A project report:

# “CMS project on Phalcon Framework”

For

# Human Resources website

By:

Amine Amejjoud - Taha El Khaoua

Naoufal Jayed - Imane Basbassi

Najib Raihan

Supervised By :

**Ms.** Sara Koulali

# Acknowledgements :

We would like to express our teacher **Ms. Sara Koulali** for giving us the opportunity to work under their guidance and help us gain immensely enriching professional experience. And give us valuable inputs and ideas right from the selection of topic for project till its successful completion.

## Objectifs:

1. Understanding the MVC Architecture.
2. Implementing MVC patterns to solve our problem.

## Problem statement:

Developing an application web which make facilities for recruiters to find experienced people's profiles in any working's field .

## Problem solution:

Using Phalcon Framework , and implement MVC patterns to create web application ( HR platform ) .

## Functional description:

1- the graduated people register in these platform .

2- they create their profiles ( personal informations + certificates + contact ).

3- Recruiters see the profiles.

4- Recruiter choose a target and contact it directly.

## Working Area:

**PhpStorm:** cross-platform IDE for PHP. it provides an editor for PHP, HTML and JavaScript .

**WAMP/XAMPP:** web server solution and interpreters for scripts written in the PHP.

**Git:** is a distributed version-control system for tracking changes in source code during software development

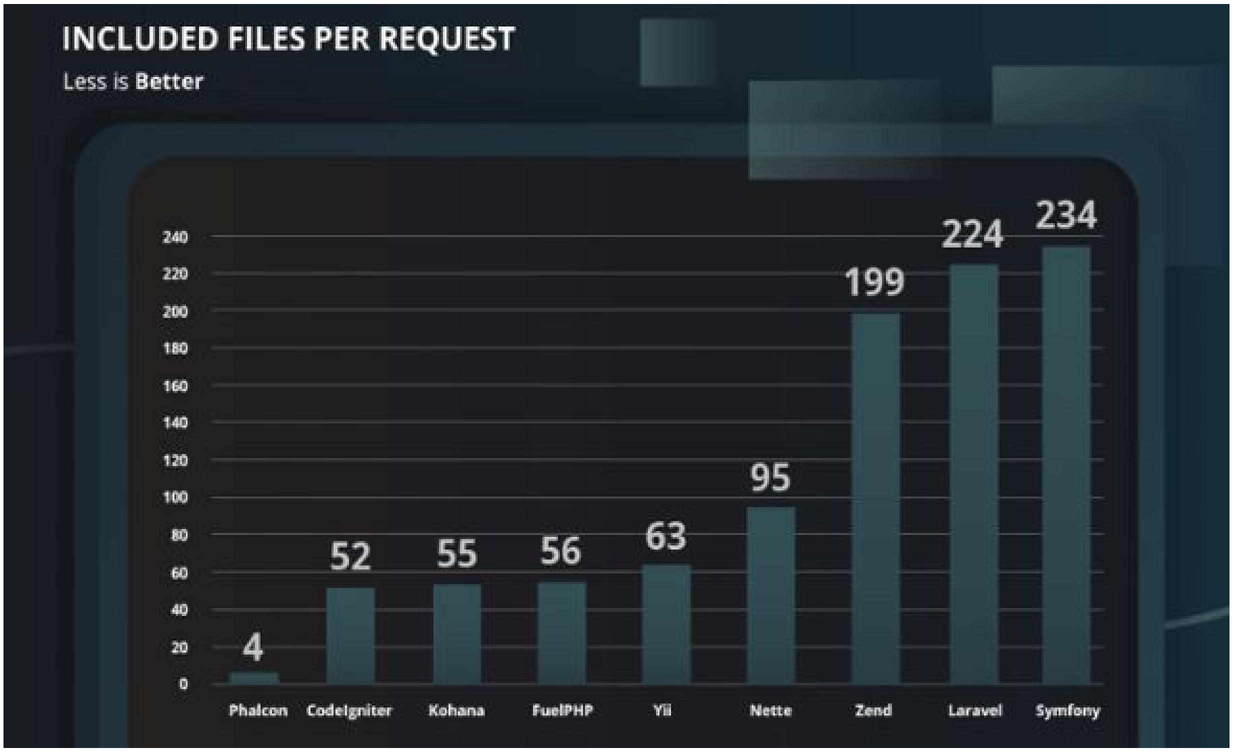
# Table of Contents

# Why Phalcon Framework:

## I. What’s phalcon:

phalcon is a high-performance PHP framework based on the model-view-controller (MVC) architecture. Its initial release dates back to 2012 and it has always been open-source. It offers some unique features in comparison to other frameworks (traditional or existing) in PHP. Following are some of the most prominent features of Phalcon:

1. Phalcon is famous for being the first framework to implement *object-relational mapping (ORM)* in C.
2. create an extension “php_phalcon.dll ” to a web server which enhances execution speeds, reduces resource utilization, and ensures higher request processing compared to other PHP frameworks.
3. Phalcon was able to complete over 1400 requests per second. Meanwhile other frameworks had a harder time.



## II. Phalcon Vs Php Frameworks:

- **Phalcon Vs. Laravel :**

  laravel can't compete with Phalcon when it comes down to speed. But , laravel is comparatively easier to learn and has better debugging support.

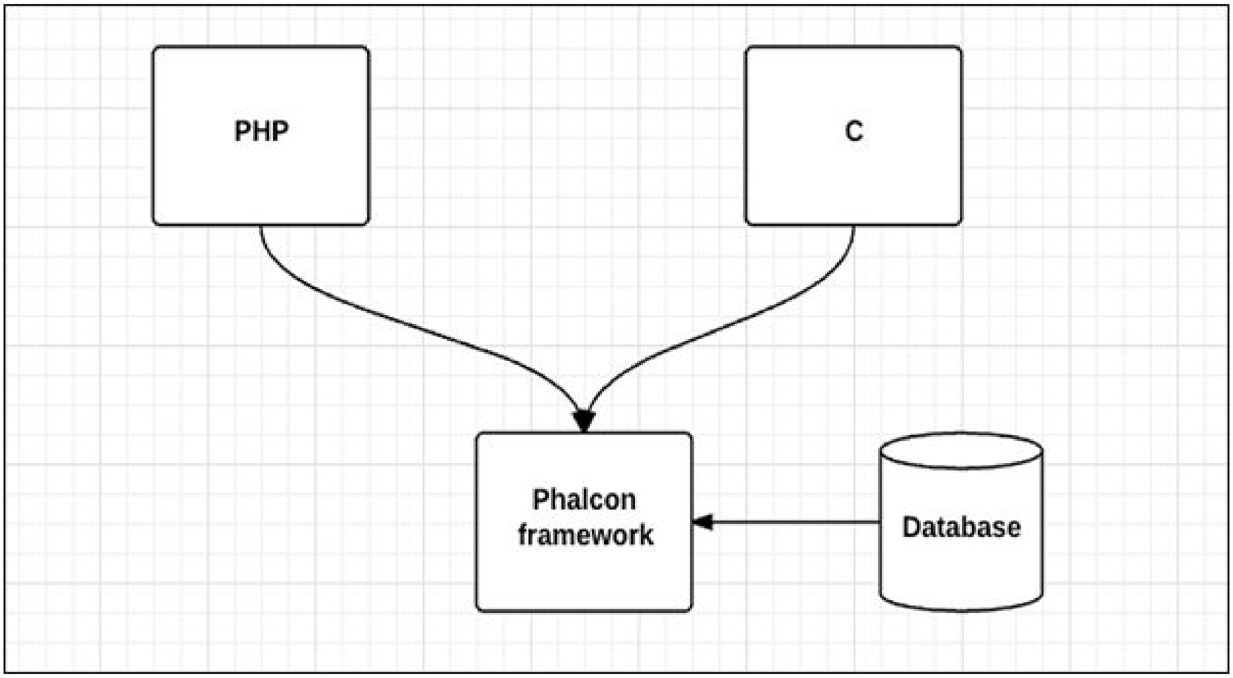
- **Phalcon Vs. CodeIgniter:**

  CodeIgniter may be able to catch up with Phalcon with appropriate hardware optimization.

- **Phalcon Vs. Symfony :**

  they are open-source projects with ORM support,Phalcon can process 50 percent more requests per second compared to Symfony.

Phalcon 's performance : Phalcon has a combination of both PHP and C; each of them can be used as a standalone module. The compilation of every request is considered on a higher speed in Phalcon which makes everything seem out-of-the-box.

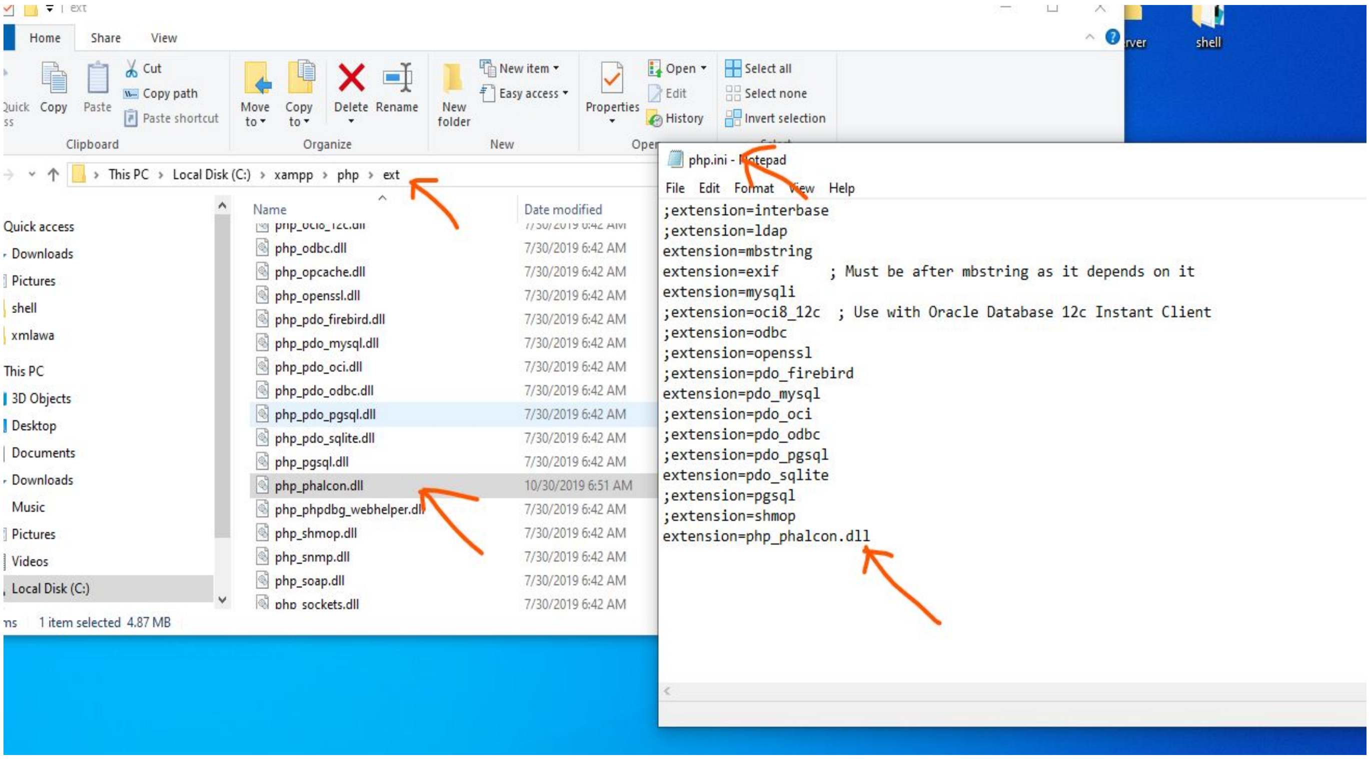


## Phalcon- Environmental Setup:

**Step 1** : Phalcon installation is completely dependent on dll file. DLL (Dynamic Link Library) creates the required package and plugins for Phalcon.

**Step 2** : **Download the required dll file from** https://phalcon.io/en-us/download/windows . Check for the appropriate configuration of the system and download the required dll file. After downloading the file, extract **phalcon-php.dll** to **/php/ext** in **the xampp folder**.

**Step 3** : Edit the path in **php.ini** file to get it configured in a similar manner of other .dll files.

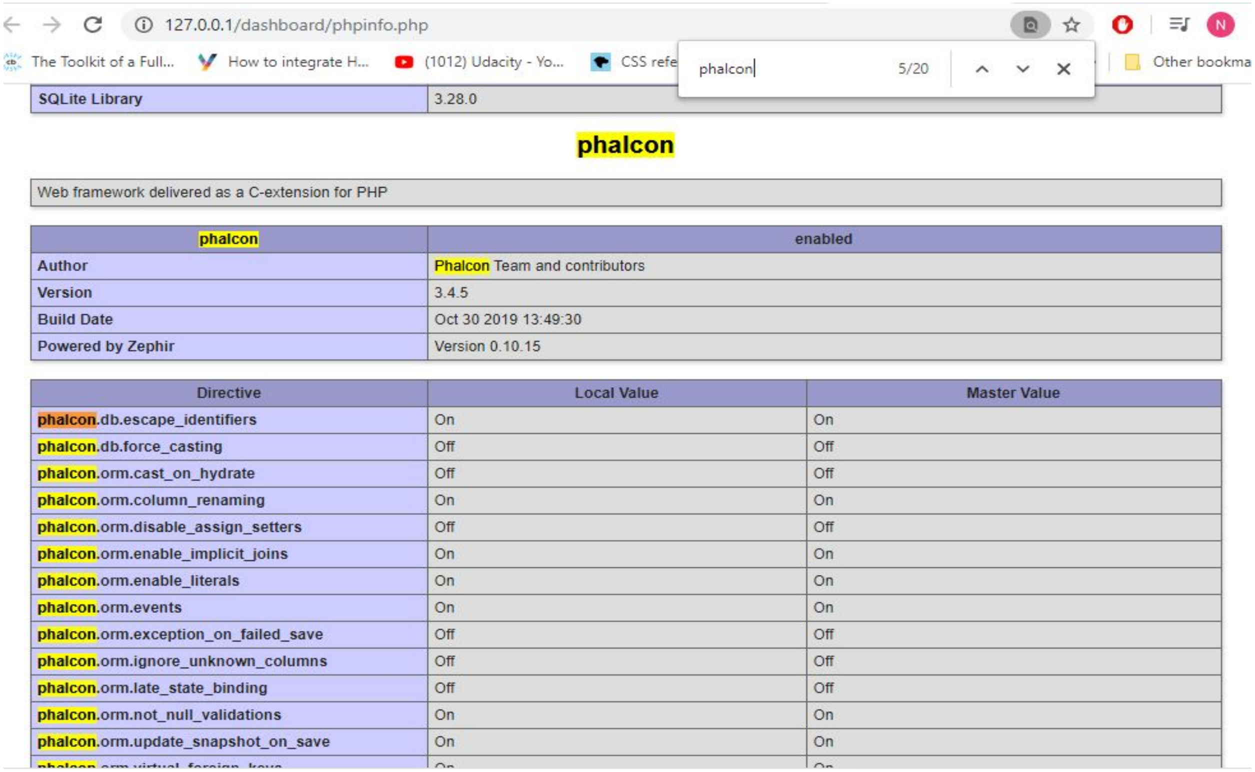


| Name | Date modified |
|---|---|
| php_oci8_12c.dll | 7/30/2019 6:42 AM |
| php_odbc.dll | 7/30/2019 6:42 AM |
| php_opcache.dll | 7/30/2019 6:42 AM |
| php_openssl.dll | 7/30/2019 6:42 AM |
| php_pdo_firebird.dll | 7/30/2019 6:42 AM |
| php_pdo_mysql.dll | 7/30/2019 6:42 AM |
| php_pdo_oci.dll | 7/30/2019 6:42 AM |
| php_pdo_odbc.dll | 7/30/2019 6:42 AM |
| php_pdo_pgsql.dll | 7/30/2019 6:42 AM |
| php_pdo_sqlite.dll | 7/30/2019 6:42 AM |
| php_pgsql.dll | 7/30/2019 6:42 AM |
| php_phalcon.dll | 10/30/2019 6:51 AM |
| php_phpdbg_webhelper.dll | 7/30/2019 6:42 AM |
| php_shmop.dll | 7/30/2019 6:42 AM |
| php_snmp.dll | 7/30/2019 6:42 AM |
| php_soap.dll | 7/30/2019 6:42 AM |
| php_sockets.dll | 7/30/2019 6:42 AM |

```
;extension=interbase
;extension=ldap
extension=mbstring
extension=exif      ; Must be after mbstring as it depends on it
extension=mysqli
;extension=oci8_12c  ; Use with Oracle Database 12c Instant Client
;extension=odbc
;extension=openssl
;extension=pdo_firebird
extension=pdo_mysql
;extension=pdo_oci
;extension=pdo_odbc
;extension=pdo_pgsql
extension=pdo_sqlite
;extension=pgsql
;extension=shmop
extension=php_phalcon.dll
```

**Step 4** : Once the path is edited, **restart the xampp/wamp** stack. It will be clearly visible in **the dashboard**, once the dll file is properly set.

