National College of Ireland

Bachelor of Science (Honours) in Computing

Software Development

2021/2022

Tegan Jennings

X18303941

X18303941@student.ncirl.ie

Student Engagement Portal

Technical Report

Contents

Executive Summary

This technical report outlines the development of my final year project, Student Engagement portal, with the main focus on project specifications, requirements and implementation.

This report highlights the importance of tracking student engagement and retention from learning platforms in order to identify students who are at risk of failing behind early on. Student engagement is a huge problem in both public and private universities, when the student engagement drops, the student retention also drops. There are three main steps to prevent low engagement and retention, they are described as: (Anon., 2016)

- Early Identification
- Early Intervention
- Continuous improvement

The student engagement portal centres around these steps in order to best support at risk students. At risk students can be identified at the glance using the Student Engagement Dashboard. Users can intervene if they identify an at-risk student. They can intervene by emailing students through the Portal. Lecturers can improve activities to keep students engaged whilst adapting to the engagement.

The student engagement portal will be very useful and important especially as online learning becomes more popular. The student engagement portal allows for authorised users to easily understand the student engagement for their module in a glance, they can also email students who are falling behind and not completing content.

The student engagement portal allows for authorised users to easily understand the student engagement for their module in a glance, they can also email students who are falling behind and not completing content. This works by allowing users to upload an excel file. Once the excel is uploaded, the information from the excel can be seen on a unique page created. On this unique dashboard page, users will see graphs and charts of the data from the excel. Users will also see a table where they can email students who are falling behind.

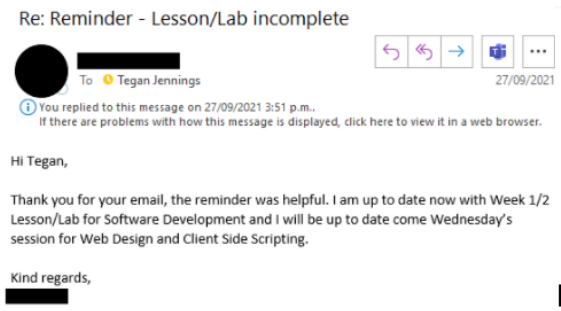
The Student Engagement Portal also has an admin portal where reports, modules and courses can be customised as well as users' information.

1.0 Introduction

1.1. Background

One of my duties as online learning support in the National College of Ireland is to track student engagement and retention on directed eLearning modules each semester. In order to track student engagement and retention, I would manually go through each module and check if students completed their required learning content for that week, and if any students didn't complete it, I would email them a a reminder/support email. However, this process was extremely time consuming, and I was left with no time to complete my other duties.

After all the positive feedback I got from students who I supported after reaching out to them based on my findings when tracking student engagement and retention, an example can be seen below, I knew that this was a very important task to complete weekly, however was too time consuming.



Re: Reminder - Lesson/Lab incomplete

[redacted]

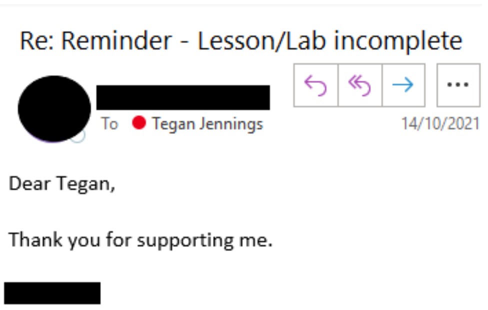
To Tegan Jennings 27/09/2021

You replied to this message on 27/09/2021 3:51 p.m..
If there are problems with how this message is displayed, click here to view it in a web browser.

Hi Tegan,

Thank you for your email, the reminder was helpful. I am up to date now with Week 1/2 Lesson/Lab for Software Development and I will be up to date come Wednesday's session for Web Design and Client Side Scripting.

Kind regards,
[redacted]



Re: Reminder - Lesson/Lab incomplete

[redacted]

To Tegan Jennings 14/10/2021

Dear Tegan,

Thank you for supporting me.

[redacted]

Because this task was very important but time consuming, I got to brainstorming and came up with a student retention portal that would be semi-automated and could perform all the tasks needed in a matter of minutes, rather than hours.

I got the motivation to develop a portal where I could drop in files and a dashboard with charts and graphs would display the student's engagement for a particular module, and from that portal I could then email any students who hasn't completed the required content at that point.

As online learning becomes more popular, we found it harder to track student's engagements in modules as when classes are held on a college campus, lecturers can see whose attending class and can monitor work being completed. However, in an online setting, the responsibility falls on the learner, to attend and complete the learning activities. On educational learning platform like Moodle, we are able to track very basically if a student has completed an activity or not and from that we can download it and then go through each student and email them.

1.2. Aims

The objective of my project the Student Engagement Portal is to give lecturers and admin staff in colleges and universities an application where they can monitor student engagement and identify and intervene with at risk students quickly.

My main aim for this project is to develop a web-based server that is independent of its data source. The server will then access mediated data and then display the data as charts and graphs in a dashboard. The server can then send generic reminder/support email to people, email can be seen in appendix 6.3.1. The main aims for the student engagement dashboard are the following:

- Web based server developed using PHP
- Application connected to a database
- Application accepts mediated data
- Application displays mediated data in a dashboard
- Dashboard can be filtered
- Students who are behind can be emailed

Each one of my aims for my project are extremely important. Useful functionality offered by the student engagement portal would be the emailing functionality. Users can view an engagement of a module and if a student is identified as behind, then an email can be sent out to the student offering support and guidance.

The Student Engagement Portal will also include an admin portal. In this admin portal, the courses, dashboards and modules can be managed. Admins can create new courses that will show up in the users' dropdowns when they are creating modules, delete courses which will delete all modules and dashboards associated with that course and deactivate courses so that they won't show up on the dropdown for users however the modules and dashboards are still on the database. Admin can also manage modules, they can create new modules, delete modules and activate or deactivate modules. Admin can also manage user accounts, they can create new users, update users and delete users.

The Student Engagement Portal will also create unique dashboards to display information given from an excel of csv file. Users will be able to view only the modules they created, and if they want to view a dashboard for one of those modules, they just have to click the module. Once the module is clicked, the dashboard will be displayed and the charts, tables and graphs will be populated with information from the file that was uploaded when that module was created.

The main focus of my project is to allow educational institutions to track student engagement and student retention on their online learning platforms, like Moodle for example. This Project will allow educational institutions to quickly and effectivity identity students who might be struggling with a module. If a student has been identified as possibility struggling, a reminder/support email can be sent out to the student

1.3. Technology

1.3.1 IntelliJ IDEA

IntelliJ IDEA is an integrated development environment (IDE) written in java. IntelliJ IDEA was developed by JetBrains. (Anon., n.d.)I am using IntelliJ IDEA as an IDE. IntelliJ IDEA allowed to create code, debug errors and design my project. With IntelliJ IDEA I was able to download plugins to run PHP files effectively. I downloaded and installed plugins like PHP, PhpClean and PHPUnit Enhancement.

PHPUnit is a plugin that provides users the ability to edit and debug, PHPUnit, Smarty, Twig and various frameworks support. I used the PHP plugin so that I could code PHP files. PhpClean is a plugin that allows for static Code Analysis for PhpStorm and IntelliJ Idea. PHPUnit Enhancement is a plugin that provides smart autocomplete, code navigation and refactoring features for mocked class methods of PHPUnit, Prophecy, and Mockery.

1.3.2 PHP

Hypertext Pre-processor (PHP) is known as a general-purpose scripting language. PHP is commonly used to develop dynamic and interactive websites. PHP can be embedded into HTML, which makes adding functionality to web pages simpler. (Anon., n.d.)

The student engagement portal is developed in the PHP language as the application is a web-based server and PHP was the language most suited to the project's needs. When looking at the project aims and objectives, such as, needing to upload external files to a database and then call those files in order to create the charts and graphs for the dashboard system and PHP has the ability to do that that.

1.3.3 HTML & CSS

The Hypertext Mark-up Language (HTML) is the standard mark-up language for documents. Cascading Style Sheets (CSS) is a style sheet language used for describing the style of documents that are written in HTML. (Anon., n.d.)

The student engagement portal is using HTML and CSS in order to design some of the main pages in the project. HTML will be used to create the structure of the web pages and the content within the pages whilst CSS will be used to create the design for those pages.

1.3.4 MySQL

MySQL is a relational database management system. Databases are used to store information. (Anon., n.d.) Information stored in a database can include any information that needs to be remembered such as a user's registration information, modules, courses and dashboards.

When I began my project, I started with a MongoDB database. MongoDB is a source-available cross-platform document-oriented database program. I had no experience using a MongoDB before and that made the implementation very difficult for me. Quickly I realised that MongoDB is not suitable for a PHP application as I was unable to seamlessly integrate the database into the application. I decided to change my database to a MySQL database using PhpMyAdmin. Once I changed to the MySQL database, I stopped having any problems with the database implementation. MySQL was the better database solution based on the applications needs.

1.3.5 PHPUnit

PHPUnit is a unit testing framework for PHP. (Anon., n.d.) PHPUnit will be used to create unit tests for the student engagement portal. Unit tests include multiple assertions, and these unit tests are the following:

- Login
- Modules
- Courses

1.3.6 GitHub

GitHub is a provider of hosting for software development projects and version control using Git. GitHub was chosen for the version control of the Student Engagement Portal. GitHub gives the ability to track all the changes made within a repository and check the history of each file in that repository. This is helpful because as the student engagement portal is being developed and features being added to the project, if something breaks, its quicker to identity where the issue may lie or roll back the application to a previous version.

1.3.7 XAMPP

XAMPP is an open-source cross-platform web server solution stack package. XAMPP includes interpreters for scripts written in the PHP programming language. (Anon., n.d.)

The student engagement portal uses XAMPP, as XAMPP allows PHP Script to run and be shown in localhost. This is one of the only ways to view PHP scripts locally.

1.4. Structure

This technical document is structured around the requirements, design and graphical user interface of my project. This technical document structure is as follows:

1. Introduction - The first part this report was the background and overview of technologies used for the project.
2. Requirements - The requirements section is split up into multiple different sections, these sections are:
    a. Functional Requirements - Functional requirements purpose is to detail what the system will accomplish.
    b. Data Requirements – Pre-processing and maintenance of data throughout projects implementation.
    c. User Requirements – How the project meets the needs and requirements for end users.
    d. Environmental Requirements - Outlines projects visual or auditory deficits.
    e. Usability Requirements - Outlines the efficiency of use, Intuitiveness, Ease of use for the project.
2. Design & Architecture – Details the design, system architecture and components used in the project.
3. Implementation – Outlines the development approaches and implementation of main algorithms/classes/functions used in the code taken to achieve project aims and goals.
4. Graphical User Interface (GUI) – Details the GUI key screens using screenshots and explain what can be seen in each one.
5. Conclusion – Conclusion of project, positive and negatives of the final project.
6. Further Development or Research – What would I do if I had more time available to do this project.
7. References – Any references used in this project to support claims or statements.
8. Appendix – Supplementary external documents such as project proposal is included in the appendix.

2.0 System

2.1. Requirements

2.1.1 Functional Requirements

1. Login System

A fully implemented and working database that is incorporated into the Student Engagement Portal that allows for the reading from the database. The purpose of this database is to hold all user's login information as well as dashboards.

MySQL is the database of choice and is used to hold all information generated by the application. The reason for having accounts at all is to link user accounts to dashboards. Registration is not an option as only authorized personnel can access this portal. When a user logs into the Student Engagement Portal, they will be shown either the admin or user home page, based on the role that is stored in the database that they were given when their account was created.

2. Upload Files

Using PHP and MySQL, users will be able to upload CSV files to the application in order to generate a dashboard on engagement based on the data within that CSV file.

In this requirement, I have a checker to check if the file being uploaded is actually a CSV file. If the file isn't a CSV file, then the file won't be uploaded to the database. If it is a CSV file, then the application will check if the file already exists within the database, if it doesn't, then the application will check the file size to make sure it's not equal to zero. If the file is a CSV, doesn't already exist and the size file isn't zero, then the file will be uploaded to the MongoDB database and the dashboard will be generated.

3. View and Generate Dashboard

To view and generate dashboard the main requirement for this project, in order to generate the dashboard, we need to be able to read csv files, and then generate charts from those CSVs and list off the students in the CSVs to allow for the emailing functionality to work. Once a file is uploaded, a dashboard should generate, and from that dashboard generating, charts and lists should generate

4. Email Students

Emailing students is an important part of student engagement. This requirement will allow users to specifically choose students from the dashboard to send a reminder/support email to, email can be seen in appendix 6.3.1.

2.1.2 Use Case Diagrams

2.1.2.1 Requirement 1: User Login

2.1.2.1.1 Description & Priority

The User is able to log into the portal and can reset password if needed. This is a crucial feature and has high priority.

2.1.2.1.2 Use Case

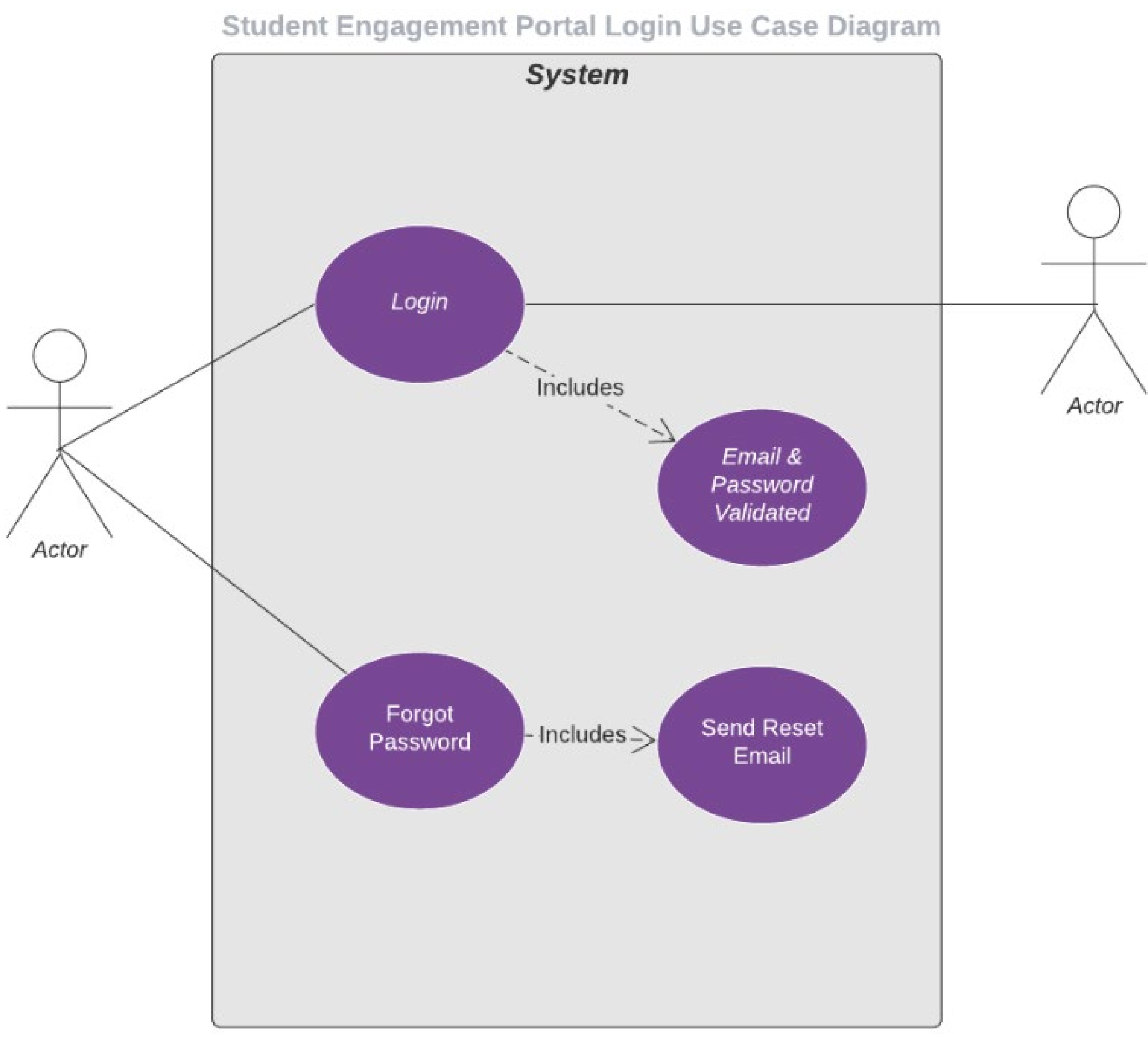
Scope

The scope of this use case is to allow users to log into the student engagement portal.

Description

This use case describes the actions taken by a user to log into the student engagement portal.

Use Case Diagram


Student Engagement Portal Login Use Case Diagram
System
Login
Includes
Email & Password Validated
Forgot Password
Includes
Send Reset Email
Actor
Actor

Flow Description

Precondition

The system is in initialisation mode, the user has received their login details and has navigated to the web-based server.

Activation

This use case starts when an <Actor> has navigated to the web-based server.

Main flow

1. The system displays the log in page.
2. The <Actor> enters in their email.
3. The <Actor> enters in their password.
4. The system validates the entered email and password and logs the actor into the system

Alternate flow

A1: Invalid Email/Password

1. The system cannot validate the entered email and password.
2. The system displays an error message.
3. The use case continues at position 2 of the main flow

Exceptional flow

N/A

Termination

The system presents the next webpage which is the homepage

Post condition

If use case was successful, the <Actor> is now logged into the system. If not, the system state is unchanged.

2.1.2.2 Requirement 2: User Uploads File

2.1.2.2.1 Description & Priority

The user will be able to upload a file and the file is verified. This is a crucial feature as the rest of the requirements cannot be completed without this requirement, therefor it has high priority.

2.1.2.2.2 Use Case

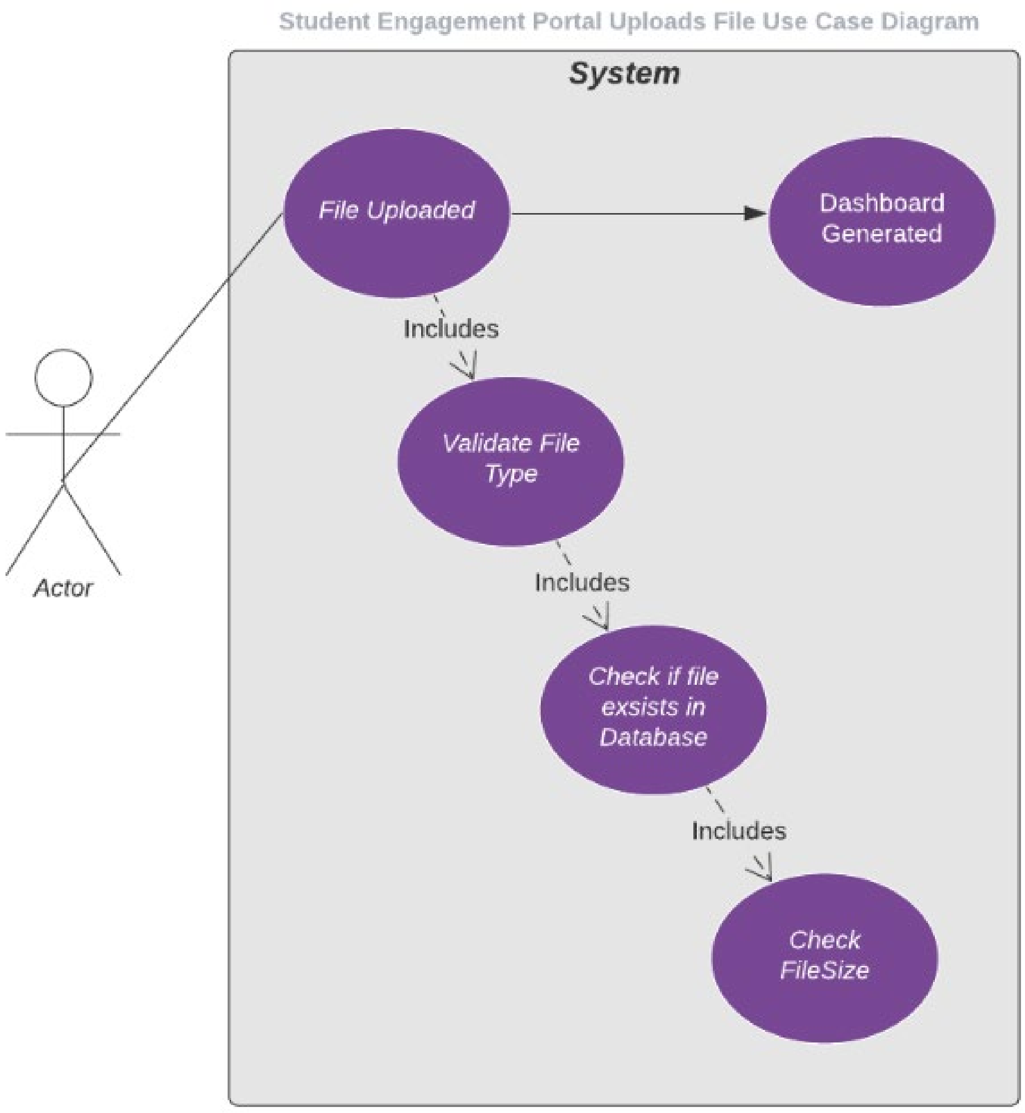
Scope

The scope of this use case is to allow users to upload a file the student engagement portal.

Description

This use case describes the actions taken by a user to upload a file.

Use Case Diagram



Flow Description

Precondition

The system has been logged into successful by an <Actor>.

Activation

This use case starts when an <Actor> is on the home page.

Main flow

1. The system displays the home page.
2. The <Actor> clicks the 'Choose File' button on the home page.
3. The <Actor> chooses a file to upload.
4. The <Actor> clicks the 'Submit' button.
5. The system validates the file uploaded.

Alternate flow

A1: File not validated

1. The system cannot valid the file submitted.
2. The system displays an error message.
3. The use case continues at position 2 of the main flow.

Exceptional flow

N/A

Termination

The system presents the generated dashboard page to <Actor>

Post condition

If use case was successful, the <Actor> can now navigate to a dashboard page on the system. If not the system state is unchanged.

2.1.2.3 Requirement 3: User Views Module Dashboard

2.1.2.3.1 Description & Priority

The user will be able to view generated dashboard page. This is high priority as it is the main functionality of this application.

2.1.2.3.2 Use Case

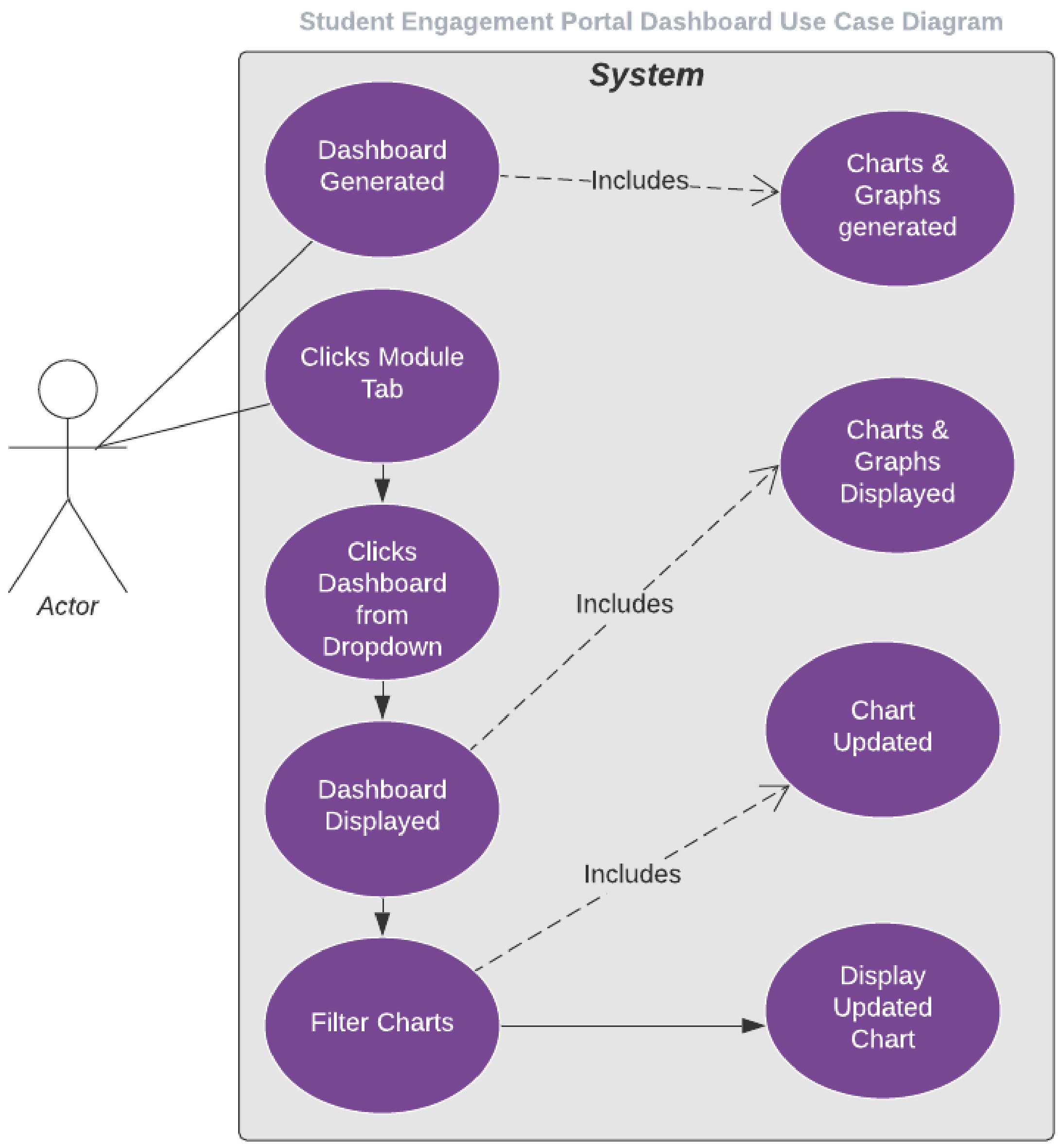
Scope

The scope of this use case is to allow users to view and filter generated dashboards.

Description

This use case describes the actions taken by a user to view a generated dashboard and filter the charts and graphs within that dashboard.

Use Case Diagram



Flow Description

Precondition

The system has successful verified a file uploaded by an <Actor>

Activation

This use case starts when an <Actor> uploads a file successfully to the system.

Main flow

1. The system identifies the file uploaded by <Actor>
2. The <Actor> clicks on the Modules tab in the top navigation bar
3. The system displays a dropdown of all dashboards available for the <Actor>
4. The <Actor> can see the new dashboard page has been created
5. The <Actor> clicks on the new dashboard page from the dropdown.
6. The system displays the new dashboard page.
7. The <Actor> clicks on a chart or graph in dashboard
8. The <Actor> filters the chart by student name, activity or week.
9. The system displays the filtered chart.

Alternate flow

A1: Dashboard not generated

1. The system failed to generate report.
2. The system reloads the home page.
3. The use case continues at position 2 of the main flow.

Exceptional flow

N/A

Termination

The system presents the charts and graphs on the Dashboard page.

Post condition

If use case was successful, the <Actor> can now view the dashboard page and can filter the graph. If not the system state is unchanged and <Actor> remains on the home page.

2.1.2.4 Requirement 5: User emails Student

2.1.2.4.1 Description & Priority

The user can choose to email students. This functionality has a high priority as this function is an important part of student engagement which is the support piece.

2.1.2.4.2 Use Case

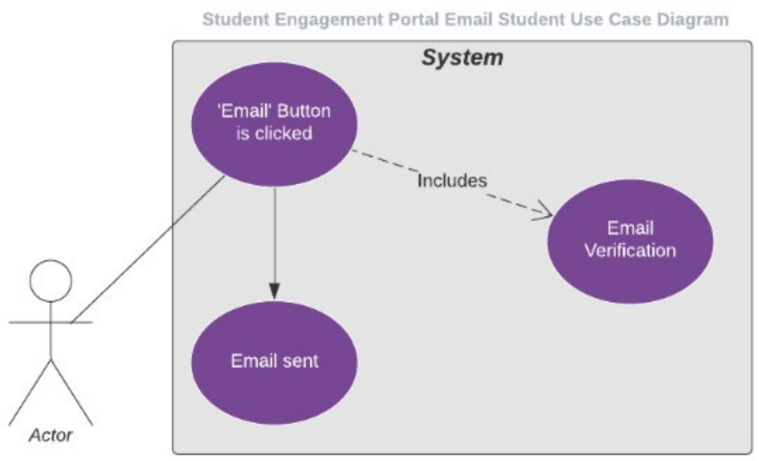
Scope

The scope of this use case is to allow users to choose students to send a reminder/support email to.

Description

This use case describes the actions taken by a user to send a reminder/support email.

Use Case Diagram


Student Engagement Portal Email Student Use Case Diagram
System
'Email' Button is clicked
Includes
Email Verification
Email sent
Actor

Flow Description

Precondition

The system has successful generated a dashboard and <Actor> has navigated to dashboard page.

Activation

This use case starts when an <Actor> navigates to a dashboard page.

Main flow

1. The system identifies that the <Actor> is on a Dashboard page.
2. The <Actor> views the list of students.
3. The <Actor> clicks on the 'Email' button corresponding to the student they want to send the reminder/support to.
4. The system identifies the student and retrieves their student email from the database.

5. The system then sends out the generic reminder/support email to the chosen student.
6. The system displays a success message.

Alternate flow

A1: Email not sent

1. The system cannot retrieve the students email from the database.
2. The system displays an error message.

Exceptional flow

N/A

Termination

The system presents a success or error message.

Post condition

If use case was successful, the <Actor> will see a success message pop up on the system. If not, the system state is unchanged, and an error message is displayed.

2.1.2.5 Requirement 6: User management

2.1.2.5.1 Description & Priority

The user can choose to view users, edit, add or delete courses. This feature has a high priority as this function is an important part of student engagement because users cannot sign up to the Student Engagement Portal.

2.1.2.5.2 Use Case

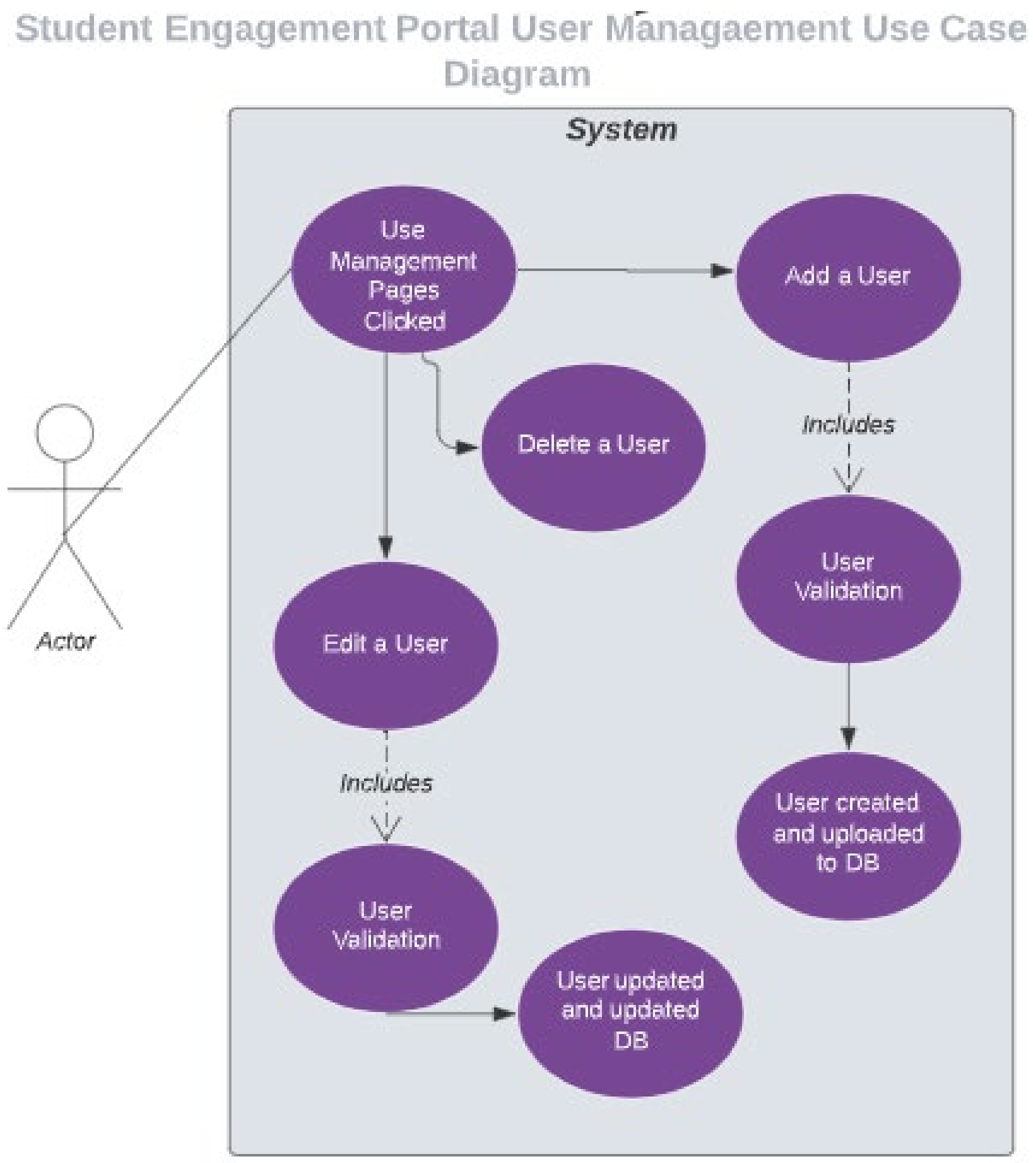
Scope

The scope of this use case is to allow users to view, delete, create and edit users.

Description

This use case describes the actions taken by a user to create a new user.

Use Case Diagram


Student Engagement Portal User Managaement Use Case Diagram
System
Use Management Pages Clicked
Add a User
Delete a User
Includes
User Validation
Edit a User
Includes
User created and uploaded to DB
User Validation
User updated and updated DB
Actor

Flow Description

Precondition

The system has navigated to the User Management page and the <Actor> is logged in as an admin.

Activation

This use case starts when an <Actor> navigates to a user management page.

Main flow

1. The system identifies that the <Actor> is on a User Management page.
2. The <Actor> views the list of users.
3. The <Actor> clicks on the ‘Add User button.
4. The system displays the add user page.
5. The <Actor> adds in name, username, email, password and role to the form.
6. The system then checks if the username already exists in the database.
7. The system displays a success message.

Alternate flow

A1: Username already exists

1. The system cannot Add user to the database as username already exists.
2. The system displays an error message.
3. The use case continues at position 3 of the main flow.

Exceptional flow

N/A

Termination

The system presents a success or error message.

Post condition

If use case was successful, the <Actor> will see the new user in the user table. If not, the system state is unchanged, and an error message is displayed.

2.1.2.6 Requirement 5: Course

2.1.2.6.1 Description & Priority

The user can choose to view courses, edit, add or delete courses.

