REPORT MANUAL OF MIDTERM EXPLANATION AND TUTORIAL (ADVANCED WEB PROGRAMMING)

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

CI Admin (Using Code Igniter Rest Api)

Posting Data to database using Restful:

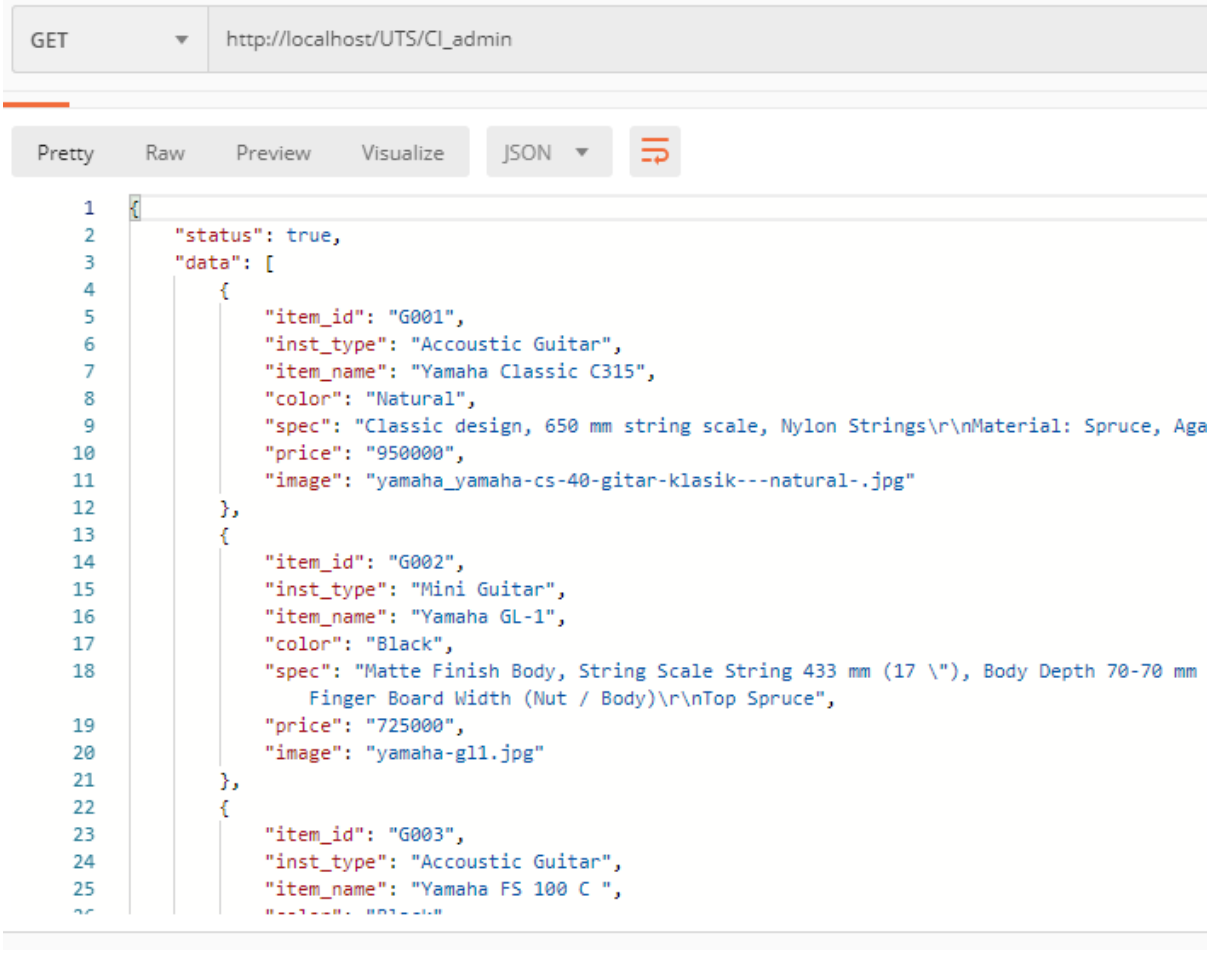
Posting data is described By the function index_get():

```
//Menampilkan data kontak
function index_get() {
    $id = $this->get('item_id');
    $type = $this->get('inst_type');

    if ($id == null and $type == null)  {
        $item = $this->item->getItem();
    } else if($id != null and $type == null){
        $item = $this->item->getItem($id);
    } else if ($id == null and $type != null){
        $item = $this->item->getItemByType($type);
    }
    if($item){
        $this->response([
            'status' => true,
            'data'=> $item
        ], REST_Controller::HTTP_OK);

    }
    else{
        $this->response([
            'status' =>false,
            'message'=>'id not found'
        ], REST_Controller::HTTP_NOT_FOUND);
    }
}
```

We can Get Items by item_id or by instrument_type:
The model is:

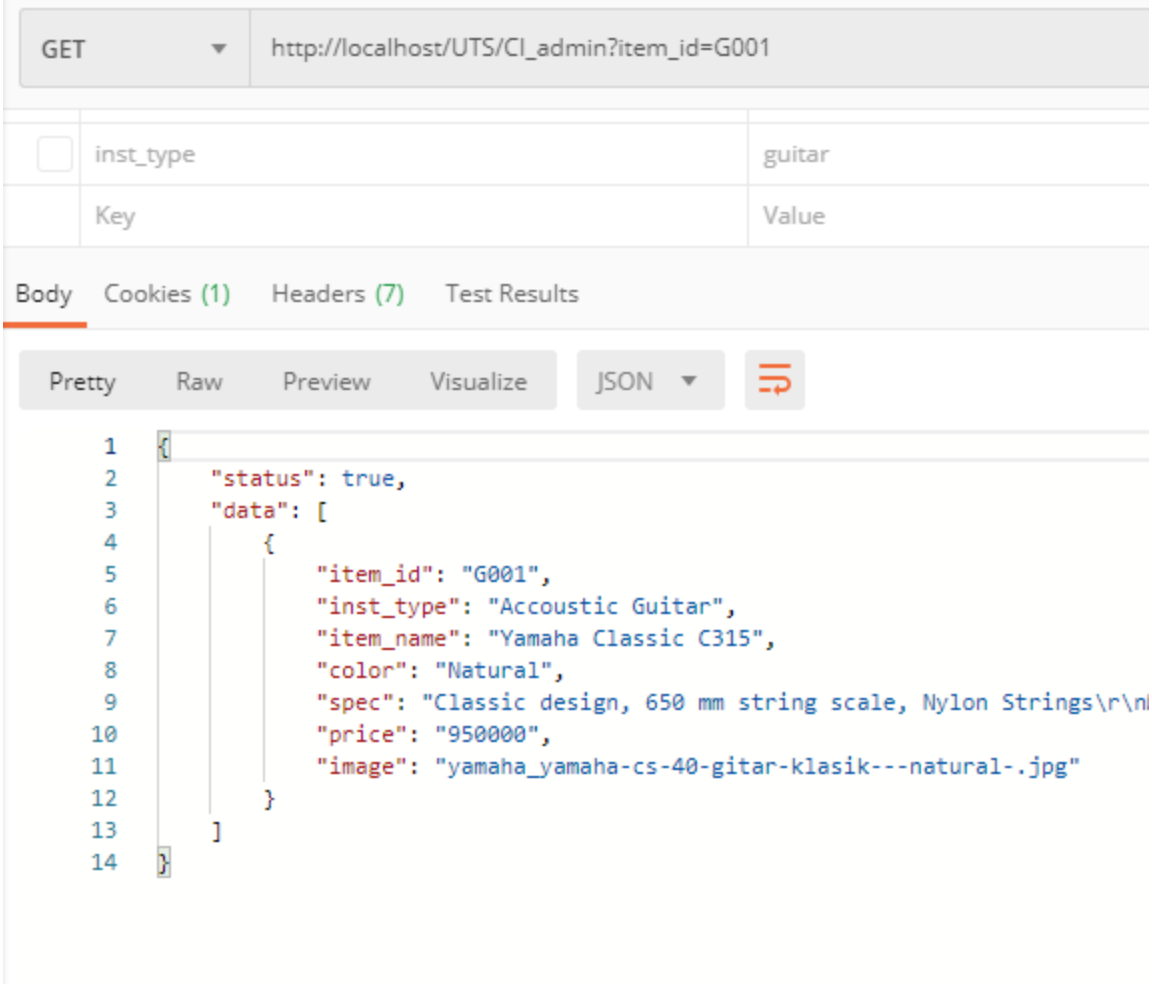
```php
public function getItem($id = null)
{
    if($id === null){
        return $this->db->get('item')->result_array();
    } else{
        return $this->db->get_where('item', ['item_id'=>$id])->result_array();
    }
}
public function getItemByType($id = null)
{
    if($id === null){
        return $this->db->get('item')->result_array();
    } else{
        // return $this->db->get_where('item', ['inst_type'=>$id])->result_array();
        $this->db->like('inst_type',$id);
        return $this->db->get('item')->result_array();
    }
}
```