**Step 5** : After downloading **the package** from https://github.com/phalcon/phalcon-devtools , set **the path variable in the system** properties.



| SQLite Library | 3.28.0 |
|---|---|

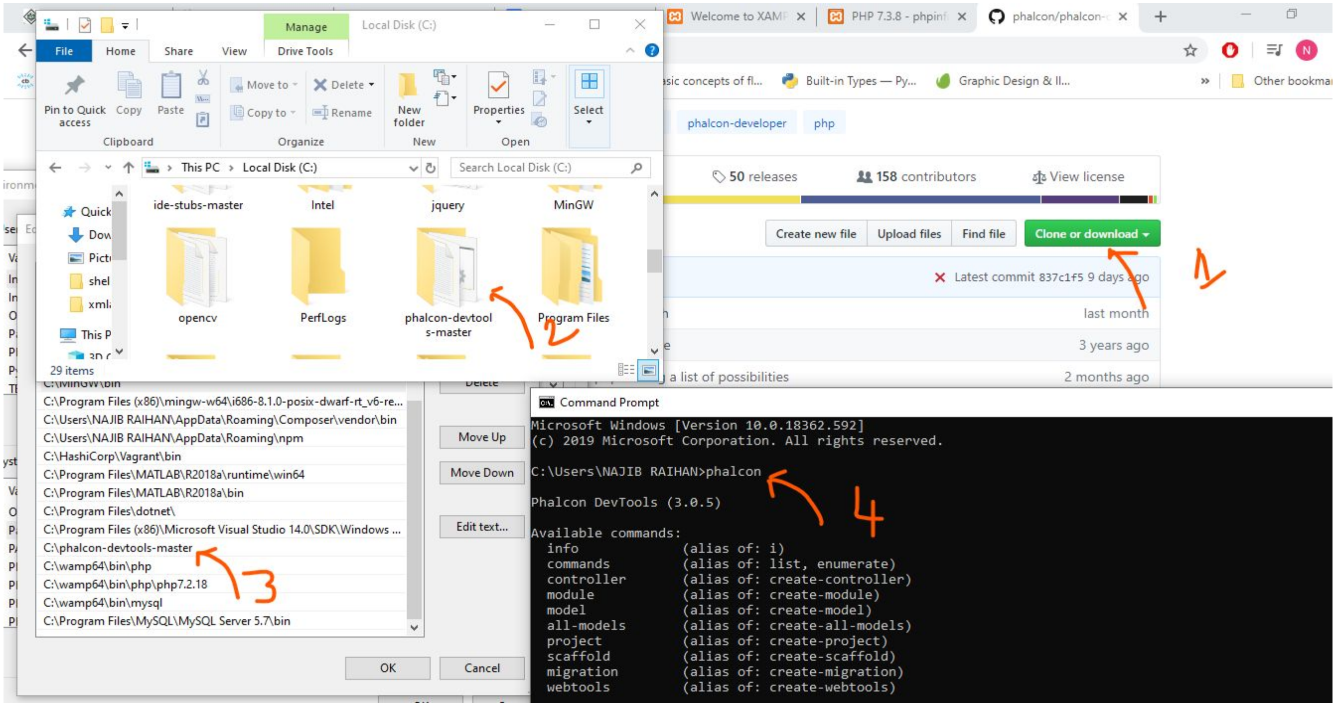
**phalcon**

Web framework delivered as a C-extension for PHP

| phalcon | enabled |
|---|---|
| Author | Phalcon Team and contributors |
| Version | 3.4.5 |
| Build Date | Oct 30 2019 13:49:30 |
| Powered by Zephir | Version 0.10.15 |

| Directive | Local Value | Master Value |
|---|---|---|
| phalcon.db.escape_identifiers | On | On |
| phalcon.db.force_casting | Off | Off |
| phalcon.orm.cast_on_hydrate | Off | Off |
| phalcon.orm.column_renaming | On | On |
| phalcon.orm.disable_assign_setters | Off | Off |
| phalcon.orm.enable_implicit_joins | On | On |
| phalcon.orm.enable_literals | On | On |
| phalcon.orm.events | On | On |
| phalcon.orm.exception_on_failed_save | Off | Off |
| phalcon.orm.ignore_unknown_columns | Off | Off |
| phalcon.orm.late_state_binding | Off | Off |
| phalcon.orm.not_null_validations | On | On |
| phalcon.orm.update_snapshot_on_save | On | On |
| phalcon.orm.virtual_foreign_keys | On | On |

**Step 6** : The dll files and Phalcon tools together help in creating the our web application. The user can verify through command prompt whether Phalcon framework has been successfully installed. The output will be displayed as shown in the following screenshot.

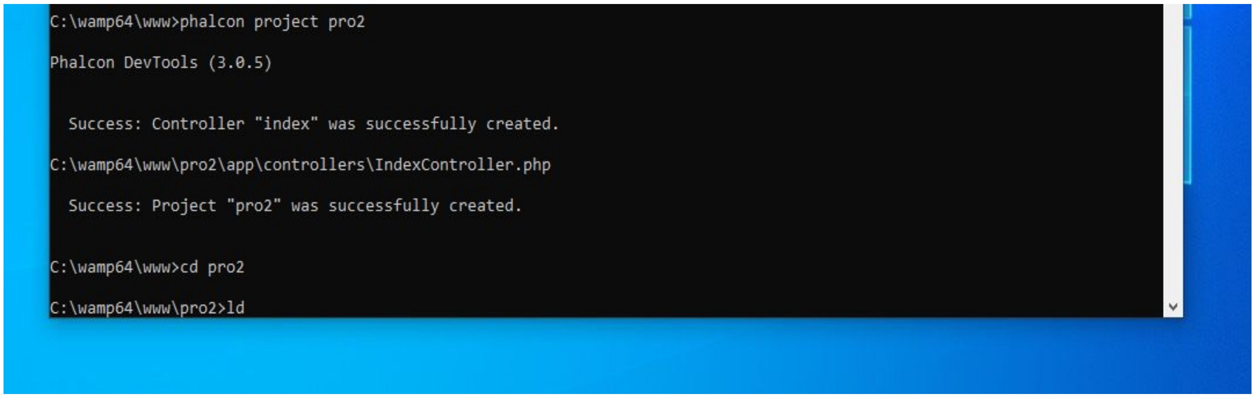


C:\MinGW\bin
C:\Program Files (x86)\mingw-w64\i686-8.1.0-posix-dwarf-rt_v6-re...
C:\Users\NAJIB RAIHAN\AppData\Roaming\Composer\vendor\bin
C:\Users\NAJIB RAIHAN\AppData\Roaming\npm
C:\HashiCorp\Vagrant\bin
C:\Program Files\MATLAB\R2018a\runtime\win64
C:\Program Files\MATLAB\R2018a\bin
C:\Program Files\dotnet\
C:\Program Files (x86)\Microsoft Visual Studio 14.0\SDK\Windows ...
C:\phalcon-devtools-master
C:\wamp64\bin\php
C:\wamp64\bin\php\php7.2.18
C:\wamp64\bin\mysql
C:\Program Files\MySQL\MySQL Server 5.7\bin

Command Prompt

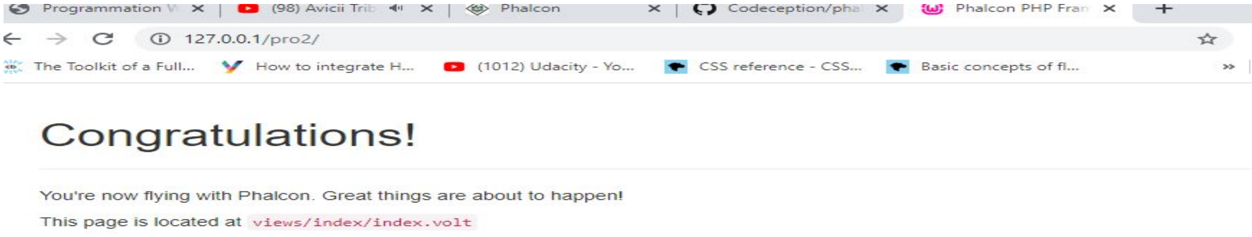
```
Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\NAJIB RAIHAN>phalcon

Phalcon DevTools (3.0.5)

Available commands:
  info           (alias of: i)
  commands       (alias of: list, enumerate)
  controller     (alias of: create-controller)
  module         (alias of: create-module)
  model          (alias of: create-model)
  all-models     (alias of: create-all-models)
  project        (alias of: create-project)
  scaffold       (alias of: create-scaffold)
  migration      (alias of: create-migration)
  webtools       (alias of: create-webtools)
```

**Step 7** : Once this necessary output is received, create a our project ”pro2” using the following command ;



```
C:\wamp64\www>phalcon project pro2

Phalcon DevTools (3.0.5)


  Success: Controller "index" was successfully created.

C:\wamp64\www\pro2\app\controllers\IndexController.php

  Success: Project "pro2" was successfully created.


C:\wamp64\www>cd pro2

C:\wamp64\www\pro2>ld
```

**Step 8** : The web application is successfully created. Click the following URL − [http://127.0.0.1/pro2](http://127.0.0.1/pro2)



Congratulations!

You're now flying with Phalcon. Great things are about to happen!

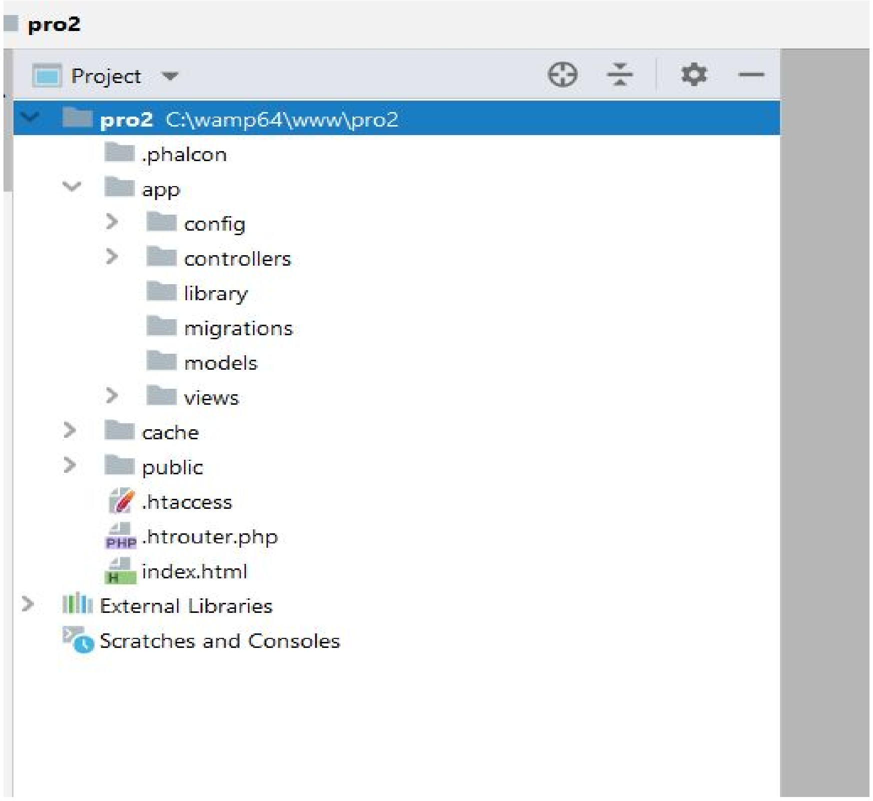
This page is located at `views/index/index.volt`

## Phalcon- Application structure:

Once a project is created, the directory structure will be visible as follows in the **xampp/htdocs** folder. Following are the folders and sub-folders of the project:

## app folder:

This folder consists of all vital script files and folders. The complete web application is designed on the basis of “app” folder. The configuration files help in assisting the necessary configuration for running the application smoothly.

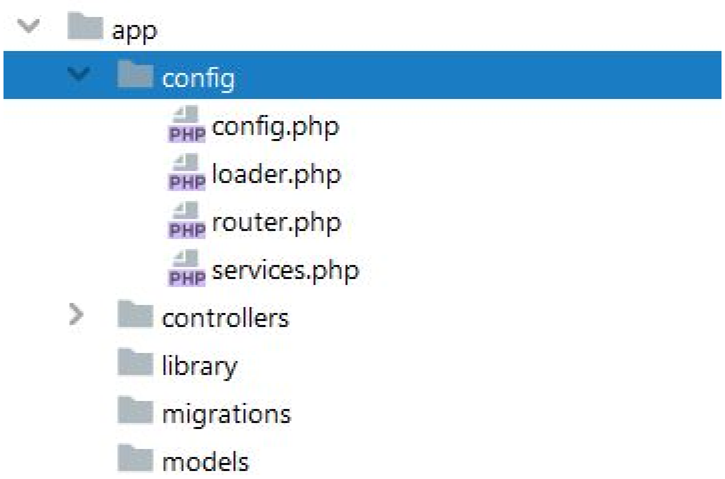


```
pro2
Project
pro2  C:\wamp64\www\pro2
    .phalcon
    app
        config
        controllers
        library
        migrations
        models
        views
    cache
    public
    .htaccess
    .htrouter.php
    index.html
External Libraries
Scratches and Consoles
```

It consists of config, controllers, library, migrations, models and views.

