email" name="email" class="inputs input-field" placeholder="Email" required>

                        <ion-icon name="mail-outline" class="icon"></ion-icon>

                    </div>

                    <div class="input-box">

                        <input type="text" name="username" class="inputs input-field" placeholder="Username" required>

                        <ion-icon name="person-outline" class="icon"></ion-icon>

                    </div>

                    <div class="input-box">

                        <input type="password" name="password" oninput="changeIcon(this.value)" id="regPassword" class="inputs input-field" placeholder="Password" required>

                        <ion-icon name="lock-closed-outline" class="icon" id="reg-pass-icon" onclick="myRegPassword()"></ion-icon>

                    </div>

                    <div class="remember-me">

                        <input type="checkbox" id="remember-me-check" name="remember\_me">

                        <label for="remember-me-check">Remember me</label>

                    </div>

                    <div class="input-box">

                        <button type="submit" class="inputs submit-btn">

                            <span>Sign Up</span>

                            <ion-icon name="arrow-forward-outline"></ion-icon>

                        </button>

                    </div>

                </div>

            </form>

        </div>

    </div>

    <script src="assets/js/main.js"></script>

    <script type="module" src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>

    <script nomodule src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>

</body>

</html>

welcome php  
<?php

require\_once \_\_DIR\_\_ . '/session\_config.php';

require 'db\_connect.php';

*// auto-logout*

if (!isset($\_SESSION['user\_id']) ||

    $\_SESSION['ip'] !== $\_SERVER['REMOTE\_ADDR'] ||

    $\_SESSION['ua'] !== ($\_SERVER['HTTP\_USER\_AGENT'] ?? '') ||

    time() - $\_SESSION['last\_activity'] > 1800) {

    session\_unset();

    session\_destroy();

    header("Location: index.php?error=session\_expired");

    exit();

}

*// update activity timestamp*

$\_SESSION['last\_activity'] = time();

*// get user data*

$stmt = $pdo->prepare("SELECT email, created\_at FROM users WHERE id = ?");

$stmt->execute([$\_SESSION['user\_id']]);

$user = $stmt->fetch();

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Dashboard | Herts2u</title>

    <link rel="stylesheet" href="assets/css/welcome.css">

    <script>

**let** timeoutWarning = setTimeout(**function**() {

        alert('Your session will expire in 2 minutes due to inactivity.');

    }, 1500000); *// 25 minutes*

    </script>

</head>

<body>

*<!-- navigation area -->*

    <header>

        <nav class="navbar section-content">

            <a href="welcome.php" class="nav-logo">

                <h2 class="logo-text">Herts2u</h2>

            </a>

            <ul class="nav-menu">

                <li class="nav-item">

                    <a href="new\_entry.php" class="nav-link">New Entry</a>

                </li>

                <li class="nav-item">

                    <a href="entries.php" class="nav-link">Entries</a>

                </li>

                <li class="nav-item">

                    <a href="student\_support.php" class="nav-link">Support</a>

                </li>

                <li class="nav-item">

                    <a href="logout.php" class="nav-link">Logout (<?= htmlspecialchars($\_SESSION['username']) ?>)</a>

                </li>

            </ul>

        </nav>

    </header>

*<!-- dashboard cotnet -->*

    <main class="welcome-section">

        <div class="section-content">

            <div class="welcome-details">

                <h1 class="title">Welcome to Herts2u</h1>

                <h2 class="subtitle">Hello, <?= htmlspecialchars($\_SESSION['username']) ?>!</h2>

*<!-- Dashboard Cards -->*

                <div class="dashboard-cards">

                    <div class="dashboard-card user-card">

                        <h3>Your Profile</h3>

                        <p><strong>Email:</strong> <?= htmlspecialchars($user['email']) ?></p>

                        <p><strong>Member since:</strong> <?= date('F j, Y', strtotime($user['created\_at'])) ?></p>

                        <p><strong>Last activity:</strong> <span id="activity-timer"></span></p>

                    </div>

                    <div class="dashboard-card quick-actions">

                        <h3>Quick Actions</h3>

                        <a href="new\_entry.php" class="action-btn">Create New Entry</a>

                        <a href="entries.php" class="action-btn">View All Entries</a>

                    </div>

                </div>

            </div>

        </div>

    </main>

    <script>

*// session timer display*

    document.addEventListener('DOMContentLoaded', **function**() {

**let** lastActivity = <?= $\_SESSION['last\_activity'] ?>;

**function** updateTimer() {

**const** now = Math.floor(Date.now() / 1000);

**const** elapsed = now - lastActivity;

**const** remaining = 1800 - elapsed; *// 30 minutes*

            if (remaining <= 0) {

                window.location.href = 'logout.php?timeout=1';

                return;

            }

**const** minutes = Math.floor(remaining / 60);

**const** seconds = remaining % 60;

            document.getElementById('activity-timer').textContent =

                `${minutes}m ${seconds}s remaining`;

        }

        updateTimer();

        setInterval(updateTimer, 1000);

        ['click', 'keypress', 'mousemove'].forEach(event **=>** {

            window.addEventListener(event, () **=>** {

                fetch('ping.php').catch(() **=>** {}); *// silent*

            });

        });

    });

    </script>

</body>

</html>

New\_entry php  
<?php

*// 1. session configuration*

require\_once \_\_DIR\_\_ . '/session\_config.php';

*// 2. Check authentication*

if (!isset($\_SESSION['username'])) {

*// Store where user was trying to go*

    $\_SESSION['redirect\_url'] = $\_SERVER['REQUEST\_URI'];

    header("Location: index.php");

    exit();

}

*// 3. generate CSRF token if not exists*

if (!isset($\_SESSION['csrf\_token'])) {

    $\_SESSION['csrf\_token'] = bin2hex(random\_bytes(32));

}

*// 4. security headers*

header("X-Frame-Options: DENY");

header("X-Content-Type-Options: nosniff");

header("X-XSS-Protection: 1; mode=block");

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <meta http-equiv="Content-Security-Policy" content="default-src 'self'; script-src 'self' 'unsafe-inline'; style-src 'self' 'unsafe-inline';">

    <title>New Entry | Herts2u</title>

*<!-- loads css for a better performance -->*

    <link rel="preload" href="assets/css/style.css" as="style">

    <link rel="preload" href="assets/css/welcome.css" as="style">

    <link rel="preload" href="assets/css/new\_entry.css" as="style">

*<!--  CSS Files -->*

    <link rel="stylesheet" href="assets/css/style.css">

    <link rel="stylesheet" href="assets/css/welcome.css">

    <link rel="stylesheet" href="assets/css/new\_entry.css">

*<!-- favicon -->*

    <link rel="icon" href="assets/images/favicon.ico" type="image/x-icon">

</head>

<body>

*<!-- header -->*

    <header>

        <nav class="navbar section-content">

            <a href="welcome.php" class="nav-logo">

                <h2 class="logo-text">Herts2u</h2>

            </a>

            <ul class="nav-menu">

                <li class="nav-item">

                    <a href="new\_entry.php" class="nav-link active">New Entry</a>

                </li>

                <li class="nav-item">

                    <a href="entries.php" class="nav-link">Entries</a>

                </li>

                <li class="nav-item">

                    <a href="student\_support.php" class="nav-link">Student Support</a>

                </li>

                <li class="nav-item">

                    <a href="logout.php" class="nav-link">Log Out</a>

                </li>

            </ul>

            <div class="hamburger">

                <span class="bar"></span>

                <span class="bar"></span>

                <span class="bar"></span>

            </div>

        </nav>

    </header>

*<!-- main body -->*

    <main style="padding-top: 80px;">

        <section class="welcome-section">

            <div class="section-content">

                <form class="welcome-details" action="process\_entry.php" method="POST" id="journalForm">

