



**Project Report**  
of  
**MVC Programming**  
On  
**Job Portal**

**Submitted by:**

**Section:KM033**

Team Members

Tarun Kumar(B65)

Aniket Sriwastva(B66)

Sahil (B67)

**School of Computer Science Engineering**  
**Lovely Professional University, Phagwara**

## **Job Portal**

### **Introduction:**

This project is aimed at developing an online search Portal for job seekers. The system is an online application that can be accessed throughout the organization and outside as well with proper login provided. This system can be used as an Online Job Portal for job seekers. Job Seekers logging should be able to upload their information in the form of a CV. Visitors/Company representatives logging in may also access/search any information put up by Job aspirants.

### **Aim:**

This system can be used as an Online Job Portal for the Placements providing to the unemployed who are seeking for a job placement. Job Seeker logging into the system and he can should be able to upload their information in the form of a CV. Visitors/Company representatives logging in may also access/search any information put up by Job Seeker.

**Github Link:** <https://github.com/tarunk99/laravel>

## **Laravel Frame Work:**

Laravel is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model view controller (MVC) architectural pattern and based on Symphony. Its latest version is Laravel 5.8, released on February 26, 2019. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in application deployment and maintenance, and its orientation toward syntactic sugar.

The whole website can have three types of users:

1. Admin: Admin can manage whole website and can make changes over whole website.
2. Company: Company or also known as job provider owners can create a profile for their company and post job requirements.
3. User: User or also called job seeker. One who needs job and can see all the job posts posted by companies and can apply for any desired job by filling up a registration form.

## **Admin features:**

Admin can manage whole website:

Manage complete jobseeker section. Like: activate/deactivate/delete/ edit jobseeker's information.

Admin user can view the jobseeker's applications for each job.

Manage complete employer section. Admin user can activate/deactivate/delete/ edit company information.

Manage posted jobs. Like: activate/deactivate/delete/edit posted job.

Manage whole website content. Dynamic CMS is included to manage the content of the website.

## **Job provider / company features:**

After registration job provider can perform following action:

Add / Edit company's profile.

Post new job vacancies.

Edit / Deactivate posted jobs.

Job provider can see the list of jobseekers who has applied for the job.

Job provider can search jobseekers.

Job provider can send message to any job seeker.

## **Job seeker features:**

After registration job seeker can perform following actions:

Search for jobs

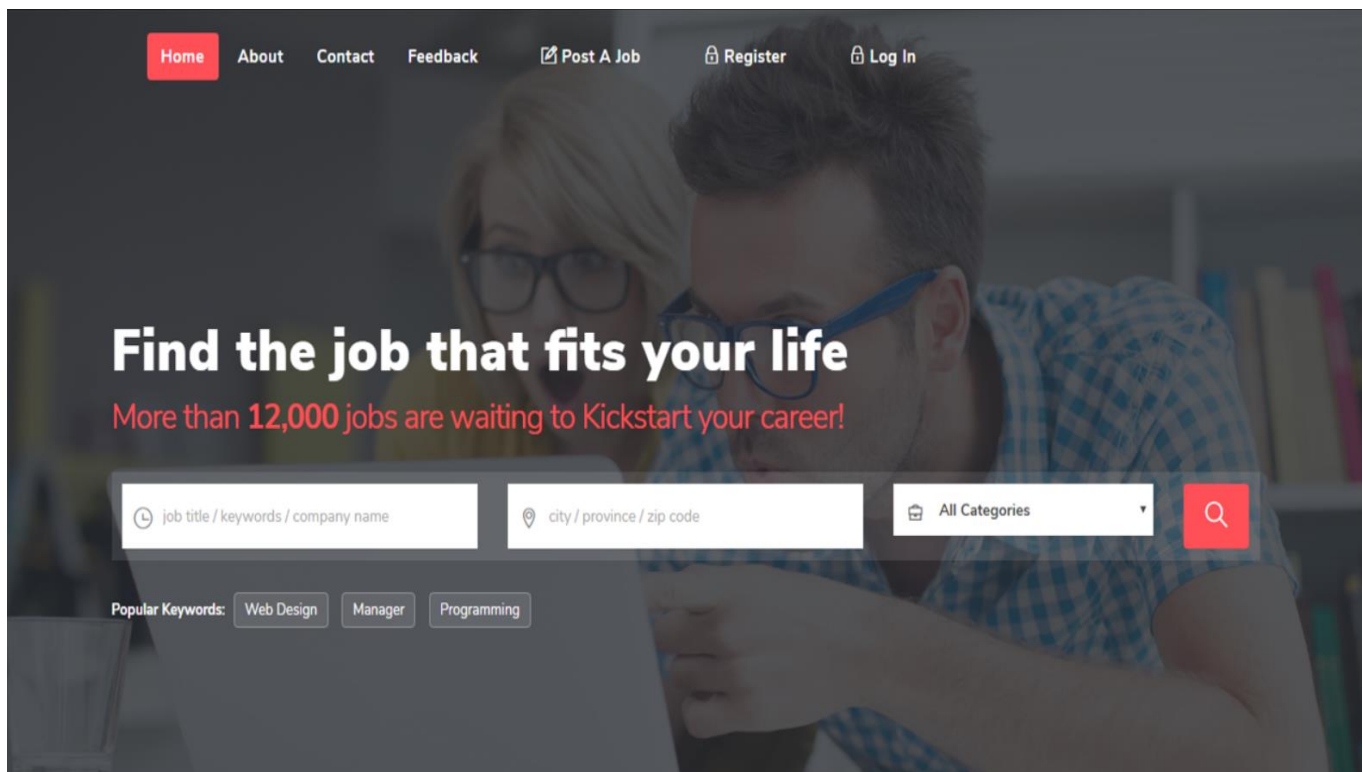
Apply Online for desire job

Add/Edit profile information including qualification, experience, and skills.

The wider areas of job searching facilitate the quick and easy access to opportunities. The increasing job opportunities and changing scenario of the business environment today has made more people to search for better career and employers to search for better potential.

## Home Page:

This is the home page of our website. Where you need to first login to access all features of website.



Filter option is also provided to users where one can set preferences according to his/her needs and can find suitable job easily. In filter option location is also provided. User can set location of job by his own.

## Sign-up Page:

Register For Job

Name

E-Mail Address

Password

Confirm Password

Mobile

Age

Location

Resume  No file chosen

Skills

Course

School/College

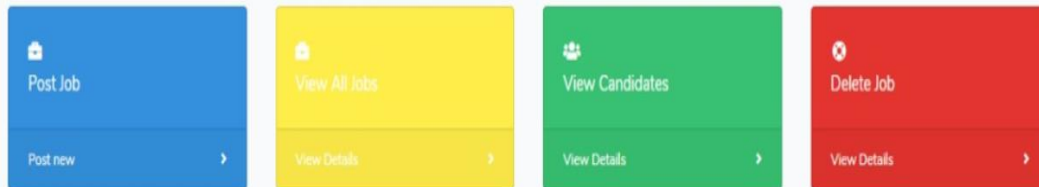
Register as

This is the signup page of website. Where new user can register and get their account. Then account will be verified by admin after checking all the details.