2.1.2.6.2 Use Case

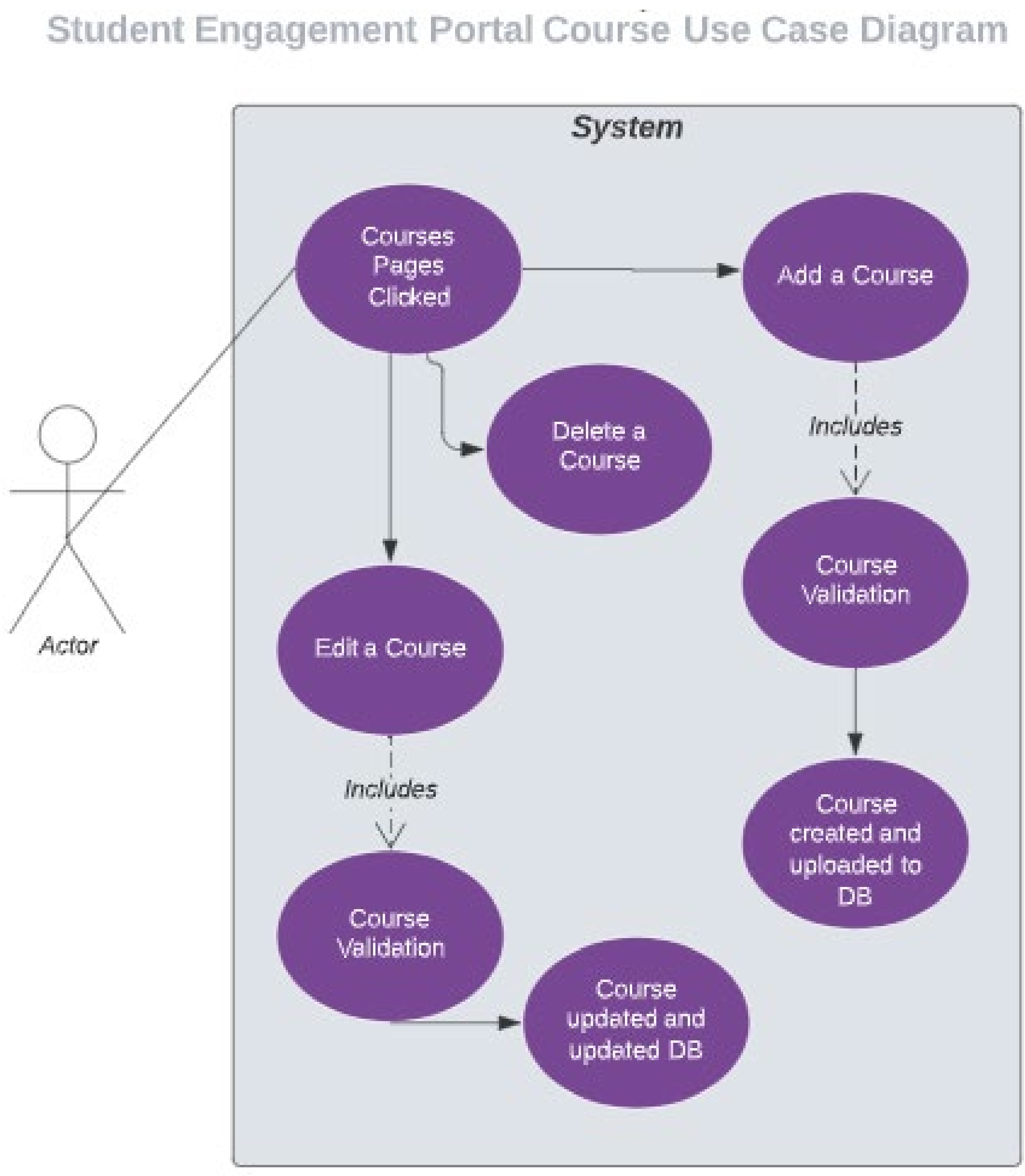
Scope

The scope of this use case is to allow users to view, delete, create and edit courses.

Description

This use case describes the actions taken by a user to create a new course.

Use Case Diagram



Flow Description

Precondition

The system has navigated to the Courses page and the <Actor> is logged in as an admin.

Activation

This use case starts when an <Actor> navigates to a course page.

Main flow

8. The system identifies that the <Actor> is on a Course page.
9. The <Actor> views the list of courses.
10. The <Actor> clicks on the ‘Add Module’ button.
11. The system displays the add course page.
12. The <Actor> adds in a course name and id.
13. The system then checks if the course ID already exists in the database.
14. The system displays a success message.

Alternate flow

A1: Course ID already exists

4. The system cannot Add course to the database as course ID already exists.
5. The system displays an error message.
6. The use case continues at position 3 of the main flow.

Exceptional flow

N/A

Termination

The system presents a success or error message.

Post condition

If use case was successful, the <Actor> will see the new course in the course table. If not, the system state is unchanged, and an error message is displayed.

2.1.2 Data Requirements

The data requirements for this project will be a csv file with students and lessons information in it. In this CSV, we would need to know the students' names, email addresses, the lessons they had to complete and if they completed them or not.

Information uploaded to the Student Engagement Portal will be stored on a MySQL database and will be linked to users using ids and foreign keys.

2.1.3 User Requirements

The user requirements for the Student Engagement Portal are that the user has an account and must be connected to the interview via WI-FI.

2.1.4 Environmental Requirements

N/A

2.1.5 Usability Requirements

When it comes to usability requires, the main requirement was to have the interface to be easy to use and understand. My project will follow an aesthetically pleasing colour palette, which will make the application easy to look at and understand.

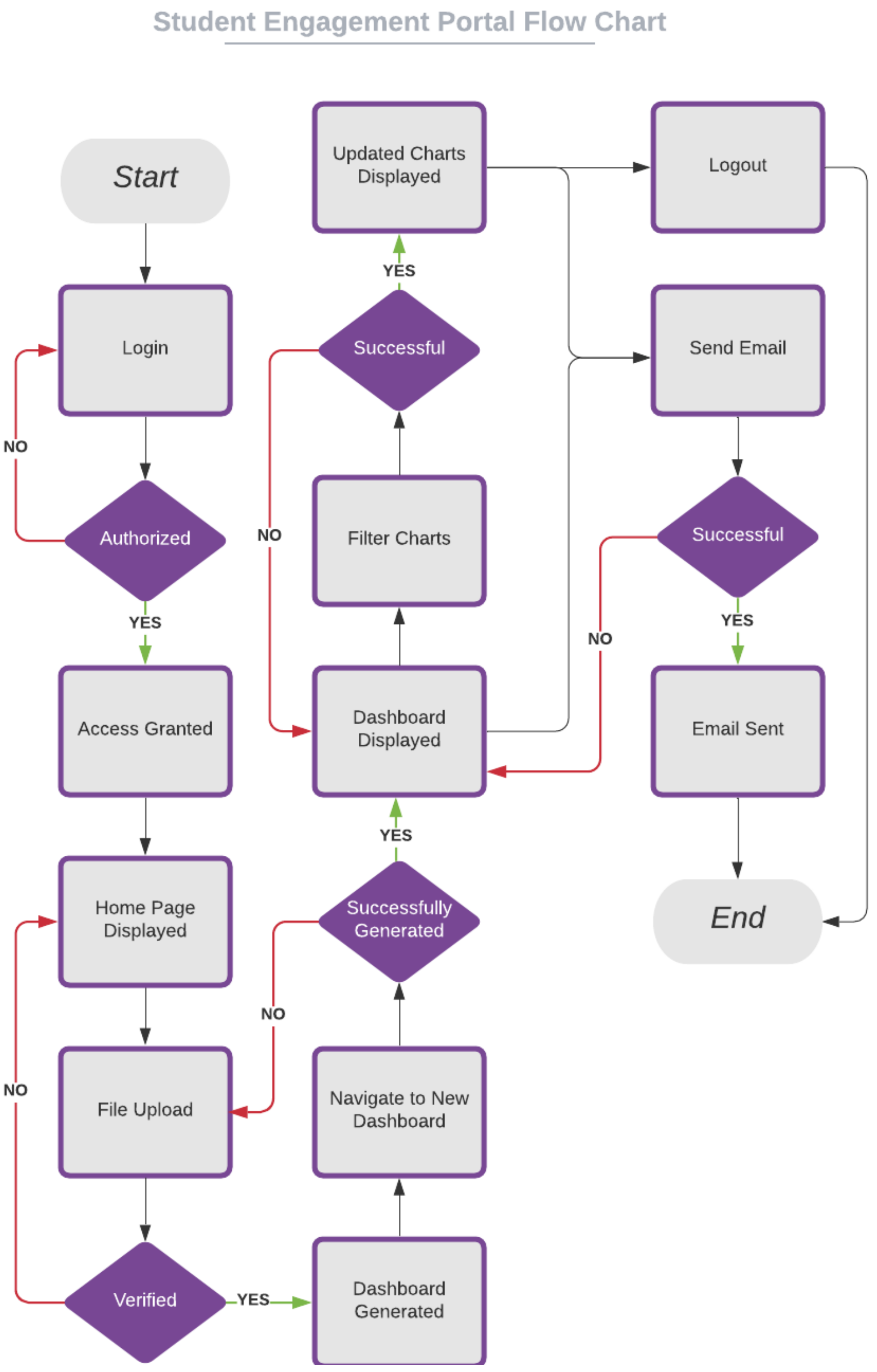
In order to keep the application interface intuitive and easy to understand, I focused on simplicity when creating my UI. Along as I kept my application efficient and I can complete my requirements, then my UI doesn't need to be complex. Considering the target audience for my application will be a mix of people with all different levels of technology experience, I really kept my UI simple but made sure the usability was still there.

Usability testing was performed on the Student Engagement Portal and changed occurred based on the findings after testing. Testing information and results can be found in 2.5.2

2.2. Design & Architecture

For the design and architecture of my project, I created a flow chart. I decided to create a flow chart as Flow Charts would allow me to create the best visual depiction of my application. My project is split up into two main components, uploading files to the MySQL database and generating dashboards with charts on them. All components of my application are viable in order to perform any of the functionality.

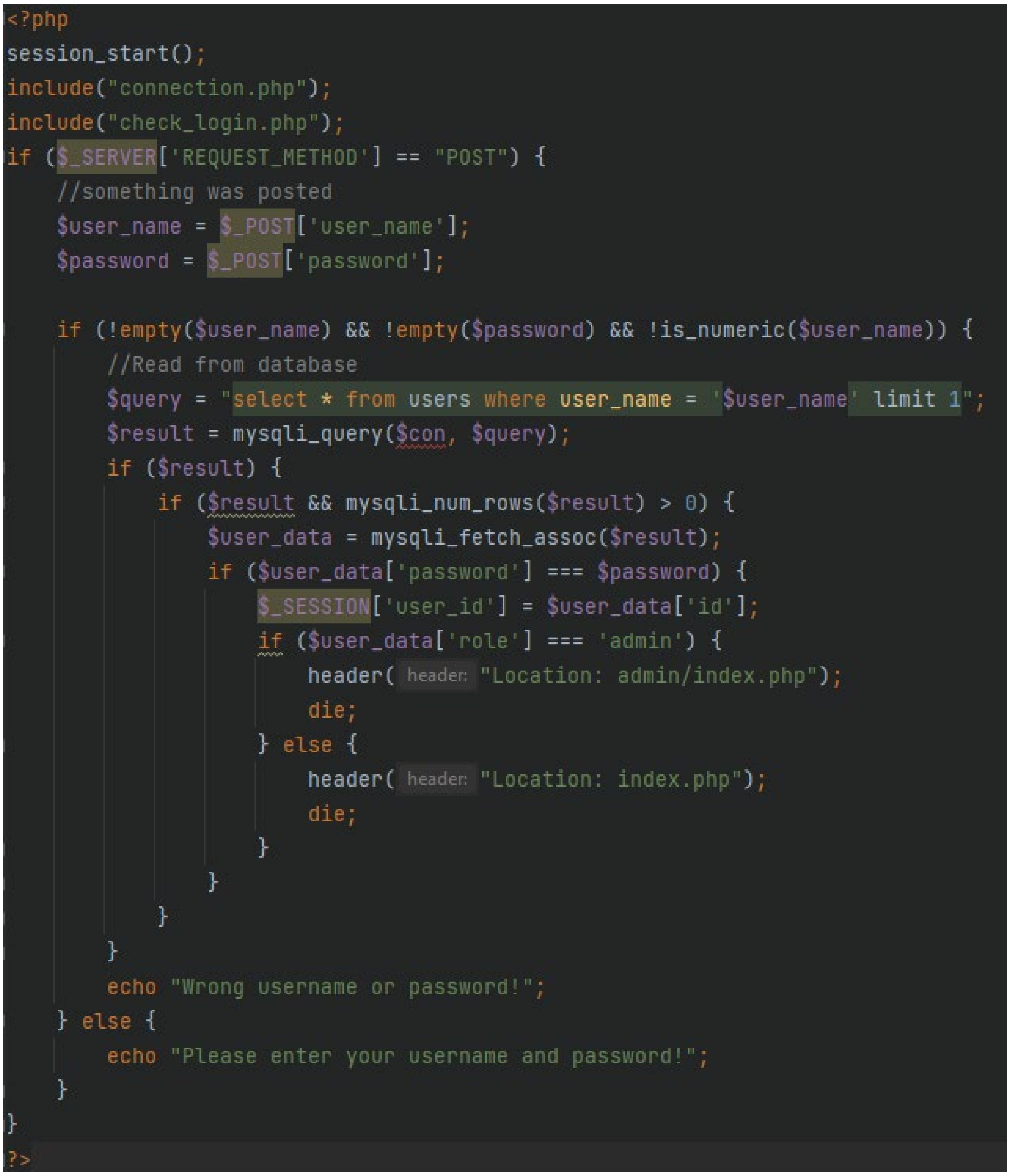
2.2.1 Flow Chart



2.3. Implementation

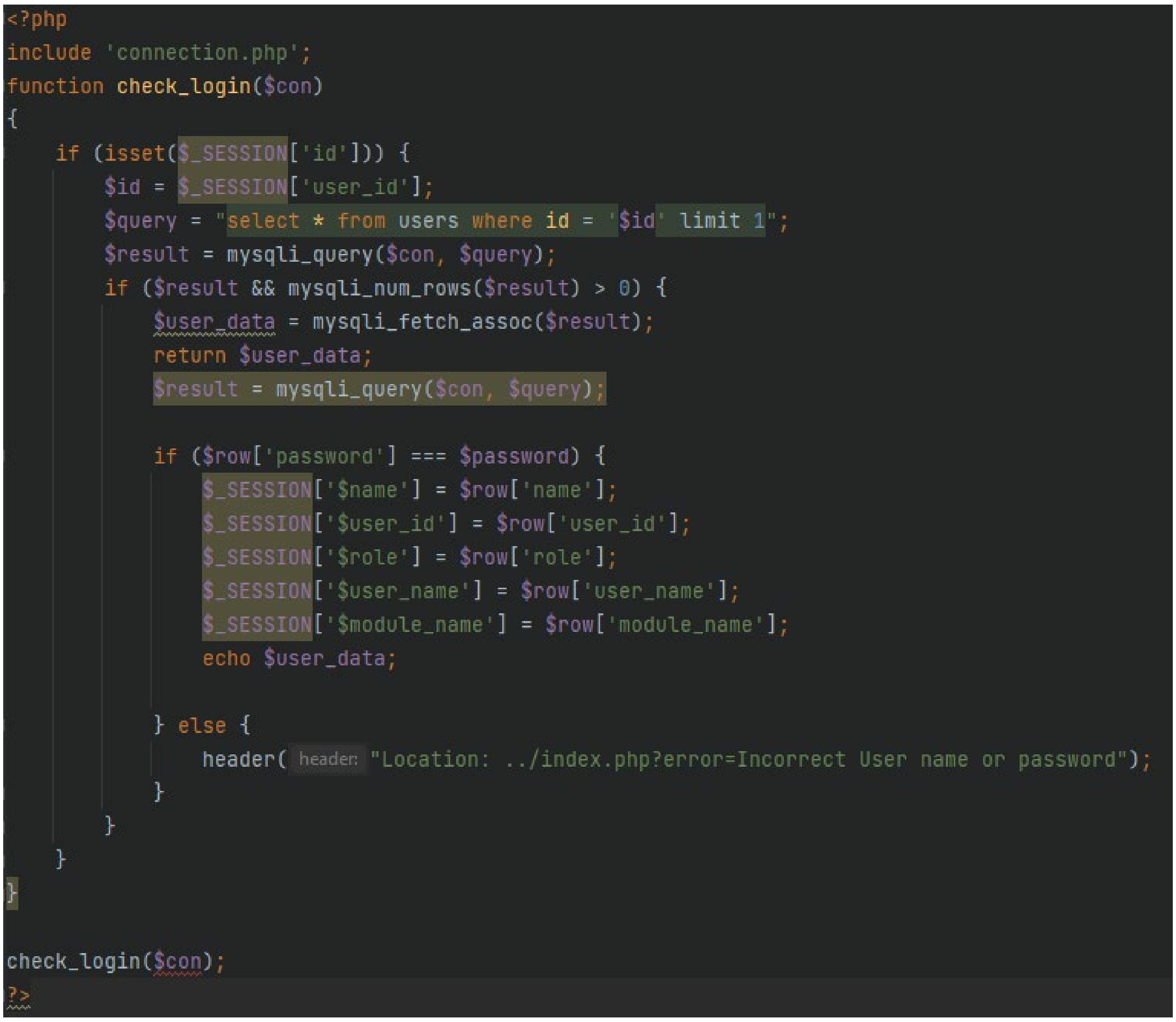
2.3.1 Login

The login activity within the Student Engagement Portal is used to authenticate users and grant them access to the application. Based on what role the user is assigned to, they will be brought to either an admin or user home page. The login activity contains validation to let users know if they have entered in their username or password incorrectly. When users are logging in a function called check_login is called, this function is called to store information in the session which can be used throughout the application, whilst the session is still running.



```php
<?php
session_start();
include("connection.php");
include("check_login.php");
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    //something was posted
    $user_name = $_POST['user_name'];
    $password = $_POST['password'];

    if (!empty($user_name) && !empty($password) && !is_numeric($user_name)) {
        //Read from database
        $query = "select * from users where user_name = '$user_name' limit 1";
        $result = mysqli_query($con, $query);
        if ($result) {
            if ($result && mysqli_num_rows($result) > 0) {
                $user_data = mysqli_fetch_assoc($result);
                if ($user_data['password'] === $password) {
                    $_SESSION['user_id'] = $user_data['id'];
                    if ($user_data['role'] === 'admin') {
                        header( header: "Location: admin/index.php");
                        die;
                    } else {
                        header( header: "Location: index.php");
                        die;
                    }
                }
            }
        }
        echo "Wrong username or password!";
    } else {
        echo "Please enter your username and password!";
    }
}
?>
```

In the image above, the login validation is shown. There are different error messages being echoed, depending on where the error is occurring, which leads to a more informative error message for the user logging in.



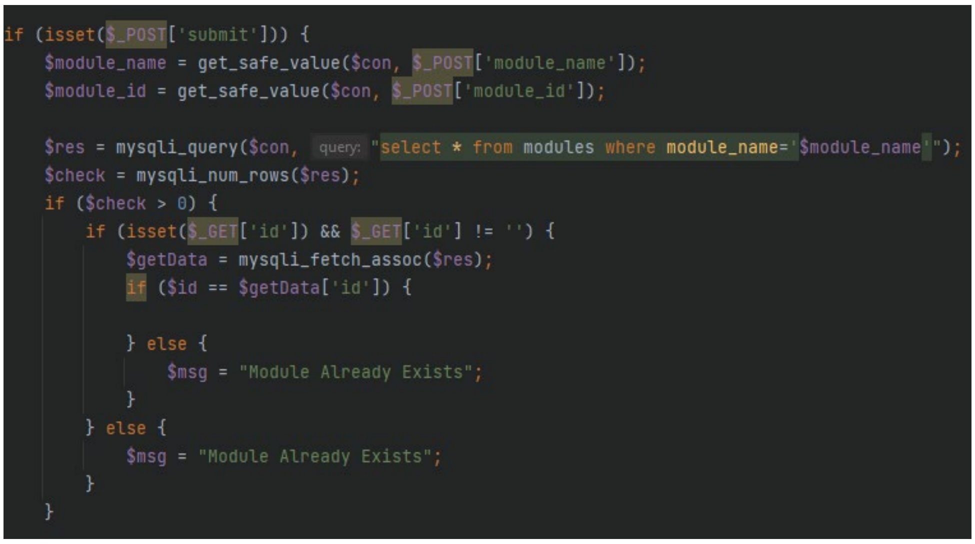
```
<?php
include 'connection.php';
function check_login($con)
{
    if (isset($_SESSION['id'])) {
        $id = $_SESSION['user_id'];
        $query = "select * from users where id = '$id' limit 1";
        $result = mysqli_query($con, $query);
        if ($result && mysqli_num_rows($result) > 0) {
            $user_data = mysqli_fetch_assoc($result);
            return $user_data;
            $result = mysqli_query($con, $query);

            if ($row['password'] === $password) {
                $_SESSION['$name'] = $row['name'];
                $_SESSION['$user_id'] = $row['user_id'];
                $_SESSION['$role'] = $row['role'];
                $_SESSION['$user_name'] = $row['user_name'];
                $_SESSION['$module_name'] = $row['module_name'];
                echo $user_data;

            } else {
                header( header: "Location: ../index.php?error=Incorrect User name or password");
            }
        }
    }
}

check_login($con);
?>
```

In the image above, the check_login function is shown. The main purpose of this function is to set some information into a session that is started when the user logs in. Sessions in PHP are used to store information in variables, these variables can then be used across multiple pages. (Anon., n.d.)

2.3.2 Upload File & Create Module

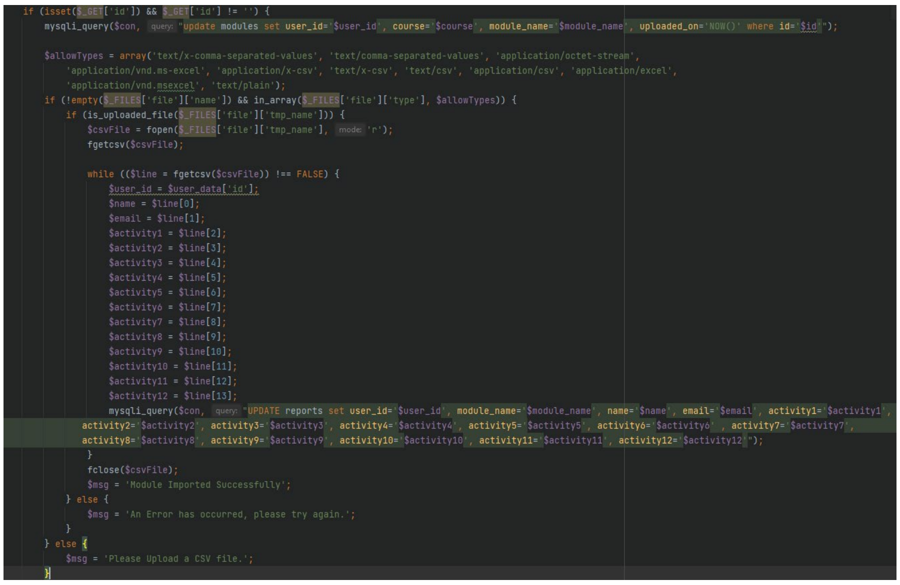
A very important activity for the student engagement portal is the Upload file activity. The purpose of this activity is to check the file type being uploaded whilst getting the module and course name. There is PHP validation implemented into this activity to check what file is being uploaded. In the validation, there is also a check to see if that module already exists, as we do not want to duplicate the module in the database.

This functionality posed a huge challenge for me, as I was needed to preform multiple different MySQL queries to add all the information into different tables in the database, like the reports and modules tables, whilst also adding in foreign keys and performing the validation checks. As I had not used PHP before, performing all these queries and validations were new to me.



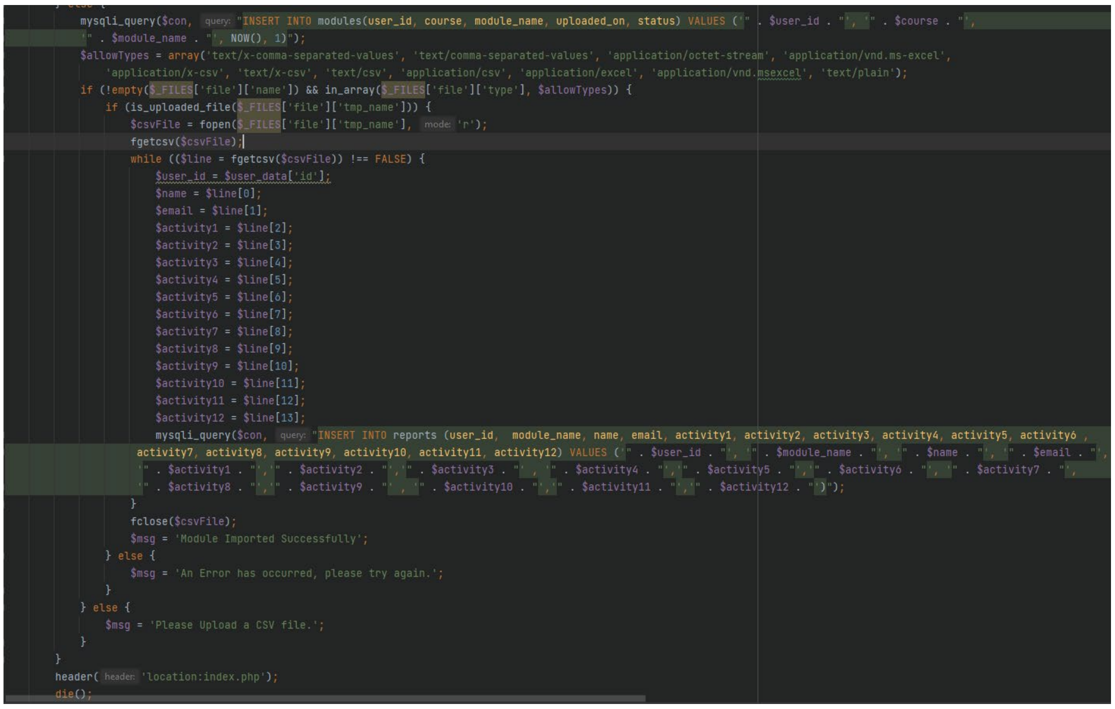
```php
if (isset($_POST['submit'])) {
    $module_name = get_safe_value($con, $_POST['module_name']);
    $module_id = get_safe_value($con, $_POST['module_id']);

    $res = mysqli_query($con, "select * from modules where module_name='$module_name'");
    $check = mysqli_num_rows($res);
    if ($check > 0) {
        if (isset($_GET['id']) && $_GET['id'] != '') {
            $getData = mysqli_fetch_assoc($res);
            if ($id == $getData['id']) {

            } else {
                $msg = "Module Already Exists";
            }
        } else {
            $msg = "Module Already Exists";
        }
    }
```

In the code snippet above, the validation is being shown. This code states that if the module name inputted is a module name already in the database, then an error will occur. If the module already exists in the database, then nothing is added to the tables in the dashboard and the user is shown an error.



```php
    if (isset($_GET['id']) && $_GET['id'] != '') {
        mysqli_query($con, "update modules set user_id='$user_id', course='$course', module_name='$module_name', uploaded_on='NOW()' where id='$id'");

        $allowTypes = array('text/x-comma-separated-values', 'text/comma-separated-values', 'application/octet-stream',
            'application/vnd.ms-excel', 'application/x-csv', 'text/x-csv', 'text/csv', 'application/csv', 'application/excel',
            'application/vnd.msexcel', 'text/plain');
        if (!empty($_FILES['file']['name']) && in_array($_FILES['file']['type'], $allowTypes)) {
            if (is_uploaded_file($_FILES['file']['tmp_name'])) {
                $csvFile = fopen($_FILES['file']['tmp_name'], 'r');
                fgetcsv($csvFile);

                while (($line = fgetcsv($csvFile)) !== FALSE) {
                    $user_id = $user_data['id'];
                    $name = $line[0];
                    $email = $line[1];
                    $activity1 = $line[2];
                    $activity2 = $line[3];
                    $activity3 = $line[4];
                    $activity4 = $line[5];
                    $activity5 = $line[6];
                    $activity6 = $line[7];
                    $activity7 = $line[8];
                    $activity8 = $line[9];
                    $activity9 = $line[10];
                    $activity10 = $line[11];
                    $activity11 = $line[12];
                    $activity12 = $line[13];
                    mysqli_query($con, "UPDATE reports set user_id='$user_id', module_name='$module_name', name='$name', email='$email', activity1='$activity1',
                activity2='$activity2', activity3='$activity3', activity4='$activity4', activity5='$activity5', activity6='$activity6' , activity7='$activity7',
                activity8='$activity8', activity9='$activity9', activity10='$activity10', activity11='$activity11', activity12='$activity12'");
                }
                fclose($csvFile);
                $msg = 'Module Imported Successfully';
            } else {
                $msg = 'An Error has occurred, please try again.';
            }
        } else {
            $msg = 'Please Upload a CSV file.';
        }
```

The code snippet above shows the code that is called when a user is editing a module. There is validation for the file being uploaded. If the file is not in the allow types arras, then nothing will be added or updated to the reports table in the database. However, in this code snippet, we are updating the modules table in the database. This table is being updated as the information from the csv file isn't going into this database. The validation is checking what file type is being uploaded. If the file type is okay, then the module information gets updated in the reports table.



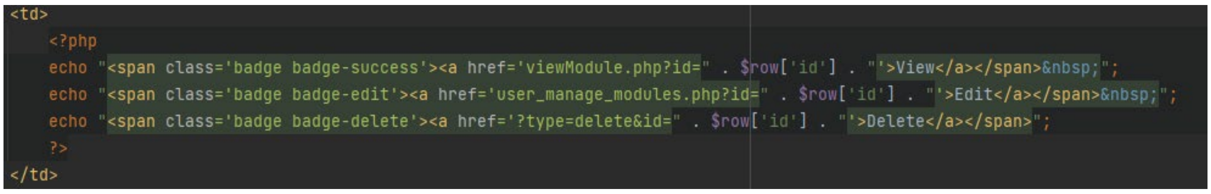
```php
        mysqli_query($con, query: "INSERT INTO modules(user_id, course, module_name, uploaded_on, status) VALUES ('" . $user_id . "', '" . $course . "',
        '" . $module_name . "', NOW(), 1)");
        $allowTypes = array('text/x-comma-separated-values', 'text/comma-separated-values', 'application/octet-stream', 'application/vnd.ms-excel',
            'application/x-csv', 'text/x-csv', 'text/csv', 'application/csv', 'application/excel', 'application/vnd.msexcel', 'text/plain');
        if (!empty($_FILES['file']['name']) && in_array($_FILES['file']['type'], $allowTypes)) {
            if (is_uploaded_file($_FILES['file']['tmp_name'])) {
                $csvFile = fopen($_FILES['file']['tmp_name'], mode: 'r');
                fgetcsv($csvFile);
                while (($line = fgetcsv($csvFile)) !== FALSE) {
                    $user_id = $user_data['id'];
                    $name = $line[0];
                    $email = $line[1];
                    $activity1 = $line[2];
                    $activity2 = $line[3];
                    $activity3 = $line[4];
                    $activity4 = $line[5];
                    $activity5 = $line[6];
                    $activity6 = $line[7];
                    $activity7 = $line[8];
                    $activity8 = $line[9];
                    $activity9 = $line[10];
                    $activity10 = $line[11];
                    $activity11 = $line[12];
                    $activity12 = $line[13];
                    mysqli_query($con, query: "INSERT INTO reports (user_id,  module_name, name, email, activity1, activity2, activity3, activity4, activity5, activity6 ,
                   activity7, activity8, activity9, activity10, activity11, activity12) VALUES ('" . $user_id . "', '" . $module_name . "', '" . $name . "', '" . $email . "',
                   '" . $activity1 . "','" . $activity2 . "','" . $activity3 . "', '" . $activity4 . "','" . $activity5 . "','" . $activity6 . "', '" . $activity7 . "',
                   '" . $activity8 . "','" . $activity9 . "', '" . $activity10 . "','" . $activity11 . "','" . $activity12 . "')");
                }
                fclose($csvFile);
                $msg = 'Module Imported Successfully';
            } else {
                $msg = 'An Error has occurred, please try again.';
            }
        } else {
            $msg = 'Please Upload a CSV file.';
        }
    }
    header( header: 'location:index.php');
    die();
```

The code snippet above shows the code that is called when a user is creating a module. There is validation for the file being uploaded like when the module is being edited. In this code snippet, we are adding the modules and reports table in the database.

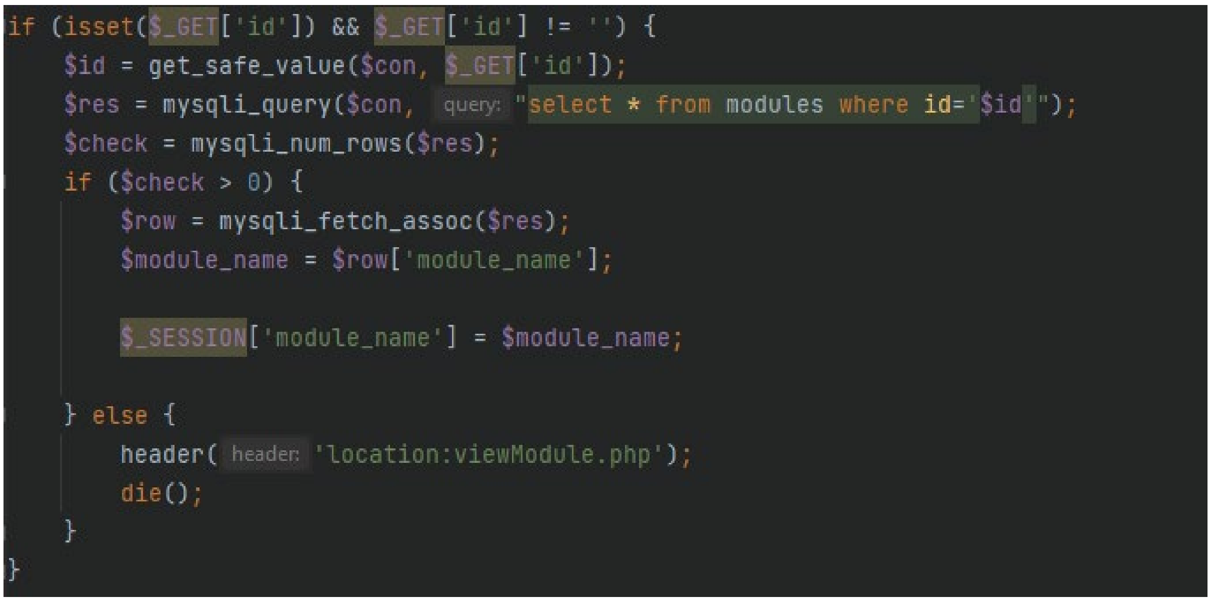
2.3.3 Dashboards and Charts

The dashboard activity is used to view module dashboard. This is the most important piece of functionality for the student engagement dashboard as this activity displays a module dashboard with charts and an email functionality, whilst being on a unique page URL.