                    <input type="hidden" name="csrf\_token" value="<?php echo htmlspecialchars($\_SESSION['csrf\_token'], ENT\_QUOTES, 'UTF-8'); ?>">

                    <h1 class="title">New Journal Entry</h1>

                    <div class="form-group">

                        <label for="entry\_title">Title</label>

                        <input type="text" id="entry\_title" name="entry\_title" required

                               maxlength="100" placeholder="Entry title"

                               pattern="[A-Za-z0-9\s\-\_,\.;:()]+"

                               title="Only letters, numbers, and basic punctuation">

                    </div>

                    <div class="form-group">

                        <label for="entry\_content">Content</label>

                        <textarea id="entry\_content" name="entry\_content" rows="6"

                                  required minlength="20" maxlength="5000"

                                  placeholder="Write your thoughts here..."></textarea>

                    </div>

                    <div class="form-actions">

                        <button type="submit" class="action-btn" id="submitBtn">Save Entry</button>

                        <a href="welcome.php" class="cancel-btn">Cancel</a>

                    </div>

                </form>

            </div>

        </section>

    </main>

*<!-- js (javascript validation) -->*

    <script src="assets/js/form-validation.js"></script>

*<!-- mobile menu functionality -->*

    <script>

        document.addEventListener('DOMContentLoaded', **function**() {

**const** hamburger = document.querySelector('.hamburger');

**const** navMenu = document.querySelector('.nav-menu');

            hamburger.addEventListener('click', **function**() {

                hamburger.classList.toggle('active');

                navMenu.classList.toggle('active');

            });

*// Close menu when clicking on a link*

            document.querySelectorAll('.nav-link').forEach(link **=>** {

                link.addEventListener('click', () **=>** {

                    hamburger.classList.remove('active');

                    navMenu.classList.remove('active');

                });

            });

*// form submission handler*

            document.getElementById('journalForm').addEventListener('submit', **function**(e) {

**const** submitBtn = document.getElementById('submitBtn');

                submitBtn.disabled = true;

                submitBtn.textContent = 'Saving...';

            });

        });

    </script>

</body>

</html>

Entries php  
<!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>Your Entries | Herts2u</title>

*<!-- shared styles -->*

    <link rel="stylesheet" href="assets/css/style.css">

*<!-- Page-specific styles -->*

    <link rel="stylesheet" href="assets/css/entries.css">

*<!-- Icons -->*

    <script type="module" src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>

    <script nomodule src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>

</head>

<body>

*<!-- single Header -->*

    <header>

        <nav class="navbar">

            <a href="welcome.php" class="nav-logo">

                <h2 class="logo-text">Herts2u</h2>

            </a>

            <ul class="nav-menu">

                <li class="nav-item">

                    <a href="new\_entry.php" class="nav-link">New Entry</a>

                </li>

                <li class="nav-item">

                    <a href="entries.php" class="nav-link active">Entries</a>

                </li>

                <li class="nav-item">

                    <a href="student\_support.php" class="nav-link">Student Support</a>

                </li>

                <li class="nav-item">

                    <a href="index.php" class="nav-link">Log Out</a>

                </li>

            </ul>

        </nav>

    </header>

*<!-- main body -->*

    <main class="entries-container">

        <section class="entries-header">

            <h1>Your Journal Entries</h1>

            <p>Reflect on your past thoughts and experiences</p>

*<!-- search Bar -->*

            <div class="search-container">

                <ion-icon name="search-outline"></ion-icon>

                <input type="text" id="searchEntries" placeholder="Search entries...">

            </div>

        </section>

*<!-- entries should be here -->*

        <section class="entries-list" id="entriesList">

*<!-- Empty state with properly linked button -->*

            <div class="empty-state">

                <ion-icon name="journal-outline"></ion-icon>

                <h2>No Entries Yet</h2>

                <p>Start writing your first journal entry to see it appear here.</p>

                <a href="new entry.html" class="btn-new-entry">

                    <ion-icon name="add-outline"></ion-icon>

                    Create New Entry

                </a>

            </div>

        </section>

    </main>

*<!-- JavaScript -->*

    <script src="assets/js/entries.js"></script>

</body>

</html>

student support php  
<?php

*// student support section*

require\_once \_\_DIR\_\_ . '/session\_config.php';   *// starts + validates session*

*//  guard: IP / UA change or 30‑min inactivity should trigger logout*

if (

    !isset($\_SESSION['user\_id']) ||

    ($\_SESSION['ip'] ?? '') !== ($\_SERVER['REMOTE\_ADDR'] ?? '') ||

    ($\_SESSION['ua'] ?? '') !== ($\_SERVER['HTTP\_USER\_AGENT'] ?? '') ||

    time() - ($\_SESSION['last\_activity'] ?? 0) > 1800

) {

    session\_unset();

    session\_destroy();

    header('Location: index.php?error=session\_expired');

    exit();

}

$\_SESSION['last\_activity'] = time();

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <title>Student Support | Herts2u</title>

*<!-- Global + page CSS -->*

    <link rel="stylesheet" href="assets/css/style.css" />

    <link rel="stylesheet" href="assets/css/student\_support.css" />

*<!-- Ionicons -->*

    <script type="module"

        src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>

    <script nomodule

        src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>

</head>

<body>

<header>

    <nav class="navbar">

        <a href="welcome.php" class="nav-logo">

            <span class="logo-text">Herts2u</span>

        </a>

        <ul class="nav-menu">

            <li><a href="new\_entry.php"       class="nav-link">New Entry</a></li>

            <li><a href="entries.php"         class="nav-link">Entries</a></li>

            <li><a href="student\_support.php" class="nav-link">Support</a></li>

            <li><a href="logout.php"          class="nav-link">

                Logout (<?= htmlspecialchars($\_SESSION['username']) ?>)

            </a></li>

        </ul>

    </nav>

</header>

<main class="support-container">

    <section class="support-hero">

        <h1>Student Support Services</h1>

        <p>We're here to help you succeed in your academic journey</p>

    </section>

    <div class="support-grid">

        <section class="support-card consent-section">

            <h2><ion-icon name="document-text-outline"></ion-icon> Journal Entry Support</h2>

            <div class="consent-toggle">

                <label>

                    <input type="checkbox" id="consentCheckbox" />

                    <span class="toggle-slider"></span>

                    <span class="toggle-text">

                        Allow advisors to view my entries when I request support

                    </span>

                </label>

                <p class="consent-note">

                    Advisors will only access entries with your explicit permission each time.

