

MyStudentHelper



Robert Foster, X17140404

National College of Ireland

BSc (Hons) in Computing

Specialisation : Software Development

Contents

Executive Summary	3
Introduction.....	3
Background	4
Aims	4
Technology	5
Structure.....	5
Requirement Specification.....	6
Moodle + MyStudentHelper Use Case Diagram	6
Functional Requirements	7
Requirement 1: User Registration.....	7
Flow Description.....	7
Requirement 2: User Login.....	8
Flow Description.....	9
Requirement 3: Setting up tickets	10
Flow Description.....	11
Requirement 4: Setting up groups.....	12
Flow Description.....	13
Non-Functional Requirements	15
Hardware Requirements.....	15
Software Requirements	15
Architecture and Design	16
Moodle core	16
Implementation	18
Testing	24
Graphical User Interface (GUI)	26
Conclusion	29
Further Development or Research	30
Bibliography	30
Project Plan	30
Reflective Journals.....	31
Objectives	38
Background.....	39

Technical Approach	40
Project Plan	41
Gantt chart.....	41
Technical Details	42
Evaluation	42

Executive Summary

The purpose of this report is to outline the benefits of MyStudentHelper, and to show how all users will interact with the Moodle system to gain access to the plug-in (MyStudentHelper).

This report will show my experience and research that has gone into the idea. MyStudentHelper will maximise grades and teach students how to break down large projects into smaller parts. I hope that it lowers anxiety levels in students especially during this year where things have been difficult enough. The report shows the requirements that will be needed to set up Moodle for development purposes. This report will also show how I implemented code, ran tests and the overall architecture of Moodle.

Introduction

MyStudentHelper will be a plugin for the Moodle learning platform that will provide a knowledgebase where each student can access and store information such as files, save notes and create tasks. A task would be created when a student is assigned a project from their respected lecturer, the student can then go to the knowledgebase block that will be displayed on the Moodle dashboard, create a ticket which would in turn break down the project into more manageable parts.

The idea behind the group section of MyStudentHelper is that most students especially in their first year will not have the opportunity to make friends and form groups to work on projects together, so I wanted to create a place where all group members can go to discuss the project and break it down to smaller manageable parts and work on it collectively.

Moodle is a learning-based management system that is designed to help students, educators, and administrators to provide an online personalised learning environment. Today, Moodle is the most used online learning environment in the world, and the Moodle platform currently supports over 170 million learners (eThink, 2021).

Background

MyStudentHelper is being created to benefit students with organization and proof of work completed, as a student myself I have always wanted a product that can break down projects into smaller requirements and help me keep track of work completed, the group section of MyStudentHelper I feel is the solution to this problem. It will also get a third-party proof work completed by each student that worked on the project. In the past I have been working on a group project but the relationship between members in the group was not good and this made it very hard for collaboration on the project. Thankfully, we got the job done but one of the members did not do any work and I thought to myself there should be a way to prove this with minimal conflict. I think it is a good idea to create it as a plug-in for Moodle as it is currently the leading learning platform globally.

The main reason why MyStudentHelper is a plug-in for Moodle is because I wanted all the relevant information about college and student in the one place. Originally it was going to be its own site, but I figured it would be more convenient for students to gain access to it through Moodle as the college they attend would have already set them up with student emails and Moodle accounts. I personally think that when I make things that have extra sign-up methods and new passwords it detours people from benefiting from MyStudentHelper.

Aims

MyStudentHelper will help students organize projects and stay on top of the requirements of the project to help with completion and to help get maximum marks. The group section will also help all members of the group by keeping track of work done by each individual so they can all keep on top of work and see if other members are working too. I think this will help with the relationship of the group and lower anxiety levels to help each student focus on the task at hand. MyStudentHelper will also benefit lectures too because if a student is falling behind, they can identify the problem and ask relevant questions earlier.

Technology

Visual Studio Code: will be used to edit the code and to update Moodle. The two main community packages that will be used in this project are:

- **PHP Debug:** This will be used to detect errors in the code with Xdebug.
- **PHP Intelephense:** Product PHP development support.
- **PHP Console:** Commands will be inputted in this to update moodle on the server when any database or file creation changes are made.
- **Vscode-database:** support for PHPMyAdmin view database.
- **Mustache:** Support for mustache files.

XAMPP: will be used for its Apache HTTP Server to host moodle and MariaDB will be the database to store the information. MySQL will also be used to communicate to the MariaDB database.

PHP: will be used for general functionality, such as creating a section on moodle known as a block to display information.

PHPUnit: will be used for running automated tests when developing the plugin.

Bootstrap 4: A framework for developing HTML, JavaScript, and CSS code to make MyStudentHelper responsive on mobile and many other capabilities.

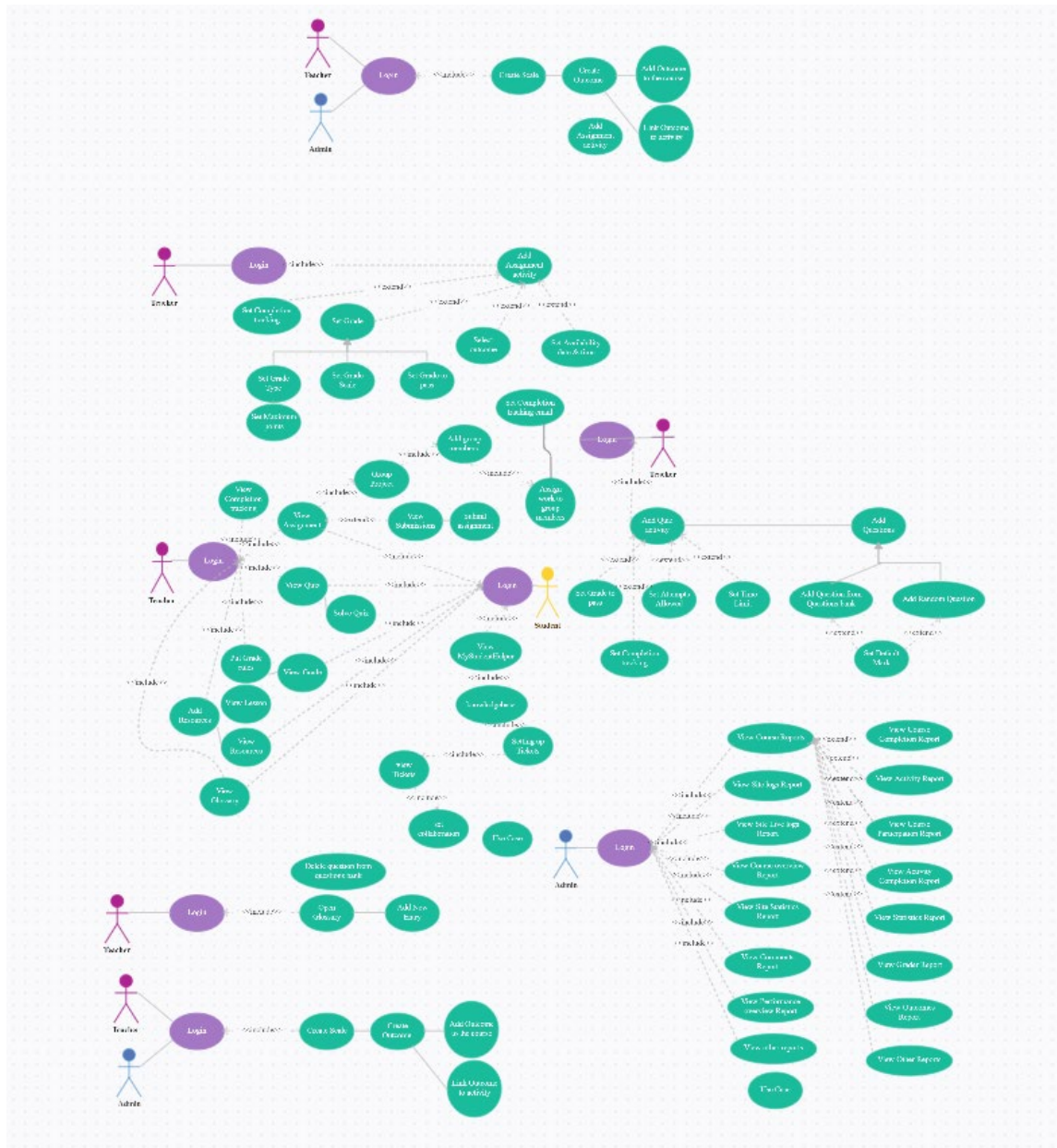
Mustache HTML: will be used when formatting the display of data outputs when following the moodle framework.

