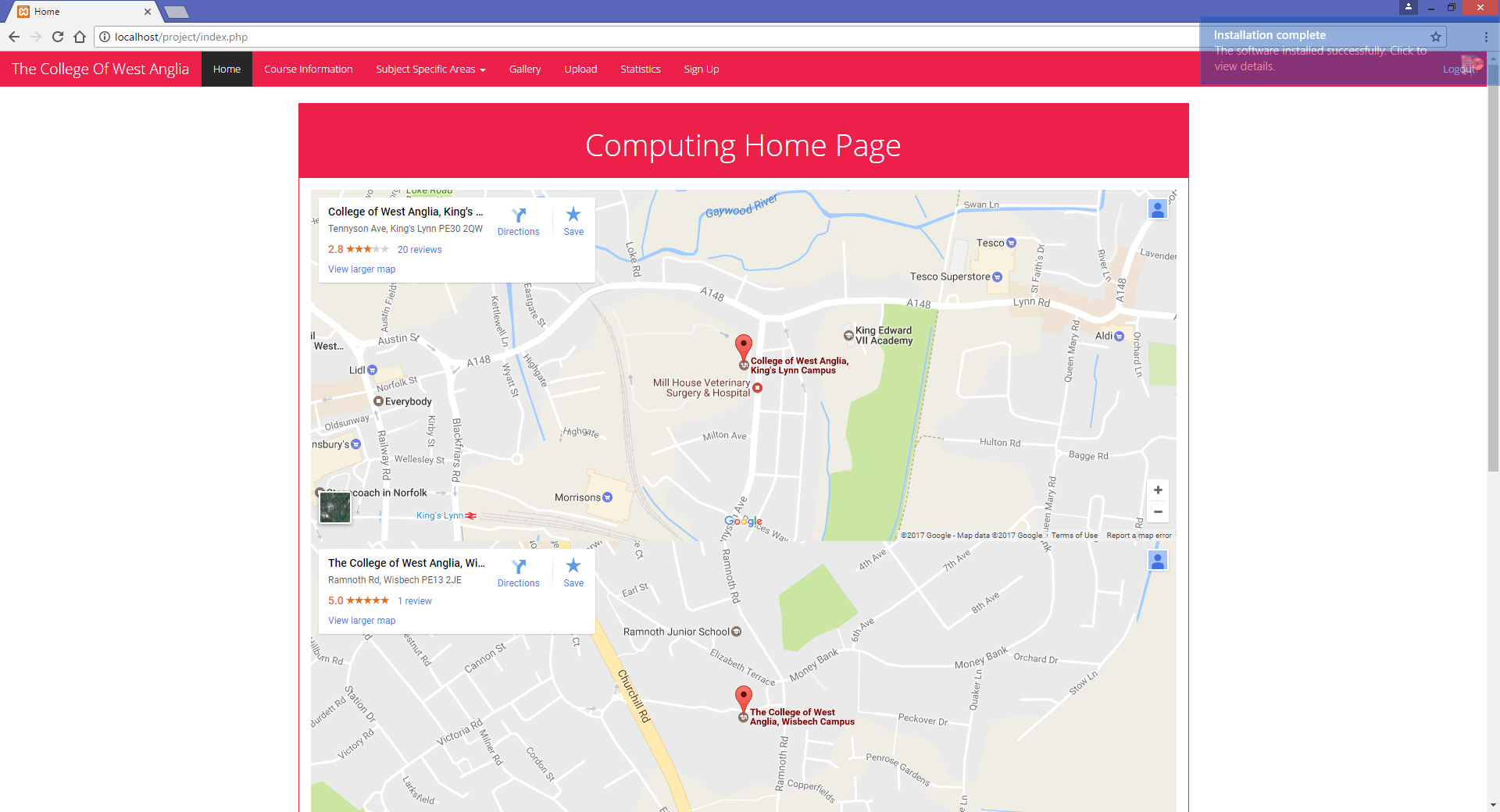
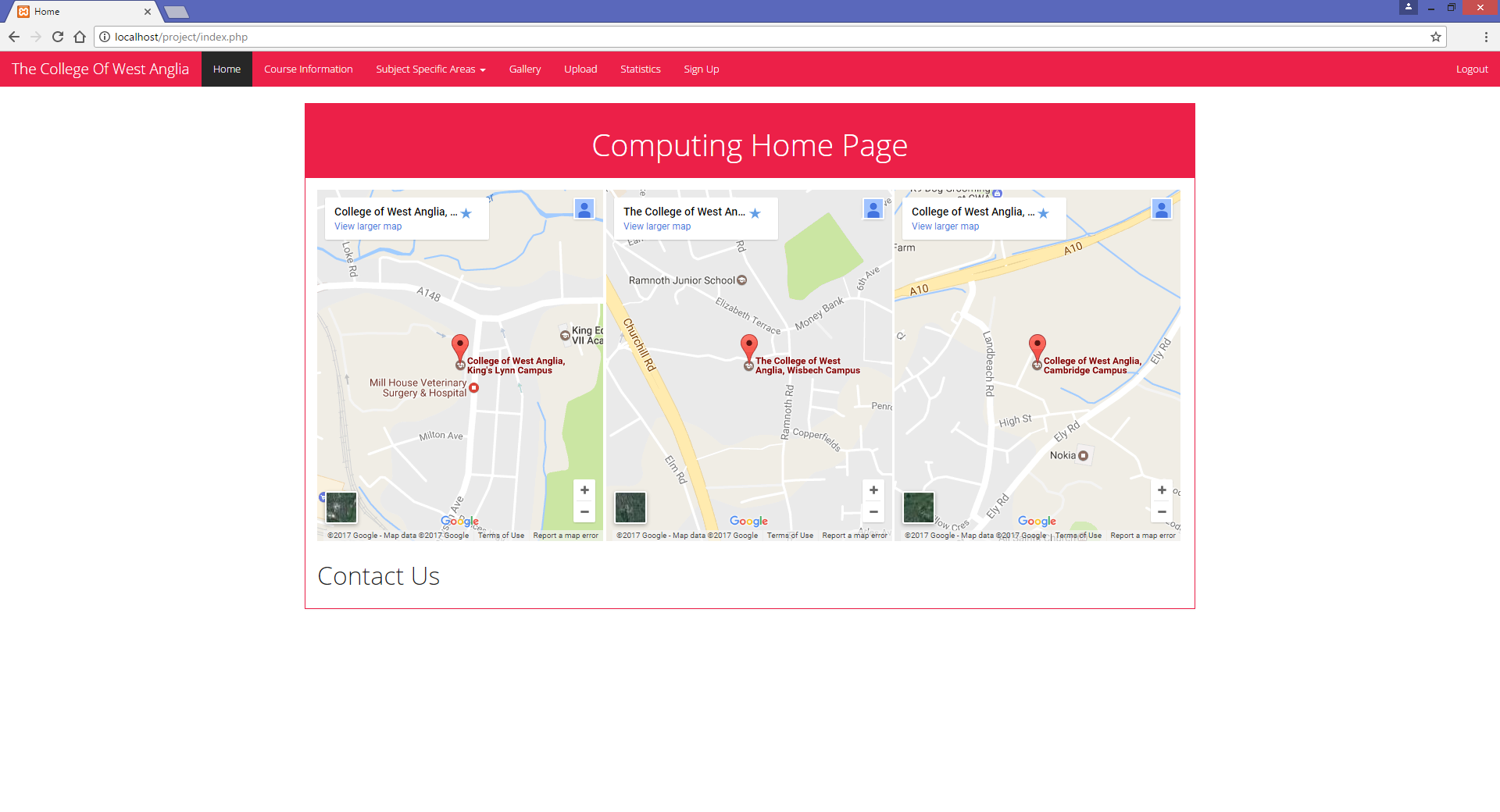
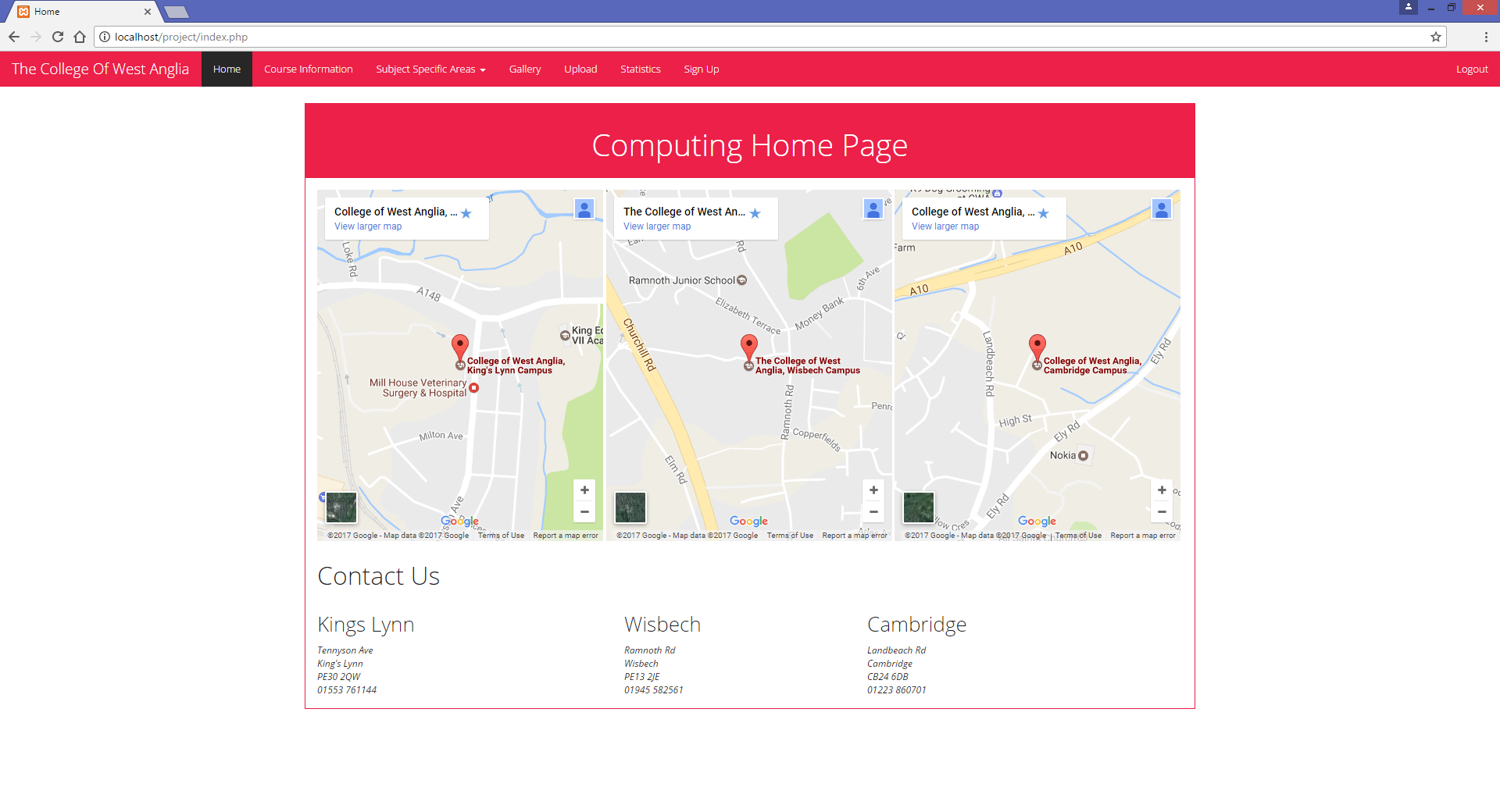
# Home Page



Here I have made the home age my contact page. I got the embedded links from google maps and put the code in to page. They are too big so will have to adjust the width