                </p>

            </div>

            <div class="request-support">

                <h3>Request Entry Review</h3>

                <select id="entrySelect" class="form-input">

                    <option value="">Select an entry to share</option>

                    <option value="entry1">Entry #1 – May 5, 2025</option>

                    <option value="entry2">Entry #2 – May 12, 2025</option>

                </select>

                <textarea class="form-input"

                          placeholder="What would you like help with?"></textarea>

                <button class="btn-support-request">

                    <ion-icon name="send-outline"></ion-icon> Send Request

                </button>

            </div>

        </section>

        <section class="support-card quick-options">

            <h2><ion-icon name="help-buoy-outline"></ion-icon> Quick Help</h2>

            <div class="option-grid">

                <div class="option-item">

                    <ion-icon name="calendar-outline"></ion-icon>

                    <h3>Book Appointment</h3>

                    <p>Schedule 1‑on‑1 time with an advisor</p>

                </div>

                <div class="option-item">

                    <ion-icon name="chatbubbles-outline"></ion-icon>

                    <h3>Live Chat</h3>

                    <p>Instant messaging with support staff</p>

                </div>

                <div class="option-item">

                    <ion-icon name="library-outline"></ion-icon>

                    <h3>Resources</h3>

                    <p>Self‑help guides and worksheets</p>

                </div>

                <div class="option-item">

                    <ion-icon name="alert-circle-outline"></ion-icon>

                    <h3>Urgent Help</h3>

                    <p>24/7 crisis support contacts</p>

                </div>

            </div>

        </section>

        <section class="support-card advisor-notes">

            <h2><ion-icon name="people-outline"></ion-icon> Advisor Communications</h2>

            <div class="notes-container">

                <div class="note">

                    <div class="note-header">

                        <span class="note-date">May 15, 2025</span>

                        <span class="note-advisor">Dr. Adebayo</span>

                    </div>

                    <p>"I reviewed your May 12 entry and suggest we discuss time‑management strategies…"</p>

                </div>

                <div class="note new">

                    <div class="note-header">

                        <span class="note-date">Today</span>

                        <span class="note-advisor">Counseling Center</span>

                    </div>

                    <p>"Your requested appointment is confirmed for Friday at 2 pm."</p>

                </div>

            </div>

            <button class="btn-view-all">

                <ion-icon name="chevron-down-outline"></ion-icon> View All Messages

            </button>

        </section>

    </div>

</main>

</body>

</html>

Process login php  
<?php

ini\_set('display\_errors', 1);

error\_reporting(E\_ALL);

require 'session\_config.php';

require 'db\_connect.php';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

*//  inputs*

    $username = trim($\_POST['username']);

    $password = trim($\_POST['password']);

*// Validate inputs*

    if (empty($username) || empty($password)) {

        header("Location: index.php?error=Username+and+password+are+required");

        exit();

    }

    try {

*// Checks user credentials*

        $stmt = $pdo->prepare("SELECT id, username, password FROM users WHERE username = ?");

        $stmt->execute([$username]);

        $user = $stmt->fetch();

        if ($user && password\_verify($password, $user['password'])) {

*// destroy any existing session*

            session\_destroy();

*// start fresh secure session*

            session\_start([

                'use\_strict\_mode' => true,

                'cookie\_httponly' => true,

                'cookie\_samesite' => 'Lax'

            ]);

*// regenerate ID before setting data*

            session\_regenerate\_id(true);

*// set session data*

            $\_SESSION = [

                'user\_id' => $user['id'],

                'username' => $user['username'],

                'last\_activity' => time(),

                'ip' => $\_SERVER['REMOTE\_ADDR'],

                'ua' => $\_SERVER['HTTP\_USER\_AGENT'] ?? '',

                'created' => time()

            ];

*// Remember me functionality is optionall*

            if (isset($\_POST['remember\_me'])) {

                $token = bin2hex(random\_bytes(32));

                setcookie('remember\_token', $token, [

                    'expires' => time() + 60\*60\*24\*30,

                    'path' => '/',

                    'secure' => isset($\_SERVER['HTTPS']),

                    'httponly' => true,

                    'samesite' => 'Lax'

                ]);

*// stores hashed token in database*

                $hashedToken = password\_hash($token, PASSWORD\_DEFAULT);

                $updateStmt = $pdo->prepare("UPDATE users SET remember\_token = ? WHERE id = ?");

                $updateStmt->execute([$hashedToken, $user['id']]);

            }

*// makes sure session was saved*

            if (isset($\_SESSION['user\_id'])) {

                header("Location: welcome.php");

                exit();

            } else {

                throw new Exception("Session data not saved");

            }

        } else {

            header("Location: index.php?error=Invalid+credentials");

            exit();

        }

    } catch (PDOException $e) {

        error\_log("Database error: " . $e->getMessage());

        header("Location: index.php?error=System+error");

        exit();

    } catch (Exception $e) {

        error\_log("Session error: " . $e->getMessage());

        header("Location: index.php?error=Session+error");

        exit();

    }

} else {

*// Non-POST request*

    header("Location: index.php");

    exit();

}

?>

Session config php  
<?php

**class** DB\_Session\_Handler **implements** SessionHandlerInterface {

**private** $pdo;

**private** $lockTimeout = 5; *// seconds*

**public** **function** \_\_construct() {

        try {

            $this->pdo = new PDO(

                'mysql:host=localhost;dbname=herts2u\_auth',

                'root',

                '',

                [

                    PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,

                    PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC,

                    PDO::ATTR\_EMULATE\_PREPARES => false,

                    PDO::ATTR\_PERSISTENT => true

                ]

            );

            $this->verifySessionTable();

        } catch (PDOException $e) {

            error\_log("SESSION INIT FAILED: " . $e->getMessage());

            throw new RuntimeException("Database connection failed.");

        }

    }

**private** **function** verifySessionTable() {

        $this->pdo->exec("

            CREATE TABLE IF NOT EXISTS sessions\_config (

                session\_id VARCHAR(128) NOT NULL PRIMARY KEY,

                data TEXT NOT NULL,

                last\_activity TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

                session\_lifetime INT NOT NULL DEFAULT 86400,

                INDEX (last\_activity)

            ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4

        ");

    }

**public** **function** open($savePath, $sessionName): bool {

        return true;

    }

**public** **function** close(): bool {

        return true;

    }

**public** **function** read($id): string {

        try {

            $stmt = $this->pdo->prepare(

                "SELECT data FROM sessions\_config WHERE session\_id = ?"

            );

            $stmt->execute([$id]);

            $result = $stmt->fetch();

            return $result['data'] ?? '';

        } catch (PDOException $e) {

            error\_log("SESSION READ ERROR: " . $e->getMessage());

            return '';

        }

    }

**public** **function** write($id, $data): bool {

        try {

            $stmt = $this->pdo->prepare("

                REPLACE INTO sessions\_config

                (session\_id, data, last\_activity, session\_lifetime)

                VALUES (?, ?, NOW(), ?)

