Table of Contents

Table of Figures

General Introduction

Overview:

Online course registration system is a web application specially designed for a University. This system provides a number of functionalities pertaining to COURSE REGISTRATION for the students as well as faculty members. Students can login, view, register, drop courses, and start studying directly online. Whereas the staff members or the teachers can login to the admin page where they can view the number of students registered for their course, add a new course they are planning to teach, drop a course they are planning to not teach anymore etc.

Problem statement:

After having continues research in the ongoing teaching and learning process in the university, I found the need of this system and strongly recommend it, because it will help a lot of students to now theirs courses better before and after registering it.

Objectives:

Project is aimed to develop a system to manage the course material online, provide an easy access to the information about the courses, get the admin the capability to add, delete, update a specific course.

Methodology:

An Online Student Course Registration System has been developed to simplify the current manual procedure. This system has been developed using PHP and MySQL. The front-end is designed using PHP, HTML, CSS, and JAVA Script and back-end is designed and managed through MySQL, and PHP

Limitations:

The limitation is that in the admin website, the page added in the add page web page isn't pre-styled as intended to be, so if the admin what a styled pages in a specific course he/she needed to style it him/herself then enter it to the web page, and that will make a little of complication with some admins.

Outline:

Chapter 1 is describing the previous work and existing systems similar to the application developed, compare between them, identify the advantages of each system, and propose a solution that have advantages for these systems.

Chapter 2 include brief definitions about what is system analysis and design, data modeling, data dictionary, and business rules

Chapter 3 includes definitions about the programming tools, programming languages, and frame works used to develop this application. And it is the project implementation which covers the overall planning of the system, and the sequence of events that lead to the creation of a new system.

Chapter 4 sheds light on the project results.

Chapter 1: Previous Works

1.1. Existing Systems:

Alison is one of the existing systems which is to the application to be developed. Alison is an online courses website which offers all people the opportunity to register in a course and starting learning directly from the website without being registered in any college or university. Alison offers over 2000 free online courses across nine distinct categories, but if you want to get a diploma or certificate to prove that you had studied the course, you need to pay an extra cost to get them. The certificate level courses require two to three hours of study while the more rigorous diploma level courses require ten to fifteen hours of study. There is no time limit for completing a course. According to the Alison website, Alison is not currently accredited by any external body, and does not intend to be accredited at any time in the future, while delivering courses with up-to-date international pedagogical standards. [Wikipedia]

Udacity began as an experiment in online learning, when Stanford instructors Sebastian Thrun and Peter Norvig elected to offer their "Introduction to Artificial Intelligence" course online to anyone, for free. Over 160,000 students in more than 190 countries enrolled. The potential to educate at a global scale was awe-inspiring, and Udacity was founded to pursue a mission to democratize education. It would take several years of intensive iteration and experimentation to clarify our focus on career advancement through mastery of in-demand skills, but today, Udacity proudly offers aspiring learners across the globe the opportunity to participate in—and contribute to—some of the most exciting and innovative fields in the world. Each course consists of several units comprising video lectures with closed captioning, in conjunction with integrated quizzes to help students understand concepts and reinforce ideas, as well as follow-up homework, which promotes a "learn by doing" model. Programming classes use the Python language; programming assignments are graded by automated grading programs on the Udacity servers. Colorado State University's Global Campus began offering transfer credit for the introductory computer science course (CS101) for Udacity students that take the final examination through a secure testing facility. In 2015, Udacity started the Nanodegree program, it is a paid credential program. Udacity also offers Nanodegree plus, which is a bit more expensive, but guarantees a job, if they fail to provide a job, the course fee is returned, although it plans to cancel the program. [Wikipedia]

1.2. Comparative Study

Both Alison and Udacity is an online course registration website that can offer an individual the ability to register, drop, and study a specific course of his/her choice. And both are not directly connected a specific university which effect their credibility. Alison offers a free online course with any certificate but if you want a certificate you should pay a specific amount to get your diploma or certificate, while in Udacity, the individual must buy a course before being able to register the course, and if the individual wants a certificate, he/she must pay extra cost.

1.3. Proposed Solution

The online course registration should be directly connected to the university or collage, so the students can register the same course from the university in the website which will give him all the materials he need to study. And if this website is directly connected to a university, it will have the credibility it need to get the trust of the students.

Chapter 2: System Analysis and design

2.1. Description of the system

2.1.1. Requirements and Business Rules

On signup, the user should enter a unique username, a valid email, and a password. On login, the user will enter his username, and their password to login to the website. The user can register a max of seven course together, and if the user want to change a course, he/she can drop that course and register any other course at any time.

The admin can enter the admin webpage to drop and add new students, the admin can also can add, update, and delete any course on the website. To add a course the admin should enter course Id, course name, course category, course description, card text, number of pages, and for each page there is a title, content, and page number. Course Id and Course name are unique.

2.1.2. Scope

The main scope of this project is to maintain a students info and to let the students register any course they need in the university.

2.1.3. Roles

The Admin is responsible for the adding new courses, deleting courses that the university stop teaching them, or updating a course that needed an update. And the admin also is responsible for the students info, how many courses they can register.

2.2. Data Model Description

2.2.1. Data Dictionary

A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them. Often a data dictionary is a centralized metadata repository.
A first step in analyzing a system of interactive objects is to identify each one and its relationship to other objects. This process is called data modeling and results in a picture of object relationships. After each data object or item is given a descriptive name, its relationship is described, or it becomes part of some structure that implicitly describes relationship. The type of data, such as text or image or binary value, is described, possible predefined default values are listed and a brief textual description is provided. This data collection can be organized for reference into a book called a data dictionary.

[1]

Attribute	Data Type	Description
AdminId	Big int	Not null, unique
AdminName	VarChar(30)	Not null
password	Text	Not null
idUsers	Big int	Not null, unique
uidUsers	VarChar(30)	Not null
emailUsers	VarChar(30)	Not null
pwdUsers	Text	Not null
courseId	Big int	Not null, unique
CourseName	VarChar(30)	Not null
category	VarChar(30)	
Description	text	
cardtext	text	
pages	Big int	
pagenb	Big int	Not null
title	VarChar(30)	
content	text	Not null

2.2.2. Conceptual data model (CDM)

A conceptual data model is the most abstract-level data model or summary-level data model. Information specific to the platform and other implementation information such as interface definition or procedures are eliminated from this data model. A conceptual data model is useful due to its simplicity. It is often used for communicating ideas and in strategic data projects.

A conceptual data model is also known as a conceptual schema.[2]

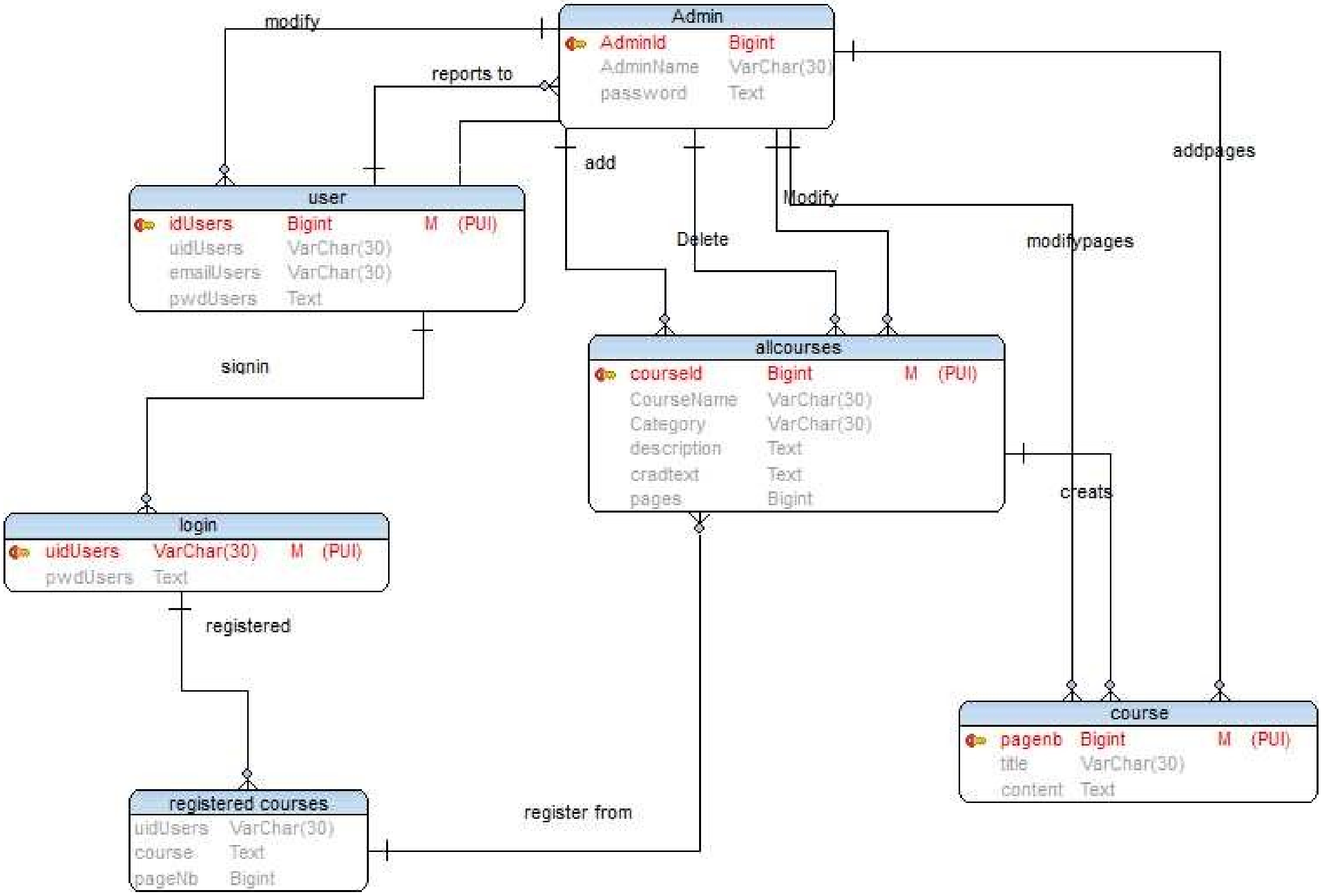


Figure 1: conceptual data model

2.2.3. Physical Data model (PDM)

A physical data model defines all of the logical database components and services that are required to build a database or can be the layout of an existing database.