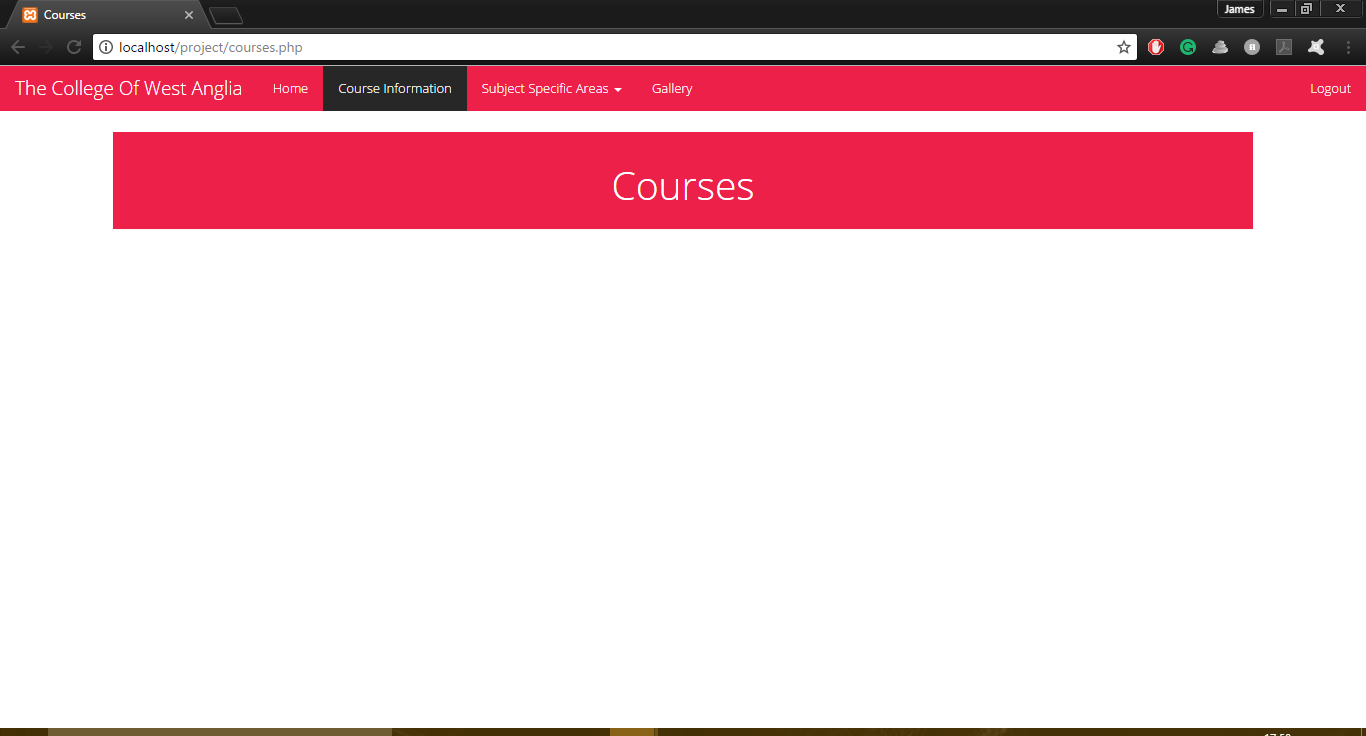


I made the width of each 33% to allow them to be next to each other. I now need to add contact information for the colleges

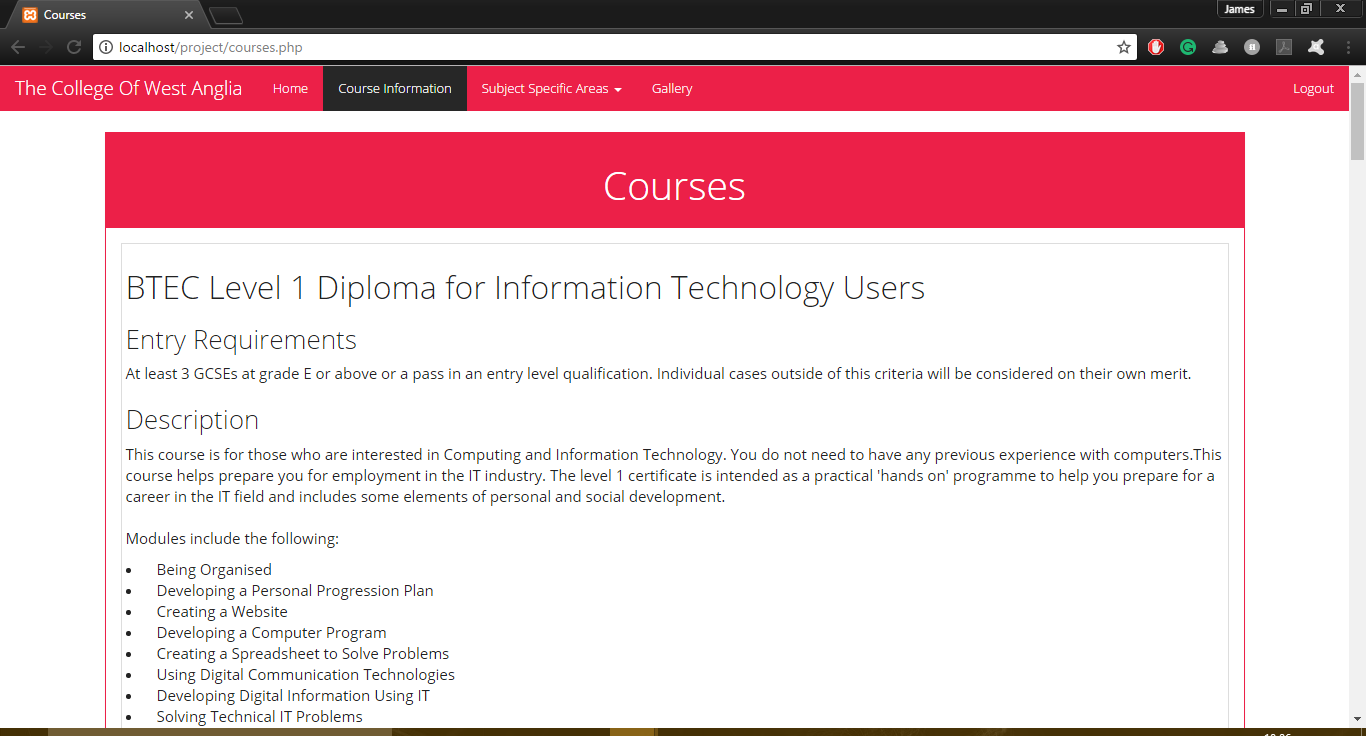


Here I put the contact inform in tables to align with the maps for stylistic purposes.

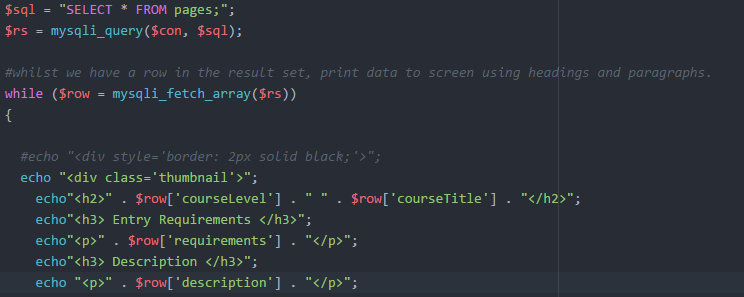
# Content Management System



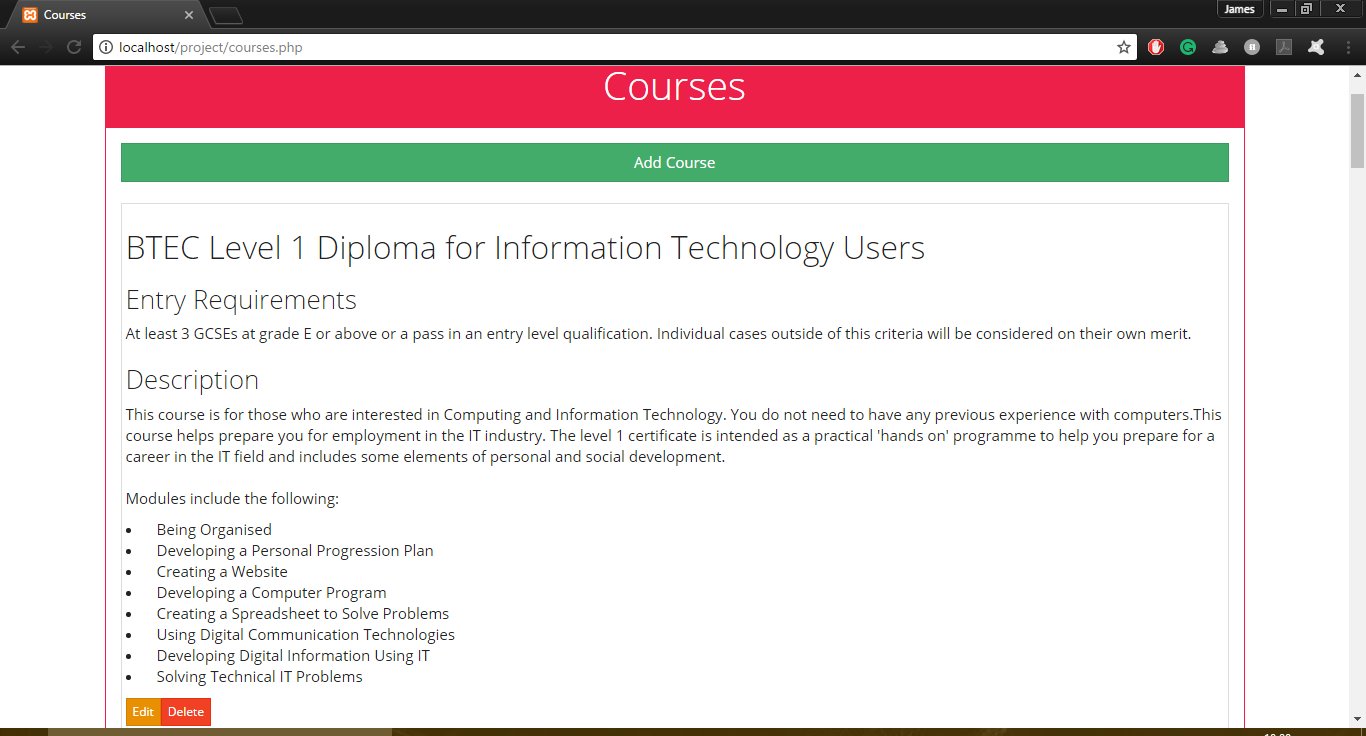
I first started adapting my CMS from a previous example into this project. Here is the basis of the design for the CMS



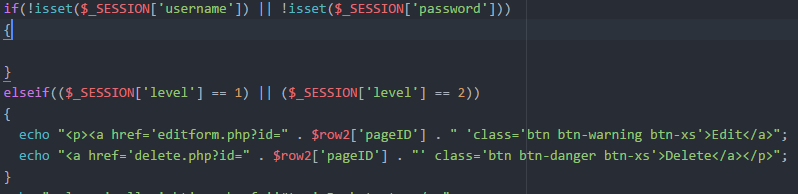
After adding an entry to the database, I later had it display in the pane



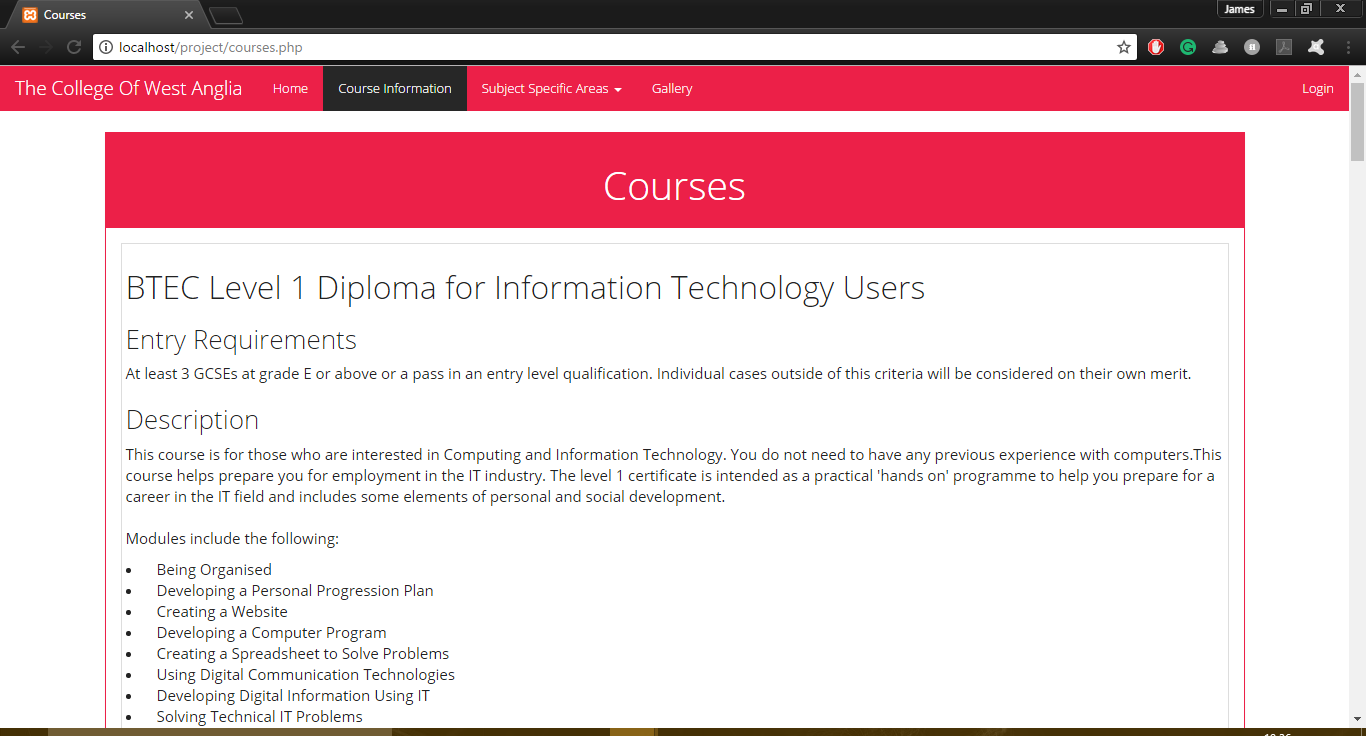
What the variable “sql” would do is select the entries from the “pages” table. Then “rs” would create the query that uses “con” which connects to the database and the “sql” variable. Then the while loops would use the variable row which would use the fetch script that uses the “rs” variable. Then it would display each entry under each heading in the table. This would continue until there are no more entries