            ");

            return $stmt->execute([

                $id,

                $data,

                ini\_get('session.gc\_maxlifetime')

            ]);

        } catch (PDOException $e) {

            error\_log("SESSION WRITE ERROR: " . $e->getMessage());

            return false;

        }

    }

**public** **function** destroy($id): bool {

        try {

            $stmt = $this->pdo->prepare("DELETE FROM sessions\_config WHERE session\_id = ?");

            return $stmt->execute([$id]);

        } catch (PDOException $e) {

            error\_log("SESSION DESTROY ERROR: " . $e->getMessage());

            return false;

        }

    }

**public** **function** gc($max\_lifetime): int|false {

        try {

            $stmt = $this->pdo->prepare("

                DELETE FROM sessions\_config

                WHERE last\_activity < DATE\_SUB(NOW(), INTERVAL ? SECOND)

            ");

            $stmt->execute([$max\_lifetime]);

            return $stmt->rowCount();

        } catch (PDOException $e) {

            error\_log("SESSION GC ERROR: " . $e->getMessage());

            return false;

        }

    }

}

*// setup session*

ini\_set('display\_errors', 1);

error\_reporting(E\_ALL);

date\_default\_timezone\_set('UTC');

*// Secure session config*

ini\_set('session.gc\_probability', 1);

ini\_set('session.gc\_divisor', 100);

ini\_set('session.gc\_maxlifetime', 86400); *// 1 day*

ini\_set('session.cookie\_lifetime', 86400);

ini\_set('session.use\_strict\_mode', 1);

ini\_set('session.use\_only\_cookies', 1);

ini\_set('session.cookie\_httponly', 1);

ini\_set('session.cookie\_secure', isset($\_SERVER['HTTPS']));

ini\_set('session.cookie\_samesite', 'Lax');

session\_set\_cookie\_params([

    'lifetime' => 86400,

    'path' => '/',

    'domain' => $\_SERVER['HTTP\_HOST'],

    'secure' => isset($\_SERVER['HTTPS']),

    'httponly' => true,

    'samesite' => 'Lax'

]);

session\_name('HERTS2U\_SESS');

*// use session handler*

$handler = new DB\_Session\_Handler();

session\_set\_save\_handler($handler, true);

register\_shutdown\_function('session\_write\_close');

*// validate session*

**function** validateSession() {

    if (!isset($\_SESSION['\_validated'])) {

        session\_regenerate\_id(true);

        $\_SESSION['\_validated'] = true;

        $\_SESSION['\_created'] = time();

        $\_SESSION['\_ip'] = $\_SERVER['REMOTE\_ADDR'] ?? '';

        $\_SESSION['\_ua'] = $\_SERVER['HTTP\_USER\_AGENT'] ?? '';

        return;

    }

    if ($\_SESSION['\_ip'] !== ($\_SERVER['REMOTE\_ADDR'] ?? '') ||

        $\_SESSION['\_ua'] !== ($\_SERVER['HTTP\_USER\_AGENT'] ?? '')

    ) {

        session\_destroy();

        throw new RuntimeException('Session security check failed');

    }

    if (time() - ($\_SESSION['\_created'] ?? 0) > 1800) {

        session\_regenerate\_id(true);

        $\_SESSION['\_created'] = time();

    }

}

*// start session*

try {

    session\_start();

    validateSession();

} catch (Exception $e) {

    error\_log("SESSION ERROR: " . $e->getMessage());

    session\_unset();

    session\_destroy();

    session\_start();

    session\_regenerate\_id(true);

    $\_SESSION = [];

    validateSession();

}

?>

Session test php  
<?php

require 'session\_config.php';

*// write test data*

$\_SESSION['debug\_time'] = date('H:i:s');

$session\_id = session\_id();

echo "<h2>Session Test</h2>";

echo "<p>Session ID: $session\_id</p>";

echo "<p>Stored data: " . $\_SESSION['debug\_time'] . "</p>";

*// vrify database write*

echo "<h3>Database Verification</h3>";

$stmt = $pdo->prepare("SELECT data FROM sessions\_config WHERE session\_id = ?");

$stmt->execute([$session\_id]);

$db\_data = $stmt->fetch();

echo "<pre>Database content: ";

print\_r($db\_data);

echo "</pre>";

?>

test login php  
<?php

session\_start();

$\_SESSION['user'] = 'test\_user'; *// force login success*

header("Location: test\_welcome.php");

exit();

?>

Test\_redirect php  
<?php

header("Location: welcome.php");

exit();

?>

Dashboard php   
<?php

session\_start();

if (!isset($\_SESSION['user\_id'])) {

    header("Location: index.php");

    exit();

}

echo "<h1>Welcome, " . htmlspecialchars($\_SESSION['username']) . "!</h1>";

echo '<a href="logout.php">Logout</a>';

?>

db\_connect php  
<?php

$host = '127.0.0.1';

$dbname = 'herts2u\_auth';

$username = 'root';

$password = '';

try {

    $pdo = new PDO(

        "mysql:host=$host;dbname=$dbname;charset=utf8mb4",

        $username,

        $password,

        [

            PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,

            PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC

        ]

    );

} catch (PDOException $e) {

    error\_log("Database error: " . $e->getMessage());

    die("Database connection error. Please try again later.");

}

?>

CSS code   
  
style css

@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');

\* {

    padding: 0;

    margin: 0;

    box-sizing: border-box;

    font-family: "Poppins", sans-serif;

}

*/\* body\*/*

body{

    background: url("../img/new.jpg");

    background-position: center;

    background-size: cover;

    background-attachment: fixed;

    background-repeat: no-repeat;

    display: flex;

    align-items: center;

    justify-content: center;

    min-height: 100vh;

    padding: 0 20px;

}

*/\* form \*/*

.form-container {

    display: flex;

    width: 100%;

    max-width: 450px;

    height: 600px;

    border: 3px solid rgba(255,255,255, 0.3);

    border-radius: 30px;

    backdrop-filter: blur(80px);

    overflow: hidden;

}

*/\*form coloumn\*/*

.form-col{

    position: relative;

    width: 100%;

    overflow: hidden;

}

.btn-box{

    display: flex;

    justify-content: center;

    gap: 10px;

    margin-top: 30px;

}

.btn{

    font-weight: 500;

    padding: 8px 30px;

    border: none;

    border-radius: 30px;

    background: rgba(255,255,255,0.2);

    color: #fff;

    box-shadow: 0 5px 10px rgba(0,0,0,0.2);

    cursor: pointer;

    transition: 0.2s;

}

.btn-1{

    background:  #74c974;

}

.btn:hover{

    opacity: 0.85;

}

*/\* form box\*/*

.form-box{

    position: absolute;

    transform: translateX(-50%);

    display: flex;

    flex-direction: column;

    align-items: center;

    width: 100%;

    padding: 0 30px;

    transition: .3s;

}

.form-title{

    margin: 40px 0;

    color: #fff;

    font-size: 28px;

    font-weight: 600;

}

*/\* login form\*/*

.login-form{

    left: 50%;

}

*/\* register form\*/*

.register-form{

    left: -50%;

}

.register-form .form-title{

    margin-block: 40px 20px;

}

*/\* form inputs\*/*

.form-inputs{

    width: 100%;

}

.input-box{

    position: relative;

}

.inputs{

    width: 100%;

    height: 55px;

    margin: 10px 0;

    color: #fff;

    border: none;

    border-radius: 50px;

    box-shadow: 0 0 10px rgba(0,0,0,0.2);

}

.input-field{

    font-size: 16px;

    padding: 0 40px 20px;

    background: rgba(255,255,255,0.2);

    outline: none;

    backdrop-filter: blur(20px);

}

::placeholder{

    color: #fff;

    font-size: 15px;

}

.input-box .icon{

    position: absolute;

    top: 50%;

    right: 20px;

    transform: translateY();

    color: #fff;

    width: 17px;

    height: 17px;

}

.forgot-pass, .remember-me{

    display: flex;

    justify-content: right;

    gap: 5px;

}

.forgot-pass a{

    color: #fff;

    text-decoration: none;

    font-size: 14px;

}

.forgot-pass a:hover{

    text-decoration: underline;

}

.remember-me label{

    color: #fff;

    font-size: 14px;

}

.submit-btn{

    display: flex;

    align-items: center;

    justify-content: center;

    gap: 10px;

    background: #74c974;

    cursor: pointer;

    transition: .3s;

}

.submit-btn:hover{

    gap: 15px;

}

.input-size .icon{

    position: absolute;

    top: 50%;

    right: 20px;

    transform: translateY(-50%);

    color: #fff;

    width: 17px;

    height: 17px;

}

Welcome css   
@import url('https://fonts.googleapis.com/css2?family=Poppins&family=Miniver&display=swap');

:root {

*/\* colours \*/*

    --white-color: #ffffff;

    --primary-color: #74c974;

    --secondary-color: #FFB6C1;

*/\* font sizes \*/*

    --font-size-s: 0.9rem;

    --font-size-n: 1rem;

    --font-size-m: 1.12rem;

    --font-size-l: 1.5rem;

    --font-size-xl: 2rem;

    --font-size-xxl: 2.3rem;

*/\* font weights \*/*

    --font-weight-normal: 400;

    --font-weight-medium: 500;

    --font-weight-semibold: 600;

    --font-weight-bold: 700;

*/\* border radius \*/*

    --border-radius-s: 8px;

    --border-radius-m: 30px;

    --border-radius-circle: 50%;

*/\* site max width \*/*

    --site-max-width: 2300px;

}

body {

    margin: 0;

    overflow-x: hidden;

    background: var(--white-color);

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

}

*/\* header and navigation bar \*/*

header {

    width: 100%;

    background: var(--primary-color);

    position: fixed;

    top: 0;

    left: 0;

    z-index: 1000;

}

.navbar {

    width: 100%;

    max-width: var(--site-max-width);

    margin: 0 auto;

    padding: 15px 20px;

    display: flex;

    justify-content: space-between;

    align-items: center;

}

.navbar .nav-logo .logo-text {

    color: var(--white-color);

    font-size: var(--font-size-xl);

    font-weight: var(--font-weight-semibold);

}

.navbar .nav-menu {

    display: flex;

    gap: 10px;

}

.navbar .nav-menu .nav-link {

    padding: 10px 18px;

    color: var(--white-color);

    background: var(--secondary-color);

    border-radius: var(--border-radius-m);

    transition: 0.3s ease;

}

.navbar .nav-menu .nav-link:hover {

    color: var(--primary-color);

    background: var(--secondary-color);

}

*/\* Welcome Section \*/*

.welcome-section {

    width: 100%;

    min-height: 100vh;

    background: var(--white-color);

    display: flex;

    justify-content: center;

    align-items: center;

    margin-top: 80px;

    padding: 40px 20px;

    box-sizing: border-box;

}

.welcome-section .section-content {

    width: 100%;

    max-width: var(--site-max-width);

    text-align: center;

}

.welcome-details {

    font-size: var(--font-size-xxl);

    color: var(--secondary-color);

    margin-bottom: 40px;

}

.welcome-details .title {

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

    margin-bottom: 20px;

    font-size: var(--font-size-xxl);

}

.welcome-details .subtitle {

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

    font-size: var(--font-size-xl);

}

*/\* ===== DaShBoarD ElEmentS ===== \*/*

.dashboard-grid {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

    gap: 30px;

    margin-top: 40px;

    text-align: left;

}

.dashboard-card {

    background: var(--white-color);

    padding: 30px;

    border-radius: var(--border-radius-m);

    box-shadow: 0 5px 15px rgba(0,0,0,0.05);

    transition: transform 0.3s ease;

}

.dashboard-card:hover {

    transform: translateY(-5px);

}

.dashboard-card h3 {

    color: var(--primary-color);

    font-size: var(--font-size-l);

    margin-bottom: 20px;

    font-weight: var(--font-weight-semibold);

}

.dashboard-card p {

    color: #555;

    font-size: var(--font-size-m);

    margin: 10px 0;

}

.dashboard-card p strong {

    color: var(--primary-color);

    font-weight: var(--font-weight-medium);

}

.action-btn {

    display: inline-block;

    background: var(--primary-color);

    color: var(--white-color);

    padding: 12px 25px;

    margin: 15px 10px 0 0;

    border-radius: var(--border-radius-m);

    font-size: var(--font-size-m);

    font-weight: var(--font-weight-medium);

    text-decoration: none;

    transition: all 0.3s ease;

}

.action-btn:hover {

    background: var(--secondary-color);

    transform: translateY(-2px);

    box-shadow: 0 4px 8px rgba(0,0,0,0.1);

}

*/\* adjustments \*/*

@media (max-width: 768px) {

    .dashboard-grid {

        grid-template-columns: 1fr;

    }

    .welcome-details .title {

        font-size: var(--font-size-xl);

    }

    .welcome-details .subtitle {

        font-size: var(--font-size-l);

    }

    .action-btn {

        display: block;

        width: 100%;

        margin: 10px 0;

    }

}  
  
new\_entry css   
*/\* ===== main layout ===== \*/*

.entry-container {

    width: 100%;

    min-height: calc(100vh - 80px);

    padding: 40px 20px;

    background: linear-gradient(to bottom, #f5fff5 0%, #e0f0e0 100%);

    display: flex;

    align-items: center;

    justify-content: center;

}

*/\* ===== entry card ===== \*/*

.entry-card {

    background: white !important;

    width: 100%;

    max-width: 700px;

    padding: 40px;

    border-radius: 16px;

    box-shadow:

        0 10px 30px rgba(116, 201, 116, 0.15),

        0 5px 15px rgba(0, 0, 0, 0.05);

    border: 1px solid rgba(116, 201, 116, 0.2);

    animation: fadeIn 0.5s ease-out;

}

@keyframes fadeIn {

    from { opacity: 0; transform: translateY(20px); }

    to { opacity: 1; transform: translateY(0); }

}

*/\* ===== typrography ===== \*/*

.entry-card h1 {

    color: #5aaf5a;

    font-size: 2.2rem;

    text-align: center;

    margin-bottom: 30px;

    position: relative;

    padding-bottom: 15px;

    font-weight: 600;

}

.entry-card h1::after {

    content: "";

    position: absolute;

    bottom: 0;

    left: 50%;

    transform: translateX(-50%);

    width: 80px;

    height: 3px;

    background: #FFB6C1;

    border-radius: 3px;

}

*/\* ===== font styles ===== \*/*

.entry-form {

    display: flex;

    flex-direction: column;

    gap: 25px;

}

.form-group {

    display: flex;

    flex-direction: column;

    gap: 8px;

    position: relative;

}

.form-group label {

    color: #5a8f5a;

    font-weight: 500;

    display: flex;

    align-items: center;

    gap: 10px;

    font-size: 1.1rem;

}

.form-group input,

.form-group textarea {

    padding: 14px 16px;

    border: 2px solid #d0e8d0;

    border-radius: 10px;

    font-family: "Poppins", sans-serif;

    font-size: 1rem;

    transition: all 0.3s ease;

}

.form-group textarea {

    min-height: 250px;

    resize: vertical;

    line-height: 1.6;

