# Overview

A4RBookstore is a web application for buying the books. This application uses core PHP (Version 7.4.21), MYSQL for database and HTML/CSS for view. For styling of the web application, Bootstrap v4.2 is used for CSS. This web application consists of the following features,

* CRUD operation
* User/Admin Login
* Cart System
* Checkout
* Search
* Form Validation
* Mailing Feature

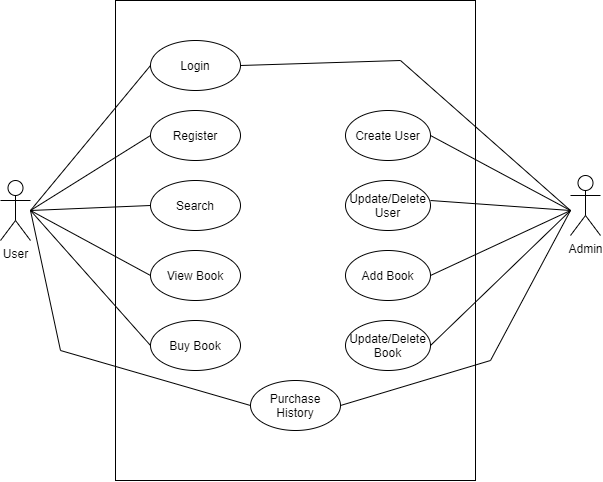
## Installation

A4R Bookstore is a web application developed using core PHP so much installation process is not required. The required application to run this application on your local servers are:

* Apache Server
* MYSQL Server
* IDE
* PHP version 7.4 or higher

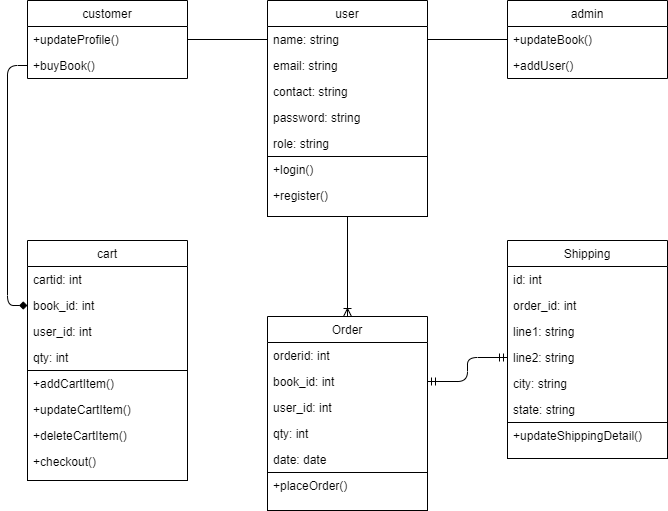
# Methodology

## Use-case Diagram



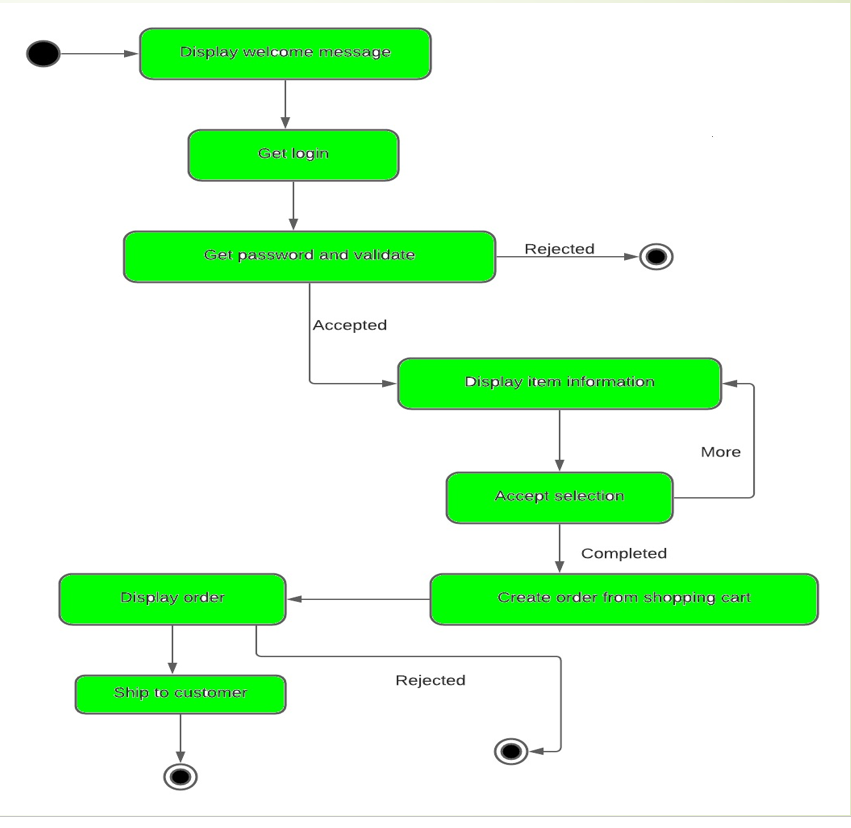
*Fig: Use-case Diagram*

## Design Class Diagram



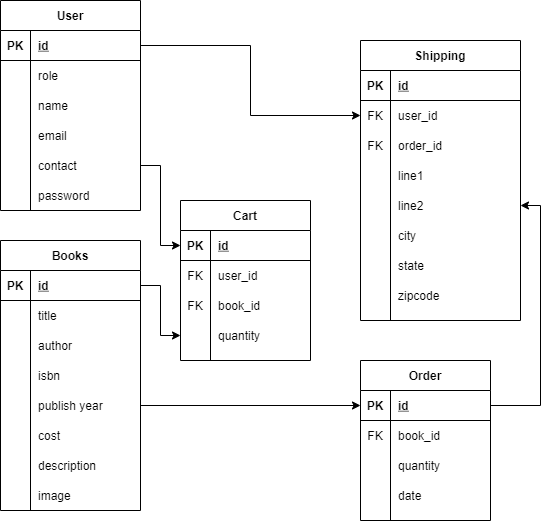
*Fig: Design Class Diagram*

## Activity Diagram



*Fig: Activity Diagram*

## Database Schema



*Fig: Database Schema*

## System Preparation

For the optimal operation of this application, this application must contain at least one admin account that is responsible for the changes in the web application such as adding, removing or updating books and at least one user who is responsible for the buying of the book.

# Tools and Technologies

For the development of this web application and bring it to life, these following tools were implemented:

### PHP

PHP is a general-purpose scripting language geared towards 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.

### MySql

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

### Apache Web Server

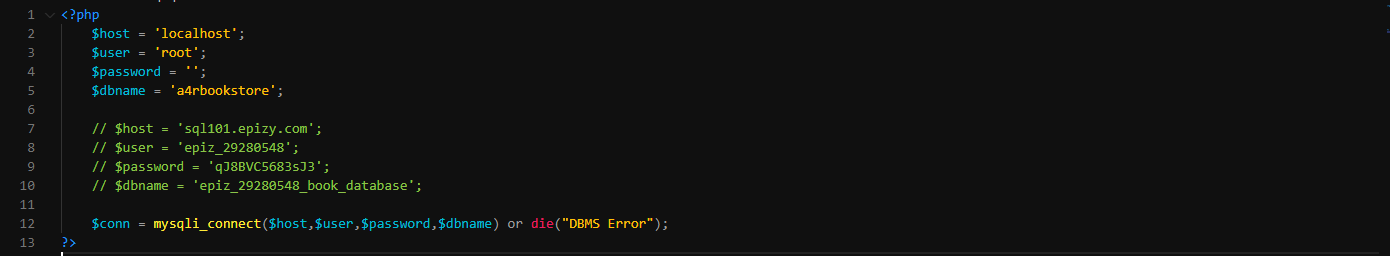
The Apache HTTP Server, colloquially called Apache, is a free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

The vast majority of Apache HTTP Server instances run on a Linux distribution, but current versions also run on Microsoft Windows, OpenVMS, and a wide variety of Unix-like systems. Past versions also ran on NetWare, OS/2 and other operating systems, including ports to mainframes.

# Code Explanation

## dbconnect.php

This page is responsible for the connection between the web application and database.



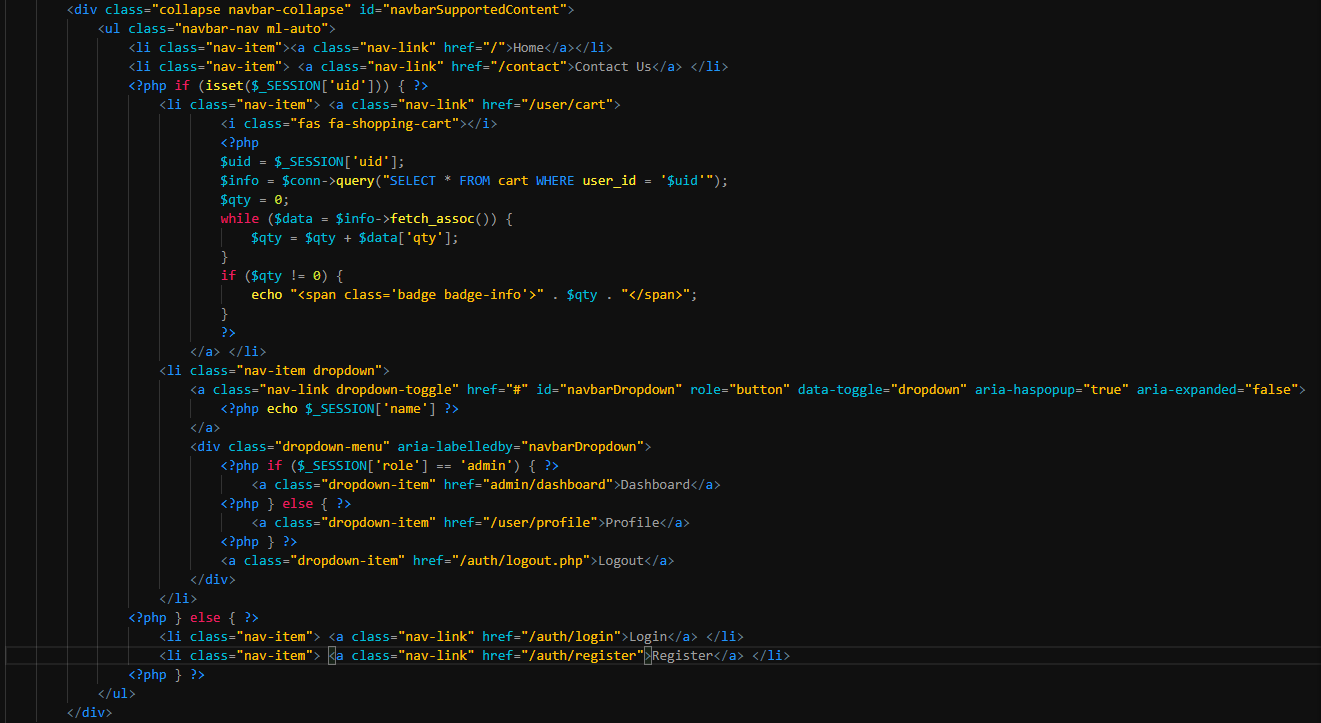
Here, host represent where the MySQL server is hosted, user and password denote the username and password respectively of the MySQL server and dbname denotes the MySQL database name.