## config folder:

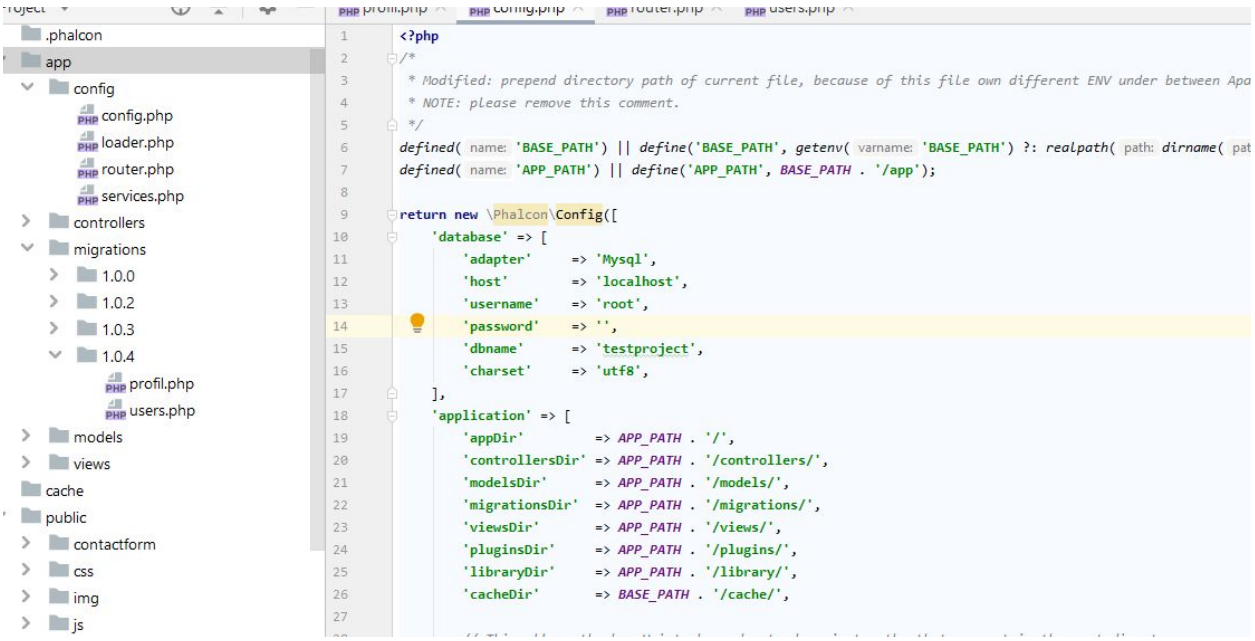
All the configuration required for the web application in Phalcon is comprised in this folder.



```
app
    config
        config.php
        loader.php
        router.php
        services.php
    controllers
    library
    migrations
    models
```

All the configuration required for the web application in Phalcon is comprised in this folder. It includes information related to :

### 1 - database connectivity: config.php



```php
<?php
/*
 * Modified: prepend directory path of current file, because of this file own different ENV under between Apa
 * NOTE: please remove this comment.
 */
defined( name: 'BASE_PATH') || define('BASE_PATH', getenv( varname: 'BASE_PATH') ?: realpath( path: dirname( pat
defined( name: 'APP_PATH') || define('APP_PATH', BASE_PATH . '/app');

return new \Phalcon\Config([
    'database' => [
        'adapter'     => 'Mysql',
        'host'        => 'localhost',
        'username'    => 'root',
        'password'    => '',
        'dbname'      => 'testproject',
        'charset'     => 'utf8',
    ],
    'application' => [
        'appDir'         => APP_PATH . '/',
        'controllersDir' => APP_PATH . '/controllers/',
        'modelsDir'      => APP_PATH . '/models/',
        'migrationsDir'  => APP_PATH . '/migrations/',
        'viewsDir'       => APP_PATH . '/views/',
        'pluginsDir'     => APP_PATH . '/plugins/',
        'libraryDir'     => APP_PATH . '/library/',
        'cacheDir'       => BASE_PATH . '/cache/',

```

### 2- Classes loading : loader.php

It extends the existing class of \Phalcon\Loader(). The loader class registers the directories which requires web application.

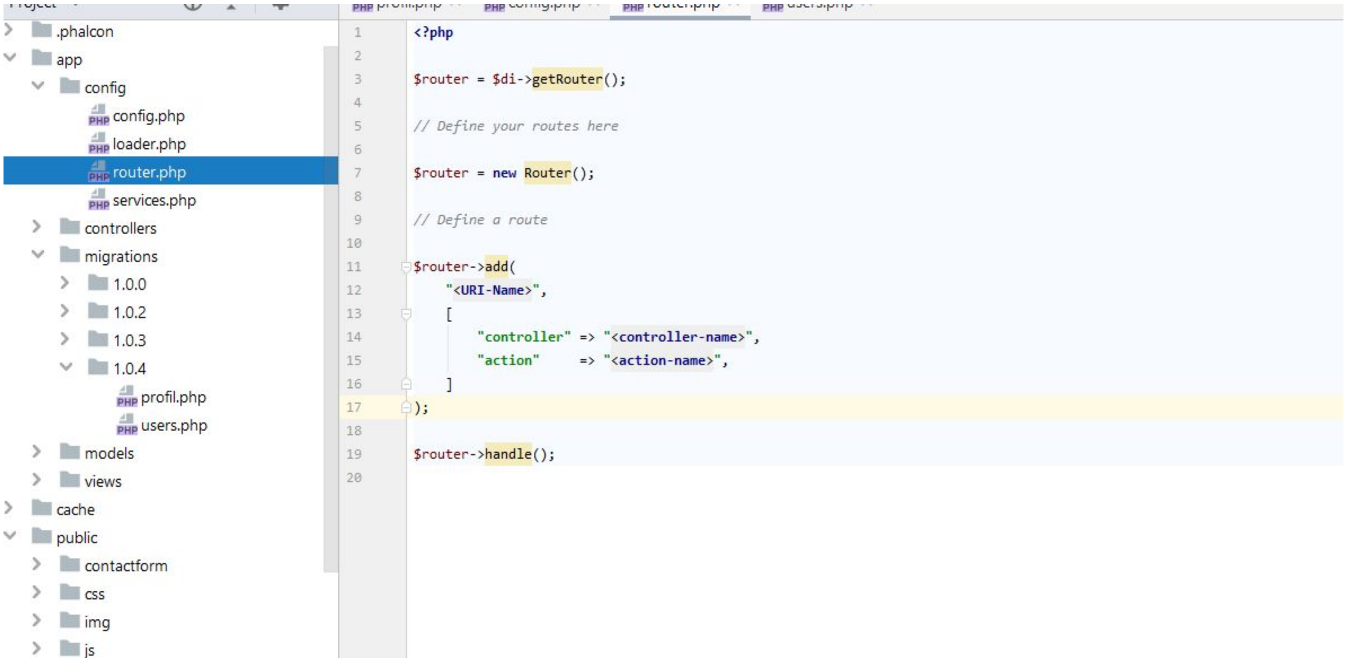
### 3 - Routing : router.php

The router component allows to define routes that are mapped to the controllers or handlers that should receive the request. A router parses a URI as per the information received.

Every router in the web application has two modes −

- MVC mode
- Match-only mode

The first mode is ideal for working with MVC applications. Following is the syntax to define a route in Phalcon.



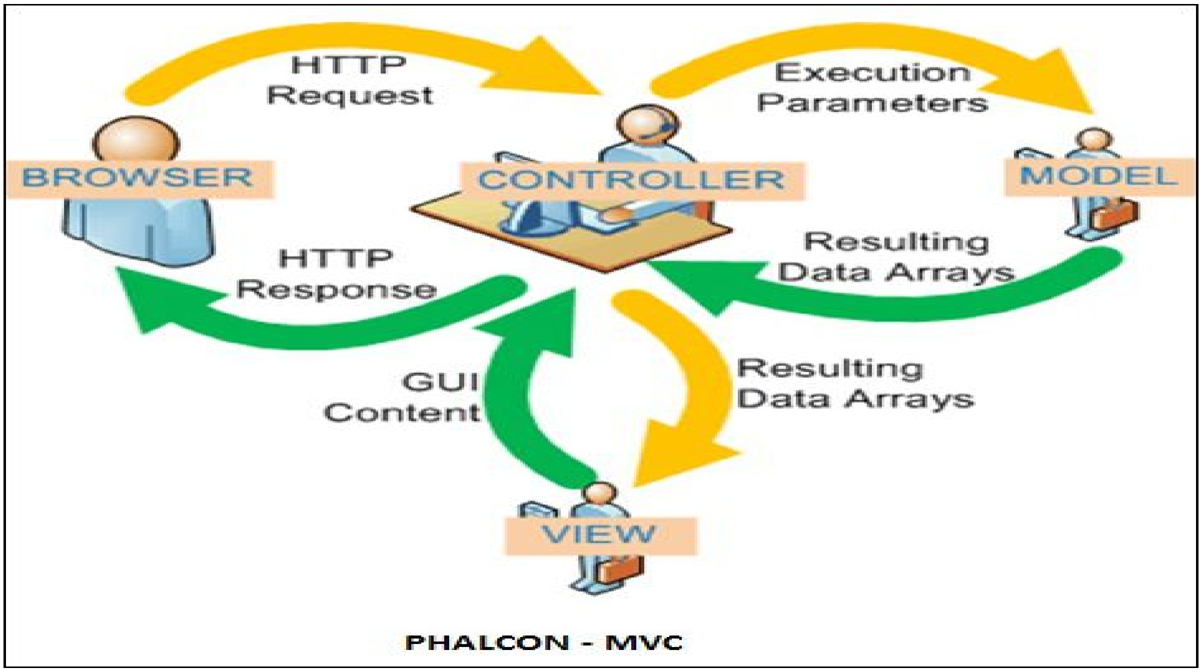
```
> .phalcon
v app
  v config
      config.php
      loader.php
      router.php
      services.php
  > controllers
  v migrations
    > 1.0.0
    > 1.0.2
    > 1.0.3
    v 1.0.4
        profil.php
        users.php
  > models
  > views
> cache
v public
  > contactform
  > css
  > img
  > js
```

```php
<?php

$router = $di->getRouter();

// Define your routes here

$router = new Router();

// Define a route

$router->add(
    "<URI-Name>",
    [
        "controller" => "<controller-name>",
        "action"     => "<action-name>",
    ]
);

$router->handle();

```

## Model View Controller (MVC) :

is a software design and structural pattern for developing web based applications. This software architectural separates the representation of information from the user's interaction with it.

The MVC model defines the web applications with three logic layers:



## Models:

Model in MVC architecture includes the logic of application. Model is the core interaction with the database. It should be able to manage updating, deleting, inserting, and fetching of records as per the user’s request.

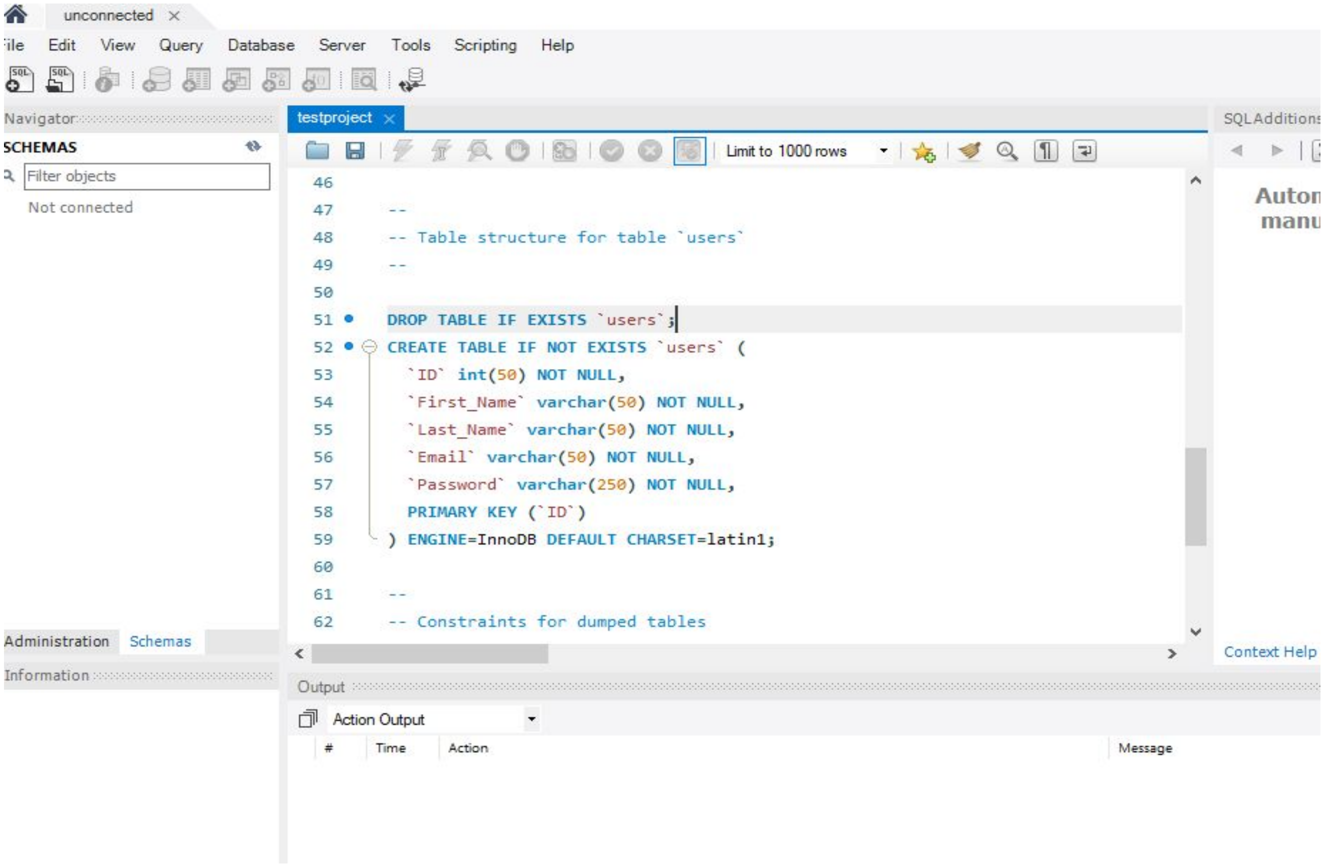
For understanding the model interaction in Phalcon PHP framework, following steps should be followed.