A physical data model consists of the table's structure, column names and values, foreign and primary keys and the relationships among the tables. [3]

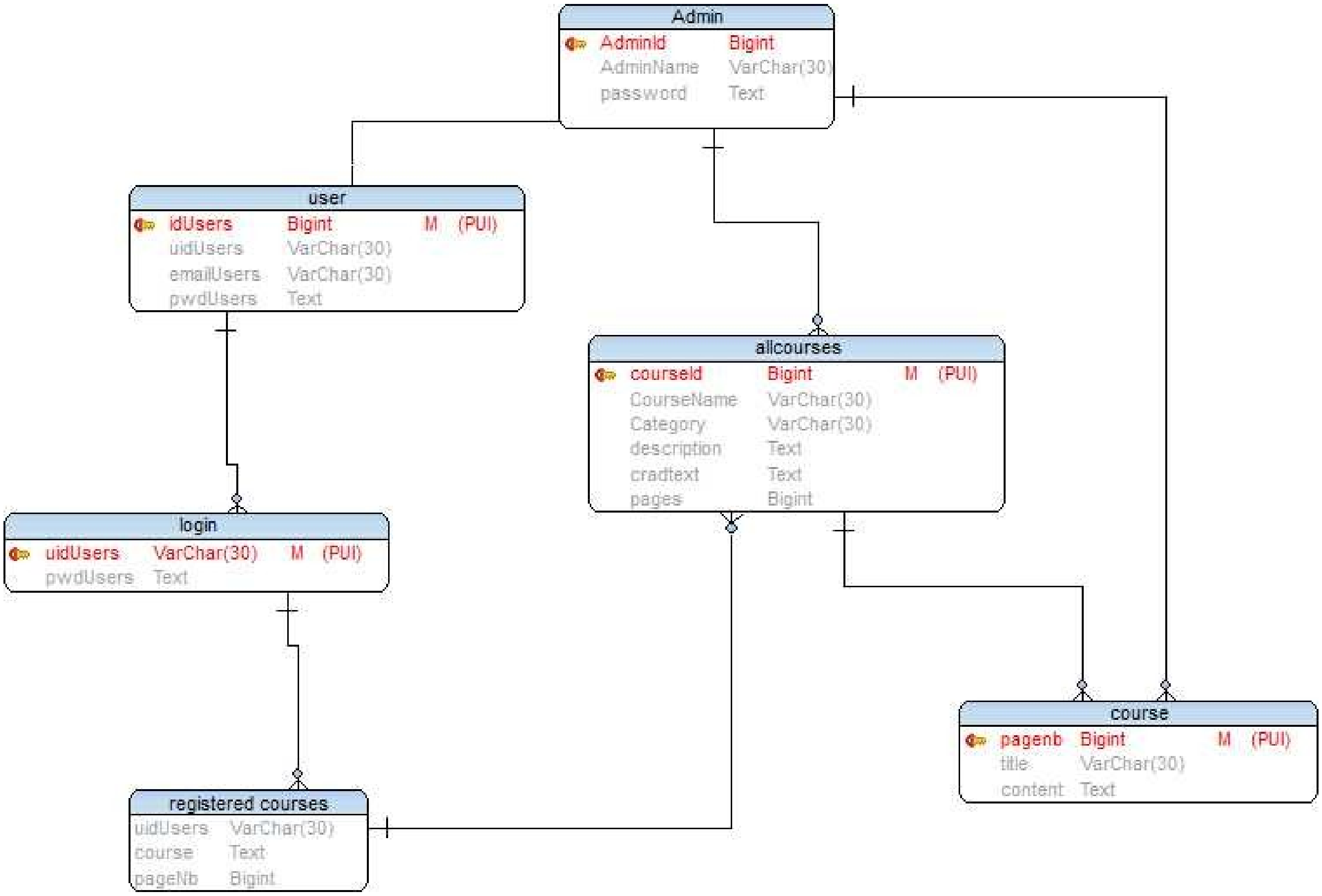


Figure 2: physical data model

Chapter 3: Implementation

3.1. Platforms and Development Tools

3.1.1. PHP

PHP is a general-purpose scripting language that is especially suited to web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994; the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of a HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response.

3.1.2. HTML

First developed by Tim Berners-Lee in 1990, HTML is short for Hypertext Markup Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.

HTML code ensures the proper formatting of text and images for your Internet browser. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance). [4]

3.1.3. CSS

This definition explains the meaning of CSS (cascading style sheets) and how using them with HTML pages is a user interface (UI) development best practice that complies with the separation of concerns design pattern.

CSS is the standard and preferred mechanism for formatting HTML pages. Conforming with the separation of concerns design pattern and best practice, cascading style sheets provide a central location in which information about what various fonts, foreground colors, background colors, italicization and emphasization should be applied to various HTML elements within a webpage. Cascading style sheets can also control how various parts of a page, such as the header, footer, body, article content, sections and asides, are laid out on the page. This is extremely helpful when content must be laid out in a dramatically different fashion depending upon whether it is being viewed on a desktop, tablet or a smartphone.[5]

3.1.4. Java Script

Javascript (JS) is a scripting languages, primarily used on the Web. It is used to enhance HTML pages and is commonly found embedded in HTML code. JavaScript is an interpreted language. Thus, it doesn't need to be compiled. JavaScript renders web pages in an interactive and dynamic fashion. This allowing the pages to react to events, exhibit special effects, accept variable text, validate data, create cookies, detect a user’s browser, etc.

HTML pages are fine for displaying static content, e.g. a simple image or text. However, most pages nowadays are rarely static. Many of today’s pages have menus, forms, slideshows and even images that provide user interaction. Javascript is the language employed by web developers to provide such interaction. Since JavaScript works with HTML pages, a developer needs to know HTML to harness this scripting language’s full potential. While there are other languages that can be used for scripting on the Web, in practice it is essentially all Javascript.[6]

3.1.5. Bootstrap

Bootstrap is a collection of reusable pieces of code written in HTML, CSS, and JavaScript. However, it is a free and open-source front end web development framework to help you create responsive websites and web applications.

Also, it saves you from having to write lengthy code, since it is intended to create responsive websites. It allows a website to work optimally on varied screen sizes from small screen size as a smartphone to a big screen size of a personal computer. Basically, it sets you free from the burden of building a device-specific site and keeping Bootstrap cheat sheets handy makes it even simpler. [7]

3.1.6. Xampp

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

3.2. System Manual

3.2.1. Admin System Manual

3.2.1.1. admin login page

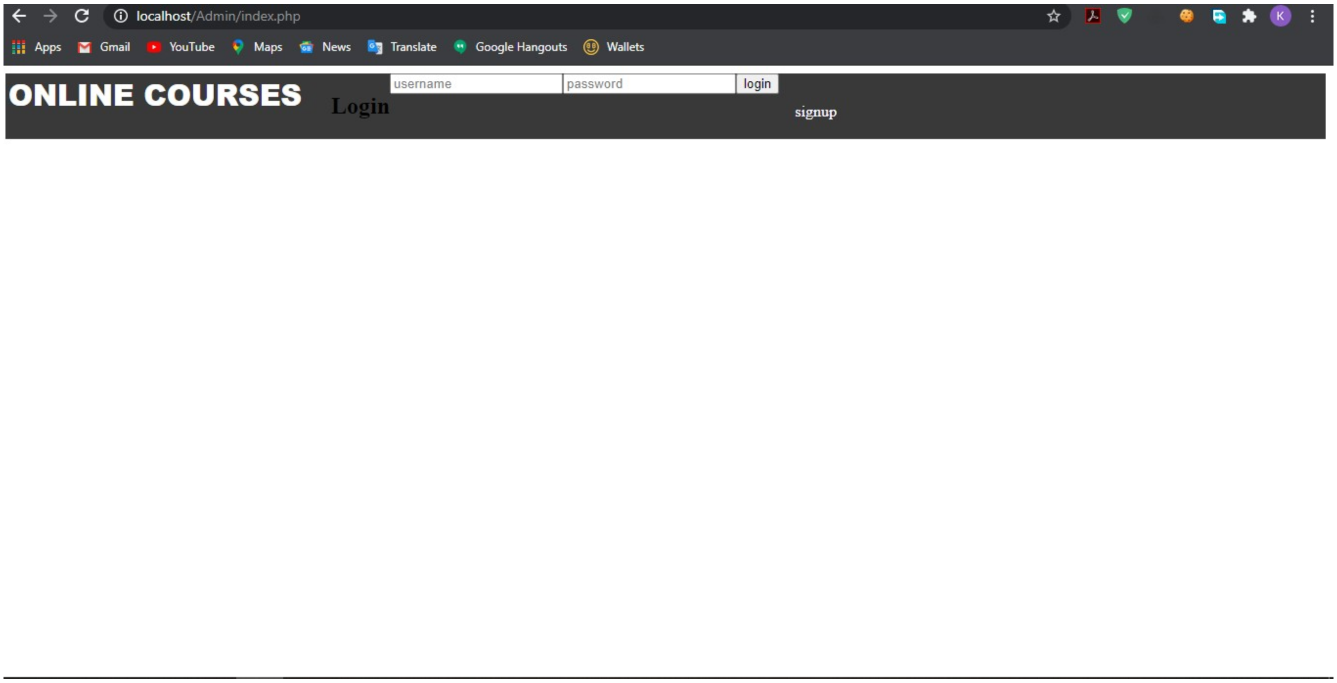


Figure 3: admin login page

In this web page, The admin can login to the website to do his role in the website like add courses and delete the courses