HighCharts is a software library that is used for creating charts. HighCharts is written in JavaScript however has wrappers available to use with PHP. I chose HighCharts as it has the most documentation and allowed for the creation of responsive charts. It was a challenge to get the information from the database tables and echo them into a HighCharts chart as I was unfamiliar with library.

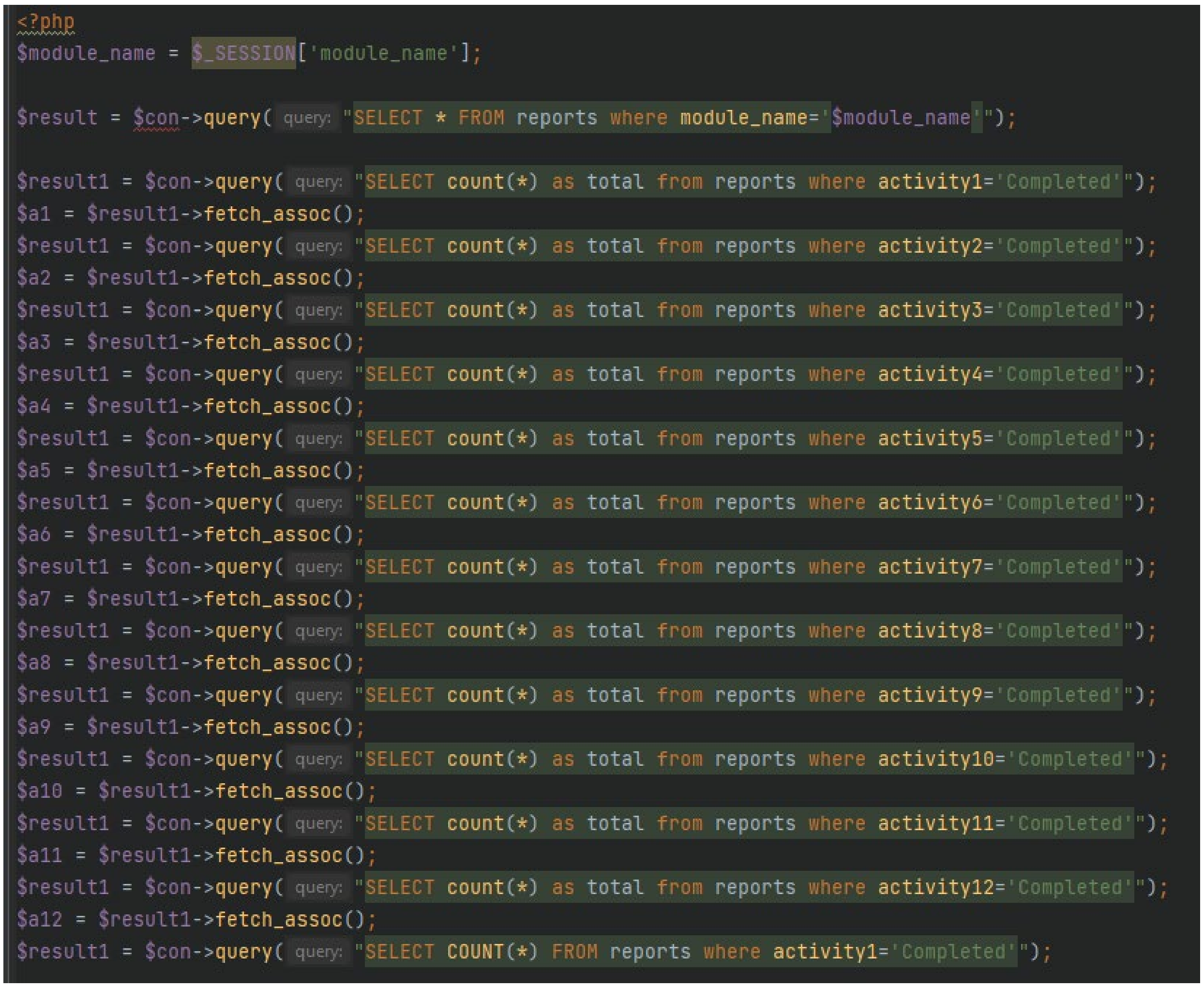


```
<td>
    <?php
    echo "<span class='badge badge-success'><a href='viewModule.php?id=" . $row['id'] . "'>View</a></span>&nbsp;";
    echo "<span class='badge badge-edit'><a href='user_manage_modules.php?id=" . $row['id'] . "'>Edit</a></span>&nbsp;";
    echo "<span class='badge badge-delete'><a href='?type=delete&id=" . $row['id'] . "'>Delete</a></span>";
    ?>
</td>
```



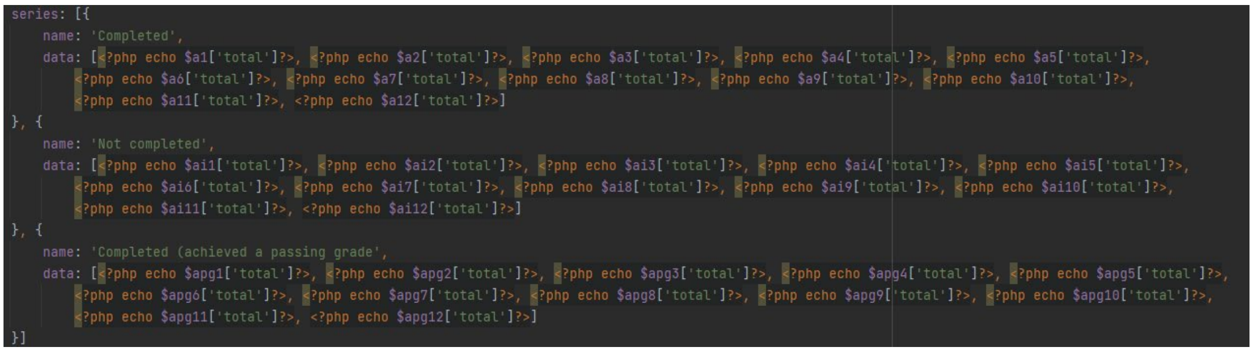
```
if (isset($_GET['id']) && $_GET['id'] != '') {
    $id = get_safe_value($con, $_GET['id']);
    $res = mysqli_query($con,  query: "select * from modules where id='$id'");
    $check = mysqli_num_rows($res);
    if ($check > 0) {
        $row = mysqli_fetch_assoc($res);
        $module_name = $row['module_name'];

        $_SESSION['module_name'] = $module_name;

    } else {
        header( header: 'location:viewModule.php');
        die();
    }
}
```

The above code depicts how the dashboards are displayed on different pages. Using the same page layout but adding in a query to the URL with the module ID, the application is able to display different module dashboards on different pages with the different information in them.



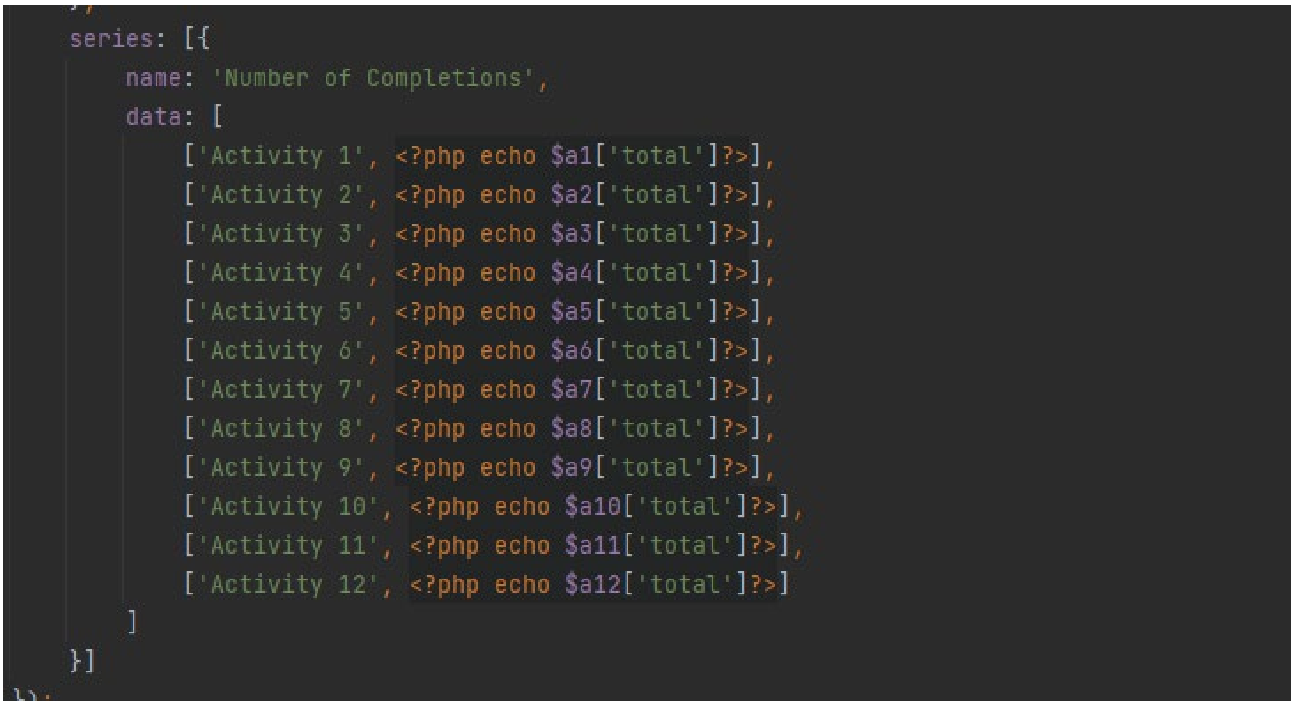
```php
<?php
$module_name = $_SESSION['module_name'];

$result = $con->query( query: "SELECT * FROM reports where module_name='$module_name'");

$result1 = $con->query( query: "SELECT count(*) as total from reports where activity1='Completed'");
$a1 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT count(*) as total from reports where activity2='Completed'");
$a2 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT count(*) as total from reports where activity3='Completed'");
$a3 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT count(*) as total from reports where activity4='Completed'");
$a4 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT count(*) as total from reports where activity5='Completed'");
$a5 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT count(*) as total from reports where activity6='Completed'");
$a6 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT count(*) as total from reports where activity7='Completed'");
$a7 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT count(*) as total from reports where activity8='Completed'");
$a8 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT count(*) as total from reports where activity9='Completed'");
$a9 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT count(*) as total from reports where activity10='Completed'");
$a10 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT count(*) as total from reports where activity11='Completed'");
$a11 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT count(*) as total from reports where activity12='Completed'");
$a12 = $result1->fetch_assoc();
$result1 = $con->query( query: "SELECT COUNT(*) FROM reports where activity1='Completed'");
```

The above code snippet, there is an example on how the application is getting information from the database tables to display in the HighCharts charts. When developing this code, loops were tried and tested however it was not as reliable as doing it the way in the code.



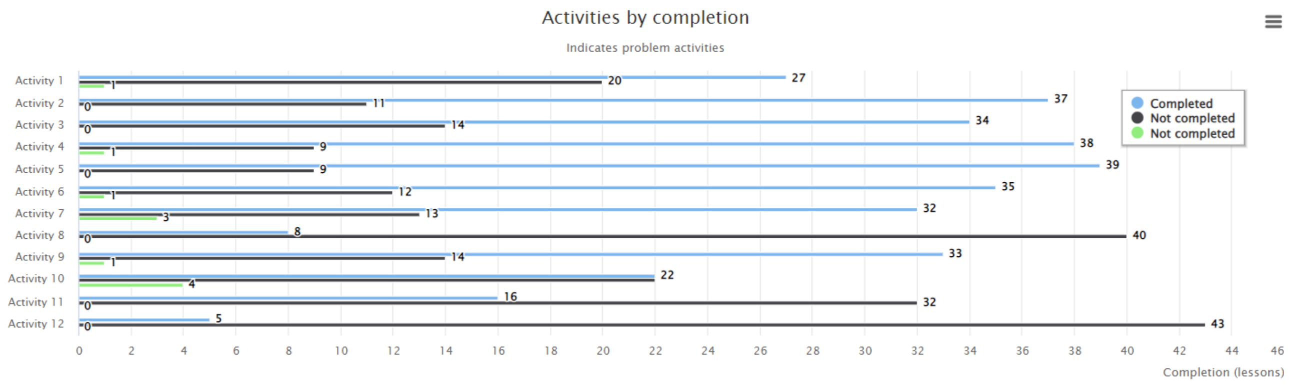
```javascript
series: [{
    name: 'Completed',
    data: [<?php echo $a1['total']?>, <?php echo $a2['total']?>, <?php echo $a3['total']?>, <?php echo $a4['total']?>, <?php echo $a5['total']?>,
        <?php echo $a6['total']?>, <?php echo $a7['total']?>, <?php echo $a8['total']?>, <?php echo $a9['total']?>, <?php echo $a10['total']?>,
        <?php echo $a11['total']?>, <?php echo $a12['total']?>]
}, {
    name: 'Not completed',
    data: [<?php echo $ai1['total']?>, <?php echo $ai2['total']?>, <?php echo $ai3['total']?>, <?php echo $ai4['total']?>, <?php echo $ai5['total']?>,
        <?php echo $ai6['total']?>, <?php echo $ai7['total']?>, <?php echo $ai8['total']?>, <?php echo $ai9['total']?>, <?php echo $ai10['total']?>,
        <?php echo $ai11['total']?>, <?php echo $ai12['total']?>]
}, {
    name: 'Completed (achieved a passing grade',
    data: [<?php echo $apg1['total']?>, <?php echo $apg2['total']?>, <?php echo $apg3['total']?>, <?php echo $apg4['total']?>, <?php echo $apg5['total']?>,
        <?php echo $apg6['total']?>, <?php echo $apg7['total']?>, <?php echo $apg8['total']?>, <?php echo $apg9['total']?>, <?php echo $apg10['total']?>,
        <?php echo $apg11['total']?>, <?php echo $apg12['total']?>]
}]
```

In the code snippet above, the data for the bar chart is being echoed into the array. Using the queries to get the information from the database, echoing in the results displays the information in the bar chart perfectly. The bar chart displays the total completed and not completed activities.

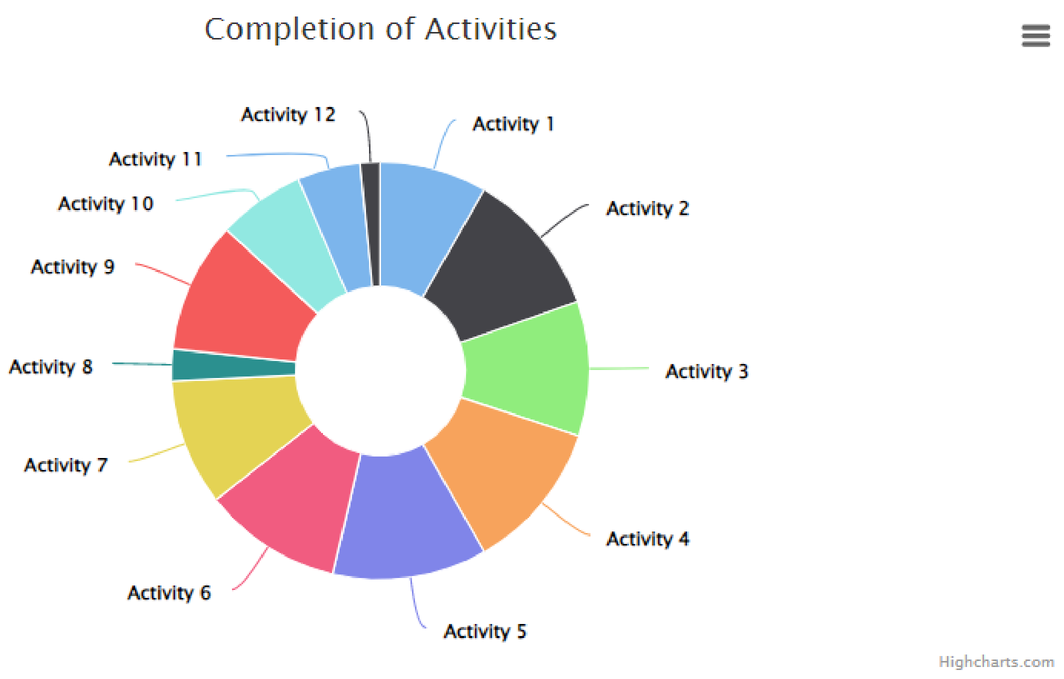


```
    series: [{
        name: 'Number of Completions',
        data: [
            ['Activity 1', <?php echo $a1['total']?>],
            ['Activity 2', <?php echo $a2['total']?>],
            ['Activity 3', <?php echo $a3['total']?>],
            ['Activity 4', <?php echo $a4['total']?>],
            ['Activity 5', <?php echo $a5['total']?>],
            ['Activity 6', <?php echo $a6['total']?>],
            ['Activity 7', <?php echo $a7['total']?>],
            ['Activity 8', <?php echo $a8['total']?>],
            ['Activity 9', <?php echo $a9['total']?>],
            ['Activity 10', <?php echo $a10['total']?>],
            ['Activity 11', <?php echo $a11['total']?>],
            ['Activity 12', <?php echo $a12['total']?>]
        ]
    }]
```

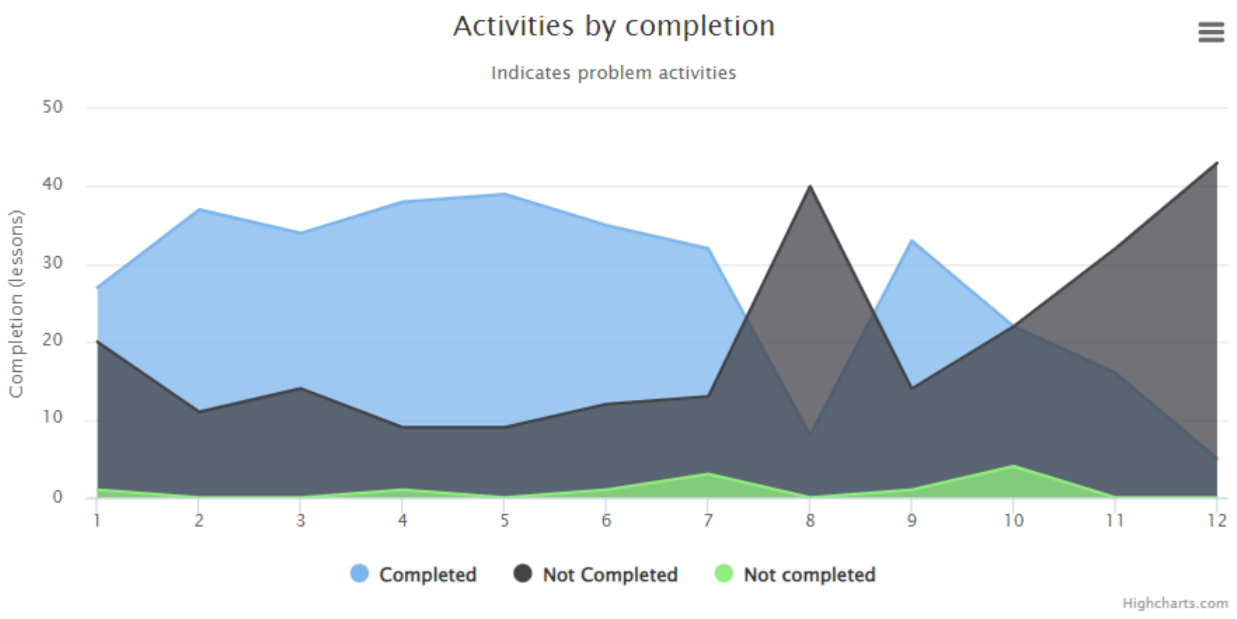
In the code snippet above, the data going into the pie chart is being shown. The pie chart displays the total completed for each activity within the database.



The image above shows the bar chart that was generated with the information within the database. This chart can be downloaded as a PNG or Jpeg.

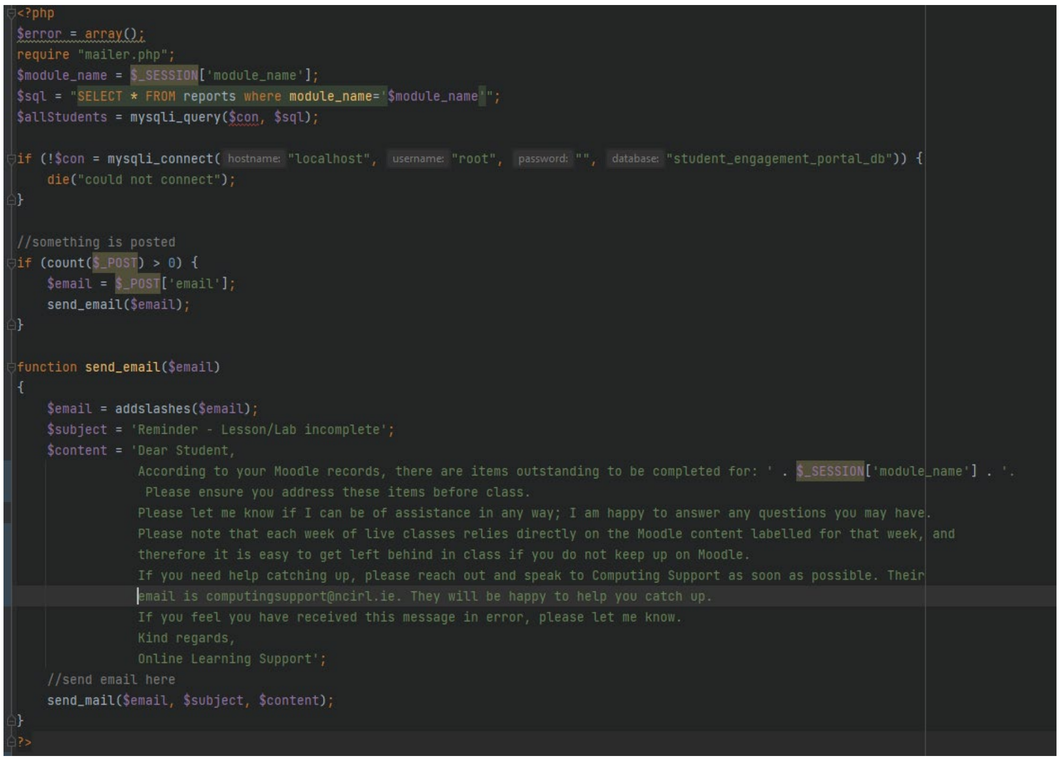


The image above shows the pie chart that was generated with the information within the database. This chart can be downloaded as a PNG or Jpeg.



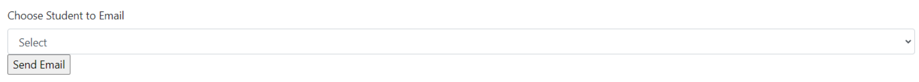
The image above shows the area chart that was generated with the information within the database. This chart can be downloaded as a PNG or Jpeg.

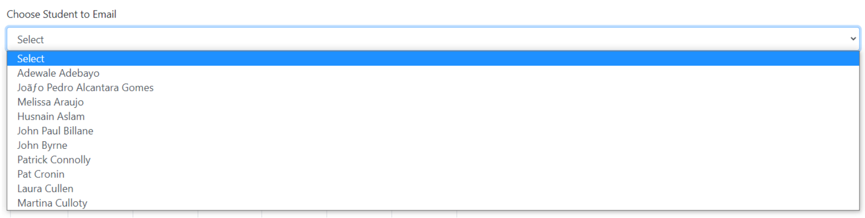
2.3.4 Email Students



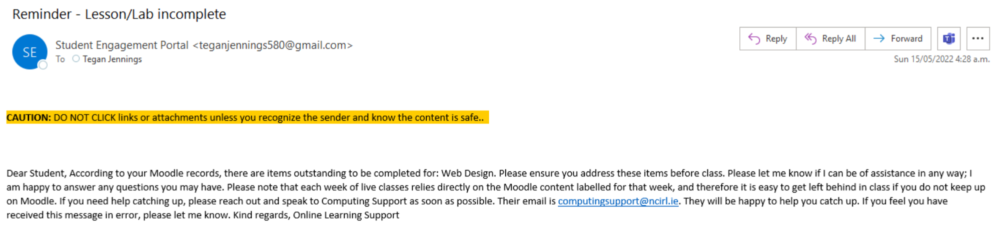
```php
<?php
$error = array();
require "mailer.php";
$module_name = $_SESSION['module_name'];
$sql = "SELECT * FROM reports where module_name='$module_name'";
$allStudents = mysqli_query($con, $sql);

if (!$con = mysqli_connect( hostname: "localhost",  username: "root",  password: "",  database: "student_engagement_portal_db")) {
    die("could not connect");
}

//something is posted
if (count($_POST) > 0) {
    $email = $_POST['email'];
    send_email($email);
}

function send_email($email)
{
    $email = addslashes($email);
    $subject = 'Reminder - Lesson/Lab incomplete';
    $content = 'Dear Student,
                According to your Moodle records, there are items outstanding to be completed for: ' . $_SESSION['module_name'] . '.
                 Please ensure you address these items before class.
                Please let me know if I can be of assistance in any way; I am happy to answer any questions you may have.
                Please note that each week of live classes relies directly on the Moodle content labelled for that week, and
                therefore it is easy to get left behind in class if you do not keep up on Moodle.
                If you need help catching up, please reach out and speak to Computing Support as soon as possible. Their
                email is computingsupport@ncirl.ie. They will be happy to help you catch up.
                If you feel you have received this message in error, please let me know.
                Kind regards,
                Online Learning Support';
    //send email here
    send_mail($email, $subject, $content);
}
?>
```

The above code shows the send email activity. This activity runs when the email student button is clicked, and a student has been selected from the dropdown. In this code, there is a set subject line and email body content.





The above screenshots are screenshots of the dropdown and button that are associated with the email student's activity. Based on the module, students' names will be displayed in the dropdown. Users can click on a student name, and then click send to send them the generic support email.



Reminder - Lesson/Lab incomplete

Student Engagement Portal <teganjennings580@gmail.com>
To Tegan Jennings

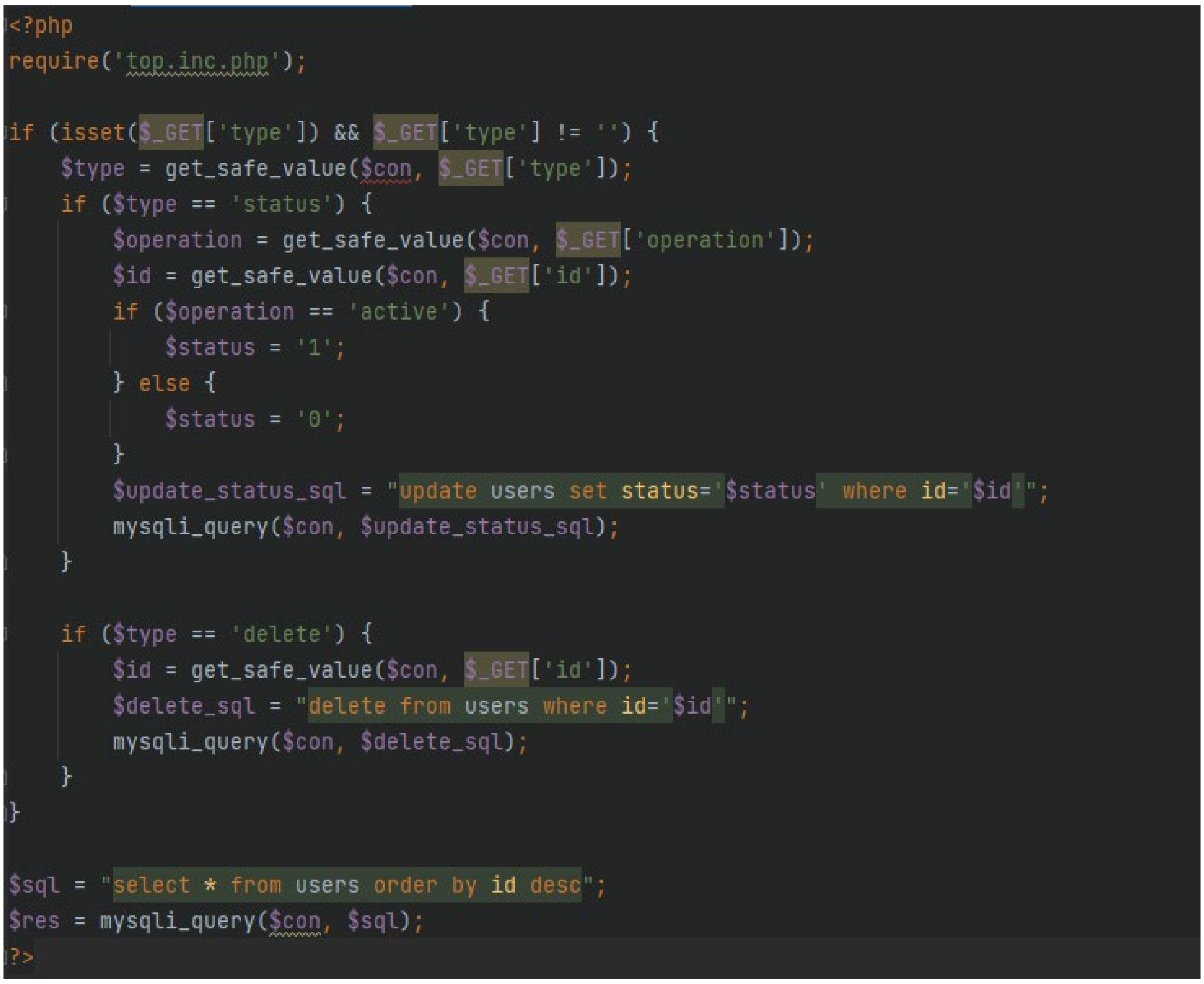
Sun 15/05/2022 4:28 a.m.

CAUTION: DO NOT CLICK links or attachments unless you recognize the sender and know the content is safe..

Dear Student, According to your Moodle records, there are items outstanding to be completed for: Web Design. Please ensure you address these items before class. Please let me know if I can be of assistance in any way; I am happy to answer any questions you may have. Please note that each week of live classes relies directly on the Moodle content labelled for that week, and therefore it is easy to get left behind in class if you do not keep up on Moodle. If you need help catching up, please reach out and speak to Computing Support as soon as possible. Their email is computingsupport@ncirl.ie. They will be happy to help you catch up. If you feel you have received this message in error, please let me know. Kind regards, Online Learning Support

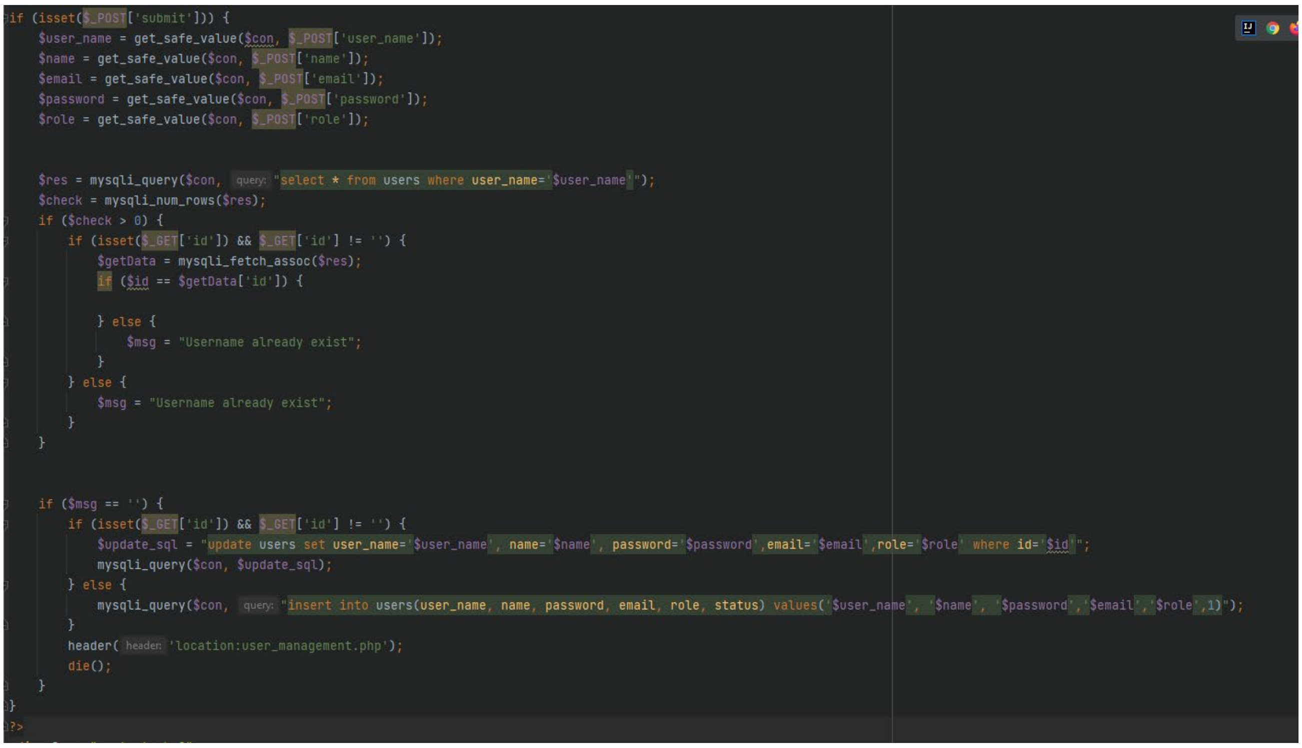
The above screenshot is a screenshot of a support email sent from the Student Engagement portal successfully.

