The Islamic University of Gaza
Faculty of Information Technology
Department of Computer Science &
Software Development

الجامعـــــــة الإســـلاميــة بغـزة
كلية تكنولوجيا المعلومات
قسم علم الحاسوب وتطوير البرمجيات

Web-based Application for jobs and training in Gaza Strip

“Gaza Jobs”

موقع الكتروني للوظائف والتدريب في قطاع غزة

"وظائف غزة"

By

Abdullah Shublaq	**120170417**
Mohammed Al-Kahlout	**120171909**
Mohammed Abuassi	**120171839**
Hashem Al-Saqqa	**120171248**

Supervised by

Prof. Alaa El-Halees

A graduation project report submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Technology

June/2021

Abstract

Finding jobs that best suits the interests and skill set is quite a challenging task for job seekers. The difficulties arise from not having enough experience, their work culture and current job openings. In addition, finding the right candidate with desired qualifications to fill their current job openings is an important task for employers. Online Job platforms have certainly made job seeking convenient on both sides.

Job's portal is the solution where the employers, as well as the job seekers, meet aiming at fulfilling their individual requirement. These websites are the cheapest as well as the fastest source of communication reaching a wide range of audience in just a single click.

This project aims to develop a platform for finding jobs and training in the Gaza Strip, named "Gaza Jobs". We used the Agile Methodology to develop this project. The main result of this project is to help students, graduates or people to connect with local companies and find jobs or training that they want. In this project, it's easy to find a job that suits your experience by the advanced search. In applying for a job, you can send any questions you have before sending the application for this job, you can apply for the job as a team so the job could be a whole project done by one team without the need to find another person for the other tasks in the project. Any company can post a training to help the beginners to develop their skills. The difference in our project that any user can join teams or create one to help each other and exchange experiences, they have a team leader that will manage all the team actions and works. Job's portal is categorized as on-demand systems and so that it is very important and useful for the students, graduates or people.

ملخص الدراسة

يعد العثور على الوظائف التي تناسب الاهتمامات والمهارات مهمة صعبة للغاية بالنسبة للباحثين عن عمل. تنشأ الصعوبات من عدم وجود خبرة كافية وثقافة العمل وفرص العمل الحالية. بالإضافة إلى ذلك، فإن العثور على المرشح المناسب بالمؤهلات المطلوبة لملء فرص العمل الحالية يعد مهمة هامة لأصحاب العمل. من المؤكد أن منصات العمل عبر الإنترنت جعلت البحث عن عمل مناسبًا لكلا الجانبين.

بوابة الوظائف هي الحل الذي يلتقي فيه أصحاب العمل والباحثون عن عمل بهدف تلبية متطلباتهم الفردية. هذه المواقع هي أرخص وأسرع مصدر للاتصال تصل إلى مجموعة واسعة من الجمهور بنقرة واحدة فقط.

يهدف هذا المشروع إلى تطوير منصة لإيجاد وظائف وتدريب في قطاع غزة تحت اسم "وظائف غزة". استخدمنا منهجية Agile لتطوير هذا المشروع. تتمثل النتيجة الرئيسية لهذا المشروع في مساعدة الطلاب أو الخريجين أو الأشخاص على التواصل مع الشركات المحلية والعثور على الوظائف أو التدريب الذي يريدونه. في هذا المشروع، من السهل العثور على وظيفة تناسب تجربتك من خلال البحث المتقدم. عند التقدم لوظيفة، يمكنك إرسال أي أسئلة لديك قبل إرسال الطلب لهذه الوظيفة، يمكنك التقدم للوظيفة كفريق حتى يمكن أن تكون الوظيفة مشروعًا كاملاً قام به فريق دون الحاجة إلى البحث عن شخص آخر من أجل عمل المهام الأخرى في المشروع. يمكن لأي شركة نشر تدريب لمساعدة المبتدئين على تطوير مهاراتهم. الفرق في مشروعنا هو أنه يمكن لأي مستخدم الانضمام إلى فرق أو إنشاء فريق لمساعدة بعضهم البعض وتبادل الخبرات، لديهم قائد فريق يدير جميع أعمال الفريق وأعماله. يتم تصنيف بوابة الوظائف على أنها أنظمة حسب الطلب ولذا فهي مهمة جدًا ومفيدة للطلاب أو الخريجين أو الأشخاص.

Epigraph Page

“When you think about the last four years, you’ll probably remember that your greatest lessons came from outside the classroom. It’s a good reminder that learning doesn’t stop just because you received a diploma.”

Dedication

As students of the Islamic University of Gaza (IUG), we dedicate this project and application to our families and friends, and everyone who lives in Gaza Strip and cares about it. We dedicate this work and give our special thanks to our supervisor and the academic staff of the Faculty of Information Technology-IUG.

Acknowledgment

First of all, we thank Allah for giving us the strength and ability to complete this project, despite all the difficulties and obstacles we’ve faced. we also thank our precious families for their infinite support and encourage. we also thank all my friends who stands by me all the time.

We also like to show our huge gratitude and grateful to our supervisor: Prof. Alaa El-Halees for sharing his guidance and experience with us during the work on this project. Without his supervision and constant help, this project would not have been possible.

Table of Contents

Tables and Figures related to chapter (2, 3, 5, 6, 7, 8)

List of Tables

List of Figures

Chapter 1

Introduction

Chapter 1
Introduction

1.1 Background and Context

These days, finding a job or training has become a basic requirement for each student, graduate, or people, and for that, some spend long days in pursuit between companies and looking for work or training, and the most important reasons that lead to this is Little networking and Lack of communication skills [1], So many countries and governments have adopted the idea of creating websites specialized in searching for jobs that make it easier for students, graduates, or people to find a job or training, and also reduce the effort they spend for that.

Online platforms support so many of the daily activities that we have become dependent on them in our personal and professional lives.

Because there are no specialized platforms for that in the Gaza Strip, we made a platform that helps to find and apply to jobs or training from governmental or private institutions or even remotely by searching for it and apply for it using their CV and also provide a great feature that enables students. graduates or people to create their teams and apply to jobs that want a team.

Gaza Jobs connects students, graduates, and people to jobs or training that need their skills in an easier and faster way than the traditional way.

1.2 Statement of the problem

With the increase in the number of students and graduates annually, it became difficult for them to reach and obtain a training or work opportunity, That leads to an increase in the level of unemployment, and the most important reasons that lead to this is Little networking and Lack of communication skills, Therefore we want to work out a solution to this problem by creating a platform that connects students and graduates with companies to enhance the chances of obtaining work or training.
Among the other problems that we faced was the problem of people communicating with companies and the difficulty of finding companies specialized in their field of interest.
Also, one of the features that are missed from other websites is able the users to creating teams, that have the ability to apply to the jobs that require many specialists in many fields.

1.3 Importance of the project

Our website is considered an on-demand application and its importance come from the main goal which aims to facilitation the communication between the job seekers and the companies and facilitating the team-building process that increases the chance of getting a job for this team, Which also all leads to a reduction in the unemployment rate.

The importance of our project come from the ease of communication between the companies and job seekers and increase the probability to find jobs for them by making a function that allows the job-seekers to create a team with many specialists in many fields that can help them to apply to the jobs that need many specialists in many fields, and the other importance of our project is to facilitation for the users looking for training by showing to them the companies that need to trainees.

1.4 Scope and Objectives

1.4.1 Scope

- The developed system will be web-based application only.
- The developed system will support Arabic Language only.
- The system will allow different types of privileges for different roles.

1.4.2 Objectives

1.4.2.1 Main Objectives

The main objective is to develop a web-based platform for users to facilitation finding jobs or training, and the companies can post their jobs or training, to make it easy for users to find this, also provide a new feature to help users make their team or find a team needs your Experiences.

1.4.2.2 Specific Objectives

- Collect and analyze the system’s requirements.
- Designing UX/UI for the system.
- Developing the Front-end for the system.
- Developing the Back-end for the system.
- Testing the final system.
- Launching the system.

1.5 Signification

- Make it easier to search for jobs and training for the client.
- Make it easier for the job seekers to make a team with each other.
- Make it easier for an employer to post jobs or training.

1.6 Limitations

- The system supports just the users from Gaza Strip.
- The first version of the system does not support make a contract.
- The first version of the system does not support online payment.

Chapter 2

Related Works

Chapter 2
Related Works

After Search on related websites, we found these websites that close to our project like jobs.ps [2], Indeed.com [3] and Linkedin.com [4]:

2.1 Jobs.ps:

Jobs.ps Ltd, located in Ramallah, is the only official, trusted and leading Job portal in Palestine. Since its beginning ten years ago as Jobs.ps Ltd, the company has played a key role in providing service and building lasting relationships by placing the highest quality candidates throughout Palestine. Jobs.ps Ltd has around One million visits per month, more than 100,000 registered job-seekers, and around 40,000 posted jobs, and almost 3000 active employer accounts.

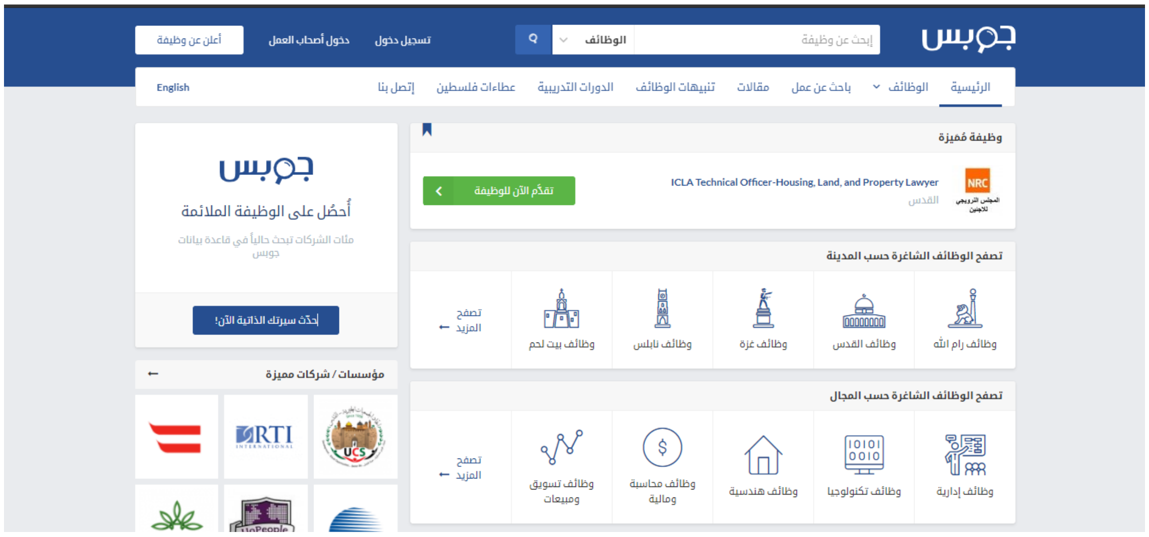
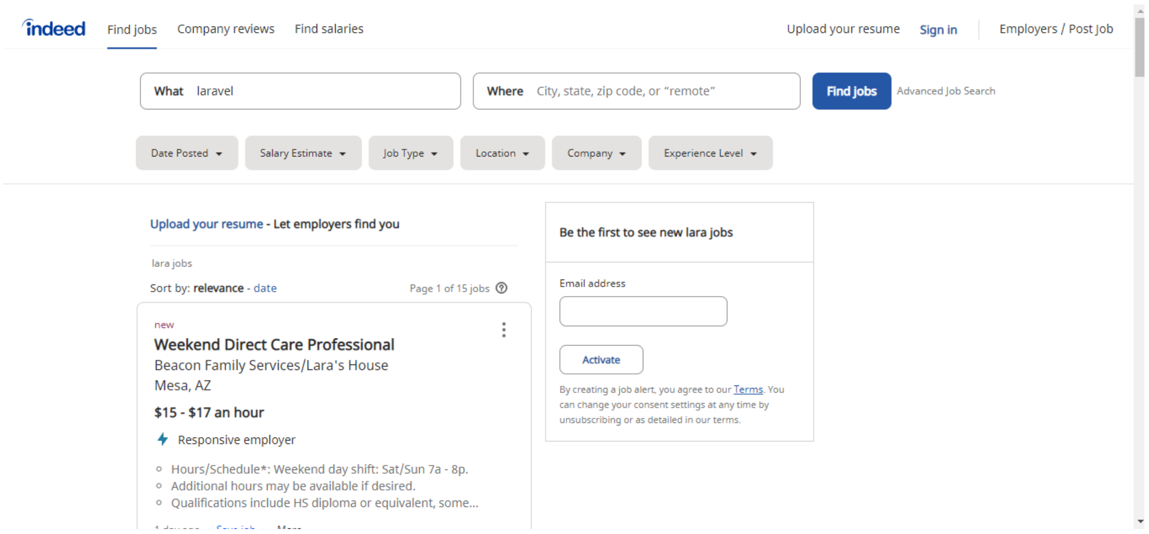