## Company Dashboard:

Post a Job

### Company dashboard

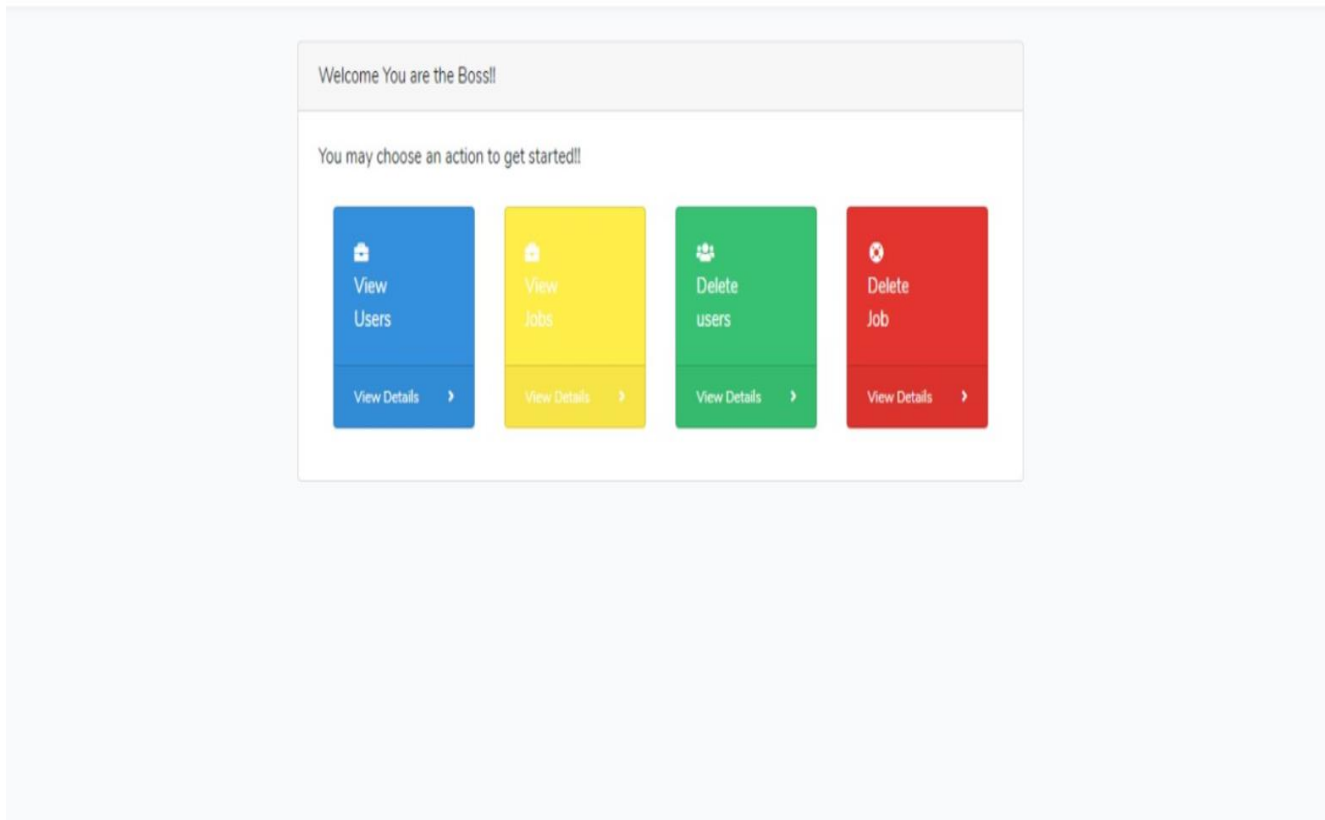


In company dashboard Admin can perform various operations such as Post job, view All jobs, View Candidates, Delete Job. Admin can delete existing jobs and also add new jobs. Admin can accept or reject the company's profile. Admin have right to delete any company's profile and also allow any company to create account on this website.

## Admin Dashboard:

Post a Job

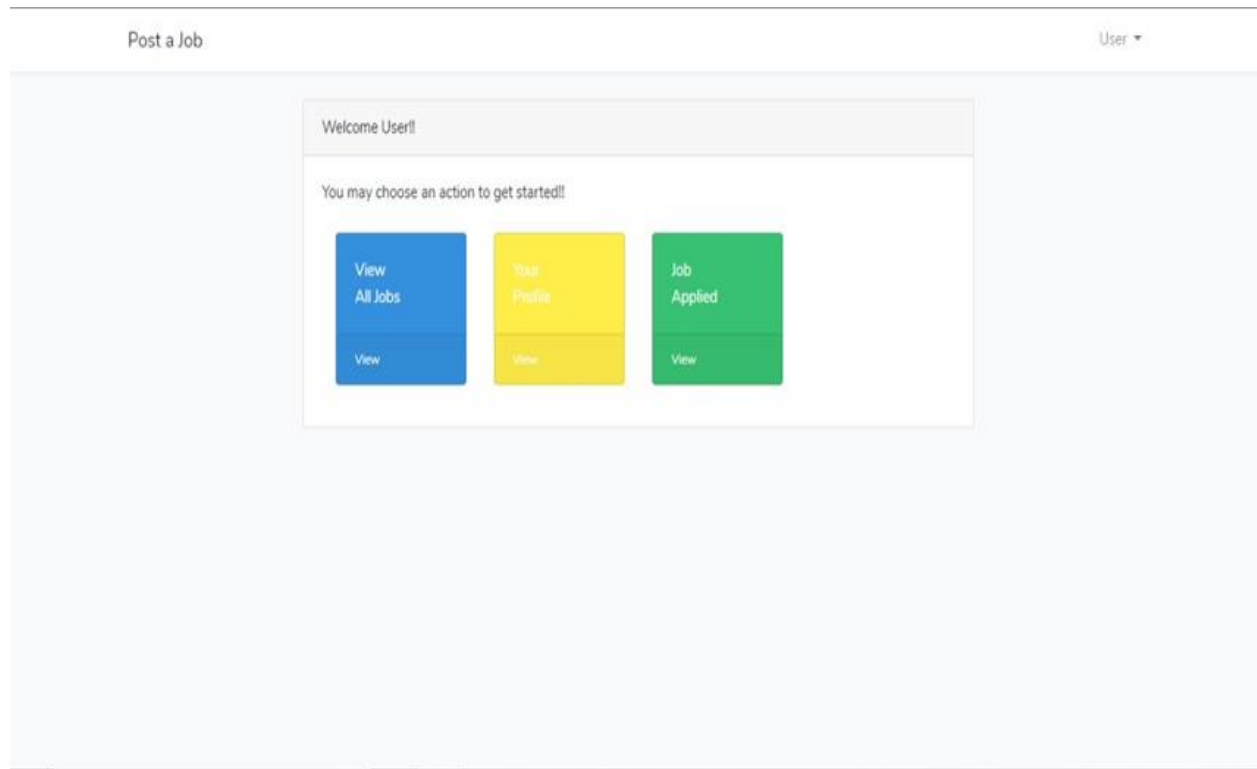
Post a Job



This is Admin's dashboard. After logging in by Admin's credentials message will be shown as You are the boss. Here Admin can view all the users/view all jobs/delete users/delete jobs.



## User's Dashboard:



This is User's Dashboard. Where one can view all available jobs that are posted by companies. Can manage his/her profile and also can see all the jobs that he/she applied for.

## Migration Code:

### Code for creating user table:

```
<?php

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateUsersTable extends Migration
{
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->string('password', 60);
            $table->dateTime('activated_at')->nullable();
            $table->enum('role', ['jobseeker', 'employer', 'admin'])->default('jobseeker');

            $table->rememberToken();
            $table->timestamps();
        });
    }

    public function down()
```

```
{  
    Schema::drop('users');  
}  
}
```

### **Code for creating Company table:**

```
<?php  
  
use Illuminate\Database\Schema\Blueprint;  
use Illuminate\Database\Migrations\Migration;  
class CreateCompaniesTable extends Migration  
{  
    public function up()  
    {  
        Schema::create('companies', function (Blueprint $table) {  
            $table->increments('id');  
            $table->string('nit')->unique();  
            $table->string('name')->unique();  
            $table->string('email');  
            $table->string('description')->nullable();  
            $table->string('website')->nullable();  
            $table->string('facebook')->nullable();  
            $table->string('twitter')->nullable();  
            $table->string('cel')->nullable();  
        });  
    }  
}
```

```
$table->string('tel')->nullable();

$table->boolean('active')->default(false);

$table->boolean('email_new_job')->default(true);

$table->boolean('show_data')->default(true);

$table->timestamps();

$table->integer('user_id')->unsigned();

$table->foreign('user_id')->references('id')->on('users');

$table->string('geo_location_id')->nullable();

$table->foreign('geo_location_id')->references('id')->on('geo_locations');

$table->integer('company_category_id')->unsigned()->nullable();

$table->foreign('company_category_id')->references('id')->on('company_categories');

});

}

public function down()

{

    Schema::drop('companies');

}

}
```