This page is included at almost every page in the web application except for those static pages. which doesn’t require database connection for operation.

These values defer as per hosting service used.

## header.php

This page includes the navigation bar used throughout the web application. Here, Login and Register link is replaced with cart and user detail such as profile or dashboard for admin and logout if user is logged in.



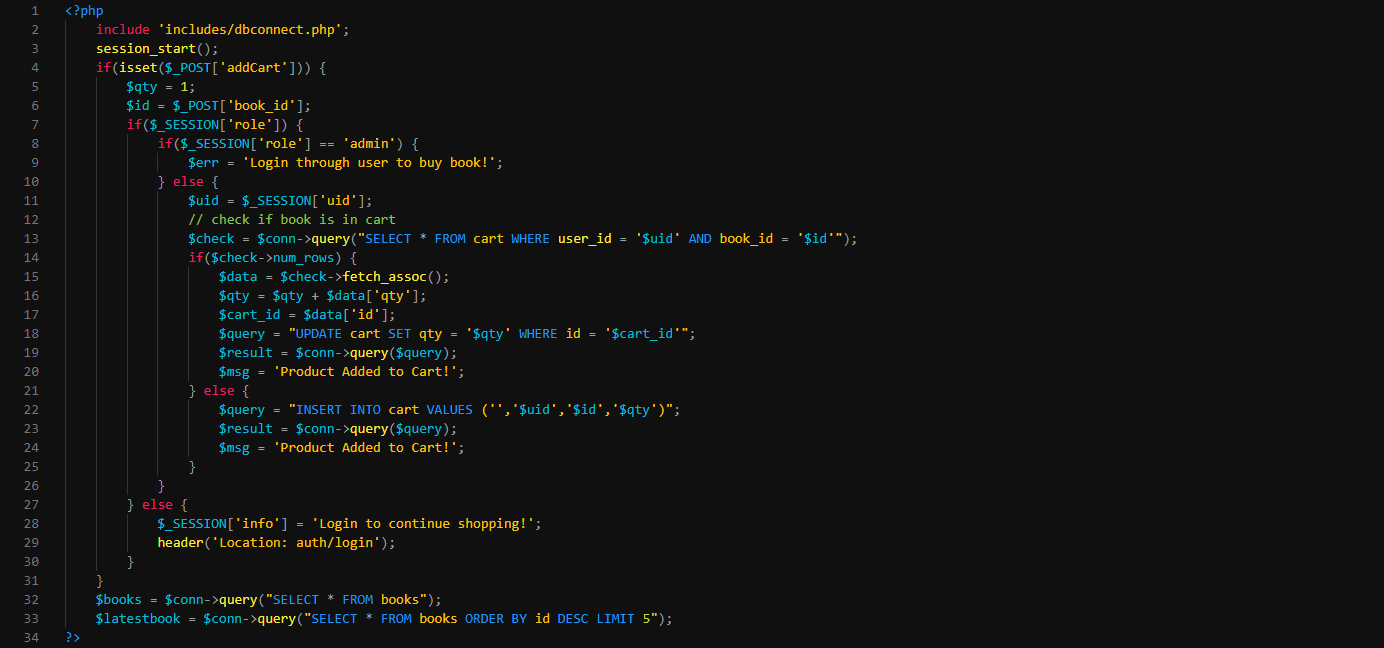
## footer.php

This page includes the footer of the web application and all the javascript used by this web application.



## index.php

The following codes are written in index.php i.e., Landing page of the web application.



These are the php function that are performed on index page.

*include ‘includes/ dbconnect.php’* is responsible of including dbconnect.php on index page which contains database connection code.

Line 32 is responsible for the extraction of all books information from the database and store as array on variable ‘$books’ and Line 33 is responsible for extracting latest 5 books details from the database and store as array on variable ‘$latestbook’.

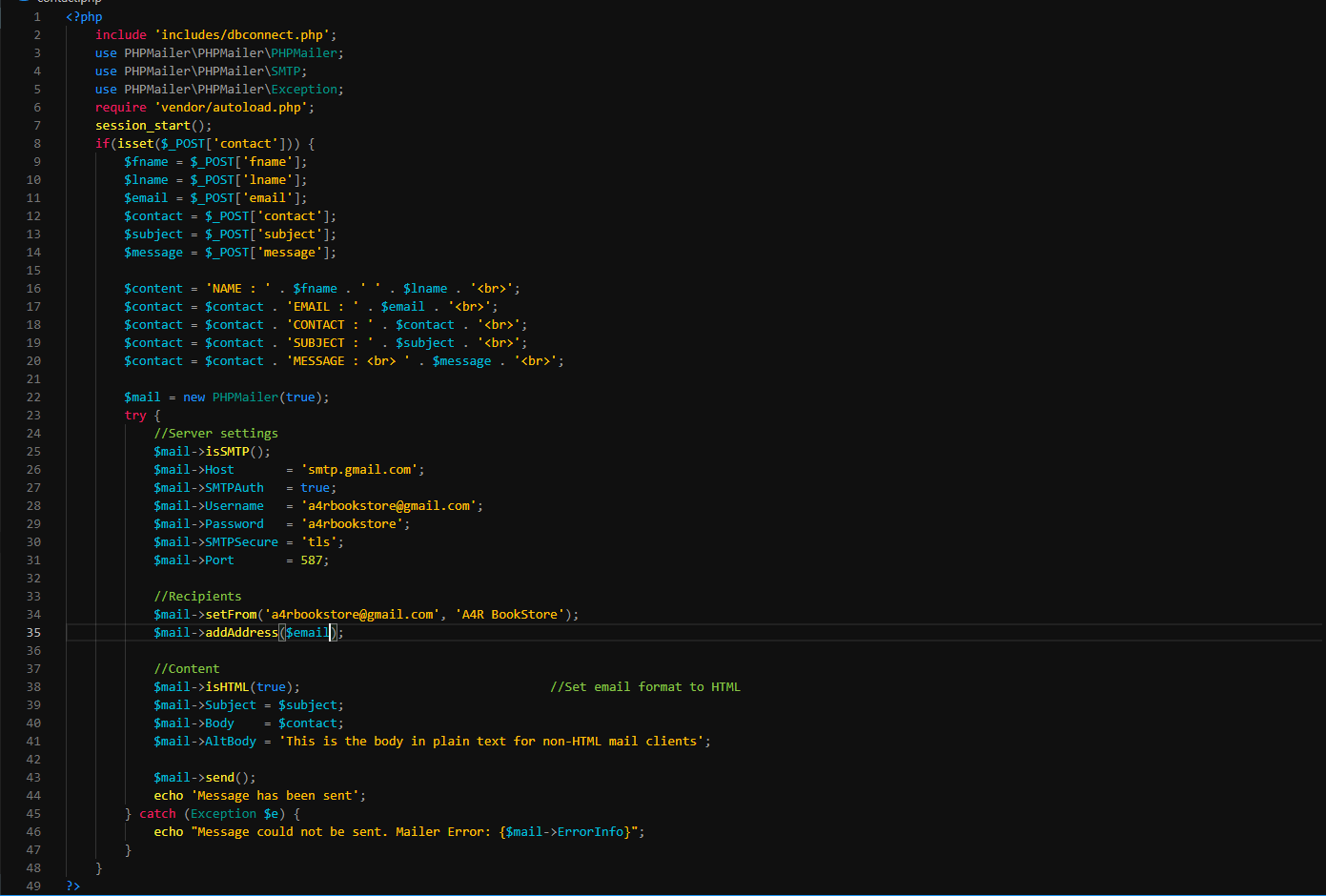
*if(isset($\_POST[‘addCart])) { …}* is responsible for adding the book to user cart. This function is only called if user click on ‘Add to cart’ on product card. This function checks whether the user is logged in or not. If not logged in, he/she is redirected to the login page or else book is added to the cart.



This above code is responsible for the displaying of the product in the page. Here, while loop runs until all the information are read from the array ‘$books’. *$books->fetch\_assoc()* is responsible for reading each rows of the $books variable.

## contact.php

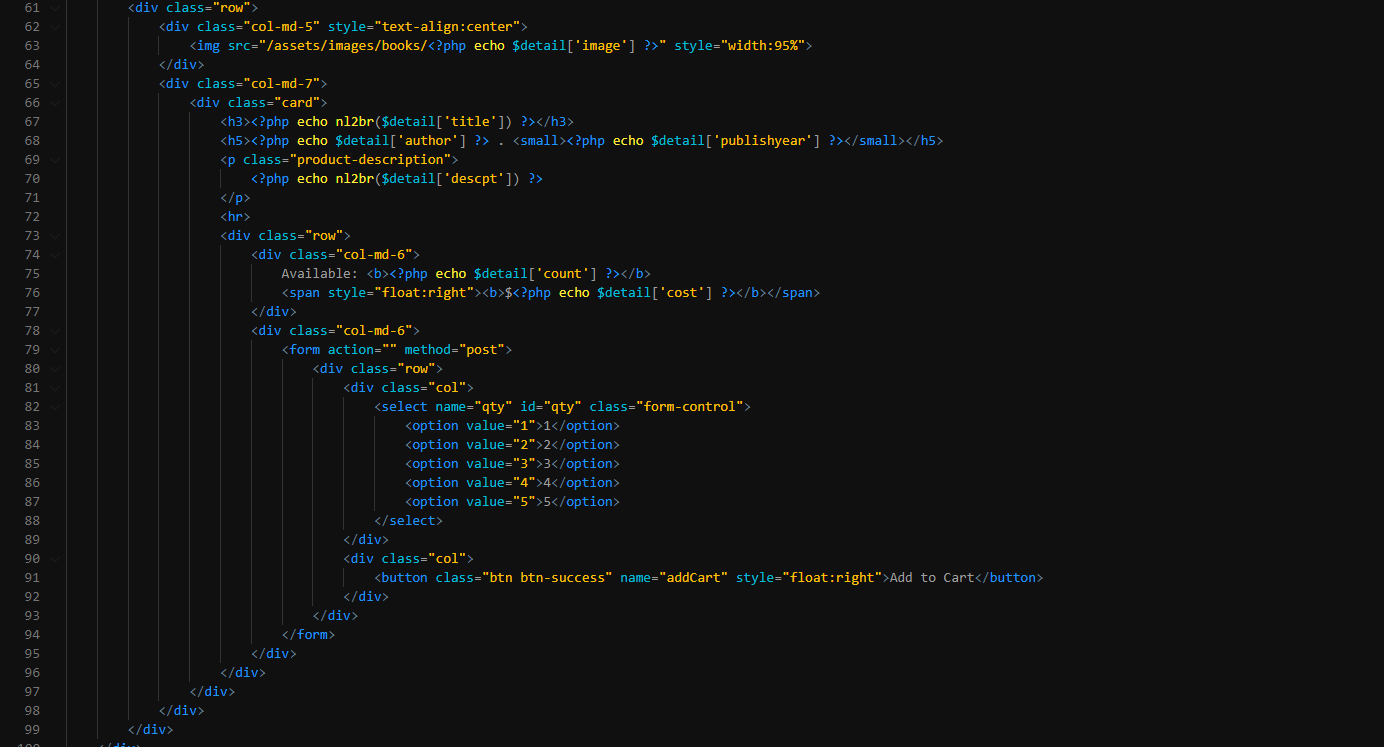
Contact page consists of contact form which sends email to required mail id on form submit. When form submitted, *if(isset($\_POST['contact'])) {..}* function is run.