Structure

This document will outline the structure of Moodle and how I will be creating a plug-in. It will display use case diagrams for both a student and educator when using Moodle, and it will also show what the requirements are needed to run this software on all devices and will give some insight on why I wanted to create this project.

Requirement Specification

Moodle + MyStudentHelper Use Case Diagram



(<https://app.creately.com/diagram>, 2020)

Functional Requirements

Requirement 1: User Registration

Description & Priority

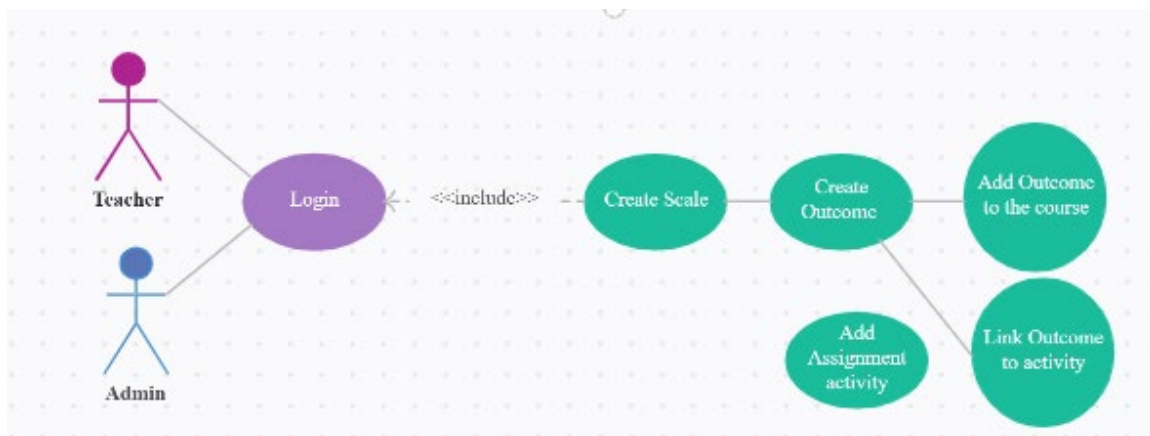
For MyStudentHelper to be successful the admin of Moodle must enrol the student to gain access to the MyStudentHelper plug-in.

Use Case

Scope: The Scope of this use case will describe how the user registers for MyStudentHelper

Description: This is an admins point of view in the registration process of MyStudentHelper

Use Case Diagram



(<https://app.creately.com/diagram>, 2020)

Flow Description

Precondition: No precondition

Activation: This use case begins when the admin of Moodle signs-up for enrolled students and adds them to their respected course.

Main flow:

1. The admin enters the correct information.
2. The admin then creates a scale.
3. The admin then creates an outcome.
4. The admin then adds the outcome to the course.
5. The admin then sets up assignments.
6. The user/Student can then access Moodle once they are sent credentials.
7. First time the user signs in they must create a new password.

Alternative flow:

1. The admin enters invalid information into the registration process.
2. The admin then tries to set up outcome.
3. The registration has been unsuccessful.
4. The user remains at the registration page.

Termination:

Moodle then stores the user information in MariaDB Database and is then directed to the login screen.

Post Condition

The user can now enter their login details.

Requirement 2: User Login

Description & Priority

For MyStudentHelper to be successful each user must login to Moodle correctly in order to access the plug-in

Use Case

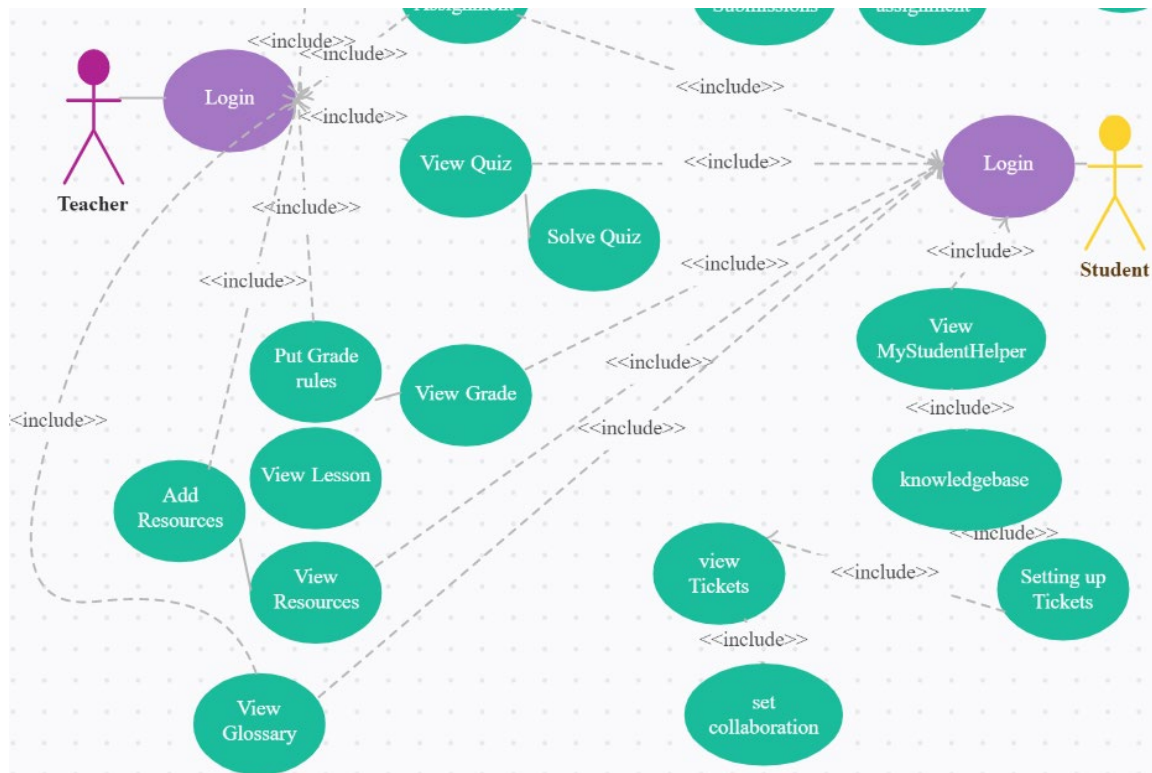
Scope

The Scope of this use case will describe how the user logs in for Moodle to access MyStudentHelper.

Description

This is a user's point of view in the login process of MyStudentHelper

Use Case Diagram



(<https://app.creately.com/diagram>, 2020)

Flow Description

Precondition

Requirement 1: User Registration

Activation

This use case begins after the admin signs up all users to Moodle.

Main flow

1. The user enters the correct information.
2. The user then clicks the login button.
3. The login has been a success.
4. The user then gets access to the main dashboard of Moodle.
5. The user can then view options depending on privileges such as the student can only view assignments, but the teacher can set up assignment rules.

Alternative flow

1. The user enters invalid information into the login process.
2. The user then clicks the login button.
3. The registration has been unsuccessful.
4. The user remains at the login page.

Termination

Moodle's main page loads correctly.

Post Condition

The user can now navigate through Moodle and click on the MyStudentHelper option.

Requirement 3: Setting up tickets

Description & Priority

The student must then navigate to the create ticket section and break down the project or assignment into smaller requirements, the plug-in will then display this on the Moodle main page.

Use Case

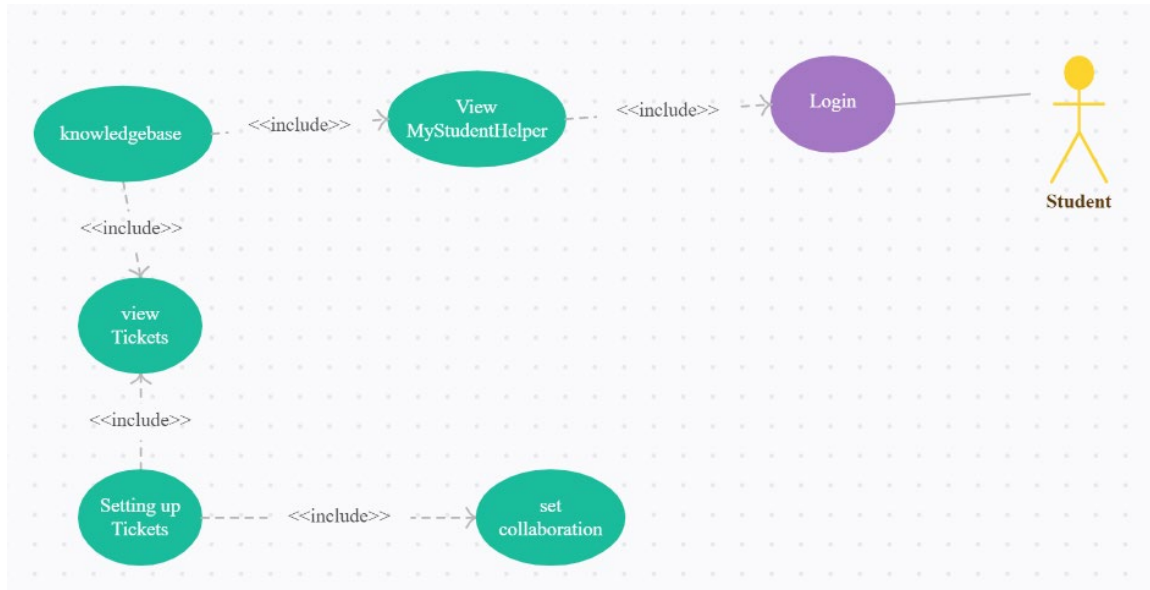
Scope

The Scope of this use case will describe how the user sets up a ticket for MyStudentHelper.