**Example:**

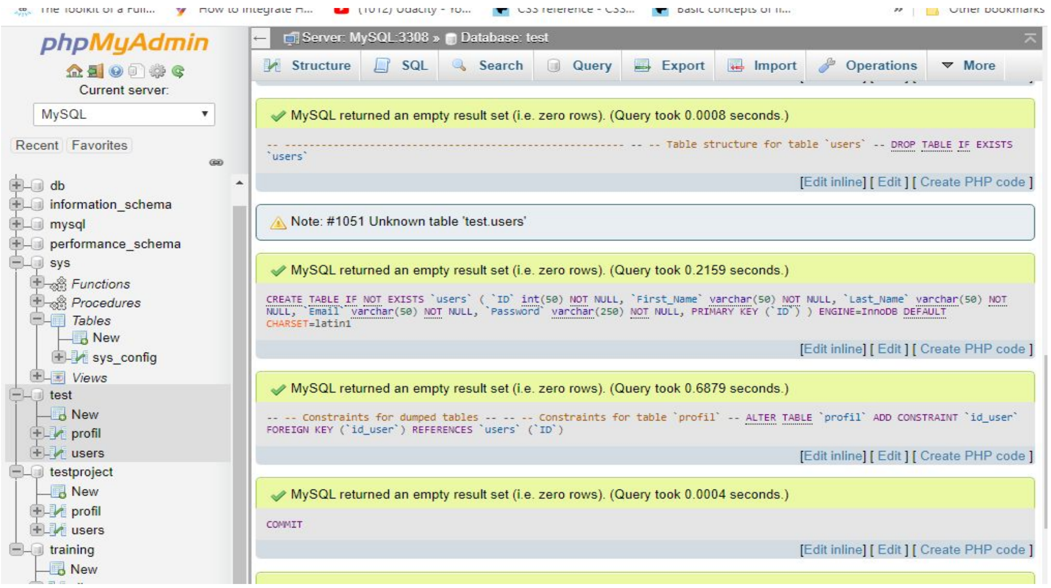
Create Users Model ;

**step 1**:Creation of database **“testproject”**.

**step 2**:In the phpmyadmin section, click the Databases tab, mention the database name and further click the **Import** to import “**testproject.sql” ,**that contains **Users** table button as shown in the following screenshot.



```

--
-- Table structure for table `users`
--

DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `ID` int(50) NOT NULL,
  `First_Name` varchar(50) NOT NULL,
  `Last_Name` varchar(50) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Password` varchar(250) NOT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Constraints for dumped tables
```



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

```
-- ------------------------------------------------------------ -- -- Table structure for table `users` -- DROP TABLE IF EXISTS
`users`
```

[Edit inline] [ Edit ] [ Create PHP code ]

Note: #1051 Unknown table 'test.users'

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.2159 seconds.)

```
CREATE TABLE IF NOT EXISTS `users` ( `ID` int(50) NOT NULL, `First_Name` varchar(50) NOT NULL, `Last_Name` varchar(50) NOT
NULL, `Email` varchar(50) NOT NULL, `Password` varchar(250) NOT NULL, PRIMARY KEY (`ID`) ) ENGINE=InnoDB DEFAULT
CHARSET=latin1
```

[Edit inline] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.6879 seconds.)

```
-- -- Constraints for dumped tables -- -- -- Constraints for table `profil` -- ALTER TABLE `profil` ADD CONSTRAINT `id_user`
FOREIGN KEY (`id_user`) REFERENCES `users` (`ID`)
```

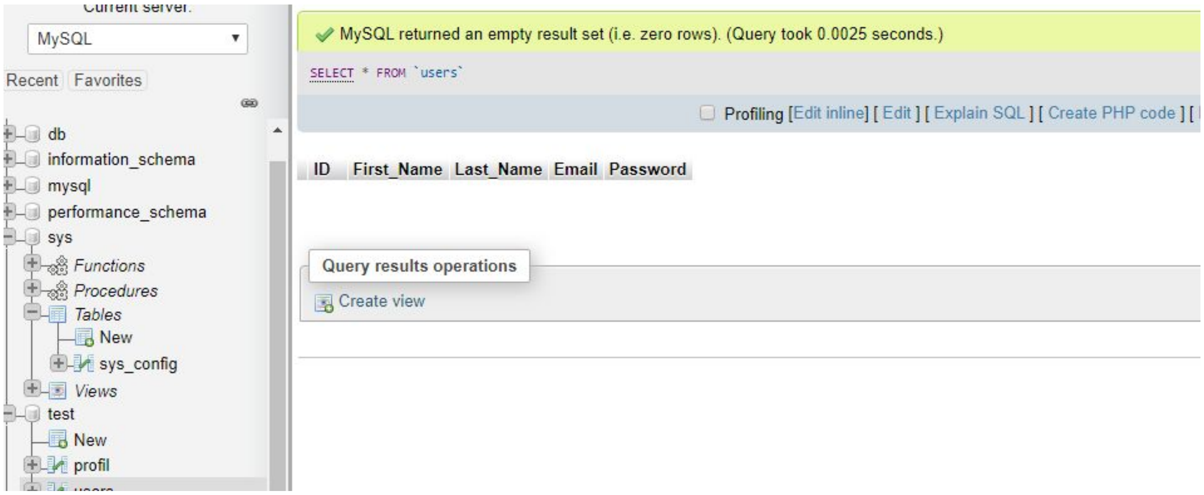
[Edit inline] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

```
COMMIT
```

[Edit inline] [ Edit ] [ Create PHP code ]

**Result:**



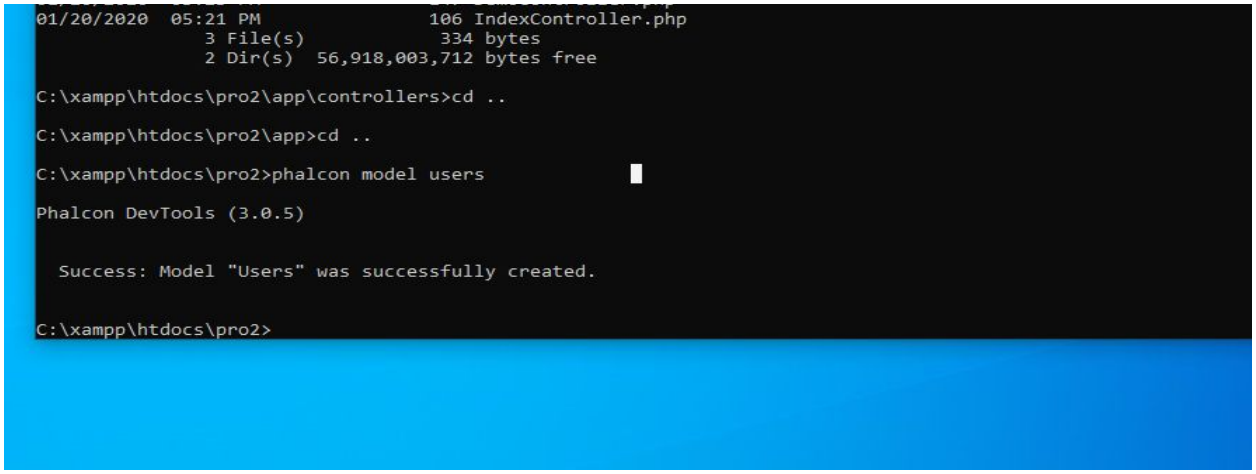
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0025 seconds.)

```
SELECT * FROM `users`
```

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

| ID | First_Name | Last_Name | Email | Password |
|---|---|---|---|---|

**step 3**:-Use the following command to create any model in Phalcon framework.

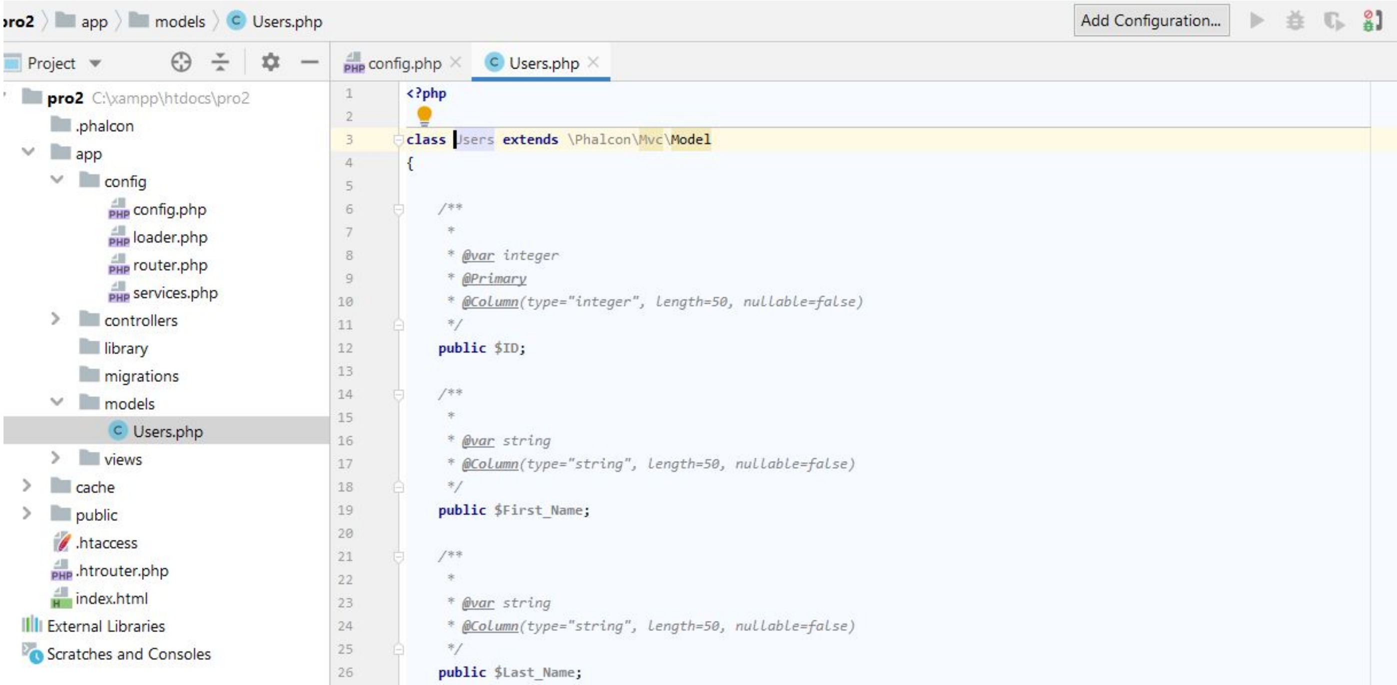


```
01/20/2020  05:21 PM               106 IndexController.php
               3 File(s)            334 bytes
               2 Dir(s)  56,918,003,712 bytes free

C:\xampp\htdocs\pro2\app\controllers>cd ..

C:\xampp\htdocs\pro2\app>cd ..

C:\xampp\htdocs\pro2>phalcon model users

Phalcon DevTools (3.0.5)


  Success: Model "Users" was successfully created.


C:\xampp\htdocs\pro2>
```

# Models folder:

Models include all the logic required to interact with the database. It is actually used for data representation.

**step 4:** Model created successfully is present in the models folder. Use the following path to view where the model is created.



```php
<?php

class Users extends \Phalcon\Mvc\Model
{

    /**
     *
     * @var integer
     * @Primary
     * @Column(type="integer", length=50, nullable=false)
     */
    public $ID;

    /**
     *
     * @var string
     * @Column(type="string", length=50, nullable=false)
     */
    public $First_Name;

    /**
     *
     * @var string
     * @Column(type="string", length=50, nullable=false)
     */
    public $Last_Name;
```

# Controller:

A controller acts as the intermediary between the user and the system (model and view). It accepts the request from the user, through the view sends it to the model. The model manipulates it and sends the response to the controller, which is displayed as the output to the end user through view.

The controller receives such user output and translates it into the appropriate messages. These messages are used by view to display as appropriate responses.

For example, if we register through a sign-up form which includes details of the user such as username, email address and password, and click the Submit button, the data inserted or posted by the user is sent through the controller with the help of associated action or function.

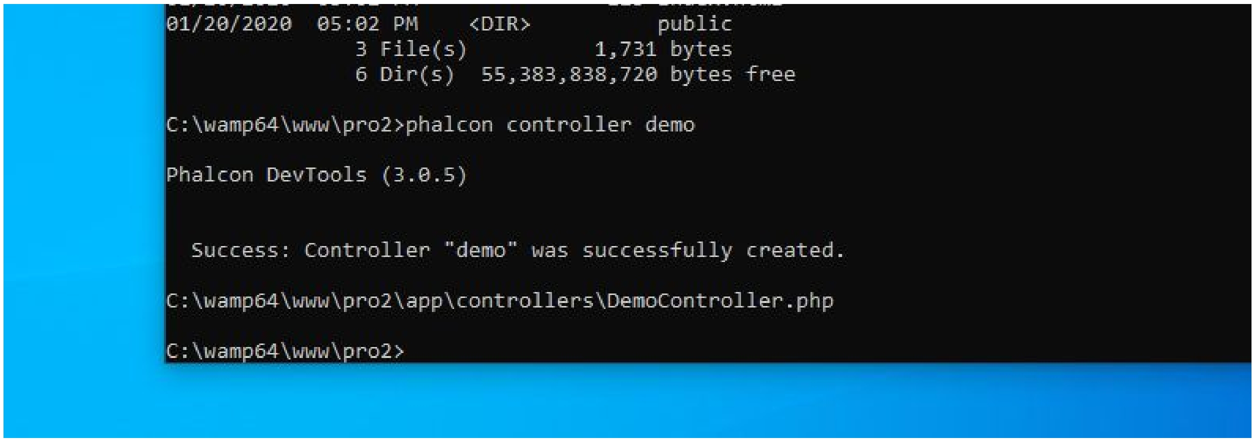
Features of a Controller:

A controller accepts inputs from the view and interacts with the associated model.

1. It helps in updating the model's state by sending commands to the model. It can also send commands to the associated view, which helps in changing the presentation of the view as per the model's state.
2. A controller acts as an intermediary between the model and the view.