This code is responsible for sending mail from the web application. It uses PHPMailer plugin for php to send mail.

## book.php

This page shows the detail of the respected book. Here, ‘*$books = $conn->query("SELECT \* FROM books WHERE id = '$id'")*’ extracts the book detail whose id is provided through the url. As in index.php, ‘*if(isset($\_POST[‘addCart])) { …}’* runs when user click on add to cart button from product detail area. But unlike index, here user can select the quantity of book to buy.



This code is responsible of extracting details from database and printing on the webpages. *‘nl2br’* converts the HTML entity code to plain text.

## login.php



Here ‘*if(isset($\_POST[‘login’])){ … }’* handles the login operation of the website. This function checks whether the user provided valid email and password or not. If the valid email and password is provided, the user is logged in and user login detail is saved on a session on that device. Until session is destroyed or timeout, the user stays logged in.

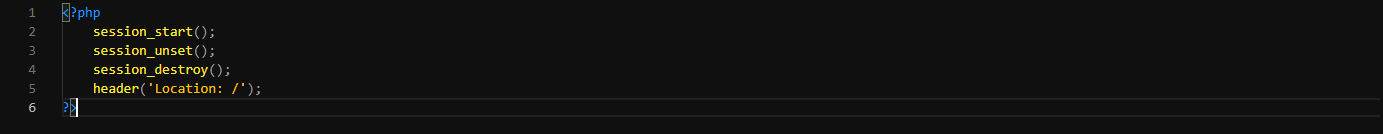
## register.php



Here ‘*if(isset($\_POST[‘register])){ … }’* handles the new user creation for the website. This function checks whether the user provided email exists on the database or not. If the email is unique and all other details is provided, the user is created and is redirected to the login page to login into website. Password is first encrypted before storing to database.

## logout.php

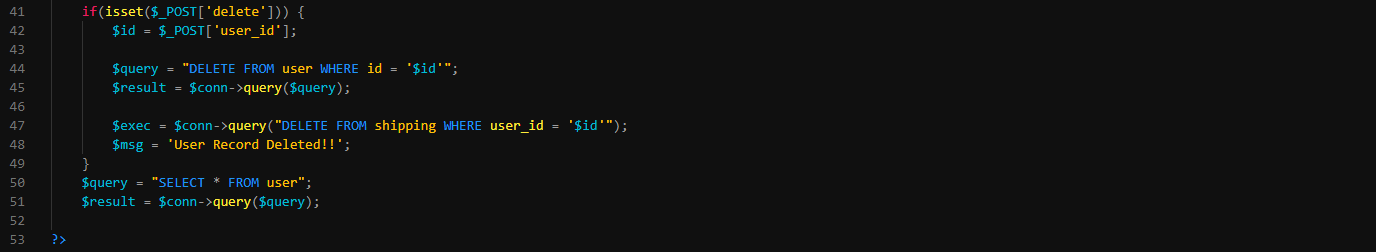
The sole responsible of this page is to destroy the current session on that device and redirect user to index or landing page.



## user.php

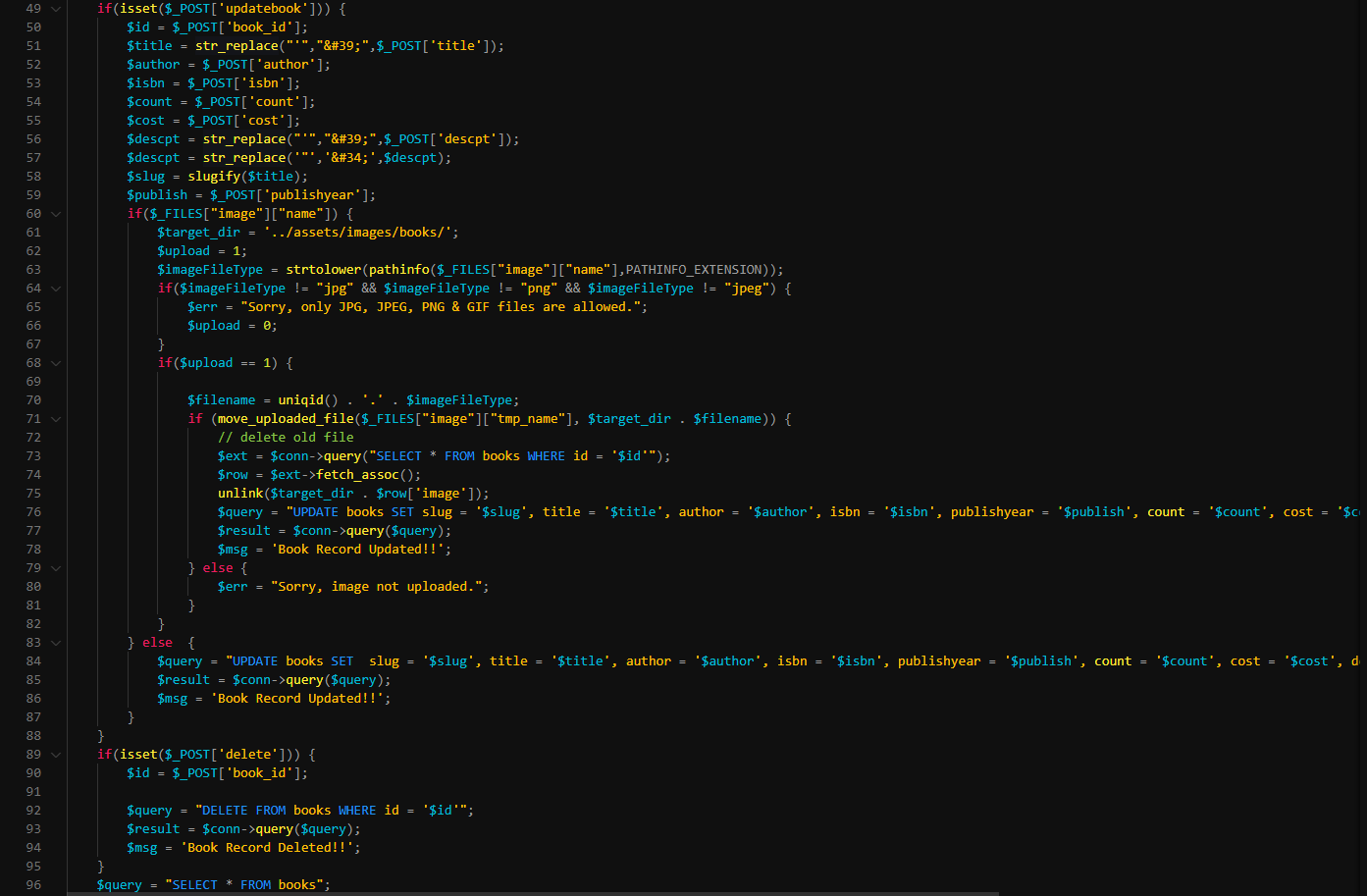
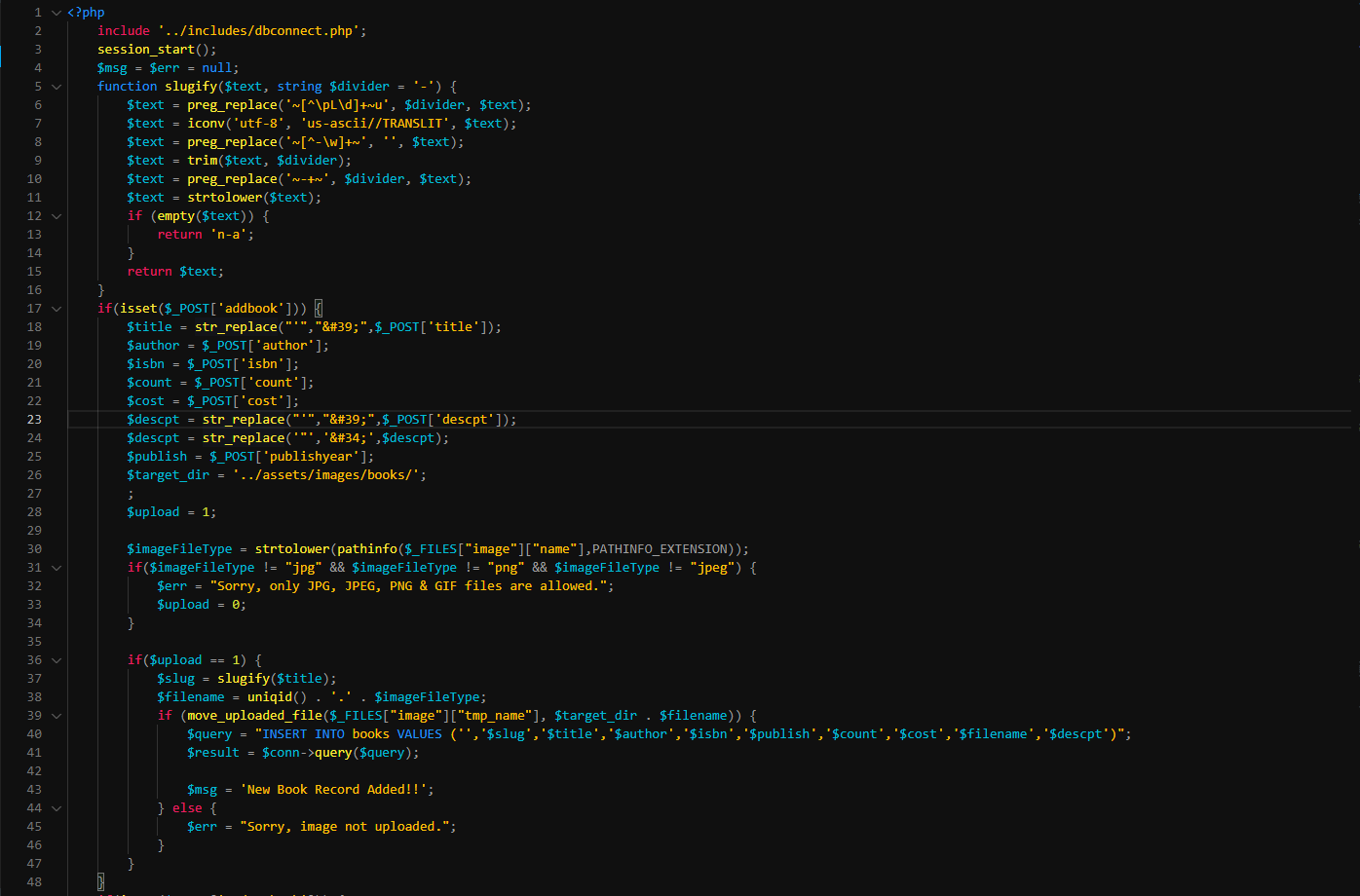
This page is admin side page which shows all user detail. Here admin can view, edit, delete or add new user. In this page, basic CRUD operation is used. ‘*if(isset($\_POST[‘adduser’])){…}*’ is used to add new user to the web application checking if email is not repeated. Password is first encrypted before storing to database. ‘*if(isset($\_POST[‘updateuser’])){…}*’ is used to update the details of the present user. ‘*if(isset($\_POST[‘delete])){…}*’ is used to delete the user from the web application.