2.3.5 User Management



```php
<?php
require('top.inc.php');

if (isset($_GET['type']) && $_GET['type'] != '') {
    $type = get_safe_value($con, $_GET['type']);
    if ($type == 'status') {
        $operation = get_safe_value($con, $_GET['operation']);
        $id = get_safe_value($con, $_GET['id']);
        if ($operation == 'active') {
            $status = '1';
        } else {
            $status = '0';
        }
        $update_status_sql = "update users set status='$status' where id='$id'";
        mysqli_query($con, $update_status_sql);
    }

    if ($type == 'delete') {
        $id = get_safe_value($con, $_GET['id']);
        $delete_sql = "delete from users where id='$id'";
        mysqli_query($con, $delete_sql);
    }
}

$sql = "select * from users order by id desc";
$res = mysqli_query($con, $sql);
?>
```

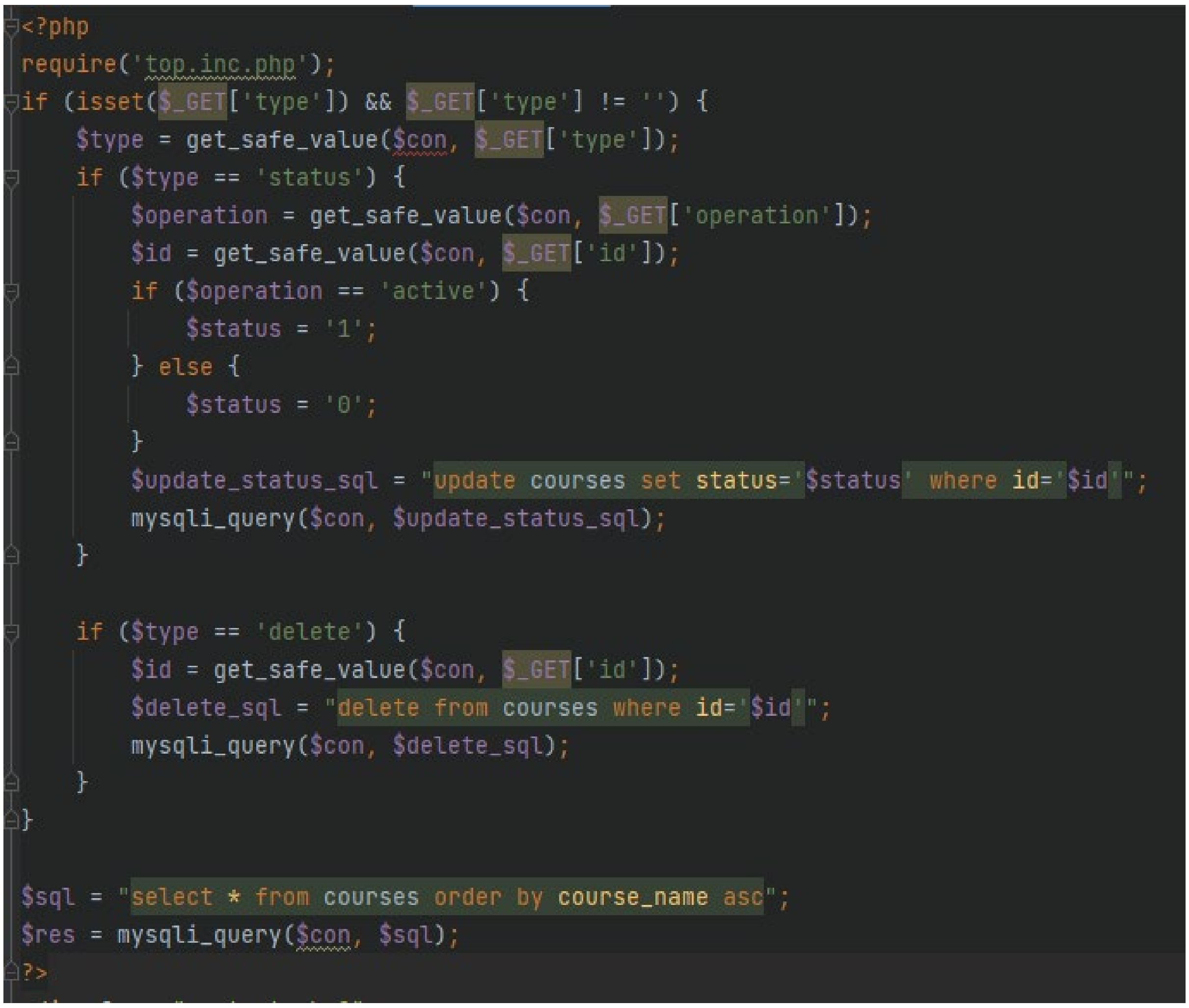
In the code snippet above, there are multiple different actions taking place. In this snippet, there is an if statement that will get the status of users and also update users' statuses. The status depicts if a user account is active or not. This code also shows the delete functionality. When the delete button is clicked, an if statement is run to drop that user from the database.



```php
if (isset($_POST['submit'])) {
    $user_name = get_safe_value($con, $_POST['user_name']);
    $name = get_safe_value($con, $_POST['name']);
    $email = get_safe_value($con, $_POST['email']);
    $password = get_safe_value($con, $_POST['password']);
    $role = get_safe_value($con, $_POST['role']);


    $res = mysqli_query($con, query: "select * from users where user_name='$user_name'");
    $check = mysqli_num_rows($res);
    if ($check > 0) {
        if (isset($_GET['id']) && $_GET['id'] != '') {
            $getData = mysqli_fetch_assoc($res);
            if ($id == $getData['id']) {

            } else {
                $msg = "Username already exist";
            }
        } else {
            $msg = "Username already exist";
        }
    }


    if ($msg == '') {
        if (isset($_GET['id']) && $_GET['id'] != '') {
            $update_sql = "update users set user_name='$user_name', name='$name', password='$password',email='$email',role='$role' where id='$id'";
            mysqli_query($con, $update_sql);
        } else {
            mysqli_query($con, query: "insert into users(user_name, name, password, email, role, status) values('$user_name', '$name', '$password','$email','$role',1)");
        }
        header( header: 'location:user_management.php');
        die();
    }
}
?>
```

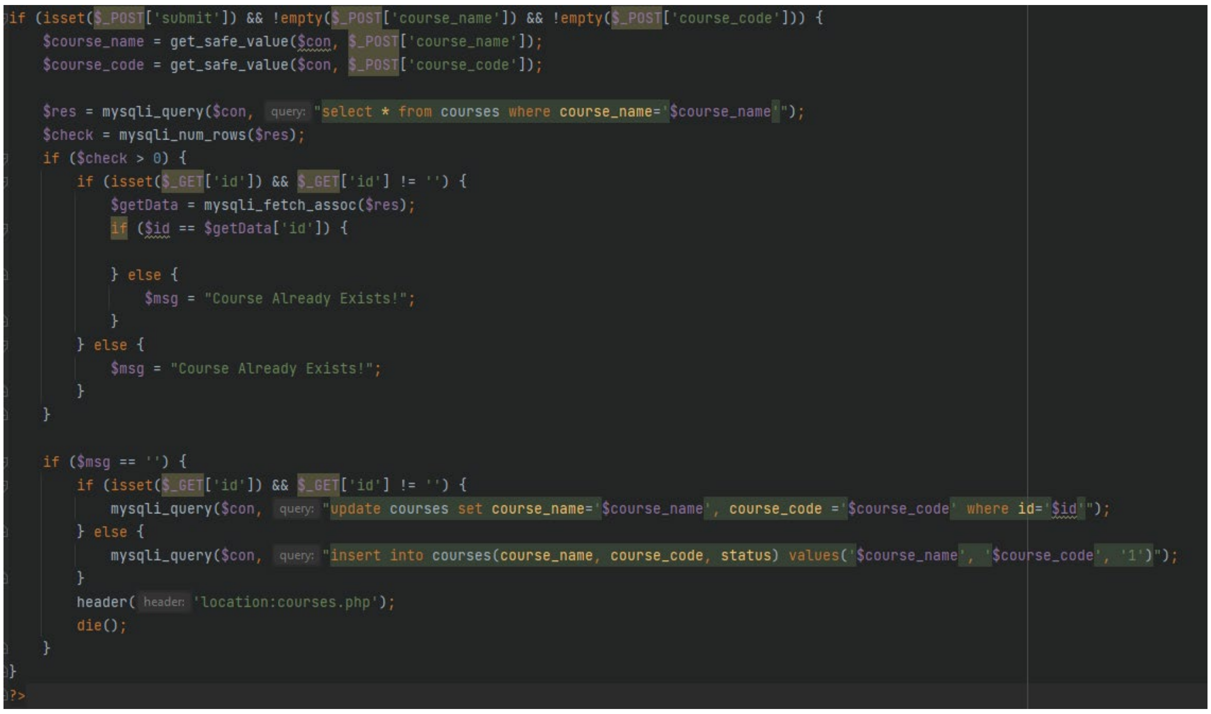
The code snippet above displays the if statement that is triggered when a user is being created or updated. The first if statement is checking if a user is already in the database, if a user does already exist, a message variable is set. If the message variable is empty, then the next if statement is triggered, and this if statement updates the users table in the database or inserts into it, depending on whether a user id already exists or not.

2.3.6 Courses



```
<?php
require('top.inc.php');
if (isset($_GET['type']) && $_GET['type'] != '') {
    $type = get_safe_value($con, $_GET['type']);
    if ($type == 'status') {
        $operation = get_safe_value($con, $_GET['operation']);
        $id = get_safe_value($con, $_GET['id']);
        if ($operation == 'active') {
            $status = '1';
        } else {
            $status = '0';
        }
        $update_status_sql = "update courses set status='$status' where id='$id'";
        mysqli_query($con, $update_status_sql);
    }

    if ($type == 'delete') {
        $id = get_safe_value($con, $_GET['id']);
        $delete_sql = "delete from courses where id='$id'";
        mysqli_query($con, $delete_sql);
    }
}

$sql = "select * from courses order by course_name asc";
$res = mysqli_query($con, $sql);
?>
```

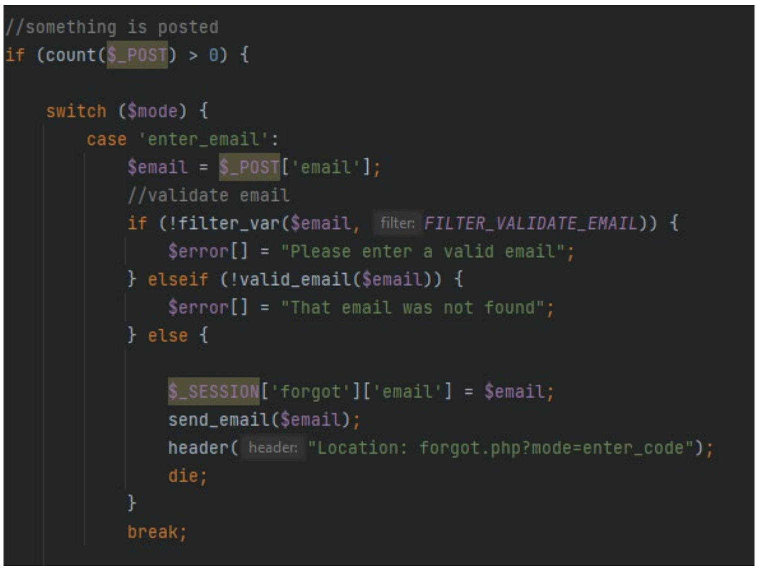
There are multiple different actions taking place in the code snippet above. In this snippet, there is an if statement that will get the status of a course and also update courses statuses. The status shows the user if a course is active or not. This code also shows the delete functionality. When the delete button is clicked, an if statement is run to drop that course from the database.



```
if (isset($_POST['submit']) && !empty($_POST['course_name']) && !empty($_POST['course_code'])) {
    $course_name = get_safe_value($con, $_POST['course_name']);
    $course_code = get_safe_value($con, $_POST['course_code']);

    $res = mysqli_query($con, query: "select * from courses where course_name='$course_name'");
    $check = mysqli_num_rows($res);
    if ($check > 0) {
        if (isset($_GET['id']) && $_GET['id'] != '') {
            $getData = mysqli_fetch_assoc($res);
            if ($id == $getData['id']) {

            } else {
                $msg = "Course Already Exists!";
            }
        } else {
            $msg = "Course Already Exists!";
        }
    }

    if ($msg == '') {
        if (isset($_GET['id']) && $_GET['id'] != '') {
            mysqli_query($con, query: "update courses set course_name='$course_name', course_code ='$course_code' where id='$id'");
        } else {
            mysqli_query($con, query: "insert into courses(course_name, course_code, status) values('$course_name', '$course_code', '1')");
        }
        header( header: 'location:courses.php');
        die();
    }
}
?>
```

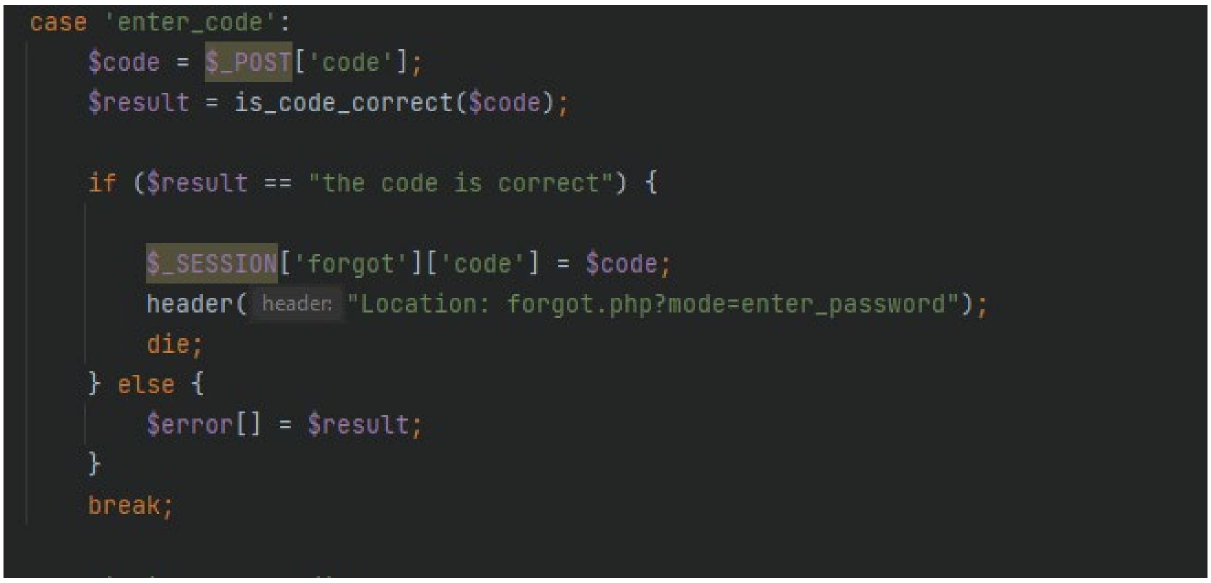
The if statement above is triggered when a course is being created or updated. The first if statement is checking if a course is already in the database, if a course does already exist, a message variable is set. If the message variable is empty, then the next if statement is triggered, and this if statement updates the courses table in the database or inserts into it, depending on whether a course id already exists or not.

2.3.7 Forgot password



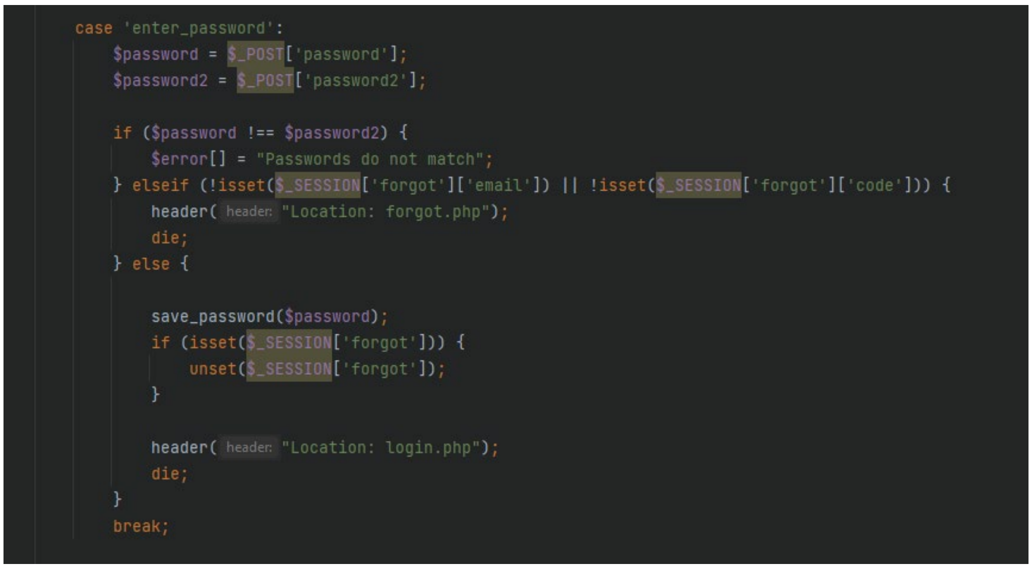
```php
//something is posted
if (count($_POST) > 0) {

    switch ($mode) {
        case 'enter_email':
            $email = $_POST['email'];
            //validate email
            if (!filter_var($email,  filter: FILTER_VALIDATE_EMAIL)) {
                $error[] = "Please enter a valid email";
            } elseif (!valid_email($email)) {
                $error[] = "That email was not found";
            } else {

                $_SESSION['forgot']['email'] = $email;
                send_email($email);
                header( header: "Location: forgot.php?mode=enter_code");
                die;
            }
            break;
```

The above code is used when a user has forgotten their password. This code is started when the user submits an email. The code then grabs the posted email and checks whether it is a valid email address or not. If it is a valid email address, then it's checked to see if it exists in the database already. The email needs to be in the database already, if it isn't then an error is displayed, if it is then an email containing a code is sent to the user. User goes to the next case.



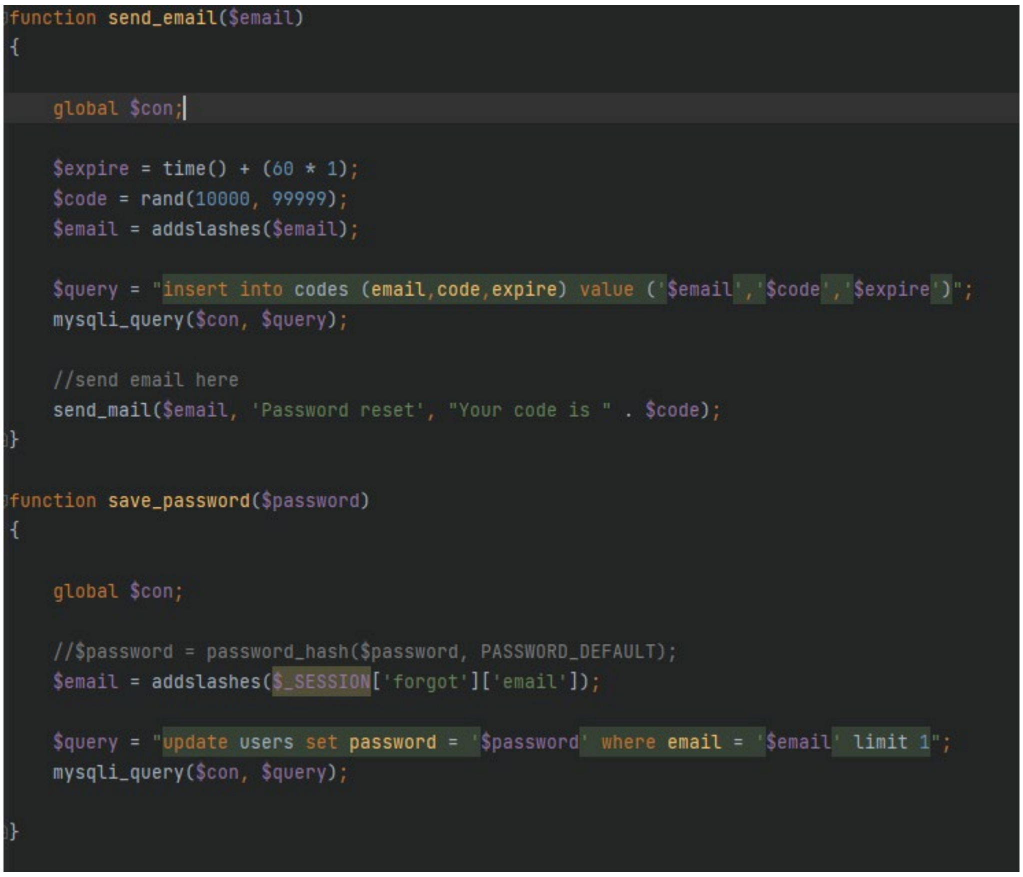
```php
case 'enter_code':
    $code = $_POST['code'];
    $result = is_code_correct($code);

    if ($result == "the code is correct") {

        $_SESSION['forgot']['code'] = $code;
        header( header: "Location: forgot.php?mode=enter_password");
        die;
    } else {
        $error[] = $result;
    }
    break;
```

The next case is the enter code case. In the snippet above, when a user clicks the submit button after entering the code, there is a check to see if the code is the one sent, if it is the user progresses to the next case, if not then an error is shown.



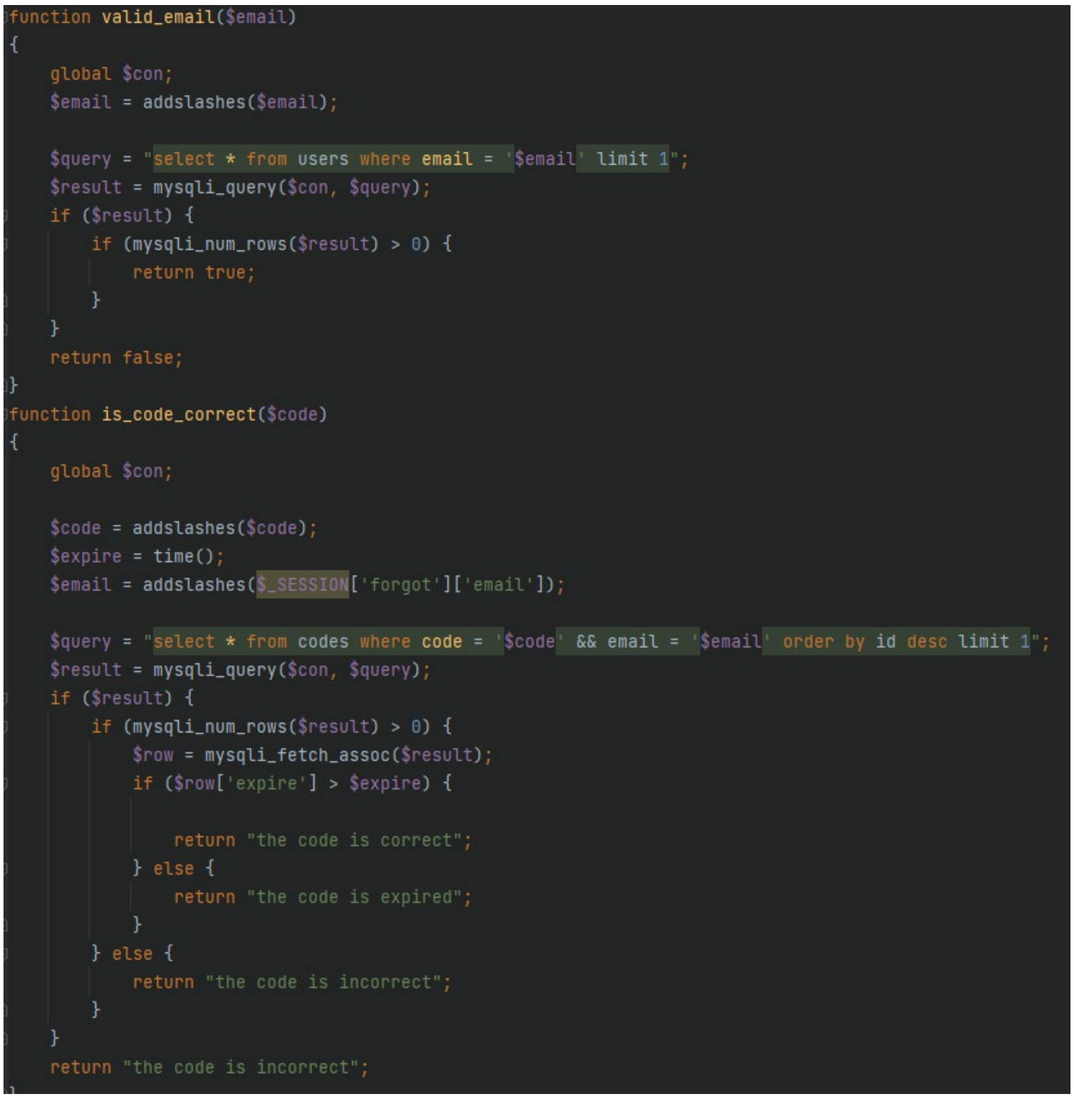
```
case 'enter_password':
    $password = $_POST['password'];
    $password2 = $_POST['password2'];

    if ($password !== $password2) {
        $error[] = "Passwords do not match";
    } elseif (!isset($_SESSION['forgot']['email']) || !isset($_SESSION['forgot']['code'])) {
        header( header: "Location: forgot.php");
        die;
    } else {

        save_password($password);
        if (isset($_SESSION['forgot'])) {
            unset($_SESSION['forgot']);
        }

        header( header: "Location: login.php");
        die;
    }
    break;
```

The next case is the enter password case. In the snippet above, when a user clicks the submit button, the passwords entered are stored in variables and checked to see if they match each other. If they match, then the user is brought back to the login page, if they don’t an error is displayed.



```
function send_email($email)
{

    global $con;

    $expire = time() + (60 * 1);
    $code = rand(10000, 99999);
    $email = addslashes($email);

    $query = "insert into codes (email,code,expire) value ('$email','$code','$expire')";
    mysqli_query($con, $query);

    //send email here
    send_mail($email, 'Password reset', "Your code is " . $code);
}

function save_password($password)
{

    global $con;

    //$password = password_hash($password, PASSWORD_DEFAULT);
    $email = addslashes($_SESSION['forgot']['email']);

    $query = "update users set password = '$password' where email = '$email' limit 1";
    mysqli_query($con, $query);

}
```

The above snippet contains two functions, one for sending emails and one for saving passwords. The send email function is used to send the code to a user when the email is validated and exists in the database. There is a MySQL query to add the code and expiry time into a codes table in the database so that the code can be checked if it was entered correctly or not. The save password function updates the users table in the database with the new password the user has set.



```php
function valid_email($email)
{
    global $con;
    $email = addslashes($email);

    $query = "select * from users where email = '$email' limit 1";
    $result = mysqli_query($con, $query);
    if ($result) {
        if (mysqli_num_rows($result) > 0) {
            return true;
        }
    }
    return false;
}
function is_code_correct($code)
{
    global $con;

    $code = addslashes($code);
    $expire = time();
    $email = addslashes($_SESSION['forgot']['email']);

    $query = "select * from codes where code = '$code' && email = '$email' order by id desc limit 1";
    $result = mysqli_query($con, $query);
    if ($result) {
        if (mysqli_num_rows($result) > 0) {
            $row = mysqli_fetch_assoc($result);
            if ($row['expire'] > $expire) {

                return "the code is correct";
            } else {
                return "the code is expired";
            }
        } else {
            return "the code is incorrect";
        }
    }
    return "the code is incorrect";
```

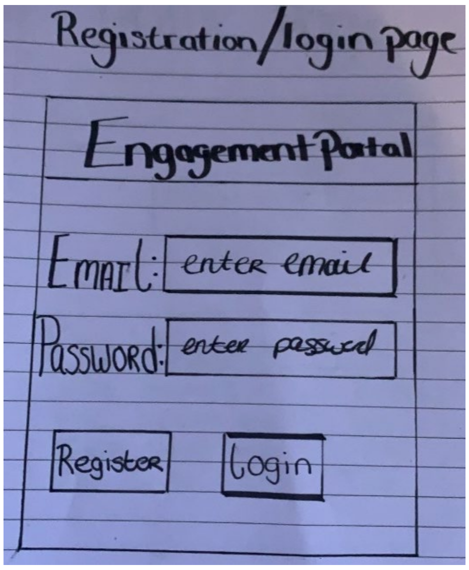
The above snippet displays two different functions, one to valid emails and the other to check if the code entered is correct. In the valid email function, there is a query to check if the inputted email exists in the database. If it does, the function returns true, if not it returns false. The is code correct function checks if the inputted code is the same as the code from the database. There is a query to grab the codes in the database to check the inputted on against. This function also checks to see if the code has expired yet or not. Depending on if it has or hasn’t, a message is returned to users.

2.4. Graphical User Interface (GUI)

2.4.1 Low Fidelity Graphical User Interface Design

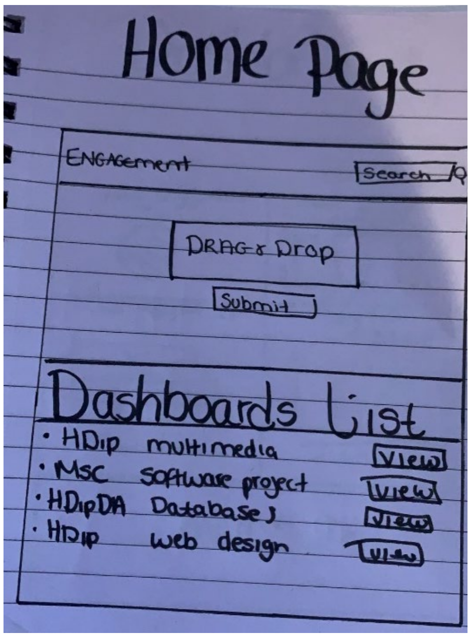
Before developing any of the user interface for the student engagement portal, low fidelity prototypes were drawn on paper.

2.4.1.1 Login page



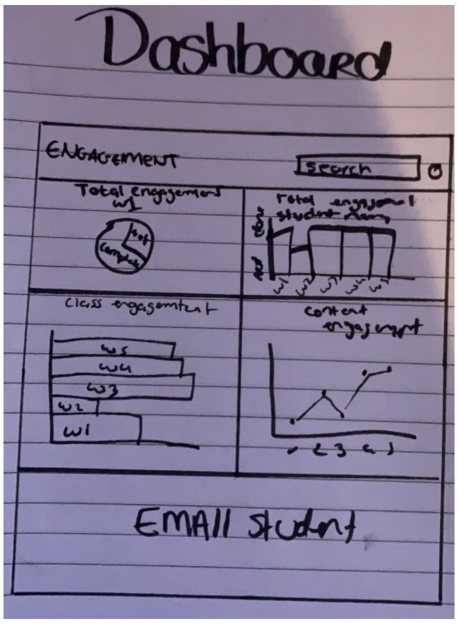
The above image is a low fidelity prototype for the student engagement portal. This prototype was a simple register/login page with a 'Register' and 'Login' button along with two inputs, email and password.

2.4.1.2 Home Page



The above image is an early prototype of the home page for the Student Engagement Portal. This prototype has a large drag and drop at the top for users to input files and then a list of dashboards that has already been created at the bottom.

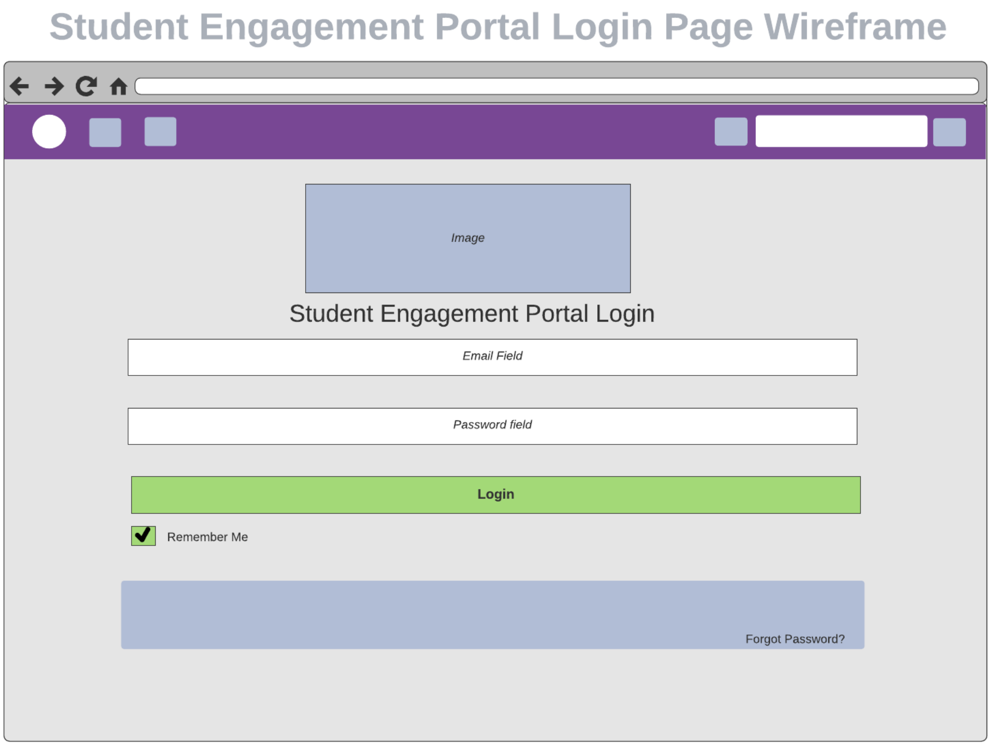
2.4.1.3 Dashboard Page



The above image is an early prototype of the dashboard page for the Student Engagement Portal. This prototype has four different generated charts on it and then a section at the bottom to email students.

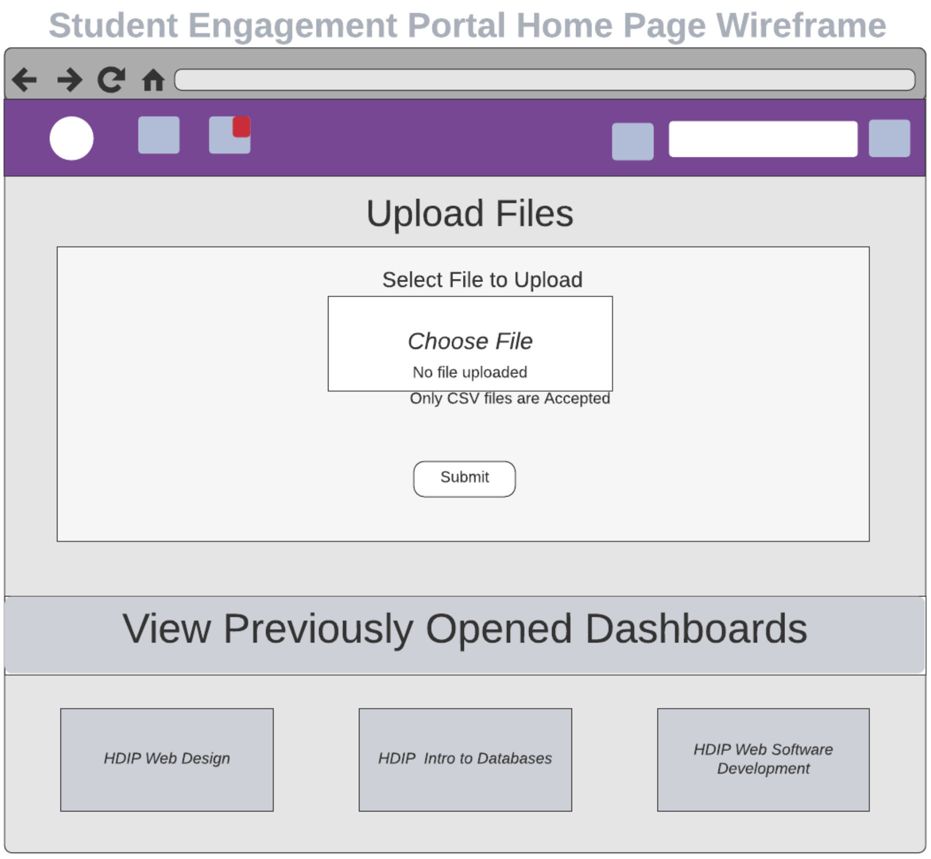
2.4.2 Wireframes

2.4.2.1 Login page


Student Engagement Portal Login Page Wireframe
Image
Student Engagement Portal Login
Email Field
Password field
Login
Remember Me
Forgot Password?

