



Inspiring Excellence

CSE470 : Software engineering
Project Report
Project Title : E-Pathshala

Table of Contents

Section No	Content	Page No
1	Introduction	
2	Functional Requirements	
3	User Manual	
4	Frontend Development	
5	Backend Development	
6	Technology (Framework, Languages)	
7	Github Repo Link	

Chapter 1: Introduction

E-Pathshala is an online learning platform. The project has been initiated to offer high-quality skill development courses. We expect that this system will attract the young people of Bangladesh to learn new skills to improve their careers at a reasonable price by buying courses related to the skill they like to learn. Students can add their friends on this platform and view which skills their friends are learning to know about those courses. Students can also claim vouchers on their every purchase which are posted by the admin on the platform occasionally. Teachers can publish their courses on this platform. This way a student-teachers friendly learning environment is maintained on this platform. Additionally, the admin role will have special access to maintain the websites.

Chapter 2: Functional and Non-Functional Requirements

Functional

1. Claim vouchers via coupon code
2. View available vouchers
3. Upload vouchers (for admins only)
4. Upload courses for teachers only
5. Search for friends
6. Add and remove friends
7. View friends profile and Own profile
8. View Friend list
9. View own purchased courses
10. Search for courses
11. View Course
12. Add and Remove to Watch later
13. Add and Remove to cart
14. View watch later and Cart
15. Show payslip before Buying courses
16. Payment via Card
17. Download payment receipt in PDF format
18. Show trending products
19. Showing cart Count
20. Available voucher list

Non-functional:

1. No Youtube content is allowed on the site.
2. Only skill related course are allowed
3. No vat on purchase
4. One user can not see others payment information
5. Users can not buy courses without login

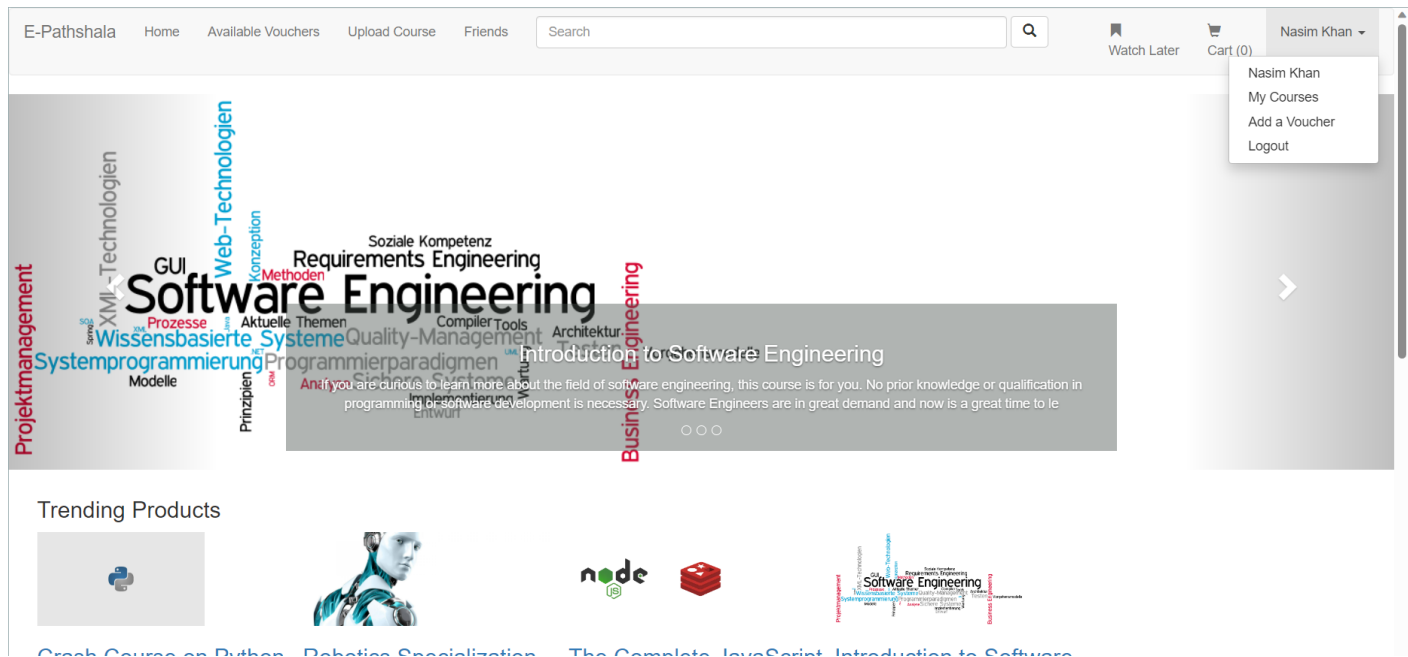
Chapter 3: User Manual

There are three major portions of our web application. There is the Students interface , teachers and admin interface. These interfaces are described below.

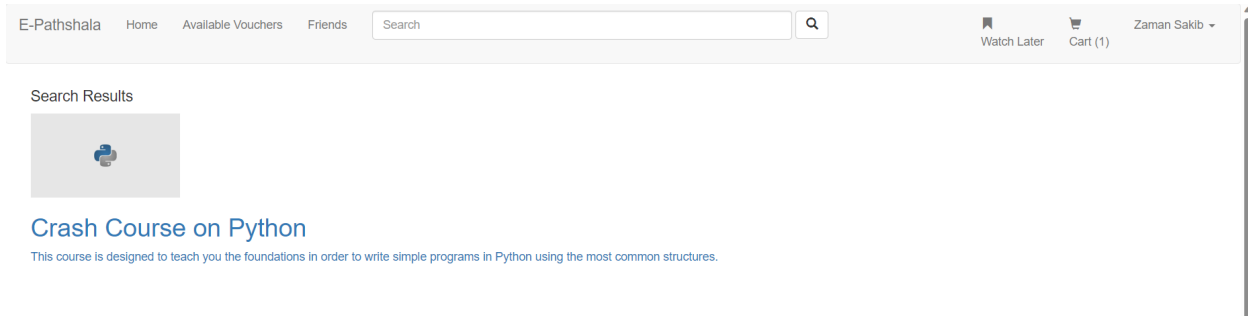
3.1 Student-Side:

Apart from login and logout there are Nine major eight modules for the user side. They are- homepage, courses, search, friends, vouchers, watch later, cart, and payments.

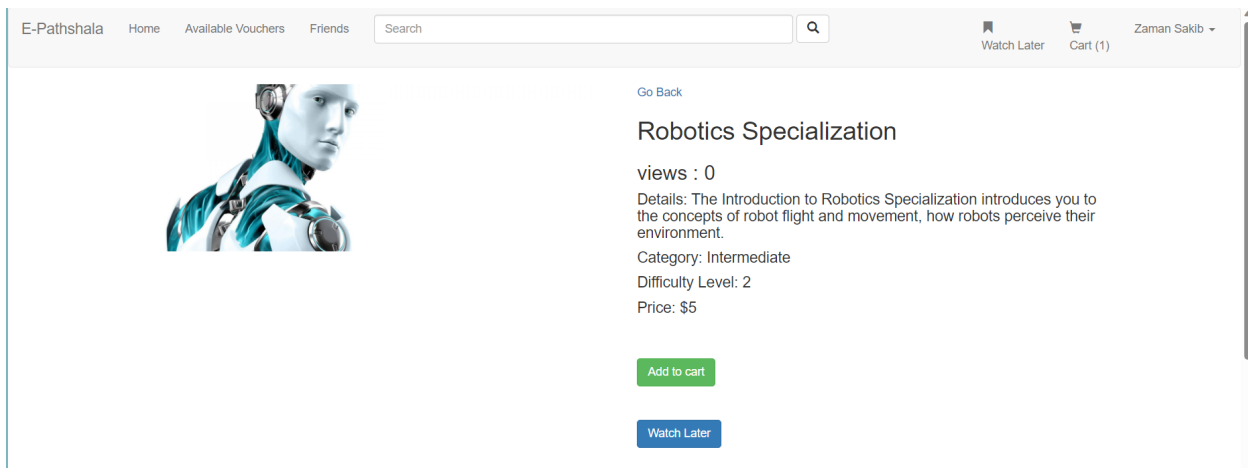
3.1.1 homepage: Both the logged and unlogged users get this homepage when they first enter this website. The homepage contains a navigation bar on top of it. On the navigation bar, unlogged users will only see the login option on their navigation bar. On the other hand, the Homepage for logged user (student interface) consists of slide shows of the suggested courses and trending courses. On the navigation bar they'll have a drop down menu button on the right navigation corner. The title of this button is the name of the user which consists of profile, My course, Add a voucher and Logout button. Beside this drop down menu, students will have a watch later, cart, friends, Available vouchers button and a search box like shown in the figure below.



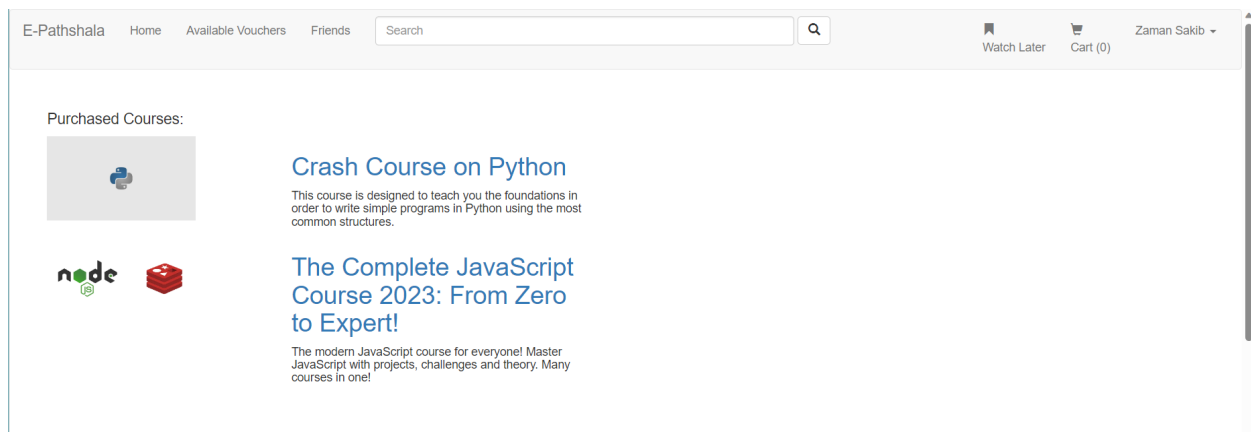
3.1.2 Search courses: Students can search courses by full course name or a part of a course name. Courses similar to the query will be displayed as search results. Example shown below.



3.1.3 Courses: By clicking on the course title or corresponding cover, users can view the course view page where they can see the syllabus and all other details such as category and price. They can add the course to cart and save in watch later.



Students can also view their purchased course from the drop down menu on the top right of the navigation bar.



3.1.4 Friends: To add someone as friends , users can search for them via their email address. By clicking on their profile name, their profile can be visited and there will be add as friend option.

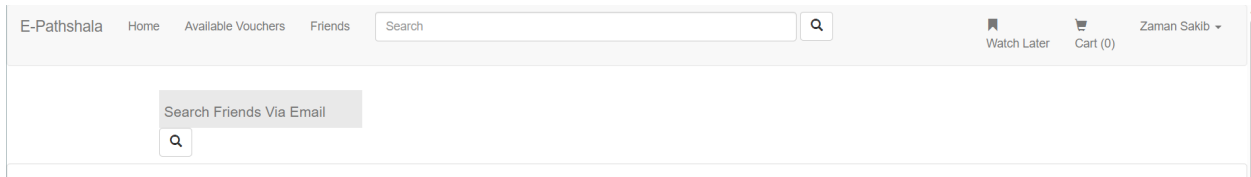


Fig: Search for friends interface

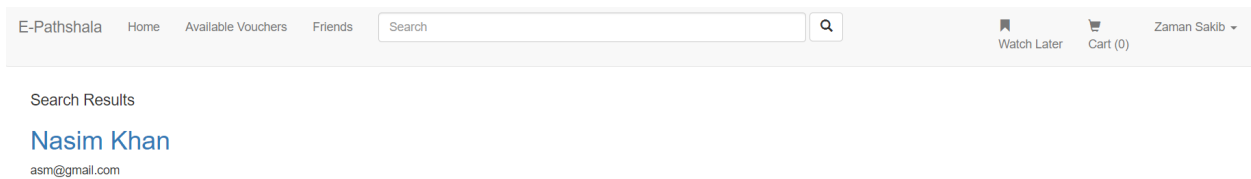


Fig: Search for friends results interface

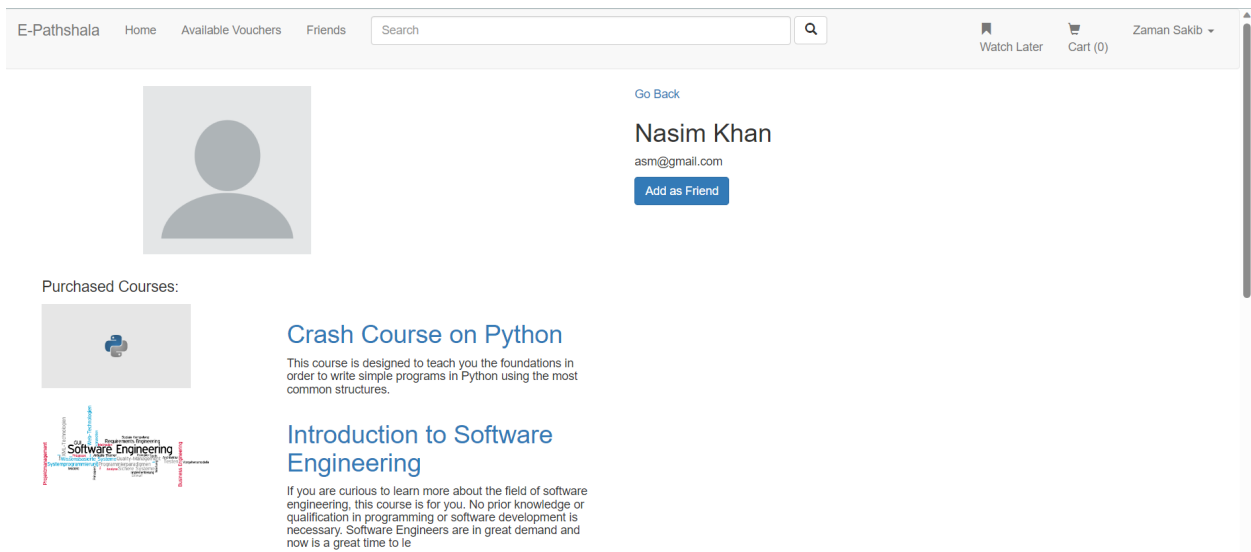


Fig: Friend profile View

Users can show their friend list and remove friends from the list.

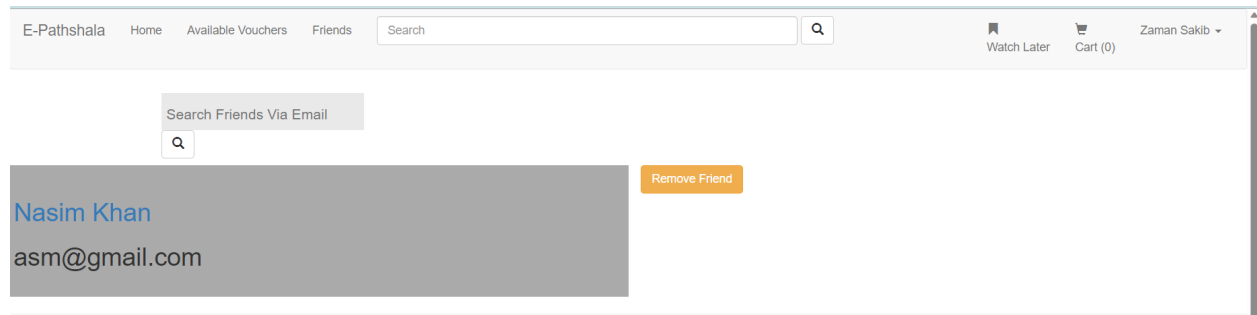


Fig: Friend list view

3.1.5 vouchers: Students can view currently available vouchers from the voucher list. And claim them by entering the coupon code.

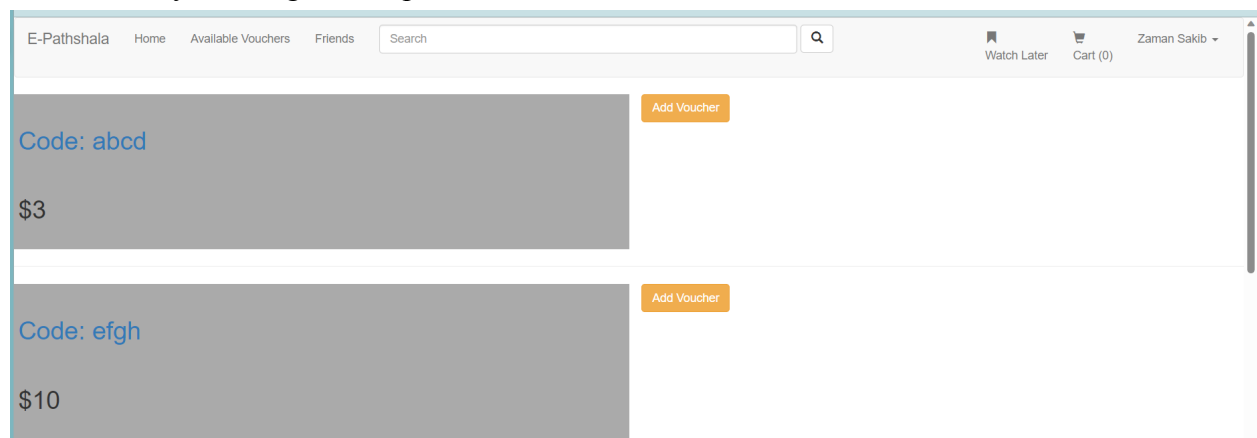


Fig: Voucher list

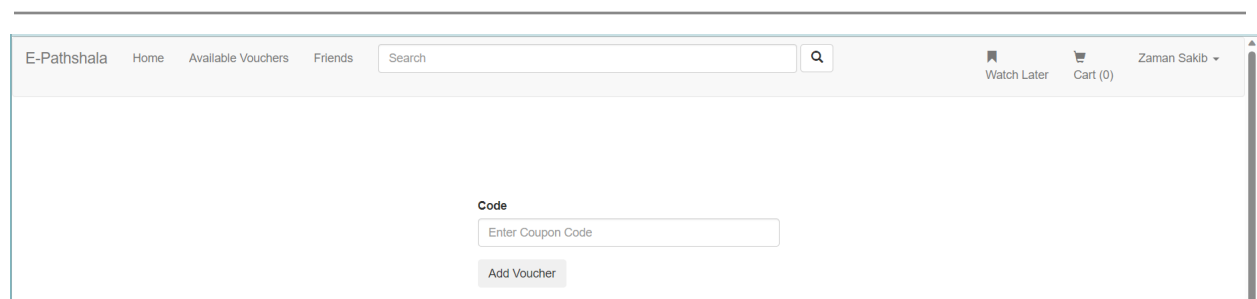


Fig: Claim voucher page

3.1.6 Watch Later: Students can save courses they like to visit later in the watch later section. From the watch later section they can go to the course detail page via clicking on the course cover or course title and remove courses from the watch later.

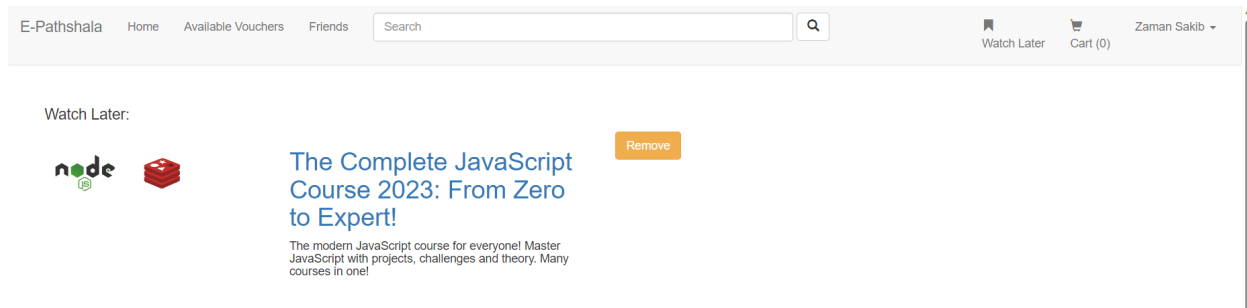


Fig: Watch Later view

3.1.7 Cart: Students can add courses they like to buy in the cart section. From the cart list section they can go to the order page via clicking on the order button to buy the courses added into the cart list and they can also remove courses from cart list.

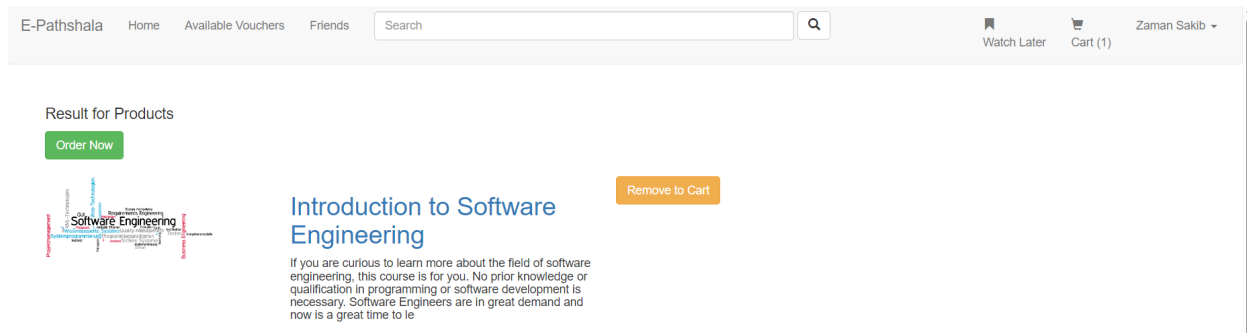


Fig: Cart List view

3.1.8 Payment: To buy courses, students have to order from the cart list. In the order page, they will see the summary of their pay slip with voucher adjustments. They have to give their shipping address and confirm the order.

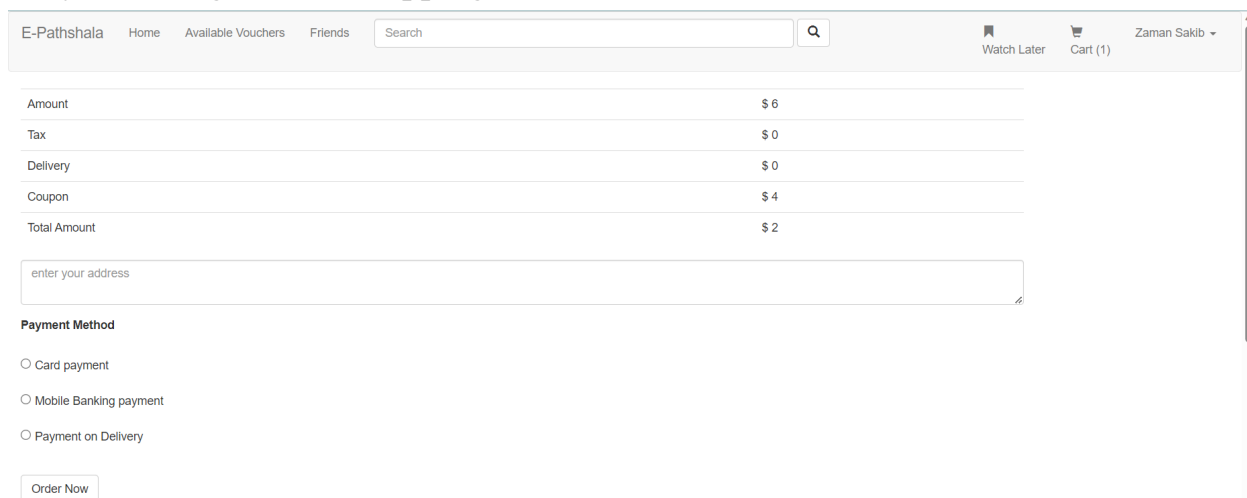
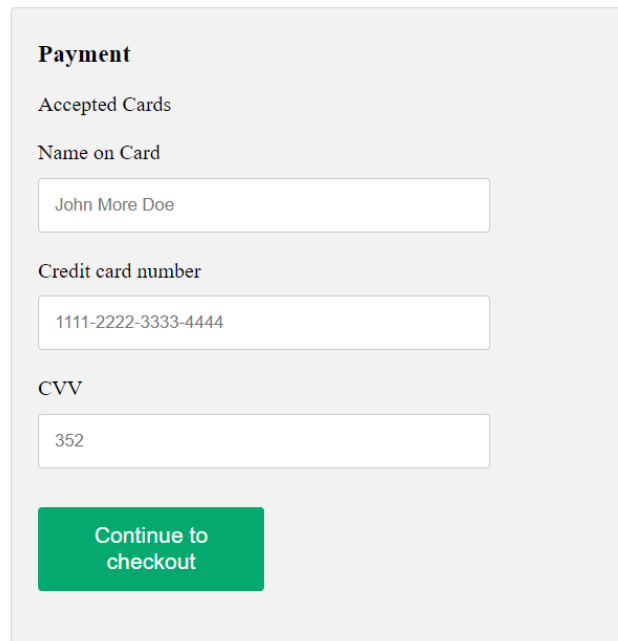


Fig: Order Page View

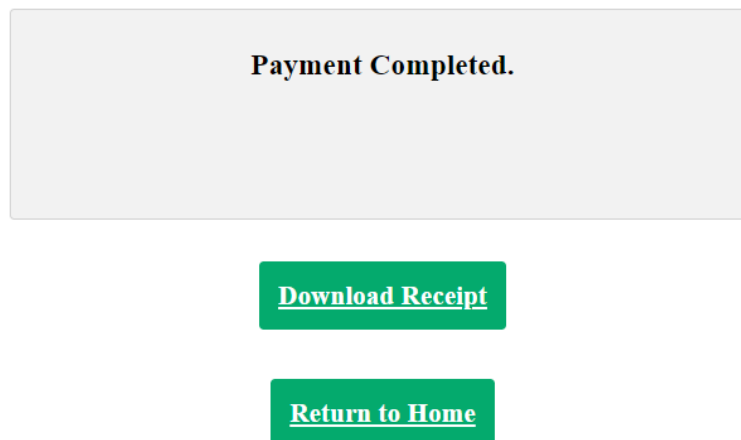
After confirming the order, the chosen payment method will pop up and users can input their payment details to complete the payment procedure.



A payment gateway form with a light gray background. It contains the following elements: a title 'Payment', a label 'Accepted Cards', a label 'Name on Card' with a text input field containing 'John More Doe', a label 'Credit card number' with a text input field containing '1111-2222-3333-4444', a label 'CVV' with a text input field containing '352', and a green button labeled 'Continue to checkout'.

Fig: Payment gateway

After completing the payment, users will get the option to download their payment receipt and return to the homepage.



A payment confirmation page with a light gray background. It features a large gray box with the text 'Payment Completed.' in bold. Below this box are two green buttons: the first is labeled 'Download Receipt' with the text underlined, and the second is labeled 'Return to Home' with the text underlined.

Fig: Payment confirmation page

3.2 Teacher-Side: Teachers can enjoy all the features of a student user and in addition they will be able to see an extra button in their navigation bar. This extra feature is an upload course button.

3.2.1 Upload Course: From the upload course button, users with a teacher's role can upload their own course to publish into the platform. They have to fill some necessary fields before uploading a course.

The screenshot shows the 'Upload Course' page in the E-Pathshala application. The navigation bar at the top includes 'E-Pathshala', 'Home', 'Available Vouchers', 'Upload Course', and 'Friends'. A search bar is located to the right of the navigation bar. On the right side of the navigation bar, there are links for 'Watch Later' and 'Cart (0)', and a user profile for 'Nasim Khan'. The main content area contains a form with the following fields: 'Course Title' (placeholder: 'Enter Course Title'), 'Category' (placeholder: 'What is the Category'), 'Description' (placeholder: 'Describe The content'), 'Difficulty Level' (placeholder: 'Enter 1,2,3 as level'), 'Cover' (placeholder: 'Enter Cover link'), and 'Price' (placeholder: 'Enter Price in \$'). A 'Publish Course' button is located at the bottom of the form.

Fig: Upload course page

3.3 Admin-Side: From an admin view, admin can publish vouchers into the website and remove them.

The screenshot shows the 'Post Voucher' page in the E-Pathshala application. The navigation bar at the top includes 'E-Pathshala', 'Home', 'Post Voucher', and 'Friends'. A search bar is located to the right of the navigation bar. On the right side of the navigation bar, there are links for 'Watch Later' and 'Cart (0)', and a user profile for 'Tushar Khan'. The main content area contains a form with the following fields: 'Coupon' (placeholder: 'Enter Coupon Code') and 'Amount' (placeholder: 'Enter Amount'). A 'Submit' button is located at the bottom of the form.

Chapter 4: Frontend Development

Front-end programming languages used in this web application are- HTML, CSS, Bootstrap and Php. To build the building block of this webpage, HTML has been used. A code snippet has shown below.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Epathshala</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/b

  <script src="https://code.jquery.com/jquery-3.7.0.min.js" integrity="sha256-2Pmv0ku
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/js/bootstrap.min.js"
</head>
<body>
  {{View::make('header')}}
  @yield('content')
  {{View::make('footer')}}

</body>
```

In all the pages, the header and the footer was created in the master. The header shows different buttons for different roles. This was maintained in the header file by checking the logged users role.

```
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
  <ul class="nav navbar-nav">
    <li class=""><a href="/">Home</a></li>
    @if(Session::has('user'))
    @if($user['role'] == 'admin')
      <li class=""><a href="/offer_push">Post Voucher</a></li>
    @else
      <li class=""><a href="/voucher_list">Available Vouchers</a></li>
    @endif
    @if($user['role'] == 'teacher')
      <li class=""><a href="/new_course">Upload Course</a></li>
    @endif
    @endif

    <li class=""><a href="/friends">Friends</a></li>

  </ul>
```

Beside using W3.CSS and bootstrap library, custom CSS file is also being used. Below a code snippet has shown

```
<style>
    .custom-login{
        height: 500px;
        padding-top:100px ;
    }
    img.slider-img{
        height: 400px !important;
    }
    .custom-product{
        height: 600px;
    }
    .slider-text{
        background-color: #49524d6e !important;
    }
    .trending-image{
        height: 100px;
    }
    .trending-item{
        float: left;
        width: 20%;
    }
    .trending-wrapper{
        margin: 30px;
    }
    .detail-img{
        height: 200px;
```

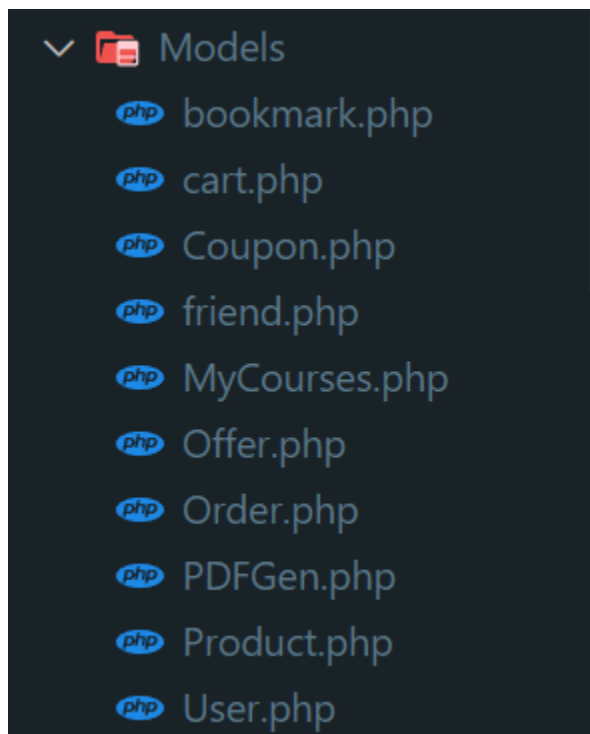
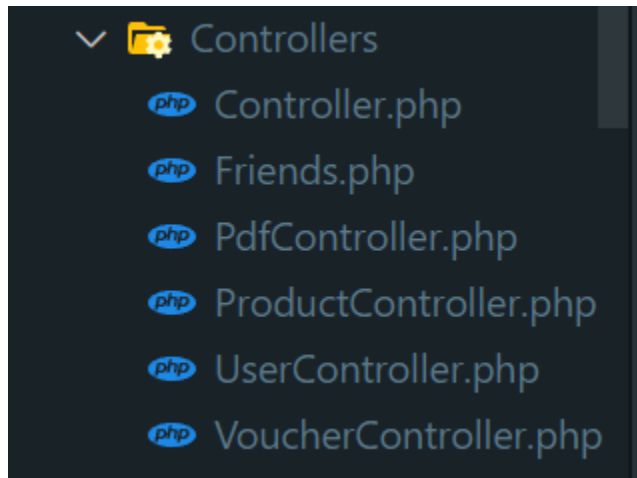
```
    color: white;
}

/* Style the search box inside the navigation bar */
.topnav input[type=text] {
    float: right;
    padding: 6px;
    border: none;
    margin-top: 8px;
    margin-right: 16px;
    font-size: 17px;
}

/* When the screen is less than 600px wide, stack the links and the search field vertically instead of h
@media screen and (max-width: 600px) {
    .topnav a, .topnav input[type=text] {
        float: none;
        display: block;
        text-align: left;
        width: 100%;
        margin: 0;
        padding: 14px;
    }
    .topnav input[type=text] {
        border: 1px solid #ccc;
    }
}
```

Chapter 5: Backend Development

In backend, Laravel framework with php and mysql was used as a local database system. The application was built following the mvc architecture. There are six major controllers with 35 routes and 10 models.



All the features related to friends are handled by the Friends controller. Similarly, cart, product,user, voucher and pdf are handled by their corresponding controllers and model to maintain the mvc architecture.

```
34
35 Route::post('/login', [UserController::class, 'login']);
36 Route::get('/', [ProductController::class, 'home']);
37 Route::get('/detail/{id}', [ProductController::class, 'detail']);
38 Route::get('/profile/{id}', [ProductController::class, 'profile']);
39 Route::get('/userProfile/{id}', [ProductController::class, 'userProfile']);
40 Route::get('/search', [ProductController::class, 'search']);
41 Route::get('/new_course', [ProductController::class, 'new_course']);
42 Route::post("/publish_course",[ProductController::class,'publish_course']);
43 Route::post('/add_to_bookmark', [ProductController::class, 'Addbookmark']);
44 Route::get("removebookmark/{id}",[ProductController::class,'removebookmark']);
45 Route::get("/bookmark",[ProductController::class,'bookmark']);
46 Route::get('/searchFriend', [Friends::class, 'searchFriends']);
47 Route::get('/friends', [Friends::class, 'friends']);
48 Route::get("removefriend/{id}",[Friends::class,'removefriend']);
49 Route::post('/addfriend', [Friends::class, 'Addfriend']);
50 Route::get("/voucher",[VoucherController::class,'voucher']);
51 Route::post("/addvoucher",[VoucherController::class,'addvoucher']);
52 Route::get('/voucher_list', [VoucherController::class, 'voucher_list']);
53 Route::get('/offer_push', [VoucherController::class, 'offer_push']);
54 Route::post("/publish_offer",[VoucherController::class,'publish_offer']);
55 Route::get("/cartlist",[ProductController::class,'cartList']);
56 Route::post('/add_to_cart', [ProductController::class, 'Addcart']);
57 Route::get("removecart/{id}",[ProductController::class,'removeCart']);
58 Route::get("/myCourse",[ProductController::class,'myCourse']);
59 Route::get("/ordernow",[ProductController::class,'orderNow']);
```

Functions like product details, friends profile view, user profile view and search works like the code snippet below.

```
function detail($id){
    $data = Product::find($id);
    return view('details',['product'=>$data]);
}

function profile($id){
    $data = User::find($id);
    $courses = MyCourses::where('user_id', $data['id']->get());
    return view('profile',['profile_data'=>$data,'course'=>$courses]);
}

function userProfile($id){
    $data = User::find($id);
    return view('userProfile',['userProfile'=>$data]);
}

function search(Request $request){

    $query = $request->input('query');
    $data = Product::where('name', 'like', '%' . $query . '%')->get();

    return view('search',['products'=>$data]);
}
```

Cart list and Watch later works in a similar fashion. The difference is they are accessing via different models thus using different tables in the database.

```
function cartList(){

    $userId=Session::get('user')['id'];
    $products= DB::table('saved_tale')
    ->join('products','saved_tale.product_id','=','products.id')
    ->where('saved_tale.user_id',$userId)
    ->select('products.*','saved_tale.id as cart_id')
    ->get();

    return view('cart',['products'=>$products]);
}
```

The Order Now function retrieves all the products that are in the cart, and joins them for ordering. Before doing so, it ensures that the user is logged in, otherwise the application redirects the user to the homepage.

```
function orderNow()  
{  
    $userId=Session::get('user')['id'];  
    $orderAll = cart::where('user_id',$userId)->first();  
    if (isset($orderAll)) {  
        $total= $products= DB::table('saved_tale')  
        ->join('products','saved_tale.product_id','=','products.id')  
        ->where('saved_tale.user_id',$userId)  
        ->sum('products.price');  
  
        return view('ordernow',['total'=>$total]);  
    } else {  
        return redirect('/');  
    }  
}
```

The payment function is shown in the below code snippet.


```

function payment(Request $request){
    // HERE
    $userId=Session::get('user')['id'];
    $allOrder= Order::where('user_id',$userId)->get();
    foreach($allOrder as $cart)
    {
        $order= new MyCourses();
        $order->product_id=$cart['product_id'];
        $order->user_id=$cart['user_id'];
        $order->save();
    }

    $request->input();
    return redirect('/complete_payment');
}

```

The publish course function creates a new object of the product model and input data into the product table from the users.

```

function publish_course(Request $request){
    if($request->session()->has('user')){
        $cart = new Product();
        $cart->name = $request->input('title');
        $cart->catagory = $request->input('catagory');
        $cart->discription = $request->input('discription');
        $cart->difficulty = $request->input('difficulty');
        $cart->cover = $request->input('cover');
        $cart->price = $request->input('price');

        $cart->views = 0;
        $cart->save();

        return redirect('/');
    }
    else{
        return redirect('/login');
    }
}

```

Technology (Framework, Languages)

Framework: Laravel with MVC architecture

Languages: Php

Design : HTML, CSS, Bootstrap

Database system: Xampp, Mysql

Github Repository

Link: <https://github.com/asm-nasim-khan/Epathshala>