Description

This is a student's point of view in the setting up tickets process of MyStudentHelper.

Use Case Diagram



(<https://app.creately.com/diagram>, 2020)

Flow Description

Precondition: User must be signed up and logged in for this.

Activation: This use case begins when the student sets up their ticket so they can prioritise work.

Main flow

1. The student sets up a title for the ticket.
2. The student then adds a description of the ticket.
3. The student then saves this ticket.
4. The ticket information is stored in the MariaDB database.
5. The student can then view the ticket on the main dashboard.
6. The student can then mark the ticket complete and will be sent a corresponding email stating this.

Alternative flow

Termination

Moodle then stores the user information in MariaDB Database and is then directed to the Moodle dashboard screen.

Post Condition

The user can now view tickets on the main dashboard in order of due date.

Requirement 4: Setting up groups.**Description & Priority**

This is when a student is asked to do a group project and they want to set-up a group project page in order of prove of work completed by each student.

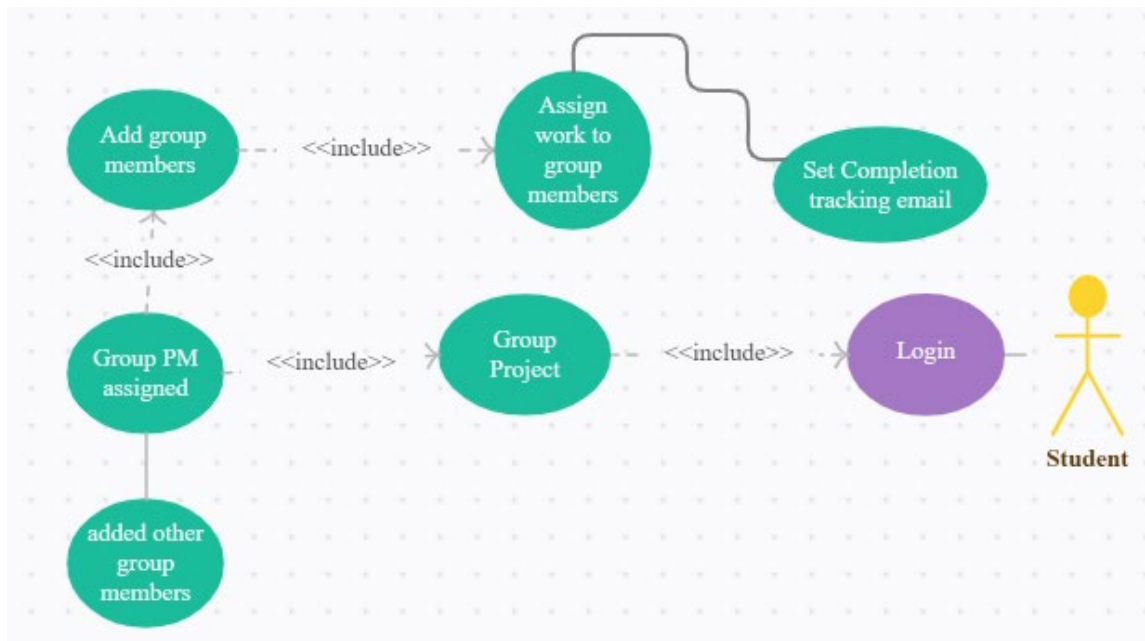
Use Case**Scope**

The Scope of this use case will describe how the user sets up a group project for MyStudentHelper.

Description

This is a student's point of view in the setting up a group in MyStudentHelper.

Use Case Diagram



(<https://app.creately.com/diagram>, 2020)

(Moodle, 2020)

Flow Description

Precondition

User must be signed up and logged in for this.

Activation

This use case begins when all the students in the group agree on a project manager and then set up the group project page.

Main flow

1. The group project manager sets of the group project page.
2. The project manager then adds other students on the project to the group.
3. All students then discuss and agree on what work should be done by each student.
4. The project manager then sets up group tickets by name and description.
5. When a task is complete the group member marks it as so.
6. All other students review the task and vote on if it is completed to the group standard.

Alternative flow**Termination**

Moodle then stores the user information in MariaDB Database and is then directed to the Moodle dashboard screen.

Post Condition

The user can now view all group members' tickets on the respected project and vote on completion.

Non-Functional Requirements

Hardware Requirements

- **Disk Space:** 200MB for all Moodle code and around 5GB free space to store all content.
- **Processor:** 1GHz (min) and 2GHz dual core or more.
- **Memory:** 512MB (min), 1GB or more but you would most likely need 8GB free space for a larger production server.

Software Requirements

This is the minimum server requirements, but it is recommending keeping you system up to date.

- **Moodle upgrade:** Moodle 3.5 or later.
- **PHP version:** minimum PHP 7.2.0 but PHP version has increased since Moodle 3.8. but PHP 7.3.x and 7.4.x are supported too.
- **PHP extension:** Mbstring is required.

Database requirements

The MyStudentHelper will support the following database requirements provided the minimum version is met as stated below.

PostgreSQL: recommended version is the latest and the minimum version is 9.6.

MySQL: recommended version is the latest and the minimum version is 5.7.

MariaDB: recommended version is the latest and the minimum version is 10.2.29.

Microsoft SQL Server: recommended version is the latest and the minimum version is 2012.

Oracle Database: recommended version is the latest and the minimum version is 11.2.

Browser requirements

Desktop:

- Google Chrome
- Firefox
- Safari
- Edge

Moodle 3.10 does not support Internet Explorer 11 and also Safari 7 is known to have issues, I would recommend the tested browsers listed above.

Mobile Requirements

Compatible with both Safari and Google Chrome.

Architecture and Design


















































































































































Moodle is structured with an application core, and plugin can be added to the application to further the capabilities of Moodle. If a developer customises Moodle without following the proper coding principles, Moodle will not recognise it as a plugin and no of the changes will be implemented correctly.

Moodle core

There are two types of users when using Moodle, Students, and teachers. The enrolment feature will add both types of users to a course. A course is made up of activities and resources such as files etc. A block is used on the dashboard part of Moodle. It is used to display some relevant information to what its associated with. A theme is all the styling for the overall Moodle look, A developer can create custom styles by implementing a new theme.

Moodle database

There are more than 250 tables in a Moodle database, there are core Moodle tables and Moodle tables for plugin. I have provided a screenshot of a sample of the database below.

Table 	Action
<input type="checkbox"/> mdl_analytics_indicator_calc	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_analytics_models	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_analytics_models_log	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_analytics_predictions	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_analytics_prediction_actions	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_analytics_predict_samples	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_analytics_train_samples	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_analytics_used_analysables	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_analytics_used_files	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assign	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignfeedback_comments	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignfeedback_editpdf_annot	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignfeedback_editpdf_cmnt	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignfeedback_editpdf_queue	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignfeedback_editpdf_quick	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignfeedback_editpdf_rot	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignfeedback_file	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignment	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignment_submissions	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignment_upgrade	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignsubmission_file	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assignsubmission_onlinetext	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assign_grades	  Browse  Structure  Search  Insert  Empty
<input type="checkbox"/> mdl_assign_overrides	  Browse  Structure  Search  Insert  Empty

Implementation

Set up Moodle: to set up Moodle for development there must be a variety of demands met.

Checking system capacity: one of the first thing a developer must do when setting up Moodle for development is to check your hardware meets all the requirements stated in the Hardware requirements section of this report, such as disk space, processor, and the memory this when a developer wants to run Moodle locally. When a developer wants to host Moodle one of the best options for it is Managed Moodle Cloud Hosting because it will support any number of users without the system crashing.