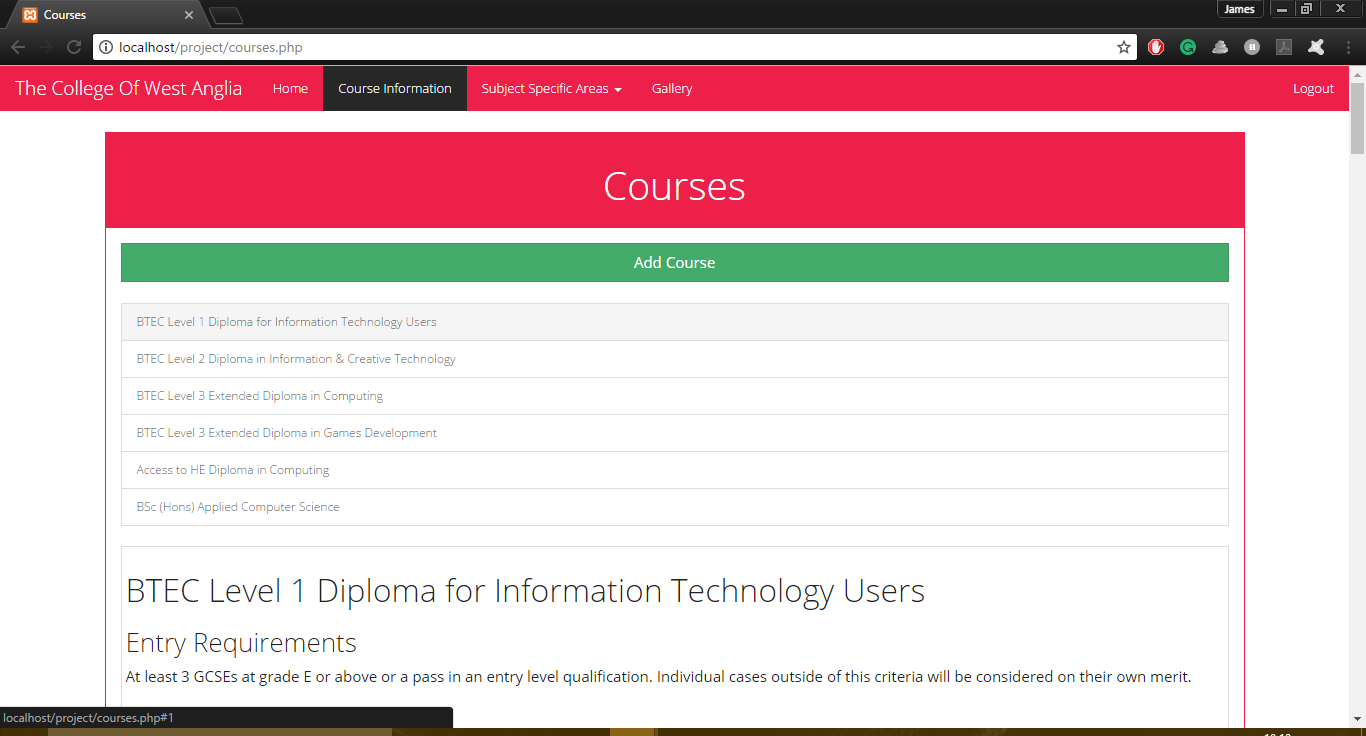


I then added buttons for adding entries, editing entries and deleting entries. Below are examples of the page if the user isn’t logged in as either staff or webmaster.

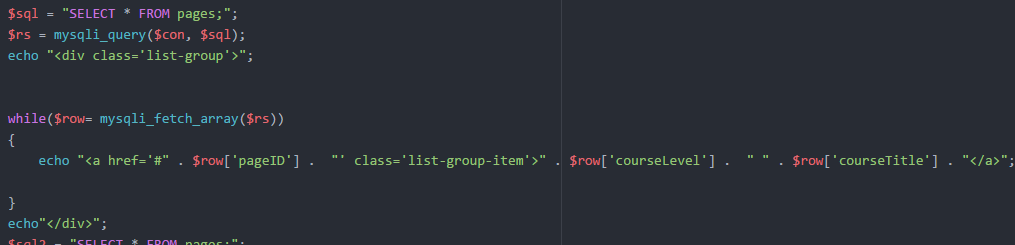


Here is the code to make that works. It checks if the user is logged in and what level they are logged in as. If they’re not logged in it will display nothing, and if they are either logged in it will display the buttons. Both buttons work by using the ID of the entry and putting it in the url when going to the forms, and then the forms would get them.

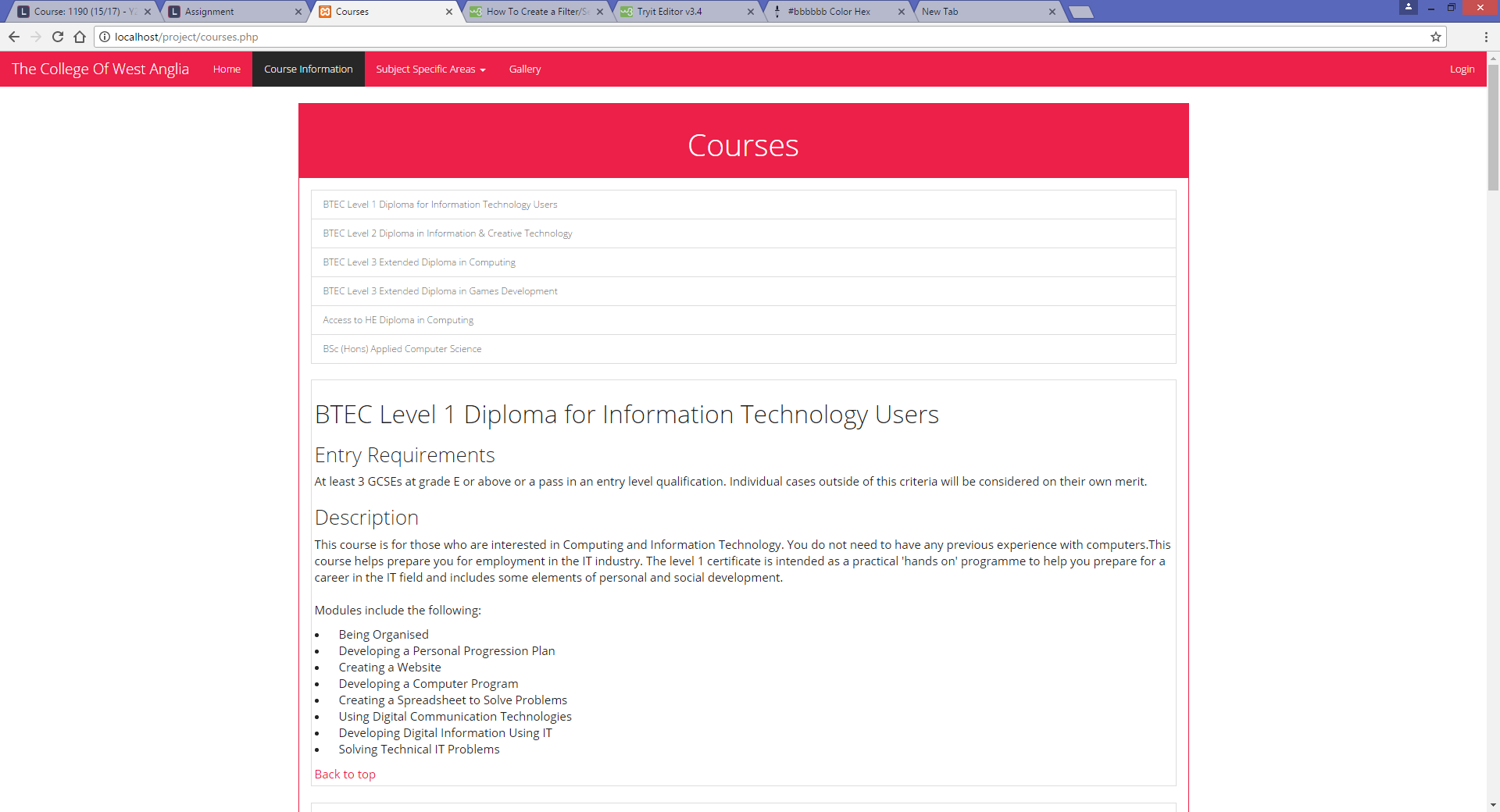




I then implemented a list group system that would jump to the selected entry in the CMS

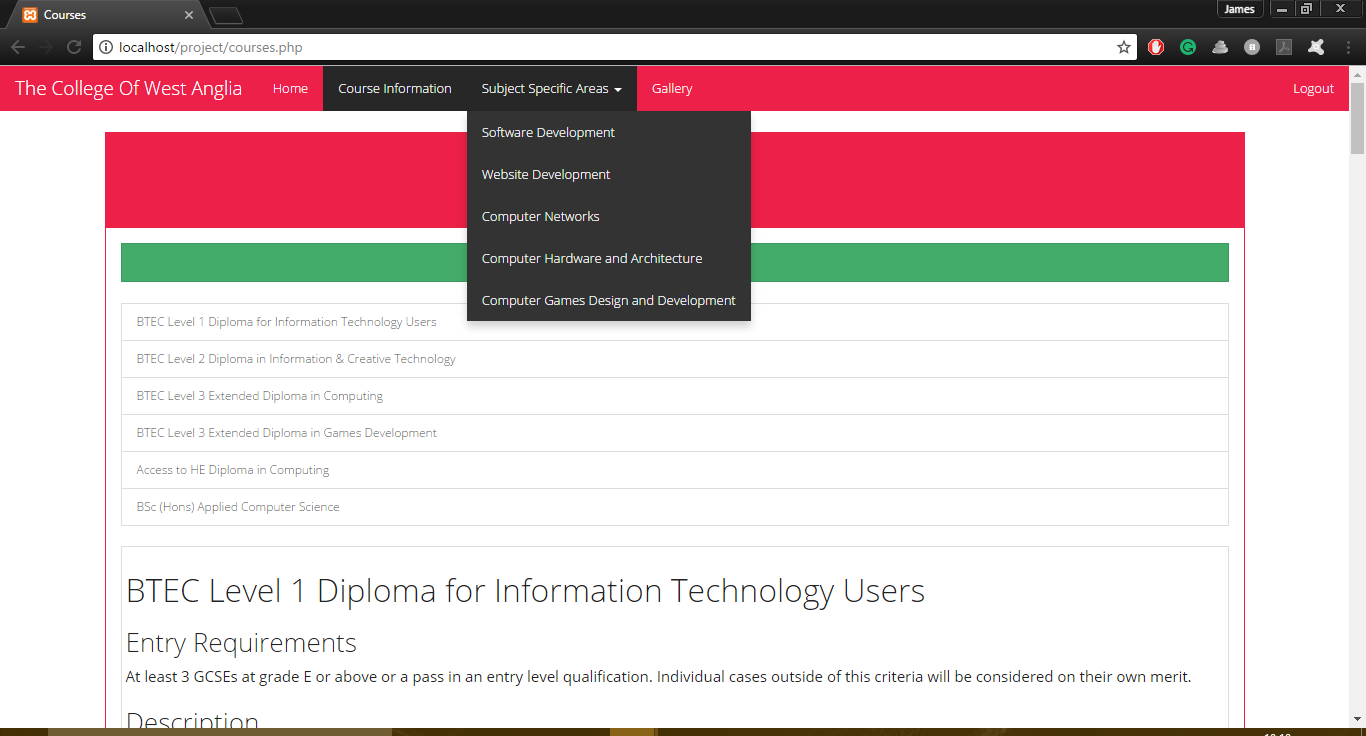


Here is the code to make the list group work. What the variable “sql” would do is select the entries from the “pages” table. Then “rs” would create the query that uses “con” which connects to the database and the “sql” variable. Then the while loops would create one list group item where the variable rows would use the fetch script that uses the “rs” variable. This list group would use the pageID’s as the link to each list, then would display the courseLevel and courseTitle from the table in the list group item. This loop would then go through each entry until it’s got through each one. I had to alter the previous code to make the page display the entries by changing the variable names by adding a “2” to the end of them as this came first in the code and they couldn’t share the same variable names