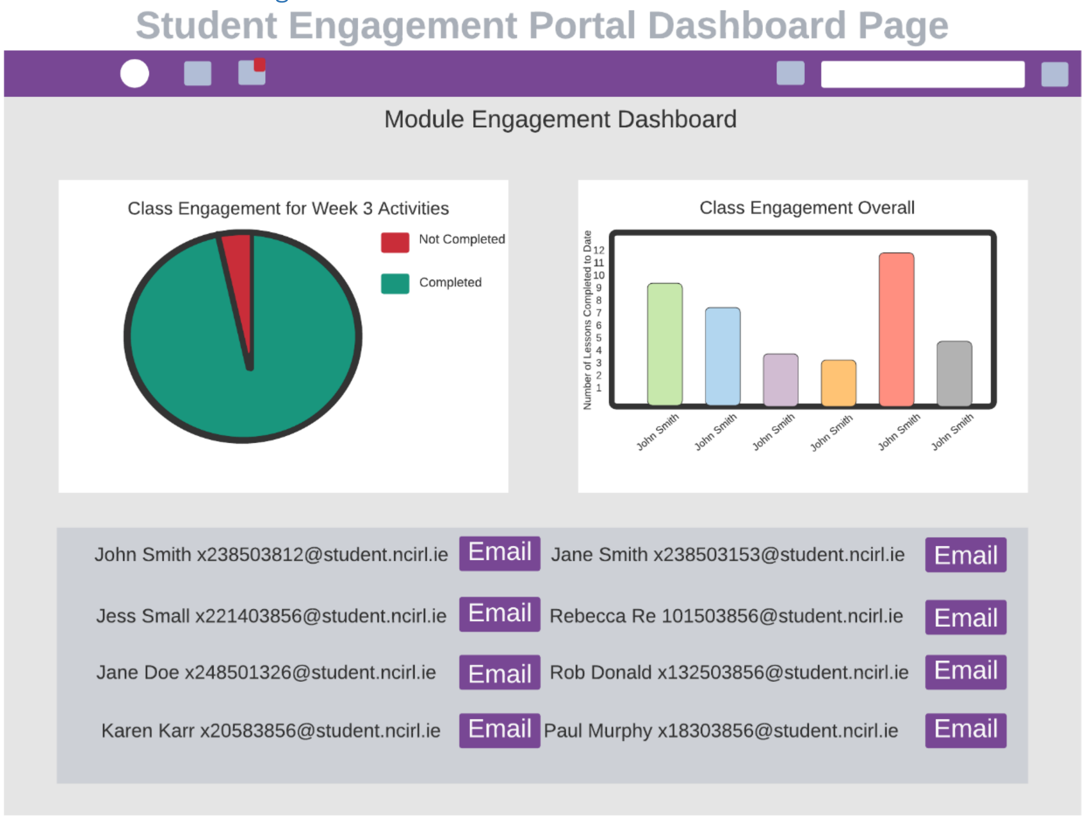
The login page wireframe is displayed above. Something very noticeable from this wireframe in comparison to the low fidelity prototype, is that there is no registration section of this page anymore. This is because for the Student Engagement portal, only authorised users can look into the application, users cannot register for it as it will contain sensitive information. The design has also become more user friendly.

2.4.2.2 Home page


Student Engagement Portal Home Page Wireframe
Upload Files
Select File to Upload
Choose File
No file uploaded
Only CSV files are Accepted
Submit
View Previously Opened Dashboards
HDIP Web Design
HDIP Intro to Databases
HDIP Web Software Development

The homepage wireframe above displays ideally how the home page will look like. The majority of the page is reserved for uploading files to create dashboards and then the rest of the home page is links to previously opened dashboard pages. A noticeable difference compared to the earlier prototype, is that the bottom section is now different. Instead of a dashboards list, it is now a previously opened dashboards section. This section makes the application more user friendly and accessible as most users will only have one or two dashboards associated to their account.

2.4.2.3 Dashboard Page


Student Engagement Portal Dashboard Page

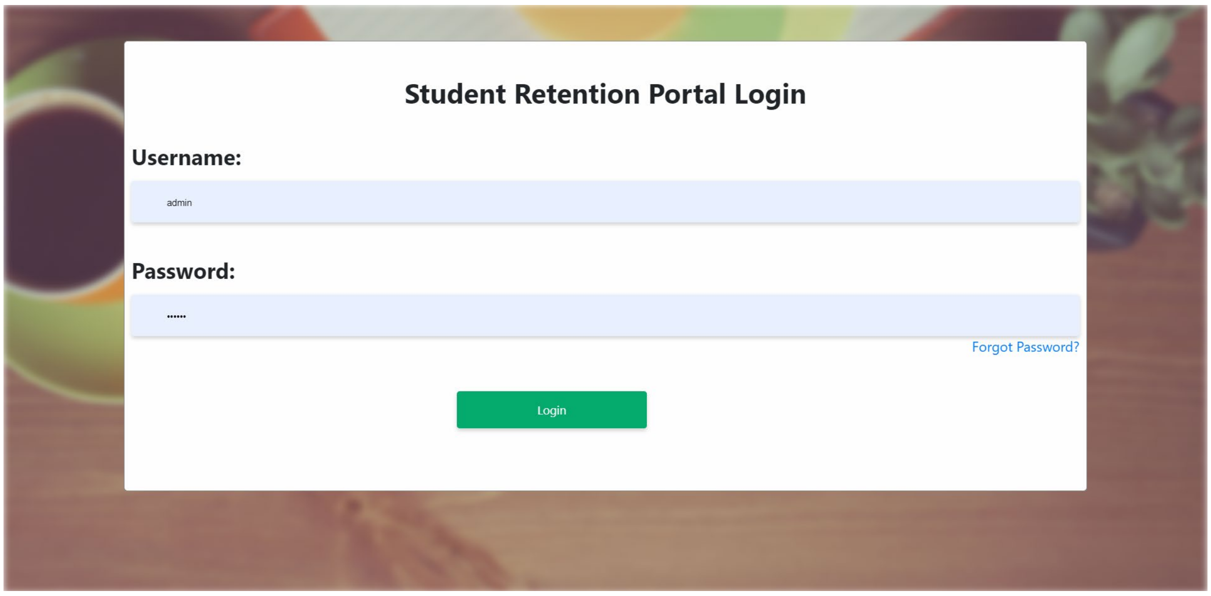
Module Engagement Dashboard

John Smith x238503812@student.ncirl.ie	Email	Jane Smith x238503153@student.ncirl.ie	Email
Jess Small x221403856@student.ncirl.ie	Email	Rebecca Re 101503856@student.ncirl.ie	Email
Jane Doe x248501326@student.ncirl.ie	Email	Rob Donald x132503856@student.ncirl.ie	Email
Karen Karr x20583856@student.ncirl.ie	Email	Paul Murphy x18303856@student.ncirl.ie	Email

The image above displays the dashboard wireframe. There is a pie chart displaying the class engagement for a particular week of content and a bar chart displaying the student's overall engagement for a particular module. In this wireframe, below the charts, there is a section where the user can choose to email particular students if they notice their engagement is down. This wireframe is very similar to the low fidelity prototype; however it has less graphs as its easier to read and less overwhelming for the user when visiting dashboard pages.

2.4.3 High Fidelity Prototypes

2.4.3.1 Login Page


Student Retention Portal Login

Username:

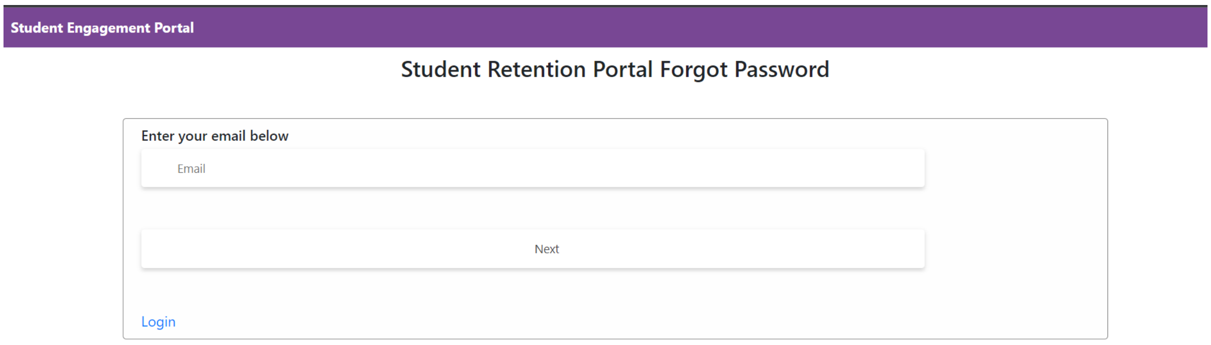
admin

Password:

••••••

In the image above, the high-fidelity prototype of the login page can be seen. There is no user registration on this page. The login page is similar to the wireframe, however it is more simplified and straight forward.

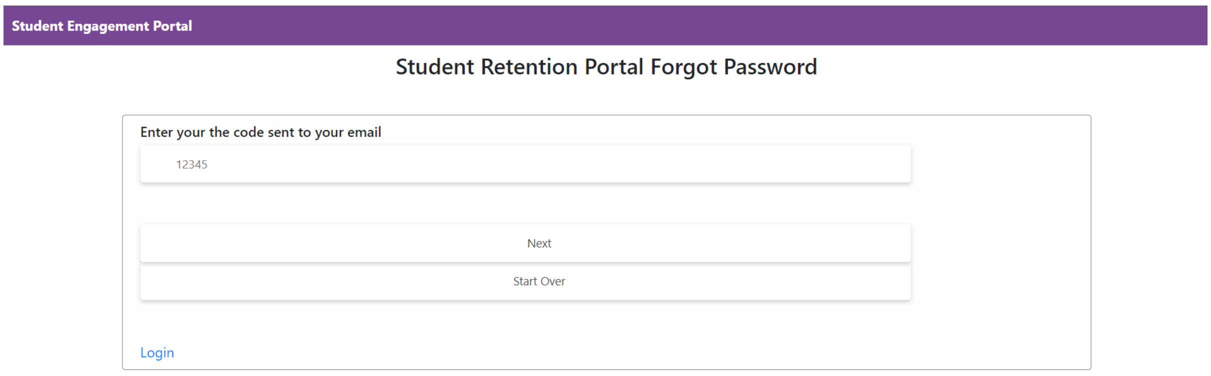
2.4.3.2 Forgot Password Page



Student Retention Portal Forgot Password

Enter your email below

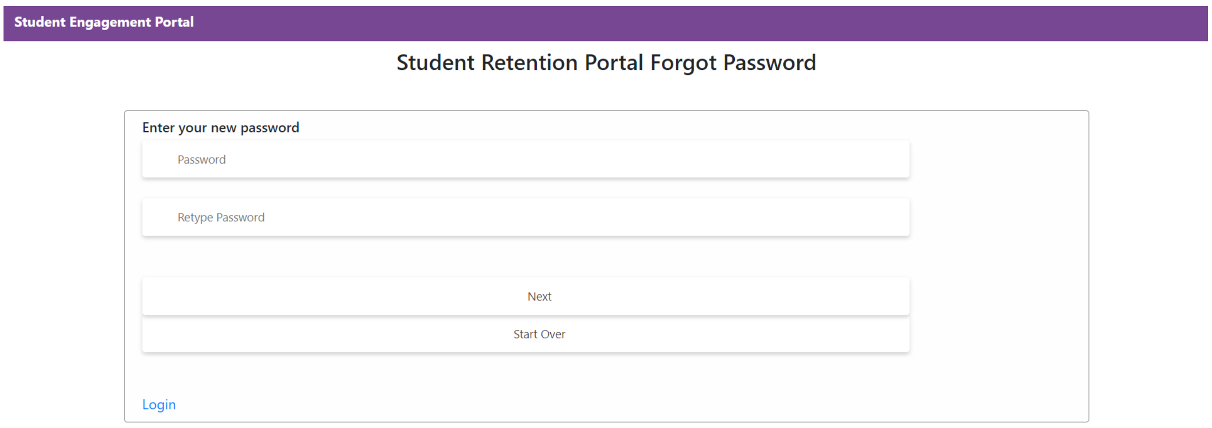
Email



Student Retention Portal Forgot Password

Enter your the code sent to your email

12345


Student Engagement Portal

Student Retention Portal Forgot Password

Enter your new password

Password

Retype Password

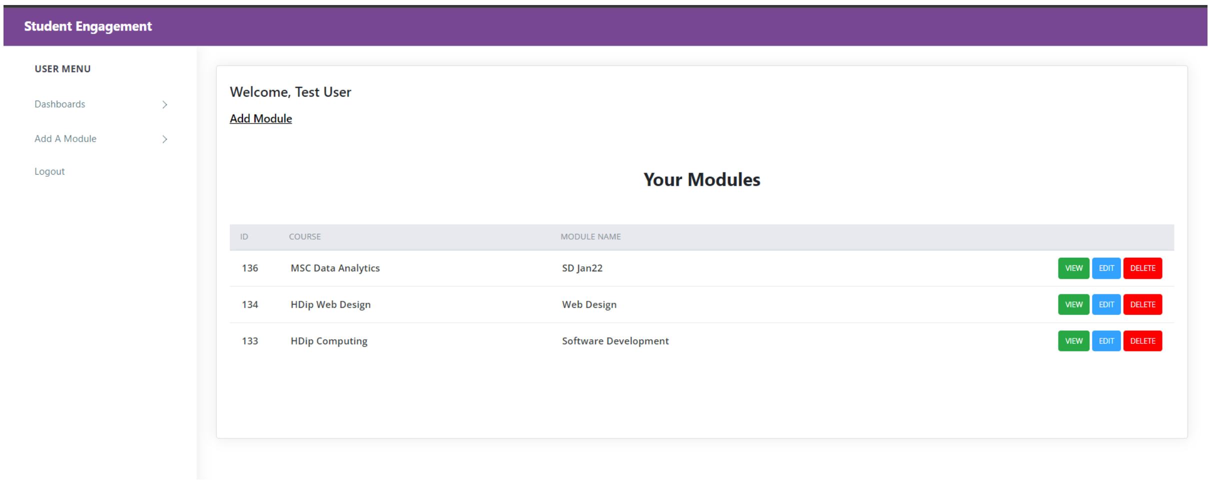
Next

Start Over

Login

The images above, are the images that are associated with the forgot password functionality. These pages were not in the wireframes or low fidelity prototypes as they were a last-minute development. These pages are simplistic and easy to navigate through.

2.4.3.3 User Home Page


Student Engagement

USER MENU

Dashboards

Add A Module

Logout

Welcome, Test User

Add Module

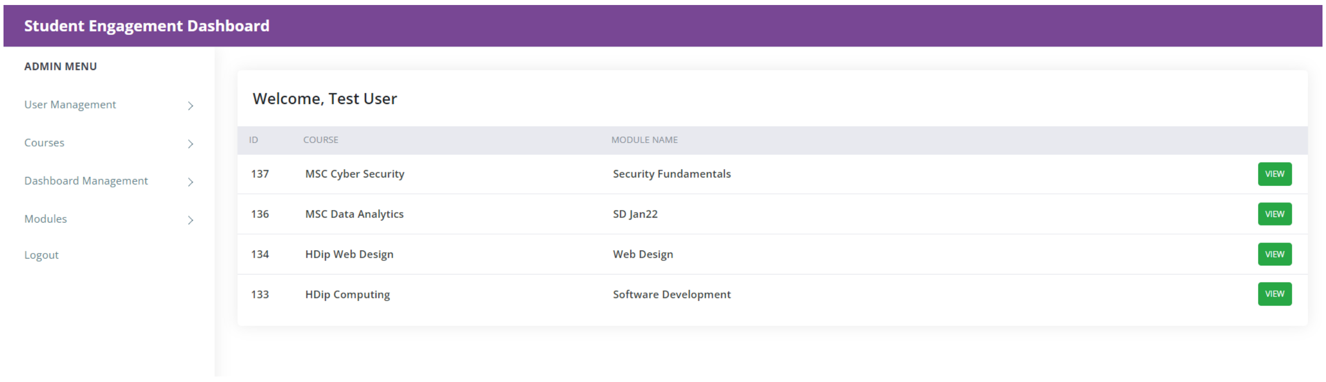
Your Modules

ID	COURSE	MODULE NAME	
136	MSC Data Analytics	SD Jan22	VIEW EDIT DELETE
134	HDip Web Design	Web Design	VIEW EDIT DELETE
133	HDip Computing	Software Development	VIEW EDIT DELETE

The image above depicts the users home page. The user home page is also the dashboard page from the side bar navigation. The user home page is very different to the wireframe. It now displays the users name and the modules they have created, if any. From the home page, users can view, edit or delete modules and their dashboards.

The graphical user interface for this page is very user friendly and easy to navigate, which was important as people with different technology experience would be using the Student Engagement Portal.

2.4.3.4 Admin Home Page

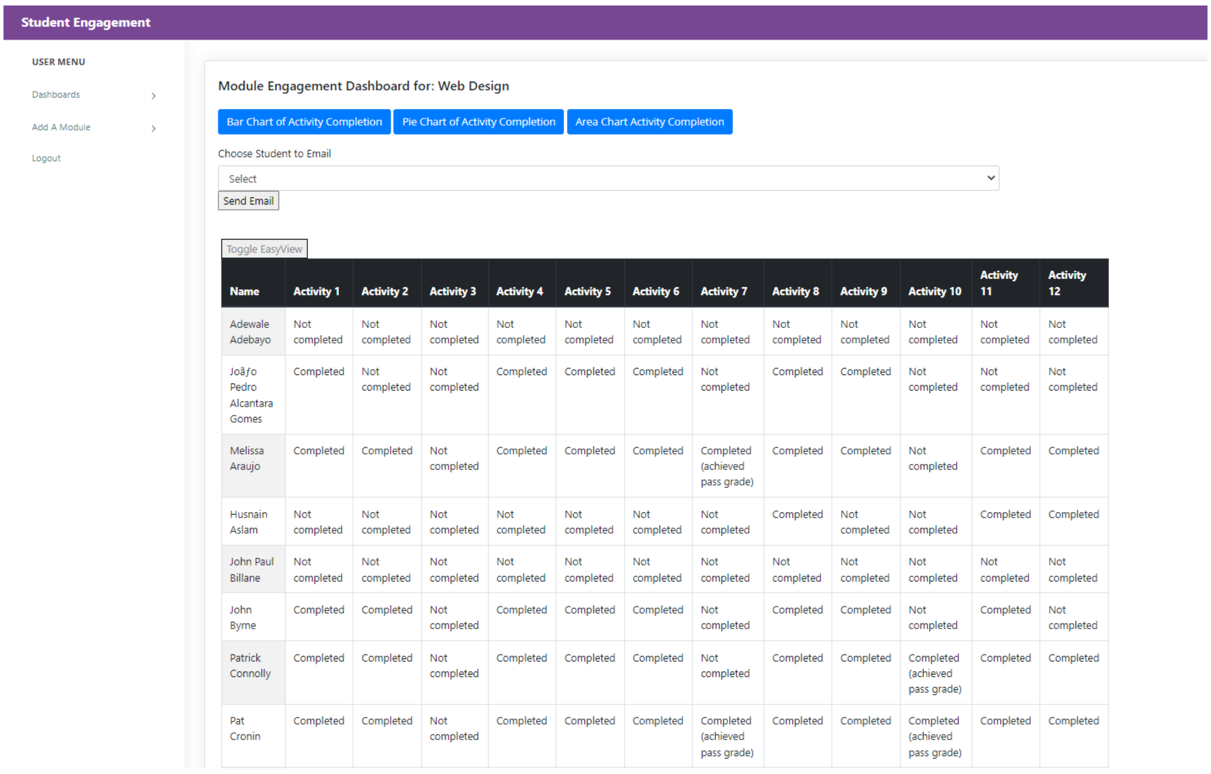


Welcome, Test User

ID	COURSE	MODULE NAME	
137	MSC Cyber Security	Security Fundamentals	VIEW
136	MSC Data Analytics	SD Jan22	VIEW
134	HDip Web Design	Web Design	VIEW
133	HDip Computing	Software Development	VIEW

The image above depicts the admin home page. The admin home page is also the dashboard management page from the side bar navigation. The admin home page has a similar layout to the user homepage however admins have a lot more access to features within the student engagement portal.

2.4.3.5 Dashboard Page

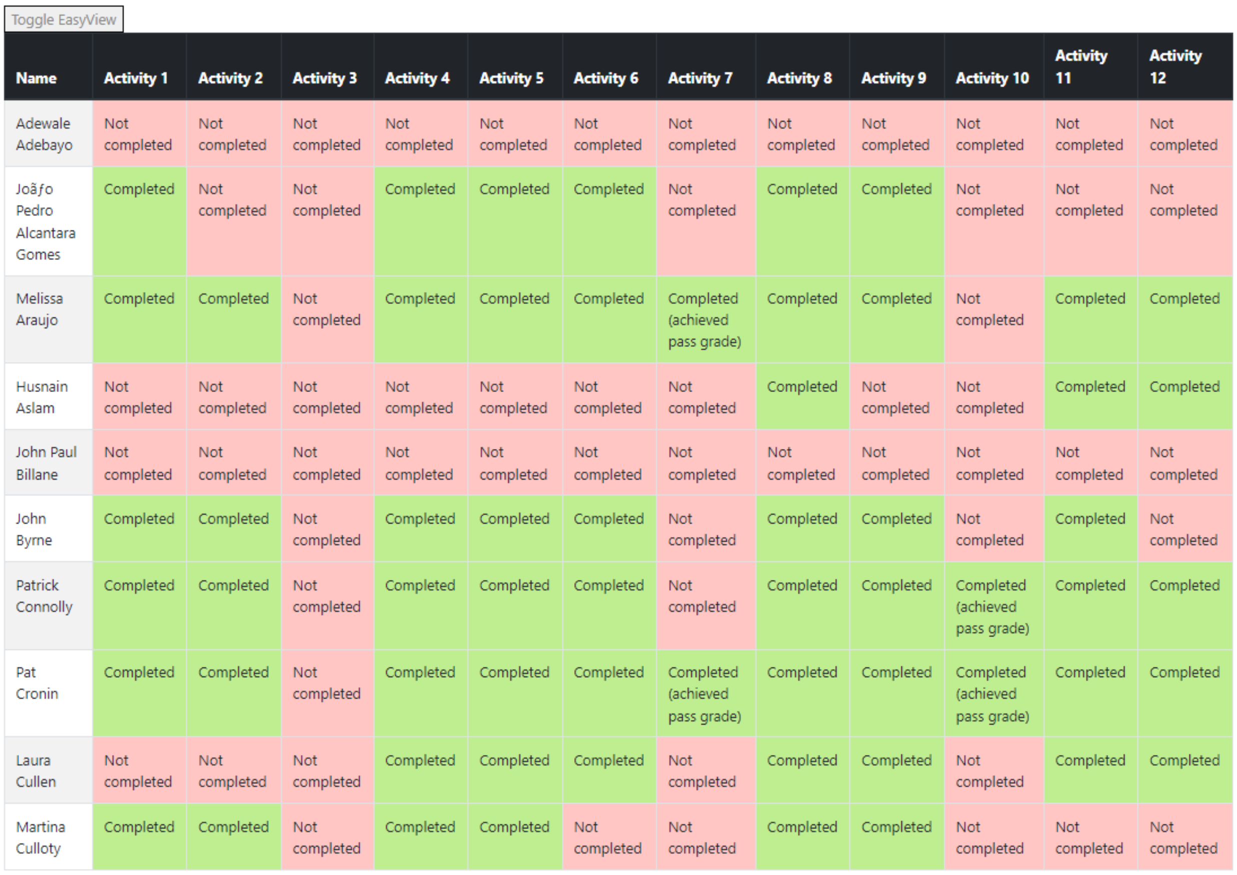

Student Engagement
USER MENU
Dashboards
Add A Module
Logout
Bar Chart of Activity Completion
Pie Chart of Activity Completion
Area Chart Activity Completion
Select
Send Email
Toggle EasyView

Module Engagement Dashboard for: Web Design

Choose Student to Email

Name	Activity 1	Activity 2	Activity 3	Activity 4	Activity 5	Activity 6	Activity 7	Activity 8	Activity 9	Activity 10	Activity 11	Activity 12
Adewale Adebayo	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed
JoÃ£o Pedro Alcantara Gomes	Completed	Not completed	Not completed	Completed	Completed	Completed	Not completed	Completed	Completed	Not completed	Not completed	Not completed
Melissa Araujo	Completed	Completed	Not completed	Completed	Completed	Completed	Completed (achieved pass grade)	Completed	Completed	Not completed	Completed	Completed
Husnain Aslam	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Completed	Not completed	Not completed	Completed	Completed
John Paul Billane	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed
John Byrne	Completed	Completed	Not completed	Completed	Completed	Completed	Not completed	Completed	Completed	Not completed	Completed	Not completed
Patrick Connolly	Completed	Completed	Not completed	Completed	Completed	Completed	Not completed	Completed	Completed	Completed (achieved pass grade)	Completed	Completed
Pat Cronin	Completed	Completed	Not completed	Completed	Completed	Completed	Completed (achieved pass grade)	Completed	Completed	Completed (achieved pass grade)	Completed	Completed

The module dashboard for the student Engagement Portal is the picture above. At the top of the page is a header that prints out the name of the module that the dashboard is for. Then there are three buttons that display the different generated charts. After that, there is a dropdown to select students to email. After the emails, there is a table that displays the students and their engagement for all activities. There is a toggle view button at the top of this table, which toggles the table to turn completed activities green and not completed, red. The main aim for the dashboard pages, was that they are easy to read and use. This dashboard is very straight forward. Images of the charts and emails can be found at 2.3.3 and 2.3.4

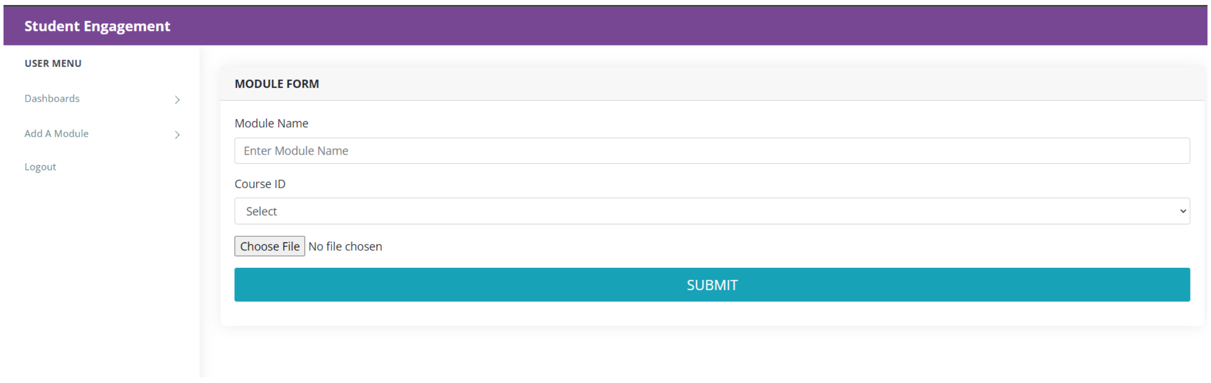


Toggle EasyView

Name	Activity 1	Activity 2	Activity 3	Activity 4	Activity 5	Activity 6	Activity 7	Activity 8	Activity 9	Activity 10	Activity 11	Activity 12
Adewale Adebayo	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed
Joãƒo Pedro Alcantara Gomes	Completed	Not completed	Not completed	Completed	Completed	Completed	Not completed	Completed	Completed	Not completed	Not completed	Not completed
Melissa Araujo	Completed	Completed	Not completed	Completed	Completed	Completed	Completed (achieved pass grade)	Completed	Completed	Not completed	Completed	Completed
Husnain Aslam	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Completed	Not completed	Not completed	Completed	Completed
John Paul Billane	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed	Not completed
John Byrne	Completed	Completed	Not completed	Completed	Completed	Completed	Not completed	Completed	Completed	Not completed	Completed	Not completed
Patrick Connolly	Completed	Completed	Not completed	Completed	Completed	Completed	Not completed	Completed	Completed	Completed (achieved pass grade)	Completed	Completed
Pat Cronin	Completed	Completed	Not completed	Completed	Completed	Completed	Completed (achieved pass grade)	Completed	Completed	Completed (achieved pass grade)	Completed	Completed
Laura Cullen	Not completed	Not completed	Not completed	Completed	Completed	Completed	Not completed	Completed	Completed	Not completed	Completed	Completed
Martina Culloty	Completed	Completed	Not completed	Completed	Completed	Not completed	Not completed	Completed	Completed	Not completed	Not completed	Not completed

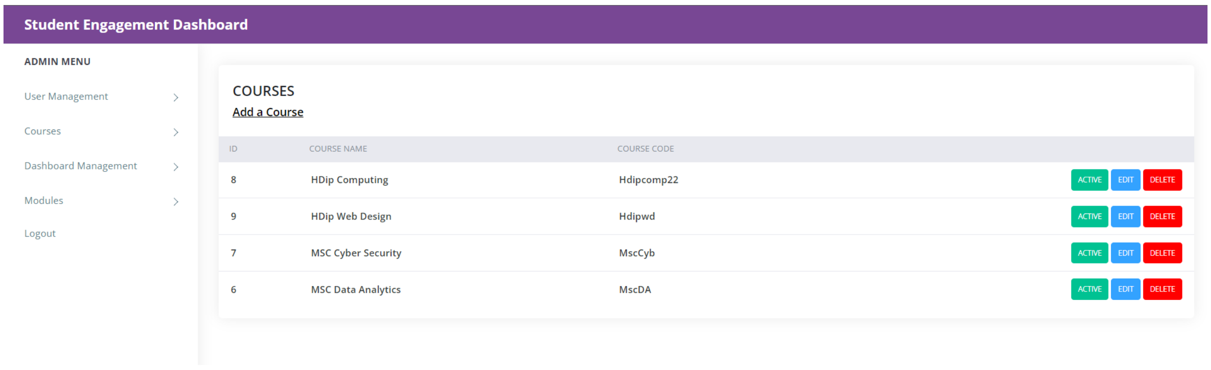
The above image is the result of clicking the 'Toggle EasyView' button at the top of the table displaying all the student engagement for activities.

2.4.3.6 Create/Edit Module



The create/edit module page has the same functionality in mind as the home page wireframe had, however its cleaner and gets more data from the user. On this page, there is a form that users are required to fill out in order to upload a csv file. Users need to add a module name and choose a course from the dropdown and choose a file on this page.

2.4.3.7 Courses Page



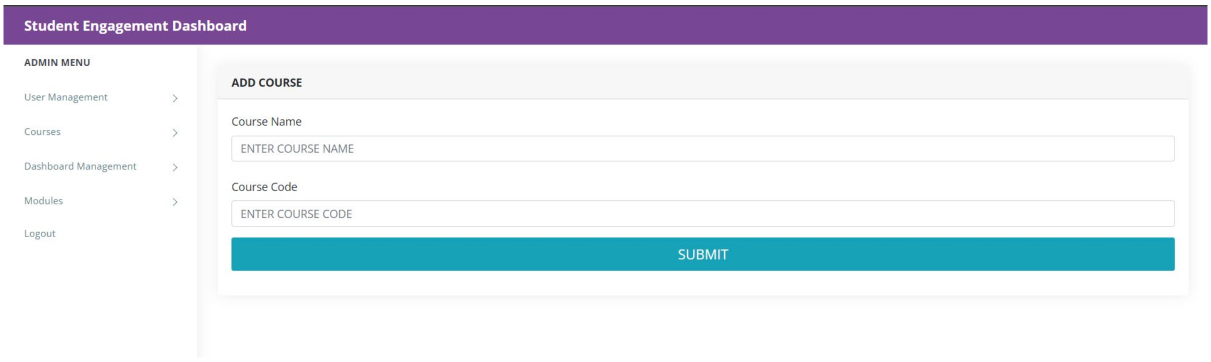
COURSES

Add a Course

ID	COURSE NAME	COURSE CODE	
8	HDip Computing	Hdipcomp22	ACTIVE EDIT DELETE
9	HDip Web Design	Hdipwd	ACTIVE EDIT DELETE
7	MSC Cyber Security	MscCyb	ACTIVE EDIT DELETE
6	MSC Data Analytics	MscDA	ACTIVE EDIT DELETE

The courses page is a simple page. It displays the courses that are in the database. From this page users can deactivate a course, edit or delete it.

2.4.3.8 Create/Edit Courses Page



ADD COURSE

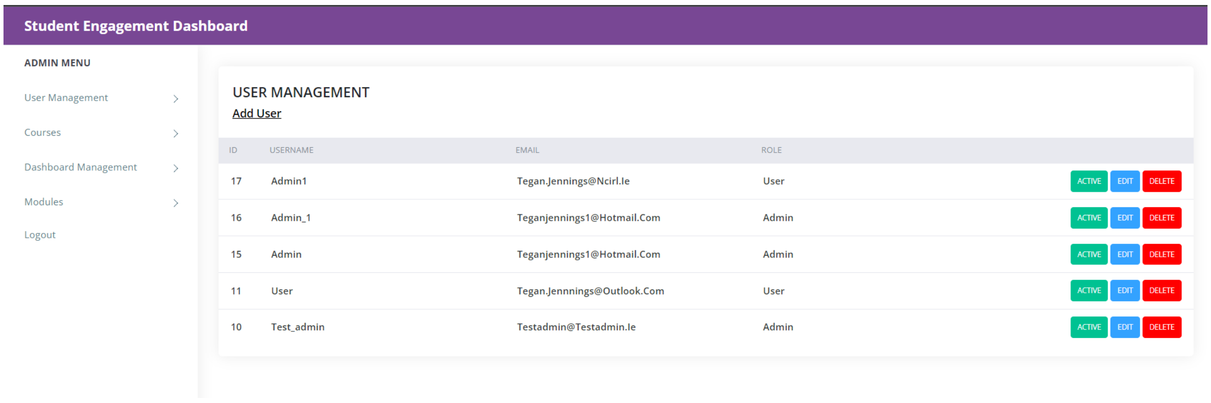
Course Name: ENTER COURSE NAME

Course Code: ENTER COURSE CODE

SUBMIT

The create/edit course page is a simple form page to get information for the database. On this page, there is a form that users are required to fill out in order to add courses for modules. Users need to add in a course name and id on this page.