Install database and web server: To set up the database server download the latest version of XAMPP, once this is completed open the XAMPP control panel and start the Apache Server and MySQL. The port number of MySQL will then have to be configured in setting up Moodle later in the installation process.

Setting up database server: To start this process find phpMyAdmin in a respected browser to do this enter `https://ip_address/phpmyadmin` into the browser address bar. The next step is to create a Moodle data directory, it should be called moodle to gain access the core of Moodle's database.

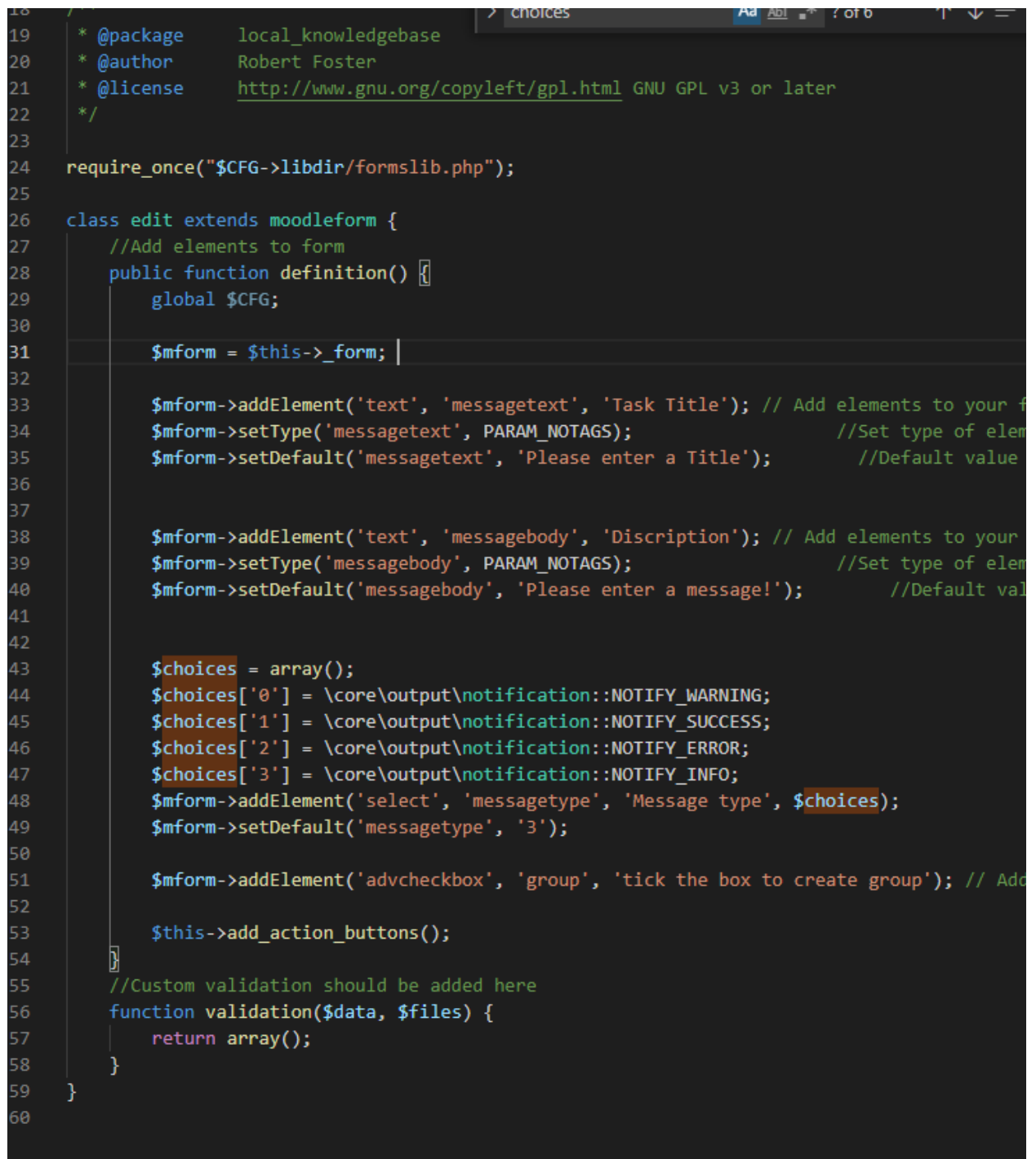
Setting up code editor: There a couple of steps to take when setting up a code environment for developing Moodle.

To implement the MyStudentHelper plugin for Moodle the following files will have to be created: Location: directory name/file name, explanation.

Root Directory: knowledgebase, this is where all the files for the plugin are stored.

Classes Directory: edit.PHP, the logic will be stored taking the input from the user and creating an entry for the knowledgebase. The file takes in information about the entry title, body, checkbox, and a type. It also uses a partial for the form to be implemented on the show page. All of this was coded following the principles that Moodle was created by. This is

shown in the screenshot below.



```
18  /*
19  * @package      local_knowledgebase
20  * @author       Robert Foster
21  * @license      http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later
22  */
23
24  require_once("$CFG->libdir/formslib.php");
25
26  class edit extends moodleform {
27      //Add elements to form
28      public function definition() {
29          global $CFG;
30
31          $mform = $this->_form;
32
33          $mform->addElement('text', 'messagetext', 'Task Title'); // Add elements to your f
34          $mform->setType('messagetext', PARAM_NOTAGS); //Set type of elem
35          $mform->setDefault('messagetext', 'Please enter a Title'); //Default value
36
37
38          $mform->addElement('text', 'messagebody', 'Discription'); // Add elements to your
39          $mform->setType('messagebody', PARAM_NOTAGS); //Set type of elem
40          $mform->setDefault('messagebody', 'Please enter a message!'); //Default val
41
42
43          $choices = array();
44          $choices['0'] = \core\output\notification::NOTIFY_WARNING;
45          $choices['1'] = \core\output\notification::NOTIFY_SUCCESS;
46          $choices['2'] = \core\output\notification::NOTIFY_ERROR;
47          $choices['3'] = \core\output\notification::NOTIFY_INFO;
48          $mform->addElement('select', 'messagetype', 'Message type', $choices);
49          $mform->setDefault('messagetype', '3');
50
51          $mform->addElement('advcheckbox', 'group', 'tick the box to create group'); // Add
52
53          $this->add_action_buttons();
54      }
55      //Custom validation should be added here
56      function validation($data, $files) {
57          return array();
58      }
59  }
60
```

DB Directory: install.xml, the logic for the database definitions in an xml format. This is shown in the screenshot below.

```
local > knowledgebase > db > install.xml
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <XMLDB PATH="local/knowledgebase/db" V
3 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 xsi:noNamespaceSchemaLocation="../../lib/xmlldb/xmlldb.xsd"
5 >
6 <TABLES>
7 <TABLE NAME="local_knowledgebase" COMMENT="each record is a message">
8 <FIELDS>
9 <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" SEQUENCE="true"/>
10 <FIELD NAME="messagetext" TYPE="text" NOTNULL="true" SEQUENCE="false"/>
11 <FIELD NAME="messagebody" TYPE="text" NOTNULL="true" SEQUENCE="false"/>
12 <FIELD NAME="messagetype" TYPE="text" NOTNULL="true" SEQUENCE="false"/>
13 <FIELD NAME="timestamp" TYPE="int" LENGTH="10" NOTNULL="true" DEFAULT="0" SEQU
14 </FIELDS>
15 <KEYS>
16 <KEY NAME="primary" TYPE="primary" FIELDS="id"/>
17 </KEYS>
18 </TABLE>
```

Templates directory: show.mustache, this is the HTML, JavaScript and Bootstrap out put that the user will see and interact with.