3.2.1.2. admin main page

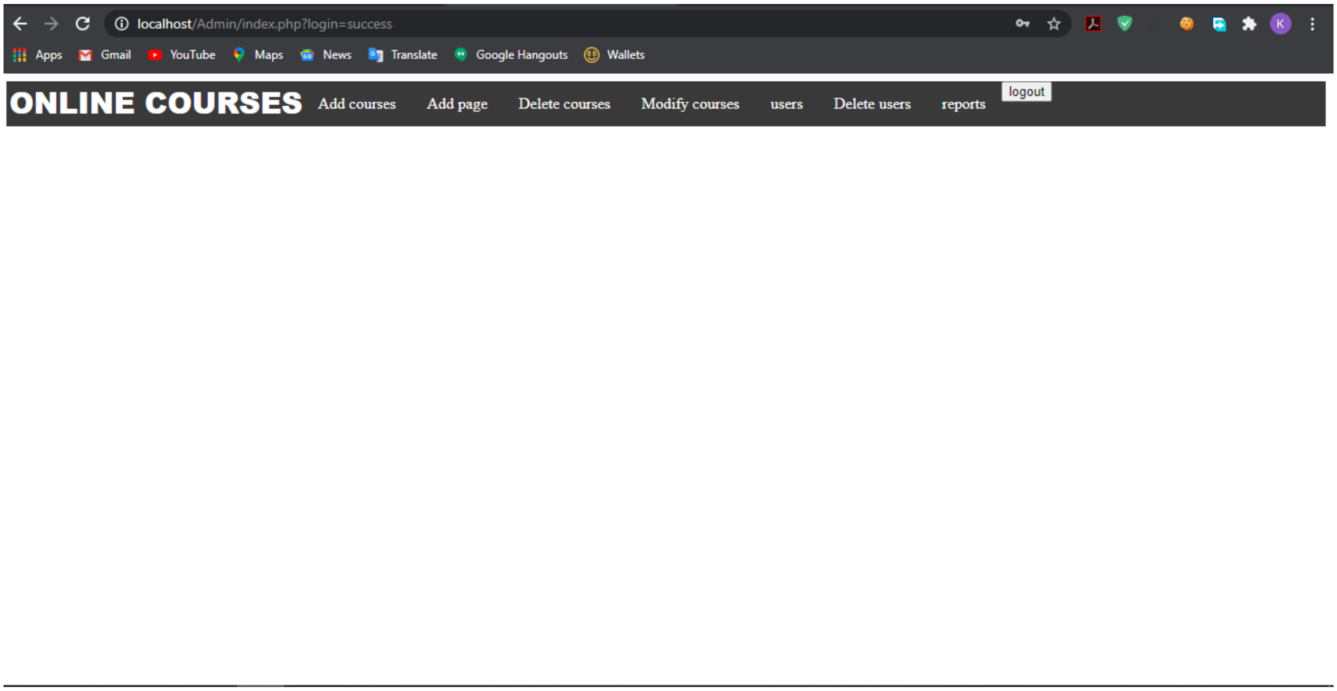


Figure 4: admin main page

In this webpage, the admin can choose to add, modify, or delete a course and see the activity of the users, delete a user, and see the reports the user send.

3.2.1.3. Add Course

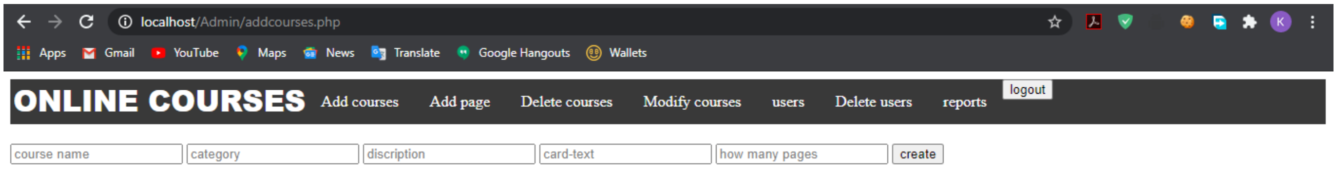


Figure 5: add course

In this web page, the admin can add a new website by entering the course name, course category like IT, management, or marketing, the course description to let the user know what the course is about, the course card text which is a small description about the course, and how many pages does the course consist of.

3.2.1.4. Add page

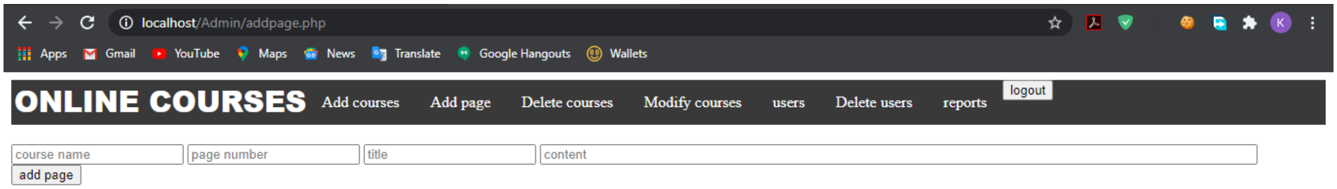


Figure 6: add page

In order to be the entry of the course completed the admin must enter the course pages, here in this webpage, the admin enters the course name, page number, title of this specific page, and the content of the page, and when the admin finish entering the course pages he/she just simply exit the page.

3.2.1.5. Delete Course

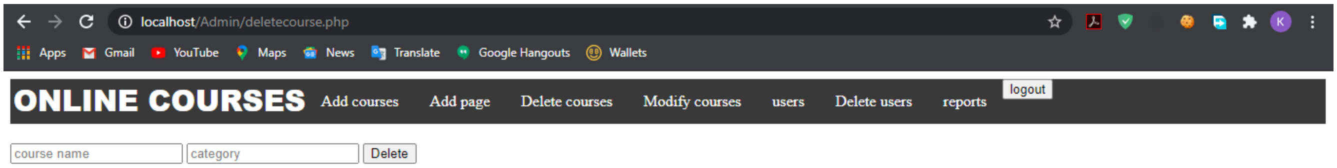


Figure 7: delete course

In this web page, the user can delete a course by just entering the course name and it’s category.

3.2.1.6. modify courses

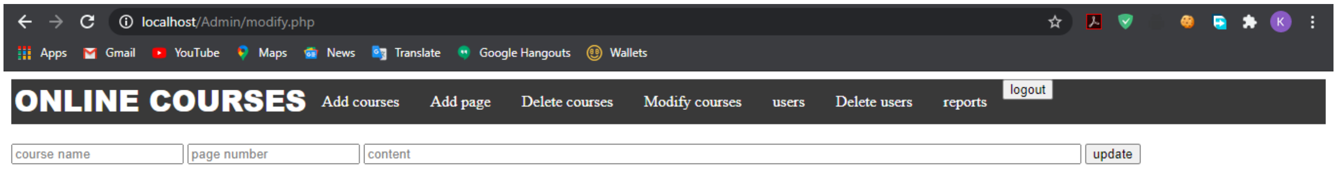
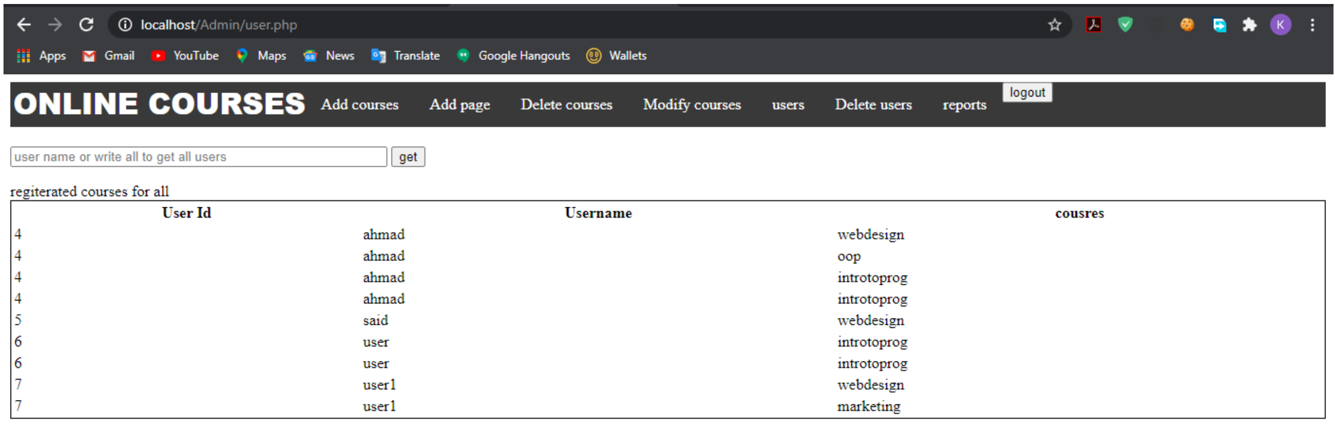


Figure 8: modify courses

In this web page, the admin can modify a specific course page by simply entering the course name, page number, and the new content of the page.