Step to Create a Controller in Phalcon:

**Step 1** : Use the following command to create an associated controller.



```
01/20/2020  05:02 PM    <DIR>          public
               3 File(s)          1,731 bytes
               6 Dir(s)  55,383,838,720 bytes free

C:\wamp64\www\pro2>phalcon controller demo

Phalcon DevTools (3.0.5)


  Success: Controller "demo" was successfully created.

C:\wamp64\www\pro2\app\controllers\DemoController.php

C:\wamp64\www\pro2>
```

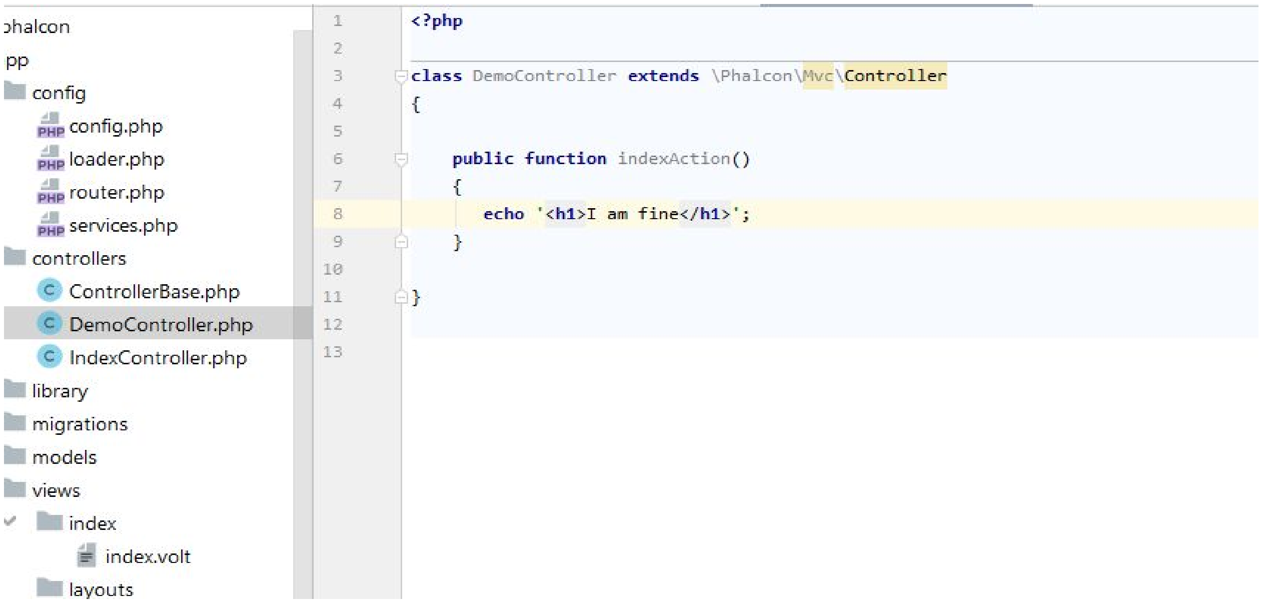
Note − The class names of the controllers must have the suffix “controller”. This implies a good naming convention which is followed in Phalcon.

By default, when the application is created in Phalcon PHP framework, it includes a controller named “IndexController”. This controller is invoked by default to trigger the actions.

This controller is extended by controller-base unlike other controllers which extends \Phalcon\Mvc\Controller.

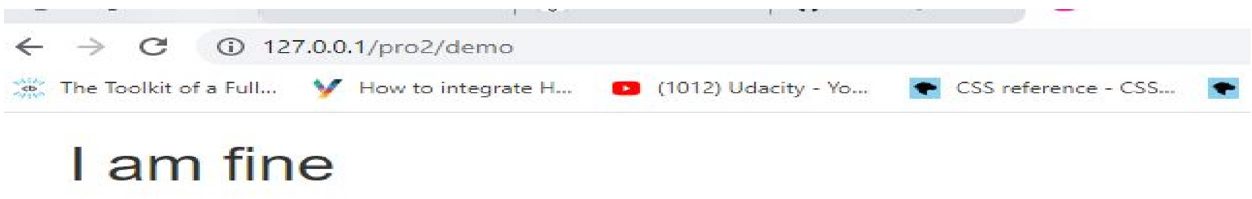
# Controllers folder:

All the controllers are included in this folder. They are used for processing requests and generating response.



```php
<?php

class DemoController extends \Phalcon\Mvc\Controller
{

    public function indexAction()
    {
        echo '<h1>I am fine</h1>';
    }

}
```

**Response** :



I am fine

## View:

A view is a visual representation of its model. View interacts with the model or its parts and gets the data necessary for the presentation from the model. This is achieved by sending requests and receiving appropriate responses. View includes all the data that end user sees.

Views are information being presented to the end user. A view can be considered as a web page with the appropriate response to be displayed. The response is received through the controller which interacts with the model.

Specifically in Phalcon, the view consists of Volt code, PHP and HTML. A set of special delimiters is available to enter in Volt mode:

{% ... %} is used to execute statements such as for-loops or assign values.

{{ ... }} prints the result of an expression to the template.

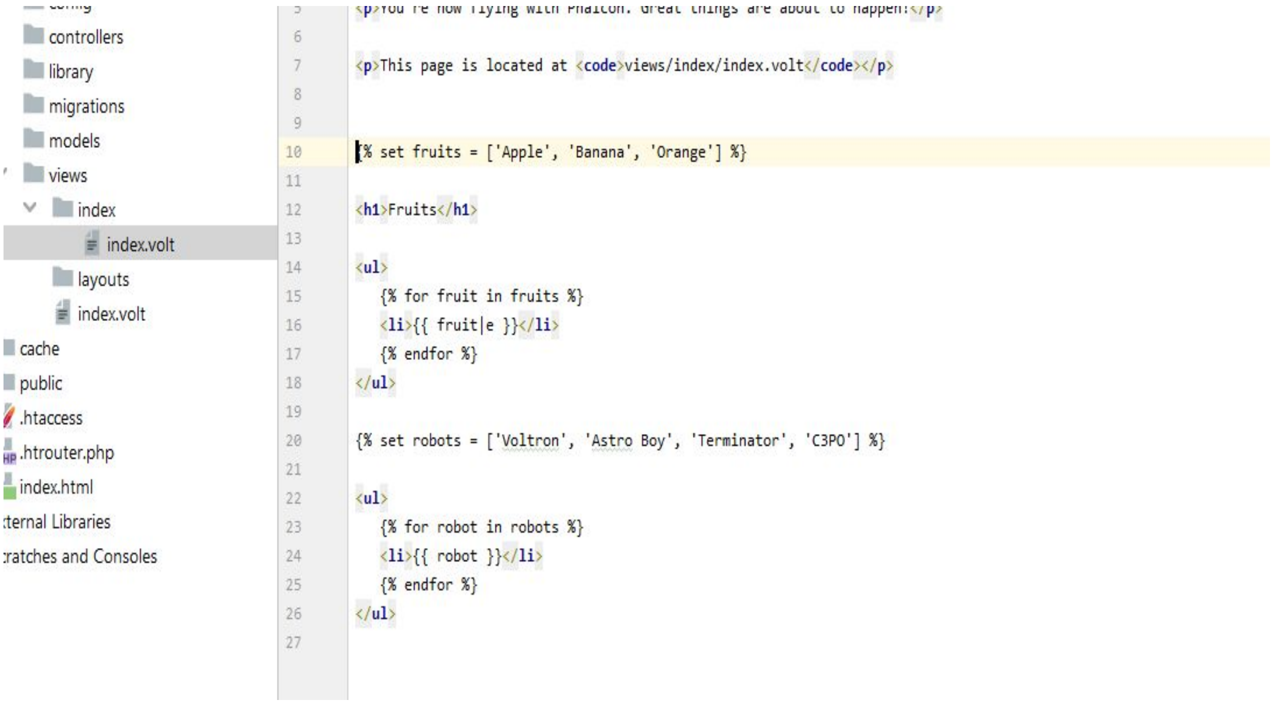
Views in Phalcon are basically classified into two types :

1. volt
2. phtml

## Difference between .volt and .phtml Files:

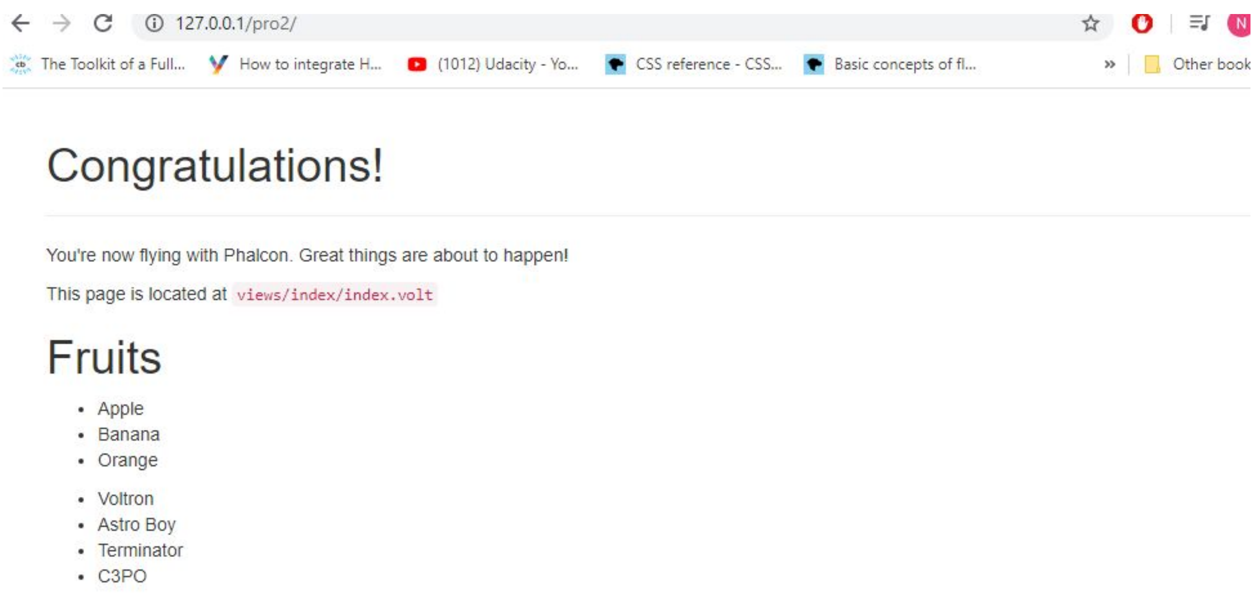
| .volt | .phtml |
|---|---|
| .volt extension is used when the template engine set up in the application is written in C. | .phtml is used when the template engine is PHP itself |
| It can be used as a stand-alone component | It cannot be used as a stand-alone component |
| Volt views are compiled to PHP code | phtml files itself includes PHP code so there is no need of compilation in Phalcon framework |

**Example:**



```
<p>You're now flying with Phalcon. Great things are about to happen!</p>

<p>This page is located at <code>views/index/index.volt</code></p>


{% set fruits = ['Apple', 'Banana', 'Orange'] %}

<h1>Fruits</h1>

<ul>
   {% for fruit in fruits %}
   <li>{{ fruit|e }}</li>
   {% endfor %}
</ul>

{% set robots = ['Voltron', 'Astro Boy', 'Terminator', 'C3PO'] %}

<ul>
   {% for robot in robots %}
   <li>{{ robot }}</li>
   {% endfor %}
</ul>

```

**Result:**



# Congratulations!

You're now flying with Phalcon. Great things are about to happen!

This page is located at `views/index/index.volt`

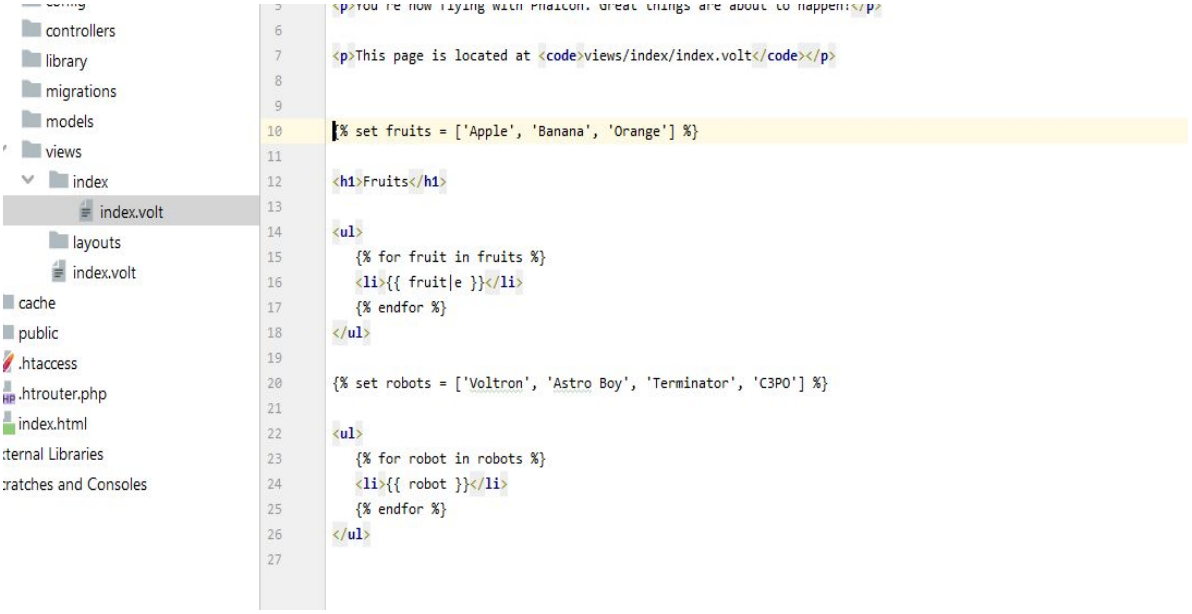
## Fruits

- Apple
- Banana
- Orange

- Voltron
- Astro Boy
- Terminator
- C3PO

## Views folder:

It constitutes all the views related to the web application. These views are displayed to the end users with the help of controllers.



```
<p>You're now flying with Phalcon. Great things are about to happen!</p>

<p>This page is located at <code>views/index/index.volt</code></p>


{% set fruits = ['Apple', 'Banana', 'Orange'] %}

<h1>Fruits</h1>

<ul>
   {% for fruit in fruits %}
   <li>{{ fruit|e }}</li>
   {% endfor %}
</ul>

{% set robots = ['Voltron', 'Astro Boy', 'Terminator', 'C3PO'] %}

<ul>
   {% for robot in robots %}
   <li>{{ robot }}</li>
   {% endfor %}
</ul>

```

## Migrations folder:

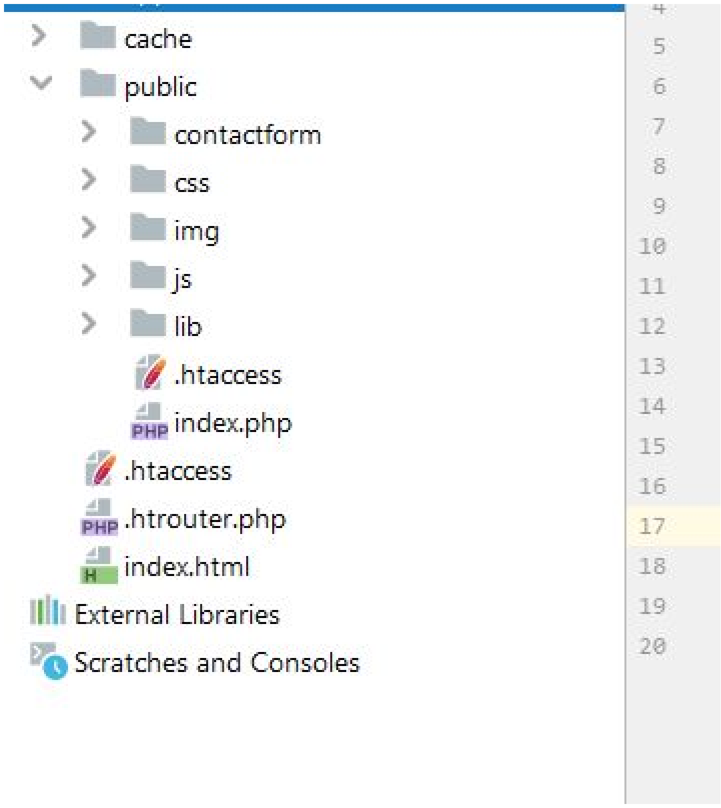
This sub-folder consists of all the files associated with data migration, which can also be used in any other framework.