Edit.php in root directory: edit.php, this is a page that will give options to display the form output to the user and follows the Moodle framework with hooks to ensure that the output of the header stays consistent with the core of Moodle. This is shown in the screenshot below.

```

27 global $DB;
28
29 $PAGE->set_url(new moodle_url('/local/knowledgebase/edit.php'));
30 $PAGE->set_context(\context_system::instance());
31 $PAGE->set_title('Edit');
32
33
34 // We want to display our form.
35 $mform = new edit();
36
37
38
39 if ($mform->is_cancelled()) {
40     // Go back to manage.php page
41     redirect($CFG->wwwroot . '/local/knowledgebase/show.php', 'You cancelled the message for
42
43
44 } else if ($fromform = $mform->get_data()) {
45
46     // Insert the data into our database table.
47     $recordtoinsert = new stdClass();
48     $recordtoinsert->messagetext = $fromform->messagetext;
49     $recordtoinsert->messagebody = $fromform->messagebody;
50     $recordtoinsert->messagetype = $fromform->messagetype;
51
52     $DB->insert_record('local_knowledgebase', $recordtoinsert);
53
54     // Go back to manage.php page
55     redirect($CFG->wwwroot . '/local/knowledgebase/show.php', 'You created a message with ti
56 }
57
58 echo $OUTPUT->header();
59 $mform->display();
60 echo $OUTPUT->footer();
61
62

```

Lib.php in root directory: lib.php, this is where the form will add notifications, store the read messages for the users so they do not display once they are read. This will store the notifications in the database. This is shown in the screenshot below.

```

18 * @package      local_knowledgebase
19 * @author       Robert Foster
20 * @license      http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later
21 */
22
23 function local_knowledgebase_before_footer() {
24     global $DB, $USER;
25     $sql = "SELECT lm.id, lm.message, lm.message_type
26             FROM {local_knowledgebase} lm
27             left outer join {local_knowledgebase_read} lmr ON lm.id = lmr.messageid
28             WHERE lmr.userid <> :userid
29             OR lmr.userid IS NULL";
30     $params = [
31         'userid' => $USER->id,
32     ];
33     $messages = $DB->get_records_sql($sql, $params);
34     foreach ($messages as $message) {
35         $type = \core\output\notification::NOTIFY_INFO;
36         if ($message->message_type === '0') {
37             $type = \core\output\notification::NOTIFY_WARNING;
38         }
39         if ($message->message_type === '1') {
40             $type = \core\output\notification::NOTIFY_SUCCESS;
41         }
42         if ($message->message_type === '2') {
43             $type = \core\output\notification::NOTIFY_ERROR;
44         }
45         \core\notification::add($message->message, $type);
46
47         $readrecord = new stdClass();
48         $readrecord->messageid = $message->id;
49         $readrecord->userid = $USER->id;
50         $readrecord->timeread = time();
51         $DB->insert_record('local_knowledgebase_read', $readrecord);
52     }
53 }
54

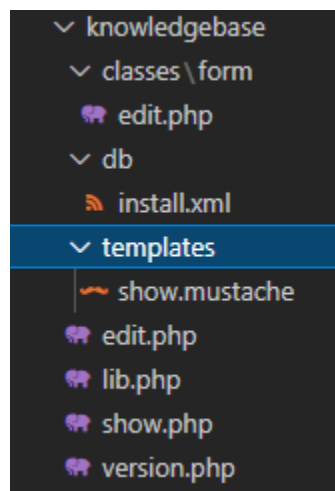
```

Show.php in root directory: show.php, this will show the GUI to create an entry for the knowledgebase and ticketing system. The user will be able to navigate to this page from the dashboard.

Version.php in root directory: version.php, this will state the version of the plugin and the version that Moodle must match to be able to run the plugin. Once this file is made an upgrade will be prompted when accessing Moodle. This is shown in the screenshot below.

```
17 /**
18  * @package    local_knowledgebase
19  * @author      Robert Foster
20  * @license     http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later
21  * @var stdClass $plugin
22  */
23
24 defined('MOODLE_INTERNAL') || die();
25 $plugin->component = 'local_knowledgebase';
26 $plugin->version   = 2020110900;    // The current module version (Date: YYYYMMDDXX)
27 $plugin->requires  = 2020110300;    // Requires this Moodle version
28
```

There is a screenshot below of the directory layout of the directories.



Testing

Moodle supports PHPUnit_intergration testing.

Database check: When the version file is created, or the database fields are updated this command must be ran to uninstall the plugin as seen in the screenshot below.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

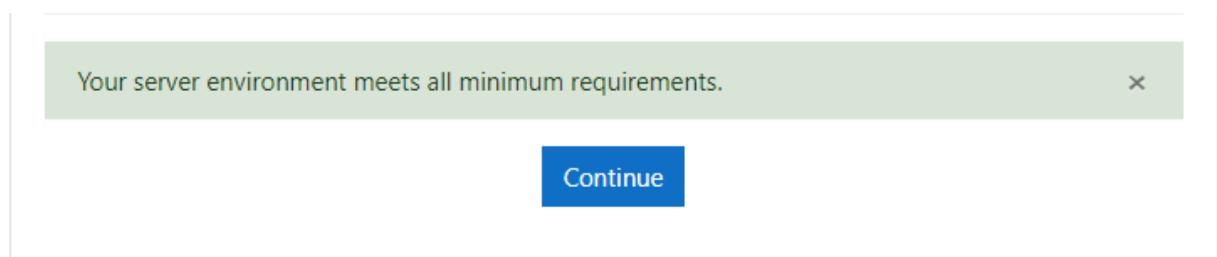
Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\xampp\htdocs\moodle> php admin/cli/uninstall_plugins.php --plugins=local_knowledgebase --run
Uninstalling: local_knowledgebase [pluginname,local_knowledgebase]
-->local_knowledgebase
++ Success ++
```

Then the developer has to go to the browser address bar and find Moodle, in my case I had to go to <http://localhost/moodle> and this would run server checks on the new plugin, Moodle locates the new files by the version number stated in the version.php file as seen in the screenshot below.

```
defined('MOODLE_INTERNAL') || die();
$plugin->component = 'local_knowledgebase';
$plugin->version = 2020110900; // The current module version (Date: YYYYMMDDXX)
$plugin->requires = 2020110300; // Requires this Moodle version
```

The server check page states the current release information. If your server meets all the intended requirements, it will state this message as seen in the screenshot below.



The next step is the plugins check this is also part of Moodle and it will state the following information as in the screenshot below.

Plugins check

This page displays plugins that may require your attention during the upgrade, such as new plugins to be installed, plugins to be upgraded, missing plugins etc. Additional plugins are displayed if there is an available update for them. It is recommended that you check whether there are more recent versions of plugins available and update their source code before continuing with this Moodle upgrade.

Check for available updates

Last check done on 12 May 2021, 3:36 PM

Plugins requiring attention

Cancel new installations (1)

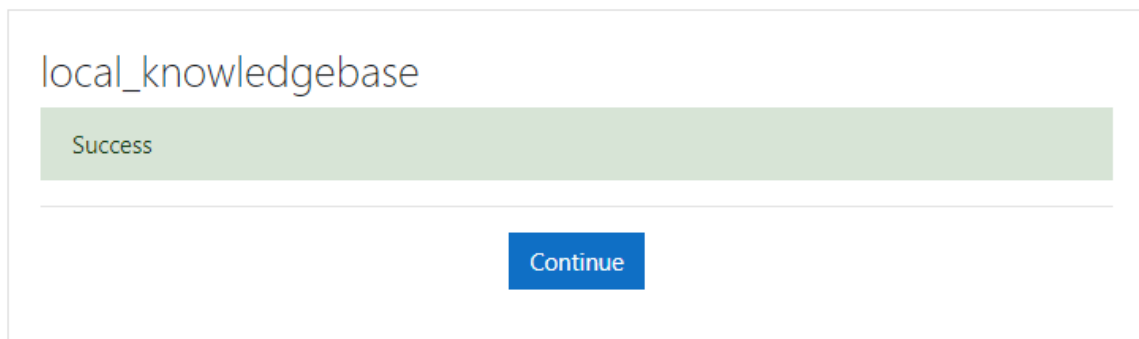
Plugins requiring attention 1

All plugins 419

Plugin name / Directory	Current version	New version	Requires	Source / Status
Local plugins				
[pluginname,local_knowledgebase] /local/knowledgebase		2020110900	• Moodle 2020110300	Additional To be installed Cancel this installation
Reload				
Upgrade Moodle database now				

The next step is to upgrade your database, this will check if there are any errors in the database code and if there are no errors it will give an output as seen below.

Upgrading to new version



If the database check is unsuccessful the developer would have to make the relevant changes and restart this entire process again.

Code testing: PHP Debug was incorporated into VS Code to detect coding errors as seen in the screenshot below.