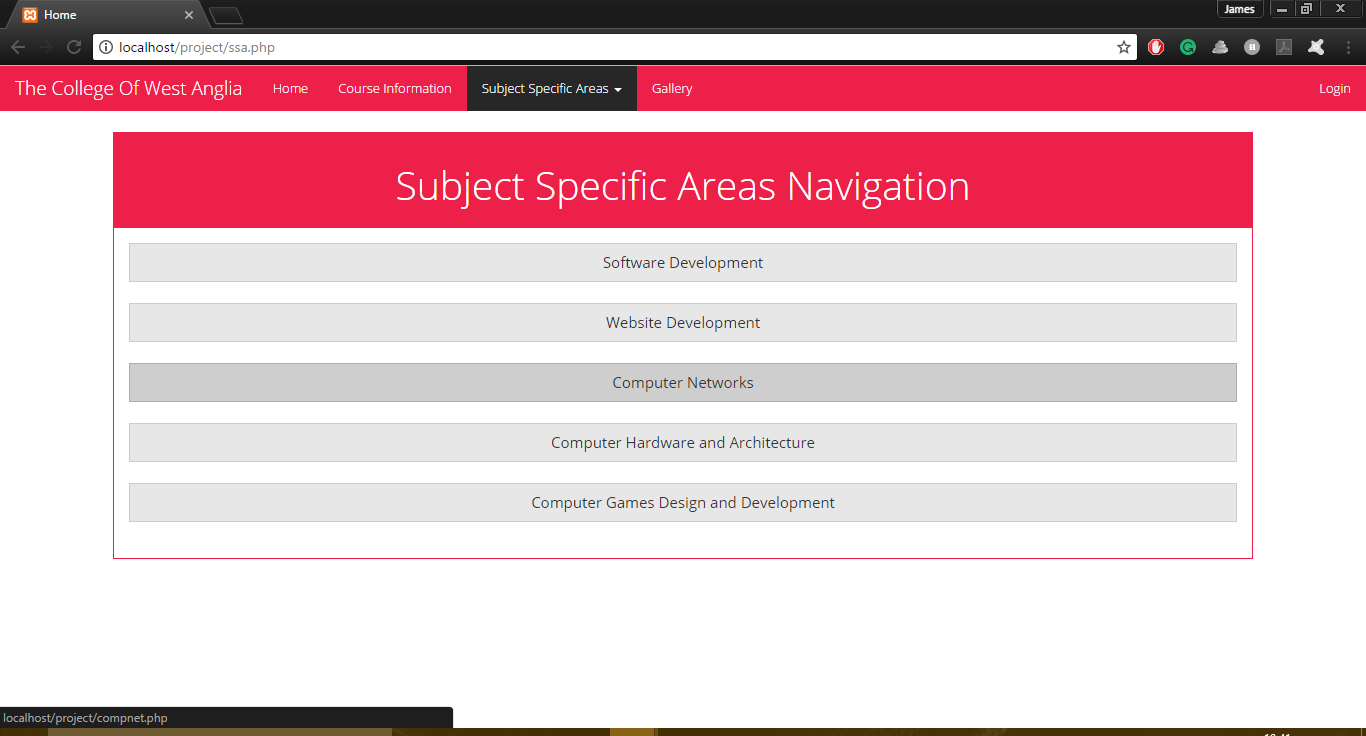


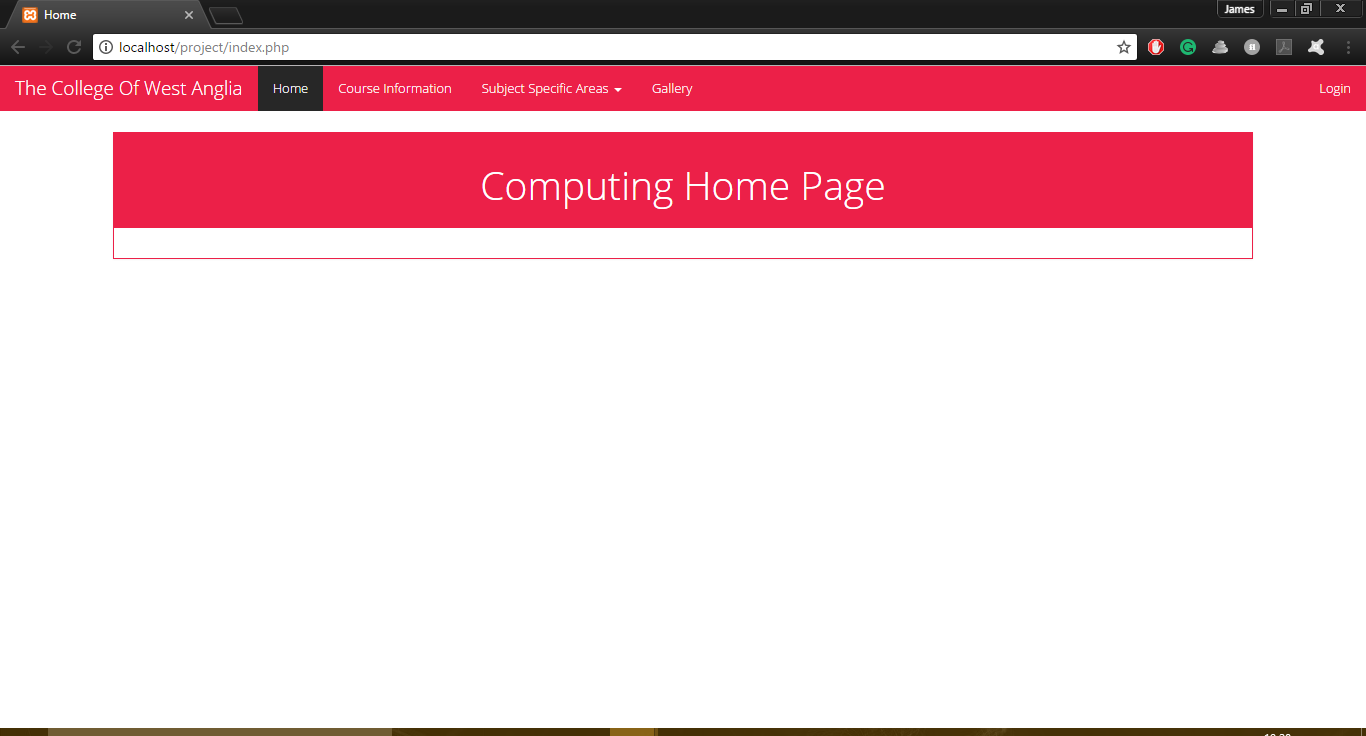
Here is where I added a back to top link to each entry. When clicked, this will put the user back to the top of the page.

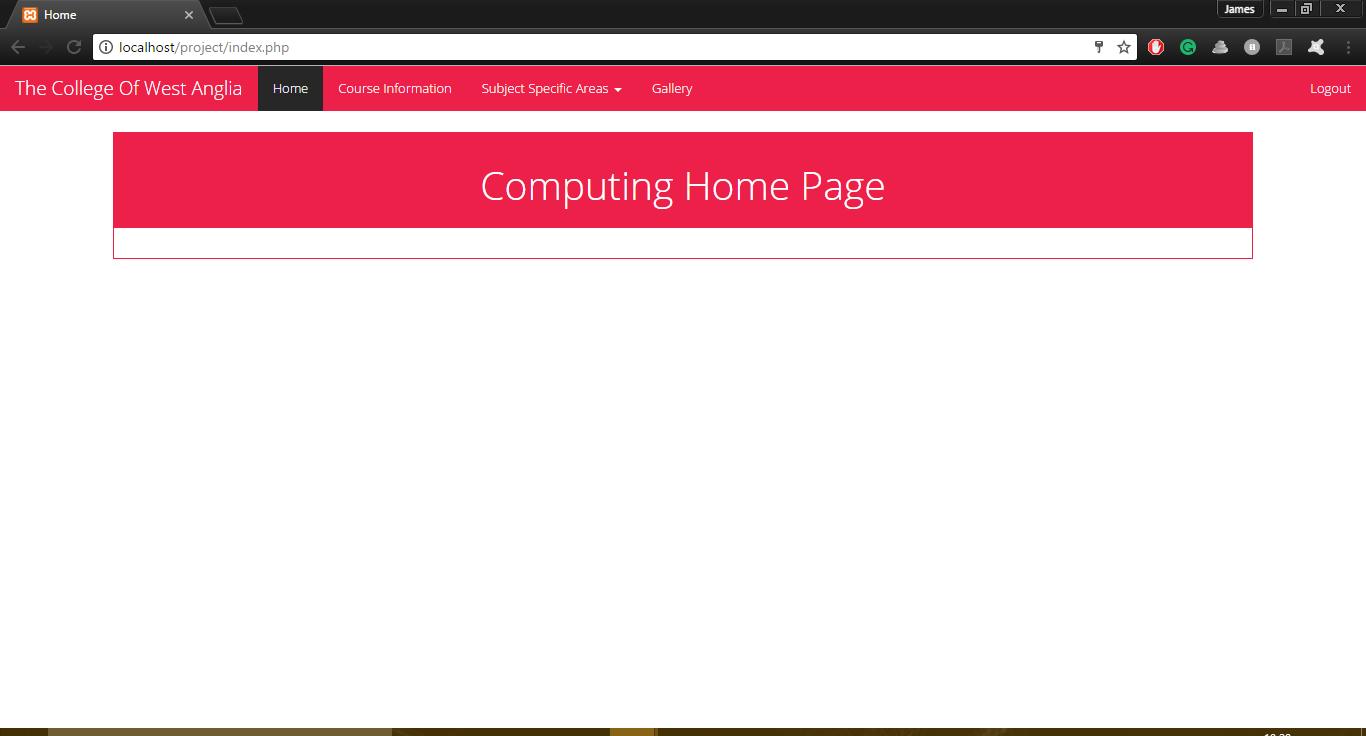
# Nav Bar



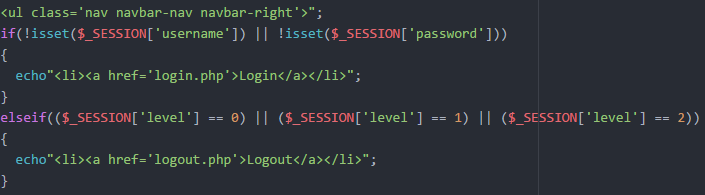
My nav bar has links to each page listed in the brief. Using JavaScript, I created a dropdown list for the Subject Specific Areas pages, and if the JavaScript doesn’t work it will direct to a page with links to each subject page as shown below