If the key is left Blank, then we will get all the items:

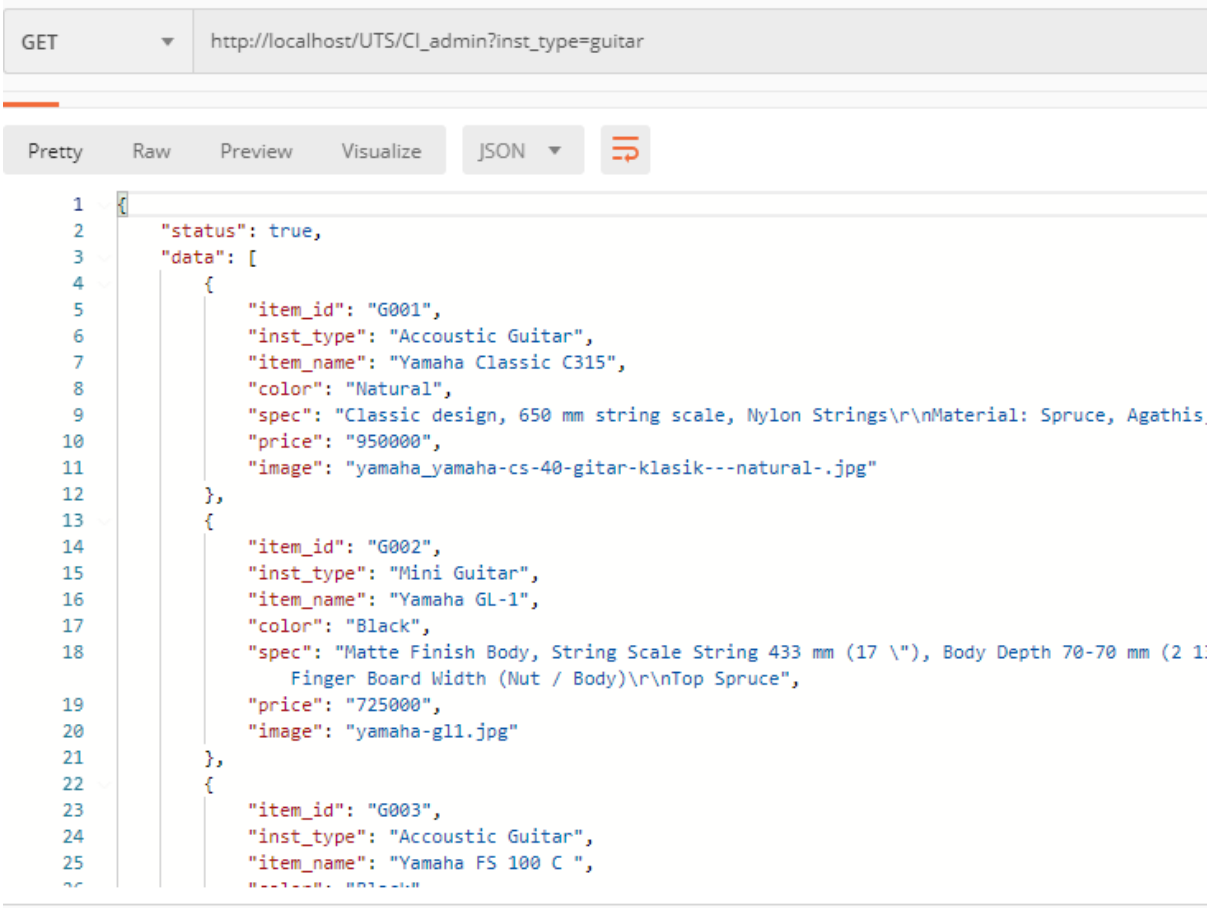



```
{
    "status": true,
    "data": [
        {
            "item_id": "G001",
            "inst_type": "Accoustic Guitar",
            "item_name": "Yamaha Classic C315",
            "color": "Natural",
            "spec": "Classic design, 650 mm string scale, Nylon Strings\r\nMaterial: Spruce, Aga
            "price": "950000",
            "image": "yamaha_yamaha-cs-40-gitar-klasik---natural-.jpg"
        },
        {
            "item_id": "G002",
            "inst_type": "Mini Guitar",
            "item_name": "Yamaha GL-1",
            "color": "Black",
            "spec": "Matte Finish Body, String Scale String 433 mm (17 \"), Body Depth 70-70 mm
                Finger Board Width (Nut / Body)\r\nTop Spruce",
            "price": "725000",
            "image": "yamaha-gl1.jpg"
        },
        {
            "item_id": "G003",
            "inst_type": "Accoustic Guitar",
            "item_name": "Yamaha FS 100 C ",
```


If the Key is item_id, then we will get the id entered like in this exmple:



If the inst_type is filled, then we will get all the instruments that match that cinstrument type like in this example:




```
{
    "status": true,
    "data": [
        {
            "item_id": "G001",
            "inst_type": "Accoustic Guitar",
            "item_name": "Yamaha Classic C315",
            "color": "Natural",
            "spec": "Classic design, 650 mm string scale, Nylon Strings\r\nMaterial: Spruce, Agathis,
            "price": "950000",
            "image": "yamaha_yamaha-cs-40-gitar-klasik---natural-.jpg"
        },
        {
            "item_id": "G002",
            "inst_type": "Mini Guitar",
            "item_name": "Yamaha GL-1",
            "color": "Black",
            "spec": "Matte Finish Body, String Scale String 433 mm (17 \"), Body Depth 70-70 mm (2 1
                Finger Board Width (Nut / Body)\r\nTop Spruce",
            "price": "725000",
            "image": "yamaha-gl1.jpg"
        },
        {
            "item_id": "G003",
            "inst_type": "Accoustic Guitar",
            "item_name": "Yamaha FS 100 C ",
```