2.4.3.9 User Management Page



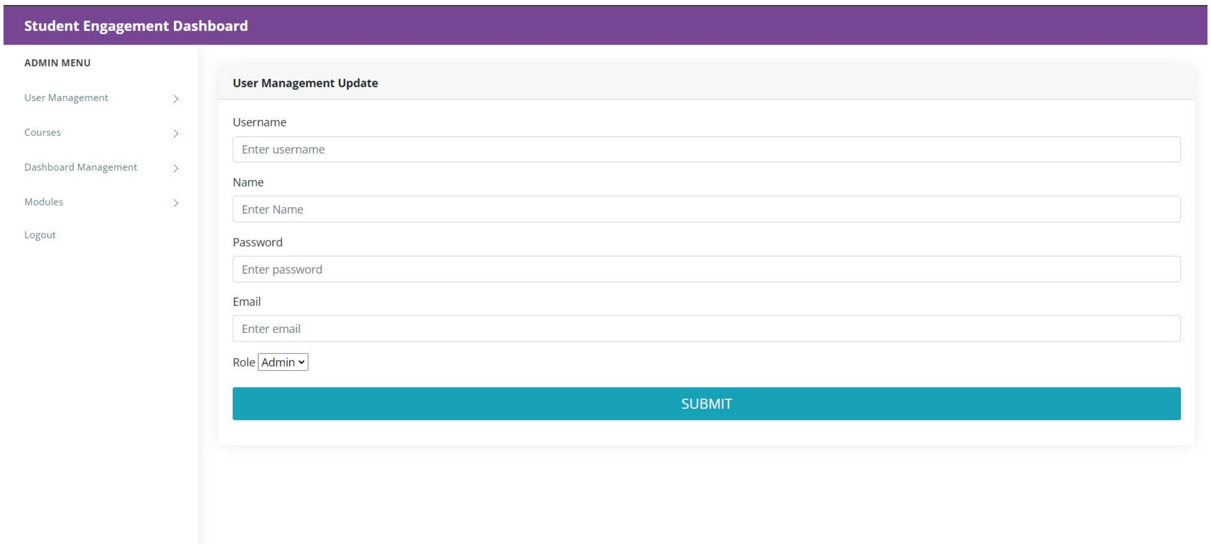
USER MANAGEMENT

Add User

ID	USERNAME	EMAIL	ROLE	
17	Admin1	Tegan.Jennings@Ncirl.Ie	User	ACTIVE EDIT DELETE
16	Admin_1	Teganjennings1@Hotmail.Com	Admin	ACTIVE EDIT DELETE
15	Admin	Teganjennings1@Hotmail.Com	Admin	ACTIVE EDIT DELETE
11	User	Tegan.Jennnings@Outlook.Com	User	ACTIVE EDIT DELETE
10	Test_admin	Testadmin@Testadmin.Ie	Admin	ACTIVE EDIT DELETE

The user management page displays all the current users in the database along with their name, email and the role they were assigned. Admins can deactivate a user from this page, they can also edit or delete a user, as well as create a new user. The interface for this page is following the layout of the other pages in order to keep consistency throughout the application.

2.4.3.10 Create/Edit Users Page



The create/edit users page follows the layout and design of the courses and modules create/edit pages. On this page, there is a form where users need to add in a username, name, password, email and the role of the user they are adding.

2.5. Testing

2.5.1 Unit Testing

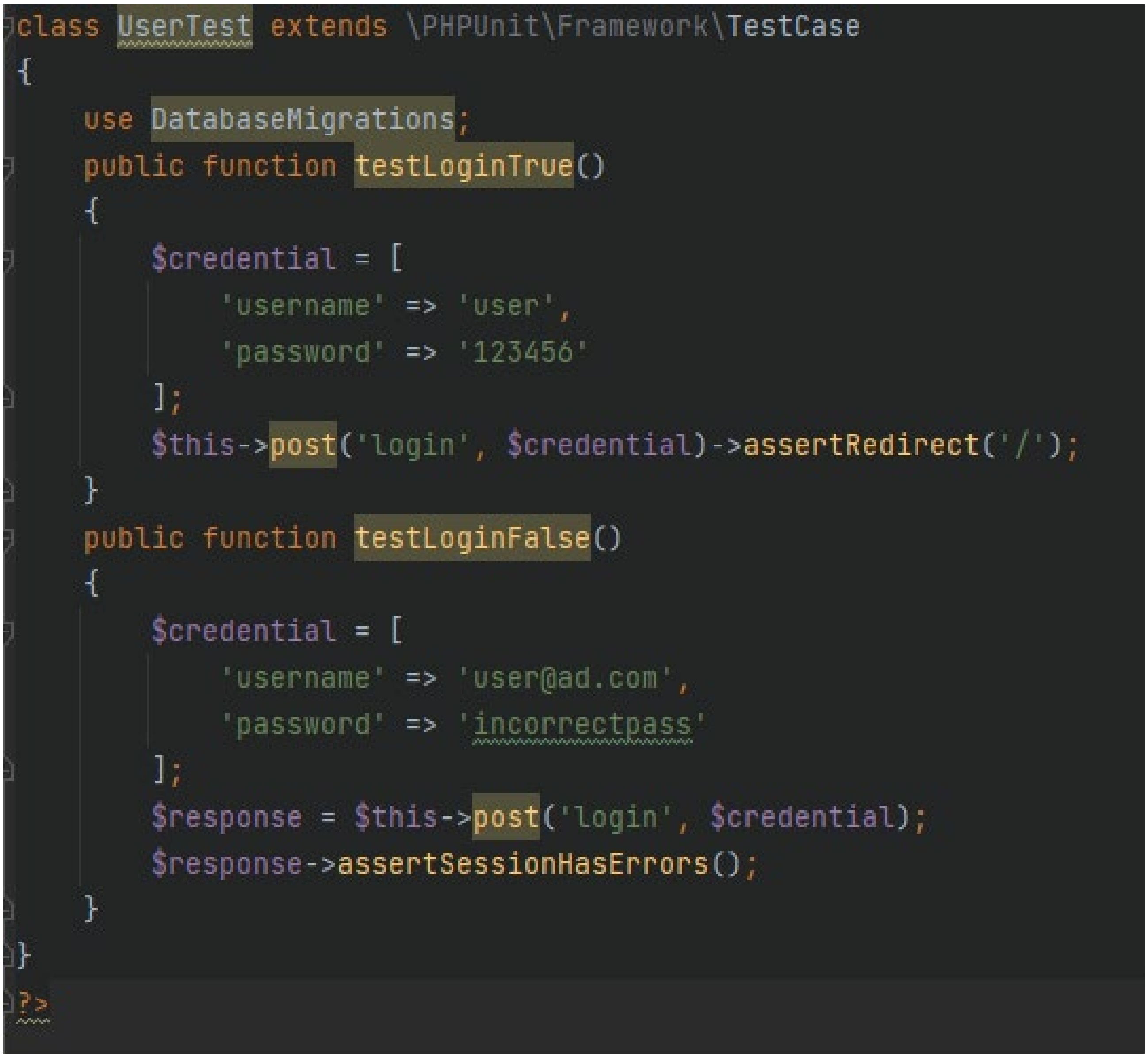
The method of testing I have chosen for my project is unit testing. Tin order to perform unit tests on the student engagement portal, PHPUnit will be utilised. I will be able to test the functionality of the application. For more information on PHPUnit, please see Technology 1.3.5.

Unit tests are important as it will check to see if the application functions as I continue to add to my project. For my application this will be very important as each part of my applications functionality relies on a previously functionality to work. For that reason, the unit tests will be extremely important for this application.

I created 3 different unit tests however these unit tests contain multiple test assertions. The unit tests are as follows:

- Login
- Modules
- Courses

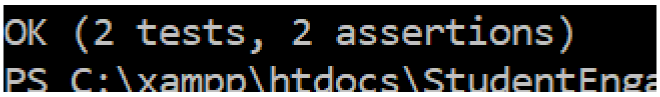
2.5.1.1 Login Unit Tests



```php
class UserTest extends \PHPUnit\Framework\TestCase
{
    use DatabaseMigrations;
    public function testLoginTrue()
    {
        $credential = [
            'username' => 'user',
            'password' => '123456'
        ];
        $this->post('login', $credential)->assertRedirect('/');
    }
    public function testLoginFalse()
    {
        $credential = [
            'username' => 'user@ad.com',
            'password' => 'incorrectpass'
        ];
        $response = $this->post('login', $credential);
        $response->assertSessionHasErrors();
    }
}
?>
```

In the test above, there are two functions. The first function in this test is to check if a user can log in successfully with the correct credentials. The second function is to make sure that users with incorrect credentials cannot log in to the Student Engagement Portal.

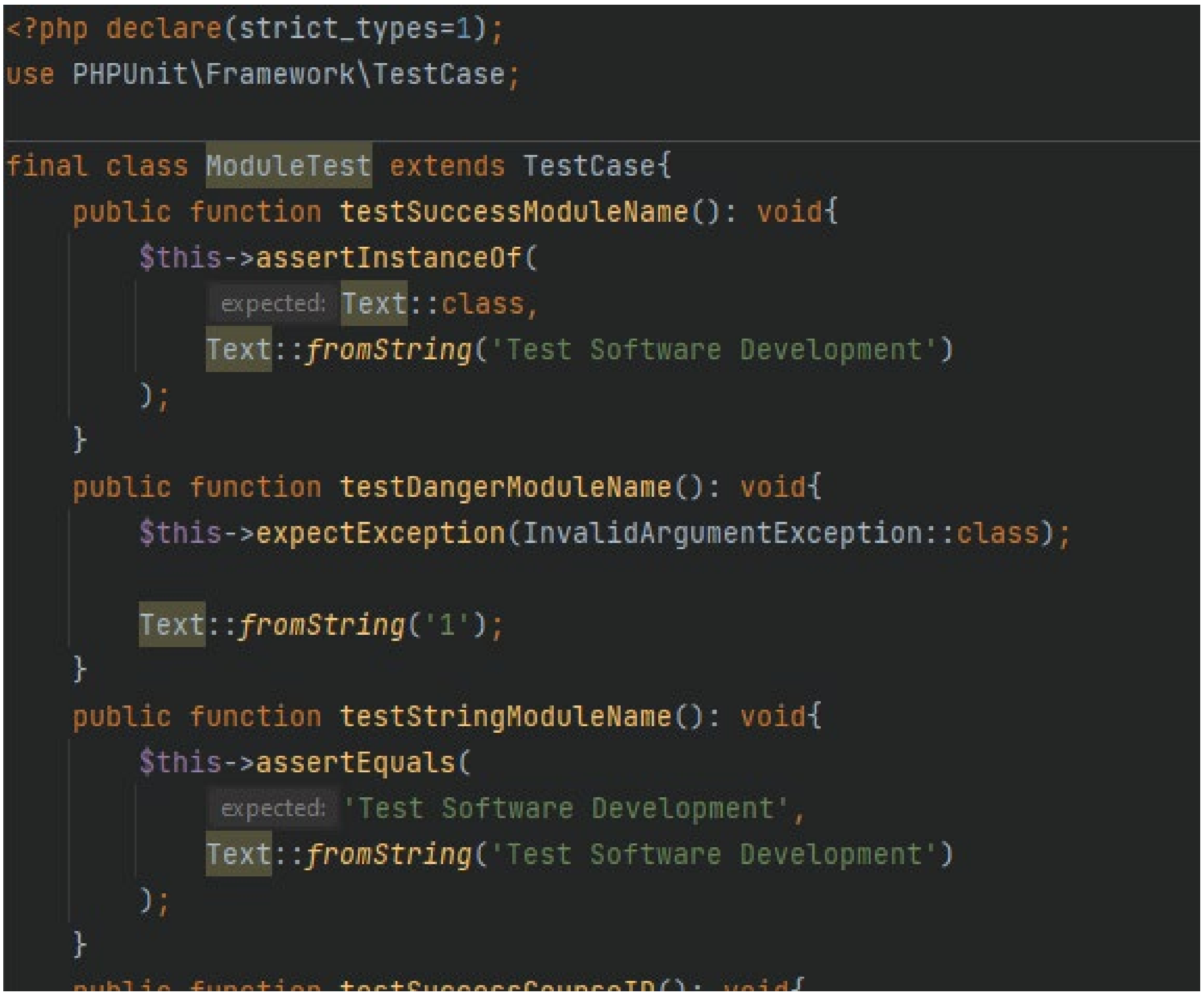
When these functions are run, we get the following result:



```
OK (2 tests, 2 assertions)
PS C:\xampp\htdocs\StudentEnga
```

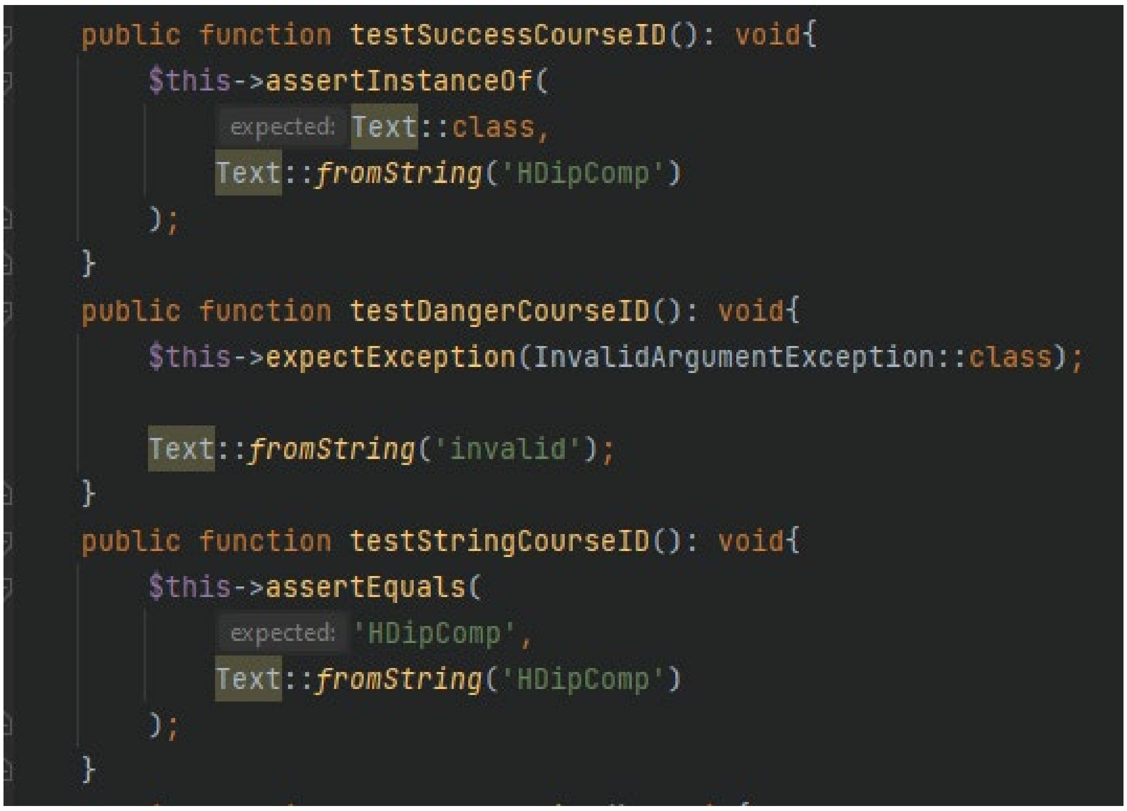
This result indicates that the tests ran successfully.

2.5.1.2 Modules Unit Tests



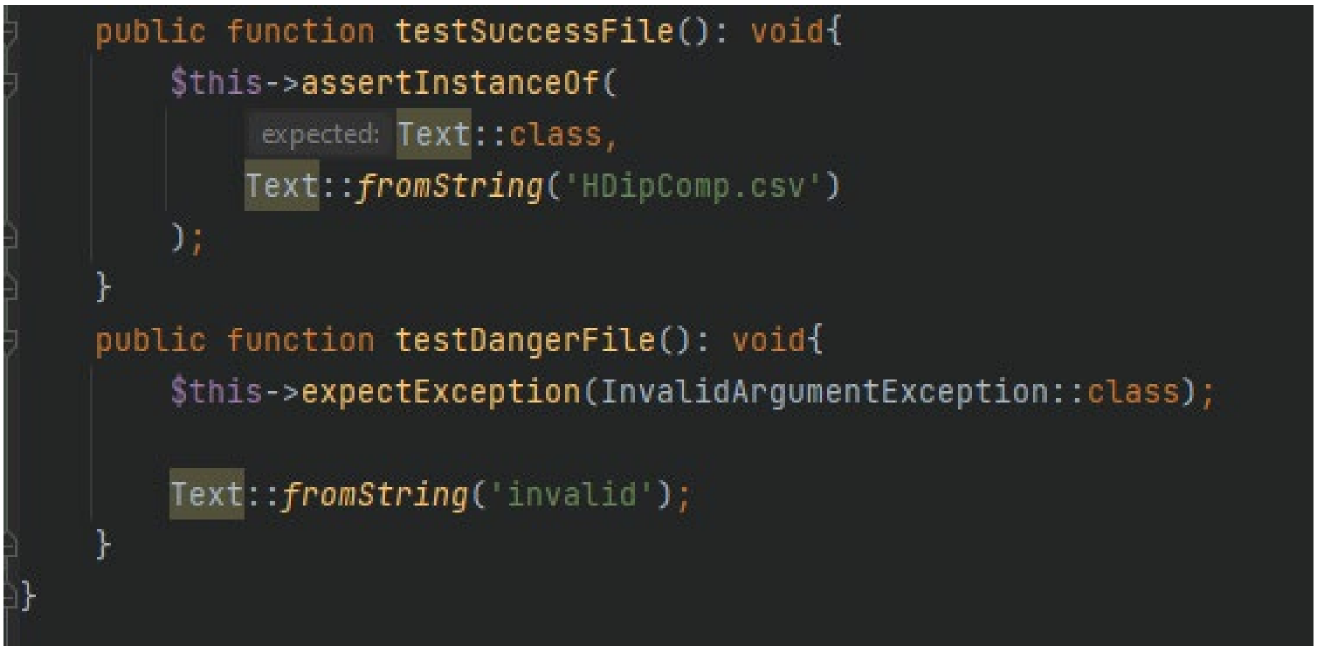
```
<?php declare(strict_types=1);
use PHPUnit\Framework\TestCase;

final class ModuleTest extends TestCase{
    public function testSuccessModuleName(): void{
        $this->assertInstanceOf(
            expected: Text::class,
            Text::fromString('Test Software Development')
        );
    }
    public function testDangerModuleName(): void{
        $this->expectException(InvalidArgumentException::class);

        Text::fromString('1');
    }
    public function testStringModuleName(): void{
        $this->assertEquals(
            expected: 'Test Software Development',
            Text::fromString('Test Software Development')
        );
    }
```

In the test above, there are three functions. The first function in this test is to check if a user can successfully add a module name to the modules upload form. The second function is to make sure that invalid module names cause errors and don't get added to the database. The third function is to check whether the module name can be sent to a string, in order to be used later.



```
    public function testSuccessCourseID(): void{
        $this->assertInstanceOf(
            expected: Text::class,
            Text::fromString('HDipComp')
        );
    }
    public function testDangerCourseID(): void{
        $this->expectException(InvalidArgumentException::class);

        Text::fromString('invalid');
    }
    public function testStringCourseID(): void{
        $this->assertEquals(
            expected: 'HDipComp',
            Text::fromString('HDipComp')
        );
    }
```

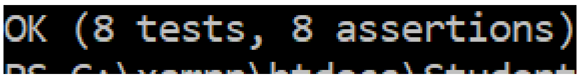
In the test above, there are three functions. The first function in this test is to check if a user can successfully add a course ID to the modules upload form. The second function is to make sure that invalid course ids cause errors and don't get added to the database. The third function is to check whether the course id can be sent to a string, in order to be used later.



```
    public function testSuccessFile(): void{
        $this->assertInstanceOf(
            expected: Text::class,
            Text::fromString('HDipComp.csv')
        );
    }
    public function testDangerFile(): void{
        $this->expectException(InvalidArgumentException::class);

        Text::fromString('invalid');
    }
}
```

In the test above, there are two functions. The first function in this test is to check if a user can successfully add a file to the modules upload form. The second function is to make sure that invalid files cause errors and don't get added to the database.

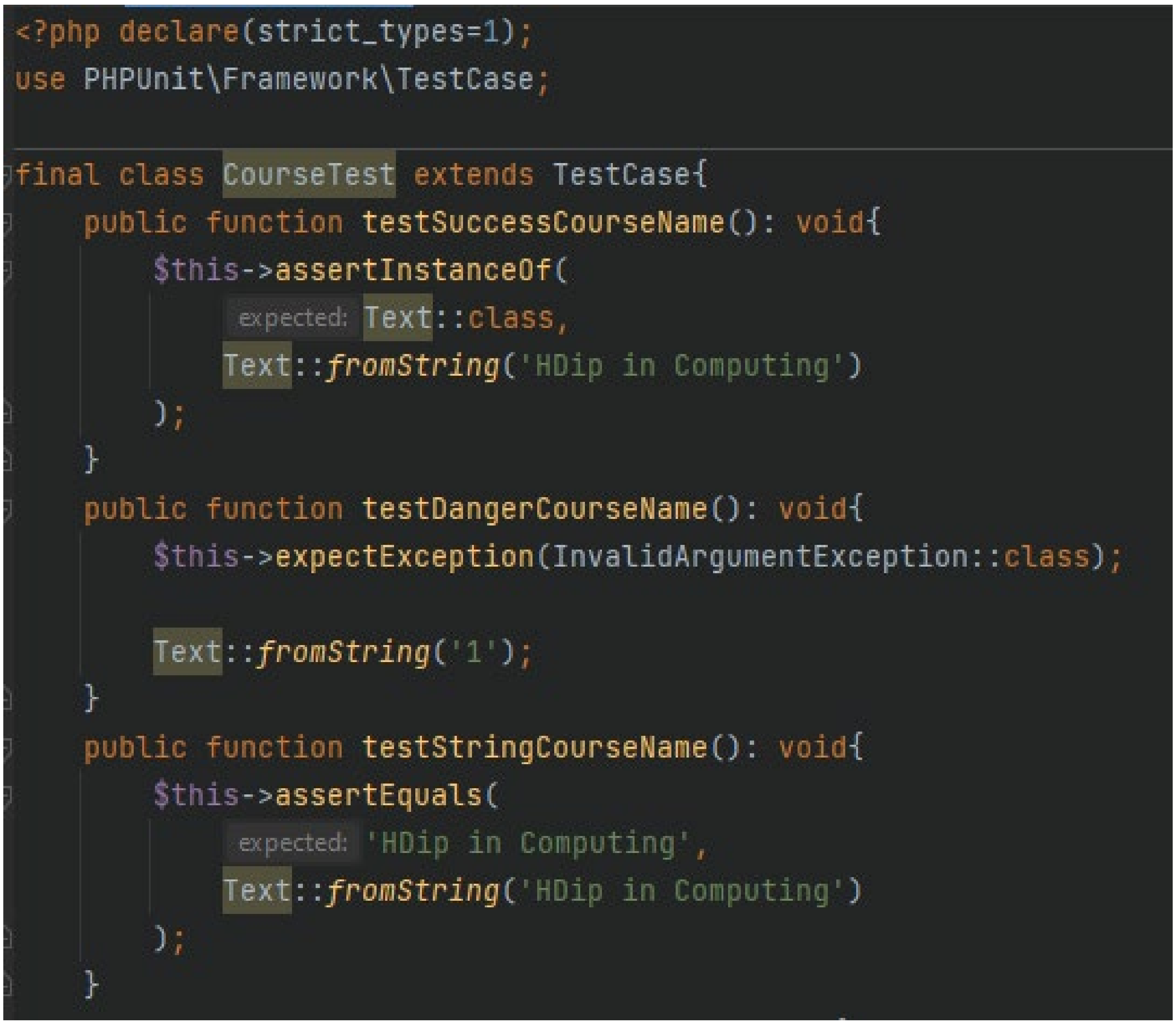
When all eight of these functions run in the Module Tests, we get the following result:



```
OK (8 tests, 8 assertions)
```

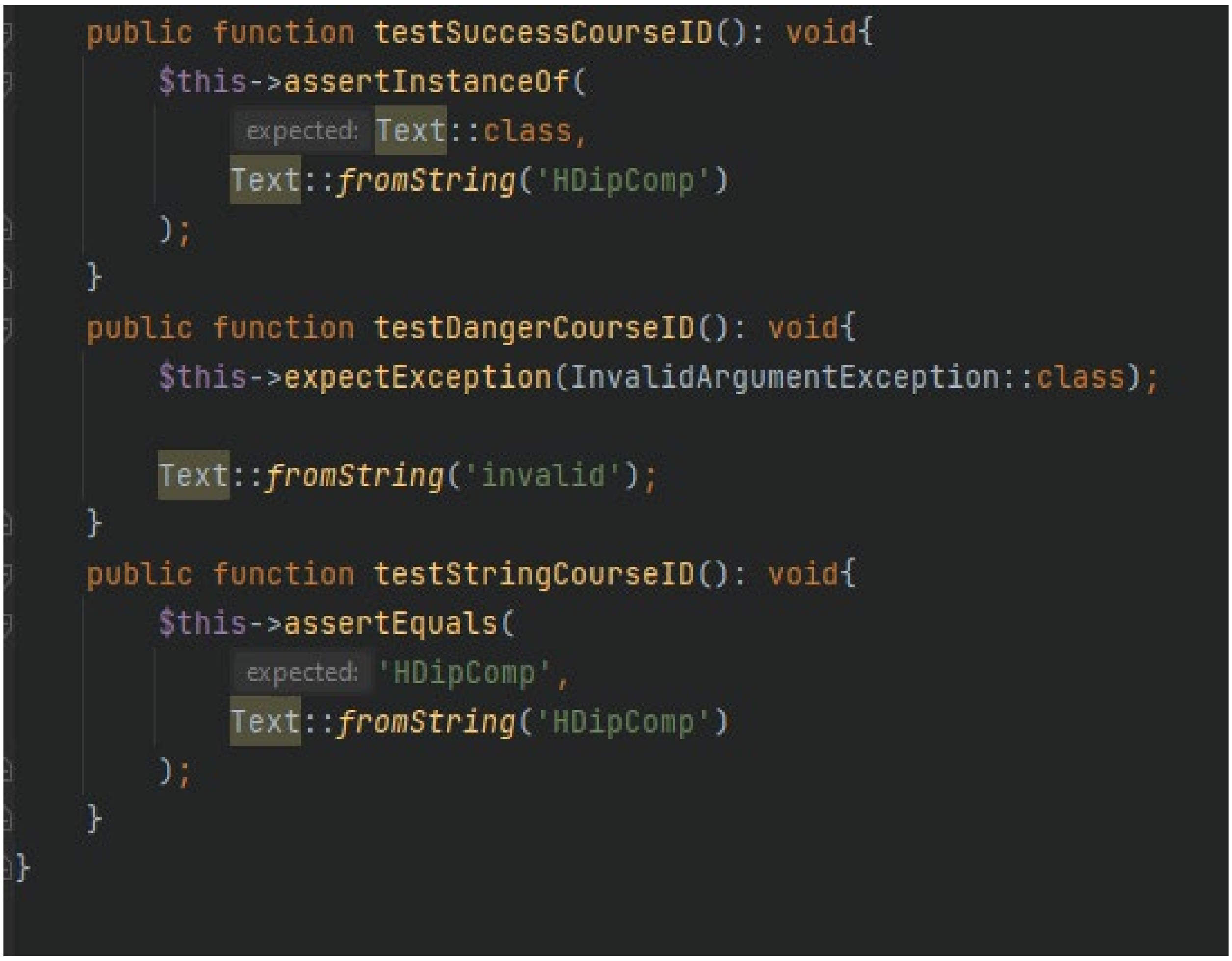
This result indicates that the tests ran successfully.

2.5.1.3 Courses Unit Tests



```php
<?php declare(strict_types=1);
use PHPUnit\Framework\TestCase;

final class CourseTest extends TestCase{
    public function testSuccessCourseName(): void{
        $this->assertInstanceOf(
            expected: Text::class,
            Text::fromString('HDip in Computing')
        );
    }
    public function testDangerCourseName(): void{
        $this->expectException(InvalidArgumentException::class);

        Text::fromString('1');
    }
    public function testStringCourseName(): void{
        $this->assertEquals(
            expected: 'HDip in Computing',
            Text::fromString('HDip in Computing')
        );
    }
```

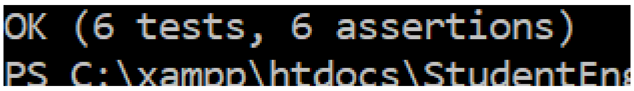
In the test above, there are three functions. The first function in this test is to check if a user can successfully add a course name to the courses upload form. The second function is to make sure that invalid courses names cause errors and don't get added to the database. The third function is to check whether the course name can be sent to a string, in order to be used later.



```
    public function testSuccessCourseID(): void{
        $this->assertInstanceOf(
            expected: Text::class,
            Text::fromString('HDipComp')
        );
    }
    public function testDangerCourseID(): void{
        $this->expectException(InvalidArgumentException::class);

        Text::fromString('invalid');
    }
    public function testStringCourseID(): void{
        $this->assertEquals(
            expected: 'HDipComp',
            Text::fromString('HDipComp')
        );
    }
}
```

In the test above, there are three functions. The first function in this test is to check if a user can successfully add a course ID to the courses upload form. The second function is to make sure that invalid course ID cause errors and don't get added to the database. The third function is to check whether the course ID can be sent to a string, in order to be used later.

When all six of these functions run in the Course Tests, we get the following result:



```
OK (6 tests, 6 assertions)
PS C:\xampp\htdocs\StudentEn
```

This result indicates that the tests ran successfully.

2.5.2 End User Testing

I completed end user testing towards the end of development for the Student Engagement Portal. I completed this testing to improve the applications layout and accessibility to improve on the design and ease of use of the application.

This testing was con by 5 different participants, each who represented a main user group who would be using the application. Some individuals work as administrators in colleges, others were technology illiterate, and some were in the teaching field. An informed consent form was signed by each of the participants to follow the research integrity guidelines, see appendix 6.4.2.

Each participant was greeted with the following introduction to the testing process once it began: "Hello, thank you for putting aside some of your time to test the student engagement portal. This application is at the end of its development so we would like some feedback to potentially improve the design."

The results of the testing are available in Appendix 6.3.3.

2.5.2.1 Trunk Test

For the trunk test, participants were presented at the dashboard and asked the following:

- What is this application?
- Can you see any major features?
- Where can you create a module?
- Where can you delete a module?
- Where can you view a module dashboard?
- Where can you email students?
- Where do you logout?

Recording of trunk test

2.5.2.2 Think Aloud Test

For think aloud testing, participants were given a number of tasks and asked to speak aloud their thinking process whilst completing those tasks. Using the think aloud testing, we were able to get a better understanding of how a user will approach the application and how easy the navigation is. The tasks are as follows:

- Log into application
- Create a Module
- View a module dashboard

Recording of think aloud test

2.5.2.3 5 Second Test

For the 5 second test, participants were shown a screenshot for 5 seconds and then after the 5 seconds, they were asked what they remembered from that screenshot. Participants were informed prior to seeing the screenshot, they would only have 5 seconds to view the screenshot. This testing was conducted to understand if the content on each page of the application was clear and concise. The questions that will be asked are as follows:

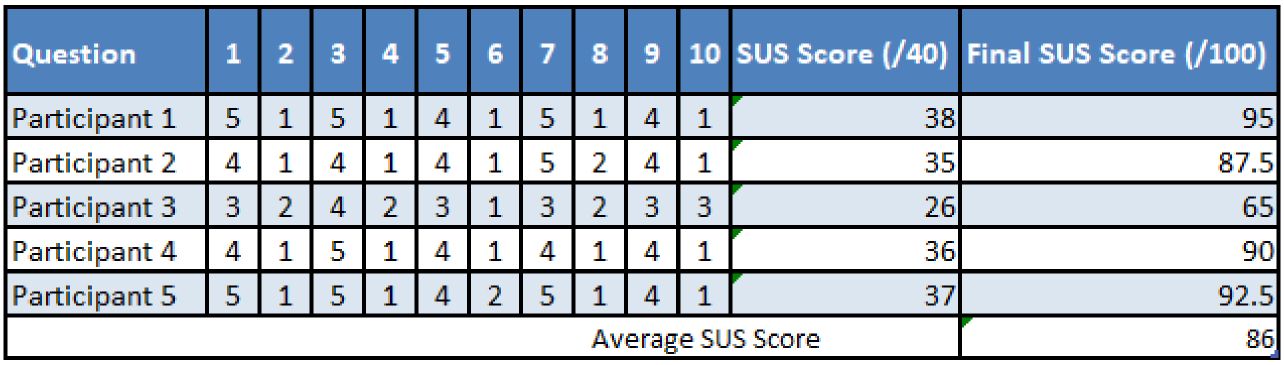
- What is this applications name?
- What is the application about and who is this project target market?
- What does the application offer in terms of features?

Recording of 5 second test

2.5.2.4 System Usability Scale Questionnaire

To debrief participants after the testing finished, participants were asked to complete an exit survey. The exit survey was the system usability scale (SUS) questionnaire. There were 10 SUS questions that were simple and very useful for receiving critical feedback.

After calculating the SUS score, the application had an average of ___ score. This was great feedback. The exit survey can be found in Appendix 6.3.3.4

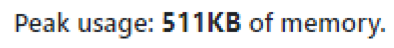


Question	1	2	3	4	5	6	7	8	9	10	SUS Score (/40)	Final SUS Score (/100)
Participant 1	5	1	5	1	4	1	5	1	4	1	38	95
Participant 2	4	1	4	1	4	1	5	2	4	1	35	87.5
Participant 3	3	2	4	2	3	1	3	2	3	3	26	65
Participant 4	4	1	5	1	4	1	4	1	4	1	36	90
Participant 5	5	1	5	1	4	2	5	1	4	1	37	92.5
Average SUS Score												86

2.6. Evaluation

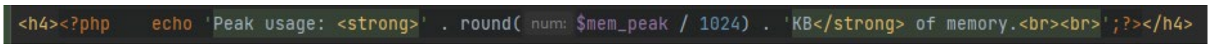
2.6.1 Ram

PHP Servers typically use 128mb of Ram per process. (Anon., 2022) When the Student Engagement Portals Ram usage was checked, it never peaked over 511kb. This is a tiny amount considering the size of the Student Engagement Portal application. However, given that PHP servers only have 128MB of RAM available to them, it's very positive the RAM amount the application was using.

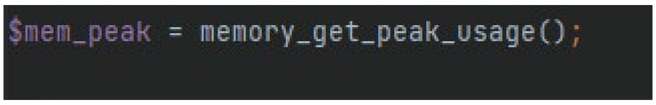


Peak usage: **511KB** of memory.

To get the RAM usage, a function was called and then echoed. The function was called memory_get_peak_usage. (Anon., n.d.)