3.2.1.7. users activity



regiterated courses for all

User Id	Username	cousres
4	ahmad	webdesign
4	ahmad	oop
4	ahmad	introtoprog
4	ahmad	introtoprog
5	said	webdesign
6	user	introtoprog
6	user	introtoprog
7	user1	webdesign
7	user1	marketing

Figure 9: users activity

In this web page, the admin can see a specific user activity by entering the username of the user, or the admin can see all the users activity by entering the word ‘all’.

3.2.1.8. Delete users

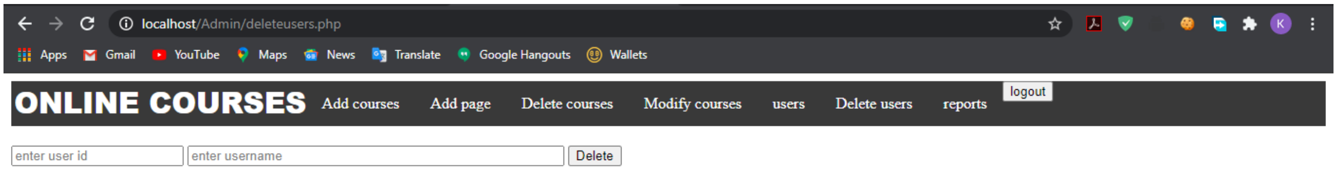
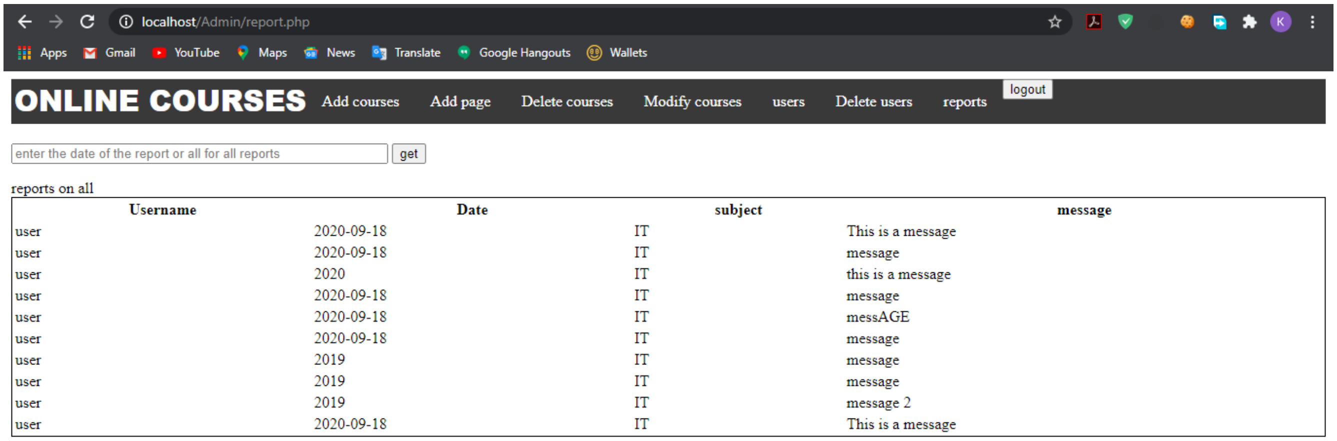


Figure 10: delete users

In this web page, the user can delete a user by entering his user id and username.

3.2.1.9. reports



reports on all

Username	Date	subject	message
user	2020-09-18	IT	This is a message
user	2020-09-18	IT	message
user	2020	IT	this is a message
user	2020-09-18	IT	message
user	2020-09-18	IT	messAGE
user	2020-09-18	IT	message
user	2019	IT	message
user	2019	IT	message
user	2019	IT	message 2
user	2020-09-18	IT	This is a message

Figure 11: reports

Here, the admin can see the reports from all dates or a specific date, and these reports consists of the username of the user who sent the report, the date of the report, subject of the report, and the message.

3.2.2. Users system manual

3.2.2.1. login page

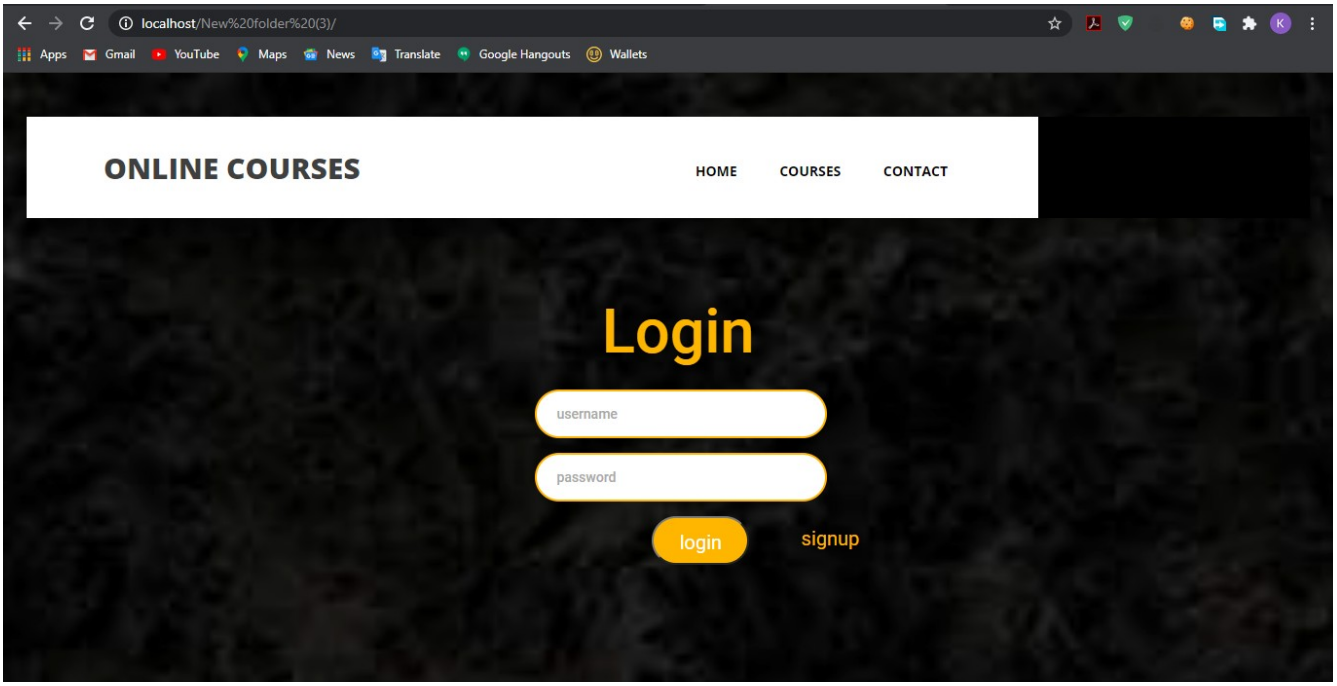


Figure 12: user login page

In this web page, the user can login to the page which is obligatory to be able to get the courses, and if the user doesn't have an username and a password, he/she can go to the sign up page.