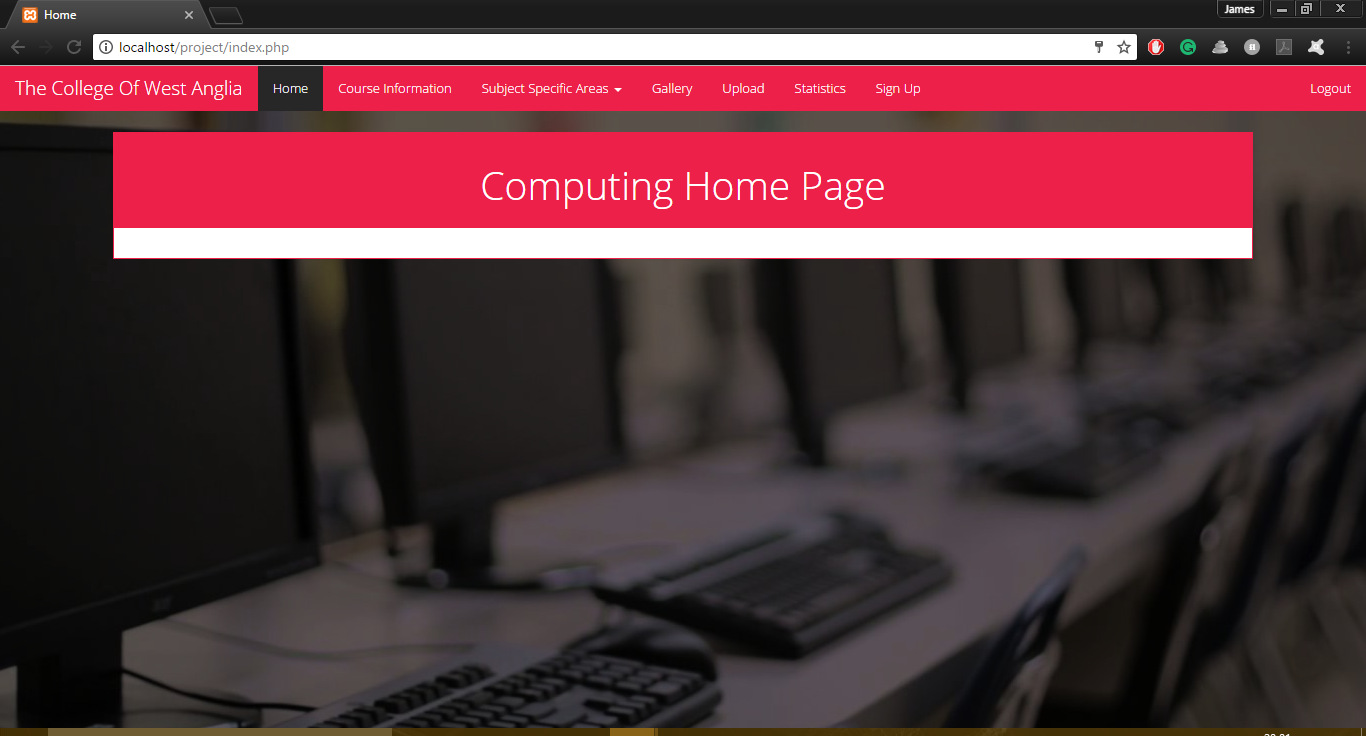




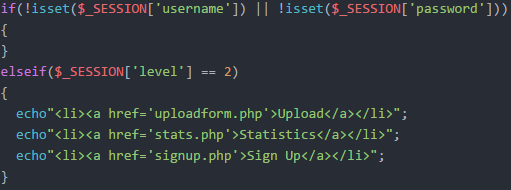
When a user isn’t logged in the link to the far right will display a login and will direct users to a login form when they click it. If they are logged in it will display logout, and will log them out and return them to the home page if they click it.



This is the code to get this to work. If the user isn’t logged in it will displayed the Login link and if they are it will display the Logout link.

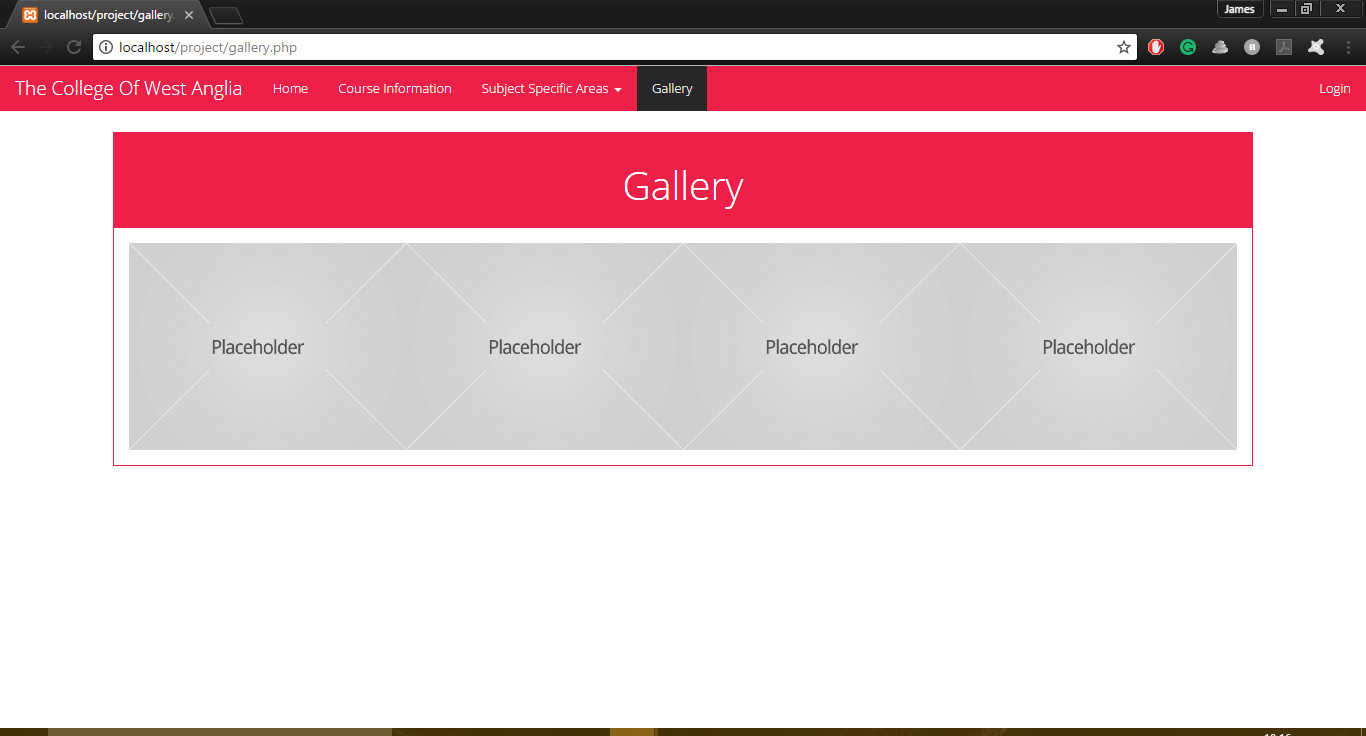


This is the nav bar when a webmaster is logged in. This allows webmaster’s to access their own specific pages

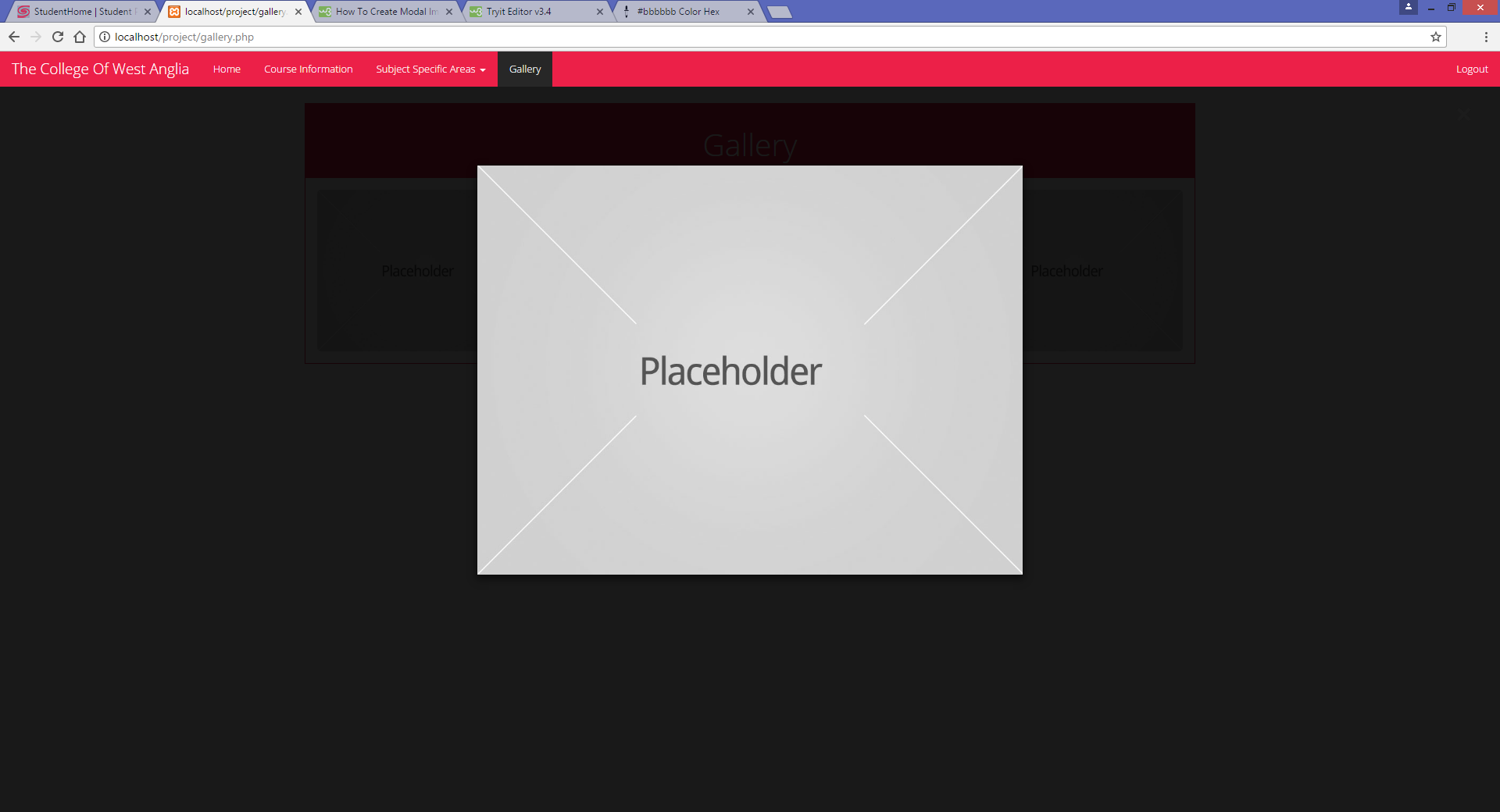


This is the code for this. Here it checks if someone is logged in. If they are it won’t display nothing. It then checks the user’s tier; if they’re not a webmaster tier it won’t display anything, it if they are it will display the three links.