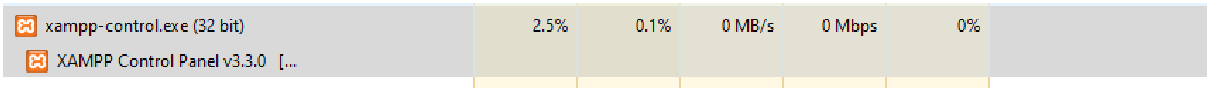
```
<h4><?php   echo 'Peak usage: <strong>' . round( num: $mem_peak / 1024) . 'KB</strong> of memory.<br><br>';?></h4>
```



```
$mem_peak = memory_get_peak_usage();
```

2.6.2 CPU

Whilst monitoring the student engagement portals CPU usage, it didn't go higher than 5%. This means that the student engagement platform doesn't require many resources in order to run and use. This would mean that the response time for loading pages and creating objects should be very quick.



```
xampp-control.exe (32 bit)        2.5%    0.1%    0 MB/s    0 Mbps    0%
  XAMPP Control Panel v3.3.0  [...
```

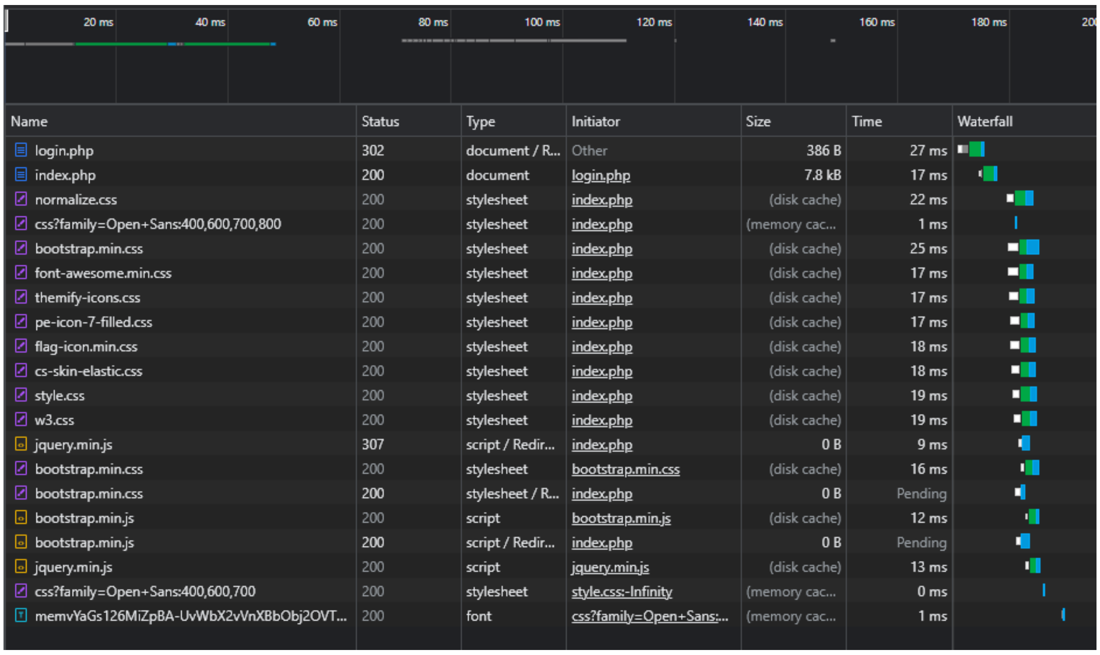
During this testing created a module and viewed the dashboard which is the expected behaviour of an average user.

2.6.3 Response Time

With ease of navigation as a requirement for the Student Engagement Platform, a quick response time was desired. Using the inspect element and the network tab, I was able to check the response time of all my pages.

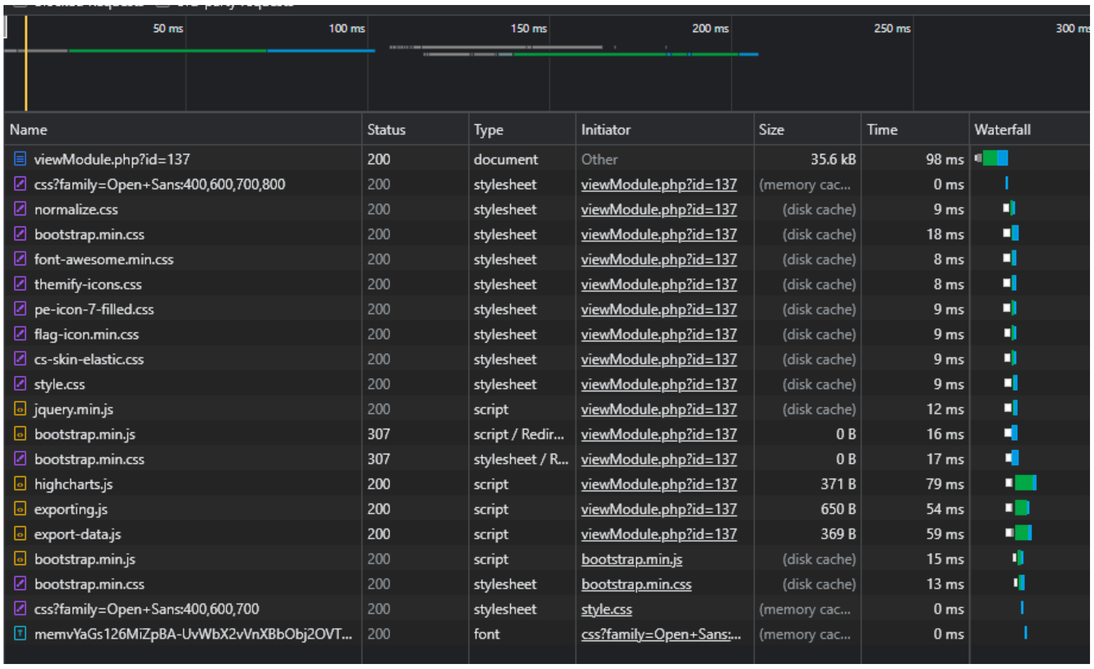
After checking the response time for all pages, the longest response time was 98ms and the quickest was 5ms. This is a huge benefit for the Student Engagement Portal as it confirms that the application has extremely quick loads time for all pages.

2.6.3.1 Login Response Time



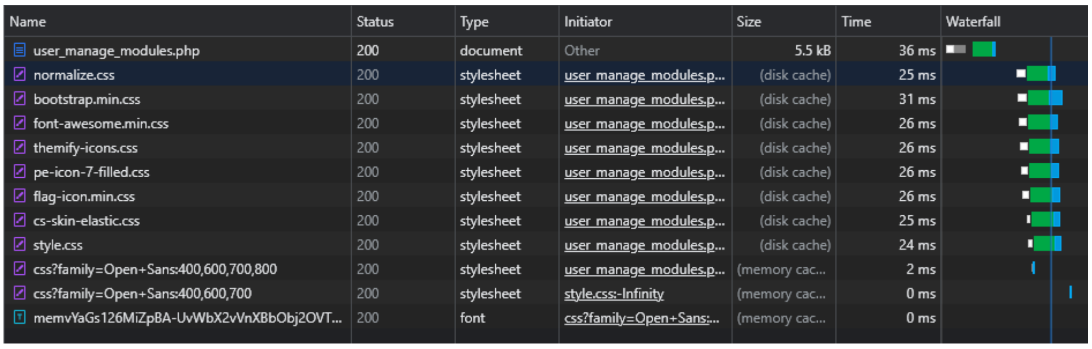
Name	Status	Type	Initiator	Size	Time	Waterfall
login.php	302	document / R...	Other	386 B	27 ms	
index.php	200	document	login.php	7.8 kB	17 ms	
normalize.css	200	stylesheet	index.php	(disk cache)	22 ms	
css?family=Open+Sans:400,600,700,800	200	stylesheet	index.php	(memory cac...	1 ms	
bootstrap.min.css	200	stylesheet	index.php	(disk cache)	25 ms	
font-awesome.min.css	200	stylesheet	index.php	(disk cache)	17 ms	
themify-icons.css	200	stylesheet	index.php	(disk cache)	17 ms	
pe-icon-7-filled.css	200	stylesheet	index.php	(disk cache)	17 ms	
flag-icon.min.css	200	stylesheet	index.php	(disk cache)	18 ms	
cs-skin-elastic.css	200	stylesheet	index.php	(disk cache)	18 ms	
style.css	200	stylesheet	index.php	(disk cache)	19 ms	
w3.css	200	stylesheet	index.php	(disk cache)	19 ms	
jquery.min.js	307	script / Redir...	index.php	0 B	9 ms	
bootstrap.min.css	200	stylesheet	bootstrap.min.css	(disk cache)	16 ms	
bootstrap.min.css	200	stylesheet / R...	index.php	0 B	Pending	
bootstrap.min.js	200	script	bootstrap.min.js	(disk cache)	12 ms	
bootstrap.min.js	200	script / Redir...	index.php	0 B	Pending	
jquery.min.js	200	script	jquery.min.js	(disk cache)	13 ms	
css?family=Open+Sans:400,600,700	200	stylesheet	style.css:-Infinity	(memory cac...	0 ms	
memvYaGs126MiZpBA-UvWbX2vVnXBbObj2OVT...	200	font	css?family=Open+Sans:...	(memory cac...	1 ms	

2.6.3.2 View Module Response Time



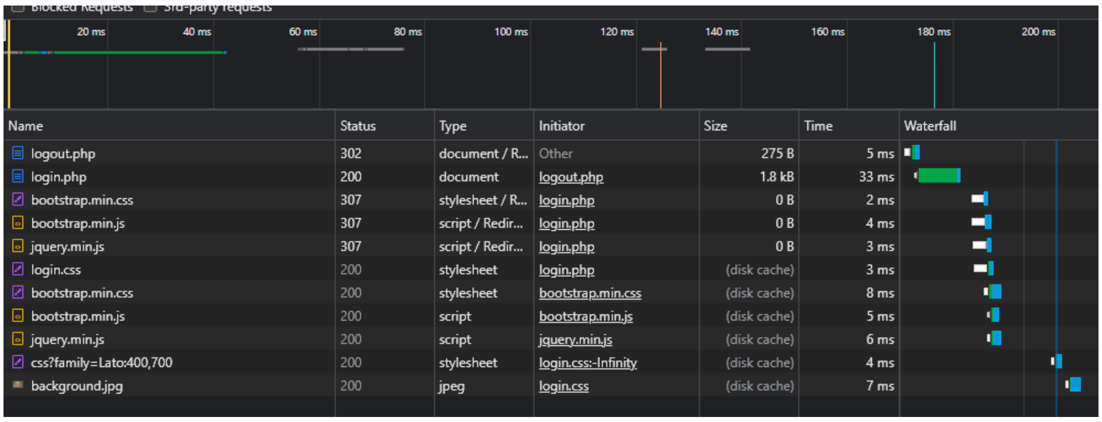
Name	Status	Type	Initiator	Size	Time	Waterfall
viewModule.php?id=137	200	document	Other	35.6 kB	98 ms	
css?family=Open+Sans:400,600,700,800	200	stylesheet	viewModule.php?id=137	(memory cac...	0 ms	
normalize.css	200	stylesheet	viewModule.php?id=137	(disk cache)	9 ms	
bootstrap.min.css	200	stylesheet	viewModule.php?id=137	(disk cache)	18 ms	
font-awesome.min.css	200	stylesheet	viewModule.php?id=137	(disk cache)	8 ms	
themify-icons.css	200	stylesheet	viewModule.php?id=137	(disk cache)	8 ms	
pe-icon-7-filled.css	200	stylesheet	viewModule.php?id=137	(disk cache)	9 ms	
flag-icon.min.css	200	stylesheet	viewModule.php?id=137	(disk cache)	9 ms	
cs-skin-elastic.css	200	stylesheet	viewModule.php?id=137	(disk cache)	9 ms	
style.css	200	stylesheet	viewModule.php?id=137	(disk cache)	9 ms	
jquery.min.js	200	script	viewModule.php?id=137	(disk cache)	12 ms	
bootstrap.min.js	307	script / Redir...	viewModule.php?id=137	0 B	16 ms	
bootstrap.min.css	307	stylesheet / R...	viewModule.php?id=137	0 B	17 ms	
highcharts.js	200	script	viewModule.php?id=137	371 B	79 ms	
exporting.js	200	script	viewModule.php?id=137	650 B	54 ms	
export-data.js	200	script	viewModule.php?id=137	369 B	59 ms	
bootstrap.min.js	200	script	bootstrap.min.js	(disk cache)	15 ms	
bootstrap.min.css	200	stylesheet	bootstrap.min.css	(disk cache)	13 ms	
css?family=Open+Sans:400,600,700	200	stylesheet	style.css	(memory cac...	0 ms	
memvYaGs126MiZpBA-UvWbX2vVnXBbObj2OVT...	200	font	css?family=Open+Sans:...	(memory cac...	0 ms	

2.6.3.3 Add/Edit Module Response Time



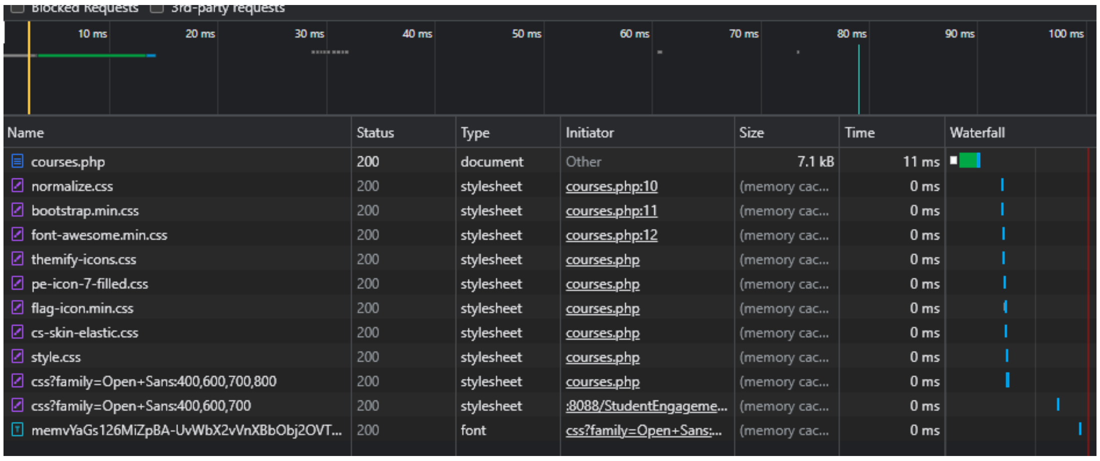
Name	Status	Type	Initiator	Size	Time	Waterfall
user_manage_modules.php	200	document	Other	5.5 kB	36 ms	
normalize.css	200	stylesheet	user_manage_modules.p...	(disk cache)	25 ms	
bootstrap.min.css	200	stylesheet	user_manage_modules.p...	(disk cache)	31 ms	
font-awesome.min.css	200	stylesheet	user_manage_modules.p...	(disk cache)	26 ms	
themify-icons.css	200	stylesheet	user_manage_modules.p...	(disk cache)	26 ms	
pe-icon-7-filled.css	200	stylesheet	user_manage_modules.p...	(disk cache)	26 ms	
flag-icon.min.css	200	stylesheet	user_manage_modules.p...	(disk cache)	26 ms	
cs-skin-elastic.css	200	stylesheet	user_manage_modules.p...	(disk cache)	25 ms	
style.css	200	stylesheet	user_manage_modules.p...	(disk cache)	24 ms	
css?family=Open+Sans:400,600,700,800	200	stylesheet	user_manage_modules.p...	(memory cac...	2 ms	
css?family=Open+Sans:400,600,700	200	stylesheet	style.css:-Infinity	(memory cac...	0 ms	
memvYaGs126MiZpBA-UvWbX2vVnXBbObj2OVT...	200	font	css?family=Open+Sans:...	(memory cac...	0 ms	

2.6.3.4 Logout Response Time



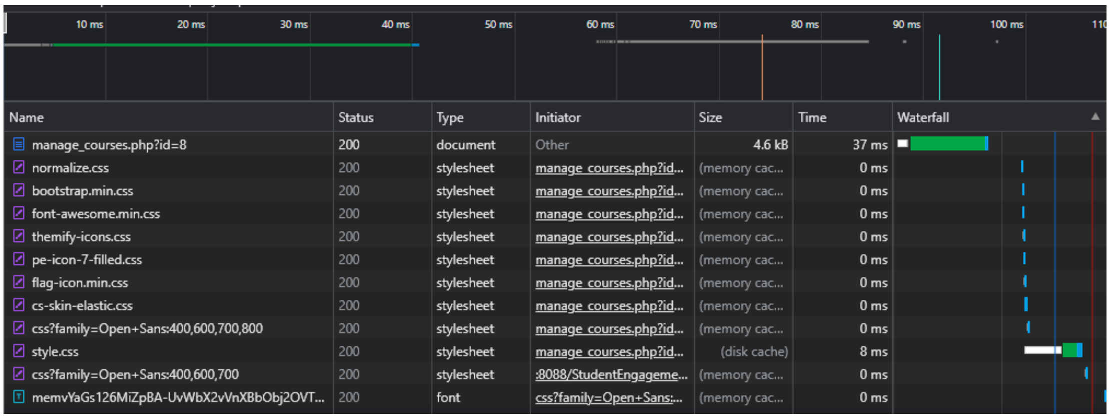
Name	Status	Type	Initiator	Size	Time	Waterfall
logout.php	302	document / R...	Other	275 B	5 ms	
login.php	200	document	logout.php	1.8 kB	33 ms	
bootstrap.min.css	307	stylesheet / R...	login.php	0 B	2 ms	
bootstrap.min.js	307	script / Redir...	login.php	0 B	4 ms	
jquery.min.js	307	script / Redir...	login.php	0 B	3 ms	
login.css	200	stylesheet	login.php	(disk cache)	3 ms	
bootstrap.min.css	200	stylesheet	bootstrap.min.css	(disk cache)	8 ms	
bootstrap.min.js	200	script	bootstrap.min.js	(disk cache)	5 ms	
jquery.min.js	200	script	jquery.min.js	(disk cache)	6 ms	
css?family=Lato:400,700	200	stylesheet	login.css:-Infinity	(disk cache)	4 ms	
background.jpg	200	jpeg	login.css	(disk cache)	7 ms	

2.6.3.5 Courses Response Time



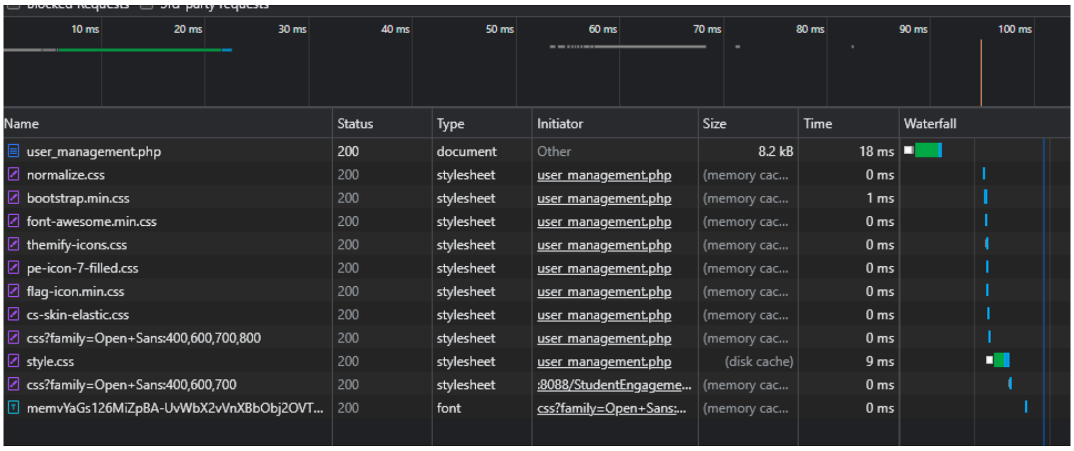
Name	Status	Type	Initiator	Size	Time	Waterfall
courses.php	200	document	Other	7.1 kB	11 ms	
normalize.css	200	stylesheet	courses.php:10	(memory cac...	0 ms	
bootstrap.min.css	200	stylesheet	courses.php:11	(memory cac...	0 ms	
font-awesome.min.css	200	stylesheet	courses.php:12	(memory cac...	0 ms	
themify-icons.css	200	stylesheet	courses.php	(memory cac...	0 ms	
pe-icon-7-filled.css	200	stylesheet	courses.php	(memory cac...	0 ms	
flag-icon.min.css	200	stylesheet	courses.php	(memory cac...	0 ms	
cs-skin-elastic.css	200	stylesheet	courses.php	(memory cac...	0 ms	
style.css	200	stylesheet	courses.php	(memory cac...	0 ms	
css?family=Open+Sans:400,600,700,800	200	stylesheet	courses.php	(memory cac...	0 ms	
css?family=Open+Sans:400,600,700	200	stylesheet	:8088/StudentEngageme...	(memory cac...	0 ms	
memvYaGs126MiZpBA-UvWbX2vVnXBbObj2OVT...	200	font	css?family=Open+Sans:...	(memory cac...	0 ms	

2.6.3.6 Create/Edit Courses Response Time



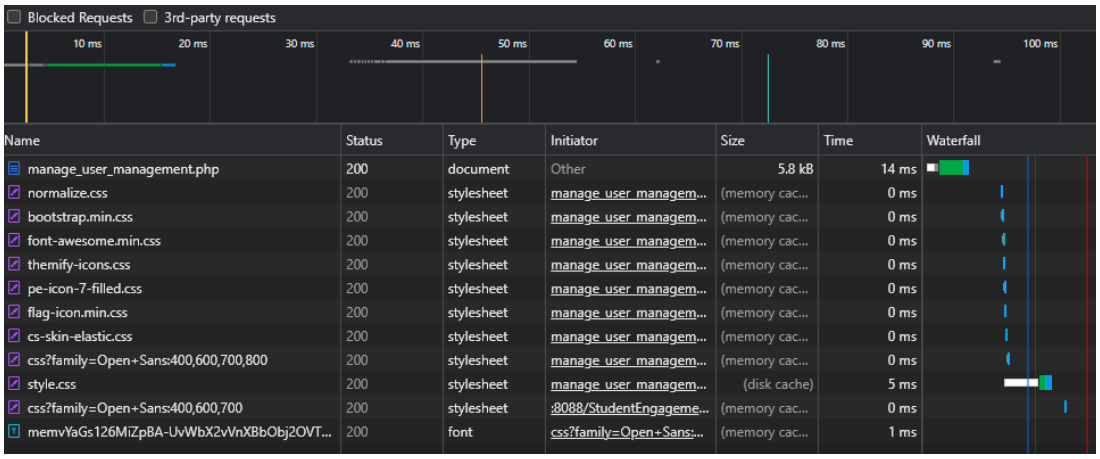
Name	Status	Type	Initiator	Size	Time	Waterfall
manage_courses.php?id=8	200	document	Other	4.6 kB	37 ms	
normalize.css	200	stylesheet	manage_courses.php?id...	(memory cac...	0 ms	
bootstrap.min.css	200	stylesheet	manage_courses.php?id...	(memory cac...	0 ms	
font-awesome.min.css	200	stylesheet	manage_courses.php?id...	(memory cac...	0 ms	
themify-icons.css	200	stylesheet	manage_courses.php?id...	(memory cac...	0 ms	
pe-icon-7-filled.css	200	stylesheet	manage_courses.php?id...	(memory cac...	0 ms	
flag-icon.min.css	200	stylesheet	manage_courses.php?id...	(memory cac...	0 ms	
cs-skin-elastic.css	200	stylesheet	manage_courses.php?id...	(memory cac...	0 ms	
css?family=Open+Sans:400,600,700,800	200	stylesheet	manage_courses.php?id...	(memory cac...	0 ms	
style.css	200	stylesheet	manage_courses.php?id...	(disk cache)	8 ms	
css?family=Open+Sans:400,600,700	200	stylesheet	:8088/StudentEngageme...	(memory cac...	0 ms	
memvYaGs126MiZpBA-UvWbX2vVnXBbObj2OVT...	200	font	css?family=Open+Sans:...	(memory cac...	0 ms	

2.6.3.7 Users Response Time



Name	Status	Type	Initiator	Size	Time	Waterfall
user_management.php	200	document	Other	8.2 kB	18 ms	
normalize.css	200	stylesheet	user_management.php	(memory cac...	0 ms	
bootstrap.min.css	200	stylesheet	user_management.php	(memory cac...	1 ms	
font-awesome.min.css	200	stylesheet	user_management.php	(memory cac...	0 ms	
themify-icons.css	200	stylesheet	user_management.php	(memory cac...	0 ms	
pe-icon-7-filled.css	200	stylesheet	user_management.php	(memory cac...	0 ms	
flag-icon.min.css	200	stylesheet	user_management.php	(memory cac...	0 ms	
cs-skin-elastic.css	200	stylesheet	user_management.php	(memory cac...	0 ms	
css?family=Open+Sans:400,600,700,800	200	stylesheet	user_management.php	(memory cac...	0 ms	
style.css	200	stylesheet	user_management.php	(disk cache)	9 ms	
css?family=Open+Sans:400,600,700	200	stylesheet	:8088/StudentEngageme...	(memory cac...	0 ms	
memvYaGs126MiZpBA-UvWbX2vVnXBbObj2OVT...	200	font	css?family=Open+Sans:...	(memory cac...	0 ms	

2.6.3.8 Create/Edit Users Response Time



Name	Status	Type	Initiator	Size	Time	Waterfall
manage_user_management.php	200	document	Other	5.8 kB	14 ms	
normalize.css	200	stylesheet	manage_user_managem...	(memory cac...	0 ms	
bootstrap.min.css	200	stylesheet	manage_user_managem...	(memory cac...	0 ms	
font-awesome.min.css	200	stylesheet	manage_user_managem...	(memory cac...	0 ms	
themify-icons.css	200	stylesheet	manage_user_managem...	(memory cac...	0 ms	
pe-icon-7-filled.css	200	stylesheet	manage_user_managem...	(memory cac...	0 ms	
flag-icon.min.css	200	stylesheet	manage_user_managem...	(memory cac...	0 ms	
cs-skin-elastic.css	200	stylesheet	manage_user_managem...	(memory cac...	0 ms	
css?family=Open+Sans:400,600,700,800	200	stylesheet	manage_user_managem...	(memory cac...	0 ms	
style.css	200	stylesheet	manage_user_managem...	(disk cache)	5 ms	
css?family=Open+Sans:400,600,700	200	stylesheet	:8088/StudentEngageme...	(memory cac...	0 ms	
memvYaGs126MiZpBA-UvWbX2vVnXBbObj2OVT...	200	font	css?family=Open+Sans:...	(memory cac...	1 ms	

3.0 Conclusions

Student Engagement Portal is an excellent one-of-a-kind idea. There are not many applications that offer an application like the Student Engagement Portal; however, none has been identified with the same functionality.

Whilst developing the Student Engagement Portal, I have learned a huge amount about various different technologies, especially PHP, MySQL, PHPMailer, HighCharts and PHPUnit, all which I had no prior experience with. I developed multiple skills such as communication, problem solving, management and programming throughout this project.

Looking at the Student Engagement Portal, I was able to develop all the objectives I set out to achieve in the project proposal. I was able to develop a web-based server, this is connected to a MySQL database. The application is able to accept mediated data and display the dashboard on a unique URL page. Students who are behind, can be emailed a generic support email.

When looking at the features, when a module is created and a file is uploaded, the application automatically saves all the columns to the database and displays data from all the columns. This is a disadvantage of this feature is that a user might only want to see the activity completion for one or two activity columns and not activity column.

To approach and fix this disadvantage, given more time, I would have done the following: When the file is uploaded, the information in the file is printed on the page with checkboxes above every column, from there users could select what columns that want to include in the dashboard and then only add the selected columns get inserted into the database.

To conclude, I consider that the Student Engagement Portal a successful project. I was able to complete all the functionality that I set out to do in my project proposal. The Student Engagement Portals final product is fit for purpose and can be used right away. The Student Engagement portal is ready to go to market with a beta version. . The little disadvantages that the student engagement portal has, are extra features that would enhance the user experience and can be resolved quickly with more development as long as time permits

4.0 Further Development or Research

Without the academic time constraint, I believe that I could had brought this application to the market as a service. With additional time, I think I could have refined this application more with additional features. There are a couple of extra features I would have added to the application to make it more complete.

Once piece of functionality I would have liked to further develop is when a user is creating or updating a module. If given more time, I would have liked to further expand on this feature.

To expand on this feature, given the time permitted, when a user clicks the submit button when creating or updating a module, the data within the file uploaded is printed under the submit button on that page. In the newly printed table, there's checkboxes at the top of each column. Users will be able to select what columns they want to add to the dashboard. Only the selected columns get inserted or updated to the database tables. This would allow for more precise dashboards and cleaning of the uploaded data.

Another feature I would have liked to add, given additional resources, is loops for the tables and charts where only the columns with data print to a table, not all the columns. This would just be beneficial to the back end of the application as it would allow for code reusability and make the applications response time quicker based on the code reusability.

5.0 References

Anon., 2016. *engage2serve.* [Online]
Available at: https://www.engage2serve.com/uk/blog/student-retention-strategies/
[Accessed 12 02 2022].

Anon., 2022. *Heroku.* [Online]
Available at: https://devcenter.heroku.com/articles/php-concurrency
[Accessed 20 04 2022].

Anon., n.d. *Alexc Web Develop.* [Online]
Available at: https://alexwebdevelop.com/monitor-script-memory-usage/#:~:text=But%20how%20can%20you%20check,memory_get_usage()%20and%20memory_get_peak_usage().
[Accessed 04 06 2022].

Anon., n.d. *MySQL.* [Online]
Available at: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html
[Accessed 06 11 2021].

Anon., n.d. *PHP.* [Online]
Available at: https://www.php.net/manual/en/intro-whatis.php
[Accessed 06 11 2021].

Anon., n.d. *PHPUnit.* [Online]
Available at: https://phpunit.de/
[Accessed 06 11 2021].

Anon., n.d. *Techopedia.* [Online]
Available at: https://www.techopedia.com/definition/7755/intellij-idea
[Accessed 06 11 2021].

Anon., n.d. *Usability.gov.* [Online]
Available at: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html
[Accessed 02 05 2022].

Anon., n.d. *W3.* [Online]
Available at: https://www.w3.org/standards/webdesign/htmlcss
[Accessed 06 11 2021].

Anon., n.d. *W3Schools.* [Online]
Available at: https://www.w3schools.com/php/php_sessions.asp
[Accessed 16 03 2022].

Anon., n.d. *XAMPP Apache + MariaDB + PHP + Perl.* [Online]
Available at: https://www.apachefriends.org/index.html
[Accessed 06 11 2021].

6.0 Appendices

6.1 Project Proposal

National College of Ireland

Project Proposal

Student Engagement Portal

05/11/2021

Bachelor of Science (Honours) in Computing

Software Development

2021/2022

Tegan Jennings

X18303941

X18303941@student.ncirl.ie

Contents

1. Objectives

The objective of my project is to allow for student engagement to be monitored on learning platforms. My application will be a web based server that is independent of its data source, it will take in mediated data and display it as a dashboard. I have a lot of aims and objectives for my project, the main objectives are as follows:

- Web based server developed using PHP
- Application connected to a database
- Application accepts mediated data
- Application displays mediated data in a dashboard
- Dashboard can be filtered
- Students who are behind can be emailed

Each one of my objectives for my project is extremely important. I need to complete one objective in order to complete the next object.

2. Background

This project idea steamed from my current role within the National College of Ireland as Learning Support. When the pandemic happened, education institutions around the world closed down and went online for an indefinite amount of time. There was no one stop shop application that would monitor student engagement online. I decided I should develop an application that allows for student engagement to be tracked, regardless of the data source.

In order to meet my objectives, I will be creating a web based server, that users can log into and then upload a file to generate a dashboard with charts and the ability to email students who are behind.

3. State of the Art

After extensive research I have not found an application that will track student engagement and display it as a dashboard, the way my application will. I have found some plugins for the platform Moodle, however my application will be independent of its data source.

There are applications out there that are school administration software, like Teach 'n go. However teach and go offers administration solutions to schools like a dashboard for student attendance and students information, however my application will be offering student engagement solutions. This is the closest application I have found at the moment to my application.

There is many Moodle plugins that can display student engagement, however my application will be independent of its data source which makes my application different to the Moodle plugins that rely on Moodle for the information.

4. Technical Approach

I will be using an agile methodology to develop my project. I will be developing a Gantt chart that allows me to develop in two week sprints. By completing two week sprints I will be able to manage my workflow well and complete milestones regularly. The Gantt chart will allow me to be able to view all the tasks I have with each submission and delegate time efficiently to each task.

To identify the requirements of my project, I firstly took note if all the activities I need to complete in order to track student engagement manually. I then created my first set of requirements from this analysis. I will be running a focus group in January 2022, in order to finalise my requirements and make any necessary changes.

The IDE I will be using is Intellij IDEA. I have used Intellij IDEA previously so I already have knowledge of how to use this IDE, so no research was needed on my IDE.

I will be using MongoDB as the database for my project. I will be researching and investigating how MongoDB works and how I will be able to save file uploads to the database. I will be using MongoDB documentation to help me set up the database and how to connect my application to the database

5. Technical Details

The IDE I will be using for this project will be Intellij. This project will be stored on GitHub to allow for version control.

At the moment I will be using the following languages to create my application:

- PHP
- HTML
- CSS
- JavaScript
- PHPUnit
- MongoDB

I will be using PHP to develop my web based server as PHP can be used to create interactive websites which is what my application will be. PHP is a server side language that I can embed into HTML

I will be using HTML, CSS and JavaScript to create a user interface and User Experience and make sure the application looks nice. The HTML will be the backbone of the application and I will have PHP embedded into the HTML for the functionality.

I will also be working with the PHPUnit to perform unit testing on my application. PHPUnit is a unit testing framework for the PHP programming language.

I will be using a MongoDB database to store user's information and store the files uploaded so that the dashboards can be generated.

6. Special Resources Required

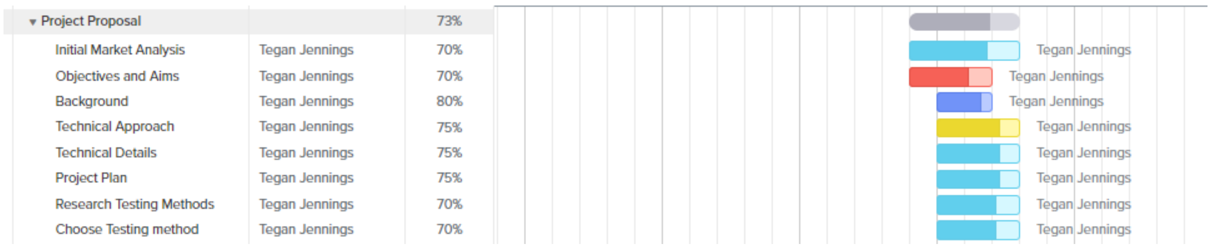
No special resources will be required for this project.

7. Project Plan

I will be creating a Gantt chart to track the activities I have to complete in order to get this project done by the deadlines. At the moment this is my Gantt chart, however it will grow and develop over time.