Figure 1 - Jobs.ps

2.2 Indeed.com:

Indeed is the #1 job site in the world1 with over 250 million unique visitors2 every month. Indeed strives to put job seekers first, giving them free access to search for jobs, post resumes, and research companies. Every day, we connect millions of people to new opportunities.



What laravel

Where City, state, zip code, or "remote"

Upload your resume - Let employers find you

lara jobs

Sort by: relevance - date

Page 1 of 15 jobs

new

Weekend Direct Care Professional

Beacon Family Services/Lara's House

Mesa, AZ

$15 - $17 an hour

Responsive employer

- Hours/Schedule*: Weekend day shift: Sat/Sun 7a - 8p.
- Additional hours may be available if desired.
- Qualifications include HS diploma or equivalent, some...

Be the first to see new lara jobs

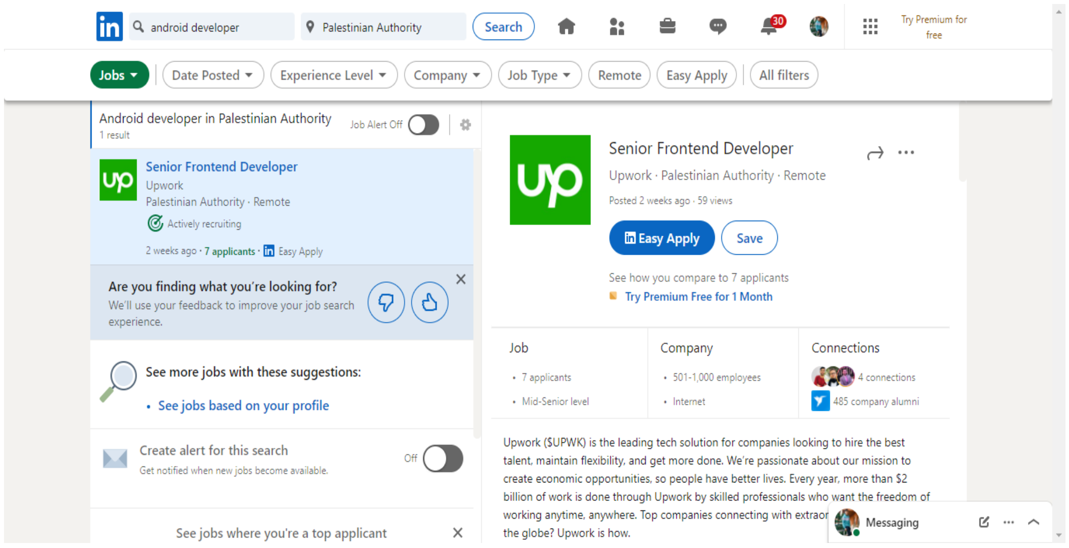
Email address

By creating a job alert, you agree to our Terms. You can change your consent settings at any time by unsubscribing or as detailed in our terms.

Figure 2 - Indeed.com

2.3 Linkedin.com:

LinkedIn began in co-founder Reid Hoffman's living room in 2002 and was officially launched on May 5, 2003.
Today, LinkedIn leads a diversified business with revenues from membership subscriptions, advertising sales and recruitment solutions under the leadership of Ryan Roslansky. In December 2016, Microsoft completed its acquisition of LinkedIn, bringing together the world’s leading professional cloud and the world’s leading professional network.


android developer
Palestinian Authority
Search
Try Premium for free

Jobs | Date Posted | Experience Level | Company | Job Type | Remote | Easy Apply | All filters

Android developer in Palestinian Authority
1 result
Job Alert Off

Senior Frontend Developer
Upwork
Palestinian Authority · Remote
Actively recruiting
2 weeks ago · 7 applicants · Easy Apply

Are you finding what you're looking for?
We'll use your feedback to improve your job search experience.

See more jobs with these suggestions:
• See jobs based on your profile

Create alert for this search
Get notified when new jobs become available.
Off

See jobs where you're a top applicant

Senior Frontend Developer
Upwork · Palestinian Authority · Remote
Posted 2 weeks ago · 59 views
Easy Apply | Save
See how you compare to 7 applicants
Try Premium Free for 1 Month

Job	Company	Connections
• 7 applicants	• 501-1,000 employees	4 connections
• Mid-Senior level	• Internet	485 company alumni

Upwork ($UPWK) is the leading tech solution for companies looking to hire the best talent, maintain flexibility, and get more done. We're passionate about our mission to create economic opportunities, so people have better lives. Every year, more than $2 billion of work is done through Upwork by skilled professionals who want the freedom of working anytime, anywhere. Top companies connecting with extraor the globe? Upwork is how.

Messaging

Figure 3 – LinkedIn

2.4 Related work summary:

After we reviewed the previous websites, we are focused on these features that are the same on our project:

1. **Search to the job by categories or specific filter:**

   you can find the job related to your selected category or by search using specific filter, and this feature are found in jobs.ps, indeed.com and linkedin.com.

2. **Make CV and portfolio:**

   So, the user can choose to make his CV manually by filling the form inputs or upload it, In the jobs.ps and linkedin.com you can just make it manually, and in indeed.com you can upload it or write it manually and in linkedin.com, but our system will allow to make it manually and he can upload it.

3. **Apply to jobs and training:**

   User can apply for a job or training with their CV and the company can see his CV and his ratings and can chat with him, in jobs.ps there a specific page for training but in indeed.com and linked.com, it shows on the jobs page.

4. **Send Inquiries:**

   The Job Seeker can send an inquire for a job and the employer can reply to it, and this feature found in indeed.com and linked.com but not found in jobs.ps.

And after that, we decide to make a feature that allows the user to make his team and apply to jobs using the team because this feature not found in any site and it will increase the probability to found and take the jobs.

Chapter 3
Methodology

Chapter 3
Methodology

3.1 Methodologies:

To develop this website, we will follow agile methodology, because the requirements of our system are not clear, so we need to get more information about the functionality of the system to design and develop in each iteration.

3.1.1 The Agile Process Flow

1. **Concept** - Projects are envisioned and prioritized
2. **Inception** - Team members are identified, funding is put in place, and initial environments and requirements are discussed
3. **Iteration/Construction** - The development team works to deliver working software based on iteration requirements and feedback
4. **Release** - QA (Quality Assurance) testing, internal and external training, documentation development, and final release of the iteration into production
5. **Production** - Ongoing support of the software
6. **Retirement** - End-of-life activities, including customer notification and migration

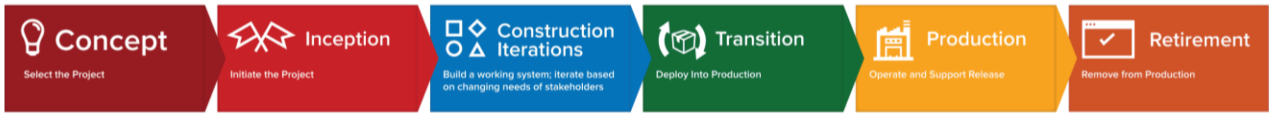


Figure 4 - The Agile Process Flow

This view presents the full Agile lifecycle model within the enterprise. In any enterprise there may be projects operating simultaneously, multiple sprints/iterations being logged on different product lines, and a variety of customers, both external and internal, with a range of business needs. [5]

3.1.2 The Agile Iteration Workflow

The Agile software development lifecycle is dominated by the iterative process. Each iteration results in the next piece of the software development puzzle - working software and supporting elements, such as documentation, available for use by customers - until the final product is complete. Each iteration is usually two to four weeks in length and has a fixed completion time. Due to its time-bound nature, the iteration process is methodical and the scope of each iteration is only as broad as the allotted time allows.

Multiple iterations will take place during the Agile software development lifecycle and each follows its own workflow. During an iteration, it is important that the customers and business stakeholders provide feedback to ensure that the features meet their needs.

A typical iteration process flow can be visualized as follows:

1. **Requirements** - Define the requirements for the iteration based on the product backlog, sprint backlog, customer and stakeholder feedback.
2. **Development** - Design and develop software based on defined requirements.
3. **Testing** - QA (Quality Assurance) testing, internal and external training, documentation development.
4. **Delivery** - Integrate and deliver the working iteration into production.
5. **Feedback** - Accept customer and stakeholder feedback and work it into the requirements of the next iteration.

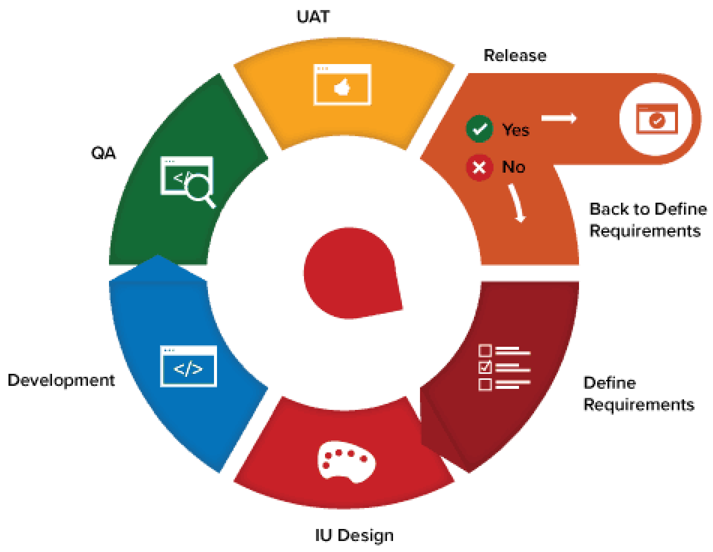


Figure 5 - The Agile Iteration Workflow

For the duration of the project, while additional features may be fed into the product backlog, the rest of the process is a matter of repeating the steps over and over until all of the items in the product backlog have been fulfilled. As a result, the process flow is more of a loop and not a linear process.