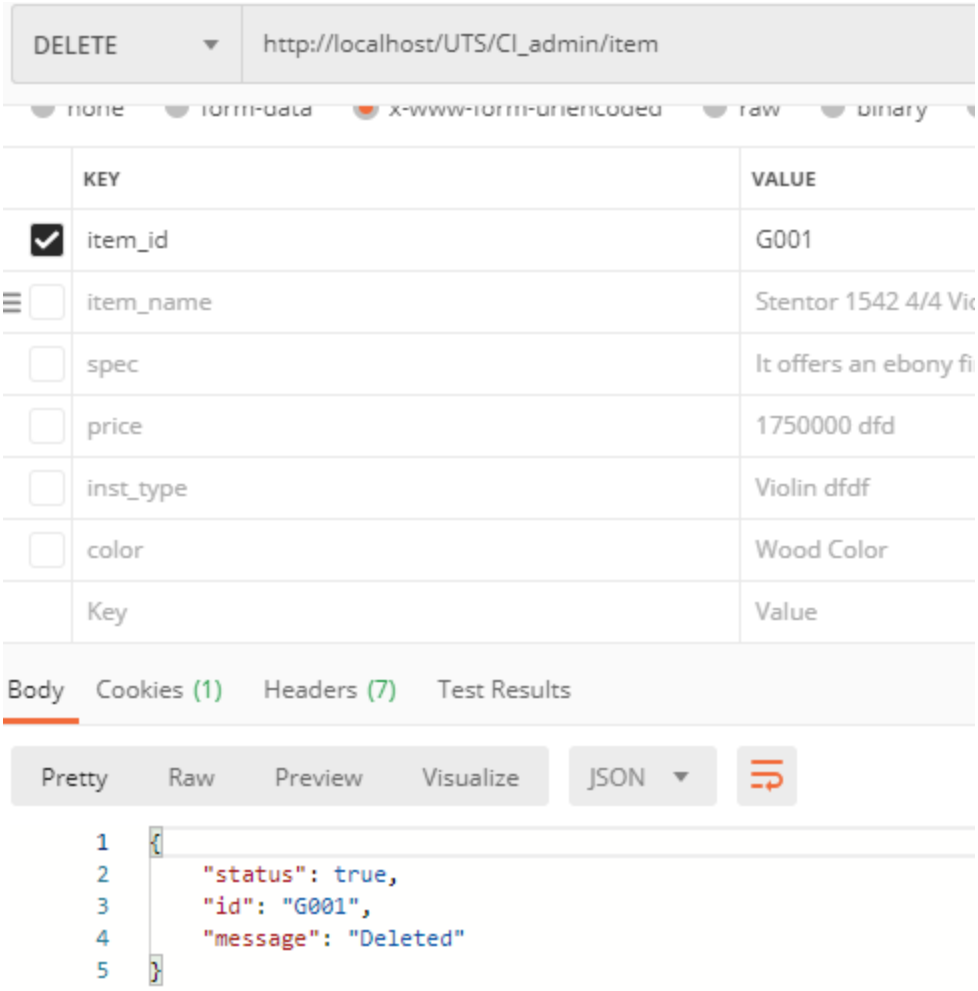

Deleting records are done by this code:

```php
public function index_delete(){
    $id=$this->delete('item_id');

    if($id === null){
        $this->response([
            'status' =>false,
            'message' =>'Provide an Id!'
        ], REST_Controller::HTTP_BAD_REQUEST);
    } else{
        if($this->item->deleteItem($id) > 0){
            $this->response([
                'status' => true,
                'id' => $id,
                'message'=> 'Deleted'
            ], REST_Controller::HTTP_OK);
        } else{
            $this->response([
                'status' =>false,
                'message'=>'id not found'
            ], REST_Controller::HTTP_BAD_REQUEST);
        }
    }
}
```

The model for delete is:

```php
public function deleteItem($id){
    $this->db->delete('item',['item_id' => $id]);
    return $this->db->affected_rows();
}
```

The deletion must be done by item_id like in this example:



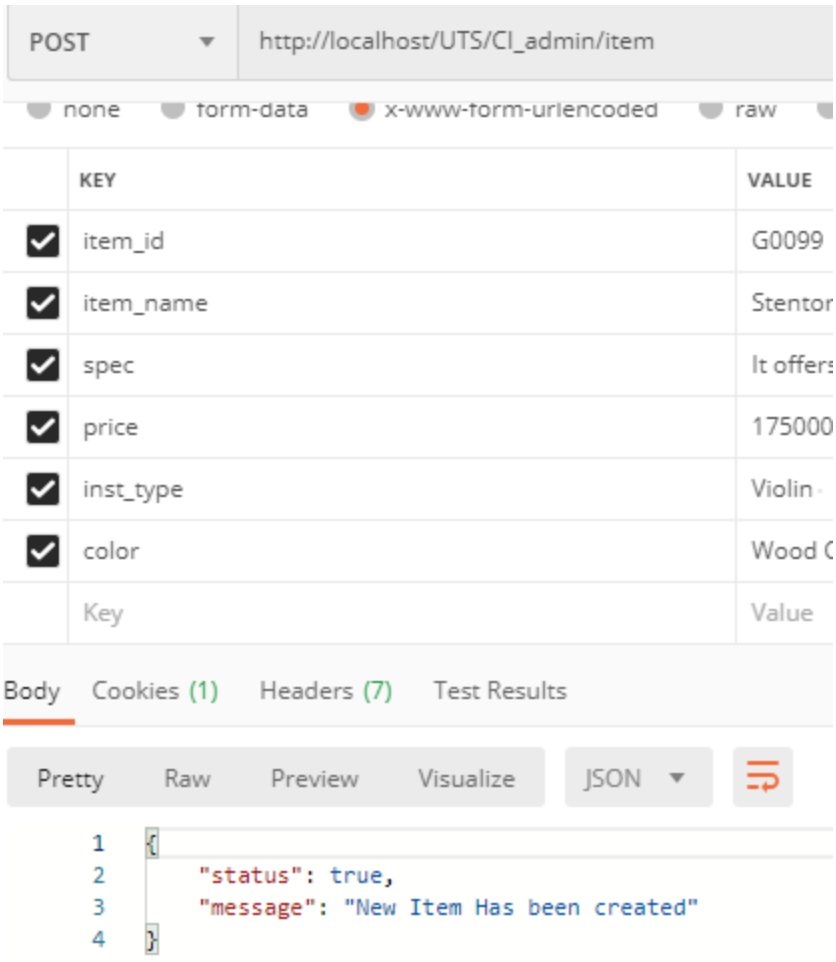
Creating a new record in the database is as follows:

```php
public function index_post(){
    $data =[
        'item_id' => $this->post('item_id'),
        'inst_type' => $this->post('inst_type'),
        'item_name' => $this->post('item_name'),
        'color' => $this->post('color'),
        'spec' => $this->post('spec'),
        'price' => $this->post('price'),
        'image' => $this->post('image'),


    ];

    if($this->item->createItem($data) > 0){
        $this->response([
            'status' => true,
            'message'=> 'New Item Has been created'
        ], REST_Controller::HTTP_CREATED);
    } else{
        $this->response([
            'status' =>false,
            'message'=>'Failed To create a New Data'
        ], REST_Controller::HTTP_BAD_REQUEST);
    }
}
```

The Model for creating is:

```php
public function createItem($data){
    $this->db->insert('item',$data);
    return $this->db->affected_rows();
}
```

And the results are:



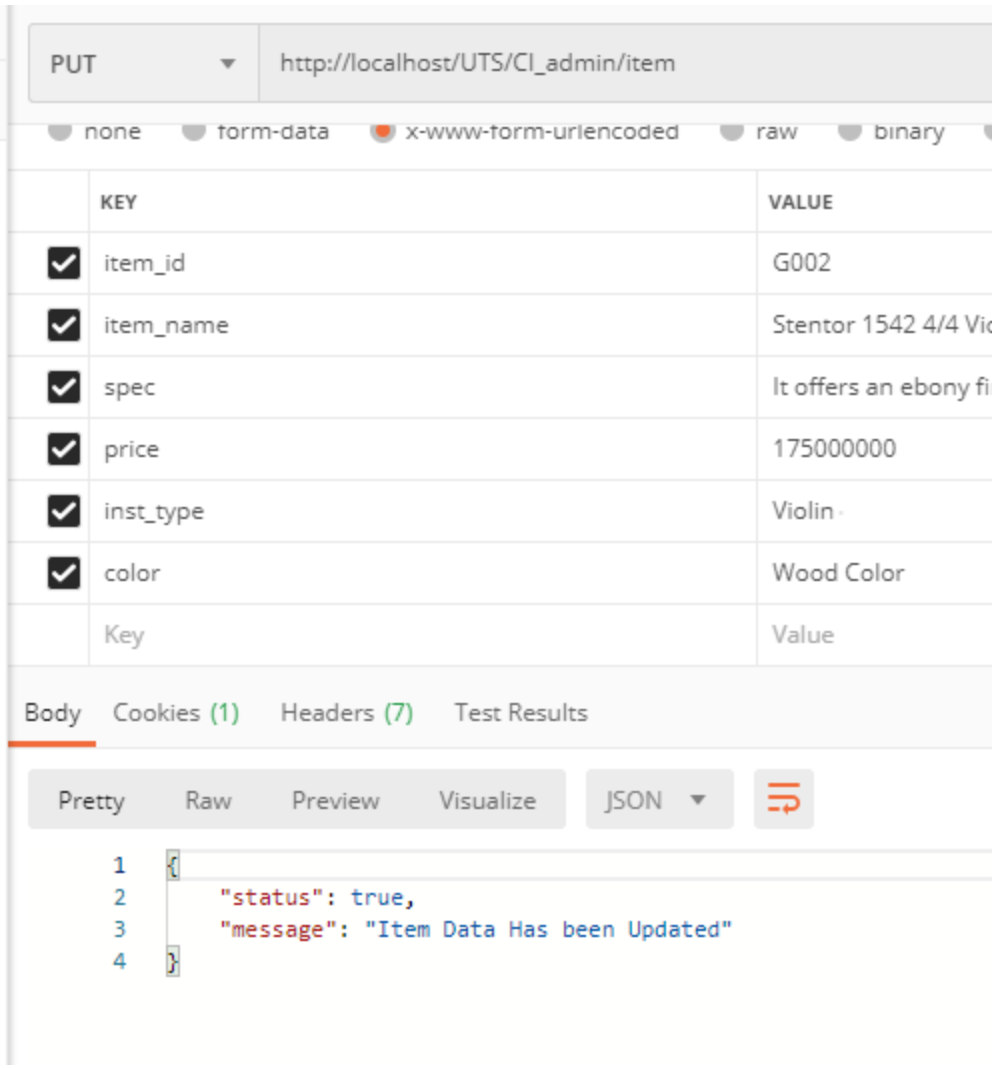
Updating a record inside the database is as follows:

```php
    public function index_put(){
        $id= $this->put('item_id');
        $data =[
            'item_id' => $this->put('item_id'),
            'inst_type' => $this->put('inst_type'),
            'item_name' => $this->put('item_name'),
            'color' => $this->put('color'),
            'spec' => $this->put('spec'),
            'price' => $this->put('price'),
            'image' => $this->post('image'),
        ];

        if($this->item->updateItem($data, $id) > 0){
            $this->response([
                'status' => true,
                'message'=> 'Item Data Has been Updated'
            ], REST_Controller::HTTP_OK);
        } else{
            $this->response([
                'status' =>false,
                'message'=>'Failed To Update Data'
            ], REST_Controller::HTTP_BAD_REQUEST);
        }
    }
    //Masukan function selanjutnya disini
}
?>
```

The Model for Update is:

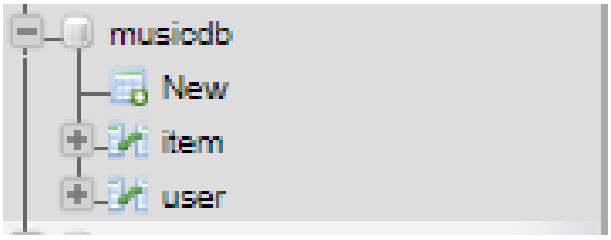
```php
public function updateItem( $data, $id){
    $this->db->update('item',$data ,['item_id' => $id]);
    return $this->db->affected_rows();
}
```

The result in Postman is:



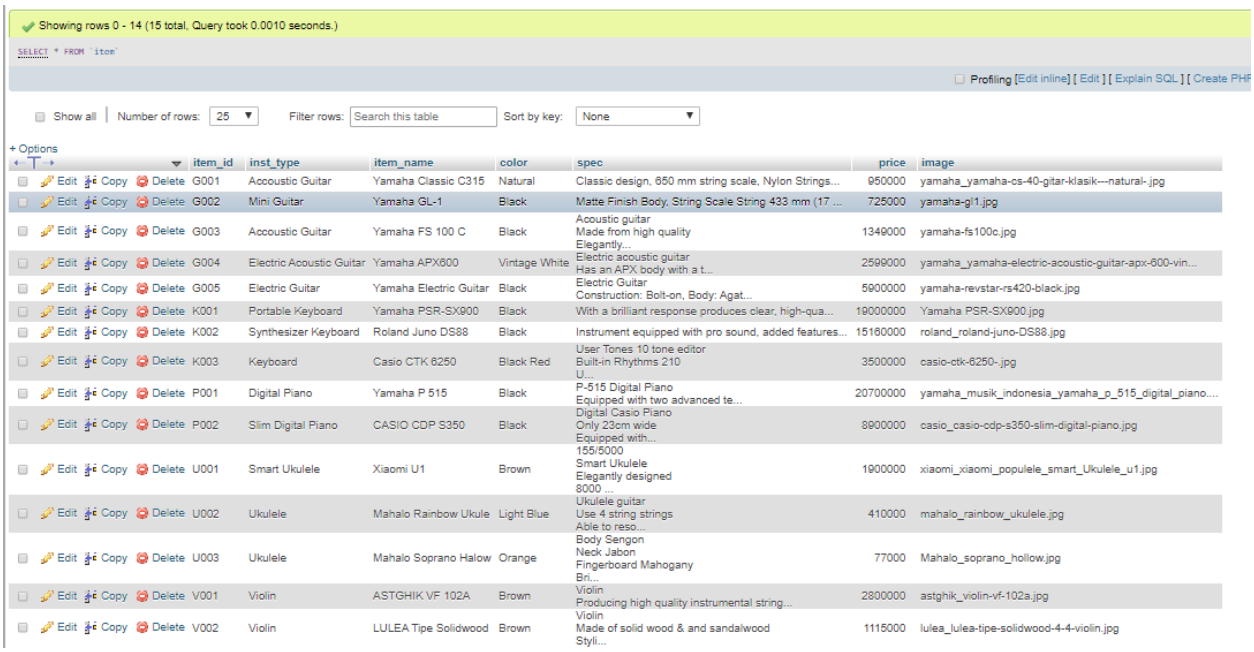
CI Client (Using code igniter php)

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

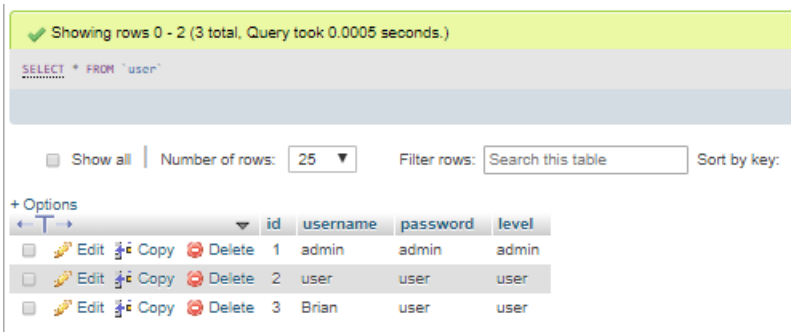
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	item_id	inst_type	item_name	color	spec	price	image
Edit Copy Delete	G001	Accoustic Guitar	Yamaha Classic C315	Natural	Classic design, 650 mm string scale, Nylon Strings...	950000	yamaha_yamaha-cs-40-gitar-klasik---natural-.jpg
Edit Copy Delete	G002	Mini Guitar	Yamaha GL-1	Black	Matte Finish Body, String Scale String 433 mm (17 ...	725000	yamaha-gl1.jpg
Edit Copy Delete	G003	Accoustic Guitar	Yamaha FS 100 C	Black	Acoustic guitar Made from high quality Elegantly...	1349000	yamaha-fs100c.jpg
Edit Copy Delete	G004	Electric Acoustic Guitar	Yamaha APX600	Vintage White	Electric acoustic guitar Has an APX body with a t...	2599000	yamaha_yamaha-electric-acoustic-guitar-apx-600-vin...
Edit Copy Delete	G005	Electric Guitar	Yamaha Electric Guitar	Black	Electric Guitar Construction: Bolt-on, Body: Agat...	5900000	yamaha-revstar-rs420-black.jpg
Edit Copy Delete	K001	Portable Keyboard	Yamaha PSR-SX900	Black	With a brilliant response produces clear, high-qua...	19000000	Yamaha PSR-SX900.jpg
Edit Copy Delete	K002	Synthesizer Keyboard	Roland Juno DS88	Black	Instrument equipped with pro sound, added features...	15160000	roland_roland-juno-DS88.jpg
Edit Copy Delete	K003	Keyboard	Casio CTK 6250	Black Red	User Tones 10 tone editor Built-in Rhythms 210 U...	3500000	casio-ctk-6250-.jpg
Edit Copy 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....
Edit Copy 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
Edit Copy Delete	U001	Smart Ukulele	Xiaomi U1	Brown	155/5000 Smart Ukulele Elegantly designed 8000 ...	1900000	xiaomi_xiaomi_populele_smart_Ukulele_u1.jpg
Edit Copy Delete	U002	Ukulele	Mahalo Rainbow Ukule	Light Blue	Ukulele guitar Use 4 string strings Able to reso...	410000	mahalo_rainbow_ukulele.jpg
Edit Copy Delete	U003	Ukulele	Mahalo Soprano Halow	Orange	Body Sengon Neck Jabon Fingerboard Mahogany Bri...	77000	Mahalo_soprano_hollow.jpg
Edit Copy Delete	V001	Violin	ASTGHIK VF 102A	Brown	Violin Producing high quality instrumental string...	2800000	astghik_violin-vf-102a.jpg
Edit Copy Delete	V002	Violin	LULEA Tipe Solidwood	Brown	Violin Made of solid wood & and sandalwood Styli...	1115000	lulea_lulea-tipe-solidwood-4-4-violin.jpg