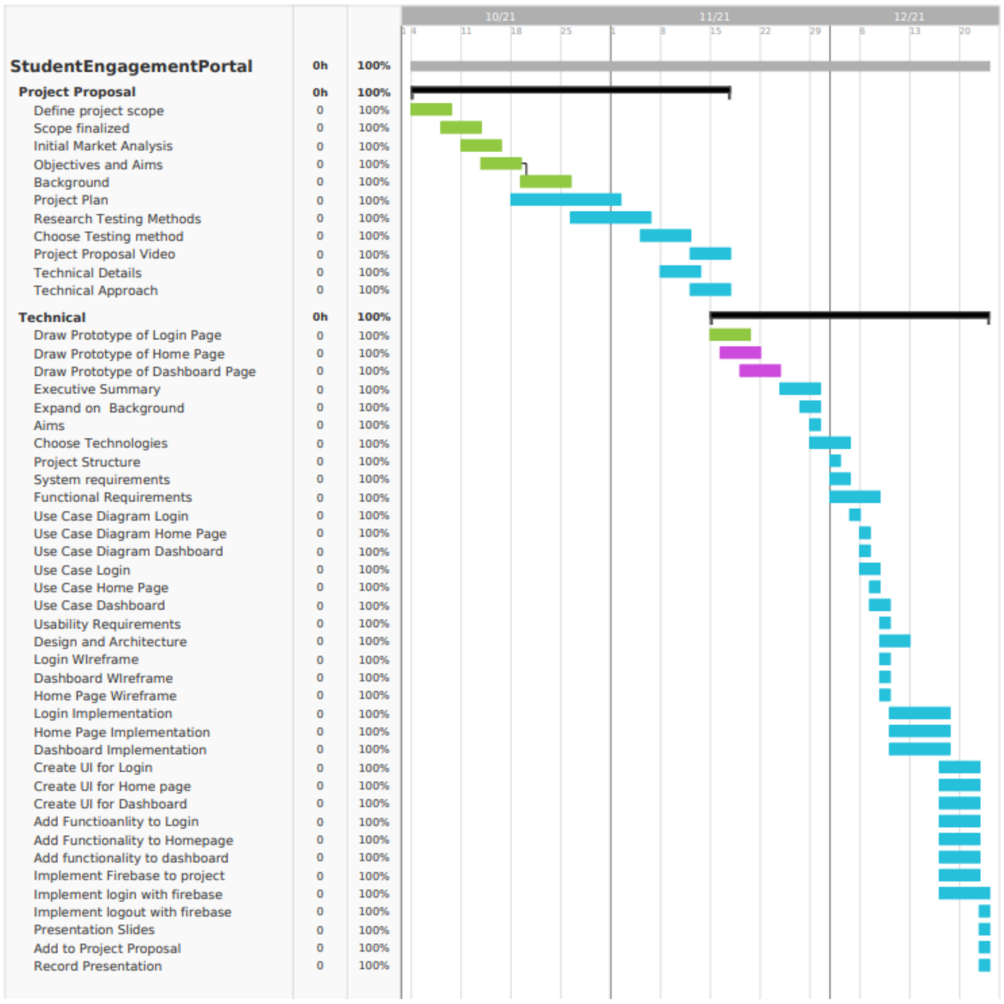
Currently my Gantt chart has the task for all of my deliverables up until the midpoint upload. I will be adding the rest of my tasks in January before semester 2 begins, so that I can allocate my time accordingly.

Gantt chart V1



▾ Project Proposal		73%
Initial Market Analysis	Tegan Jennings	70%
Objectives and Aims	Tegan Jennings	70%
Background	Tegan Jennings	80%
Technical Approach	Tegan Jennings	75%
Technical Details	Tegan Jennings	75%
Project Plan	Tegan Jennings	75%
Research Testing Methods	Tegan Jennings	70%
Choose Testing method	Tegan Jennings	70%

Gantt chart V2



StudentEngagementPortal	**0h**	**100%**
Project Proposal	**0h**	**100%**
Define project scope	0	100%
Scope finalized	0	100%
Initial Market Analysis	0	100%
Objectives and Aims	0	100%
Background	0	100%
Project Plan	0	100%
Research Testing Methods	0	100%
Choose Testing method	0	100%
Project Proposal Video	0	100%
Technical Details	0	100%
Technical Approach	0	100%
Technical	**0h**	**100%**
Draw Prototype of Login Page	0	100%
Draw Prototype of Home Page	0	100%
Draw Prototype of Dashboard Page	0	100%
Executive Summary	0	100%
Expand on Background	0	100%
Aims	0	100%
Choose Technologies	0	100%
Project Structure	0	100%
System requirements	0	100%
Functional Requirements	0	100%
Use Case Diagram Login	0	100%
Use Case Diagram Home Page	0	100%
Use Case Diagram Dashboard	0	100%
Use Case Login	0	100%
Use Case Home Page	0	100%
Use Case Dashboard	0	100%
Usability Requirements	0	100%
Design and Architecture	0	100%
Login WIreframe	0	100%
Dashboard WIreframe	0	100%
Home Page Wireframe	0	100%
Login Implementation	0	100%
Home Page Implementation	0	100%
Dashboard Implementation	0	100%
Create UI for Login	0	100%
Create UI for Home page	0	100%
Create UI for Dashboard	0	100%
Add Functioanlity to Login	0	100%
Add Functionality to Homepage	0	100%
Add functionality to dashboard	0	100%
Implement Firebase to project	0	100%
Implement login with firebase	0	100%
Implement logout with firebase	0	100%
Presentation Slides	0	100%
Add to Project Proposal	0	100%
Record Presentation	0	100%

8. Testing

The method of testing I have chosen for my project is unit testing. Unit testing relies on mock objects being created to test sections of code that are not yet part of a complete application. Mock objects fill in for the missing parts of the program.

To test my application, I will be using PHPUnit in order to create and run unit tests for the PHP code in my project. I will be able to test the functionality of the application.

Unit tests are important as it will check to see if the application functions as I continue to add to my project. For my application this will be very important as each part of my applications functionality relies on a previously functionality to work. For that reason, the unit tests will be extremely important for my application.

I will be creating dummy data of student's engagement also to check my dashboard functionality. I will be doing manual systems testing to check that all the functionality on the dashboard and the filtering works.

I will be also manually testing the email functionality and send out engagement emails to fake email accounts I create to see if the students get personalized emails.

6.2 Reflective Journals

Supervision & Reflection October Document

Student Name	Tegan Jennings
Student Number	X18303941
Course	BSHCSD4

Month: October

What? Reflect on what has happened in your project this month?

This month I came up with my project idea and I uploaded my project pitch. At the beginning of October, I had a rough idea of what my projects idea was going to be and how I might approach it. However, I needed to expand on my idea and the visibility of the project.

My project idea is a web-based server that takes in mediated data and displays a dashboard with that data. The server will also have user permissions. My project will be centred on student engagement in online courses. My project will display the mediated data in a dashboard for stakeholders to see student's engagement. My project will also allow for students to be emailed if they don't engage in the class.

I recorded a project pitch video this month. My project pitch video outlined what I have said above about my project idea. My project pitch video was reviewed by 2 people, and I am thankful that my project idea has been approved.

So What? Consider what that meant for your project progress. What were your successes? What challenges still remain?

So now that my project idea has been approved, I can begin my project proposal and start choosing what technologies to use for my project. This also means that I can begin working on my project proposal and other documentation that will accompany my project.

One challenge that I have at the moment, is that I don't have any PHP knowledge. This is a problem as my project will be a PHP web-based server so I will need to know PHP in order to start development. On top of that, I will need to decide what technical approach to take and develop my project plan more.

Now What? What can you do to address outstanding challenges? This month to solve the current problem with lack of PHP knowledge, I will be completing some YouTube tutorials and some Code Academy courses so that when I start developing my project, I have some PHP knowledge to get me started. I also will be researching more in-depth and developing this project more. I will be creating a Gantt chart this month so that I can continuously monitor my progress and breakdown my tasks, activities and milestones.	
Student Signature: Tegan Jennings	Tegan Jennings

Supervision & Reflection November Document

Student Name	Tegan Jennings
Student Number	X18303941
Course	BSHCSD4

Month: November

What? Reflect on what has happened in your project this month? For the majority of this month, I focused on my project's requirements. I have spent a lot of time this month investigating different ways to get information from people willing to participant in an interview with me. After extensive research and investigating, I decided to run a focus group with a select few individuals that might see some benefit from my project. I had to clearly choose what prompts I would give to participants of the focus group so that I could efficiently get the information I needed with the help of all the participants. I also needed to create a permission form for participants to sign in order to agree to be a part of the focus group. I also completed a lot of PHP tutorials over the last month, in order to understand PHP fully, so that I can code my project efficient. I didn't have any PHP knowledge before, so I prioritise learning the language to the best of my ability so that I was able to develop my project.
So What? Consider what that meant for your project progress. What were your successes? What challenges still remain? Now that I have most of the ethics application completed, I can now finish my application and submit it. Once I have my ethics application submitted, I will run my focus group. After researching focus groups, I believe I will be able to efficiently run a focus group and get the information I need from it. After my focus group, I will be able to gather more project requirements and begin to complete more documentation and so now that I know how I will be running my focus group to gather more of my project requirements, I need to complete my ethics application form fully and submit it to the ethics committee. Now that I have a lot more knowledge of PHP, I will be starting the development of my project next month. I will start with creating the web-based server with minimal functionality to begin with, then build off that.

Now What? What can you do to address outstanding challenges? My current challenge is managing my time. I am working against a lot of different deadlines, so I am sticking to my Gantt chart that I have created for this project so that I can meet my projects milestones and deliverables. Other than that, I don't have any other challenges I am facing at the present.	
Student Signature: Tegan Jennings	Tegan Jennings

Supervision & Reflection December Document

Student Name	Tegan Jennings
Student Number	X18303941
Course	BSHCSD4

Month: December

What? Reflect on what has happened in your project this month?

This month I focused on coding my project prototype, now that I had clear requirements outlined for my project. I was able to create the barebones of my project and how the layout will look when I am finished. The project now resembles how the functionality will work, I have created the basic functionality for the project and the UI for it. I also started the technical report documentation and filled in what I could.

So What? Consider what that meant for your project progress. What were your successes? What challenges still remain?

Now that I have started the development of my project, I will now need to research APIs that might be able to help me with the generation of the dashboard pages and charts.

I need to now expand on the functionality I have now coded in order to get all the functionality working to 100% as right now the functionality is very basic. I am pleased to have the application started and have a clear image in my head of where the project is going and what I need to do to complete it.

I will be running a focus group too. This focus group will be run to redefine my project requirements and make sure I'm not missing anything.

Now What? What can you do to address outstanding challenges?

My current challenge is my database, after connecting my application to the database, my database worked for the login initially but now my database won't work. I need to investigate why that happened and what change caused my database to stop interacting with my application.

Another challenge I am having is getting files to save to the database. I need the files that are uploaded to save to the database so that the application can generate a dashboard page. I will be reading the MongoDB documentation and following tutorials in order to address this challenge.

Student Signature: Tegan Jennings	Tegan Jennings

Supervision & Reflection January Document

Student Name	Tegan Jennings
Student Number	X18303941
Course	BSHCSD4

Month: January

What? Reflect on what has happened in your project this month? This month I focused on my applications database. After failing to use Firebase as my database. I changed to MongoDB. Because of this I spent most of this month reading documentations on MongoDB and implementing it into my project.
So What? Consider what that meant for your project progress. What were your successes? What challenges still remain? Now that I understand MongoDB and have it almost completely implemented, I can now work on the bigger functionality pieces of my project.
Now What? What can you do to address outstanding challenges? The next challenge I will face is starting the development of the first functionality piece which is uploading an excel to the database.

Student Signature: Tegan Jennings	Tegan Jennings

Supervision & Reflection February Document

Student Name	Tegan Jennings
Student Number	X18303941
Course	BSHCSD4

Month: February

What? Reflect on what has happened in your project this month? This month, like last week I continued to focus on the database for my application. I have been trying to implement the database into the application correctly so that when users upload an excel, the file will be saved in the database.
So What? Consider what that meant for your project progress. What were your successes? What challenges still remain? I have my database implementation almost completed. Once it is completed I will be able to work on the next functionality of the project. However, until the database is implemented fully and functioning, I cannot work on the other functionality of my application.
Now What? What can you do to address outstanding challenges? My next challenge will be a huge challenge. Which is getting the data from the excel in the database and create the functionality for the application to automatically generate a web page with the student retention being displayed using charts and graphs.

Student Signature: Tegan Jennings	Tegan Jennings

Supervision & Reflection March Document

Student Name	Tegan Jennings
Student Number	X18303941
Course	BSHCSD4

Month: March

What? Reflect on what has happened in your project this month? This month I worked on my documentation more. After meeting with my supervisor, I focused most of my energy on updating parts of my documentation that needed more information. I also took all the comments my supervisor made and implemented them into my documentation.
So What? Consider what that meant for your project progress. What were your successes? What challenges still remain? Next month I will be focusing on my functionality of my project. I will need to work hard to overcome the challenges that I will face when implementing my functionality.
Now What? What can you do to address outstanding challenges? My next challenge will be to create a unique URL when an excel is uploaded to the database and store that URL for a specified amount of time, for my project it will be a week. I will be watching tutorials and creating the functionality with trial and error.

Student Signature: Tegan Jennings	Tegan Jennings

Supervision & Reflection April Document

Student Name	Tegan Jennings
Student Number	X18303941
Course	BSHCSD4

Month: April

What? Reflect on what has happened in your project this month?

This month I worked on my functionality. I was able to get csv files to save to database and print the information from the database. I also was able to get the dashboard started and one of the charts is working with the information in the database.

So What? Consider what that meant for your project progress. What were your successes? What challenges still remain?

Now that I have some of the functionality done, I need to add the rest of the functionality and then finish the functionality pieces and build on my database. I need to add loops and remove anything I've hardcoded from the database. I need to clean up the code and then start unit testing.

Now What? What can you do to address outstanding challenges?

My next challenge will be to create a way to create new tables in the database when a csv file is uploaded. This has been very challenging to do as I need to create a new table, then add the information from the csv file to that table and then grab the information from the table to add to the dashboard. I also need to add loops to count columns in the database and then only show the number of columns in the database into a table.

Student Signature: Tegan Jennings	Tegan Jennings

6.3 Other materials used

6.3.1 Reminder/Support Generic Email

Dear <Student_Name>,

According to your Moodle records, there are items outstanding to be completed for <Module>. Please ensure you address these items before class.

Before class, please ensure you have the following completed: <Outstanding_Items>

Please let me know if I can be of assistance in any way; I am happy to answer any questions you may have.

Please note that each week of live classes relies directly on the Moodle content labelled for that week, and therefore it is easy to get left behind in class if you do not keep up on Moodle.

If you need help catching up, please reach out and speak to Computing Support as soon as possible. Their email is computingsupport@ncirl.ie. They will be happy to help you catch up.

If you feel you have received this message in error, please let me know.

Kind regards,

Tegan Jennings
Online Learning Support

6.3.2 Testing Consent Form

Student Engagement Portal

Testing Consent Form

- I ____________________________________ agree to voluntarily participant in the testing of the Student Engagement Portal.
- I am aware of the Student Engagement Portal and its aims and objectives.
- I understand that no information that could be used to identify me will be collected or stored, with the exception of my signature below.
- I understand that I can withdraw from testing at any time by request.
- I understand that I am not obliged to test the student engagement portal.
- I understand that I am free to contact the testing host to seek further information or clarification on anything asked or covered during or after the testing has concluded.

Signature of testing participant

I understand what is involved in this testing and I agree to participate in the study.

_____________________	________
Signature of participant	Date

Signature of researcher

I believe the participant is giving informed consent to participate in this study.

______________________	________
Signature of researcher	Date

6.3.3 Testing Results

6.3.3.1 Trunk Test Results

Participant 1

Task Name	Task Goal	Time Taken (seconds)	Expected Behaviour	Actual Behaviour
What is this application?	The participant must be able to recall the application name and purpose.	15	The participant identifies the application name and understands its purpose.	The participant identifies the application name as Student Engagement Portal.
Can you see any major features?	The participant must identify the key features offered.	8	The participant clearly identifies key functionality.	The participant clearly identifies key functionality.
Where can you create a module?	The participant must be able to locate where to create a module.	5	The participant locates where to create a module in the application and creates one.	Participant located where to create a module.
Where can you delete a module?	The participant must be able to locate where to delete a module.	5	The participant locates where to delete a module in the application and deletes one.	Participant located where to delete a module.
Where can you view a module dashboard?	The participant must be able to locate where the module dashboards are.	5	The participant locates a module dashboard and views it.	The participant located a module dashboard.
Where can you email students?	The participant must be able to locate where to email students.	20	The participant locates the email students' section and sends an email.	Participant viewed a module and located email section.
Where do you logout?	The participant must locate the logout functionality.	6	The participant locates the logout option located on the left side navigation.	Participant located the logout.

Participant 2

Task Name	Task Goal	Time Taken (seconds)	Expected Behaviour	Actual Behaviour
What is this application?	The participant must be able to recall the application name and purpose.	25	The participant identifies the application name as Student Engagement Portal.	Participant identified the applications name.
Can you see any major features?	The participant must identify the key features offered.	13	The participant clearly identifies key functionality.	Participant identified most of the key functionality
Where can you create a module?	The participant must be able to locate where to create a module.	4	The participant locates where to create a module in the application and creates one.	Participant located where to create a module.
Where can you delete a module?	The participant must be able to locate where to delete a module.	7	The participant locates where to delete a module in the application and deletes one.	Participant located where to delete a module.
Where can you view a module dashboard?	The participant must be able to locate where the module dashboards are.	8	The participant locates a module dashboard and views it.	The participant located a module dashboard.
Where can you email students?	The participant must be able to locate where to email students.	9	The participant locates the email students' section and sends an email.	Participant viewed a module and located email section.
Where do you logout?	The participant must locate the logout functionality.	6	The participant locates the logout option located on the left side navigation.	Participant located the logout.

<u>Participant 3</u>

Task Name	Task Goal	Time Taken (seconds)	Expected Behaviour	Actual Behaviour
What is this application?	The participant must be able to recall the application name and purpose.	7	The participant identifies the application name as Student Engagement Portal.	Participant identified applications name.
Can you see any major features?	The participant must identify the key features offered.	6	The participant clearly identifies key functionality.	Participant identified the key functionality
Where can you create a module?	The participant must be able to locate where to create a module.	6	The participant locates where to create a module in the application and creates one.	Participant located where to create a module.
Where can you delete a module?	The participant must be able to locate where to delete a module.	4	The participant locates where to delete a module in the application and deletes one.	Participant located where to delete a module.
Where can you view a module dashboard?	The participant must be able to locate where the module dashboards are.	6	The participant locates a module dashboard and views it.	The participant located a module dashboard.
Where can you email students?	The participant must be able to locate where to email students.	8	The participant locates the email students' section and sends an email.	Participant viewed a module and located email section.
Where do you logout?	The participant must locate the logout functionality.	2	The participant locates the logout option located on the left side navigation.	Participant located the logout.

Participant 4

Task Name	Task Goal	Time Taken (seconds)	Expected Behaviour	Actual Behaviour
What is this application?	The participant must be able to recall the application name and purpose.	10	The participant identifies the application name as Student Engagement Portal.	Participant identified home page name.
Can you see any major features?	The participant must identify the key features offered.	5	The participant clearly identifies key functionality.	Participant identified the key functionality
Where can you create a module?	The participant must be able to locate where to create a module.	6	The participant locates where to create a module in the application and creates one.	Participant located where to create a module.
Where can you delete a module?	The participant must be able to locate where to delete a module.	8	The participant locates where to delete a module in the application and deletes one.	Participant located where to delete a module.
Where can you view a module dashboard?	The participant must be able to locate where the module dashboards are.	8	The participant locates a module dashboard and views it.	The participant located a module dashboard.
Where can you email students?	The participant must be able to locate where to email students.	55	The participant locates the email students' section and sends an email.	Participant edited a module and then viewed a module and located email section.
Where do you logout?	The participant must locate the logout functionality.	2	The participant locates the logout option located on the left side navigation.	Participant located the logout.

Participant 5

Task Name	Task Goal	Time Taken (seconds)	Expected Behaviour	Actual Behaviour
What is this application?	The participant must be able to recall the application name and purpose.	9	The participant identifies the application name as Student Engagement Portal.	Participant identified the application name and understood its purpose
Can you see any major features?	The participant must identify the key features offered.	6	The participant clearly identifies key functionality.	Participant identified some of the key functionality
Where can you create a module?	The participant must be able to locate where to create a module.	6	The participant locates where to create a module in the application and creates one.	Participant located where to create a module.
Where can you delete a module?	The participant must be able to locate where to delete a module.	6	The participant locates where to delete a module in the application and deletes one.	Participant located where to delete a module.
Where can you view a module dashboard?	The participant must be able to locate where the module dashboards are.	6	The participant locates a module dashboard and views it.	The participant located a module dashboard.
Where can you email students?	The participant must be able to locate where to email students.	38	The participant locates the email students' section and sends an email.	Participant views a module and located email section.
Where do you logout?	The participant must locate the logout functionality.	3	The participant locates the logout option located on the left side navigation.	Participant located the logout.

6.3.3.2 Think Aloud Test Results

Participant 1

Task Name	Task Goal	Time Taken (seconds)	Expected Behaviour	Actual Behaviour
Log into application	The participant logs into their account and is shown the correct home page whether they are a user or admin.	5	The participant successfully logs into the application.	Participant logged into application
Create a Module	The participant clicks the module section of the side navigation and then clicks add module, fills I the form and submits.	20	The participant successfully creates a new module.	Participant created new module but didn't upload a csv file.
View a module dashboard	The participant must click on the dashboard section in the side navigation then choose a module dashboard to view and view it.	9	The participant views a module dashboard from the dashboard section.	Participant viewed the dashboard.

Participant 2

Task Name	Task Goal	Time Taken (seconds)	Expected Behaviour	Actual Behaviour
Log into application	The participant logs into their account and is shown the correct home page whether they are a user or admin.	11	The participant successfully logs into the application.	Participant logged into application
Create a Module	The participant clicks the module section of the side navigation and then clicks add module, fills I the form and submits.	17	The participant successfully creates a new module.	Participant created new module.
View a module dashboard	The participant must click on the dashboard section in the side navigation then choose a module dashboard to view and view it.	4	The participant views a module dashboard from the dashboard section.	Participant viewed the dashboard.

Participant 3

Task Name	Task Goal	Time Taken (seconds)	Expected Behaviour	Actual Behaviour
Log into application	The participant logs into their account and is shown the correct home page whether they are a user or admin.	40	The participant successfully logs into the application.	Participant logged into application after 1 unsuccessfully try.
Create a Module	The participant clicks the module section of the side navigation and then clicks add module, fills I the form and submits.	150	The participant successfully creates a new module.	Participant created new module but tried to upload a png. Then uploaded a csv
View a module dashboard	The participant must click on the dashboard section in the side navigation then choose a module dashboard to view and view it.	15	The participant views a module dashboard from the dashboard section.	Participant viewed the dashboard.

Participant 4

Task Name	Task Goal	Time Taken (seconds)	Expected Behaviour	Actual Behaviour
Log into application	The participant logs into their account and is shown the correct home page whether they are a user or admin.	5	The participant successfully logs into the application.	Participant logged into application successfully
Create a Module	The participant clicks the module section of the side navigation and then clicks add module, fills I the form and submits.	12	The participant successfully creates a new module.	Participant created new module and uploaded a file.
View a module dashboard	The participant must click on the dashboard section in the side navigation then choose a module dashboard to view and view it.	4	The participant views a module dashboard from the dashboard section.	Participant viewed the dashboard.

<u>Participant 5</u>

Task Name	Task Goal	Time Taken (seconds)	Expected Behaviour	Actual Behaviour
Log into application	The participant logs into their account and is shown the correct home page whether they are a user or admin.	70	The participant successfully logs into the application.	Participant reset password and then logged in successfully.
Create a Module	The participant clicks the module section of the side navigation and then clicks add module, fills I the form and submits.	22	The participant creates a new module from the modules section.	Participant created a new module and uploaded a file.
View a module dashboard	The participant must click on the dashboard section in the side navigation then choose a module dashboard to view and view it.	33	The participant views a module dashboard from the dashboard section.	Participant viewed dashboard and emailed a student

6.3.3.3 5 Second Test Results

Participant 1

Task Name	Task Goal	Actual Behaviour
What is the applications name?	The participant recalls application as Student Engagement Portal.	Participant recalled module name.
What is the application about and who is this project target market?	The participant recalls the application is about college engagement for modules.	Participant recalled the application is about module and student information and said it was for college admins.
What functionality does this application offer?	The participant recalls the module and dashboard functionality.	Participant recalled most functionality

Participant 2

Task Name	Task Goal	Actual Behaviour
What is the applications name?	The participant recalls application as Student Engagement Portal.	Participant recalled application name.
What is the application about and who is this project target market?	The participant recalls the application is about college engagement for modules.	Participant understood what this application is for.
What functionality does this application offer?	The participant recalls the module and dashboard functionality.	Participant understood all functionality

Participant 3

Task Name	Task Goal	Actual Behaviour
What is the applications name?	The participant recalls application as Student Engagement Portal.	Participant recalled application name.
What is the application about and who is this project target market?	The participant recalls the application is about college engagement for modules.	Participant recalled the application being for a college
What functionality does this application offer?	The participant recalls the module and dashboard functionality.	Participant understood all functionality

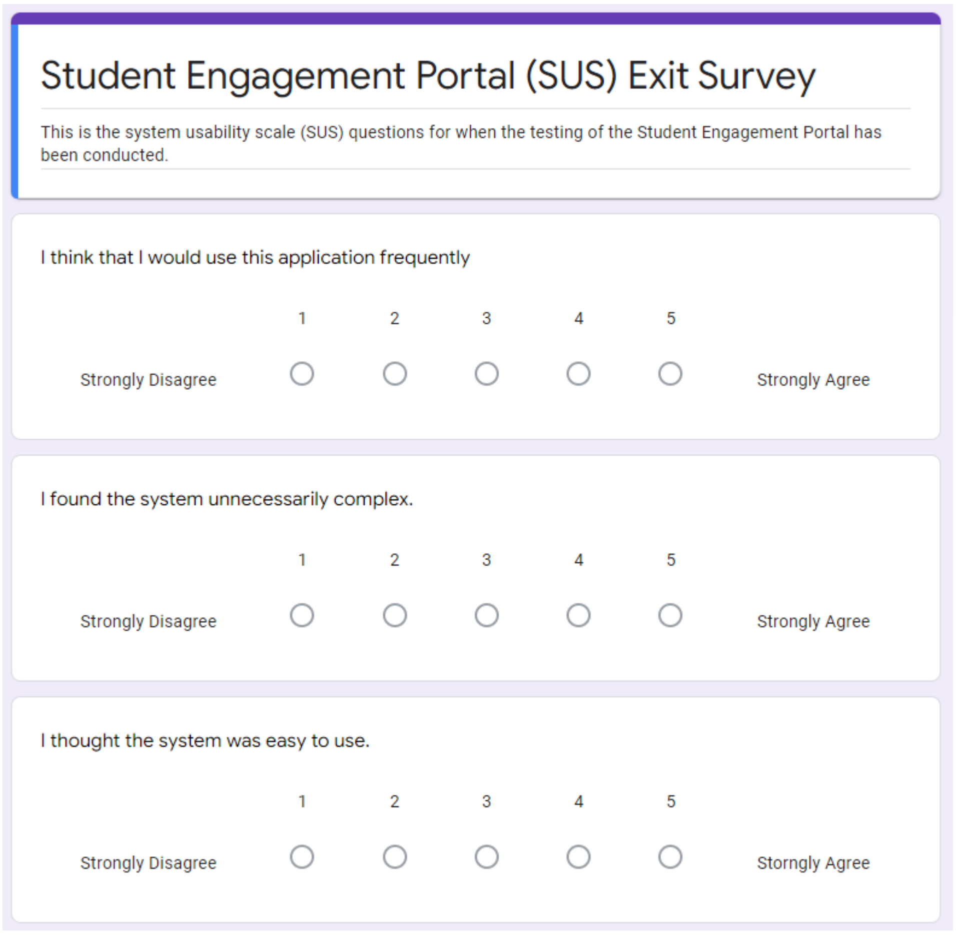
Participant 4

Task Name	Task Goal	Actual Behaviour
What is the applications name?	The participant recalls application as Student Engagement Portal.	Participant called application by Module section.
What is the application about and who is this project target market?	The participant recalls the application is about college engagement for modules.	Participant recalled the application managing module information
What functionality does this application offer?	The participant recalls the module and dashboard functionality.	Participant understood some functionality

Participant 5

Task Name	Task Goal	Actual Behaviour
What is the applications name?	The participant recalls application as Student Engagement Portal.	Participant called application by Module section.
What is the application about and who is this project target market?	The participant recalls the application is about college engagement for modules.	Participant recalled the application managing dashboards with student information on them
What functionality does this application offer?	The participant recalls the module and dashboard functionality.	Participant understood most functionality

6.3.3.4 System Usability Scale Exit Survey



Student Engagement Portal (SUS) Exit Survey

This is the system usability scale (SUS) questions for when the testing of the Student Engagement Portal has been conducted.

I think that I would use this application frequently

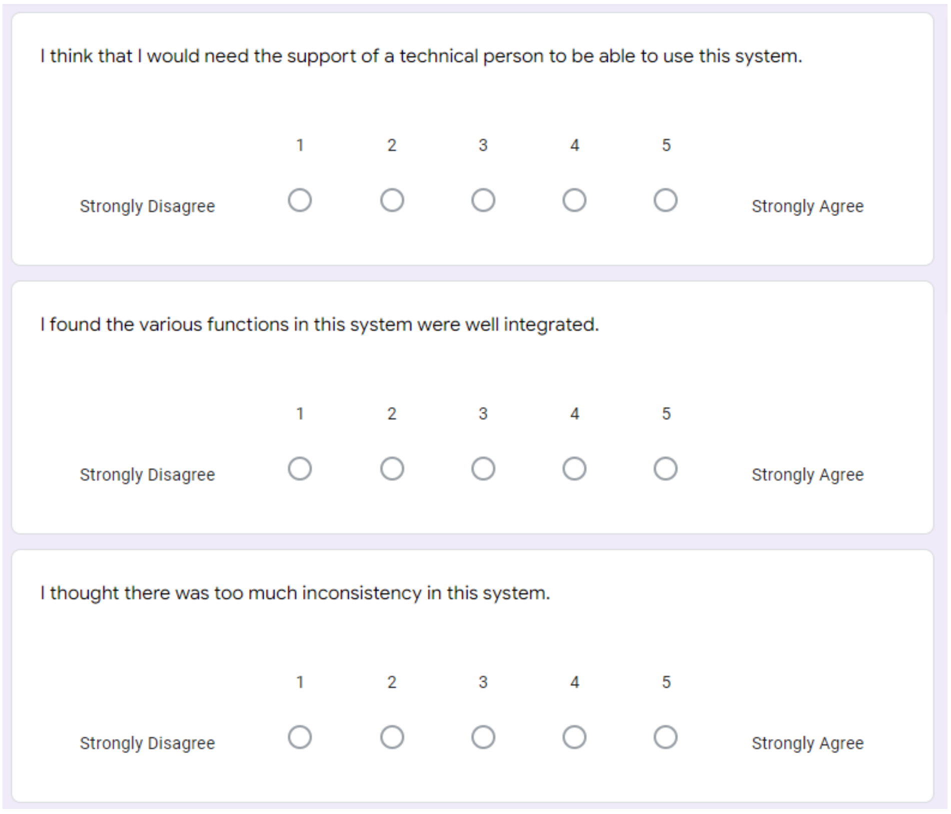
	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I found the system unnecessarily complex.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I thought the system was easy to use.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Stornly Agree



I think that I would need the support of a technical person to be able to use this system.

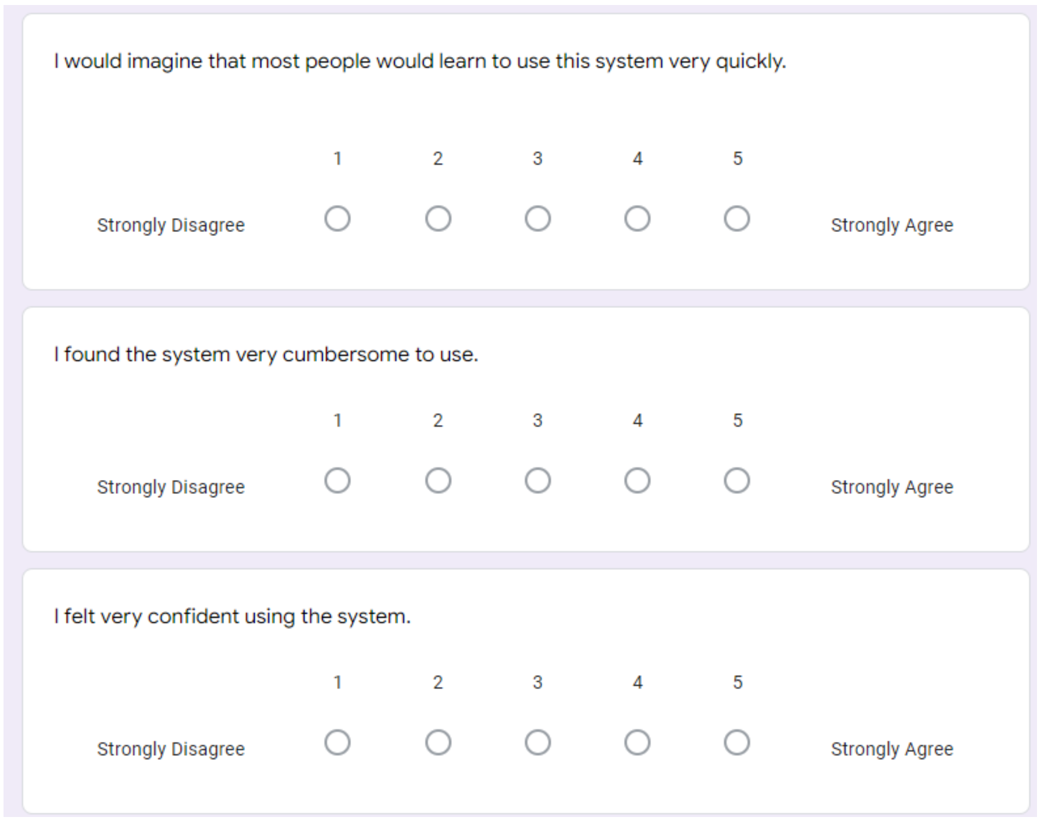
	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I found the various functions in this system were well integrated.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I thought there was too much inconsistency in this system.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree



I would imagine that most people would learn to use this system very quickly.

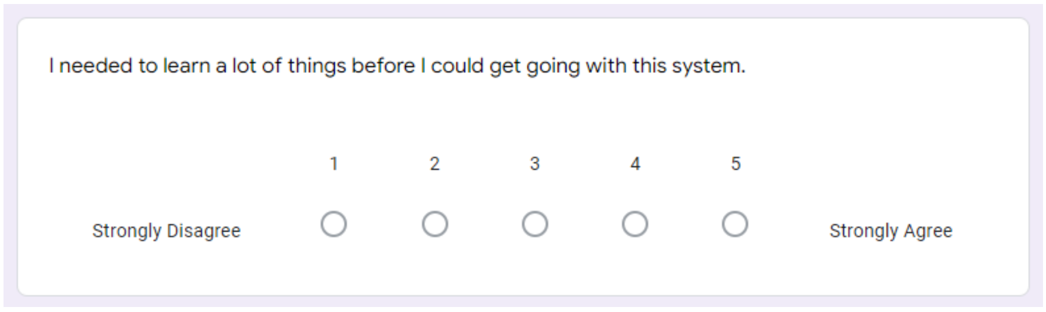
	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I found the system very cumbersome to use.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I felt very confident using the system.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree



I needed to learn a lot of things before I could get going with this system.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree