



REPORT MANUAL OF FINAL PROJECT EXPLANATION AND TUTORIAL (ADVANCED WEB PROGRAMMING)

Name	:	Brian Sayudha & Rakotoaritsoa Fenohasina
Class / NIM	:	2G / 1841720158 2G / 1841720221
Major	:	D-IV Informatics Engineering

Admin (Using Laravel)

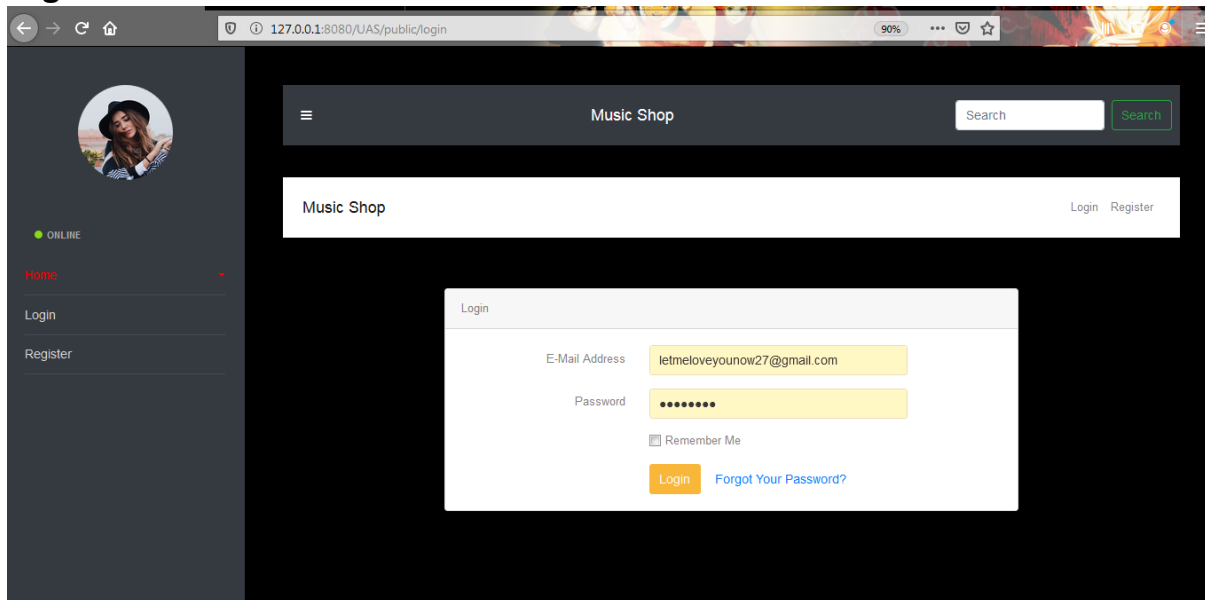
Introduction:

Our project is a Music shop which sells instruments and the front end is made with Code Igniter and the back End is made with Laravel.

Back End or Admin site:

Features:

Login:



The login looks like this. No feature works in here before the admin logs in. So we need to log in first before being able to change anything.

We can also register.

Music Shop

Search

Register

Music Shop

Login Register

Register

Name

E-Mail Address

Password

Confirm Password

Register

Once Logged in, we can browse :

Music Shop

Search

Add Bass Data

Instrument List

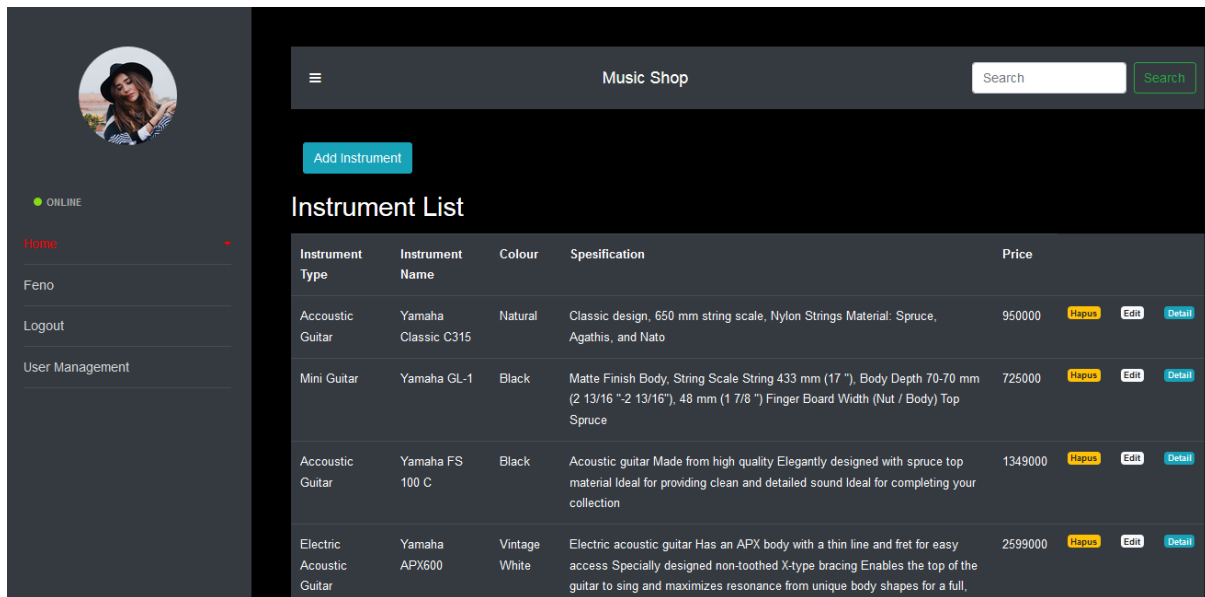
Instrument Type	Instrument Name	Colour	Spesification	Price
Accoustic Guitar	Yamaha Classic C315	Natural	Classic design, 650 mm string scale, Nylon Strings Material: Spruce, Agathis, and Nato	950000 Hapus Edit Detail
Mini Guitar	Yamaha GL-1	Black	Matte Finish Body, String Scale String 433 mm (17 "), Body Depth 70-70 mm (2 13/16 ~2 13/16"), 48 mm (1 7/8 ") Finger Board Width (Nut / Body) Top Spruce	725000 Hapus Edit Detail
Accoustic Guitar	Yamaha FS 100 C	Black	Acoustic guitar Made from high quality Elegantly designed with spruce top material Ideal for providing clean and detailed sound Ideal for completing your collection	1349000 Hapus Edit Detail
Electric Acoustic Guitar	Yamaha APX600	Vintage White	Electric acoustic guitar Has an APX body with a thin line and fret for easy access Specially designed non-toothed X-type bracing Enables the top of the guitar to sing and maximizes resonance from unique body shapes for a full	2599000 Hapus Edit Detail

Menu:

The menus are:

- Home
- Guitar
- Piano
- Bass
- Violin
- Logout and User management.

We can also do a searcth and add new instruments and make CRUD.



Here is the list of all guitar:

The code:

To explain it, we use a master as a template. Here is the master:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
  <link rel="stylesheet" href="http://localhost:8080/UAS/assets/css/style.css">
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
</head>

<body bgcolor="#000000">
  <div class="wrapper d-flex align-items-stretch">
    <nav id="sidebar">
      <div class="p-4 pt-5">
        <a href="#" class="img logo rounded-circle mb-5" style="background-image: url(http://127.0.0.1:8080/UAS/assets/img/logo.png);"></a>
      </div>
      <div class="profile-usertitle">
        <div class="profile-usertitle-name">
          <div></div>
        </div>
        <div class="profile-usertitle-status"><span class="indicator label-success"></span>Online</div>
      </div>
      <div class="clear"></div>
      <div class="divider"></div>
      <a href="#" sidebar></a>
      <ul class="list-unstyled components mb-5">
```

```

@guest
<li class="nav-item">
  <a class="nav-link" href="{{ route('login') }}">{{ __('Login') }}</a>
</li>
  @if (Route::has('register'))
    <li class="nav-item">
      <a class="nav-link" href="{{ route('register') }}">{{ __('Register') }}</a>
    </li>
  @endif
  @else
    <li class="nav-item dropdown">
      <a id="navbarDropdown" href="#" role="button" data-toggle="dropdown" aria-expanded="false">
        {{ Auth::user()->name }} <span class="caret"></span>
      </a>
      <div aria-labelledby="navbarDropdown">
        <a class="dropdown-item" href="{{ route('logout') }}">
          {{ __('Logout') }}
        </a>
        <a href="{{ route('users.index') }}">
          User Management
        </a>
        <form id="logout-form" action="{{ route('logout') }}" method="POST">
          @csrf
        </form>
      </div>
    </li>
  @endguest
</nav>

<div id="content" class="p-4 p-md-5">
  <nav class="navbar navbar-dark bg-dark">
    <button type="button" id="sidebarCollapse" class="btn btn-dark">
      <i class="fa fa-bars"></i>
      <span class="sr-only">Toggle Menu</span>
    </button>
    <a class="navbar-brand" href="#">Music Shop</a>
    <form class="form-inline my-2 my-lg-0" action="http://127.0.0.1:8080/UAS/public/bass/search" method="get">
      <input class="form-control mr-sm-2" type="search" placeholder="Search" name="search">
      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
    </form>
  </nav>

  <!-- End of navigation bar -->

  <!-- content -->

  @yield('konten')

  <!-- end of content -->
</div>
</body>

```

Then, Here is the code to print all the guitars:
The controller:

```

public function guitar(){
    $search = 'Guitar';
    $search2 = 'G';
    $item= \DB::table('item')
    ->where('inst_type', 'LIKE', '%'.$search.'%')
    ->orWhere('item_id', 'LIKE', '%'.$search2.'%')
    ->get();
    return view('bassindex',['item' => $item]);
}

```

The View:

```

@extends('master')

<!-- Fille the content -->
@section('konten')
<div class="container">

<div class="row mt-3">
    <div class="col-md-6">
        <a href="admin/bass/add" class="btn btn-info">Add Instrument </a>
    </div>
</div>

<div class="row mt-3">
    <div class="col-md-20">
        <h2 style="color: white">Instrument List</h2>
        <table class="table table-striped table-dark">
            <tr>
                <th scope="col">Instrument Type</th>
                <th scope="col">Instrument Name</th>
                <th scope="col">Colour</th>
                <th scope="col">Spesification</th>
                <th scope="col">Price</th>
            </tr>
            @foreach($item as $itm)
                <tr>
                    <td>{{ $itm->inst_type }}</td>
                    <td>{{ $itm->item_name }}</td>
                    <td>{{ $itm->color }}</td>
                    <td>{{ $itm->spec }}</td>
                    <td>{{ $itm->price }}</td>
                    <td> <a href="http://127.0.0.1:8080/UAS/public/bass/hapus/{{ $itm->item_id }}" class="badge badge-danger">Hapus</a>
                    <td> <a href="http://127.0.0.1:8080/UAS/public/bass/edit/{{ $itm->item_id }}" class="badge badge-warning">Edit</a>
                    <td> <a href="http://127.0.0.1:8080/UAS/public/bass/detail/{{ $itm->item_id }}" class="badge badge-info">Detail</a>
                </tr>
            @endforeach
        </table>
    </div>
</div>
</div>

```

The view is the same for all the instrument types but the function in the controller are different like we can see in here:

```

public function guitar(){
    $search = 'Guitar';
    $search2 = 'G';
    $item= \DB::table('item')
    ->where('inst_type', 'LIKE', '%'.$search.'%')
    ->orWhere('item_id', 'LIKE', '%'.$search2.'%')
    ->get();
    return view('bassindex',['item' => $item]);
}

public function piano(){
    $search = 'violin';
    $search2 = 'P';
    $item= \DB::table('item')
    ->where('inst_type', 'LIKE', '%'.$search.'%')
    ->orWhere('item_id', 'LIKE', '%'.$search2.'%')
    ->get();
    return view('bassindex',['item' => $item]);
}

public function violin(){
    $search = 'violin';
    $search2 = 'V';
    $item= \DB::table('item')
    ->where('inst_type', 'LIKE', '%'.$search.'%')
    ->orWhere('item_id', 'LIKE', '%'.$search2.'%')
    ->get();
    return view('bassindex',['item' => $item]);
}

```

To make the search, we needed to create a search method in the controller:

```

public function search(Request $request){
    $search= $request->get('search');
    $item= \DB::table('item')
    ->where('inst_type', 'LIKE', '%'.$search.'%')
    ->orWhere('item_name', 'LIKE', '%'.$search.'%')
    ->get();
    return view('bassindex',['item' => $item]);
}
}

```

And for the CRUD, Here are the codes:

ADD:

View:

```

</div>
<div class="card-body">
  @if(count($errors)>0)
    <div class="alert alert-danger" role="alert">
      <ul>
        @foreach($errors->all() as $error)
          <li>{{ $error }}</li>
        @endforeach
      </ul>
    </div>
  @endif

  <form action="http://127.0.0.1:8080/UAS/public/bass/simpan" method="POST">
    {{csrf_field()}}
    <div class="form-group">
      <label for="item_id">Item id</label>
      <input type="text" name="item_id" class="form-control" id="item_id" placeholder="Item Id">
    </div>
    <div class="form-group">
      <label for="inst_type">Bass type</label>
      <select class="form-control" name="inst_type" id="inst_type" >
        @foreach($bass_inst_type as $key)
          <option value="{{ $key }}" selected>{{ $key }}</option>
        @endforeach
      </select>
    </div>
    <div class="form-group">
      <label for="item_name">Bass Name</label>
      <input type="text" name="item_name" class="form-control" id="item_name" placeholder="Instrument Name">
    </div>
    <div class="form-group">
      <label for="color">Colour</label>
      <select class="form-control" name="color" id="color">
        @foreach($bass_color as $key)

```

Here is the Controller:

```

public function add(){
    $bass_inst_type = ['Double Bass','Electric Bass','Accoustic Bass','Semi-Accoustic'];
    $bass_color = ['Brown','Orange','Natural','Black','White','Black Red','Light Blue','Yellow'];
    return view('bassadd',['bass_inst_type' => $bass_inst_type , 'bass_color' => $bass_color]);
}

public function simpan(Request $request){

    $this->validate($request,[
        'item_id' => 'required',
        'inst_type' => 'required',
        'item_name' => 'required',
        'color' => 'required',
        'spec' => 'required',
        'price' => 'required|numeric'
    ]);

    \DB::table('item')->insert([
        'item_id' => $request->item_id,
        'inst_type' => $request->inst_type,
        'item_name' => $request->item_name,
        'color' => $request->color,
        'spec' => $request->spec,
        'price' => $request->price
    ]);
    //Take data from the Mahasiswa Table in Database:
    $item = \DB::table('item')->get();

    //Send data to view:
    return view('bassindex',['item' => $item]);
}

```

We have a form validation in the add method

The result:

Form Adding Bass Data

Item id

Item Id

Bass type

Semi-Accoustic

Bass Name

Instrument Name

Colour

Dark Blue

Specification

Speculation

Price

Price of the Instrument

Go back

Submit

Form validation:

Form Adding Bass Data

- The price field is required.

Item id

Item Id

Bass type

Semi-Accoustic

Bass Name

Instrument Name

Colour **Dark Blue**

Specification

Speculation

Form Adding Bass Data

- The price field is required.

Item id

1233

Bass type

Electric Bass

Bass Name

Bass be

Colour **Dark Blue**

Specification

High Pitch

Price

60000

[Go back](#)

[Submit](#)

Type	Name					
Semi-Accoustic	DS55555	Dark Blue	SDSDS	255554	Hapus	Edit Detail
Electric Bass	Bass be	Dark Blue	High Pitch	60000	Hapus	Edit Detail
Double Bass	Bass Buffle	Natural	Big bass profound	500000	Hapus	Edit Detail
Acoustic Bass	Yamaha	Natural	Classical	650000	Hapus	Edit Detail

Details:

Bass Details

DS55555

Instrument Type : Semi-Accoustic

Color : Dark Blue

Spesification : SDSDS

Price : Rp. 255554

Go back

Code:

```

files / views / bassdetail.php
@extends('master')

<!-- Fille the content -->
@section('konten')

<div class="container">
<div class="row mt-3">
<div class="col-md-6">
<div class="card">
<div class="card-header">
    Bass Details
</div>
@foreach($bass as $bass)
<div class="card-body">
<h5 class="card-title">{{ $bass->item_name }}</h5>
<p class="card-text">
<label for=""><b>Instrument Type</b> : </label>
        {{ $bass->inst_type }}
    </p>
<p class="card-text">
<label for=""><b>Color</b> : </label>
        {{ $bass->color }}
    </p>
<p class="card-text">
<label for=""><b>Spesification</b> : </label>
        {{ $bass->spec }}
    </p>
<p class="card-text">
<label for=""><b>Price</b> : Rp.</label>
        {{ $bass->price }}
    </p>
<a href="http://127.0.0.1:8080/UAS/public/bass" class="btn btn-info">Go back</a>
    </div>
</div>
@endforeach
</div>
</div>
</div>
</div>
@endsection

```

Controller:

```

public function detail($id){

    $bass = \DB::table('item')->where('item_id',$id)->get();

    return view('bassdetail', ['bass' => $bass]);

}

public function edit($id){

```

Delete:

Controller:

```

public function hapus($id){

    $bass = \DB::table('item')->where('item_id',$id)->delete();

    $item = \DB::table('item')->get();

    //Send data to view:
    return view('bassindex',['item' => $item]);

}

```

Edit:

Controller:

```

public function edit($id){
    $bass_inst_type = ['Double Bass','Electric Bass','Accoustic Bass','Semi-Accoustic'];
    $bass_color = ['Brown','Orange','Natural','Black','White','BlacK Red','Light Blue','Yellow','Mix','Blue','Dark Blue'];
    $bass = \DB::table('item')->where('item_id',$id)->get();

    return view('bassedit', ['bass' => $bass, 'bass_inst_type' => $bass_inst_type , 'bass_color' => $bass_color]);
}

public function update(Request $request){

    \DB::table('item')->where('item_id',$request->item_id)->update([
        'inst_type' => $request->inst_type,
        'item_name' => $request->item_name,
        'color' => $request->color,
        'spec' => $request->spec,
        'price' => $request->price
    ]);

    //Take data from the Mahasiswa Table in Database:
    $item = \DB::table('item')->get();

    //Send data to view:
    return view('bassindex',['item' => $item]);

}

```


View:

```

('master')

le the content -->
('konten')
ss="container"
class="row mt-3"
<div class="col-md-6">
  <div class="card">
    <div class="card-header">
      Form Editing Bass Data
    </div>
    <div class="card-body">
      @if(count($errors)>0)
        <div class="alert alert-danger" role="alert">
          <ul>
            @foreach($errors->all() as $error)
              <li>{{ $error }}</li>
            @endforeach
          </ul>
        </div>
      @endif
      <form action="http://127.0.0.1:8080/UAS/public/bass/update" method="POST">
        {{csrf_field()}}
        @foreach($bass as $bass)
          <input type="hidden" name="item_id" value="{{ $bass->item_id }}"><br>
          <div class="form-group">
            <label for="inst_type">Bass type</label>
            <select class="form-control" name="inst_type" id="inst_type" >
              @foreach($bass_inst_type as $key)
                <option value="{{ $key }}" selected="{{ $key }}"></option>
              @endforeach
            </select>
          </div>
          <div class="form-group">
            <label for="item_name">Bass Name</label>
            <input type="text" name="item_name" class="form-control" id="item_name" placeholder="Instrument Name" value="{{ $bass->item_name }}">
          </div>
          <div class="form-group">
            <label for="color">Colour</label>
            <select class="form-control" name="color" id="color">
              @foreach($bass_color as $key)
                <option value="{{ $key }}" selected="{{ $key }}"></option>
              @endforeach
            </select>
          </div>
          <div class="form-group">
            <label for="spec">Specification</label>
            <input type="text" name="spec" class="form-control" id="spec" placeholder="Speculation" value="{{ $bass->spec }}">
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

```

Music Shop

Form Editing Bass Data

Bass type

Semi-Acoustic

Bass Name

Yamaha Classic C315

Colour

Dark Blue

Specification

Classic design, 650 mm string scale, Nylon StringsMaterial: Spruce

Price

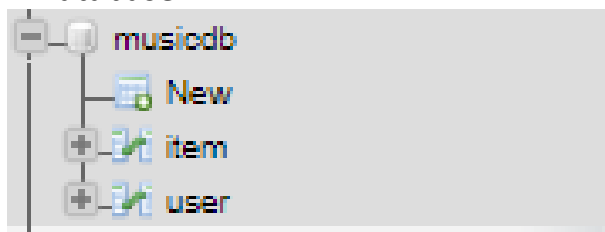
950000

Go back

Submit

CI Client (Using code igniter php)

I. Database



The database name is Musicdb and it contains 2 table, item and user

- Table Item

Showing rows 0 - 14 (15 total, Query took 0.0010 seconds.)

SELECT * FROM 'item'

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	item_id	inst_type	item_name	color	spec	price	image
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	G001	Acoustic Guitar	Yamaha Classic C315	Natural	Classic design, 850 mm string scale, Nylon Strings...	950000	yamaha_yamaha-cs-40-gitar-klasik--natural-.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	G002	Mini Guitar	Yamaha GL-1	Black	Matte Finish Body, String Scale String 433 mm (17 ...	725000	yamaha-gl1.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	G003	Acoustic Guitar	Yamaha FS 100 C	Black	Acoustic guitar Made from high quality Elegantly...	1349000	yamaha-fs100c.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	G004	Electric Acoustic Guitar	Yamaha APX800	Vintage White	Electric acoustic guitar Has an APX body with a t...	2599000	yamaha_yamaha-electric-acoustic-guitar-apx-800-vin...
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	G005	Electric Guitar	Yamaha Electric Guitar	Black	Electric Guitar Construction: Bolt-on, Body: Agat...	5900000	yamaha-revstar-rs420-black.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	K001	Portable Keyboard	Yamaha PSR-SX900	Black	With a brilliant response produces clear, high-qua...	19000000	Yamaha PSR-SX900.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	K002	Synthesizer Keyboard	Roland Juno DS88	Black	Instrument equipped with pro sound, added features...	15160000	roland_roland-juno-ds88.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	K003	Keyboard	Casio CTK 6250	Black Red	User Tones 10 tone editor Built-in Rhythms 210 U...	3500000	casio-ctk-6250-.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	P001	Digital Piano	Yamaha P 515	Black	P-515 Digital Piano Equipped with two advanced te...	20700000	yamaha_musik_indonesia_yamaha_p_515_digital_piano...
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	P002	Slim Digital Piano	CASIO CDP S350	Black	Digital Casio Piano Only 23cm wide Equipped with...	8900000	casio_casio-cdp-s350-slim-digital-piano.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	U001	Smart Ukulele	Xiaomi U1	Brown	Smart Ukulele Elegantly designed 8000 ...	1900000	xiaomi_xiaomi_populele_smart_Ukulele_u1.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	U002	Ukulele	Mahalo Rainbow Ukule	Light Blue	Ukulele guitar Use 4 string strings Able to reso...	410000	mahalo_rainbow_ukulele.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	U003	Ukulele	Mahalo Soprano Halow	Orange	Body Sengon Neck Jabon Fingerboard Mahogany Bri...	77000	Mahalo_soprano_hollow.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	V001	Violin	ASTGHIK VF 102A	Brown	Violin Producing high quality instrumental string...	2800000	astghik_violin-vf-102a.jpg
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	V002	Violin	LULEA Tipe Solidwood	Brown	Violin Made of solid wood & and sandalwood Styl...	1115000	lulea_lulea-tipe-solidwood-4-4-violin.jpg

Table item contain the instrument that the music shop will sell, the item table contain item_id, instrument type, instrument name, color of the instrument, spec or description per item, price, and image for instrument

- User table

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

SELECT * FROM 'user'

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key:

	id	username	password	level
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	admin	admin	admin
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	user	user	user
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	Brian	user	user

From this user table is contain the user that login in the music shop store

II. Login

1. Controller

This controller will control the login page, from this we set the login , process, and logout. At the proses_login we can see that our input will be processed to get the data, if the level user is admin, will go to the admin, then for the user, will go to the user. Admin can't go to user url, and user can't go to the admin url. All of the data must through the login


```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class login extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');
        $this->load->helper('form');
        $this->load->model('login_model');
        $this->load->library('form_validation');
        $this->load->library('session');
    }

    public function index()
    {
        $data['title'] = 'Login Form';
        $this->load->view('template/header_login', $data);
        $this->load->view('login/index');
        $this->load->view('template/footer');
    }

    public function proses_login()
    {
        $username=htmlspecialchars($this->input->post('uname1'));
        $password=htmlspecialchars($this->input->post('pwd1'));

        $ceklogin=$this->login_model->login($username,$password);

        if($ceklogin){
            foreach($ceklogin as $row);

            $this->session->set_userdata('user', $row->username);
            $this->session->set_userdata('level', $row->level);

            if($this->session->userdata('level')== "admin"){
                redirect('admin/guitar');
            }
        }
    }
}
```

```

        }
        elseif($this->session->userdata('level') == "user"){
            redirect('user/guitar/');
        }
        else{
            redirect('login/index', 'refresh');
        }
    }
    else {
        $data['message'] = "Username And/Or Password Wrong";
        $data['title'] = 'Login';
        $this->load->view('template/header_login', $data);
        $this->load->view('login/index');
        $this->load->view('template/footer');
    }
}

public function logout()
{
    $this->session->sess_destroy();
    redirect('login', 'refresh');
}
}

/* End of file login.php */
?>

```

2. Model

This model to get the input data and check it with database or user that registered (user table)

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class login_model extends CI_Model {

    function login($username,$password){
        $this->db->select('username,password,level');
        $this->db->from('user');
        $this->db->where('username', $username);
    }
}

```

```

        $this->db->where('password', $password);
        $this->db->limit(1);

        $query=$this->db->get();
        if($query->num_rows()==1){
            return $query->result();
        }else{
            return false;
        }
    }
}

/* End of file login_model.php */

?>

```

3.Views

This views is to set the interface of the login,

```

<?= form_open('login/proses_login'); ?>
<div class="login-dark">
    <div class="container py-5">
        <div class="row">
            <div class="col-md-12 ">
                <div class="row">
                    <div class="col-lg-6 mx-auto">
                        <!-- form card login -->
                        <div class="card text-white border-info bg-
dark mb-3 ">
                            <div class="card-header border-info">
                                <h3 class="mb-0">Login</h3>
                            </div>
                            <div class="illustration"><i class="icon i
on-ios-locked-outline"></i></div>
                            <div class="alert alert-
info" role="alert">
                                <?php
                                    if(isset($message)){
                                        echo $message;
                                    }
                                    else{

```

```

        echo "Please Insert your username and
password";
    }
    ?>
</div>
    <div class="card-body">
        <form class="form" role="form" autocomplete="off" id="formLogin" novalidate="" method="POST">
            <label for="uname1">Username</label>
            <div class="input-group">
                <div class="input-group-prepend">
                    <div class="input-group-text"><i class="fa fa-user"></i></div>
                </div>
                <input type="text" class="form-control form-control-lg rounded-0" name="uname1" id="uname1" placeholder="username" required="">
            </div>
            <div class="form-group">
                <div class="invalid-feedback">Oops, you missed this one.</div>
            </div>
            <label>Password</label>
            <div class="input-group">
                <div class="input-group-prepend">
                    <div class="input-group-text"><i class="fa fa-key"></i></div>
                </div>
                <input type="password" class="form-control form-control-lg rounded-0" id="pwd1" name="pwd1" placeholder="*****" required="" autocomplete="new-password">
            </div>
            <div class="invalid-feedback">Enter your password too!</div>
            <br>

```

```

        <button type="submit" class="btn btn-
outline-success btn-block active" id="btnLogin">Login</button>
        </form>
    </div>
<!--/card-block-->
</div>
<!-- /form card login -->

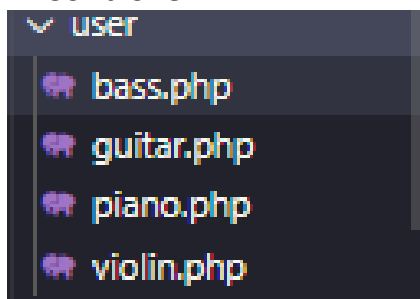
    </div>
</div>
<!--/row-->

    </div>
<!--/col-->
</div>
<!--/row-->
</div>
<!--/container-->
</div>
<?= form_close() ?>

```

III. CodeIgniter user

1. Controller



There are some controllers in our CI user, the controller is separated the types of the music item database, because at the database the item is put together, we separate it per type to get the user more comfortable and easier to find some instrument

1.2. Controller Code

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

```

```

class guitar extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->
>API = 'http://localhost/UTS_web/CI_client/api/guitarAPI';
        $this->load->helper('url');
        $this->load->helper('form');
        $this->load->model('guitar_model');
        $this->load->library('Curl');
        $this->load->library('session');

        if($this->session->userdata('level')!="user"){
            redirect('login','refresh');
        }

    }

    //Get data with api
    function index() {
        if($this->session->userdata('level') == "user"){

            $result = $this->curl->simple_get($this->API);

            $guitar = [
                'guitar' => json_decode($result,true),
                'title'   => 'Guitar',
            ];

            $this->load->view('template/header_user', $guitar);
            $this->load->view('user/guitar/index', $guitar);
            $this->load->view('template/footer_user');
        }else{
            redirect(base_url());
        }
    }

    // for function button to get done

```

```

    public function detail($id)
    {
        $data['title']='Detail Data';
        $data['guitar'] = $this->guitar_model->getGuitarID($id);
        $this->load->view('template/header_user', $data);
        $this->load->view('user/guitar/detail', $data);
        $this->load->view('template/footer_user');
    }

    public function buy($id)
    {
        $data['title']='Buy Guitar';
        $data['guitar'] = $this->guitar_model->getGuitarID($id);
        $this->load->view('template/header_user', $data);
        $this->load->view('user/guitar/buy', $data);
        $this->load->view('template/footer_user');
    }
}

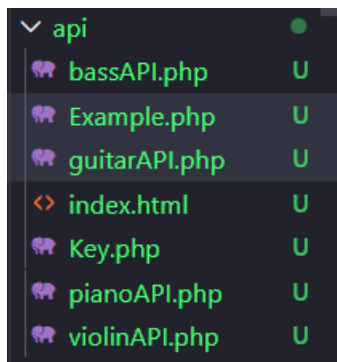
```

End of file guitar.php */

From this we can see this is the guitar controller, its contain to control the guitar views only. Because this is only user website, the controller only contain get data and detail per item only. For this controller, we edit the index get data, so we can get the data from database with restful API. We used curl library to get the data and decode it with json_decode. This controller will redirect to API controller that will get the data from model.

All of controller is same but the different is only at the models and the views url. Per controller have their own models (will be discussed next) for example guitar have guitar model, bass have bass model, etc.

1.3. Controller API



Code for guitarAPI.php

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . 'libraries/REST_Controller.php';
require APPPATH . 'libraries/Format.php';

class guitarAPI extends REST_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');
        $this->load->helper('form');
        $this->load->model('login_model');
        $this->load->model('guitar_model');
        $this->load->library('form_validation');
        $this->load->library('session');
    }
    //API GET
    public function index_get()
    {
        $id = $this->get('item_id');
        if($id === null){
            $guitar = $this->guitar_model->getGuitarData();
        }else{
            $guitar = $this->guitar_model->getGuitarData($id);
        }
    }
}
```



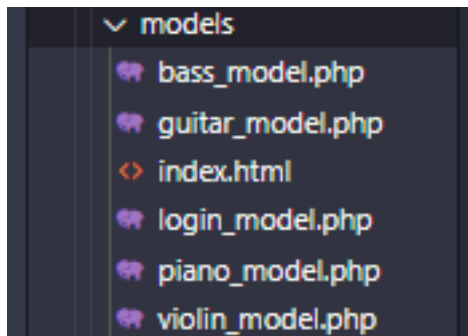
```

        if($guitar){
            $this->response([
                'status' =>true,
                'data' => $guitar
            ], REST_Controller::HTTP_OK);
        }else{
            $this->response([
                'status' => false,
                'data' => 'id not found'
            ], REST_Controller::HTTP_NOT_FOUND);
        }
    }
}

```

From this code, we get the API to get the data from model

4. Models



```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class guitar_model extends CI_Model {

    // Get the data guitar in the database
    public function getGuitarData()
    {
        $this->db->like('item_id','g');
        $this->db->or_like('item_id','u');
        return $this->db->get('item')->result_array();
    }

    // Get the data description from id

```

```

        public function getGuitarID($id)
        {
            return $this->db->get_where('item', ['item_id'=> $id])-
>row_array();
        }
        public function searchData()
        {
            $keyword=$this->input->post('keyword');
            $this->db->like('item_name', $keyword);
            return $this->db->get('item')->result_array();
        }
    }
    /* End of file guitar_model.php */
    ?>

```

From This we can see the guitar model, from this guitar model, only get the data from database and get the id of the data. You can see at the guitar data there are

```

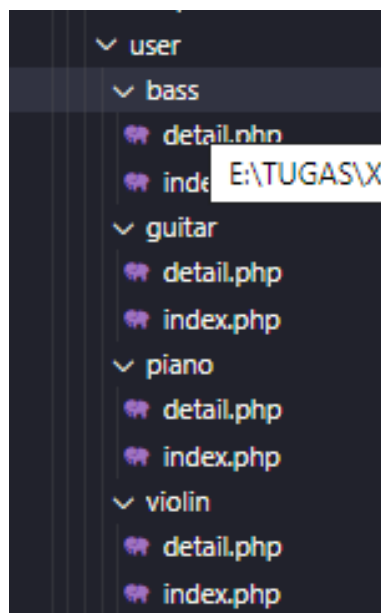
$this->db->like('item_id', 'g');
$this->db->or_like('item_id', 'u');

```

This code is to get only the guitar type to display at the website

All of the models type are same but the different is at the get data, the get data is per type or per name. we use like query to separate the data per type

3.Views



1. views code

User/guitar/index.php

```
<div class="container">
<div class="row mt-3">
  <div class="col-md-20">
    <h2 style="color: white">Guitar List</h2>
    <div class="row mt-2">
      <?php foreach($guitar['data'] as $gtr): ?>
        <?php
          // In this code is used for limit the spec description
to 75 Char only
          if (strlen($gtr['spec']) > 75) {
            $gtr['spec'] = substr($gtr['spec'], 0, 75) . "...";
          }
        ?>
        <!-- Card Content -->
        <div class="col-md-4">
          <div class="card-group">

            <div class="card text-white bg-dark border-
danger mb-3 " style="width: 18rem;">
              <!--
- In image we call the image name from database and then call the imag
e from the folder -->
              

              <div class="card-body">
                <h5 class="card-title text-
white"><?php echo $gtr['item_name']; ?></h5>
                <p class="card-
text"><?php echo $gtr['spec']; ?></p>
                <a href="<?= base_url(); ?>user/guitar/detail/<?=
$gtr['item_id']; ?>" class="btn btn-danger">Detail</a></td>
                <a href="<?= base_url(); ?>user/guitar/buy/<?= $gt
r['item_id']; ?>" class="btn btn-warning">Buy Now !!</a></td>
                <div class="card-footer text-white">
                  <small class="text-muted text-
white">Instrument Type : <?php echo $gtr['inst_type']; ?> <br></small>
```

```

                <small class="text-muted text-
white">Item Id :<?php echo $gtr['item_id'];?> </small>
            </div>
        </div>
    </div>
</div>
<!-- End of card content -->
<?php endforeach; ?>
</div>
</script>

```

This are the user/guitar/index.php views. From this we set the views to card, the card itself will display item image, item name, spec, and the item id, from this card we can see the detailed information of per item. All of the views is same, the different is at the foreach of the data

2.Detail views

```

<div class="container">
    <div class="row mt-3">
        <div class="col-md-6">
            <!-- Card Content -->
            <div class="card">
                <div class="card-header">
                    Guitar Details
                </div>
                <div class="card-body">
                    <h5 class="card-
title"><?= $guitar['item_name']; ?></h5>
                    <p class="card-text">
                        <label for=""><b>Instrument Type</b> : </label>
                    </p>
                    <?= $guitar['inst_type'];?>
                    </p>
                    <p class="card-text">
                        <label for=""><b>Color</b> : </label>
                        <?= $guitar['color'];?>
                    </p>
                    <p class="card-text">

```



```

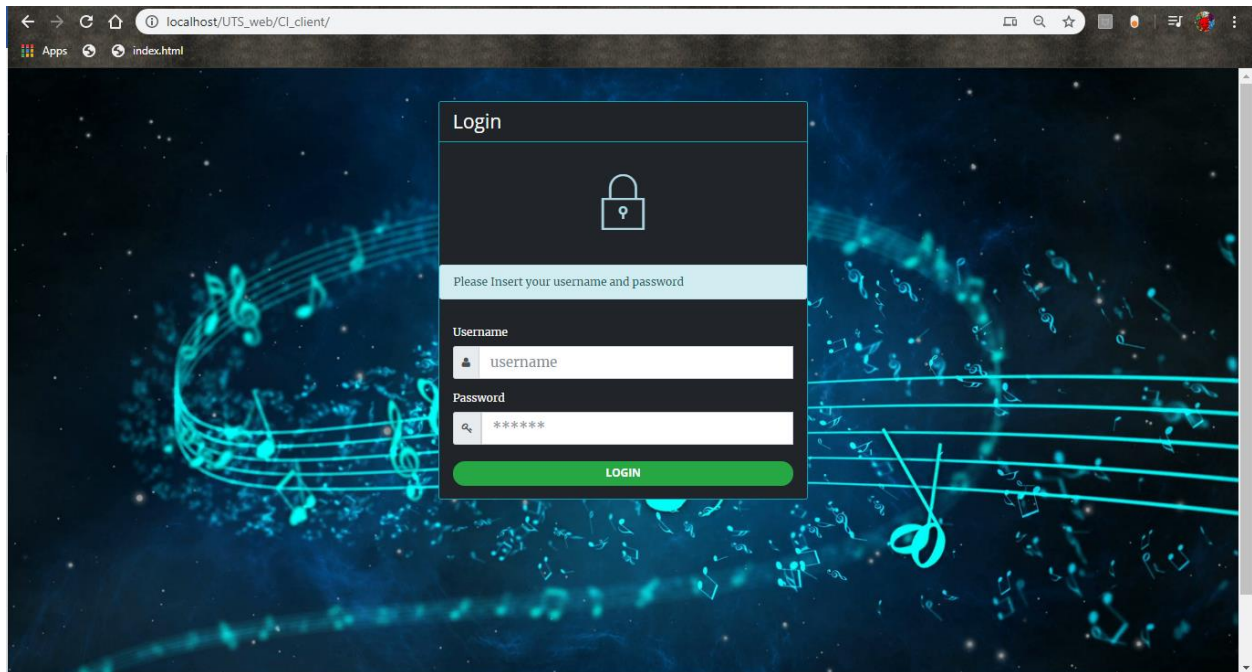
        </p>
        <p class="card-text">
            <label for=""><b>Color</b> : </label>
            <?= $guitar['color'];?>
        </p>
        <p class="card-text">
            <label for=""><b>Spesification</b> : </label>
            <?= $guitar['spec'];?>
        </p>
        <p class="card-text">
            <label for=""><b>Price</b> : Rp.</label>
            <?= $guitar['price'];?>
        </p>
        <a href="<?= base_url(); ?>user/guitar" class="btn
btn-
warning" onClick="return confirm('Thanks For Purchasing')">Buy!!</a>
        <a href="<?= base_url();?>user/guitar" class="btn
btn-info">Go back</a>
    </div>
</div>
<!-- End of card content -->
</div>
</div>
</div>

```

The buy view, is to buy the instrument, its like the detail views, but added with buy button and later the customer will get the order

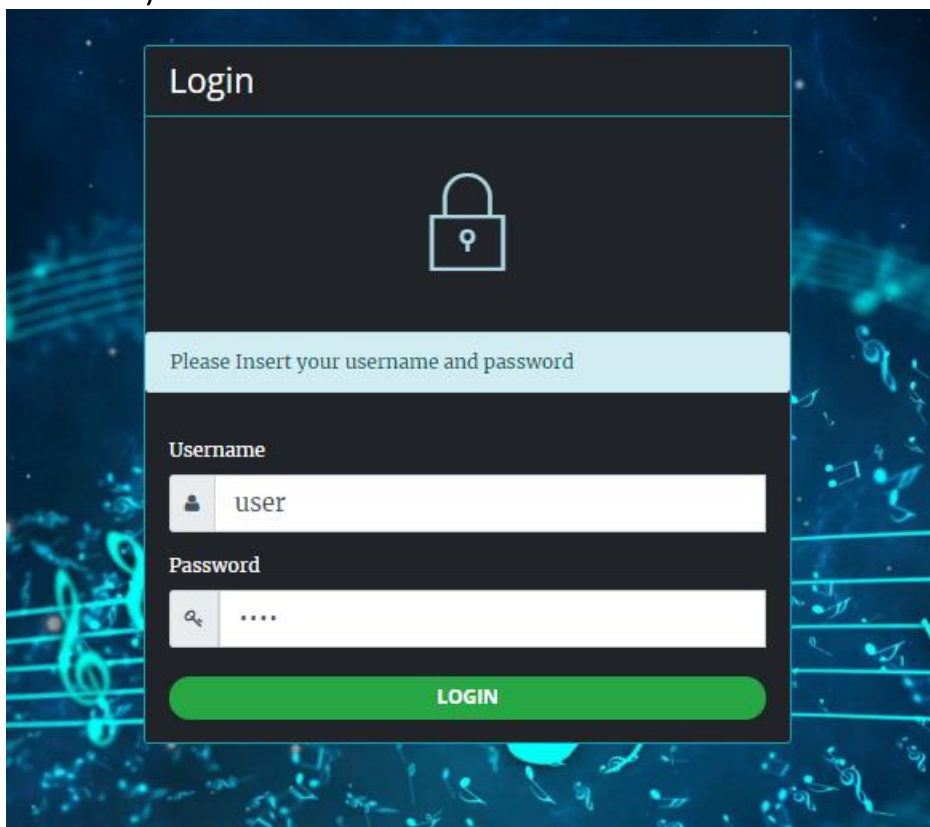
TUTORIAL OF USER

1. First open your web browser and write this url : http://localhost/UTS_web/CI_client/
The url will automatically routes you to the login, because the login is default controller (but first thing dont forget copy the CI folder into htdocs at xampp folder and import the database to the phpmyadmin)

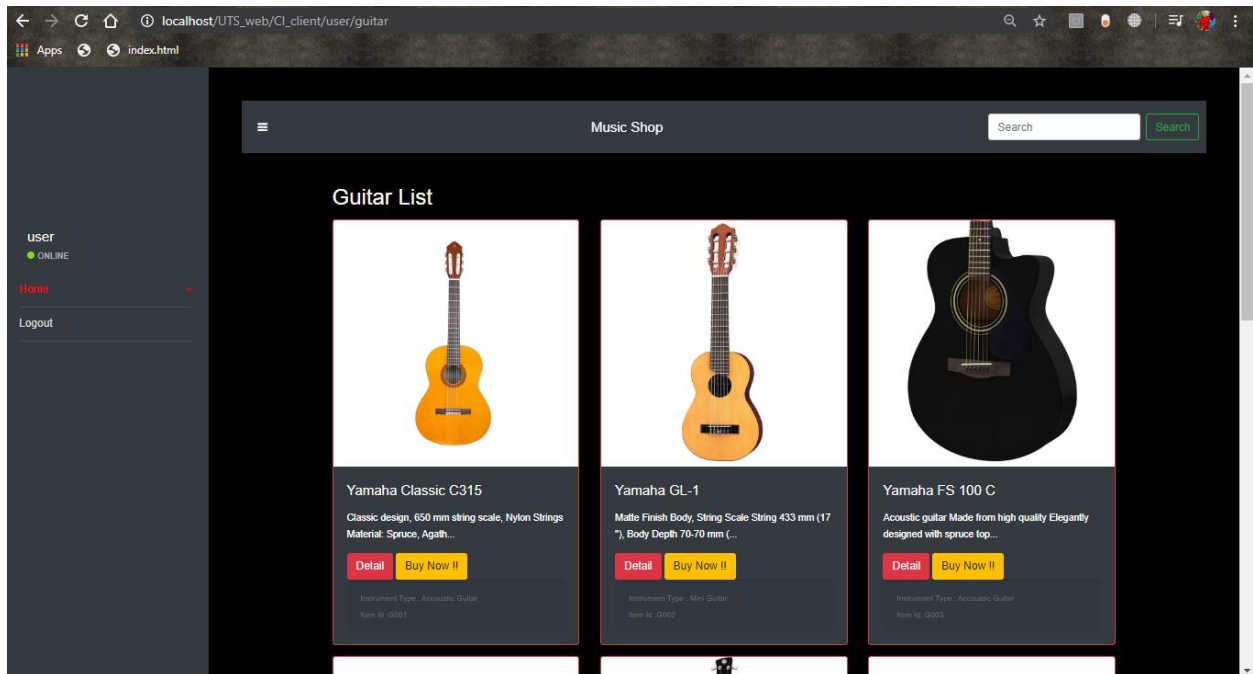


It will popped up interface like this.

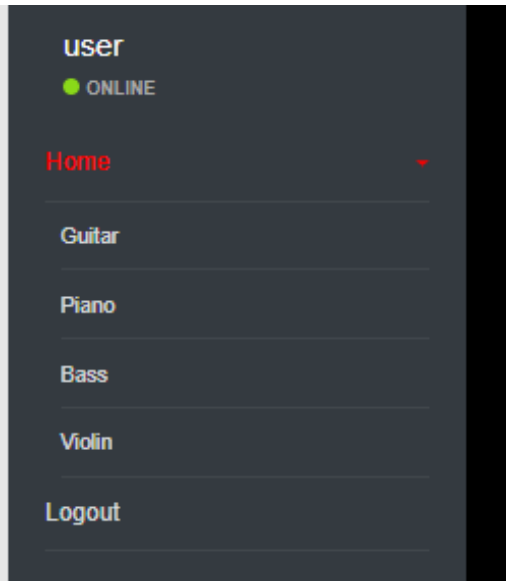
2. Input the user name and password and then click login. For example username is **user** and password is **user** (the username and password is get from the database)



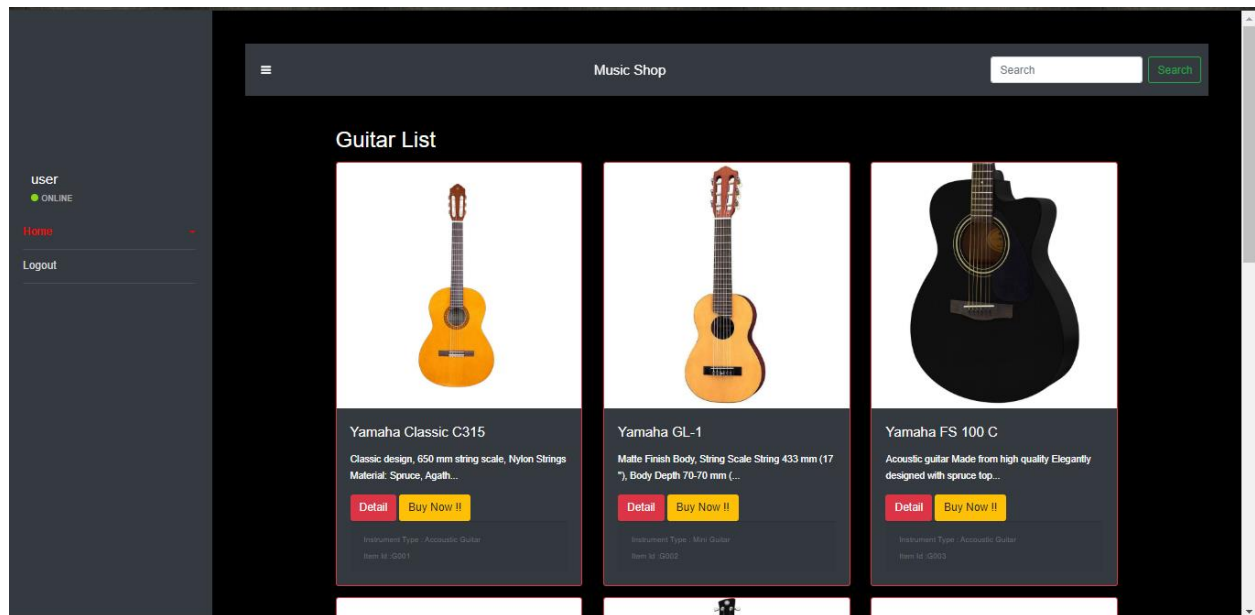
3. After you logged in, it will popped up like this



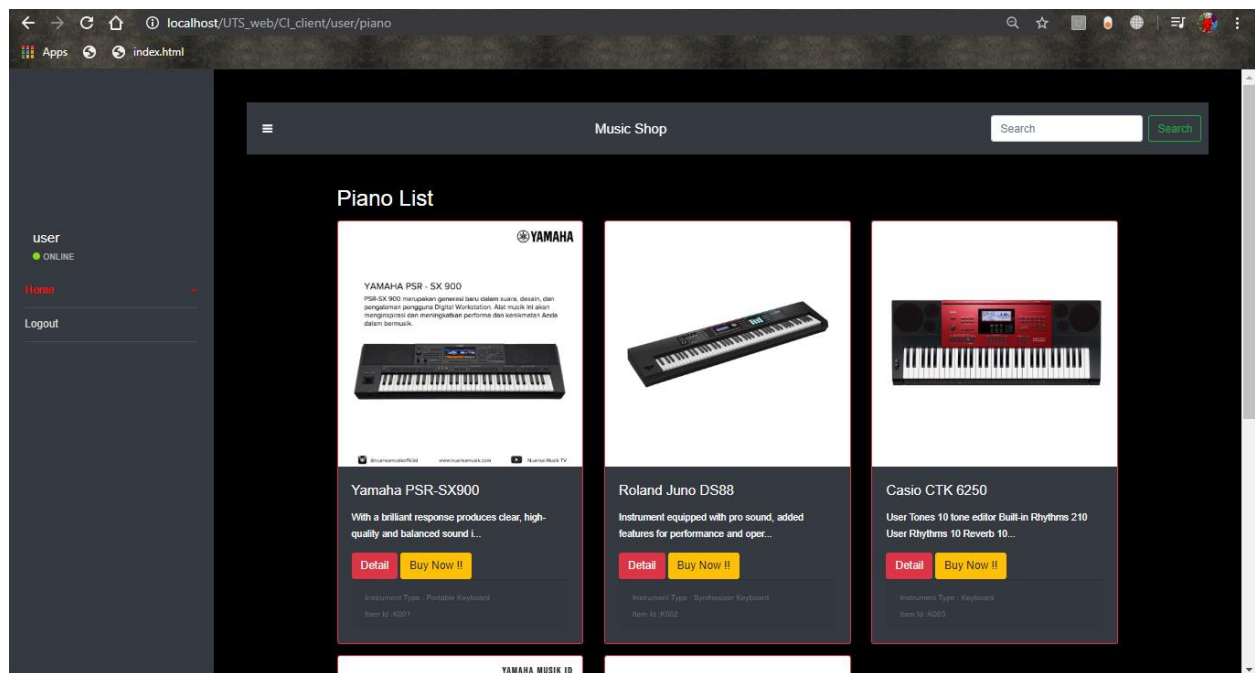
4. You can search the instrument type in this side bar



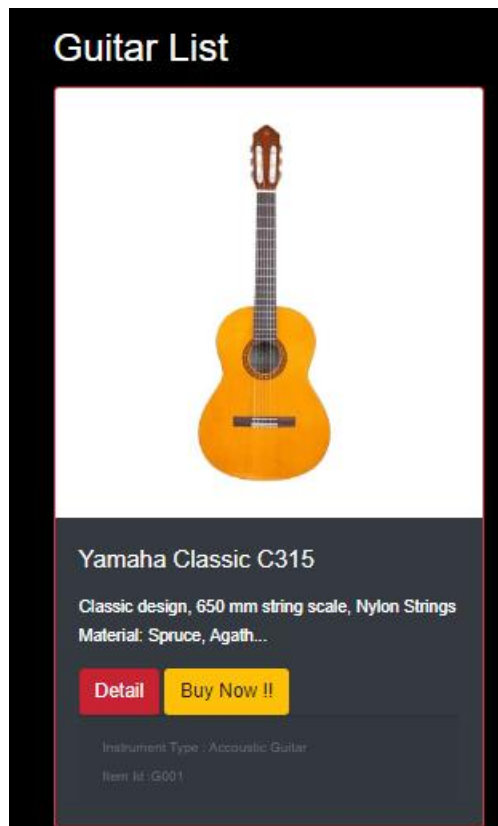
Per side bar will give you some information of instrument per type
For example the guitar, it will guitar list only



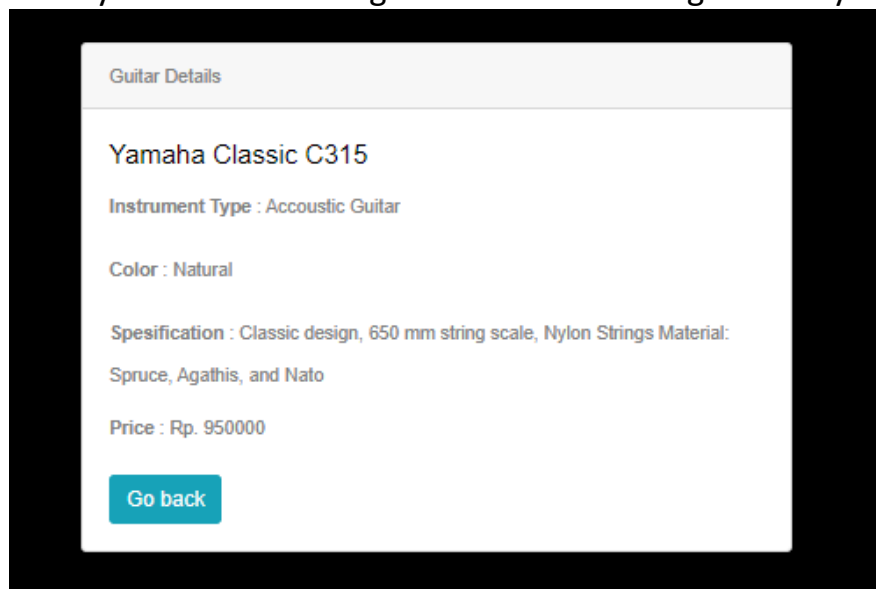
Piano list



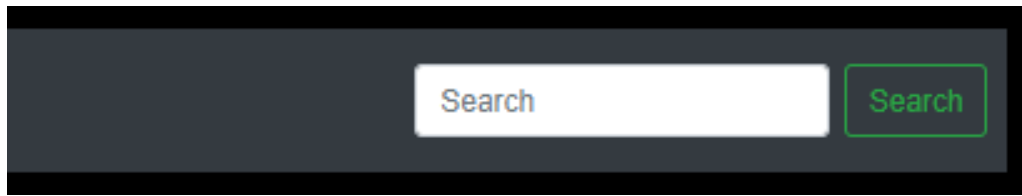
5. From this you can see the detail per item, just by click the detail button



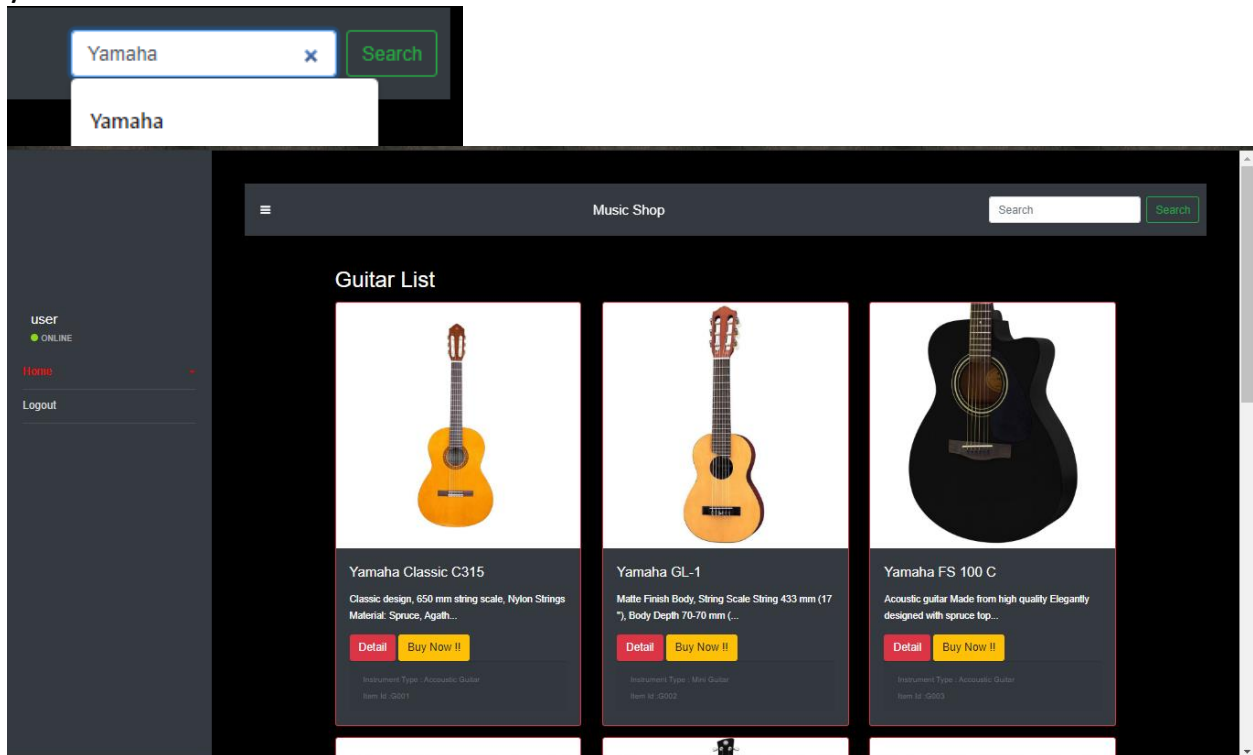
If you click it, it will popped up the detailed version of the instrument, after you done you can click the “go back” button and go to the your last opened list



6. You can search the instrument that you want, by search it the name of the instrument itself at the navbar



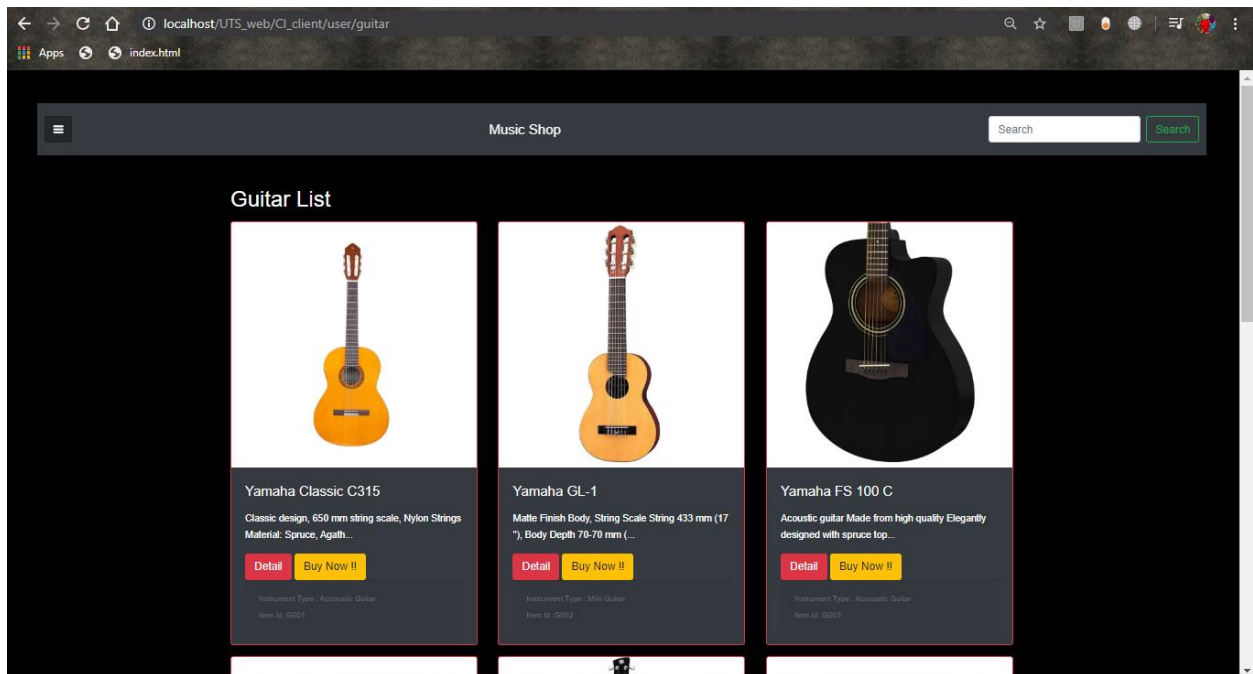
For example i search for yamaha, and it will popped up the instrument with yamaha name



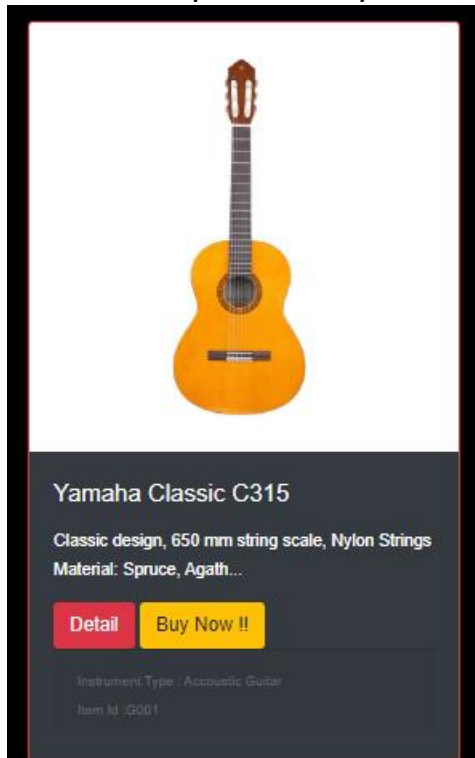
7. if you disturbed by the side bar, you can hide it by click the toogle menu at the side bar



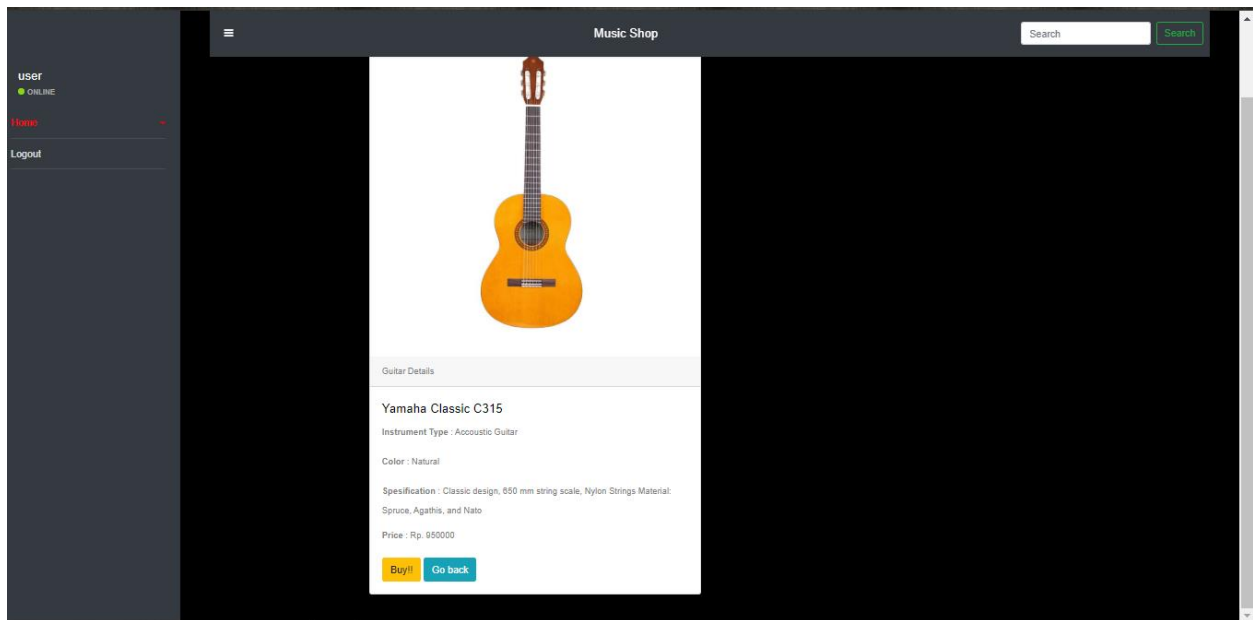
And the sidebar is hidden



8. from this you can buy the item (click buy now)

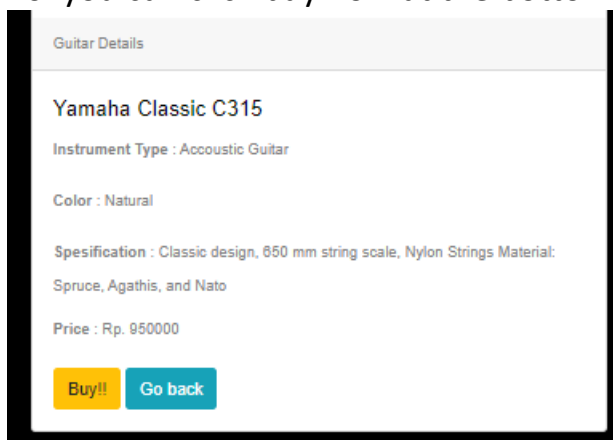


9. it will redirect to this

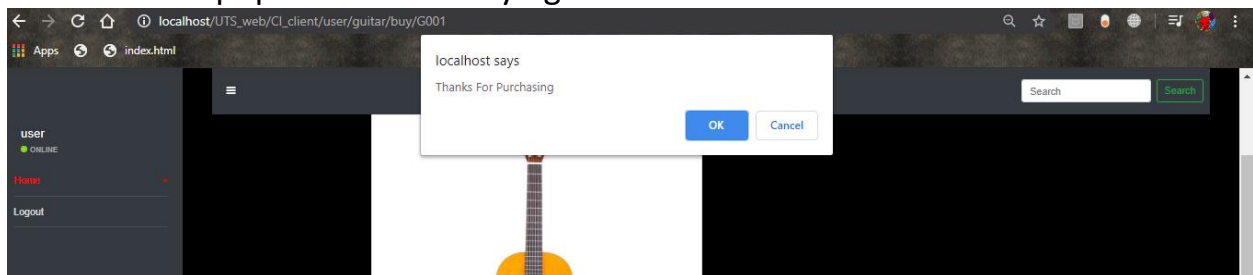


It contain the detailed data plus the image of the instrument,

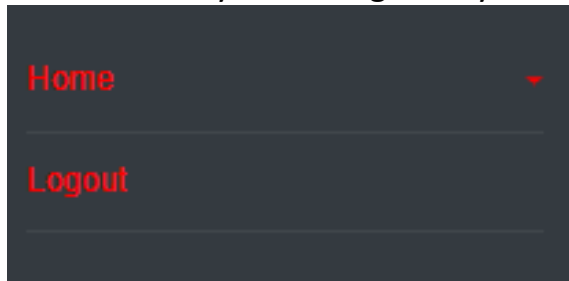
10. you can click buy now at the bottom



And then will pop the alert of buying



11. after that you can logout if you done it (logout is at the side bar)



If you click it you will back at the login page.

LINK OF GITHUB (CODE)

https://github.com/Brian-Kraosio/UAS_Web