```
25
26 class edit extends moodleform {
27     //Add elements to form
28     public function definition() {
29         global $CFG;
30     }
```

Graphical User Interface (GUI)

At the login section of Moodle the users can enter their details for accessing the site. This can be seen in the screenshot below.

MyStudentHelper-Moodle

Forgotten your username or password?

Cookies must be enabled in your browser ?

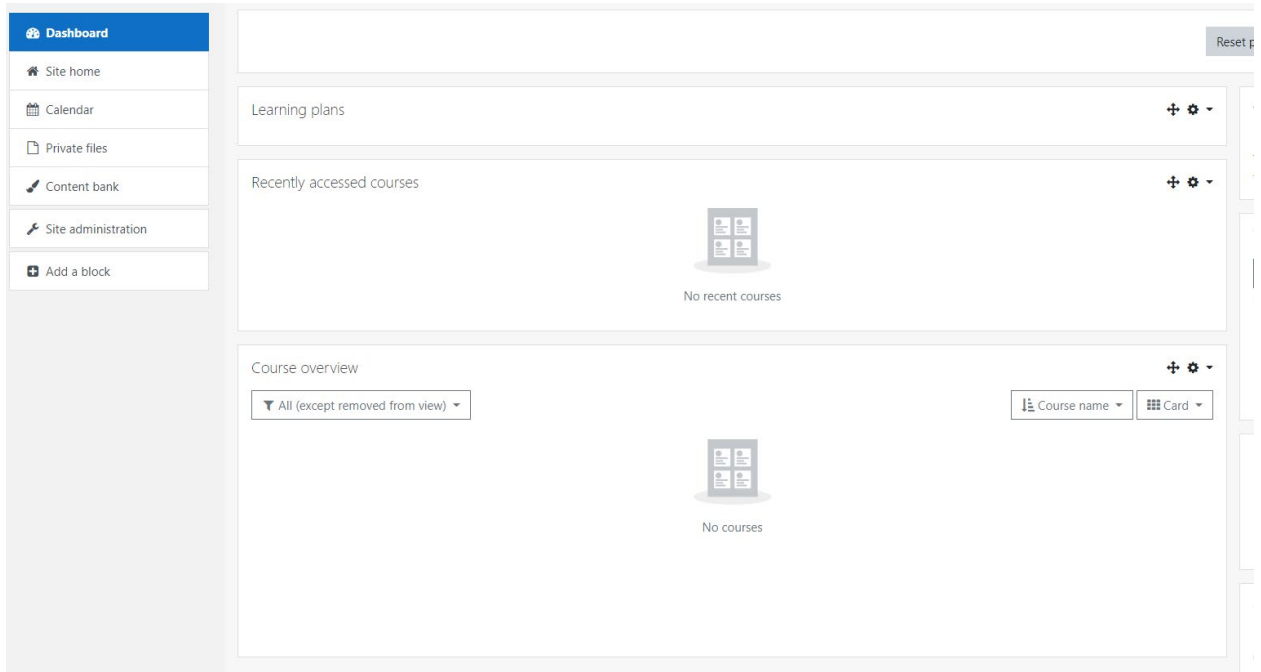
Some courses may allow guest access

Log in as a guest

☒ Remember username

Log in

The next screen the user will see is the dashboard part of Moodle that is essentially the main interface of Moodle. This is shown in the screenshot below.



From here the user can navigate to the MyStudentHelper section of the website. When a student wants to make an entry to the knowledgebase or set up a ticket, they can by navigating to the form page of MyStudentHelper. The form will have input fields as seen below.

Task Title	<input type="text" value="Please enter a Title"/>
Discription	<input type="text" value="Please enter a message!"/>
Message type	<input type="text" value="info"/>
<input type="checkbox"/> tick the box to create group	
<input type="button" value="Save changes"/> <input type="button" value="Cancel"/>	

Once an entry is made it can be viewed on the show page, the data from the entry is displayed using moustache HTML as seen in the screenshot below.

You created a message with title Assignment 3

Assignment 3

robert

DIARY ENTRIES

DIARY TITLES

Assignment 1Assignment 2Assignment 3

Create Entry

Conclusion

During this project I really tried to challenge myself by creating a plugin for Moodle, I had no previous experience in working with php or the Moodle framework. MyStudentHelper has some functionality but not all, this was due to hardware issues as stated in my reflective journal. I understand how to implement the plugin and all the fundamentals associated with it.

The whole idea of MyStudentHelper was a viable one in my opinion especially because of Covid restrictions, I know many students have found it very difficult and I wanted a way to connect people and lower anxiety levels when completing college assignments. I think the solution is MyStudentHelper because it will allow a student to privately write notes that the student can access when signing into the Moodle platform. I think it would be a very good idea especially for software students to be able to take projects and break them down into smaller manageable tasks.

The group section of the MyStudentHelper will bring students together by working on the group projects while at home, I think this would work because the students will have a common goal to reach and this could even create friendships, I know this because I have created friendships through working on projects in college and I prefer to work that way.

Overall, I am happy with my achievements on this project considering the circumstances and the hardware issues I faced. I have a good understanding of PHP code and will consider using it in the future

on projects, I also got to showcase my XML skills as I am familiar with this from past assignments when in university. The part I found the most frustrating was using the moustache HTML templates for outputting of the information from the database, this really bothered me because I consider myself a good web developer but in newer languages like Ruby.

Further Development or Research

If I had more time, I would try new applications using moustache HTML to get used to how it works because there is not a lot of information about it online. I would create a stand-alone journal to study the functionality and incorporate it into my development Moodle server.

I would have implemented a ticketing system that showed up on the dashboard of the Moodle website as a block indicating using the warning functionality I incorporated when the due date is near, or the user could make their own customisable due dates for the ticketing system. An email confirmation will also be incorporated to send to the student on completion of the ticket, this email would also store the relevant information about the work completed when working on a group project.

I would also love to implement this idea and submit it to Moodle, and they could run integration tests on it and publish it live as a downloadable plugin.

Bibliography

eThink. (2021, May 2). *What is Moodle?* Retrieved from ethinkeducation:

<https://ethinkeducation.com/uk/what-is-moodle-guide/#what-is-Moodle>

<https://app.creately.com/diagram>. (2020, December 20). Retrieved from Creately.com:

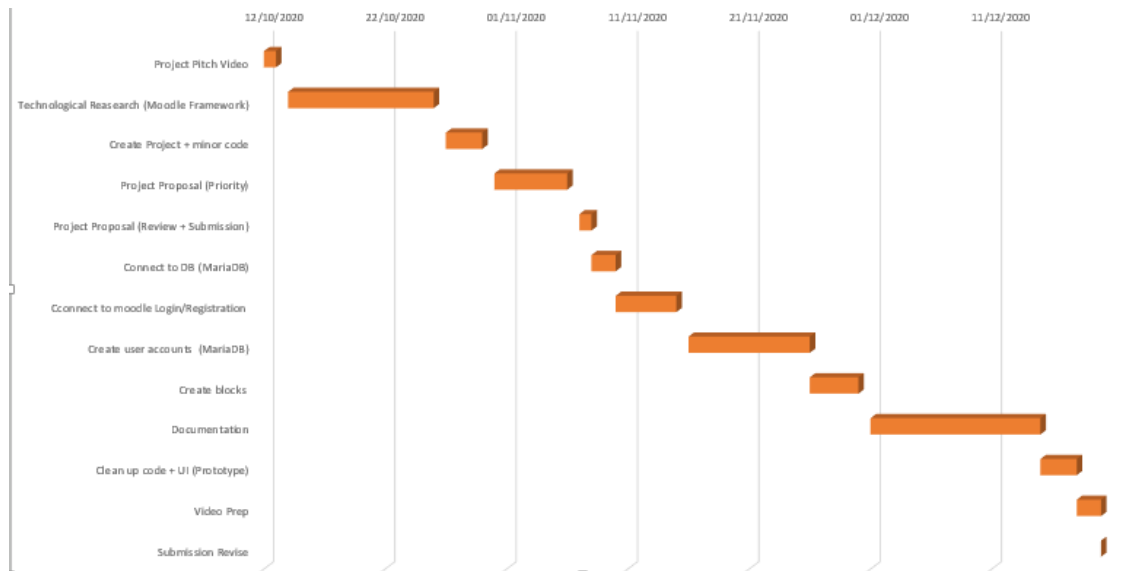
<https://creately.com/>

Moodle. (2020, December 20). *Moodle documentation* . Retrieved from Moodle.org:

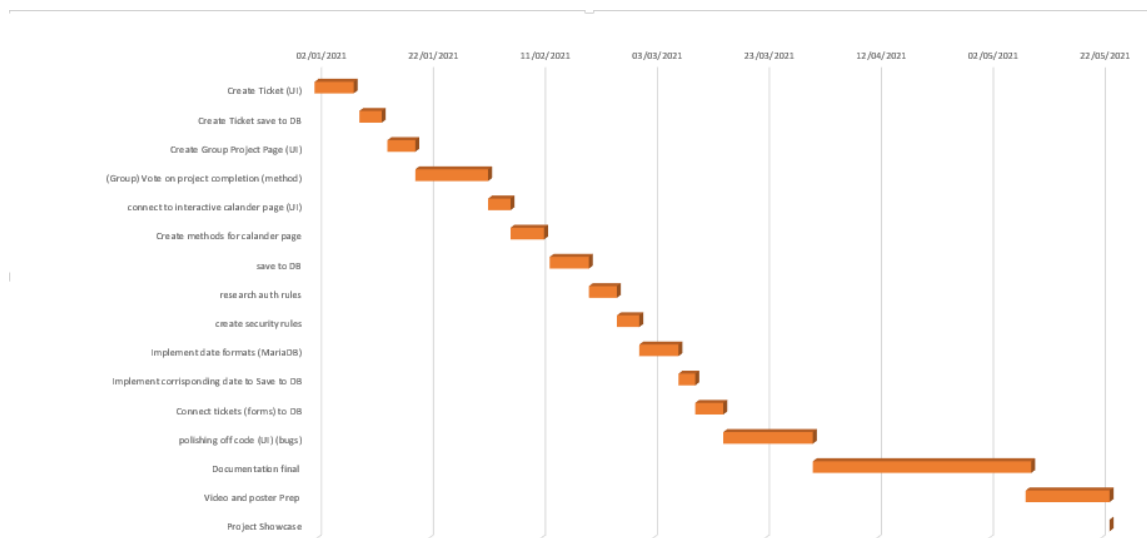
<https://moodle.org/>

Project Plan

2020



2021



Reflective Journals

October 2020

I researched how I am going to build my project this month and got started on making small code changes to the app just to set up the project and have it connected to my IDE.

I have decided to use AngularJS framework because it's used by a lot of companies today and I would love to work with JS. I completed the AngularJS tour of hero's tutorial to just familiarise myself with the

framework and where the components go. I also have decided to use SCSS to style my website, also because it is used in the industry today and it's very easy to make modifications to the website in a future stage of development.

I then booked a slot with computing support to go over the use of SCSS and AngularJS together and how to upload projects to GitHub correctly. I just want to try and make sure I don't face any roadblocks while doing this project.

I then created my project pitch video explaining my project and uploaded it, and have currently complete my project proposal I just want to print it out and go over it and make changes to any errors or rewording of the document to get it to a professional standard.

I have been really enjoying working from home I think I get way more done. I have been at my desk from 9am to 6pm every day since college started back and I want to keep this trend going because I feel as if it improves my work and it makes my work more manageable.

The great thing is I would really use my website right now and its features so I am excited about the idea and getting started to build it.

November 2020

So, this month I started creating a prototype to get a feel of what my website would look like, I wasn't happy and after speaking to my supervisor I have decided to still create the same project but not as its own website but rather as a plug-in for Moodle. This means that all of the student's information will be stored in the same place.

I had to do a lot of research on how to develop a Moodle plug-in, I found on the Moodle website that they have documentation to show me how to create this. I needed to download XAMPP and I had issues with ports so I had to do some error fixing this took me a couple days. I then had to host a blank version of Moodle on XAMPP but I got this done fairly quickly.