## Cache folder:

This directory includes data related to caching, which helps in improving the performance.

## Public folder:

It includes all the folders for asset management purpose which comprises of CSS, JavaScript, files to be uploaded, and some meta data.



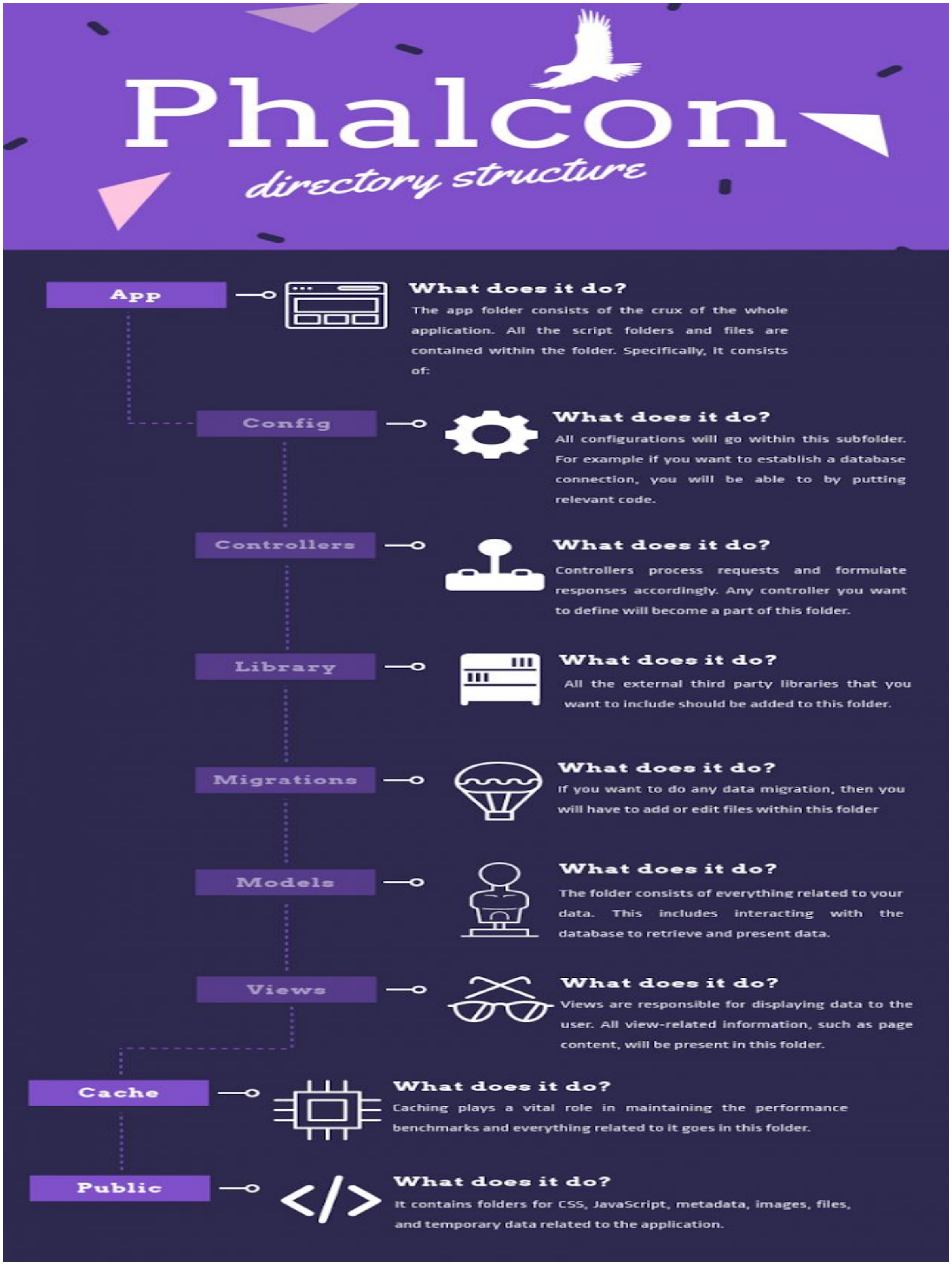
```
> cache
v public
    > contactform
    > css
    > img
    > js
    > lib
      .htaccess
      index.php
  .htaccess
  .htrouter.php
  index.html
External Libraries
Scratches and Consoles
```

## .htaccess File

Web servers running on Apache Web Server software use .htaccess as a configuration file. When it is placed in a directory, all the necessary configuration is loaded as soon as the server is started.

For example, it is possible to configure a website so that it will available only to specific IP addresses with .htaccess file.

**Phalcon Structure:**



# Phalcon

*directory structure*

**App**

**What does it do?**

The app folder consists of the crux of the whole application. All the script folders and files are contained within the folder. Specifically, it consists of:

**Config**

**What does it do?**

All configurations will go within this subfolder. For example if you want to establish a database connection, you will be able to by putting relevant code.

**Controllers**

**What does it do?**

Controllers process requests and formulate responses accordingly. Any controller you want to define will become a part of this folder.

**Library**

**What does it do?**

All the external third party libraries that you want to include should be added to this folder.

**Migrations**

**What does it do?**

If you want to do any data migration, then you will have to add or edit files within this folder

**Models**

**What does it do?**

The folder consists of everything related to your data. This includes interacting with the database to retrieve and present data.

**Views**

**What does it do?**

Views are responsible for displaying data to the user. All view-related information, such as page content, will be present in this folder.

**Cache**

**What does it do?**

Caching plays a vital role in maintaining the performance benchmarks and everything related to it goes in this folder.

**Public**

**What does it do?**

It contains folders for CSS, JavaScript, metadata, images, files, and temporary data related to the application.

# Application Design Document :

## Models :

### -User

This model consists of user login and registration. It maintains the user's information and also user's files. It consists of two tables, one to include user's personal information and other to maintain user's profile ( Our recruiters sees that ). The input to database for personal information is given by the user in registration page.

### -Profile

This model consists of user build his own profile.

## Specification of Functions 's App :

In this part ,we will  to identify the services that must be provided by our application, we will detail the specific needs which consist of functional needs and non-functional needs:

**1- Functional needs :**

- The graduated people must have an account (Login).
- The graduated people must have a username and password to Connect (Signup).
- The application must allow recruiters to see directly user 's profile , without going through Signup 's operation.
- Each candidate is defined by his own information.

**1- Non-Functional needs :**

In order to  guarantee satisfaction of the users, some constraints must be taken into our application:

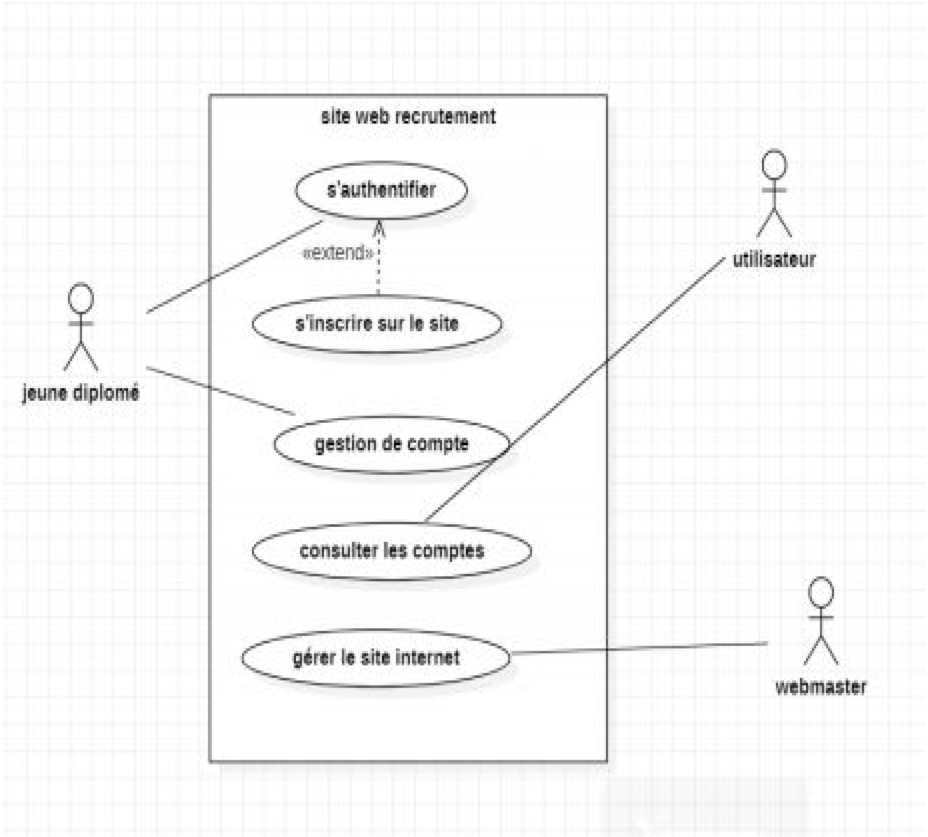
- **user-friendly**: The application must provide a simple view for the user in order to facilitate the operation of the application services.
- **Speed** : The application must ensure the speed of response and the reliability of the result obtained
- **Security**: The application must ensure the confidentiality of the data of users.

# Use case diagrams:

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

## Use case 1 : recruiting website

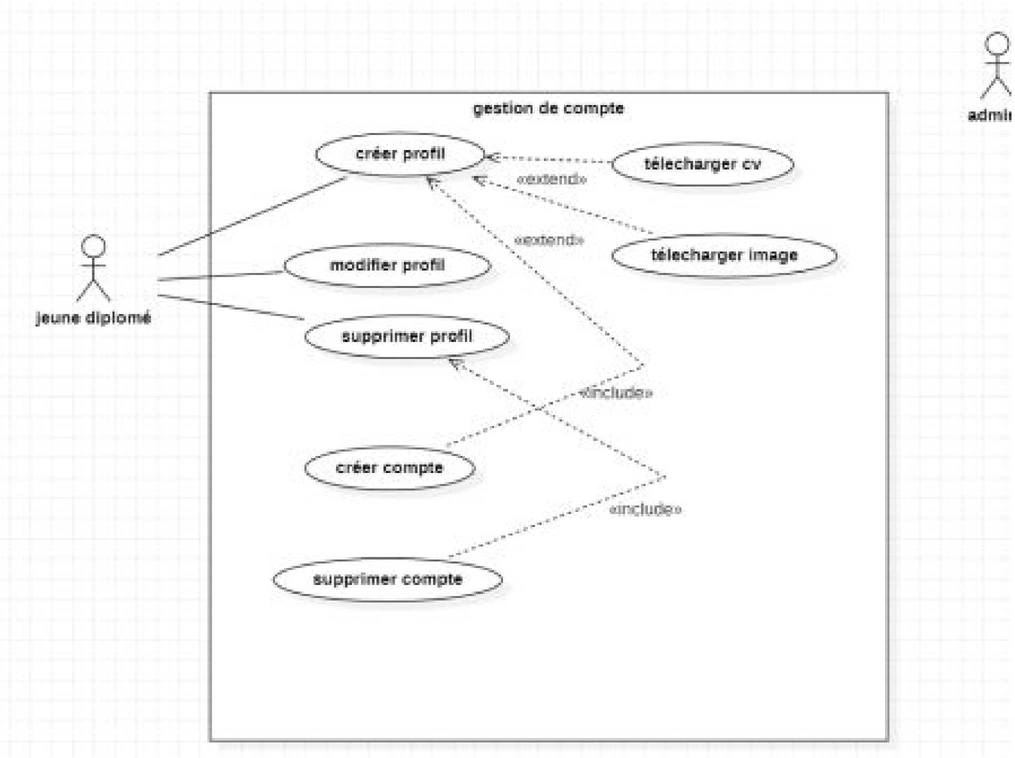
The figure shows the diagram relating to the use case: **recruiting website**. By choosing to register, the candidate fills out a form.



## Use case 2 : account management

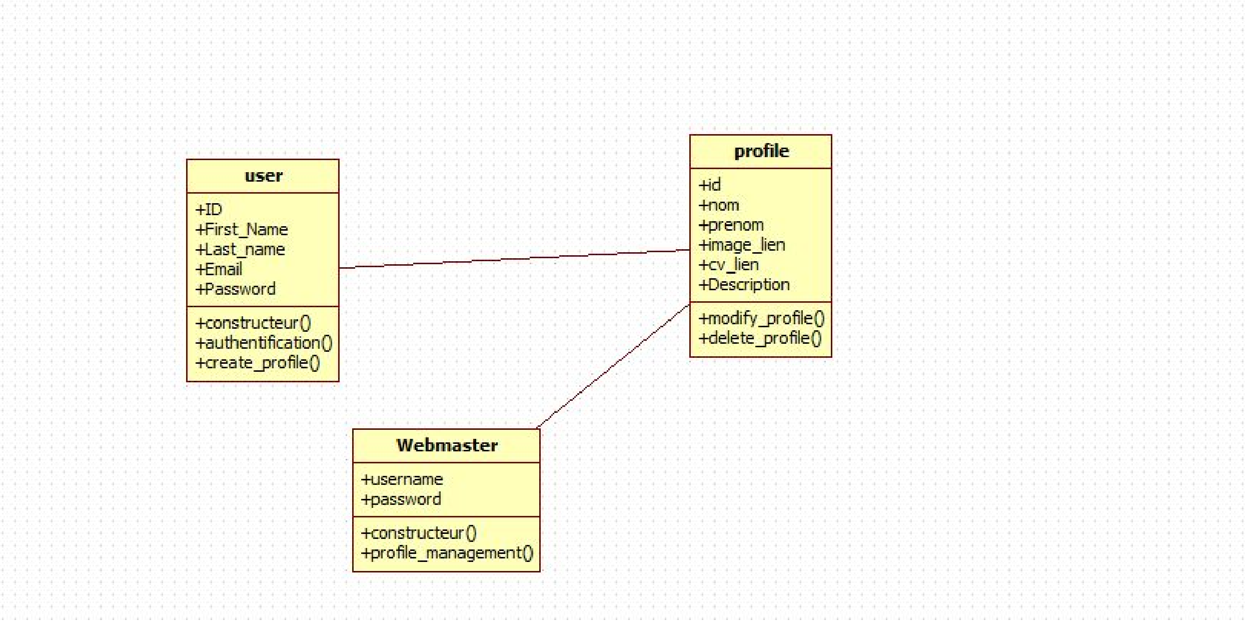
The young graduate uploads his CV and his picture when creating a profile on the website. he  can modify their informations , as he can delete his profile when he don’t need it yet.

Suppression of the profile implies the deletion of the account.