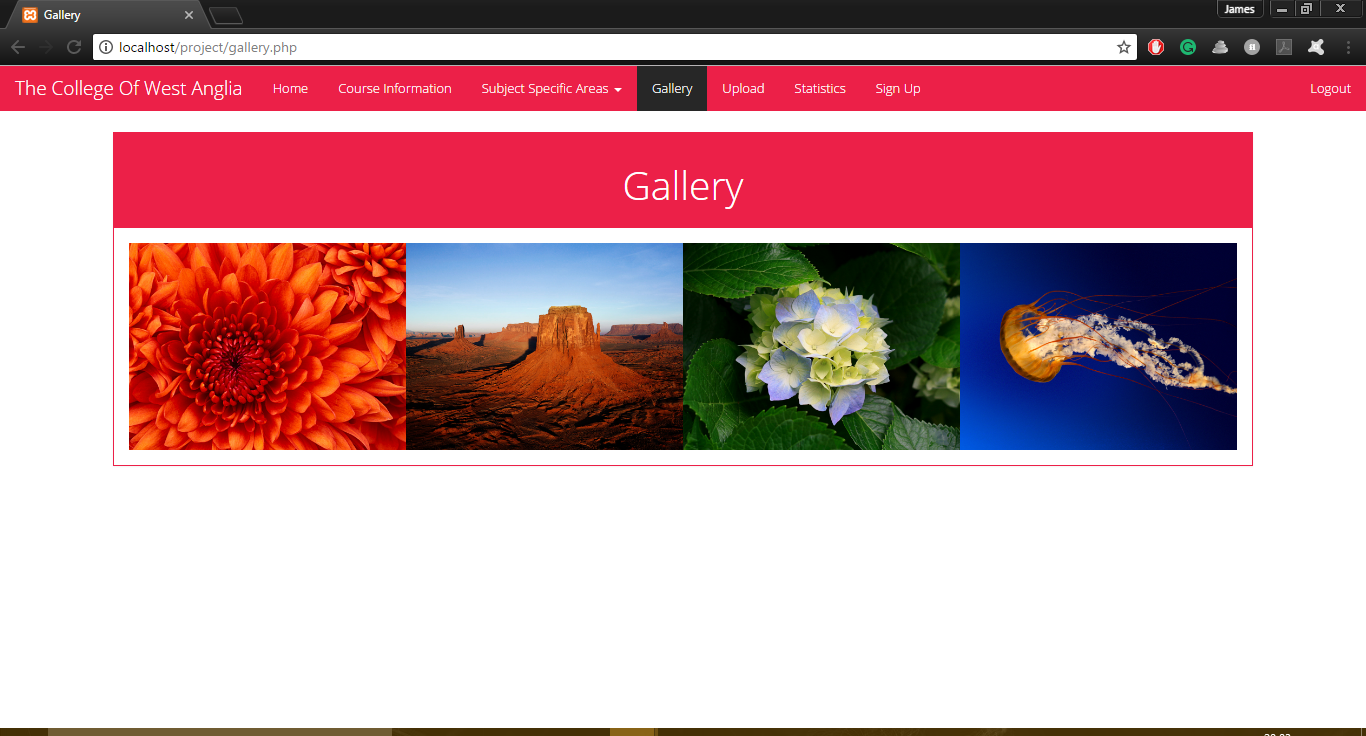
# Gallery

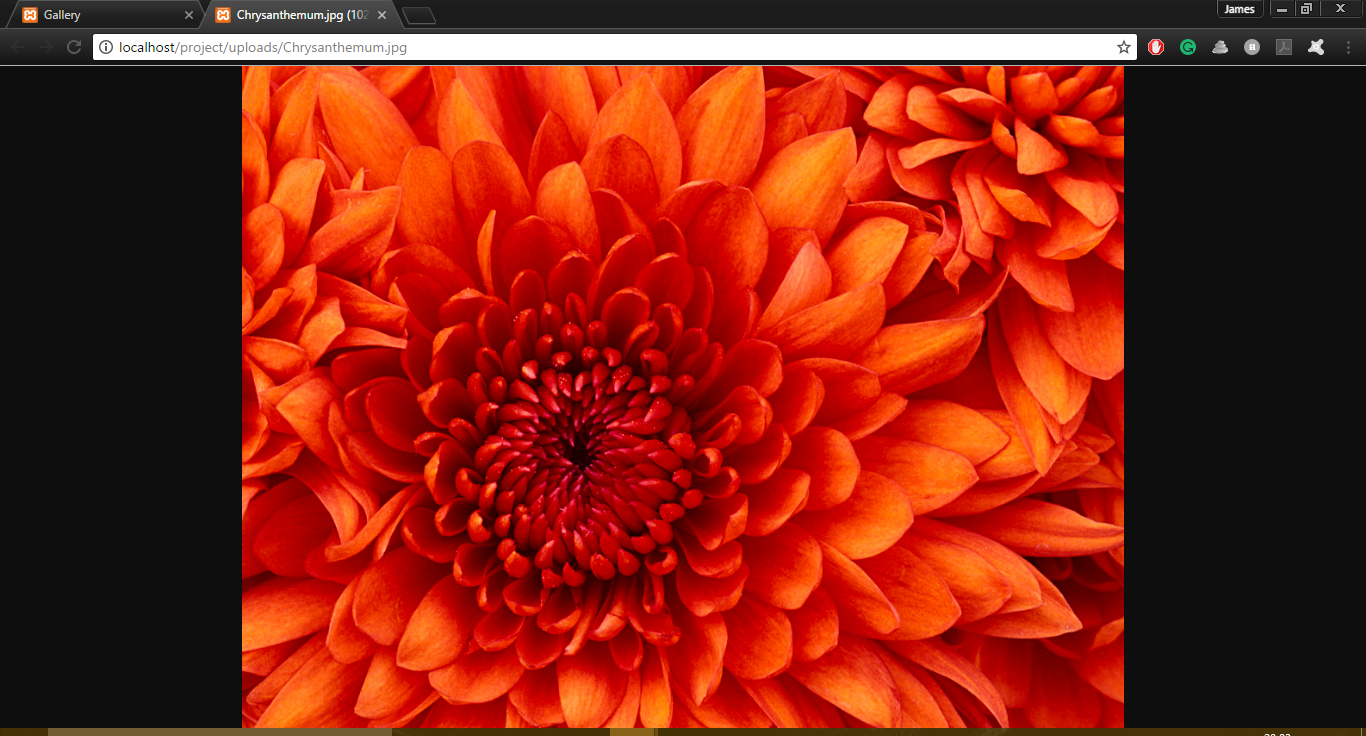


Here is where the gallery of images will be displayed. For now, there are 4 identical placeholder images in the uploads folder that the page is pulling them from. These images will be replaced when finalised and the gallery will be a grid of images pulled from the folder.



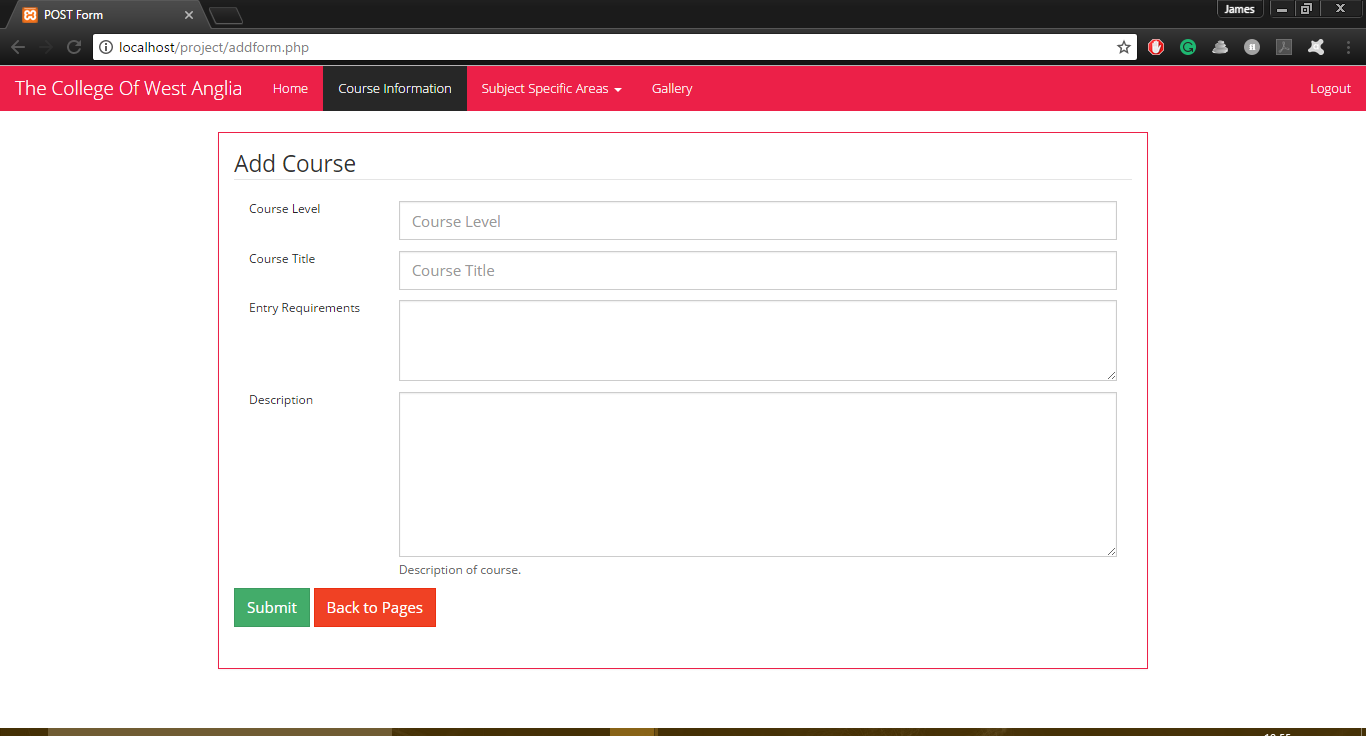
Here I’ve got the images to worst as a modal system. When clicking on the image it will zoom in and make it the main focus of the page. The issue I have now is that it only works on the first image





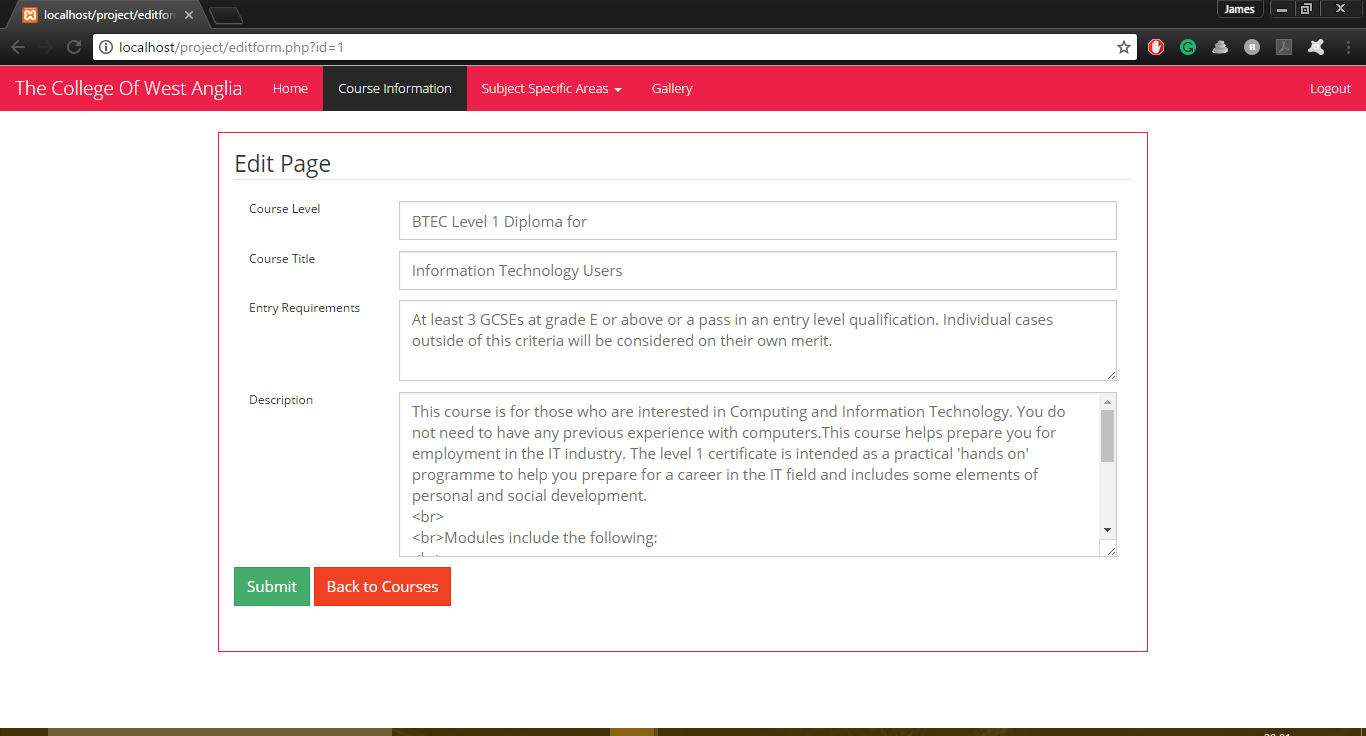
The original idea I had proved to be too difficult to accomplish in the timeframe I’ve been given so I changed the idea and made it a simple image gallery that will open the image in a new page

# Add Form



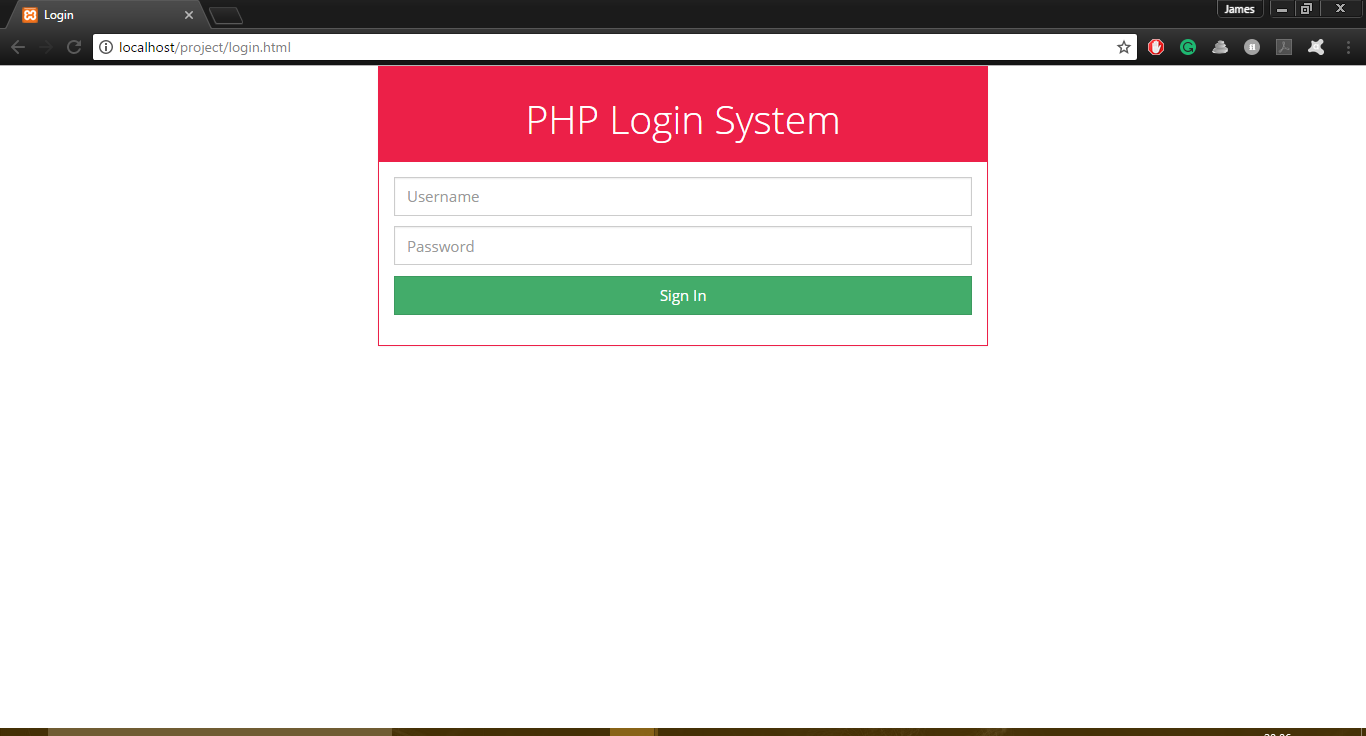
Here is the form the staff and webmasters will be able to access to add a new course into the database. Upon hitting submit it will add the information in each field to the database and will then be displayed on the courses page.

# Edit Form

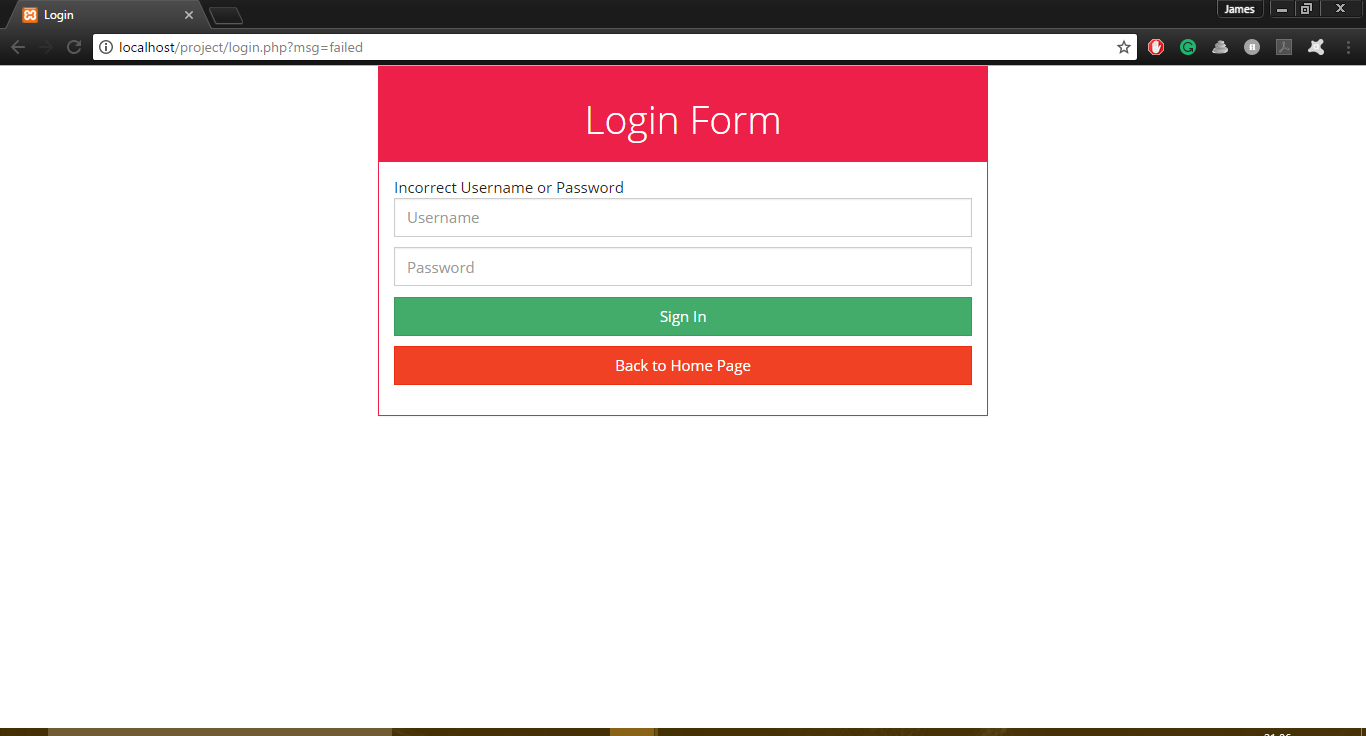


Here is the edit form. It looks and works similarly to the add form. It pulls the information of the selected entry and places them in each field, and upon hitting submit it changes the entry in the database to what is in the altered field(s).

# Sign In Form

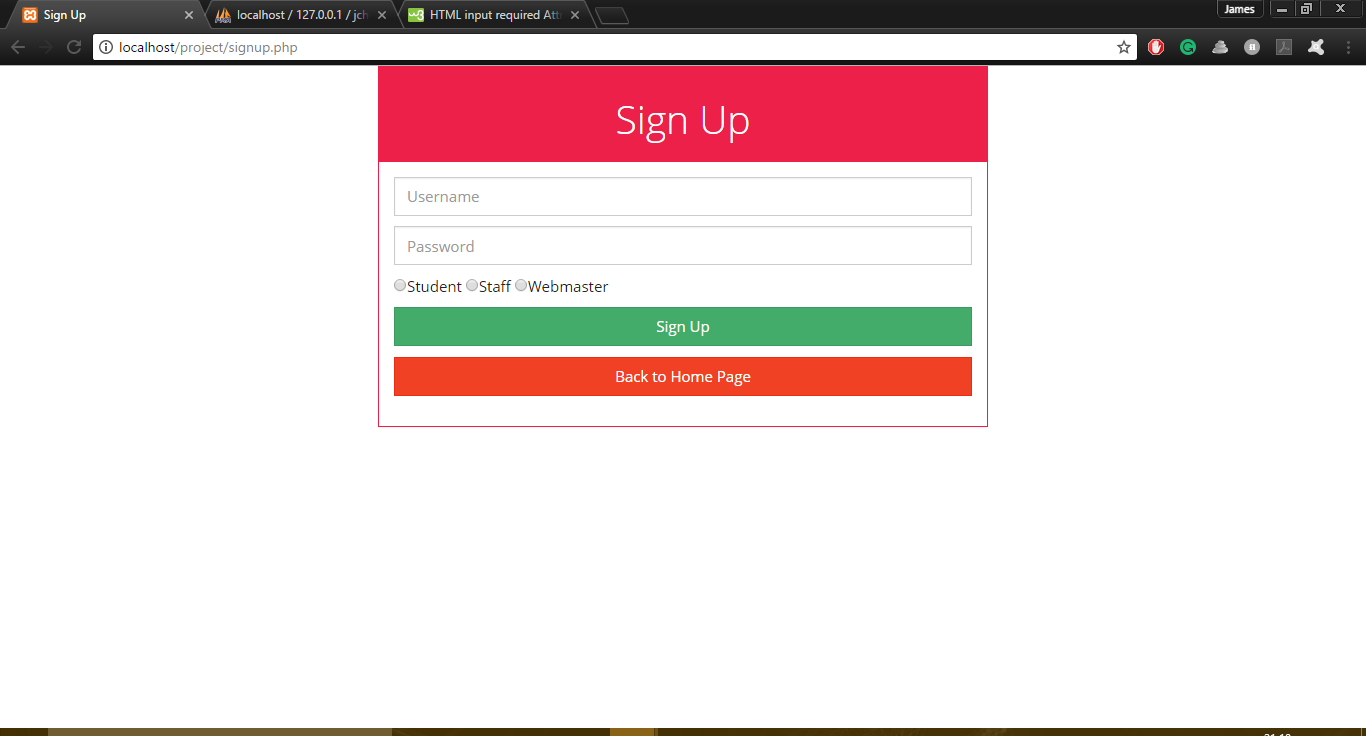


This is the sign in form. Here users can enter their details and if they are valid they will be directed to the home page. I have not yet made anything for non-existent users; the plan is to include a “username or password is invalid message on the page

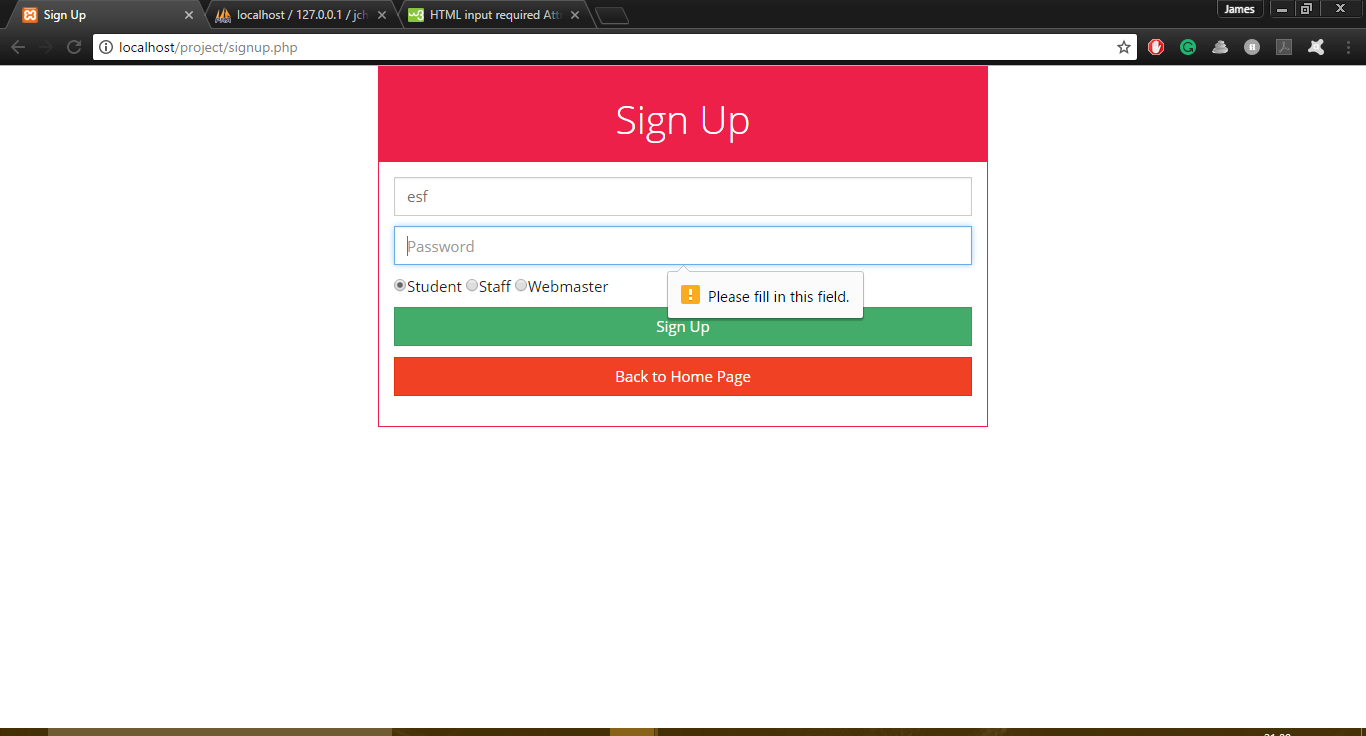


Now when someone enters an incorrect username or password it will display a line of text informing them