I then got started on my mid-point documentation again because I had to change my proposal to fit the new updated version of my project.

December, 2020

This month I felt a little short on my work because I had other classes to work on such as mobile technology and an AI chess game I have been developing as part of my course in NCI. I got that finished and submitted last week and then focused on my software project documentation and the prototype.

I also took a few lessons on developing Moodle through YouTube and documentation on the Moodle website. I am going to submit my December reflective journal before Christmas because I am afraid I will forget to do this.

I then worked on my user requirements and use case diagrams, once I completed all of this and my prototype I then made the video for the presentation. I don't like doing this at all and would much rather be interviewed because I have no experience in creating videos and editing videos. I am kind of worried that I will be marked on this negatively but I'm happy that I have done the best I can.

January, 2021

At the start of this month, I focused on trying to set up a local version of Moodle and to create my first plug in which was going to be a messaging plug in just for practice to get used to developing on Moodle.

I found tutorials on YouTube on how to download and set up a local version of Moodle. I then downloaded XAMPP and configured it. I had to set up a phpMyAdmin server and create a Moodle database to store the local Moodle data. It was my first time using phpMyAdmin, but I got the hang of it quickly enough my laptop did take ages to load the database information, and this took up time and I was worried that it will slow me down and frustrate me further.

I then had to install all the relevant requirements for Moodle and make small changes to the config file in XAMPP. I kept running into problems with the installation process on my laptop and every time I must restart it took ages to get going. I found myself getting very frustrated and had to take a break from doing the project and try focus on other classes.

February, 2021

At the start of this month, I looked at the YouTube tutorials I found online on creating a Moodle plugin, I wanted to have a good understanding of the structure of creating and editing plugin with my local version of Moodle.

I found a plugin tutorial on YouTube on how to create a messaging service with Moodle, I started following the tutorial and making changes in the code to create my plugin. I ran into problems because every time I wanted to make a change, I had to update Moodle, this took a toll on my laptop and every time I had to make the update it could take up to an hour even for small changes.

I have had my laptop for the whole way through college, and I was planning on getting a PC and had some money saved so I spent the rest of the month saving money and following tutorials online but found it very hard to make any changes as Moodle took so long to update. I was a bit gutted this month because I wanted to get a lot done for my Software Project, but my hardware issues slowed me down.

March,2021

At the start of this month, I knew that I would have had to focus on order projects that are due for my order modules, so I knew it would be hard for me to focus on Moodle and development.

I had a look at the Moodle documentation online, and I started to populate my local version of Moodle by setting up mock courses, assignments, and other activities to see how it all works. I wanted to set up a few users and add them to groups to see how it works from an Admin point of view to get a better understanding of the code that I would need to complete my project. I had to follow YouTube tutorials to get an understanding on how to populate Moodle and set it up from an admin point of view. Once I did all this, I started making small changes in the code but ran into the same problems with Moodle updating.

I started to get worried about the local version of Moodle being so slow and focused on documentation a little and setting up the Showcase profile. I then had to focus on distributed systems project that was coming up soon.

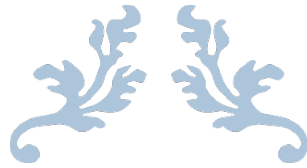
April,2021

At the start of this month, I bought my PC online and was waiting for it to be shipped I was excited for this as I was getting a high-powered custom PC. I taught to myself that I will focus on upcoming projects and exams in other modules that I had at this month, and once my computer arrives, I will focus more on developing Moodle.

I had to redownload Moodle on the new computer and configure it but with the new computer this process did not take a lot of time as I have done it before. I did not want to completely forget about my software project, so I did a lot of documentation this month and I made a lot of diagrams for the

technical report. I thought this was a good idea because I did not have loads of time to focus on the software project because I had a Ruby on rails project due that was worth 100 percent of the module.

Overall, this month even though I had a lot to do in order modules, I was happy with the work that I got done because I reset up Moodle and configured it this process takes a lot of time as you have a lot of fields to fill out and a lot of reading to do to make sure that you are setting up Moodle correctly. I did install the plugin that I previously made to see how quick the process sped up and I was very happy with the results.



MyStudentHelper



Robert Foster, X17140404

National College of Ireland

BSc (Hons) in Computing

Specialisation : Software Development

Contents

Objectives	38
Background.....	39
Technical Approach.....	40
Project Plan	41
Gantt chart.....	41
Technical Details	42
Evaluation	42

Objectives

The objective of MyStudentHelper will be a moodle plug-in ticking system for students that will manage the workload for the users, by creating a task manager for assignments based on priority of due date. The tasks that need to be completed will be listed on moodle in order from top to bottom on what needs to be done. The user will create their own tickets and the system will ask for a brief description of the task, this will help with keeping on track of the workload and have a clear to do list and this can lower anxiety levels in students and give a better experience of managing multiple projects and tasks.

- Scenario: A student in NCI has started their year and they have a part time job and three different modules with three assignments for each module. They go to the MyStudentHelper section on moodle and then input their assignments and due dates, they then check MyStudentHelper and see all of their projects listed by date.

There will also be a group section that will help manage the workload between group members, there will be a project manager and they will set up the group project page for the rest of the members. Each member will be advised to discuss the workload and agree on what member should complete each task, the project manager will then input this data into the system and allocate the workload. Each individual on the project will then have their own tasks and they can tick off each task on completion, the rest of the group members can then review the work and vote if the work is completed, they can also write comments on general feedback about the task completed.