}

.form-group input:focus,

.form-group textarea:focus {

    border-color: #74c974;

    box-shadow: 0 0 0 3px rgba(116, 201, 116, 0.2);

    outline: none;

}

*/\* ===== button styles ===== \*/*

.form-actions {

    display: flex;

    gap: 15px;

    justify-content: center;

    margin-top: 30px;

}

.btn-submit {

    background: #74c974;

    color: white;

    box-shadow: 0 4px 15px rgba(116, 201, 116, 0.3);

    padding: 12px 28px;

    border-radius: 50px;

    font-weight: 500;

    border: none;

    cursor: pointer;

    transition: all 0.3s ease;

    font-size: 1rem;

}

.btn-cancel {

    background: #FFB6C1;

    color: white;

    box-shadow: 0 4px 15px rgba(255, 182, 193, 0.3);

    padding: 12px 28px;

    border-radius: 50px;

    text-decoration: none;

    text-align: center;

    transition: all 0.3s ease;

    font-size: 1rem;

}

.btn-submit:hover,

.btn-cancel:hover {

    transform: translateY(-2px);

    box-shadow: 0 6px 20px rgba(116, 201, 116, 0.4);

}

*/\* =====  ===== \*/*

.form-group.invalid input,

.form-group.invalid textarea {

    border-color: #ff6b6b;

    background-color: #fff5f5;

}

.form-group.valid input,

.form-group.valid textarea {

    border-color: #74c974;

}

*/\* ===== responsice design ===== \*/*

@media (max-width: 768px) {

    .entry-card {

        padding: 30px 20px;

    }

    .entry-card h1 {

        font-size: 1.8rem;

    }

    .form-actions {

        flex-direction: column;

        gap: 12px;

    }

    .btn-submit,

    .btn-cancel {

        width: 100%;

        padding: 14px;

    }

}

*/\* ===== micro interactions ===== \*/*

.form-group input:not(:placeholder-shown),

.form-group textarea:not(:placeholder-shown) {

    background: #f9f9f9;

}

.btn-submit:active,

.btn-cancel:active {

    transform: translateY(1px);

    box-shadow: 0 2px 10px rgba(116, 201, 116, 0.3);

}

*/\* ===== acessibilty ===== \*/*

.form-group input:focus-visible,

.form-group textarea:focus-visible {

    outline: 2px solid #5aaf5a;

    outline-offset: 2px;

}

Entries CSS  
:root {

*/\* colours\*/*

    --white-color: #ffffff;

    --primary-color: #74c974;

    --secondary-color: #FFB6C1;

*/\* font size\*/*

    --font-size-s:0.9rem;

    --font-size-n:1rem;

    --font-size-m:1.12rem;

    --font-size-1:1.5rem;

    --font-size-l:1.5rem;

    --font-size-xl:2rem;

    --font-size-xxl:2.3rem;

*/\* font weight\*/*

    --font-weight-normal: 400;

    --font-weight-medium:500;

    --font-weight-semibold:600;

    --font-weight-bold:700;

*/\*border radies\*/*

    --border-radius-s:8px;

    --border-radius-m:30px;

    --border-radius-circle:50%;

*/\*site max width\*/*

    --border-radius-s:8px;

    --border-radius-m:30px;

    --border-radius-circle:50%;

*/\*site max width\*/*

    --site-max-width: 2300px

}

*/\* ===== main layout ===== \*/*

.entries-container {

    width: 100%;

    min-height: calc(100vh - 80px);

    padding: 40px 20px;

    background: linear-gradient(to bottom, #f5fff5 0%, #e0f0e0 100%);

}

*/\* make header full width \*/*

header {

    width: 100%;

    background: var(--primary-color);

    position: fixed;

    top: 0;

    left: 0;

    z-index: 1000;

}

header {

    background: var(--primary-color);

}

header .navbar{

    display: flex;

    padding: 20px;

    align-items: center;

    justify-content: space-between;

}

.navbar .nav-logo .logo-text{

    color: var(--white-color);

    font-size: var(--font-size-xl);

    font-weight: var(--font-weight-semibold);

}

.navbar .nav-menu{

    display: flex;

    gap: 10px;

}

.navbar .nav-menu .nav-link{

    padding: 10px 18px;

    color: var(--white-color);

    background: var(--secondary-color);

    border-radius: var(--border-radius-m);

    transition: 0.3 ease;

}

.navbar .nav-menu .nav-link:hover{

    color: var(--primary-color);

    background: var(--secondary-color);

}

*/\* ===== header styles ===== \*/*

.entries-header {

    text-align: center;

    margin-bottom: 40px;

    animation: fadeIn 0.5s ease-out;

}

.entries-header h1 {

    color: #5aaf5a;

    font-size: 2.2rem;

    margin-bottom: 10px;

    font-weight: 600;

}

.entries-header p {

    color: #7a9e7a;

    font-size: 1.1rem;

    margin-bottom: 20px;

}

.search-container {

    max-width: 500px;

    margin: 0 auto;

    position: relative;

}

.search-container ion-icon {

    position: absolute;

    left: 15px;

    top: 50%;

    transform: translateY(-50%);

    color: #7a9e7a;

}

#searchEntries {

    width: 100%;

    padding: 12px 20px 12px 45px;

    border: 2px solid #d0e8d0;

    border-radius: 50px;

    font-family: "Poppins", sans-serif;

    font-size: 1rem;

    transition: all 0.3s ease;

}

#searchEntries:focus {

    border-color: #74c974;

    box-shadow: 0 0 0 3px rgba(116, 201, 116, 0.2);

    outline: none;

}

*/\* ===== entries list ===== \*/*

.entries-list {

    max-width: 900px;

    margin: 0 auto;

    display: grid;

    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));

    gap: 25px;

    padding: 20px;

}

.entry-card {

    background: white;

    border-radius: 12px;

    box-shadow:

        0 5px 15px rgba(116, 201, 116, 0.1),

        0 3px 8px rgba(0, 0, 0, 0.05);

    border: 1px solid rgba(116, 201, 116, 0.15);

    padding: 25px;

    transition: all 0.3s ease;

    animation: fadeIn 0.5s ease-out;

    position: relative;

    overflow: hidden;

}

.entry-card:hover {

    transform: translateY(-5px);

    box-shadow:

        0 8px 25px rgba(116, 201, 116, 0.2),

        0 5px 15px rgba(0, 0, 0, 0.08);

}

.entry-card h2 {

    color: #5aaf5a;

    font-size: 1.4rem;

    margin-bottom: 10px;

    padding-bottom: 10px;

    border-bottom: 2px solid #FFB6C1;

}

.entry-card .entry-date {

    color: #7a9e7a;

    font-size: 0.9rem;

    margin-bottom: 15px;

    display: flex;

    align-items: center;

    gap: 5px;

}

.entry-card .entry-content {

    color: #555;

    line-height: 1.6;

    margin-bottom: 20px;

    display: -webkit-box;

    -webkit-line-clamp: 5;

    -webkit-box-orient: vertical;

    overflow: hidden;

    text-overflow: ellipsis;

}

.entry-card .entry-actions {

    display: flex;

    gap: 10px;

}

.entry-card .btn-read,

.entry-card .btn-delete {

    padding: 8px 15px;

    border-radius: 50px;

    font-size: 0.9rem;

    border: none;

    cursor: pointer;

    transition: all 0.2s ease;

    display: flex;

    align-items: center;

    gap: 5px;