Table item containt 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:

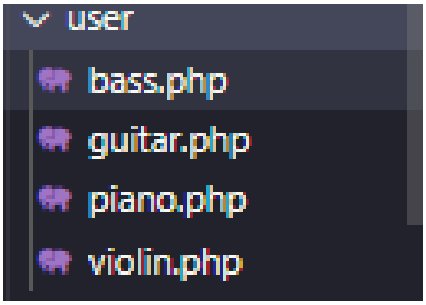
+ Options

	id	username	password	level
Edit Copy Delete	1	admin	admin	admin
Edit Copy Delete	2	user	user	user
Edit Copy Delete	3	Brian	user	user

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

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

2. Controller Code

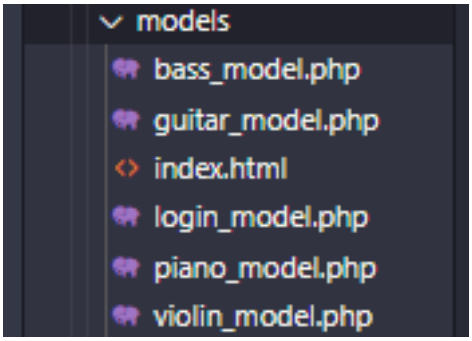
```php
<?php

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

    class guitar extends CI_Controller {

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

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

        }
        // To get guitar data

        public function index()
        {
            $data['title']='MUSIC SHOP';
            $data['name']='$name';
            $data['guitar'] = $this->guitar_model->getGuitarData();
```

```
            if($this->input->post('keyword')){
                $data['guitar']=$this->guitar_model->searchData();
            }
            $this->load->view('template/header_user', $data);
            $this->load->view('user/guitar/index', $data);
            $this->load->view('template/footer_user');
        }

// 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');
        }
    }

    /* End of file guitar.php */
```

From this we can see this is the guitar controller, its contain to control the gitar views only. Because this is only user website, the controller only containt get data and detail per item only.

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

2. Models




```
<?php
```

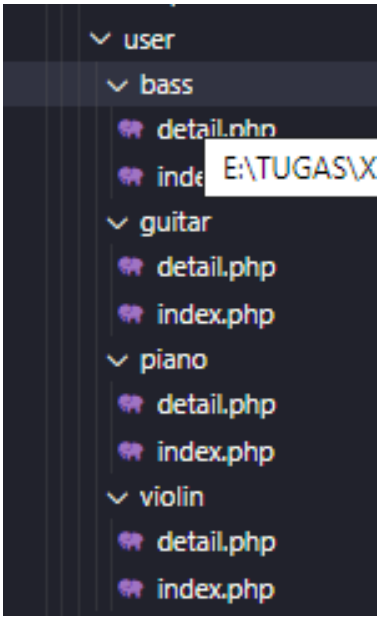
```php
    defined('BASEPATH') OR exit('No direct script access allowed');

    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

```php
$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

3.Views



1. views code

```
<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 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 -->
```

```
                        <img class="card-img-
top" src="<?php echo base_url().'assets/img/'.$gtr['image'] ?>"  alt="
Card image cap">
                    <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>
                    <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>
    </div>
    <!-- End of card content -->
    <?php endforeach; ?>
    </div>
```

This are the guitar/index.php views. From this we set the views to card, the card itslef will displat 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

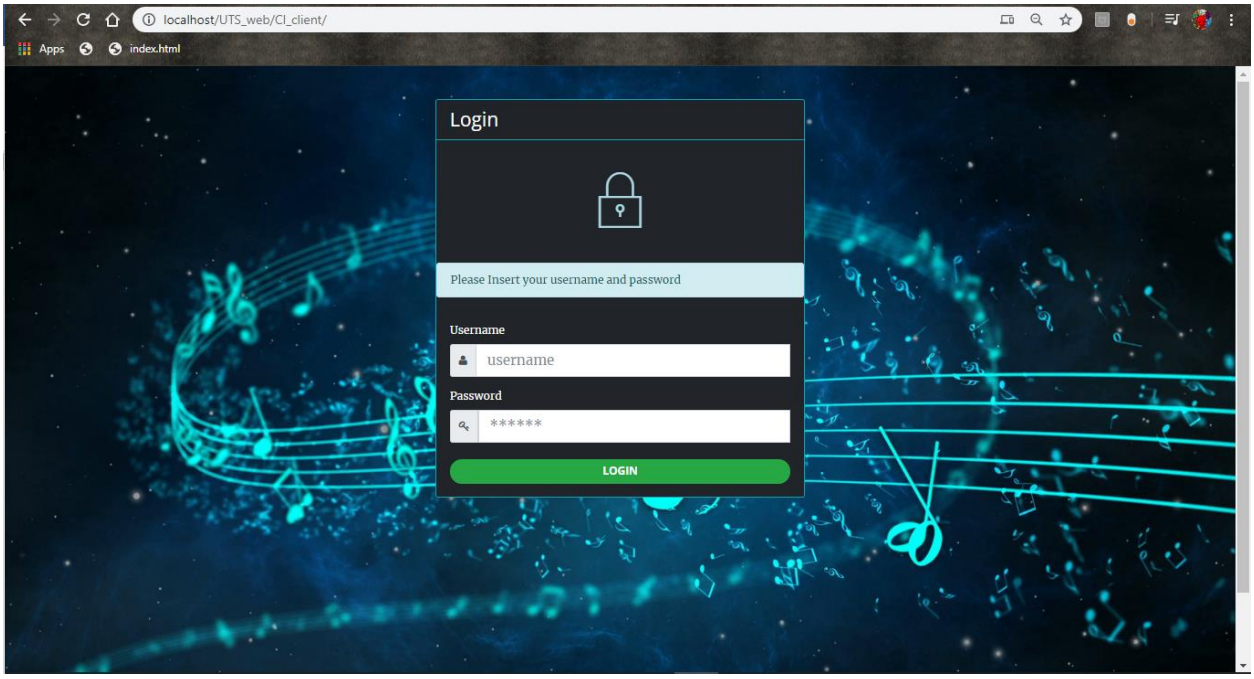
```
<div class="container">
    <div class="row mt-3">
        <div class="col-md-6">
            <!-- Card Content -->
            <div class="card">
                <div class="card-header">
                    Guitar Details
                </div>
                <div class="card-body">
```

```
                    <h5 class="card-
title"><?= $guitar['item_name']; ?></h5>
                    <p class="card-text">
                        <label for=""><b>Instrument Type</b> : </label
>
                        <?= $guitar['inst_type'];?>
                    </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-info">Go back</a>
                </div>
            </div>
            <!-- End of card content -->
        </div>
    </div>
</div>
```