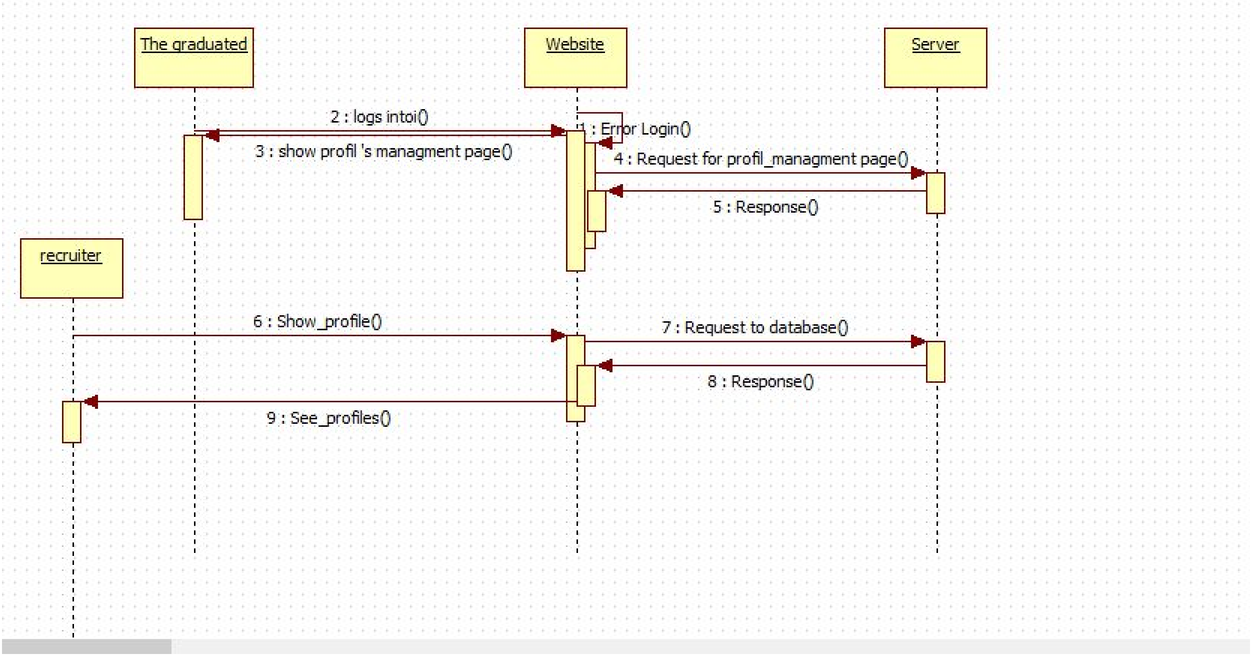
# Class diagram:

In software engineering, a class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.



# Sequence diagram:

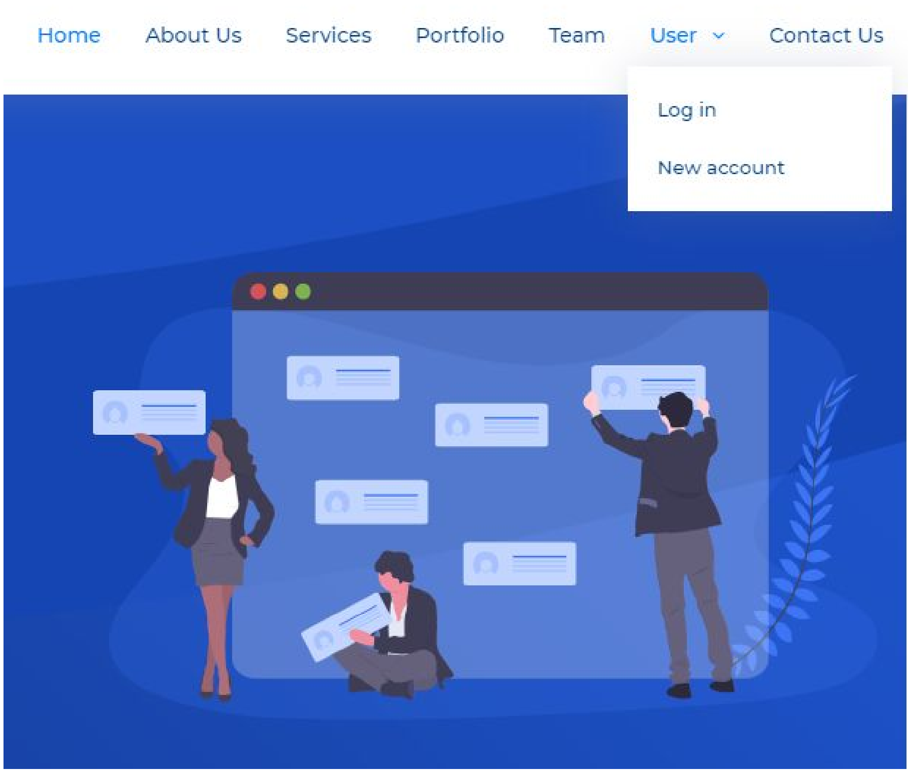
A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.



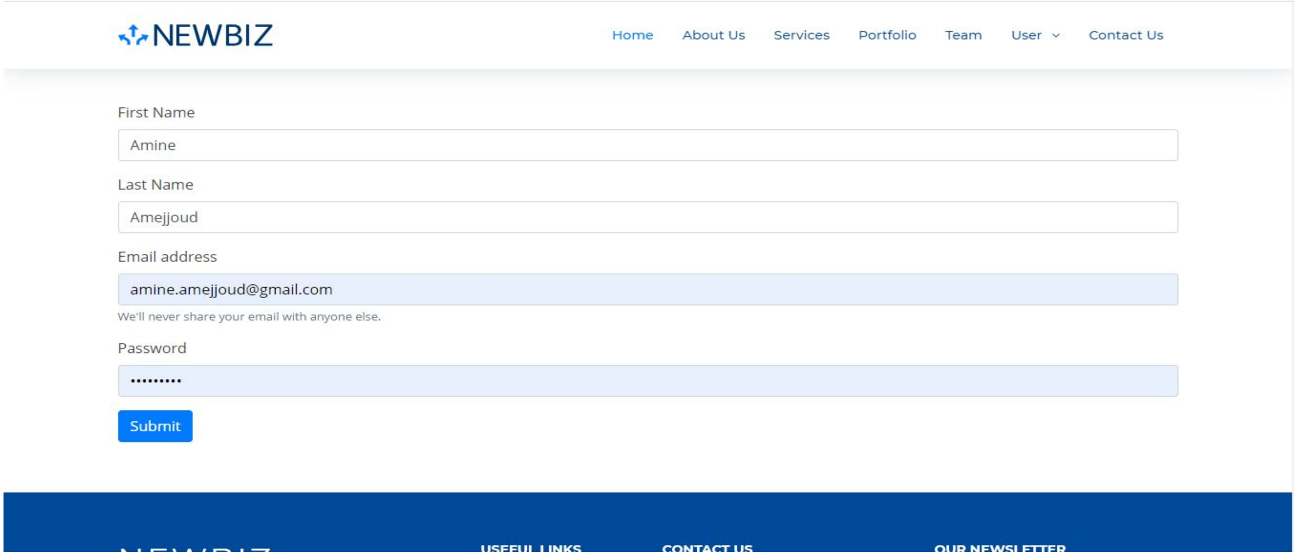
# Testing Our Application:

## Home page Module:

- User clicks in **New account** button , to create his own profile:



## Sign Up Module:



First Name
Amine

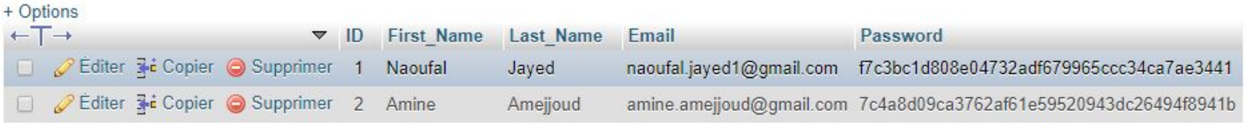
Last Name
Amejjoud

Email address
amine.amejjoud@gmail.com

We'll never share your email with anyone else.

Password
•••••••••

- the graduated young 's informations are saved in database.



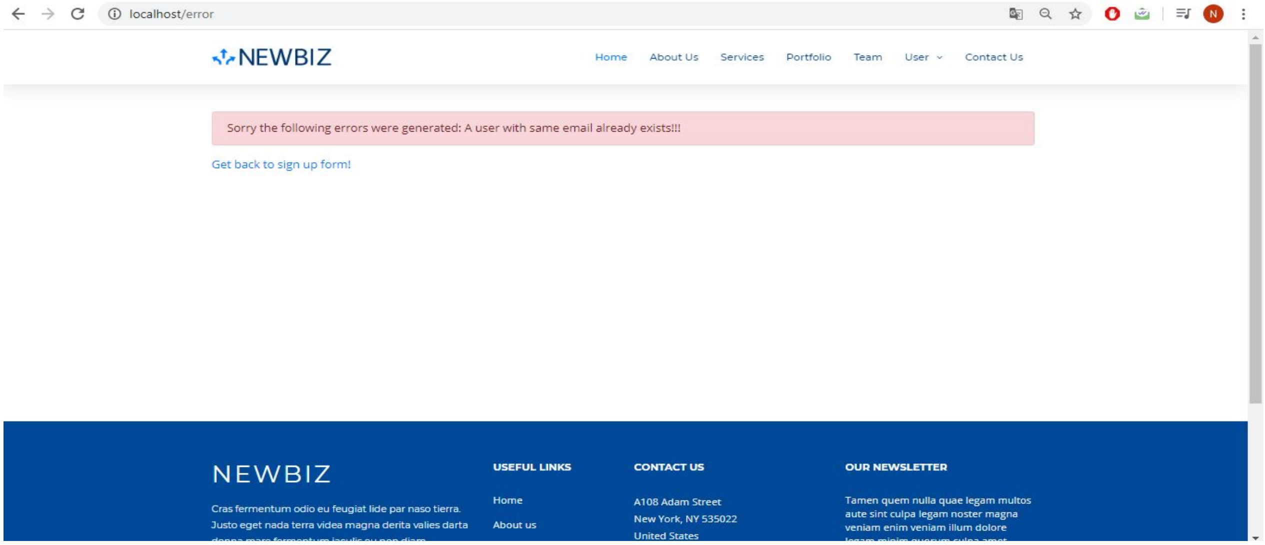
+ Options

| | ID | First_Name | Last_Name | Email | Password |
|---|---|---|---|---|---|
| □ Éditer Copier Supprimer | 1 | Naoufal | Jayed | naoufal.jayed1@gmail.com | f7c3bc1d808e04732adf679965ccc34ca7ae3441 |
| □ Éditer Copier Supprimer | 2 | Amine | Amejjoud | amine.amejjoud@gmail.com | 7c4a8d09ca3762af61e59520943dc26494f8941b |

- register 's error show in 2 case :

  -informations already exists in database

  -email not exists



Sorry the following errors were generated: A user with same email already exists!!!

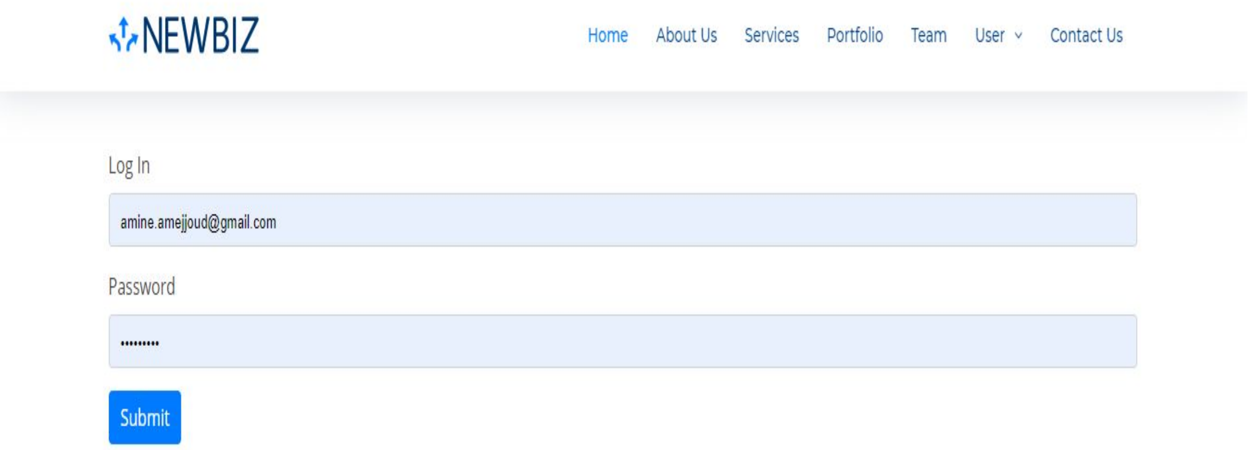
Get back to sign up form!

Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra videa magna derita valies darta

A108 Adam Street
New York, NY 535022
United States

Tamen quem nulla quae legam multos aute sint culpa legam noster magna veniam enim veniam illum dolore

## login 's Module:



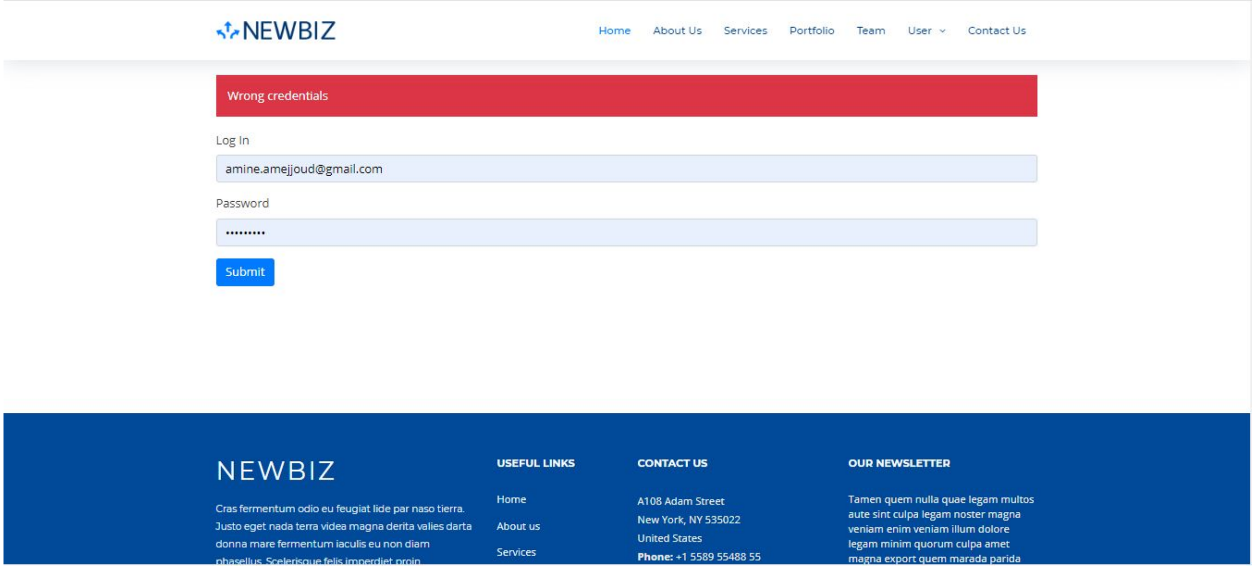
Log In

amine.amejjoud@gmail.com

Password

•••••••••

- login ‘s error :