}

.entry-card .btn-read {

    background: #74c974;

    color: white;

}

.entry-card .btn-delete {

    background: #FFB6C1;

    color: white;

}

.entry-card .btn-read:hover,

.entry-card .btn-delete:hover {

    opacity: 0.9;

    transform: translateY(-1px);

}

.empty-state {

    grid-column: 1 / -1;

    text-align: center;

    padding: 50px 20px;

    color: #7a9e7a;

}

.empty-state ion-icon {

    font-size: 3rem;

    margin-bottom: 20px;

    color: #d0e8d0;

}

.empty-state h2 {

    color: #5a8f5a;

    margin-bottom: 15px;

}

.empty-state p {

    margin-bottom: 25px;

}

.empty-state .btn-new-entry {

    padding: 12px 25px;

    background: #74c974;

    color: white;

    border-radius: 50px;

    text-decoration: none;

    display: inline-flex;

    align-items: center;

    gap: 8px;

    transition: all 0.3s ease;

}

.empty-state .btn-new-entry:hover {

    background: #5aaf5a;

    transform: translateY(-2px);

    box-shadow: 0 5px 15px rgba(116, 201, 116, 0.3);

}

*/\* modal styles \*/*

.modal-overlay {

    position: fixed;

    top: 0;

    left: 0;

    right: 0;

    bottom: 0;

    background: rgba(0, 0, 0, 0.5);

    display: flex;

    align-items: center;

    justify-content: center;

    z-index: 1000;

    opacity: 0;

    visibility: hidden;

    transition: all 0.3s ease;

}

.modal-overlay.active {

    opacity: 1;

    visibility: visible;

}

.modal-content {

    background: white;

    width: 90%;

    max-width: 700px;

    max-height: 90vh;

    border-radius: 16px;

    padding: 30px;

    position: relative;

    overflow-y: auto;

    transform: translateY(20px);

    transition: all 0.3s ease;

}

.modal-overlay.active .modal-content {

    transform: translateY(0);

}

.modal-close {

    position: absolute;

    top: 20px;

    right: 20px;

    background: none;

    border: none;

    font-size: 1.5rem;

    cursor: pointer;

    color: #7a9e7a;

}

.modal-title {

    color: #5aaf5a;

    margin-bottom: 10px;

    padding-right: 30px;

}

.modal-date {

    color: #7a9e7a;

    margin-bottom: 20px;

}

.modal-text {

    line-height: 1.8;

    color: #333;

    white-space: pre-wrap;

}

*/\* animation \*/*

@keyframes fadeIn {

    from { opacity: 0; transform: translateY(10px); }

    to { opacity: 1; transform: translateY(0); }

}

*/\* Responsive Design \*/*

@media (max-width: 768px) {

    .entries-list {

        grid-template-columns: 1fr;

    }

    .entries-header h1 {

        font-size: 1.8rem;

    }

}

student support css

@import url('https://fonts.googleapis.com/css2?family=Poppins&family=Miniver&display=swap');

:root {

*/\* colours \*/*

    --white-color: #ffffff;

    --primary-color: #74c974;

    --secondary-color: #FFB6C1;

*/\* font sizes \*/*

    --font-size-s: 0.9rem;

    --font-size-n: 1rem;

    --font-size-m: 1.12rem;

    --font-size-l: 1.5rem;

    --font-size-xl: 2rem;

    --font-size-xxl: 2.3rem;

*/\* font weights \*/*

    --font-weight-normal: 400;

    --font-weight-medium: 500;

    --font-weight-semibold: 600;

    --font-weight-bold: 700;

*/\* border radius \*/*

    --border-radius-s: 8px;

    --border-radius-m: 30px;

    --border-radius-circle: 50%;

*/\* site max width \*/*

    --site-max-width: 2300px;

}

body {

    margin: 0;

    overflow-x: hidden;

    background: var(--white-color);

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

}

*/\* header and navigation bar \*/*

header {

    width: 100%;

    background: var(--primary-color);

    position: fixed;

    top: 0;

    left: 0;

    z-index: 1000;

}

.navbar {

    width: 100%;

    max-width: var(--site-max-width);

    margin: 0 auto;

    padding: 15px 20px;

    display: flex;

    justify-content: space-between;

    align-items: center;

}

.navbar .nav-logo .logo-text {

    color: var(--white-color);

    font-size: var(--font-size-xl);

    font-weight: var(--font-weight-semibold);

}

*/\* pink buttons for navigation links \*/*

.navbar .nav-menu {

    display: flex;

    gap: 10px;

}

.navbar .nav-menu .nav-link {

    padding: 10px 18px;

    color: var(--white-color);

    background: var(--secondary-color);

    border-radius: var(--border-radius-m);

    transition: 0.3s ease;

    text-decoration: none;

    font-weight: var(--font-weight-semibold);

}

.navbar .nav-menu .nav-link:hover {

    color: var(--primary-color);

    background: var(--secondary-color);

    cursor: pointer;

}

.support-container {

    margin-top: 80px;

    max-width: var(--site-max-width);

    margin-left: auto;

    margin-right: auto;

    padding: 20px;

    box-sizing: border-box;

    background: var(--white-color);

}

.support-hero {

    text-align: center;

    margin-bottom: 40px;

}

.support-hero h1 {

    font-size: var(--font-size-xxl);

    color: var(--primary-color);

    margin-bottom: 10px;

}

.support-hero p {

    font-size: var(--font-size-m);

    color: #444;

}

*/\* support grid layout \*/*

.support-grid {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(320px, 1fr));

    gap: 30px;

}

*/\* support card styling \*/*

.support-card {

    background: var(--white-color);

    padding: 30px;

    border-radius: var(--border-radius-m);

    box-shadow: 0 5px 15px rgba(0,0,0,0.05);

    transition: transform 0.3s ease;

}

.support-card:hover {

    transform: translateY(-5px);

}

*/\* section titles \*/*

.support-card h2 {

    color: var(--primary-color);

    font-size: var(--font-size-l);

    margin-bottom: 20px;

    font-weight: var(--font-weight-semibold);

    display: flex;

    align-items: center;

    gap: 10px;

}

*/\* consent toggle  \*/*

.consent-toggle label {

    display: flex;

    align-items: center;

    gap: 10px;

    font-weight: var(--font-weight-medium);

    cursor: pointer;

}

.consent-note {

    font-size: var(--font-size-s);

    color: #666;

    margin-top: 8px;

}

.option-grid {

    display: grid;

    grid-template-columns: repeat(auto-fit, minmax(140px, 1fr));

    gap: 20px;

}

.option-item {

    text-align: center;

    padding: 15px;

    background: #f9f9f9;

    border-radius: var(--border-radius-s);

    box-shadow: 0 2px 8px rgba(0,0,0,0.05);

}

.option-item ion-icon {

    font-size: 2rem;

    color: var(--primary-color);

    margin-bottom: 10px;

}

*/\* advisor notes \*/*

.notes-container {

    max-height: 200px;

    overflow-y: auto;

    margin-bottom: 20px;

}

.note {

    background: #f1f8f1;

    padding: 15px;

    border-radius: var(--border-radius-s);

    margin-bottom: 15px;

    font-size: var(--font-size-n);

    color: #333;

}

.note.new {

    background: var(--secondary-color);

    color: #fff;