## Code for creating Application table:

```
<?php
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateApplicationsTable extends Migration
{
    public function up()
    {
        Schema::create('applications', function (Blueprint $table) {
            $table->increments('id');
            $table->string('intro');
            $table->boolean('preselected')->default(false);
            $table->boolean('accepted')->default(false);
            $table->timestamps();
            $table->integer('resume_id')->unsigned();
            $table->foreign('resume_id')->references('id')->on('resumes');
            $table->integer('job_id')->unsigned();
            $table->foreign('job_id')->references('id')->on('jobs');
        });
    }
    public function down()
    {
        Schema::drop('applications');
```

```
}  
}
```

## Controller Code:

### Admin Controller:

```
<?php  
namespace App\Http\Controllers\Admin;  
use Illuminate\Http\Request;  
use App\Http\Requests;  
use App\Http\Controllers\Controller;  
class AdminController extends Controller  
{  
    protected $userFacade;  
    public function __construct(UserFacade $userFacade)  
    {  
        $this->userFacade = $userFacade;  
    }  
    public function index()  
    {  
        return view('admin.home');  
    }  
  
    public function editAccount(User $user)
```

```
{  
    return view('account.form', [  
        'user' => $user,  
        'formData' => ['route' => ['users.update', $user], 'method' => 'POST', 'files'  
=> true]  
    ]);  
}  
  
public function postEditAccount(UpdateRequest $request, User $user)  
{  
    $this->userFacade->updateUser($request->all(), $user);  
    if($user->isJobseeker()) {  
        return redirect()->route('admin.registers');  
    }  
    return redirect()->route('admin.companies.index');  
}  
}
```

## Companies Controller:

```
<?php

namespace App\Http\Controllers\Admin;

use App\Entities\Company;
use App\Entities\Job;
use App\Http\Requests;

class CompaniesController extends ResourceController
{
    protected $routePrefix = 'companies';

    protected $viewPath = 'portal.companies';

    protected $facade;

    protected $emailService;

    function __construct(CompanyService $service, EmployerFacade $facade,
        EmailService $emailService)
    {
        $this->service = $service;

        $this->emailService = $emailService;

        $this->facade = $facade;
    }

    public function index()
    {
        return $this->view('lists', []);
    }
}
```



```
public function active(Company $company)
{
    $user = $company->user;
    if($company->is_active) {
        $company->active = 0;
        $user->activated_at = null;
    }
    else {
        $company->active = 1;
        $user->activated_at = Carbon::now()->toDateTimeString();
        $this->emailService->notifyActiveUser($user);
    }
    $company->save();
    $user->save();
    return ['success' => true];
}

public function destroy(Company $company)
{
    return $this->service->deleteModel($company);
}

}
```

## Register Controller:

```
<?php

namespace App\Http\Controllers\Validations;

use App\Http\Controllers\Controller;
use App\Http\Requests;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;

class RegisterController extends Controller
{
    protected $rules = [

        'name'          => 'required|max:255',
        'email'          => 'required|email|max:255|unique:users',
        'role'           => 'required|in:jobseeker,employer',
        'password'       => 'required|confirmed|min:6',
        'company'        => 'required_if:role,employer|unique:companies,name',
        'nit'            => 'required_if:role,employer|unique:companies',
        'terms-employer' => 'required_if:role,employer',
        'terms-jobseeker' => 'required_if:role,jobseeker'
    ];

    public function validation(Request $request)
    {

```

```
if($request->get('validate') && array_key_exists($request->get('validate'),
$this->rules)) {

    $validator = Validator::make( $request->all(), [

        $request->get('validate') => $this->rules[$request->get('validate')]

    ]);

    if($validator->passes()){

        return 'true';

    }

}

return 'false';

}

}
```

## Routes:

```
Route::get('show-post-job/{id}', 'JobsController@showPostJob')->
middleware('authentication');
```

```
Route::get('save-job', 'JobsController@store')->middleware('authentication');
```

```
Route::get('remove-job/{id}', 'JobsController@destroy')->
middleware('authentication');
```