The detail view, contain the data of the item itself, the data will describe all of the information that user needs

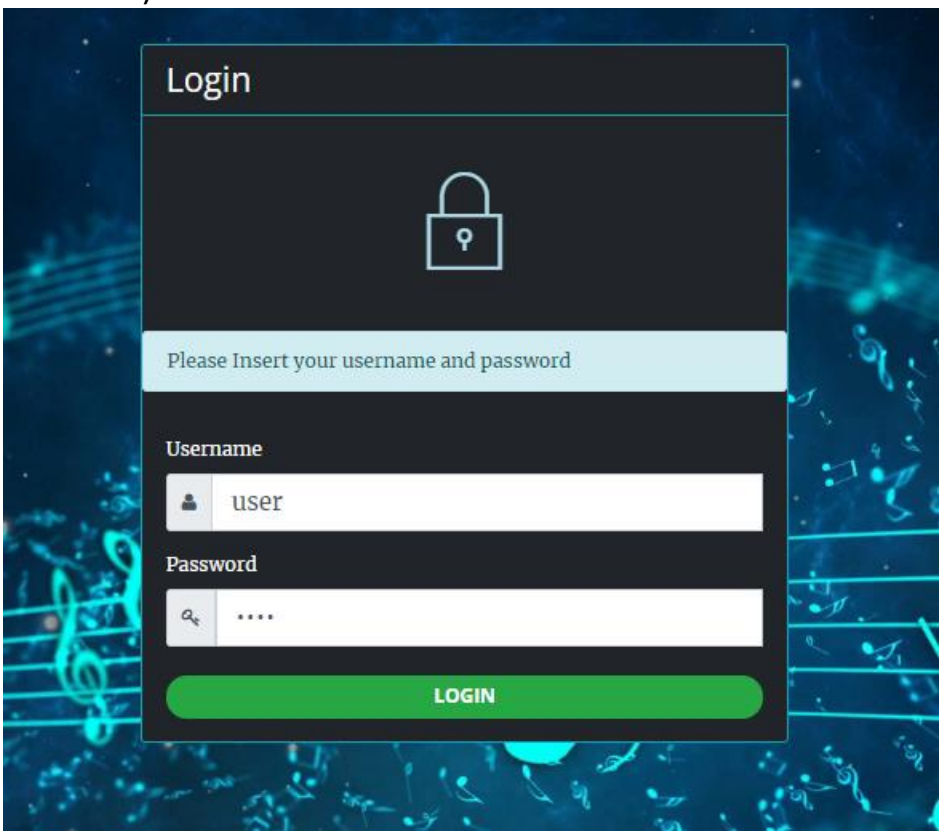
TUTORIAL OF USER

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)

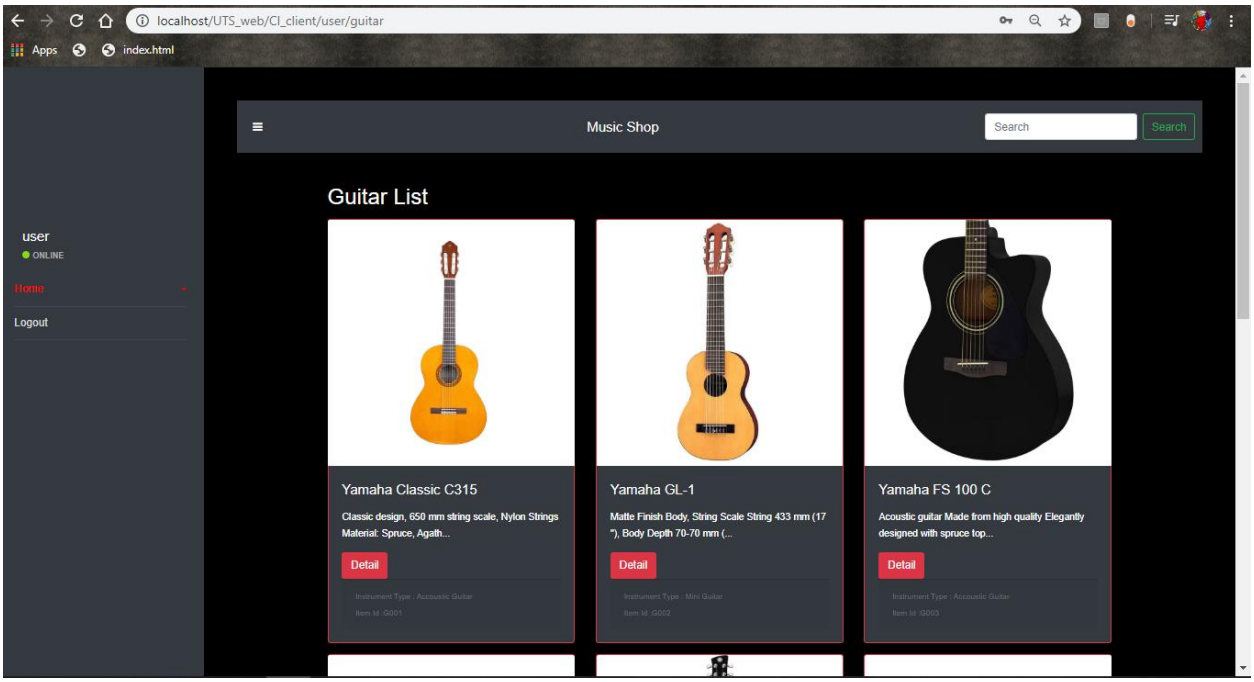


It will popped up interface like this.

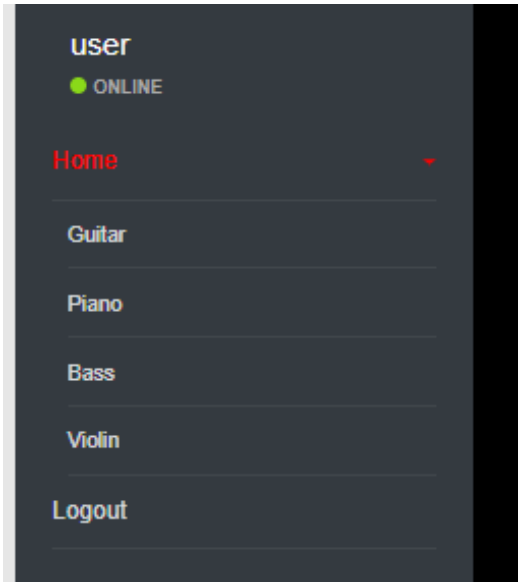
2. Input the user name and password and then click login. For example username is **user** and password is **user** (the username and password is get from the database)