}

.note-header {

    display: flex;

    justify-content: space-between;

    font-weight: var(--font-weight-semibold);

    margin-bottom: 8px;

}

.btn-view-all {

    background: var(--primary-color);

    color: var(--white-color);

    padding: 10px 25px;

    border: none;

    border-radius: var(--border-radius-m);

    font-weight: var(--font-weight-semibold);

    cursor: pointer;

    transition: background 0.3s ease;

}

.btn-view-all:hover {

    background: var(--secondary-color);

    color: var(--primary-color);

}

@media (max-width: 768px) {

    .support-grid {

        grid-template-columns: 1fr;

    }

    .option-grid {

        grid-template-columns: 1fr 1fr;

    }

}

# JavaScript code

Entries.js  
document.addEventListener('DOMContentLoaded', **function**() {

**const** entriesList = document.getElementById('entriesList');

**const** searchInput = document.getElementById('searchEntries');

**let** entries = JSON.parse(localStorage.getItem('journalEntries')) || [];

*// initial display*

    displayEntries(entries);

*// search functionality*

    if (searchInput) {

        searchInput.addEventListener('input', **function**() {

**const** searchTerm = this.value.toLowerCase();

**const** filteredEntries = entries.filter(entry **=>**

                entry.title.toLowerCase().includes(searchTerm) ||

                entry.content.toLowerCase().includes(searchTerm)

            );

            displayEntries(filteredEntries);

        });

    }

*// event delegation for entry actions*

    entriesList.addEventListener('click', **function**(e) {

**const** entryCard = e.target.closest('.entry-card');

        if (!entryCard) return;

**const** entryId = parseInt(entryCard.dataset.id);

*// read button*

        if (e.target.classList.contains('btn-read')) {

            e.preventDefault();

            showEntryModal(entryId);

        }

*// delete button*

        if (e.target.classList.contains('btn-delete')) {

            deleteEntry(entryId);

        }

    });

});

*// display all entries*

**function** displayEntries(entries) {

**const** entriesList = document.getElementById('entriesList');

    if (entries.length === 0) {

        entriesList.innerHTML = `

            <div class="empty-state">

                <ion-icon name="journal-outline"></ion-icon>

                <h2>No Entries Yet</h2>

                <p>Start writing your first journal entry to see it appear here.</p>

                <a href="new entry.html" class="btn-new-entry">  <!-- CHANGED THIS LINE -->

                    <ion-icon name="add-outline"></ion-icon>

                    Create New Entry

                </a>

            </div>

        `;

        return;

    }

    entriesList.innerHTML = entries.map(entry **=>** `

        <div class="entry-card" data-id="${entry.id}">

            <h2>${entry.title}</h2>

            <div class="entry-date">

                <ion-icon name="calendar-outline"></ion-icon>

                ${entry.date}

            </div>

            <div class="entry-content">

                ${entry.content.length > 150 ?

                    entry.content.substring(0, 150) + '...' :

                    entry.content}

            </div>

            <div class="entry-actions">

                <button class="btn-read">

                    <ion-icon name="eye-outline"></ion-icon>

                    Read

                </button>

                <button class="btn-delete">

                    <ion-icon name="trash-outline"></ion-icon>

                    Delete

                </button>

            </div>

        </div>

    `).join('');

}

Form\_validates.js  
document.addEventListener('DOMContentLoaded', **function**() {

**const** form = document.querySelector('form');

    form.addEventListener('submit', **function**(e) {

**const** content = document.getElementById('entry\_content').value.trim();

        if (content.length < 20) {

            e.preventDefault();

            alert('Content should be at least 20 characters long.');

            return false;

        }

        return true;

    });

});

Main.js

*// element selections*

**const** loginBtn = document.querySelector("#Login");

**const** registerBtn = document.querySelector("#Register");

**const** loginForm = document.querySelector(".login-form");

**const** registerForm = document.querySelector(".register-form");

*// password field elements*

**const** logInputField = document.getElementById('logPassword');

**const** logInputIcon = document.getElementById('log-pass-icon');

**const** regInputField = document.getElementById('regPassword');

**const** regInputIcon = document.getElementById('reg-pass-icon');

*// form toggle functionality*

loginBtn.addEventListener('click', () **=>** {

    loginBtn.style.backgroundColor = "#74c974";

    registerBtn.style.backgroundColor = "rgba(255,255,255,0.2)";

    loginForm.style.left = "50%";

    registerForm.style.left = "-50%";

    loginForm.style.opacity = 1;

    registerForm.style.opacity = 0;

});

registerBtn.addEventListener('click', () **=>** {

    loginBtn.style.backgroundColor = "rgba(255,255,255,0.2)";

    registerBtn.style.backgroundColor = "#74c974";

    loginForm.style.left = "150%";

    registerForm.style.left = "50%";

    loginForm.style.opacity = 0;

    registerForm.style.opacity = 1;

});

*// password visibility toggle functions*

**function** myLogPassword() {

    if(logInputField.type == "password") {

        logInputField.type = "text";

        logInputIcon.name = "eye-off-outline";

    } else {

        logInputField.type = "password";

        logInputIcon.name = "eye-outline";

    }

    logInputIcon.style.cursor = "pointer";

}

**function** myRegPassword() {

    if(regInputField.type == "password") {

        regInputField.type = "text";

        regInputIcon.name = "eye-off-outline";

    } else {

        regInputField.type = "password";

        regInputIcon.name = "eye-outline";

    }

    regInputIcon.style.cursor = "pointer";

}

*// change icon when typing*

**function** changeIcon(value) {

    if(value.length > 0) {

        logInputIcon.name = "eye-outline";

        regInputIcon.name = "eye-outline";

    } else {

        logInputIcon.name = "lock-closed-outline";

        regInputIcon.name = "lock-closed-outline";

    }

}

*// form submission handler*

document.querySelector('.login-form').addEventListener('submit', **function**(e) {

*// Get form values*

**const** username = this.querySelector('input[type="text"]').value;

**const** password = this.querySelector('input[type="password"]').value;

*// Simple validation*

    if(!username || !password) {

        e.preventDefault(); *// Stop form submission*

        alert('Please enter both username and password');

    }

*// If valid, form will submit to welcome.html automatically*

});

New entry.js

document.getElementById('entryForm').addEventListener('submit', **function**(e) {

    e.preventDefault();

**const** title = document.getElementById('entryTitle').value;

**const** content = document.getElementById('entryContent').value;

*// create entry object*

**const** entry = {

        id: Date.now(), *// unique ID*

        title,

        content,

        date: new Date().toLocaleDateString('en-GB', {

            day: 'numeric',

            month: 'short',

            year: 'numeric'

        }),

        timestamp: new Date().toISOString() *// For sorting*

    };

*// get existing entries or create empty array*

**let** entries = JSON.parse(localStorage.getItem('journalEntries')) || [];

*// Add new entry at beginning (newest first)*

    entries.unshift(entry);

*// save to localStorage*

    localStorage.setItem('journalEntries', JSON.stringify(entries));

*// reset form*

    this.reset();

*// redirect to entries page*

    window.location.href = 'entries.html';

});

document.getElementById('entryContent').addEventListener('input', **function**() {

**const** charCount = this.value.length;

    console.log(`Characters: ${charCount}`); *// can display this in UI*

});