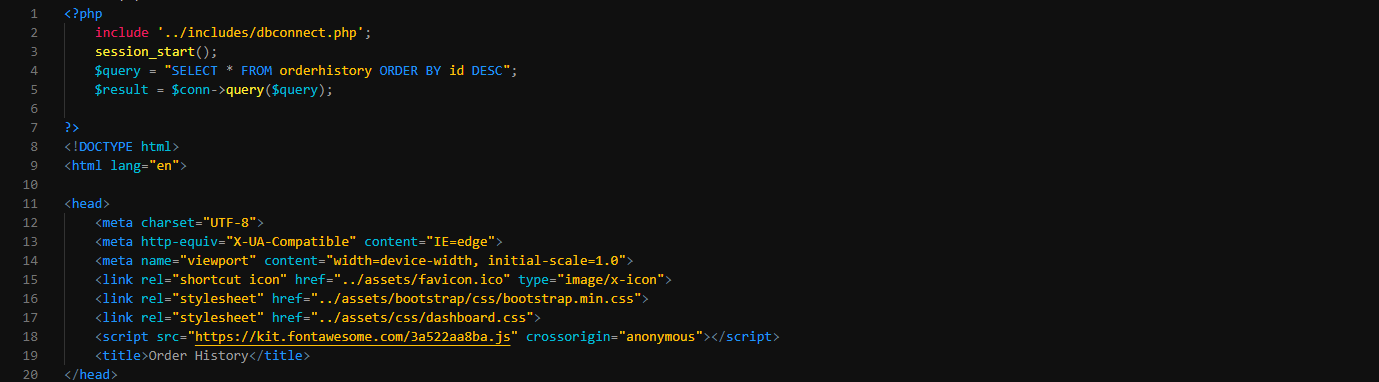
## book-list.php

This page is same as user.php but is for books. Here image can also be uploaded. Image upload is handled by ‘*move\_uploaded\_file()’* operation provided by php. ‘*slugify()*’ is used to generate slug of the book using its title for the URL friendly text. ‘*str\_replace("'","&#39;",$\_POST['descpt'])’* and ‘*str\_replace(‘”’,"&#34;",$\_POST['descpt'])’* is used to replace single quotation (‘) and double quotation (“) to their respect HTML entity number to store in database.



## order.php

This page displays all the order placed by the user. This page only includes simple extract query in reverse order i.e. latest order first.



## dashboard.php

This is index of admin after they login. This page only includes basic information such as the user count, book count, book sold, etc. and latest user created, book added and order placed.



## profile.php

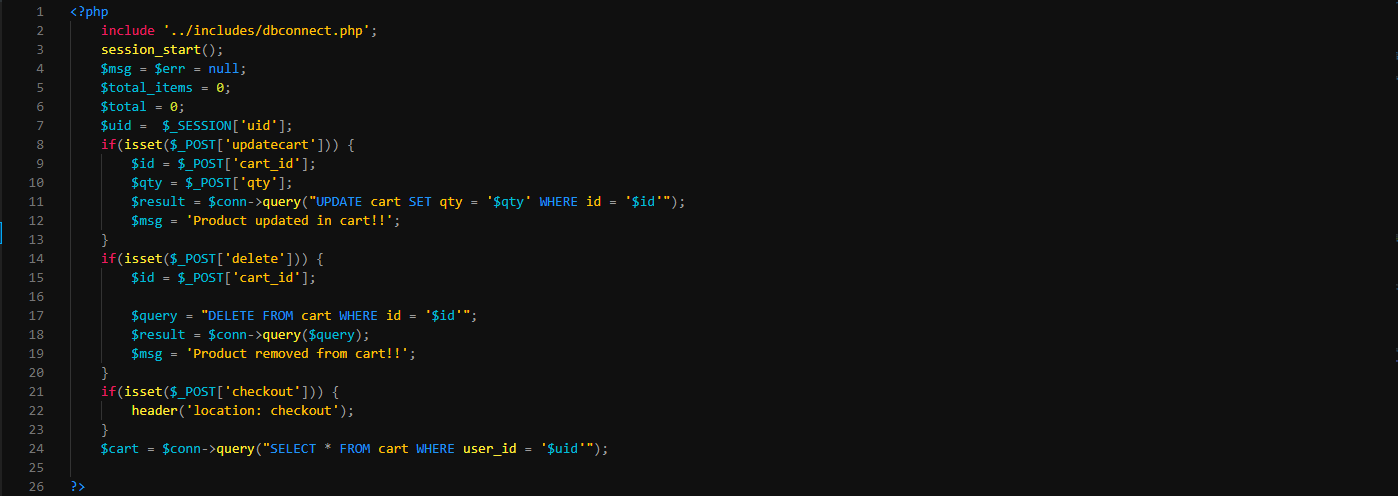
This page displays user detail to them. From this page user can update their basic information as well as see their order history.



*‘if(isset($\_POST[‘update’])){…}’* is responsible for the updating the user detail to database. This checks if user has updated password or not, if no others detail except password is updated or else all details is updated. Password is first encrypted before storing to database.

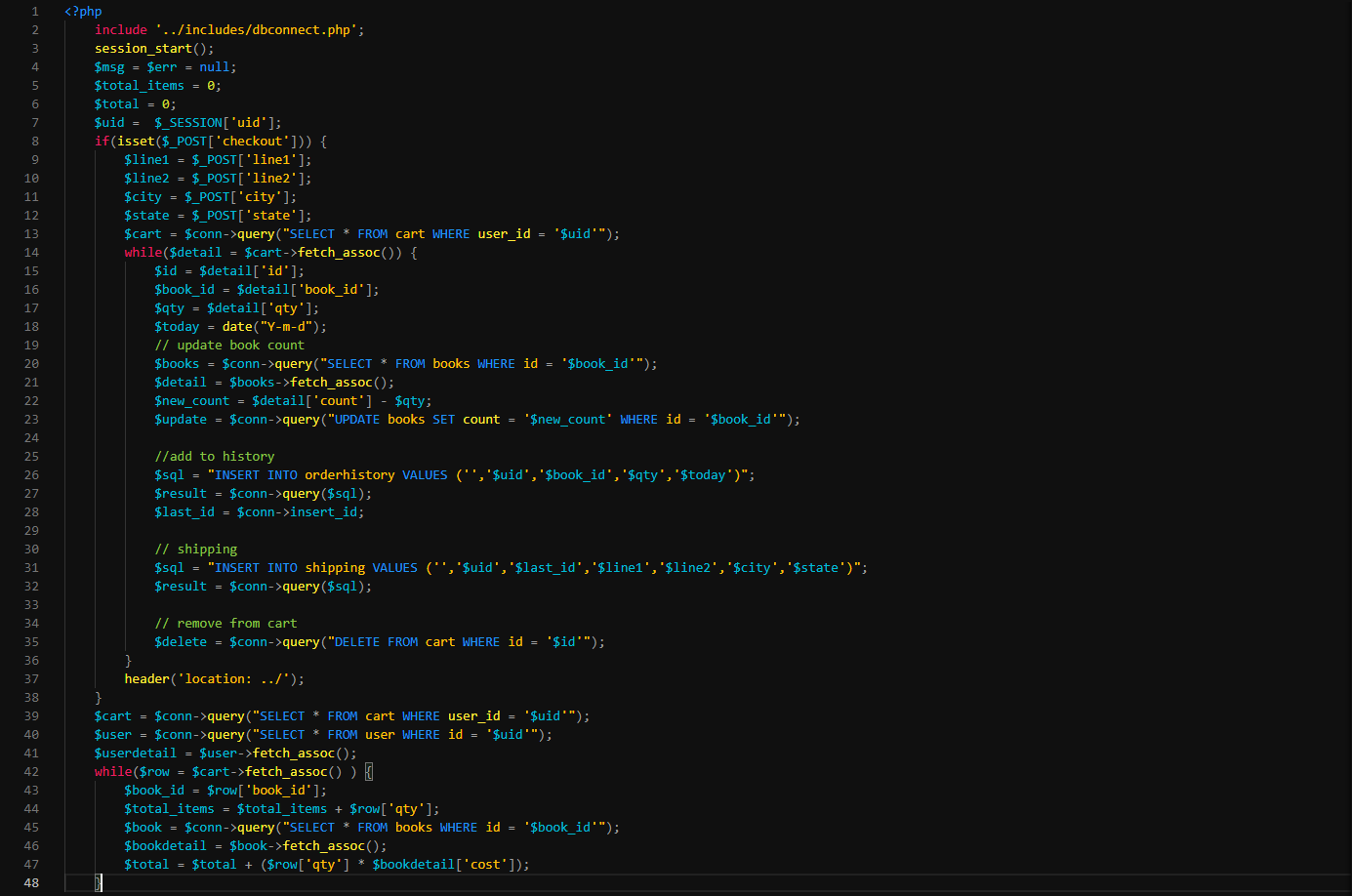
## cart.php

This page shows all the items user has added to the cart. Here, user can update or delete cart items. Total amount the user has to pay is also shown on this page with total number of items on cart. After user presses checkout, they are redirected to the checkout page where they fill up their shipping and card detail.



## checkout.php

This page is last page to place order. Until this page is filled and submitted, order is not placed. Here user enters their shipping address and card info such as card number and cvv.



After user press pay, the order is then placed or the product remains in cart. All items from cart are then moved to another table ‘*orderhistory’* clearing the cart for next purchase.

# Beautify URL

In php, all the GET operation variable is shown in the url as well as the file extension is shown.

i.e.

<http://localhost/a4r/user/profile.php>

<http://localhost/a4r/book.php?slug=zed&id=4>

To hide those elements there are many ways but here we are using ‘*.htaccess’* which customize the url as we want. In this application ‘*.htaccess’* is used to remove those elements.



After using ‘*.htaccess’* new url looks like:

<http://localhost/a4r/user/profile/>

<http://localhost/a4r/book/zed/4/>