3.2.2.2. Sign up

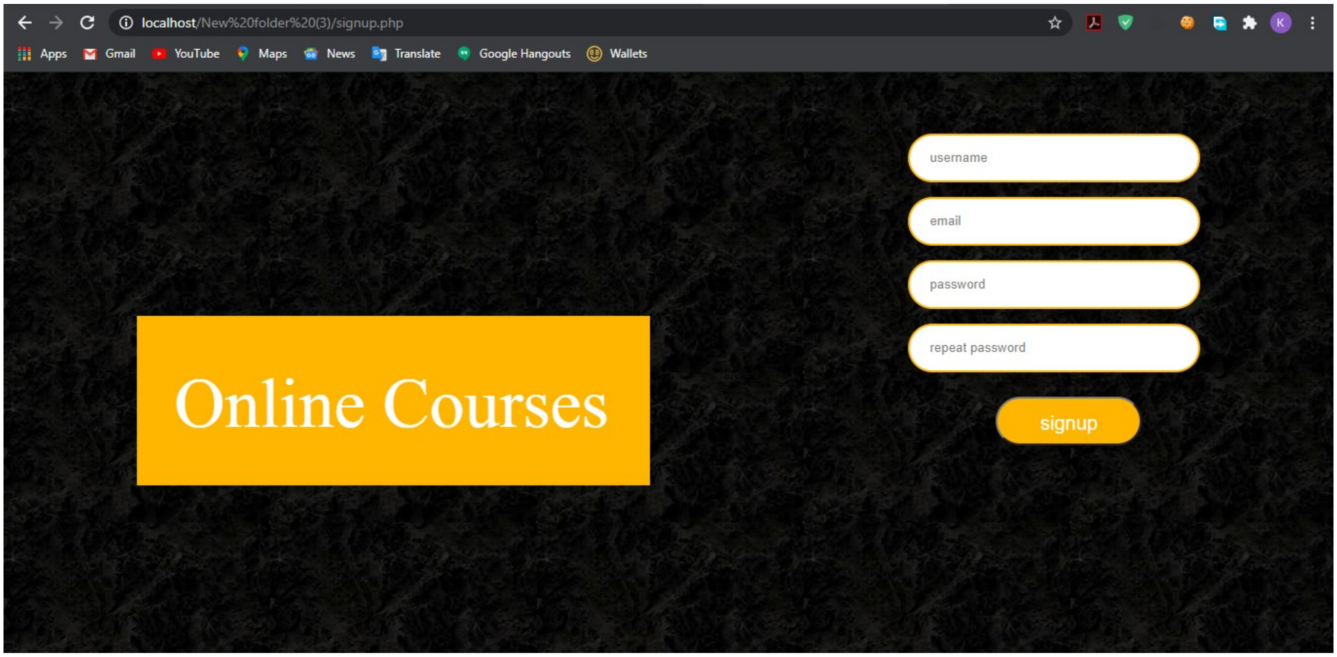
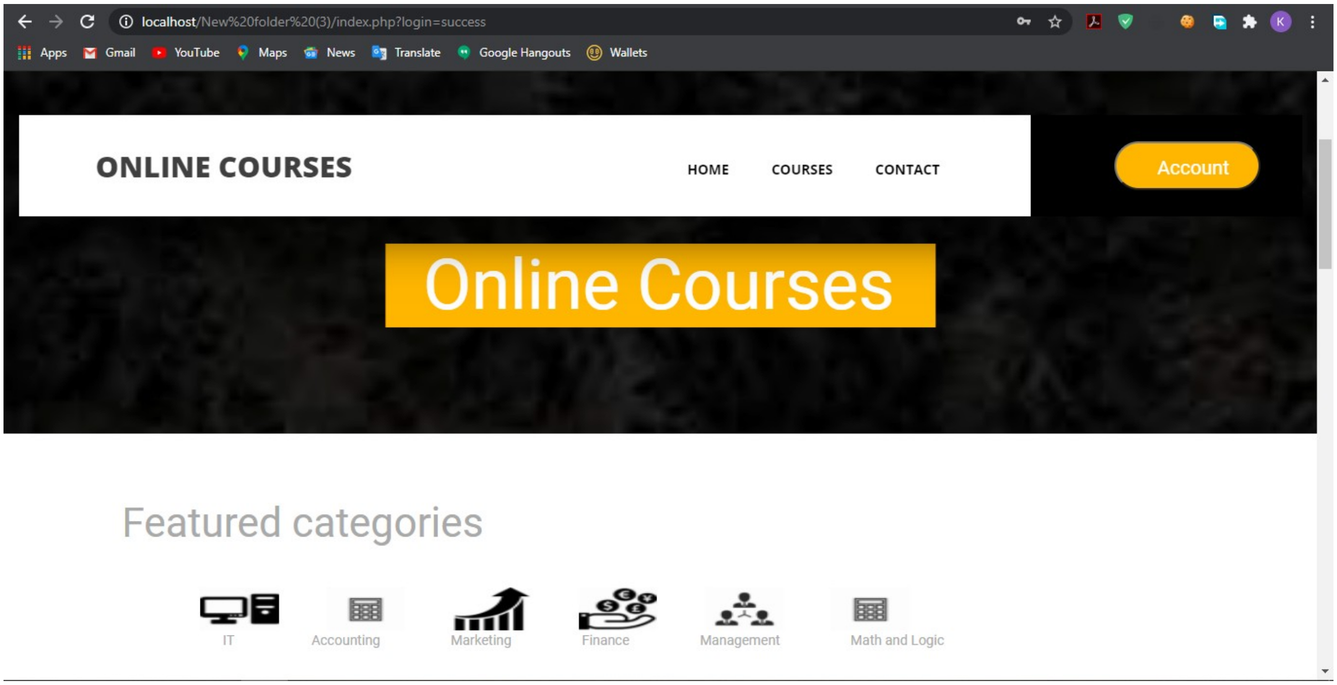


Figure 13: user sign up page

In the sign up page, the user can register in the web page by simply entering a unique username, a valid email, a strong password, and after that he/she can log in to the web page.

3.2.2.3.Home page



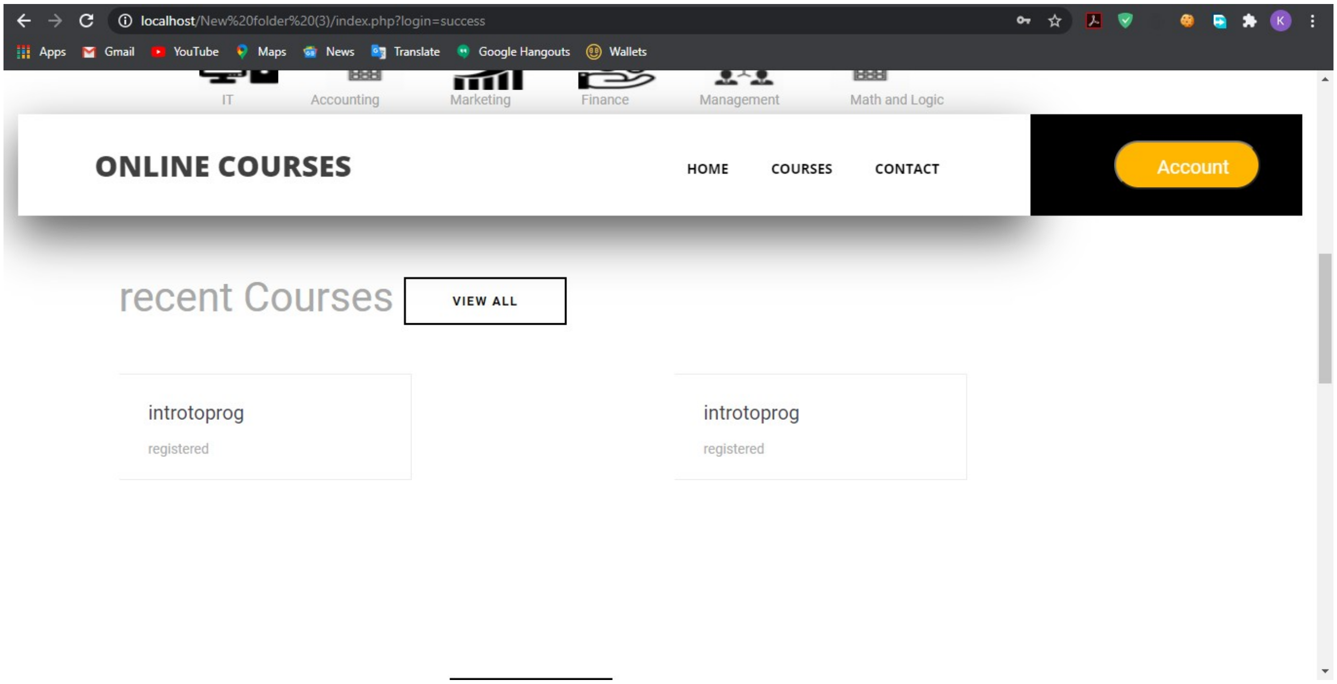


Figure 15: user home page part 2

In the home page, the user can get some general info about the website if the user is new to the website, and if the user is an old user. he/she can directly go to the recent course and continue where they left.