Wrong credentials

Log In

amine.amejjoud@gmail.com

Password

•••••••••

NEWBIZ

Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra videa magna derita valies darta donna mare fermentum iaculis eu non diam phasellus. Scelerisque felis imperdiet proin

USEFUL LINKS

Home

About us

Services

CONTACT US

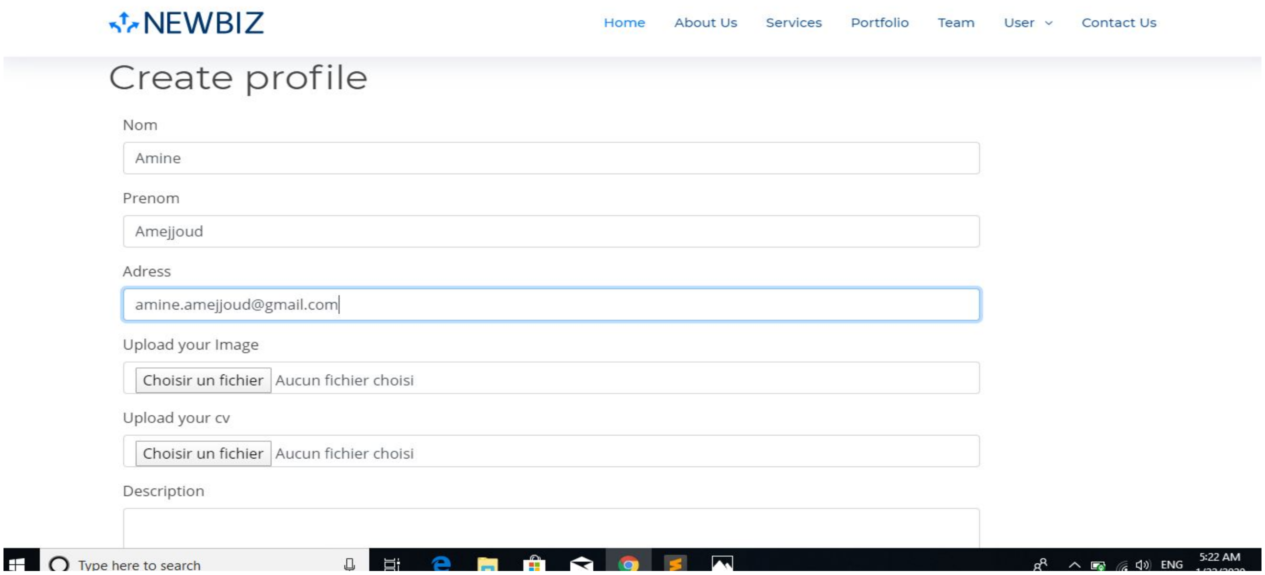
A108 Adam Street
New York, NY 535022
United States
**Phone:** +1 5589 55488 55

OUR NEWSLETTER

Tamen quem nulla quae legam multos aute sint culpa legam noster magna veniam enim veniam illum dolore legam minim quorum culpa amet magna export quem marada parida

## Profile ‘s Module:

### Create profile:



Create profile

Nom

Amine

Prenom

Amejjoud

Adress

amine.amejjoud@gmail.com

Upload your Image

Aucun fichier choisi

Upload your cv

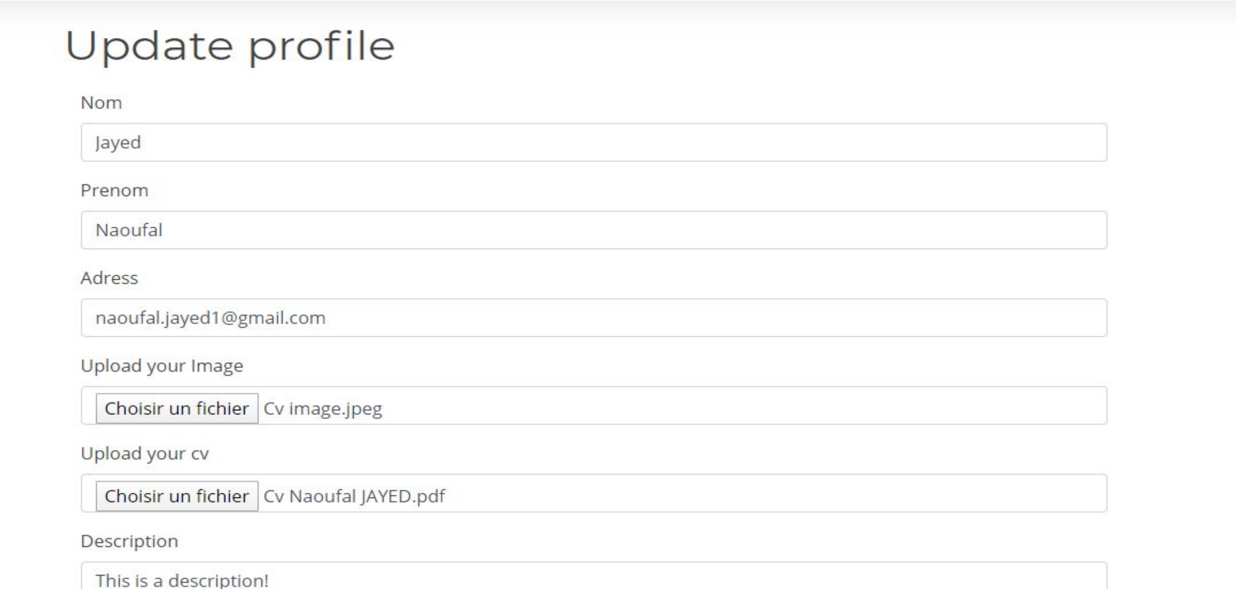
Aucun fichier choisi

Description

Users enter the following information:

- Full Name , address
- face picture , resume
- description of their experiences

## Update profile:



Update profile

Nom

Jayed

Prenom

Naoufal

Adress

naoufal.jayed1@gmail.com

Upload your Image

Choisir un fichier Cv image.jpeg

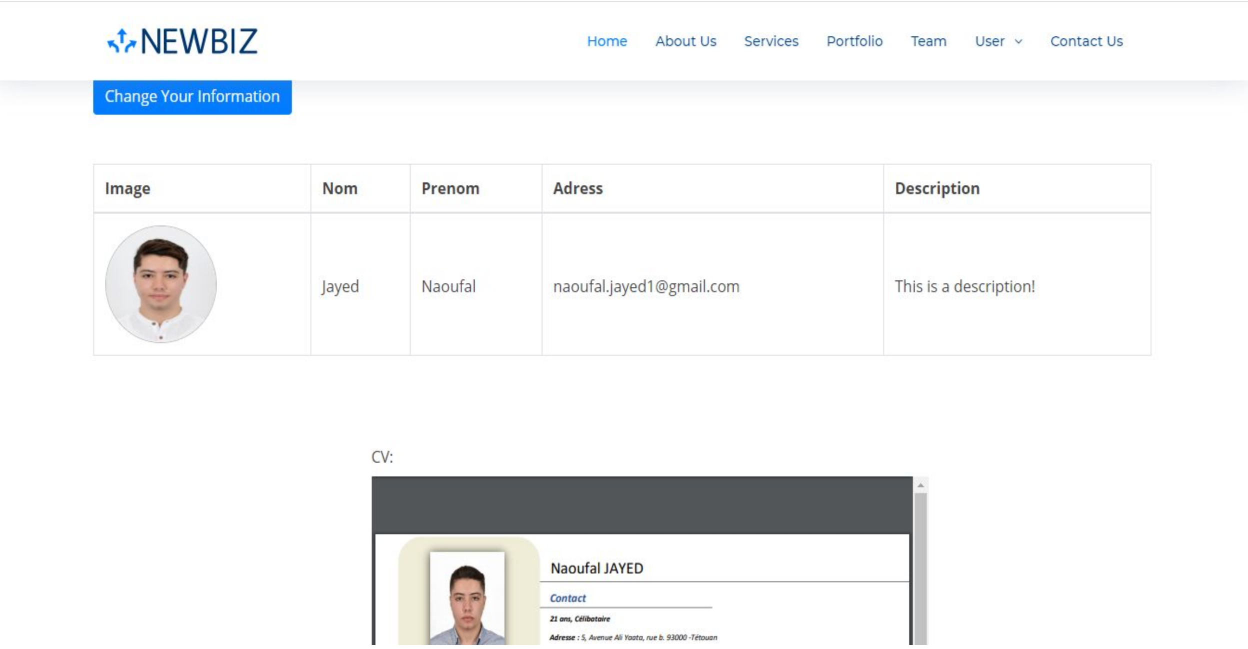
Upload your cv

Choisir un fichier Cv Naoufal JAYED.pdf

Description

This is a description!

## Read profile:



| Image | Nom | Prenom | Adress | Description |
|---|---|---|---|---|
|  | Jayed | Naoufal | naoufal.jayed1@gmail.com | This is a description! |

CV:

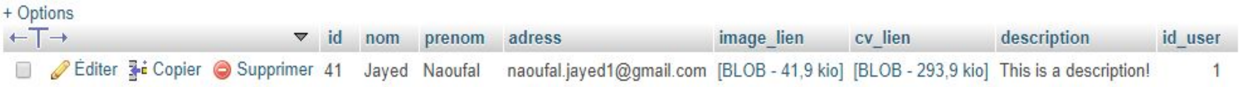
Naoufal JAYED

Contact

21 ans, Célibataire

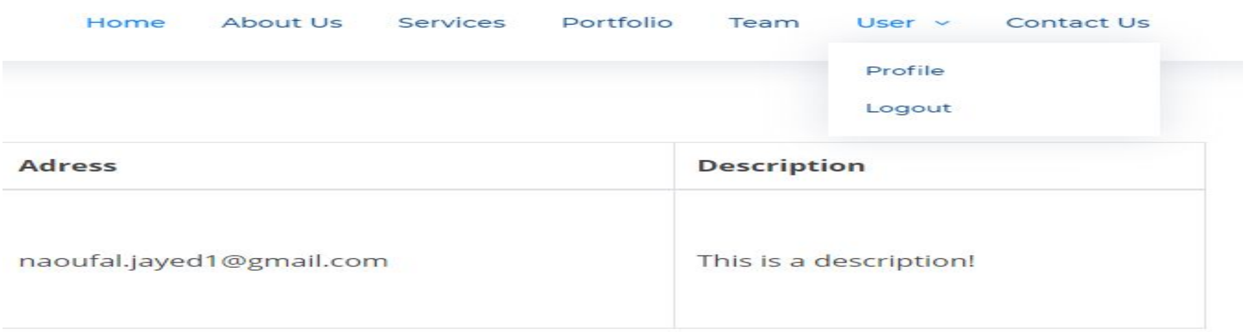
Adresse : 5, Avenue Ali Yaata, rue b. 93000 -Tétouan

## profile 's database:



|  | id | nom | prenom | adress | image_lien | cv_lien | description | id_user |
|---|---|---|---|---|---|---|---|---|
| □ Éditer Copier Supprimer | 41 | Jayed | Naoufal | naoufal.jayed1@gmail.com | [BLOB - 41,9 kio] | [BLOB - 293,9 kio] | This is a description! | 1 |

## Logout Module:



| Adress | Description |
|---|---|
| naoufal.jayed1@gmail.com | This is a description! |

Naoufal JAYED

Contact

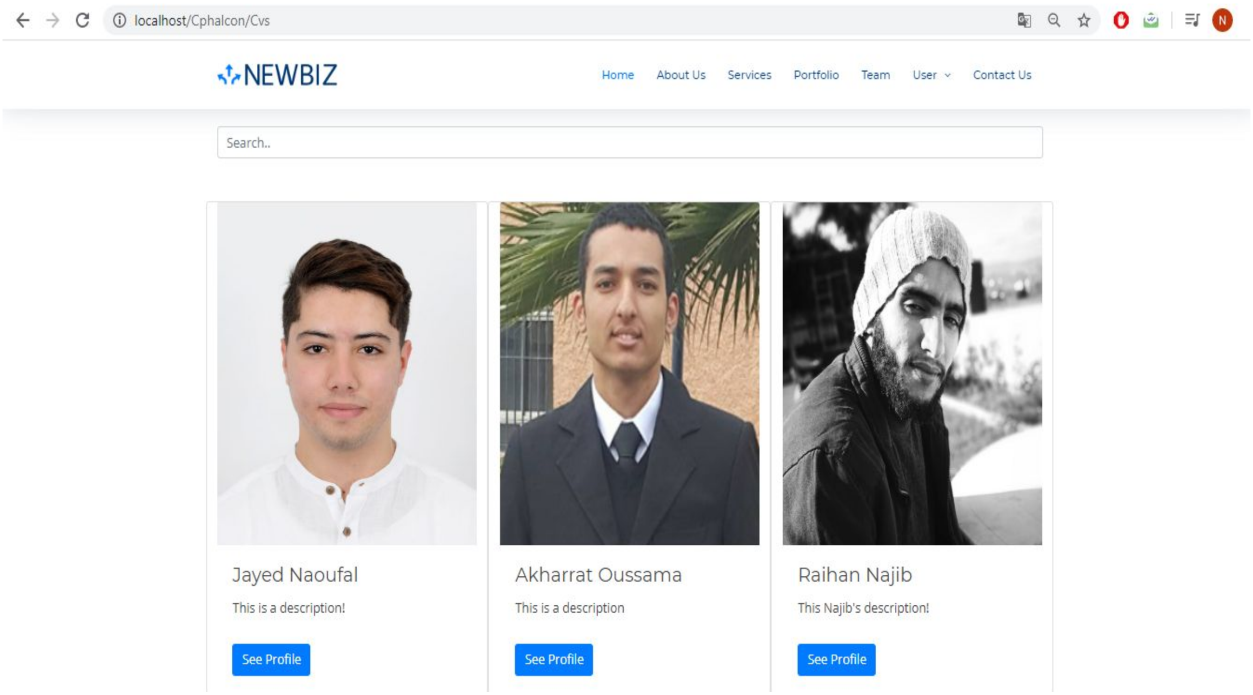
21 ans, Célibataire

Adresse : 5, Avenue Ali Yaata, rue b. 93000 -Tétouan

## Note:

Session 's user will destroy , when the user logout.

## Recruiter page 's  Module



Jayed Naoufal

This is a description!

Akharrat Oussama

This is a description

Raihan Najib

This Najib's description!

## Conclusion:

The purpose and objective of our Application  is achieved. By providing extremely rich graphical user interface, web page designing is easy and in an aesthetic form. Flexibility in designing makes users satisfied, even a novice user can dream and accomplish their wish of get a work.

## References :

- https://docs.phalcon.io/4.0/en/introduction
- https://www.hostinger.com/tutorials/phalcon-tutorial
- https://www.javatpoint.com/phalcon-tutorial
- https://github.com/phalcon
- https://en.wikipedia.org/wiki/Phalcon_(framework)