When each task is completed there will be a receipt sent to the email of the user stating that the task complete as proof. This will show the lecturer proof that the individual completed to work and will help with the self-evaluation forms for when the project is being marked, it will also highlight at an early stage on what team members is working or not working so they can identify the problem when it's not at too late of a stage and will increase the groups grades.

- Scenario: Four students get selected to be in a group project and they all have to work together and complete it remotely, one member of the group sets up a group project using

MyStudentHelper and invites the other students working on the project. All students agree on what tasks they should complete and once completed the other students can vote if it's up to standard or completed, once it's agreed that the task is complete a receipt gets sent to the student as proof of completion.

Background

I have experience in developing application both professionally and for college assignments, I could be working on up to three projects at any given time. I have had trouble keeping track of work that was already completed and the requirement's left for the project. The issues I face would be tracking down what class or method that I was last working on. I have been keeping a programming journal for the past five years now I use it like a diary, but it has started to get old and worn down over time. I kept running into issues such as searching through the copy for too long and then having to research a solution online of a problem I have solved in the past or finding multiple to do lists for the same subject. This caused me to lose productivity and has made me anxious in the past.

I also have experience in working for an IT company during my academic internship in my third year of computing in NCIRL. The internship went on for six months and I learned so much about the industry in that time. The IT Company used a knowledge base but it only stored text in it and you could only create a title for the document but you did not have a search bar to locate it, so as a new employee I spent a lot of time reading information that I did not require.

I also gained experience with IT issues from customers, the customer would call the company and log an IT issue if there was any extra information required the developer would have to email the customer and this could take some time depending on availability. I think if there was an allocation system that ties in with a developer's personal requirement's system these problems could be avoided, saving the company time and money.

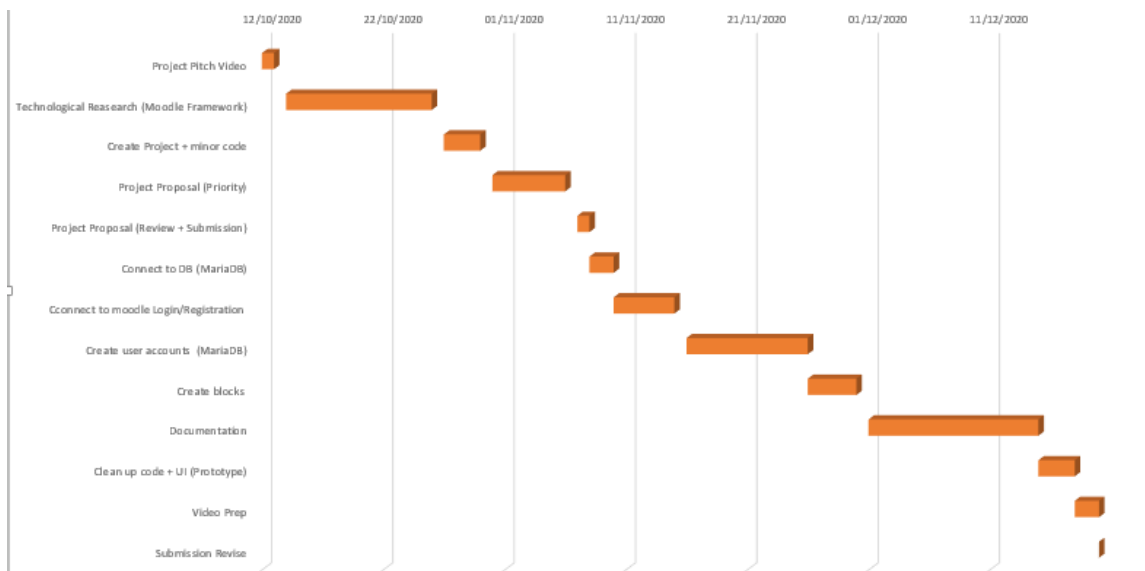
Technical Approach

- A first time user of MyStudentHelper will click on the plug-in and will already be logged in through the moodle system.
- A returning user will login to moodle with username/email and password.
- The user will then be at the main page of moodle and there will be a MyStudentHelper section where they can view the dashboard and tickets that they have previously setup.
- The dashboard will display a navigation bar consisting of the MyStudentHelper logo that when clicked it will return the user to the MyStudentHelper section on moodle and there will be options on the navigation bar will be create ticket and group page.
- The create ticket option when clicked will display a popup box that will prompt the user for a subject name, due date, and a description. It will automatically create a timestamp when a ticket is created too. When the ticket is complete, MyStudentHelper will send a receipt via email to show completion of work with a reminder to submit work for college.
- The group section will be the same as the create ticket with some extra features such as assigning work to other users and a voting system that will mark the ticket as complete. It will then automatically send a receipt to the user on completion of work.

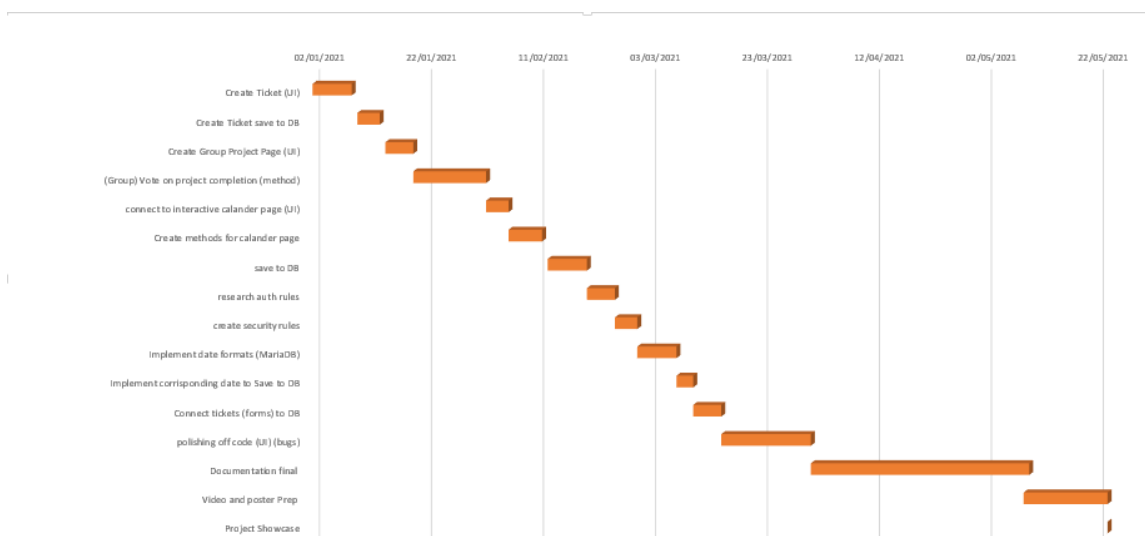
Project Plan

Gantt chart

2020



2021



Technical Details

- **PHP:** Will be used for general functionality, such as creating a section on moodle known as a block to display information.
- **CSS:** CSS language will be used for styling and positioning MyStudentHelper.
- **ATOM:** Atom is an open-source integrated development environment that the website will be built on, it will need the following community packages that are easily downloadable and installable.
- **PHP Debug:** This will be used to detect errors in the code.
- **PHP Console:** Commands will be inputted in this to update moodle on the server when any changes are made.
- **XAMPP:** will be used for its Apache HTTP Server to host moodle and MariaDB will be the database to store the information. MySQL will also be used to communicate to the MariaDB database.

Evaluation

Overall I think if I had a product such as MyStudentHelper when I started out in college it would have improved my grades over all by having all my requirements for projects in the same place corresponding with due dates, this will help because most student have many modules and many tasks for each module, so to have them all on the same page it will help prioritize what assignment to work on at any given time.

Over all I think if I had a product like MyStudentHelper while I was starting out as a programming student I think it would of increased my grades. That is the whole thinking behind MyStudentHelper side

of things is to organise projects and the correct way, I think this would have also increased my chances of staying working on a few personal programs that could have improved my programming skills. I would love if this project actually was downloaded by students to help improve grades. The idea of increasing grades and MyStudentHelper being a benefit to that is the whole idea.

I think that the group projects page would have also helped me in college because I had a small circle of friends in college and I was always nervous about being in a group project with people I don't know, I was worried that if I did work and sent them the code that they would take it and act like they made it. At the end of most group projects the students are asked to fill out a self-evaluation form and I was always nervous about students saying I didn't do much work to increase their own grades because they can say they did more work on the project overall, as a student I didn't know if that would actually effect my grades but it still made me anxious. The receipt part of the group page will help prove the work done by each student and avoid all of this conflict. I think this would hugely help students work together too because if somebody is struggling with a part of the project they can identify it early and work together on a solution.