# Sign Up Form



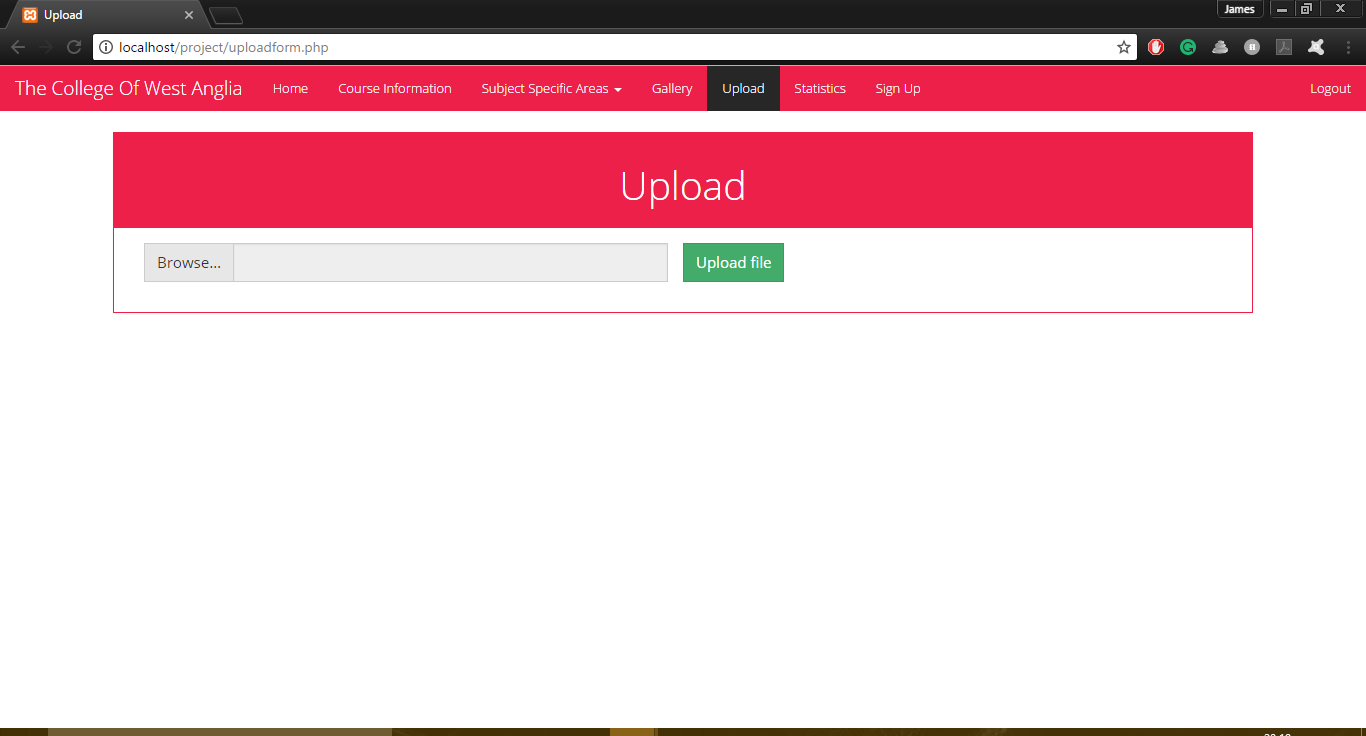
Here is the form the webmasters use to sign up a new user. The issue I have with this is that the form still works when a field isn’t filled out



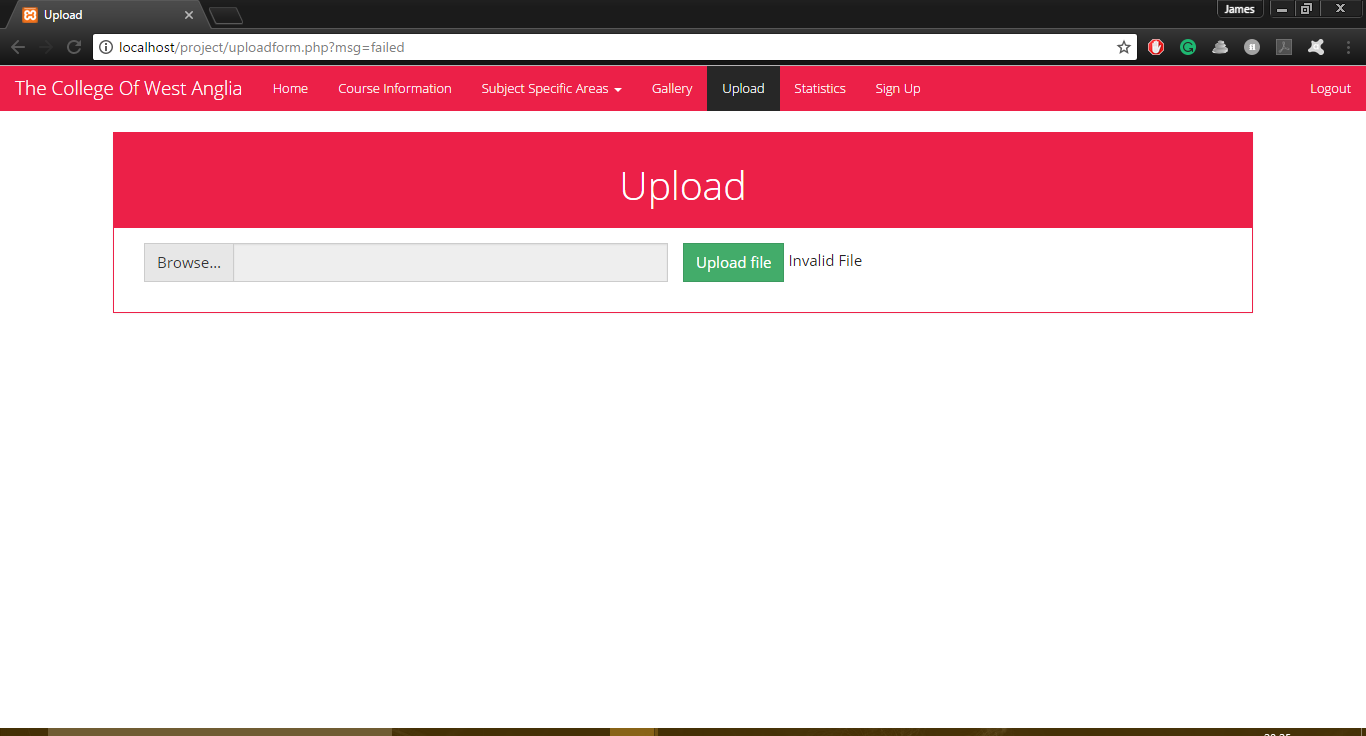
I fixed this issue by adding the required tag inside the input script. So this will appear when they try continue without filling all fields.

An issue I had was the hashing of passwords. It would not login as it wouldn’t recognise the hashed password

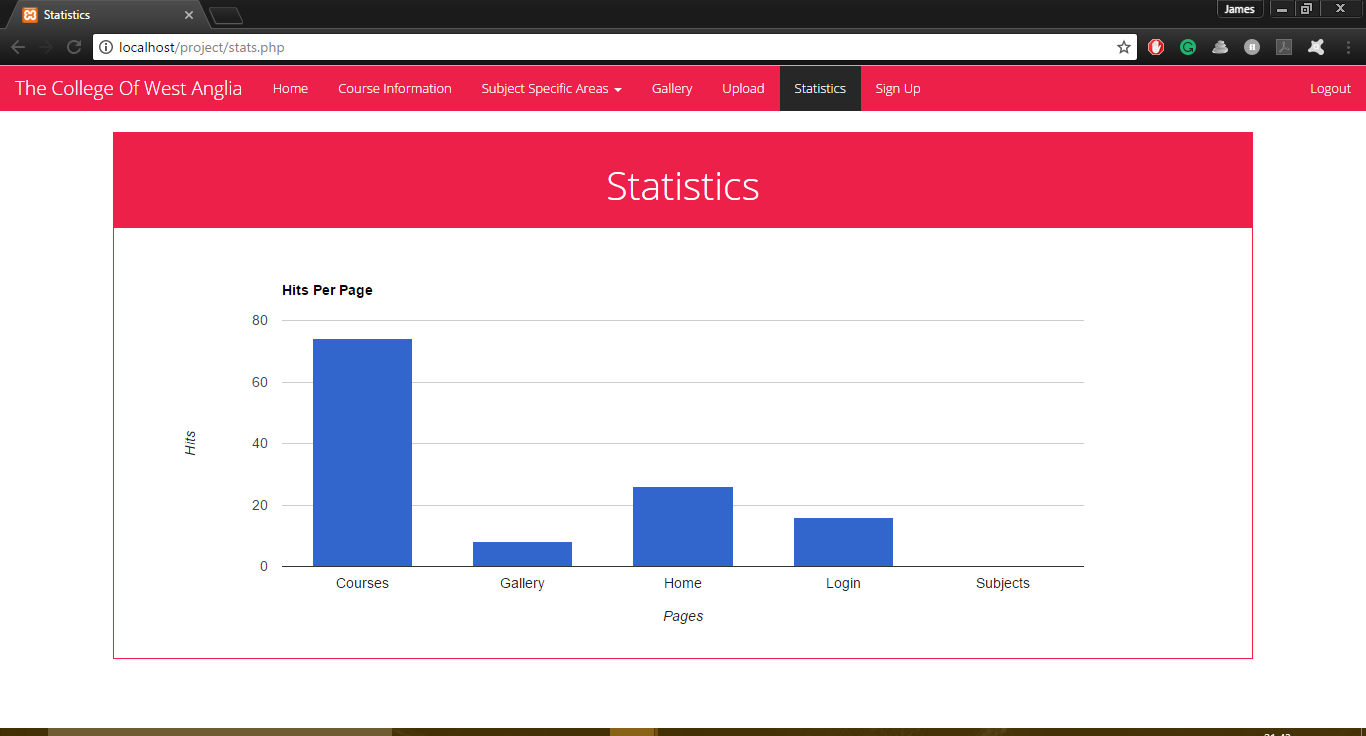
# Upload Form



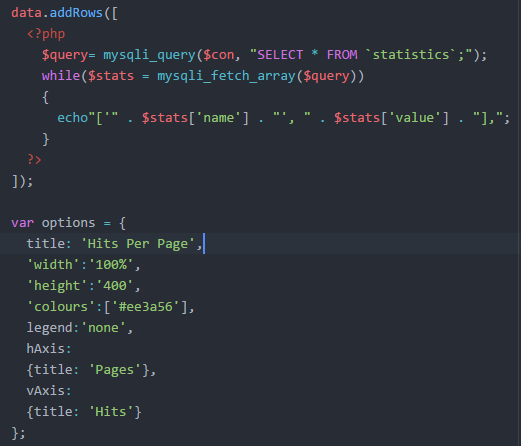
This is the upload form. When the browse button is clicked, it will bring up the explorer for the user to choose an image. When it’s chosen, it will display the file name in the box next to it. If the user has chosen a valid file it will upload to the folder and be displayed in the gallery. If not it will display an error saying it’s invalid.



# Statistics



Here is the page for the statistics. This shows the amount of visits to each of the different pages available to anyone who views the page. I used the example rom google charts (<https://jsfiddle.net/api/post/library/pure/>) and altered it to read from the table in the database



Here is the code I used to make it work. The variable “query” selects all entries in the “statistics” page using the “con” variable to connect to it. Then the while loop creates a new variable called “stats” which use the “query” variable to get the entries from the table. They are then read into the rows of the chart. The loop will continue until it has gone through every entry. I then styled the table to show the title and label the axis accordingly.