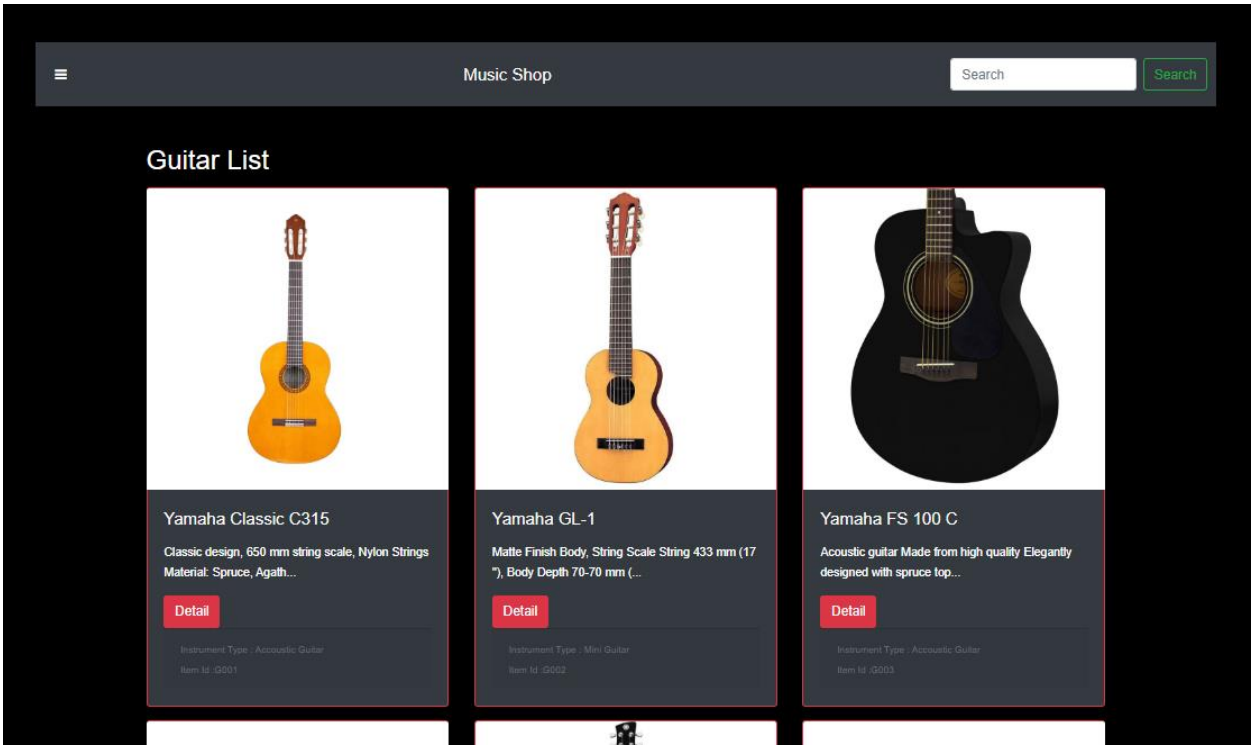
3. After you logged in, it will popped up like this



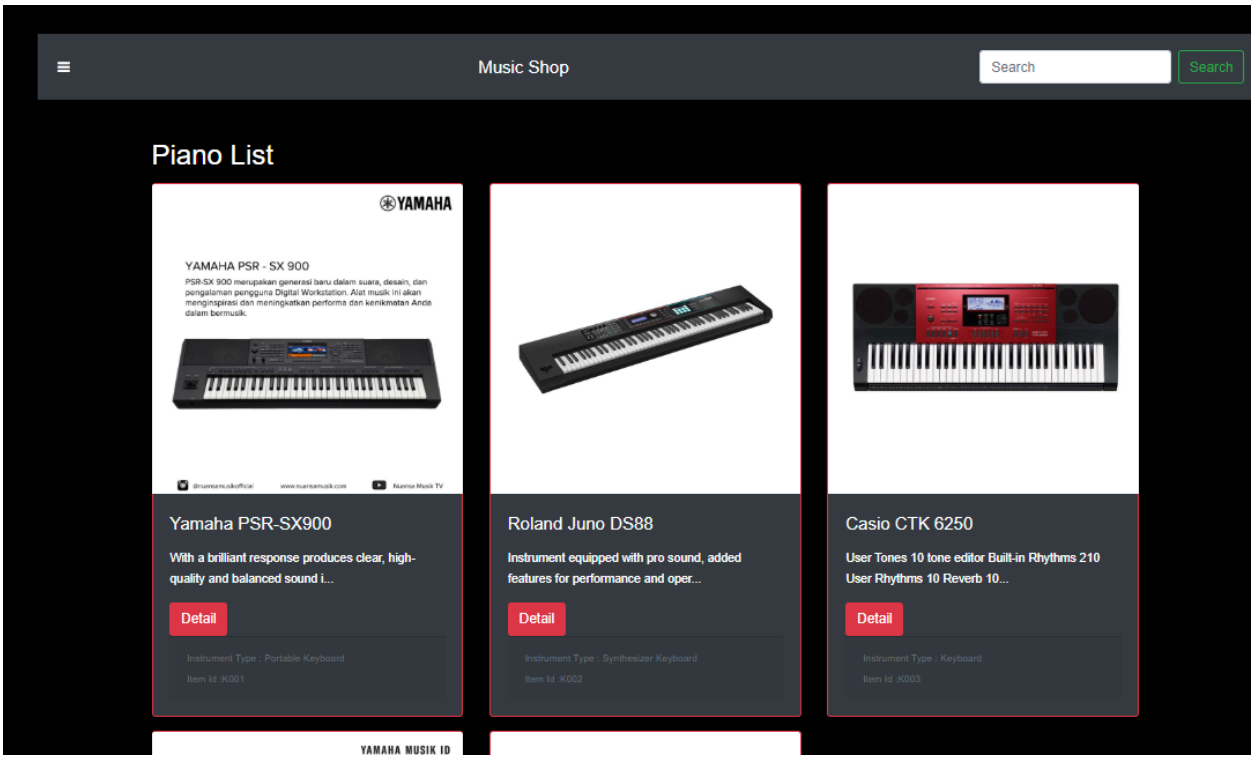
4. You can search the instrument type in this side bar



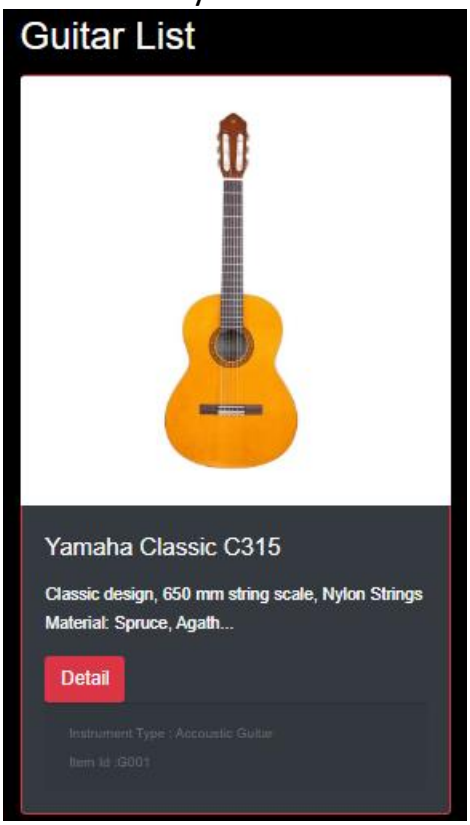
Per side bar will give you some information of instrument per type
For example the guitar, it will guitar list only


Music Shop
Search
Search
Guitar List
Yamaha Classic C315
Classic design, 650 mm string scale, Nylon Strings Material: Spruce, Agath...
Detail
Yamaha GL-1
Matte Finish Body, String Scale String 433 mm (17 "), Body Depth 70-70 mm (...
Detail
Yamaha FS 100 C
Acoustic guitar Made from high quality Elegantly designed with spruce top...
Detail

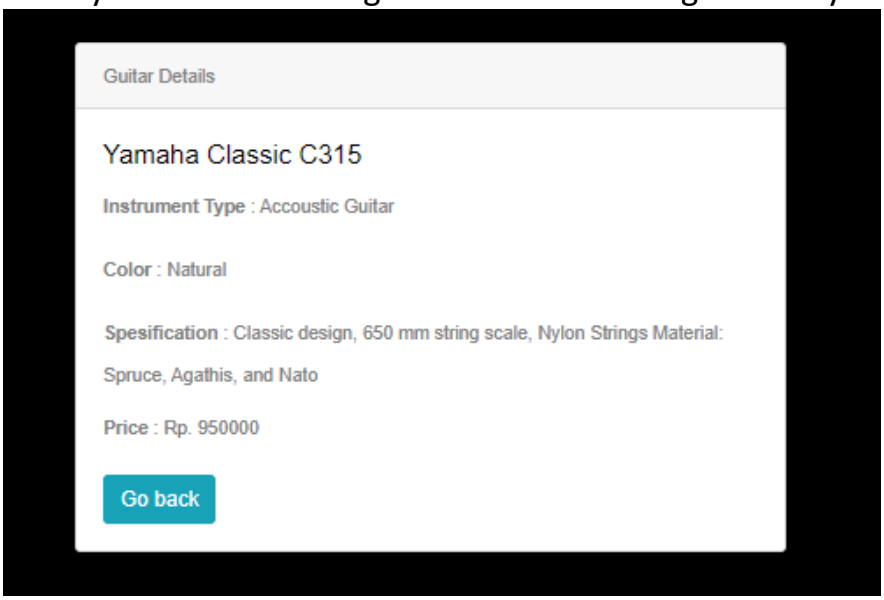
Piano list



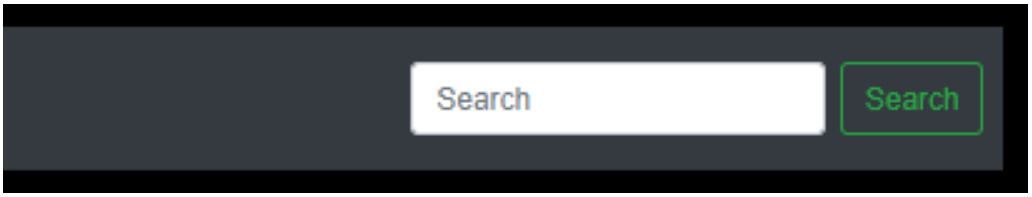
5. From this you can see the detail per item, just by click the detail button



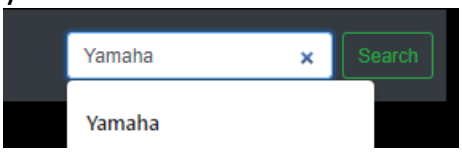
If you click it, it will popped up the detailed version of the instrument, after you done you can click the “go back” button and go to the your last opened list

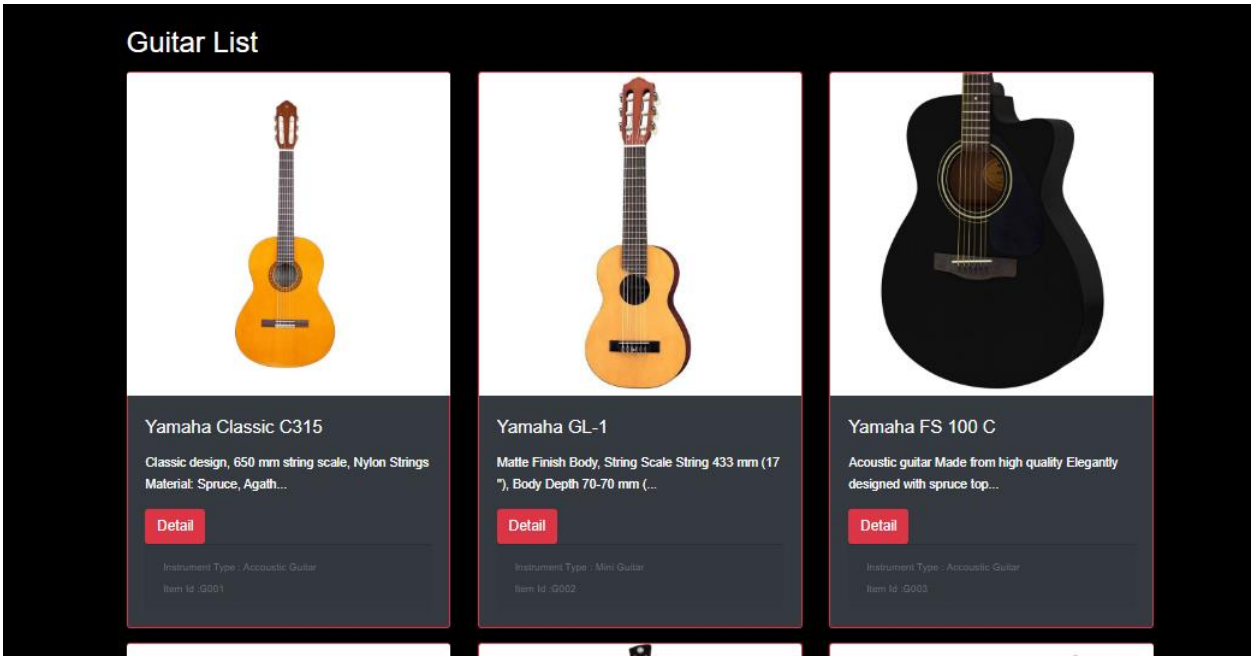


6. You can search the instrument that you want, by search it the name of the instrument itself at the navbar



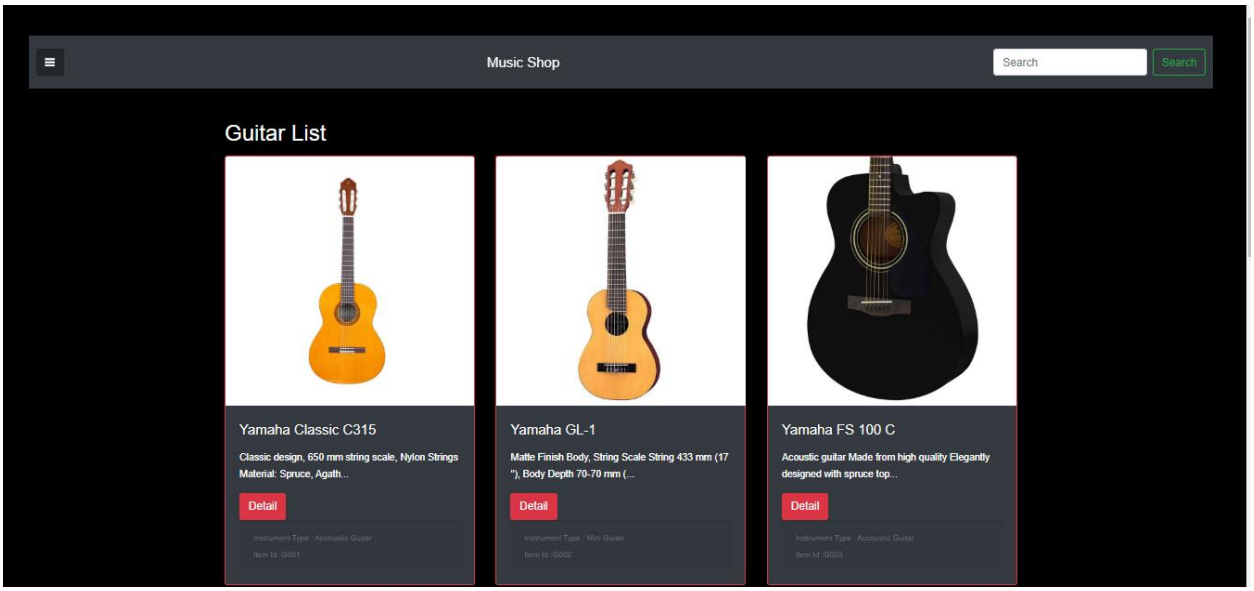
For example i search for yamaha, and it will popped up the instrument with yamaha name



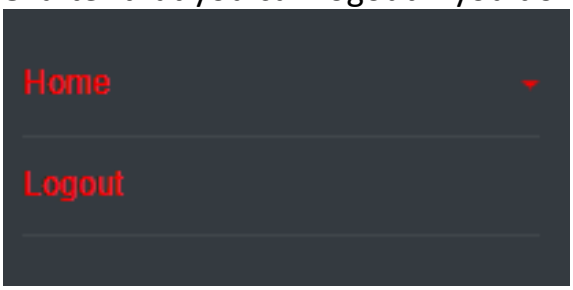


7. if you disturbed by the side bar, you can hide it by click the toogle menu at the side bar

And the sidebar is hidden



8. 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.