3.2.2.4. courses

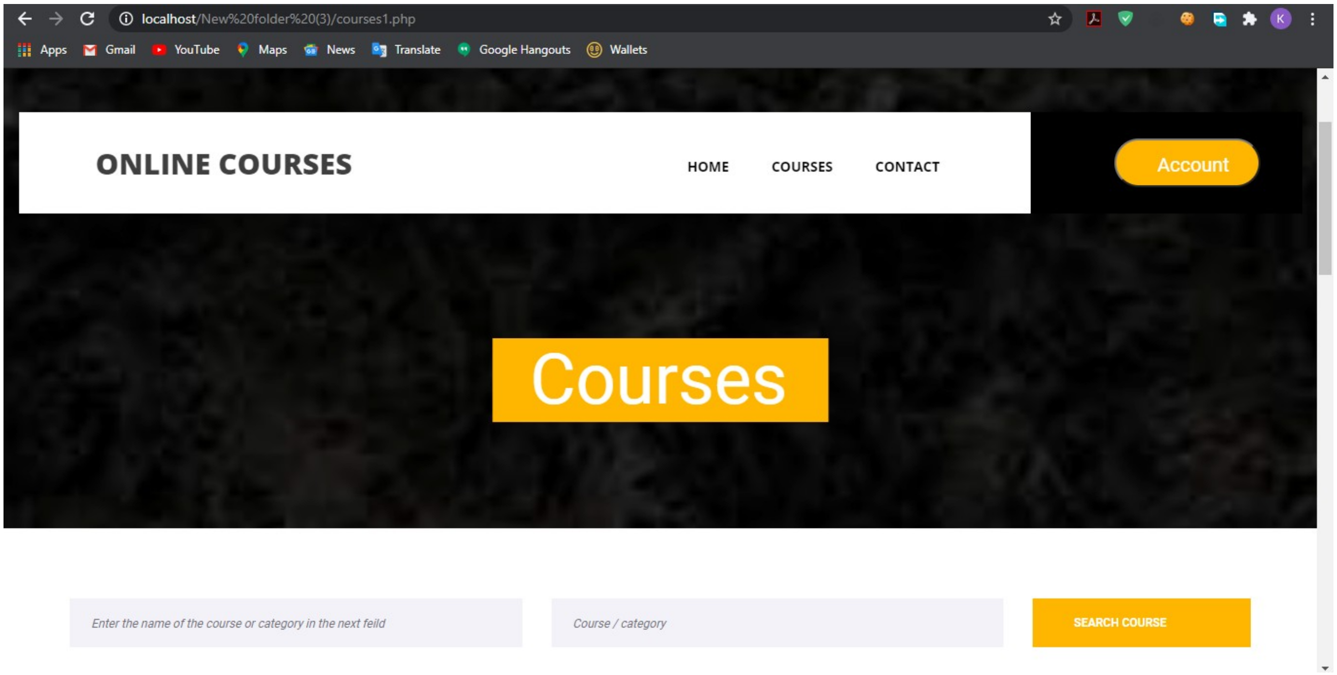
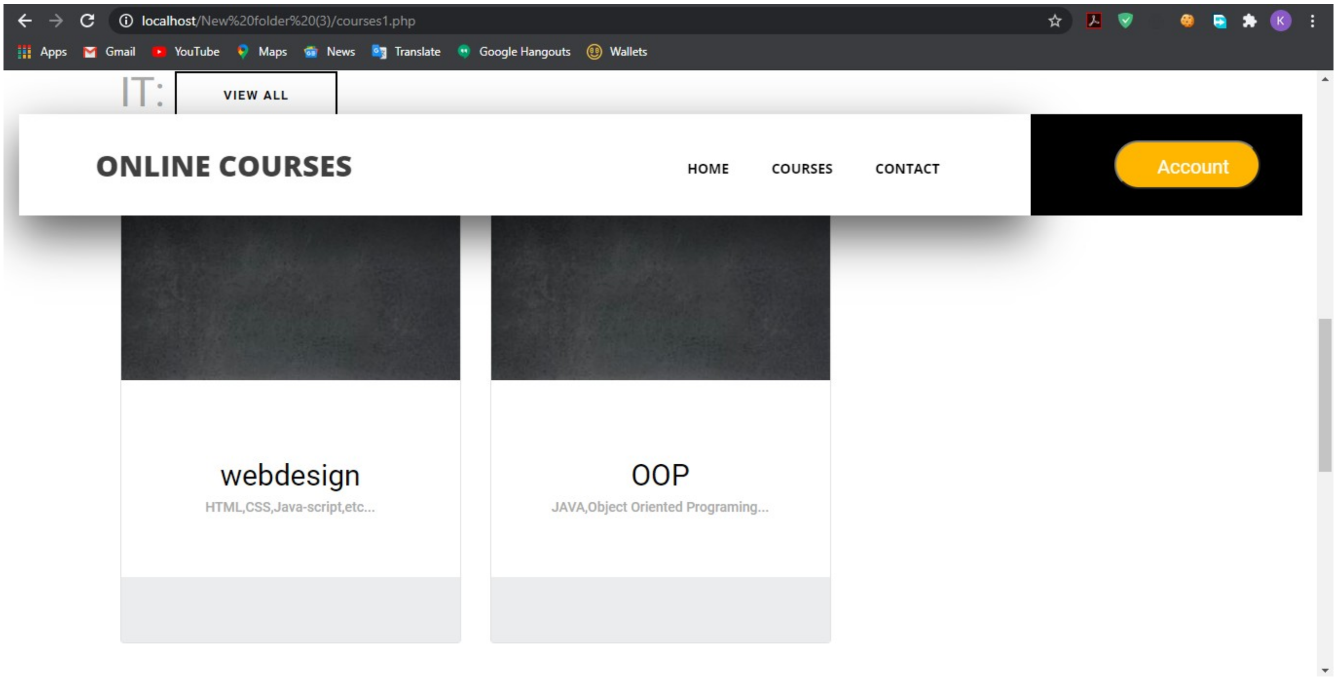


Figure 16: user courses page



Here, the user can chose between different course to register in.

3.2.2.5. course description

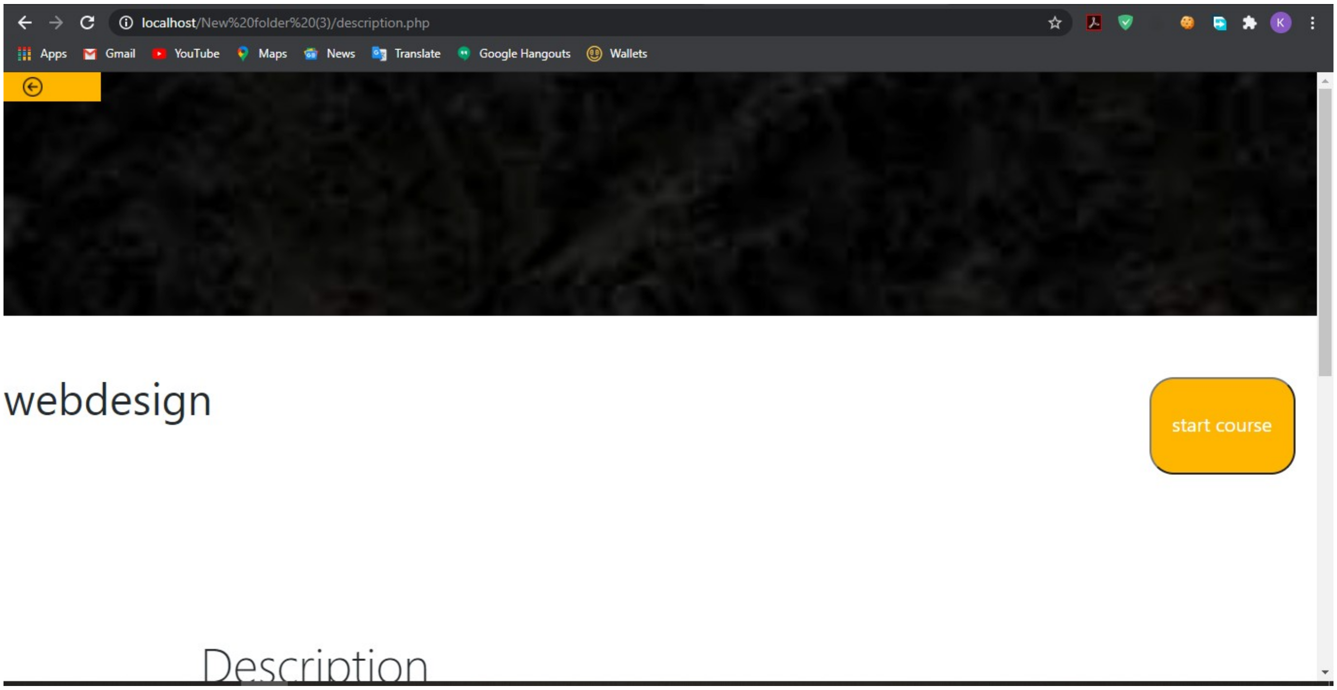
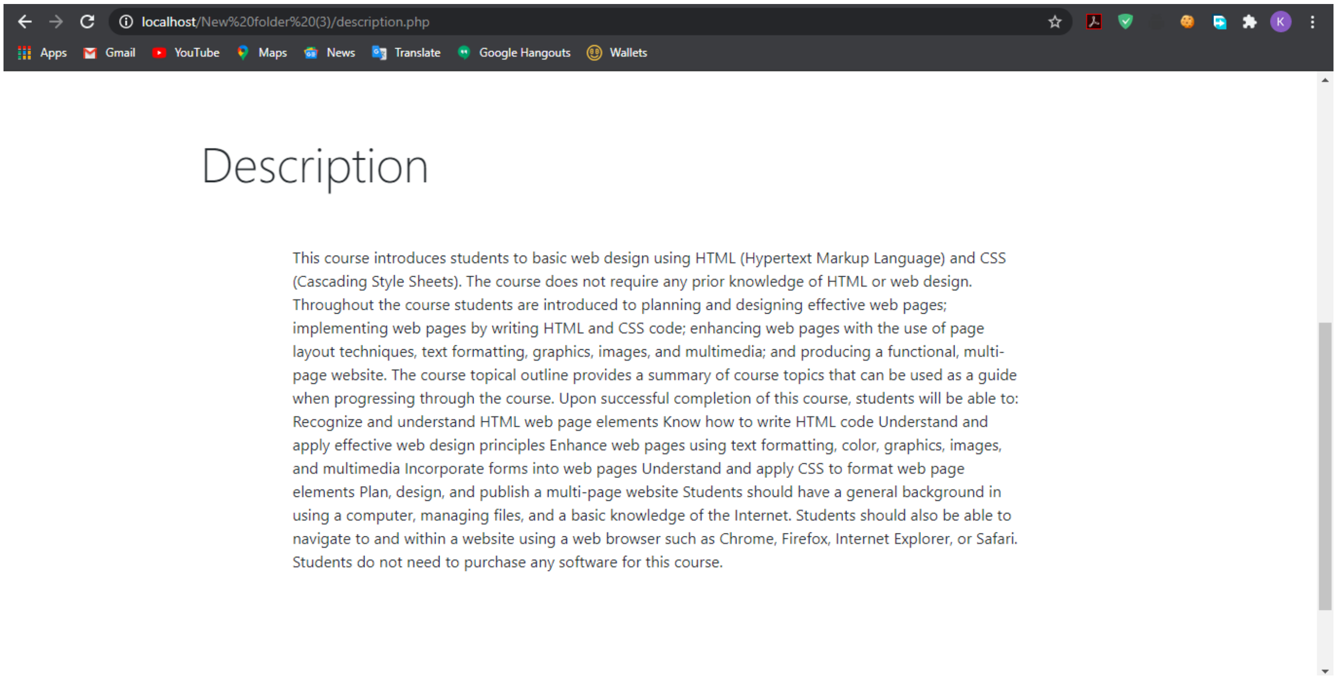


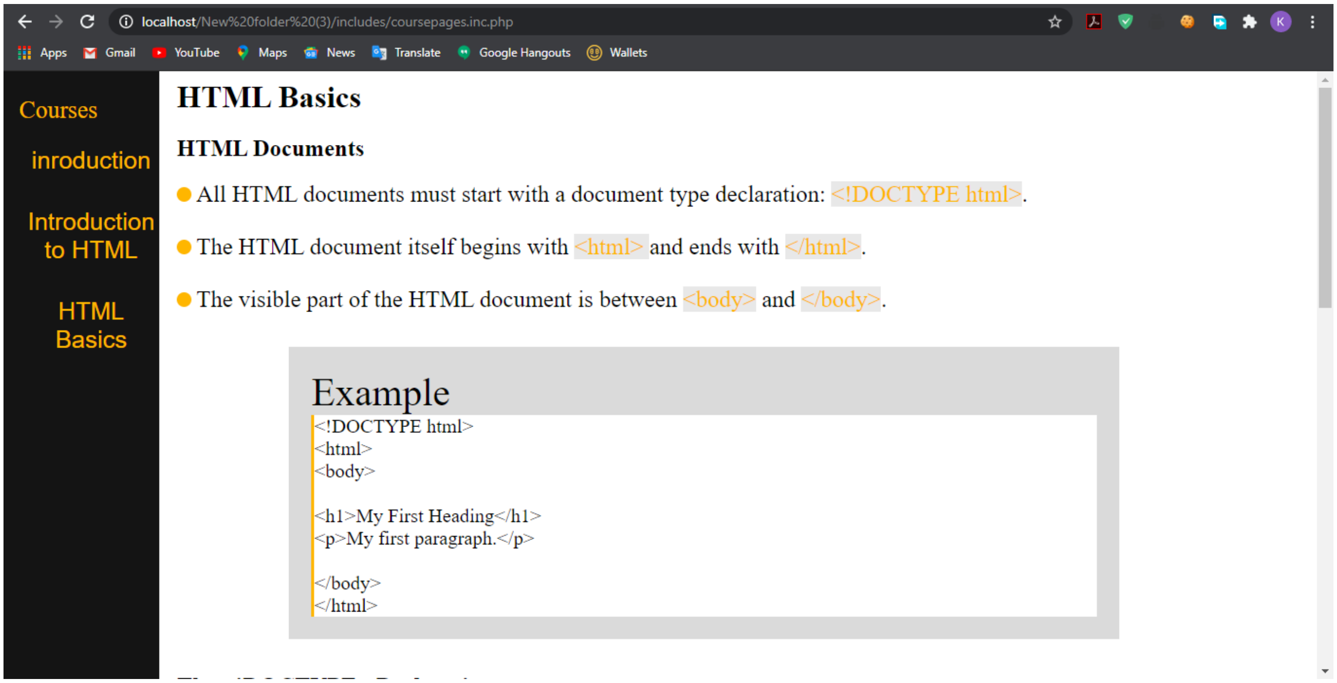
Figure 17: course description page



This course introduces students to basic web design using HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets). The course does not require any prior knowledge of HTML or web design. Throughout the course students are introduced to planning and designing effective web pages; implementing web pages by writing HTML and CSS code; enhancing web pages with the use of page layout techniques, text formatting, graphics, images, and multimedia; and producing a functional, multi-page website. The course topical outline provides a summary of course topics that can be used as a guide when progressing through the course. Upon successful completion of this course, students will be able to: Recognize and understand HTML web page elements Know how to write HTML code Understand and apply effective web design principles Enhance web pages using text formatting, color, graphics, images, and multimedia Incorporate forms into web pages Understand and apply CSS to format web page elements Plan, design, and publish a multi-page website Students should have a general background in using a computer, managing files, and a basic knowledge of the Internet. Students should also be able to navigate to and within a website using a web browser such as Chrome, Firefox, Internet Explorer, or Safari. Students do not need to purchase any software for this course.

Here, the user can get all the info he need about the course like what the course is about, what does it consist of, and what kind of study it is. And if the user wants to register in this course, he/she just have to press the ‘start course’ button.

3.2.2.6. course info



HTML Basics

HTML Documents

- All HTML documents must start with a document type declaration: <!DOCTYPE html>.
- The HTML document itself begins with <html> and ends with </html>.
- The visible part of the HTML document is between <body> and </body>.

Example

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Figure 18: course info page

In this web page, the user can start studying this course, and easy moves between chapters, this web page could consist of the chapters of the course on the left of the web page, and the info of the chapter and some examples.

3.2.2.7. contact

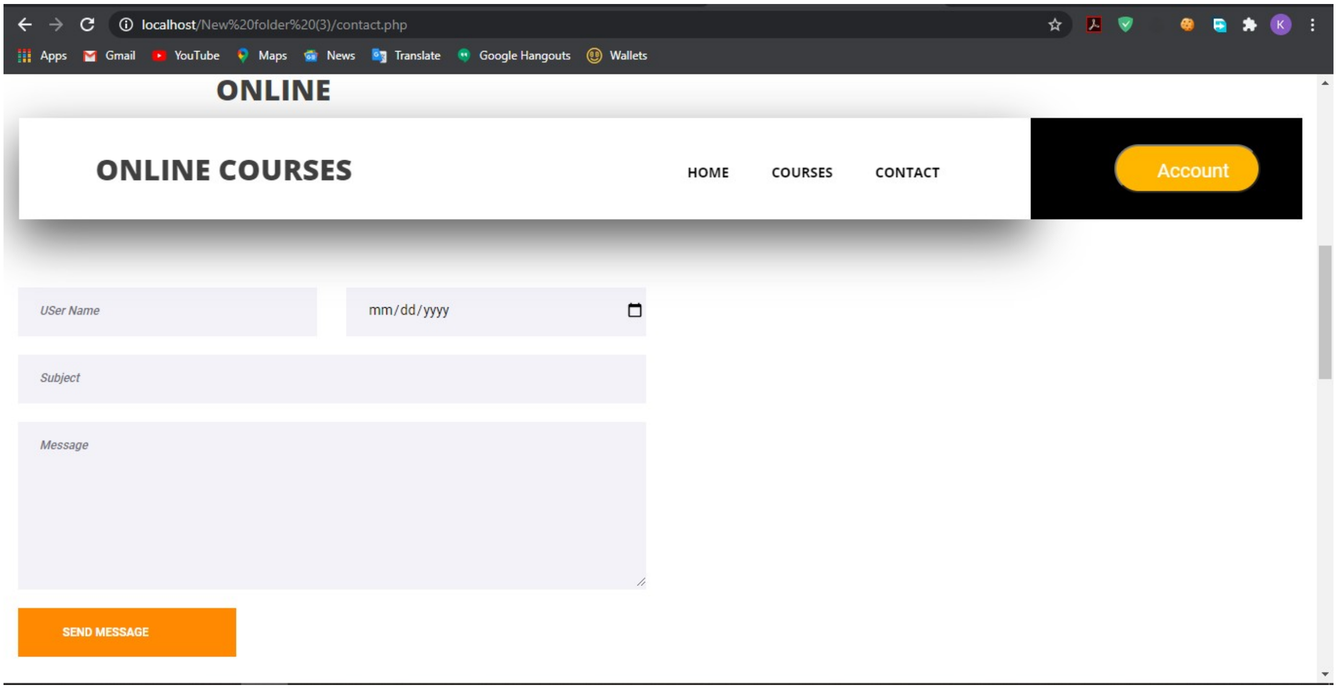


Figure 19: contact page

In this web page, the user can contact the admin of the page if he/she had a problem or asking for help. The user should enter the username, the date, the subject of the message, and the message.

General Conclusion, Perspectives and Recommendations

Conclusion

In conclusion, the online course registration could change the way of getting a course in a university in to a much easier and less time consuming way, that will help the student in getting the needed time for studying the course, and the student can get a course any time and any where thanks to this website. Also, the teacher will have a benefit from this program, because the teacher will be able to search his course with his/her students easily and with no complications on this website and even if he/she don't know how to code, the website will do all the work instead of them.

Limitations

The limitation is that in the admin website, the page added in the add page web page isn't pre-styled as intended to be, so if the admin what a styled pages in a specific course he/she needed to style it him/herself then enter it to the web page, and that will make a little of complication with some admins.

Perspectives

In my perspective, this project is very useful in the future, since the future is already heading to remotely studying and working, so using this kind of programs is obligatory in the future. This program can connect the student with the university any time and any where through the existing online courses.

Recommendations

I highly recommend using this program and I also recommend always updating this program for more user friendly interface which will get for this program more interest in.

References

Data dictionary, TechTarget, https://searchapparchitecture.techtarget.com/definition/data-dictionary

Conceptual data model, techopedia, https://www.techopedia.com/definition/28026/conceptual-data-model

physical data model techopedia,
[https://www.techopedia.com/definition/30600/physical-data-model]

HTML, Computer Hope, https://www.computerhope.com/jargon/h/html.htm

CSS (cascading style sheets), TheServerSide,

https://www.theserverside.com/definition/cascading-style-sheet-CSS#:~:text=This%20definition%20explains%20the%20meaning%20of%20CSS%20%28cascading,preferred%20mechanism%20for%20formatting%20HTML%20pages.%20Conforming%20with

JavaScript, techopedia,
https://www.techopedia.com/definition/3929/javascript-js

What is Bootstrap? Definition & Tutorial for Beginners, TemplateToaster,

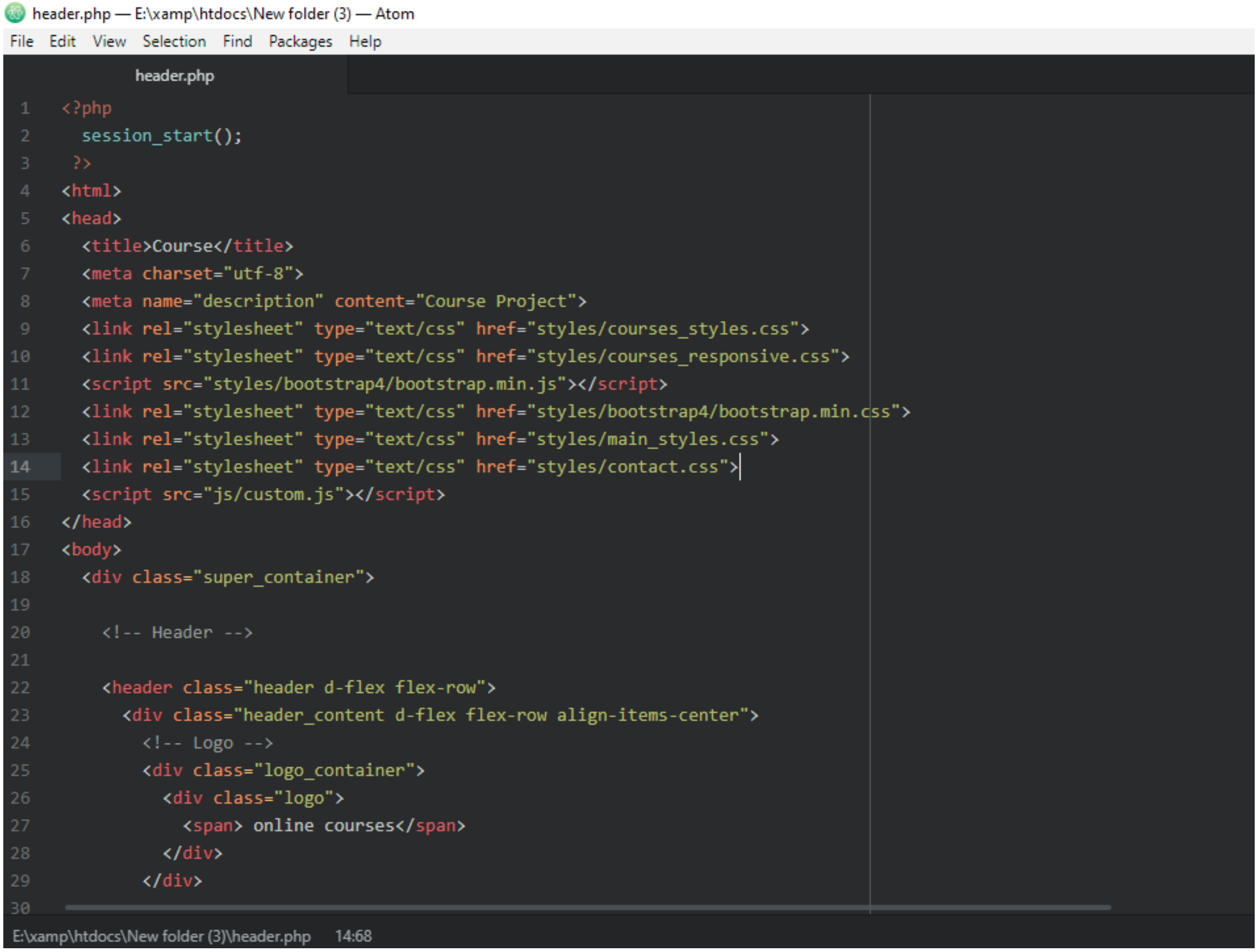
https://blog.templatetoaster.com/what-is-bootstrap/

Kevin Yank, PHP & MYSQL NOVICE TO NINJA, fifth edition, SitePoint, 2012

Len Silverston, The Data Model Resource Book, volume 1, 1997

Genius Guide, Welcome to HTML5 & CSS3, volume 2, Imagine Publishing Ltd, 2015

Appendix 1


header.php — E:\xamp\htdocs\New folder (3) — Atom
File
Edit
View
Selection
Find
Packages
Help
header.php
E:\xamp\htdocs\New folder (3)\header.php
14:68

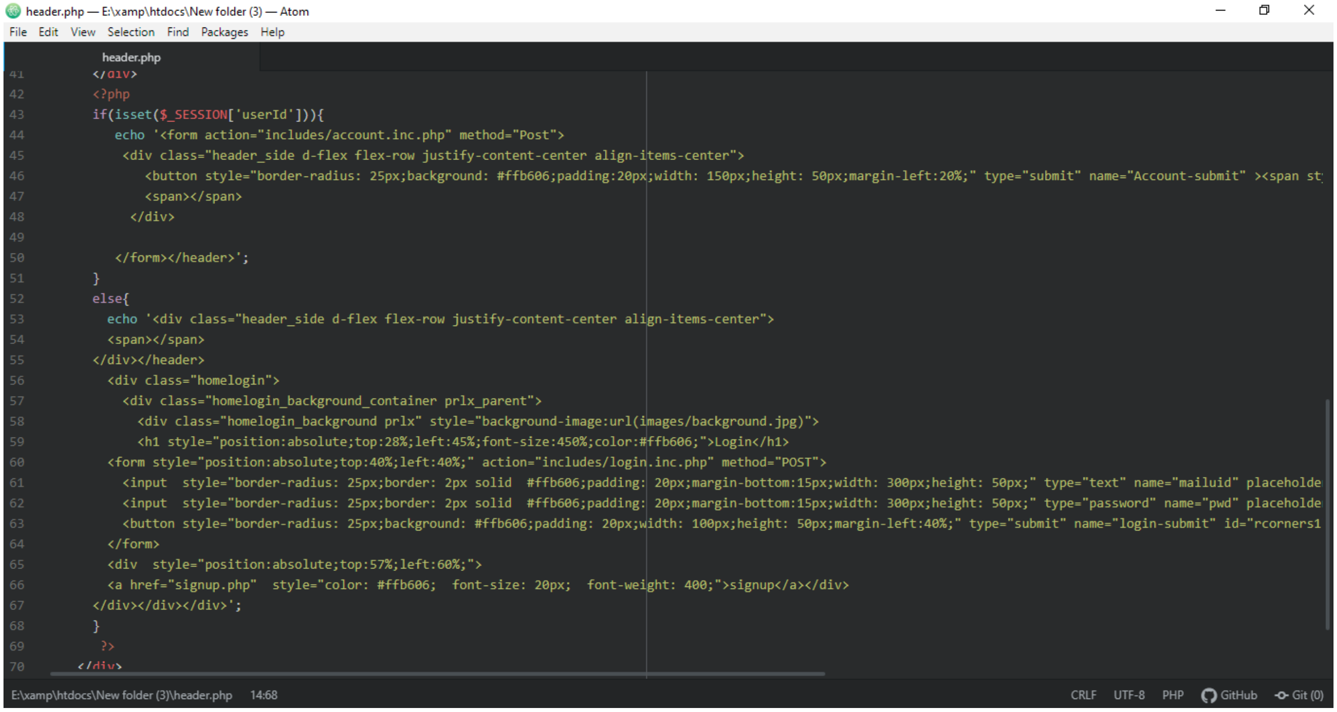
```
<?php
  session_start();
 ?>
<html>
<head>
  <title>Course</title>
  <meta charset="utf-8">
  <meta name="description" content="Course Project">
  <link rel="stylesheet" type="text/css" href="styles/courses_styles.css">
  <link rel="stylesheet" type="text/css" href="styles/courses_responsive.css">
  <script src="styles/bootstrap4/bootstrap.min.js"></script>
  <link rel="stylesheet" type="text/css" href="styles/bootstrap4/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="styles/main_styles.css">
  <link rel="stylesheet" type="text/css" href="styles/contact.css">
  <script src="js/custom.js"></script>
</head>
<body>
  <div class="super_container">

    <!-- Header -->

    <header class="header d-flex flex-row">
      <div class="header_content d-flex flex-row align-items-center">
        <!-- Logo -->
        <div class="logo_container">
          <div class="logo">
            <span> online courses</span>
          </div>
        </div>
```

Figure 20: code sample 1

Appendix 2


header.php — E:\xamp\htdocs\New folder (3) — Atom
File
Edit
View
Selection
Find
Packages
Help
header.php
E:\xamp\htdocs\New folder (3)\header.php
14:68
CRLF
UTF-8
PHP
GitHub
Git (0)

```
        </div>
        <?php
        if(isset($_SESSION['userId'])){
           echo '<form action="includes/account.inc.php" method="Post">
            <div class="header_side d-flex flex-row justify-content-center align-items-center">
               <button style="border-radius: 25px;background: #ffb606;padding:20px;width: 150px;height: 50px;margin-left:20%;" type="submit" name="Account-submit" ><span st
               <span></span>
             </div>

           </form></header>';
        }
        else{
          echo '<div class="header_side d-flex flex-row justify-content-center align-items-center">
          <span></span>
        </div></header>
          <div class="homelogin">
            <div class="homelogin_background_container prlx_parent">
              <div class="homelogin_background prlx" style="background-image:url(images/background.jpg)">
              <h1 style="position:absolute;top:28%;left:45%;font-size:450%;color:#ffb606;">Login</h1>
          <form style="position:absolute;top:40%;left:40%;" action="includes/login.inc.php" method="POST">
            <input  style="border-radius: 25px;border: 2px solid  #ffb606;padding: 20px;margin-bottom:15px;width: 300px;height: 50px;" type="text" name="mailuid" placeholde
            <input  style="border-radius: 25px;border: 2px solid  #ffb606;padding: 20px;margin-bottom:15px;width: 300px;height: 50px;" type="password" name="pwd" placeholde
            <button style="border-radius: 25px;background: #ffb606;padding: 20px;width: 100px;height: 50px;margin-left:40%;" type="submit" name="login-submit" id="rcorners1
          </form>
          <div  style="position:absolute;top:57%;left:60%;">
          <a href="signup.php"  style="color: #ffb606;  font-size: 20px;  font-weight: 400;">signup</a></div>
        </div></div></div>';
        }
         ?>
      </div>
```

Figure 21: code sample 2