3.3 Time Table:

#	Tasks	Iterations	Duration	Start	Finish
Step 1	Design and implement the Front-end	Requirement Development Testing Delivery Feedback	30 Days	01/01/2021	01/02/2021
Step 2	Design the database	Requirement Development Testing Delivery Feedback	15 Days	01/02/2021	16/02/2021
Step 3	Develop the dashboard	Requirement Development Testing Delivery Feedback	30 Days	16/02/2021	16/03/2021
Step 4	Develop the user pages	Requirement Development Testing Delivery Feedback	30 Days	16/03/2021	16/04/2021

Table 1 - Project Time Table

Chapter 4
Requirements

Chapter 4
Requirements

In our project, we did our requirements by distributing a survey to 40 students and talk to people in the field.

4.1 Requirements Collection

According to the survey we published, it became clear to us that there is a large percentage of people who are looking for work or training are facing difficulty in that, and also employers face some problems in publishing his jobs and training.

More info about the survey questions is available in Appendix 1.

The following section shows the results of the survey:

- **76.9%** of job seekers have faced problems.
- The problem that faces the job Seekers:
    - عدم المعرفة بالشركات التي تحتاج او تقبل متدربين
    - عدم معرفة جميع الشركات التي تطلب الوظائف
    - لا يوجد منصة تساعد على إيجاد عمل او تدريب
    - لا اعرف من اين ابدا البحث
    - عدم معرفة الشركات التي تعمل في مجال اختصاصي
    - قلة فرص العمل
    - الأمور غير منظمة والاعلانات فقط عبر الفيس بوك
    - إيجاد فرص عمل مناسبة
    - لا أجد التدريب المناسب لي
- **43.6%** of job seekers their knowledge of companies that work in their field is low.
- The functions and features that should be in the site and it will help the job seekers:
    - سهولة البحث عن العمل الذي يناسب خبرتي
    - اشتراط وجود تفاصيل كافية للعمل
    - ان يحتوي على طريقة عرض مرتبة
    - سهولة تقديم الطلبات
    - تصنيف التخصصات ووجود عدد ساعات العمل وطبيعته ومكانه
    - توثيق الشركات والافراد المعلنين للوظائف
    - امكانية تقديم على العمل من خلال الموقع
    - تصنيف الوظائف حسب التخصصات
    - يمكن للشخص رؤية الموقع دون تسجيل دخول بدون السماح للخصائص الموجودة للمسجل مسبقا
    - المساعدة في الاستفسارات عن الوظيفة المعلنة

- **97.4%** of job seekers said that this site will help.
- **66.7%** of employers have faced problems when they post a job or training.
- The problems that faced the employers when they post a job or training.
  - بسبب الافتقار الى موقع يربط الشركات بالوظائف، والاعتماد بشكل كبير على موقع فيس بوك فقط.
  - عدم وجود منصة واحدة للتعامل معها والاضطرار للتعامل مع أكثر من عنوان
  - قلة معرفتي بأساليب الدعاية، والنشر، كثرة الأسئلة والاستفسارات عن وظيفة ولا يوجد لدى وقت للمتابعة، لا أجد الجمهور المناسب المهتم بالتدريب، لا اعلم كيف اوضح جميع تفاصيل العمل

- The functions and features that should be in the site and it will help the employers:
  - امكانية نشر الوظيفة
  - وجود بيانات حول المتدرب تشمل توجه الطالب ومهاراته الحالية
  - تقسيم الموقع الى اقسام للوظائف أو للتدريب
  - اعداد قوائم نختار منهم وظيفة مسوق وظيفة كاتب محتوى ..... الخ
  - كتابة تفاصيل الوظيفة وان تكون اجبارية مثل اسم الوظيفة .... عدد ساعات العمل ... الراتب .... اوقات الدوام ... المهام المطلوبة.. شروط تقديم الطلب .... وهكذا

- **83.3%** of employers said that this site will help them.

Our website can help to solving the problem. It can help jobseekers to search about the jobs and training in an easy way, and can help employers to publish their jobs and training.

4.2 System Requirements:

From the survey and by talking with people in the field, we defined the functional and nonfunction requirements.

4.2.1 Functional Requirements:

1 - Employer:

- The employer shall be able to sing up to the site.
- The employer shall be able to login to the site.
- The employer shall be able to verify his account.
- The employer shall be able to edit his account information.
- The verified employer shall be able to add jobs or training.
- The employer shall be able to display all job seekers that applied to his job or training.
- The employer shall be able to display all Inquiries that the job seekers sent to his job or training.
- The employer shall be able to search for job seekers.

2 - Job seeker:

- The job seeker shall be able to sign up to the site.
- The job seeker shall be able to login to the site.
- The job seeker shall be able to search for jobs or training.
- The job seeker shall be able to verify his account.
- The job seeker shall be able to edit his account information.
- The verified job seeker shall be able to apply to jobs or training.
- The job seekers shall be able to send an inquiry for a job or training.
- The job seeker shall be able to search for employers.

3 - System administrator:

- The system administrator shall be able to display a preview of the website statistics.
- The system administrator shall be able to (add, display, edit, delete) roles.
- The system administrator shall be able to (add, display, edit, delete) admins account.
- The system administrator shall be able to attach role to admin account.
- The system administrator shall be able to (add, display, edit, delete) employers account.

- The system administrator shall be able to (add, display, edit, delete) job seeker account.
- The system administrator shall be able to (add, display, edit, delete) regions.
- The system administrator shall be able to (add, display, edit, delete) categories.
- The system administrator shall be able to (add, display, edit, delete) jobs and training.
- The system administrator shall be able to verify job seekers and employers accounts.

4 - The guest:

- The guest shall be able to browse the jobs.
- The guest shall be able to browse the training.
- The guest shall be able to browse the job seekers.
- The guest shall be able to browse the employers.
- The guest shall be able to send a message in contact us.

4.2.2 Non-Functional Requirements:

4.2.2.1 Performance:

- Interactions with the server which require processing should take less than three seconds.
- Over reasonably common internet connection speeds, the server should respond to client requests in less than one second.
- Querying the database should take less than one second.
- The user interface for the software shall be compatible with any browser.

4.2.2.2 Security:

- Passwords should be stored as bcrypt hashes.
- The System will check for permission on any page.

4.2.2.3 Usability:

- 85% of users shall find the system easy to learn.
- 70% shall recommend the system to others.

Chapter 5
Analysis

Chapter 5
Analysis

In this chapter, we will define actors in our project, such as Employer, Job Seeker and administrator.

5.1 Actors

In our project we propose four actors:

5.1.1 Employer:

- Create new account.
- Login to the website.
- Verify account.
- Edit account information.
- Add new jobs.
- Add new training.
- Search for job seekers.
- Display and reply to the Inquiries on his job or training.
- Display Job Seekers that applied to his job or training.

5.1.2 Job Seeker:

- Create new account.
- Login to the website.
- Verify account.
- Edit account information.
- Search for jobs and training.
- Search for Companies.
- Apply to job or training.
- send Inquiry for a job or training.
- Save a job or training on the favorites list.
- Create new teams.

5.1.3 Admin:

- Login to the dashboard.
- Edit account information.
- Manage employers, job seekers and admins accounts.
- Create new roles.
- Attach role to an admin.
- Mange Categories.
- Manage Regions.
- Mange Jobs and Training.
- Manage jobs and training applications.
- Manage teams.

5.1.4 Guest:

- Search for jobs and training.
- Search for employers.
- Search for job seekers.
- Send a message in contact us.

5.2 Use Case Diagram:

5.2.1 General Use Case:

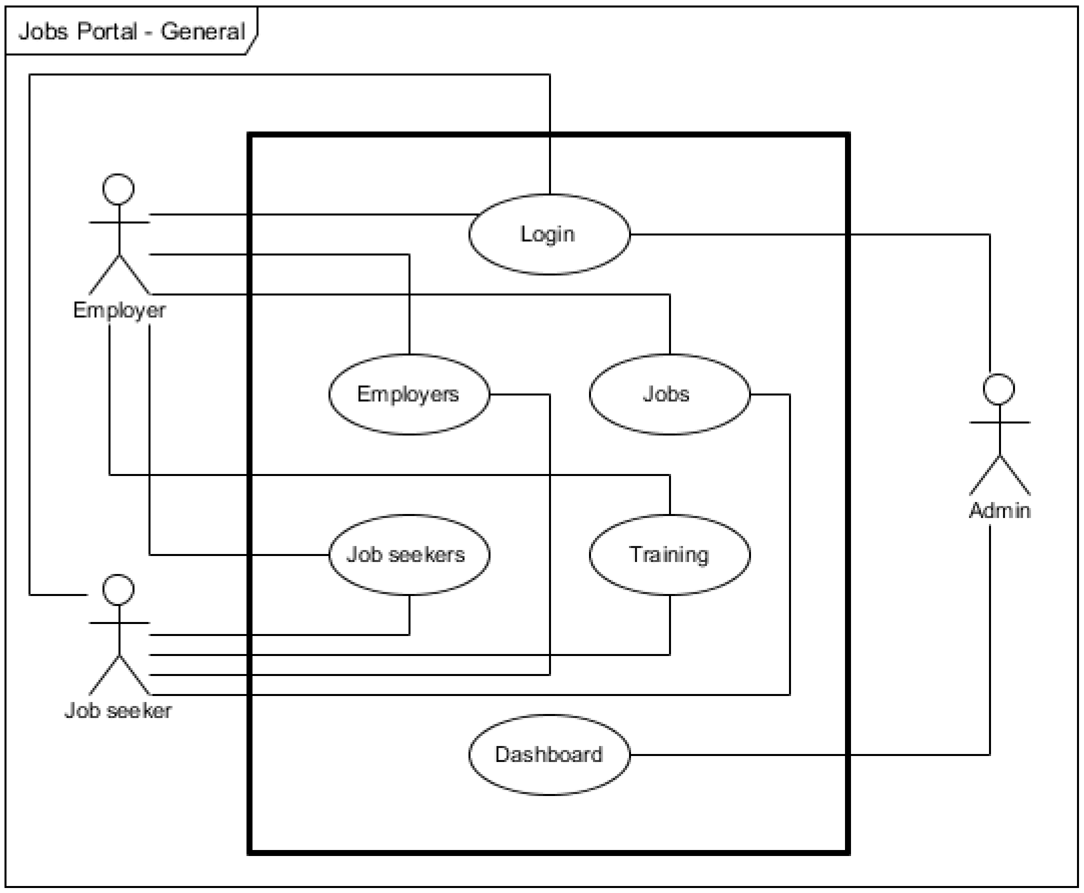


Figure 6 – General Use Case

5.2.2 Employer Use Case:

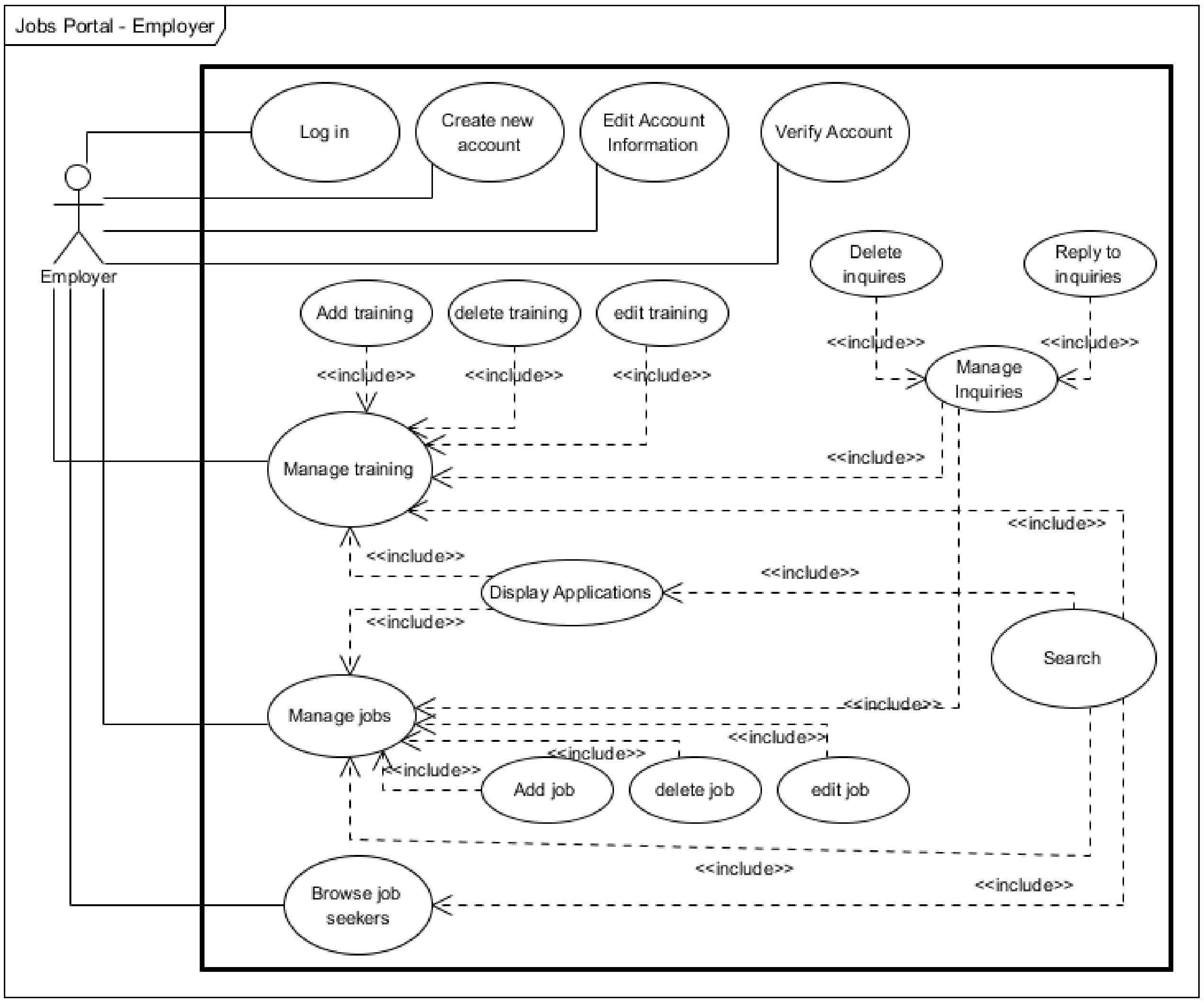


Figure 7 - Employer Use Case

5.2.3 Employer Use Case Table:

5.2.3.1 Login Use Case:

Indenter	EUC_01
Name	Login
Goal	Login to the system
Actors	Employer
Pre-conditions	User registered in the system
Post-conditions	Login user to the system and store token in the session
Primary Scenario	1. User open login page 2. User Enter Email and Password 3. The system validating the data 4. The system check if the user exists and credentials was correct 5. The system forwards the user to home page
Alternative Scenario	
Exceptional Scenarios	4.a The request validation was failed 4.b The user not registered in the system

Table 2 - Login Use Case

5.2.2.2 Create New Account Use Case:

Indenter	EUC_02
Name	Create new account
Goal	Register to the system
Actors	Employer
Pre-conditions	The User open the system
Post-conditions	The system creates a new user and login him
Primary Scenario	1. User open register page 2. User fill the register form 3. The system validating the request data 4. The system registers the user and login him
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed 3.b The system determine that this user was already registered.

Table 3 - Create New Account Use Case

5.2.2.3 Edit Account Information Use Case:

Indenter	EUC_03
Name	Edit Account Information
Goal	Change user profile data
Actors	Employer
Pre-conditions	The User login to the system
Post-conditions	User profile data was updated
Primary Scenario	1. The user open edit profile page 2. The user fill and update the form 3. The system validating the request data 4. The system updates the profile data
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed

Table 4 - Edit Account Information Use Case

5.2.2.4 Verify Account Use Case:

Indenter	EUC_04
Name	Verify Account
Goal	Make user account verified
Actors	Employer
Pre-conditions	The User login to the system
Post-conditions	Use account will be verified and some functionality will be unlocked
Primary Scenario	1. The user opens the verify account page 2. The user fills the form and upload the attachments 3. The system validates the request data 4. The system saves the user verify data in the database and make his account verified
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed 3.b The uploaded attachments exceed the allowed size or they type not allowed

Table 5 - Verify Account Use Case

5.2.2.5 Manage Training Use Case:

Indenter	EUC_05
Name	Manage Training
Goal	The user manages his training
Actors	Employer
Pre-conditions	The User login to the system
Post-conditions	Display all the training added by this user
Primary Scenario	1. The user opens manage training page 2. The system will return all the training added by this user
Alternative Scenario	
Exceptional Scenarios	

Table 6 - Manage Training Use Case

5.2.2.6 Add Training Use Case:

Indenter	EUC_06
Name	Add Training
Goal	Publish new training offer
Actors	Employer
Pre-conditions	The User login to the system and verified
Post-conditions	A new Training offer will publish and the jobseekers can apply to it
Primary Scenario	1. The user opens the add training page 2. The user fills the form inputs 3. The system validates the request data 4. The system saves the training in the database
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed

Table 7 - Add Training Use Case

5.2.2.7 Edit Training Use Case:

Indenter	EUC_07
Name	Edit Training
Goal	Modify Training offer data
Actors	Employer
Pre-conditions	The User login to the system and verified
Post-conditions	The training offer data will be updated
Primary Scenario	1. The user opens the edit training page 2. The user edits the data in the form 3. The system validates the request data 4. The system updates the training in the database
Alternative Scenario	
Exceptional Scenarios	1.a The training not found 2.b The user not the owner of this training 3.a The request validation was failed

Table 8 - Edit Training Use Case

5.2.2.8 Delete Training Use Case:

Indenter	EUC_08
Name	Delete Training
Goal	Delete Training offer
Actors	Employer
Pre-conditions	The User login to the system and verified
Post-conditions	The training offer data will be deleted from the database
Primary Scenario	1. The user clicks on delete button 2. The system asks for confirm 3. The system deletes the training from the database
Alternative Scenario	
Exceptional Scenarios	3.a The training not found 3.b The user not the owner of this training

Table 9 - Delete Training Use Case

5.2.2.9 Manage Job Use Case:

Indenter	EUC_09
Name	Manage Job
Goal	The user manages his jobs
Actors	Employer
Pre-conditions	The User login to the system
Post-conditions	Display all the jobs added by this user
Primary Scenario	3. The user opens manage jobs page 4. The system will return all the jobs added by this user
Alternative Scenario	
Exceptional Scenarios	

Table 10 - Manage Job Use Case

5.2.2.10 Add Job Use Case:

Indenter	EUC_10
Name	Add Job
Goal	Publish new job offer
Actors	Employer
Pre-conditions	The User login to the system and verified
Post-conditions	A new Job offer will publish and the jobseekers can apply to it
Primary Scenario	1. The user opens the add job page 2. The user fills the form inputs 3. The system validates the request data 4. The system saves the training in the database
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed

Table 11 - Add Job Use Case

5.2.2.11 Edit Job Use Case:

Indenter	EUC_11
Name	Edit Job
Goal	Modify job offer data
Actors	Employer
Pre-conditions	The User login to the system and verified
Post-conditions	The job offer data will be updated
Primary Scenario	1. The user opens the edit job page 2. The user edits the data in the form 3. The system validates the request data 4. The system updates the training in the database
Alternative Scenario	
Exceptional Scenarios	1.a The job not found 2.b The user not the owner of this job 3.a The request validation was failed

Table 12 - Edit Job Use Case

5.2.2.12 Delete Job Use Case:

Indenter	EUC_12
Name	Delete Job
Goal	Delete Job offer
Actors	Employer
Pre-conditions	The User login to the system and verified
Post-conditions	The job offer data will be deleted from the database
Primary Scenario	1. The user clicks on delete button 2. The system asks for confirm 3. The system deletes the job from the database
Alternative Scenario	
Exceptional Scenarios	3.a The job not found 3.b The user not the owner of this job

Table 13 - Delete Job Use Case

5.2.2.13 Search Use Case:

Indenter	EUC_13
Name	Search
Goal	Search for job or training
Actors	Employer
Pre-conditions	The User login to the system
Post-conditions	Display the training or jobs that match the search
Primary Scenario	1. The user open display job or training page 2. The use fills the search form 3. The system will return the data that match the search
Alternative Scenario	
Exceptional Scenarios	

Table 14 - Search Use Case

5.2.2.14 Display Applications Use Case:

Indenter	EUC_14
Name	Display Applications
Goal	Display Applications for job or training
Actors	Employer
Pre-conditions	The User login to the system and verified
Post-conditions	Display the job seekers that applied to his job or training
Primary Scenario	1. The user clicks on display applications button for the job or training 2. The system will return the job seekers that applied to the job or training
Alternative Scenario	
Exceptional Scenarios	2.a The job or training not found 2.b The user not the owner of this job or training

Table 15 - Display Application Use Case

5.2.2.15 Manage Inquiries Use Case:

Indenter	EUC_15
Name	Manage Inquiries
Goal	The user managed inquiries of his job or training
Actors	Employer
Pre-conditions	The User login to the system and verified
Post-conditions	Display all the inquiries added by job seekers for his job or training
Primary Scenario	1. The user opens inquiries page 2. The system will return all the inquiries for his job or training
Alternative Scenario	
Exceptional Scenarios	2.a The job or training not found 2.b The user not the owner of this job or training

Table 16 - Manage Inquiries Use Case

5.2.2.16 Reply to Inquiries Use Case:

Indenter	EUC_16
Name	Reply to Inquiries
Goal	The user reply to the inquiries of his job or training
Actors	Employer
Pre-conditions	The User login to the system and verified
Post-conditions	Reply to the inquiries added by job seekers for his job or training
Primary Scenario	1. The user opens inquiries page 2. The user add reply to inquiry 3. The system validates the request data 4. The system will add the reply to the inquiry in the database and send a notification for the job seeker
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed 4.a The job or training not found 4.b The user not the owner of this job or training

Table 17 - Reply Ti Inquiries Use Case

5.2.2.17 Delete Inquiries Use Case:

Indenter	EUC_17
Name	Delete Inquiries
Goal	The user reply to the inquiries of his job or training
Actors	Employer
Pre-conditions	The User login to the system and verified
Post-conditions	Reply to the inquiries added by job seekers for his job or training
Primary Scenario	1. The user clicks on delete button 2. The system asks for confirm 3. The system deletes the inquiry from the database
Alternative Scenario	
Exceptional Scenarios	3.a The inquiry not found 3.b The user not the owner of this job or training

Table 18 - Delete Inquiries Use Case

5.2.2.18 Display Job Seekers Use Case:

Indenter	EUC_18
Name	Browse Job Seekers
Goal	Browse all job seekers
Actors	Employer
Pre-conditions	The User login to the system
Post-conditions	Return all job seekers
Primary Scenario	1. The user open job seekers page 2. The system will return all job seekers
Alternative Scenario	
Exceptional Scenarios	

Table 19 - Display Job Seekers Use Case

5.2.3 Job Seeker Use Case:

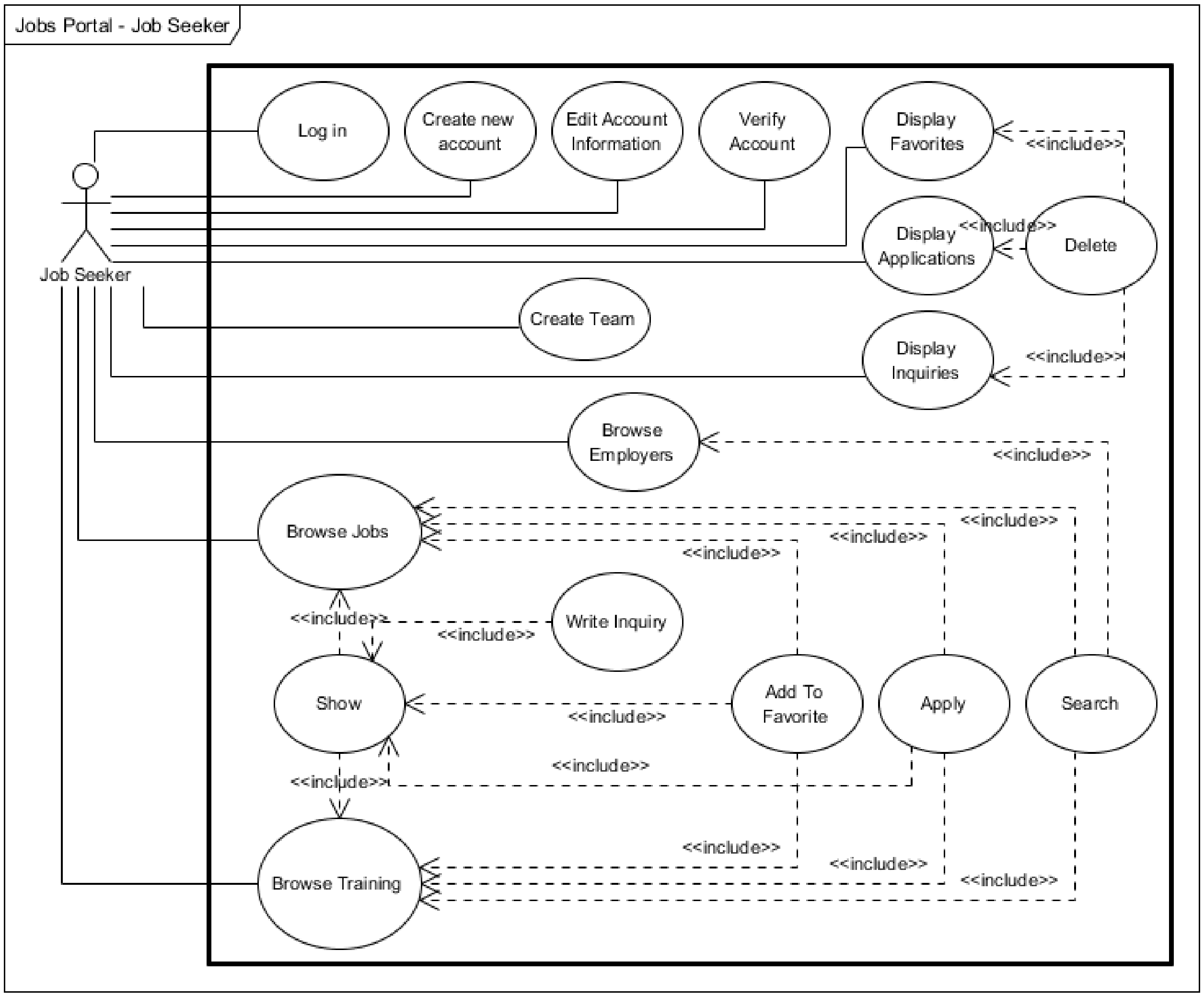


Figure 8 - Job Seeker Use Case

5.2.4 Job Seeker Use Case Table:

5.2.4.1 Login Use Case:

Indenter	JSUC_01
Name	Login
Goal	Login to the system
Actors	Job Seeker
Pre-conditions	User registered in the system
Post-conditions	Login user to the system and store token in the session
Primary Scenario	1. User open login page 2. User Enter Email and Password 3. The system validating the data 4. The system check if the user exists and credentials was correct 5. The system forwards the user to home page
Alternative Scenario	
Exceptional Scenarios	4.a The request validation was failed 4.b The user not registered in the system

Table 20 - Login Use Case

5.2.4.2 Create New Account Use Case:

Indenter	JSUC_02
Name	Create new account
Goal	Register to the system
Actors	Job Seeker
Pre-conditions	The User open the system
Post-conditions	The system creates a new user and login him
Primary Scenario	1. User open register page 2. User fill the register form 3. The system validating the request data 4. The system registers the user and login him
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed 3.b The system determine that this user was already registered.

Table 21 - Create Account Use Case

5.2.4.3 Edit Account Information Use Case:

Indenter	JSUC_03
Name	Edit Account Information
Goal	Change user profile data
Actors	Job Seeker
Pre-conditions	The User login to the system
Post-conditions	User profile data was updated
Primary Scenario	1. The user open edit profile page 2. The user fill and update the form 3. The system validating the request data 4. The system updates the profile data
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed

Table 22 - Edit Account Informatio Use Case

5.2.4.4 Verify Account Use Case:

Indenter	JSUC_04
Name	Verify Account
Goal	Make user account verified
Actors	Job Seeker
Pre-conditions	The User login to the system
Post-conditions	Use account will be verified and some functionality will be unlocked
Primary Scenario	1. The user opens the verify account page 2. The user fills the form and upload the attachments 3. The system validates the request data 4. The system saves the user verify data in the database and make his account verified
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed 3.b The uploaded attachments exceed the allowed size or they type not allowed

Table 23 - Verify Account Use Case

5.2.4.5 Display Favorites Use Case:

Indenter	JSUC_04
Name	Display Favorites
Goal	View all favorites list
Actors	Job Seeker
Pre-conditions	The User login to the system
Post-conditions	Return all favorites that added by the user
Primary Scenario	1. The user opens the favorites page 2. The system returns all the favorites that added by the user
Alternative Scenario	
Exceptional Scenarios	2.a Unauthorized

Table 24 - Display Favorites Use Case

5.2.4.6 Display Applications Use Case:

Indenter	JSUC_05
Name	Display Applications
Goal	View all jobs or training applications
Actors	Job Seeker
Pre-conditions	The User login to the system
Post-conditions	Return all jobs or training applications that added by the user
Primary Scenario	1. The user opens the display applications page 2. The system returns all the applications that added by the user
Alternative Scenario	
Exceptional Scenarios	2.b Unauthorized

Table 25 - Display Application Use Case

5.2.4.7 Display Inquiries Use Case:

Indenter	JSUC_06
Name	Display Inquiries
Goal	View all inquiries
Actors	Job Seeker
Pre-conditions	The User login to the system
Post-conditions	Return all inquiries added by the user and its replies
Primary Scenario	1. The user opens the display inquiries page 2. The system returns all the inquiries that added by the user and its replies
Alternative Scenario	
Exceptional Scenarios	2.b Unauthorized

Table 26 - Display Inquiries Use Case

5.2.4.8 Delete favorite Use Case:

Indenter	JSUC_08
Name	Delete
Goal	Delete favorite or application or inquiry
Actors	Job Seeker
Pre-conditions	The User login to the system
Post-conditions	Add new team on the database
Primary Scenario	1. The user clicks delete button 2. The system deletes the favorite or application or inquiry from the database
Alternative Scenario	
Exceptional Scenarios	2.a The favorite or application or inquiry not found 2.b Unauthorized

Table 27 – Delete Favorite Use Case

5.2.4.9 Create Team Use Case:

Indenter	JSUC_09
Name	Create Team
Goal	Create new team
Actors	Job Seeker
Pre-conditions	The User login to the system and verified
Post-conditions	Add new team on the database
Primary Scenario	1. The user opens the team page 2. The user fills the input name and add team members 3. The system validates the request data 4. The system adds a new team to the database
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed 3.b Team member already in another team 3.c Team member not verified

Table 28 - Create Team Use Case

5.2.4.10 Browse Employers Use Case:

Indenter	JSUC_10
Name	Browse Employers
Goal	View all employers
Actors	Job Seeker
Pre-conditions	The User login to the system
Post-conditions	View all employers
Primary Scenario	1. The open view employers page 2. The system will return all employers
Alternative Scenario	
Exceptional Scenarios	

Table 29 - Browse Employers Use Case

5.2.4.11 Search Use Case:

Indenter	JSUC_11
Name	Search
Goal	Search for Employers or Jobs or training
Actors	Job Seeker
Pre-conditions	The User login to the system
Post-conditions	Display all records that match the search
Primary Scenario	1. The user fills the search form 2. The system will return the data that match the search
Alternative Scenario	
Exceptional Scenarios	

Table 30 - Search Use Case

5.2.4.12 Browse Jobs Use Case:

Indenter	JSUC_12
Name	Browse Jobs
Goal	Display all jobs offer
Actors	Job Seeker
Pre-conditions	The User login to the system
Post-conditions	View all jobs offer
Primary Scenario	1. The user open jobs page 2. The system will return all the jobs
Alternative Scenario	
Exceptional Scenarios	

Table 31 - Browse Jobs Use Case

5.2.4.13 Browse Training Use Case:

Indenter	JSUC_13
Name	Browse Training
Goal	Display all training offer
Actors	Job Seeker
Pre-conditions	The User login to the system
Post-conditions	View all training offer
Primary Scenario	1. The user open training page 2. The system will return all the training
Alternative Scenario	
Exceptional Scenarios	

Table 32 - Browse Training Use Case

5.2.4.14 Show Use Case:

Indenter	JSUC_14
Name	Show
Goal	Display more info about the item
Actors	Job Seeker
Pre-conditions	The User login to the system
Post-conditions	Display view item page
Primary Scenario	1. The user clicks show details button 2. The system will return the details about the selected item
Alternative Scenario	
Exceptional Scenarios	2.b The item not found

Table 33 - Show Use Case

5.2.4.15 Write Inquiry Use Case:

Indenter	JSUC_15
Name	Write Inquiry
Goal	Send inquiry about a job or training to the employer
Actors	Job Seeker
Pre-conditions	The User login to the system and verified
Post-conditions	Send inquiry to the employer
Primary Scenario	1. The user open show detail page 2. The user fills the inquiry input 3. The system validates the request data 4. The system adds the inquiry in the database and send a notification for the employer
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed

Table 34 - Write Inquiry Use Case

5.2.4.16 Add to Favorite Use Case:

Indenter	JSUC_16
Name	Add To Favorite
Goal	Add item into the favorite list
Actors	Job Seeker
Pre-conditions	The User login to the system and verified
Post-conditions	Add item was added to the favorite list
Primary Scenario	1. The user clicks on save in favorite button 2. The system will save the item in the favorite list
Alternative Scenario	
Exceptional Scenarios	

Table 35 - Add to Favorite Use Case

5.2.4.17 Apply Use Case:

Indenter	JSUC_16
Name	Apply
Goal	Apply to job or training
Actors	Job Seeker
Pre-conditions	The User login to the system and verified
Post-conditions	The Application will be sent to the job or training
Primary Scenario	1. The user clicks on the apply button 2. The system will add the application to the database
Alternative Scenario	
Exceptional Scenarios	

Table 36 - Apply Use Case

5.2.5 Admin Use Case:

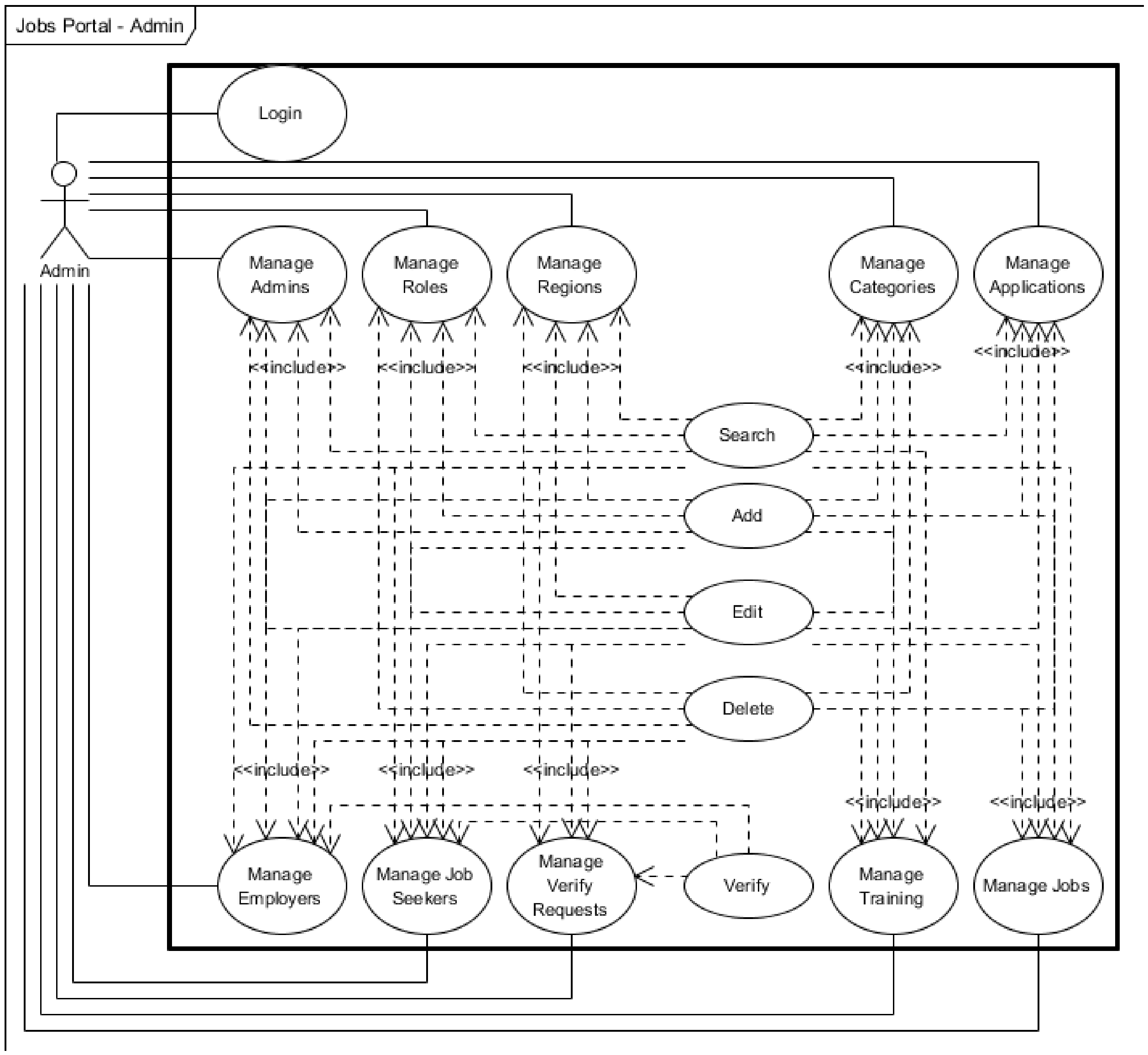


Figure 9 - Admin Use Case

5.2.6 Admin Use Case Table:

5.2.6.1 Login Use Case:

Indenter	AUC_01
Name	Login
Goal	Login to the system
Actors	Admin
Pre-conditions	Admin registered in the system
Post-conditions	Login admin to the system and store token in the session
Primary Scenario	1. Admin open login page 2. Admin Enter Email and Password 3. The system validating the data 4. The system check if the admin exists and credentials was correct 5. The system forwards the admin to dashboard page
Alternative Scenario	
Exceptional Scenarios	4.a The request validation was failed 4.b The admin not registered in the system

Table 37 - Login Use Case

5.2.6.2 Manage Admins Use Case:

Indenter	AUC_02
Name	Mange Admins
Goal	Display all admins data and manage it
Actors	Admin
Pre-conditions	Admin authenticated and has permission to display admins
Post-conditions	The system will return all admins
Primary Scenario	1. Admin open manage admins page 2. The system will return all admins
Alternative Scenario	
Exceptional Scenarios	1.a The admin not authenticated 1.b The admin not have permission

Table 38 - Manage Admins Use Case

5.2.6.3 Manage Roles Use Case:

Indenter	AUC_03
Name	Manage Roles
Goal	Display all roles data and manage it
Actors	Admin
Pre-conditions	Admin authenticated and has permission to display roles
Post-conditions	The system will return all roles
Primary Scenario	1. Admin open manage roles page 2. The system will return all roles
Alternative Scenario	
Exceptional Scenarios	1.a The admin not authenticated 1.b The admin not have permission

Table 39 - Manage Roles Use Case

5.2.6.4 Manage Regions Use Case:

Indenter	AUC_04
Name	Manage Regions
Goal	Display all regions data and manage it
Actors	Admin
Pre-conditions	Admin authenticated and has permission to display regions
Post-conditions	The system will return all regions
Primary Scenario	1. Admin open manage regions page 2. The system will return all regions
Alternative Scenario	
Exceptional Scenarios	1.a The admin not authenticated 1.b The admin not have permission

Table 40 - Manage Regions Use Case

5.2.6.5 Manage Categories Use Case:

Indenter	AUC_05
Name	Manage Categories
Goal	Display all categories data and manage it
Actors	Admin
Pre-conditions	Admin authenticated and has permission to display categories
Post-conditions	The system will return all categories
Primary Scenario	3. Admin open manage categories page 4. The system will return all categories
Alternative Scenario	
Exceptional Scenarios	1.a The admin not authenticated 1.b The admin not have permission

Table 41 - Manage Categories Use Case

5.2.6.6 Manage Employers Use Case:

Indenter	AUC_06
Name	Manage Employers
Goal	Display all employers data and manage it
Actors	Admin
Pre-conditions	Admin authenticated and has permission to display employers
Post-conditions	The system will return all employers
Primary Scenario	5. Admin open manage employers page 6. The system will return all employers
Alternative Scenario	
Exceptional Scenarios	1.a The admin not authenticated 1.b The admin not have permission

Table 42 - Manage Employers Use Case

5.2.6.7 Manage Job Seekers Use Case:

Indenter	AUC_07
Name	Manage Job Seekers
Goal	Display all job seekers data and manage it
Actors	Admin
Pre-conditions	Admin authenticated and has permission to display job seekers
Post-conditions	The system will return all job seekers
Primary Scenario	7. Admin open manage job seekers page 8. The system will return all job seekers
Alternative Scenario	
Exceptional Scenarios	1.a The admin not authenticated 1.b The admin not have permission

Table 43 - Manage Job Seekers Use Case

5.2.6.8 Manage Verify Requests Use Case:

Indenter	AUC_08
Name	Manage Verify Requests
Goal	Display all verify requests data and manage it
Actors	Admin
Pre-conditions	Admin authenticated and has permission to display verify requests
Post-conditions	The system will return all verify requests
Primary Scenario	9. Admin open manage verify requests page 10. The system will return all verify requests
Alternative Scenario	
Exceptional Scenarios	1.a The admin not authenticated 1.b The admin not have permission

Table 44 - Manage Verify Requests Use Case

5.2.6.9 Manage Training Use Case:

Indenter	AUC_09
Name	Manage Training
Goal	Display all training data and manage it
Actors	Admin
Pre-conditions	Admin authenticated and has permission to display training
Post-conditions	The system will return all training
Primary Scenario	11. Admin open manage training page 12. The system will return all training
Alternative Scenario	
Exceptional Scenarios	1.a The admin not authenticated 1.b The admin not have permission

Table 45 - Manage Training Use Case

5.2.6.10 Manage Jobs Use Case:

Indenter	AUC_10
Name	Manage Jobs
Goal	Display all jobs data and manage it
Actors	Admin
Pre-conditions	Admin authenticated and has permission to display jobs
Post-conditions	The system will return all training
Primary Scenario	13. Admin open manage jobs page 14. The system will return all jobs
Alternative Scenario	
Exceptional Scenarios	1.a The admin not authenticated 1.b The admin not have permission

Table 46 - Manage Jobs Use Case

* In the following use cases the "item" references to admins or roles or regions or categories or applications or employers or job seekers or verify requests or training or jobs.

5.2.6.11 Search Use Case:

Indenter	AUC_11
Name	Search
Goal	Search for the “item”
Actors	Admin
Pre-conditions	The admin authenticated and have permission to display “item”
Post-conditions	The system will return all records that match the search
Primary Scenario	1. The admin fills the search form 2. The system will return the data that match the search
Alternative Scenario	
Exceptional Scenarios	

Table 47 - Search Use Case

5.2.6.12 Add Use Case:

Indenter	AUC_12
Name	Add
Goal	Add new “item”
Actors	Admin
Pre-conditions	The admin authenticated and have permission to add “item”
Post-conditions	The system will create new “item”
Primary Scenario	1. The admin opens the add “item” page 2. The admin fills the form data 3. The system validates the request data 4. The system adds new “item” in the database
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed

Table 48 - Add Use Case

5.2.6.13 Edit Use Case:

Indenter	AUC_13
Name	Edit
Goal	Edit the selected “item”
Actors	Admin
Pre-conditions	The admin authenticated and have permission to edit “item”
Post-conditions	The system will edit the “item” data
Primary Scenario	1. The admin opens the edit “item” page 2. The admin fills the form data 3. The system validates the request data 4. The system edit “item” data in the database
Alternative Scenario	
Exceptional Scenarios	3.a The request validation was failed

Table 49 - Edit Use Case

5.2.6.14 Delete Use Case:

Indenter	AUC_14
Name	Delete
Goal	Delete the selected "item"
Actors	Admin
Pre-conditions	The admin authenticated and have permission to delete "item"
Post-conditions	The system will delete the "item"
Primary Scenario	1. The admin clicks on delete button 2. The system asks for confirm 3. The system deletes the job from the database
Alternative Scenario	
Exceptional Scenarios	3.a The "item" not found 3.b Unauthorized

Table 50 - Delete Use Case

5.2.6.15 Verify Use Case:

Indenter	AUC_15
Name	Verify
Goal	Verify Employer or Job Seeker account
Actors	Admin
Pre-conditions	The admin authenticated and have permission to verify employer or job seeker
Post-conditions	The system will verify employer or job seeker account
Primary Scenario	1. The admin clicks on verify button 2. The system verifies employer or job seeker
Alternative Scenario	
Exceptional Scenarios	2.a The employer or job seeker not found 2.b Unauthorized

Table 51 - Verify Use Case

5.2.7 Sequence Diagram:

We did sequence diagrams for Login, Create Account, Send Verification request, verify account, Post job or training and apply to job or training.

5.2.7.1 Login Sequence Diagram:

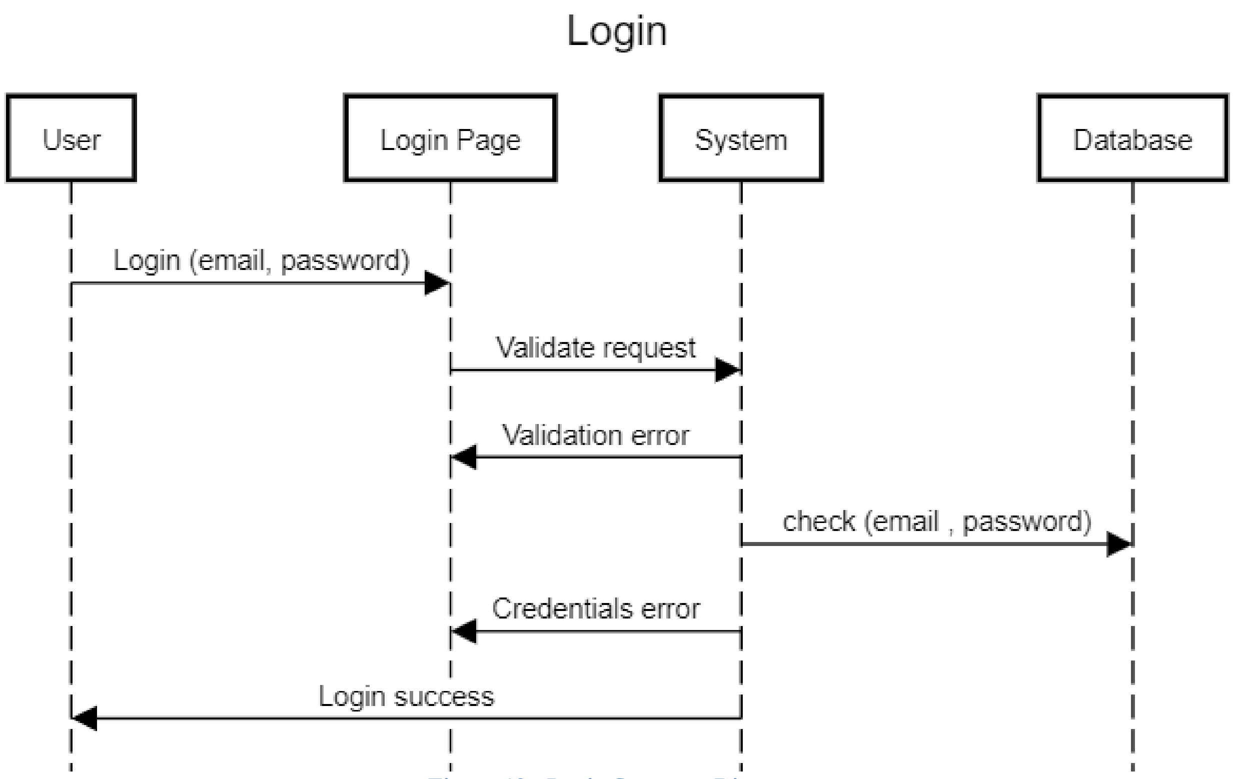


Figure 10 - Login Sequence Diagram

5.2.7.2 Create Account Sequence Diagram:

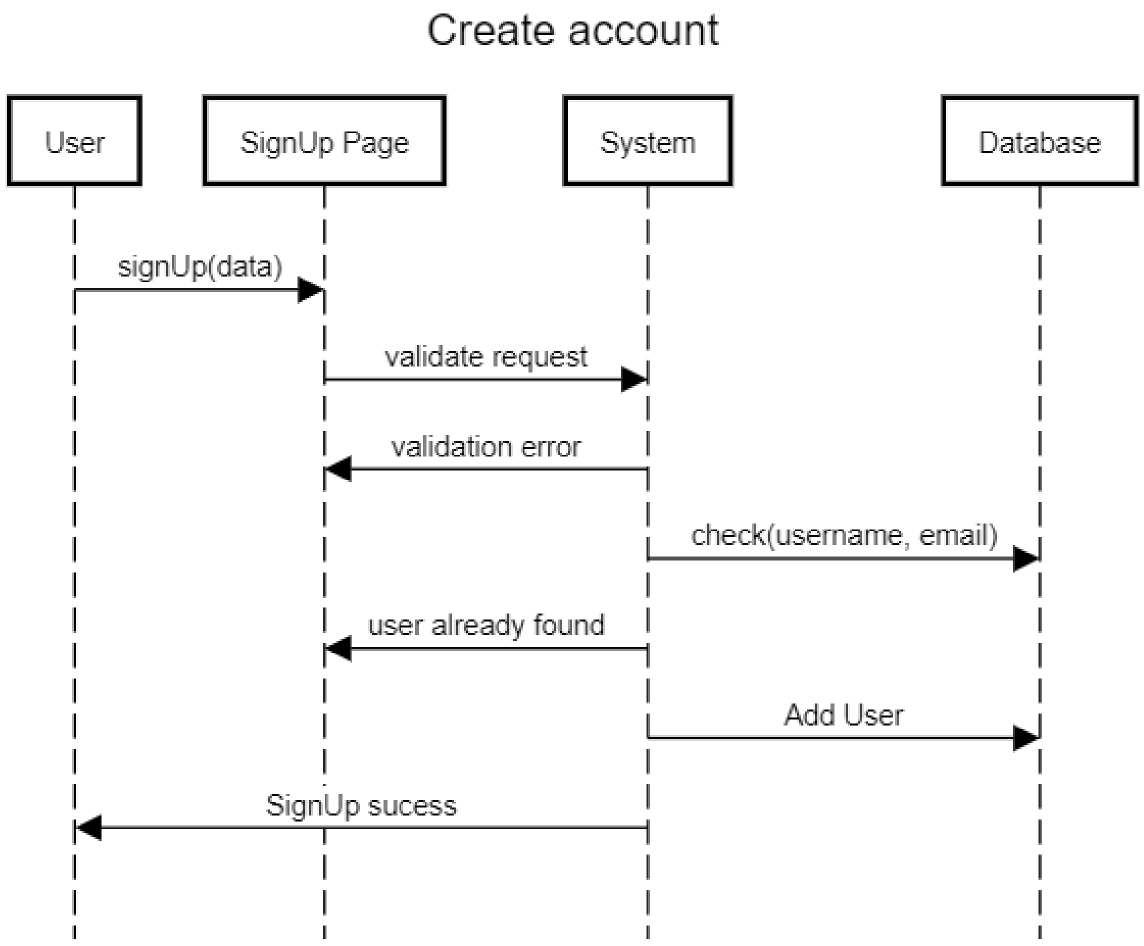


Figure 11 - Create Account Sequence Diagram

5.2.7.3 Send Verification request Sequence Diagram:

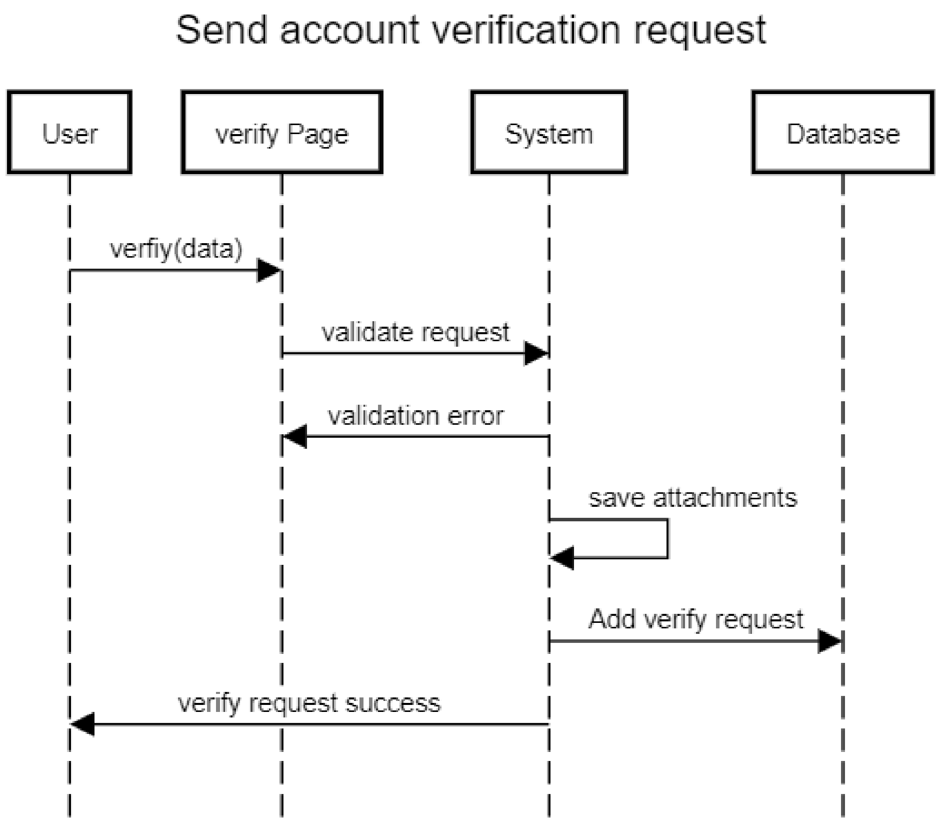


Figure 12 - Send Account verification request

5.2.7.4 Verify account Sequence Diagram:

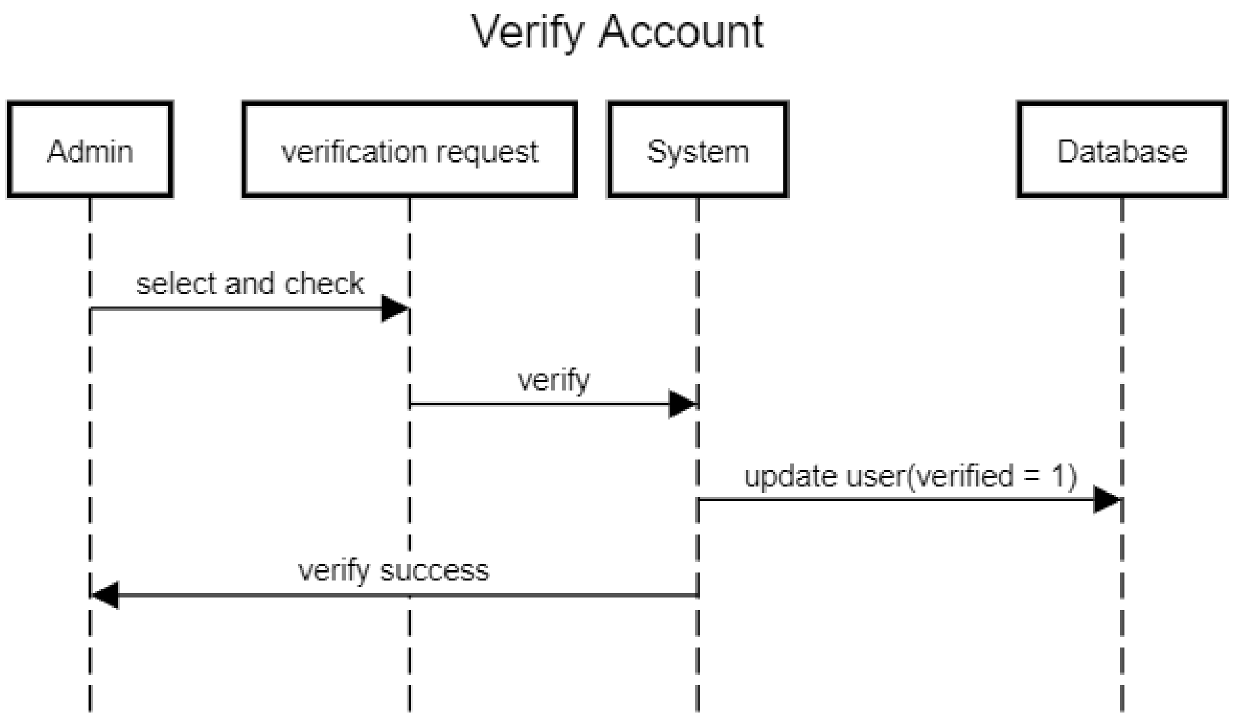


Figure 13 - Verify Account Sequence Diagram

5.2.7.5 Post Job or training Sequence Diagram:

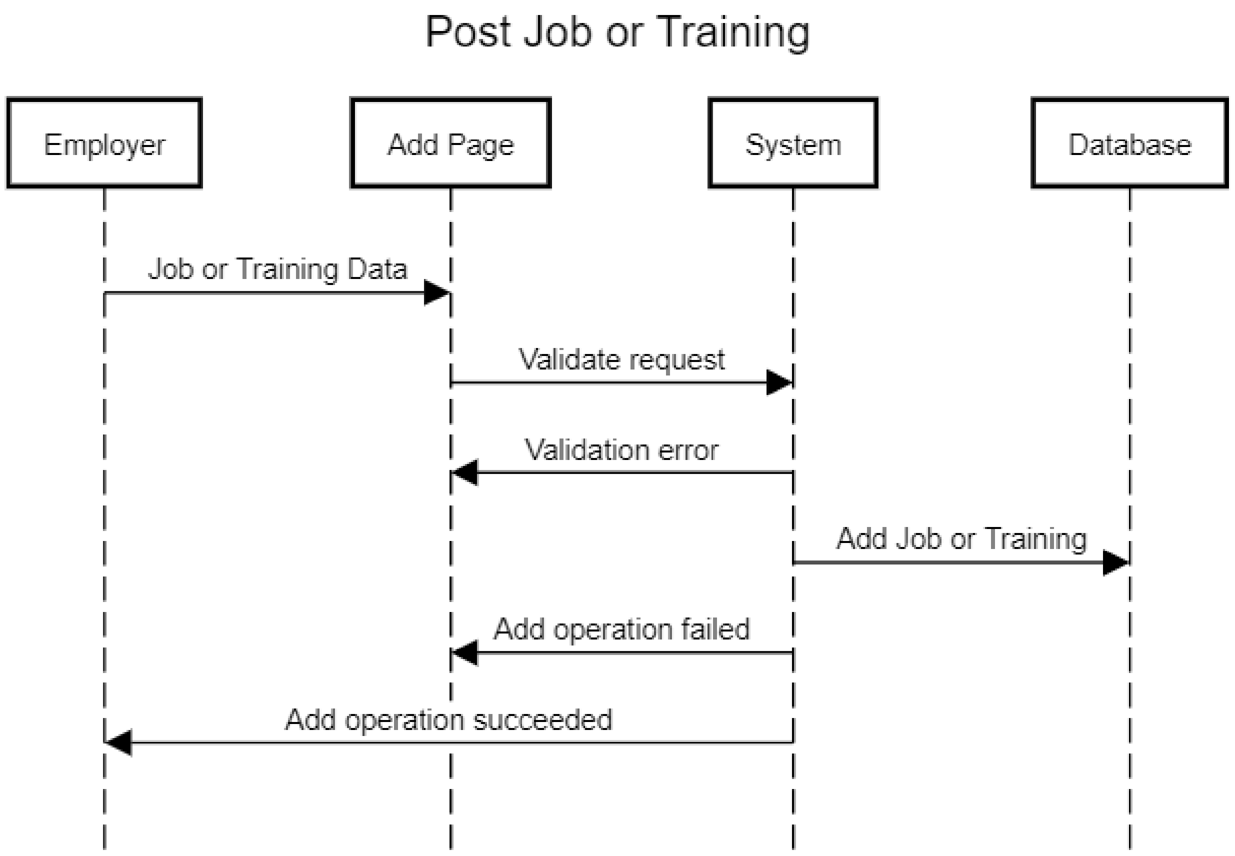


Figure 14 - Post Job or training Sequence Diagram

5.2.7.6 Apply to Job or training Sequence Diagram:

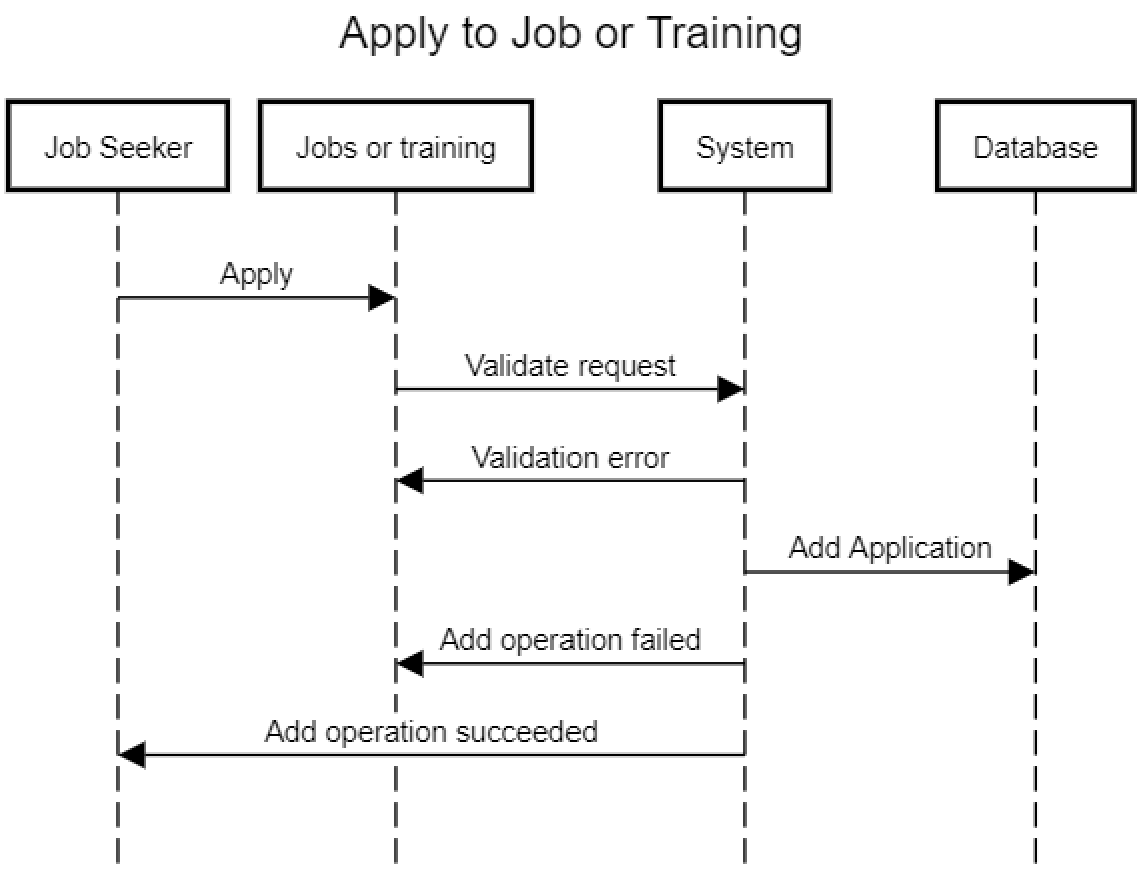


Figure 15 - Apply to job or training sequence diargram

Chapter 6

Design

Chapter 6
Design

6.1. Architecture Design:

We did our implementation using Laravel. Laravel is a PHP-based web framework that is largely based on the MVC architecture.
This diagram shows in figure16 the four main components of the Laravel application:

1. **Model:** In Laravel, each of the database table has a corresponding "Model" that allow us to interact with that table. Models gives you the way to retrieve, insert, and update information into your data table.[6]
2. **View:** Views are meant to hold the presentation logic in any Laravel application. Views separates presentation logic from the application logic. View can be a complete web page or parts of page like header and footer.[7]
3. **Controller:** Controller is the entry point for any MVC framework-based application, it receives the incoming HTTP requests and process it communicating with models and views, then return the results back to the web browser. In Laravel, controllers are completely responsible for handling the application logic.[8]
4. **Router:** Routing is one of the core components of Laravel framework, it is simply a mechanism that performs the mapping for your requests to specific controller action.[9]

In this system, we created a new route file for the Dashboard and register it in RouteServiceProvider, also we created two controller's folder one for the Dashboard and the other for the Front, and create three Auth Controllers one for the Admin and the two others for the Employer and Job Seeker, and we created two view folders one for the Dashboard and the other for the Front, and we created two layouts' folders one for the Dashboard and one for the Front.

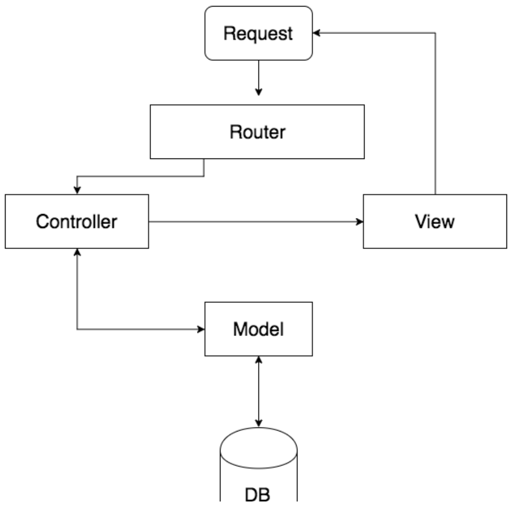


Figure 16 - Architecture Design

6.2. ER with tables:

Our system consists of 26 tables.

6.2.1. ER Diagram:

6.2.1.1 Job Seeker Diagram:

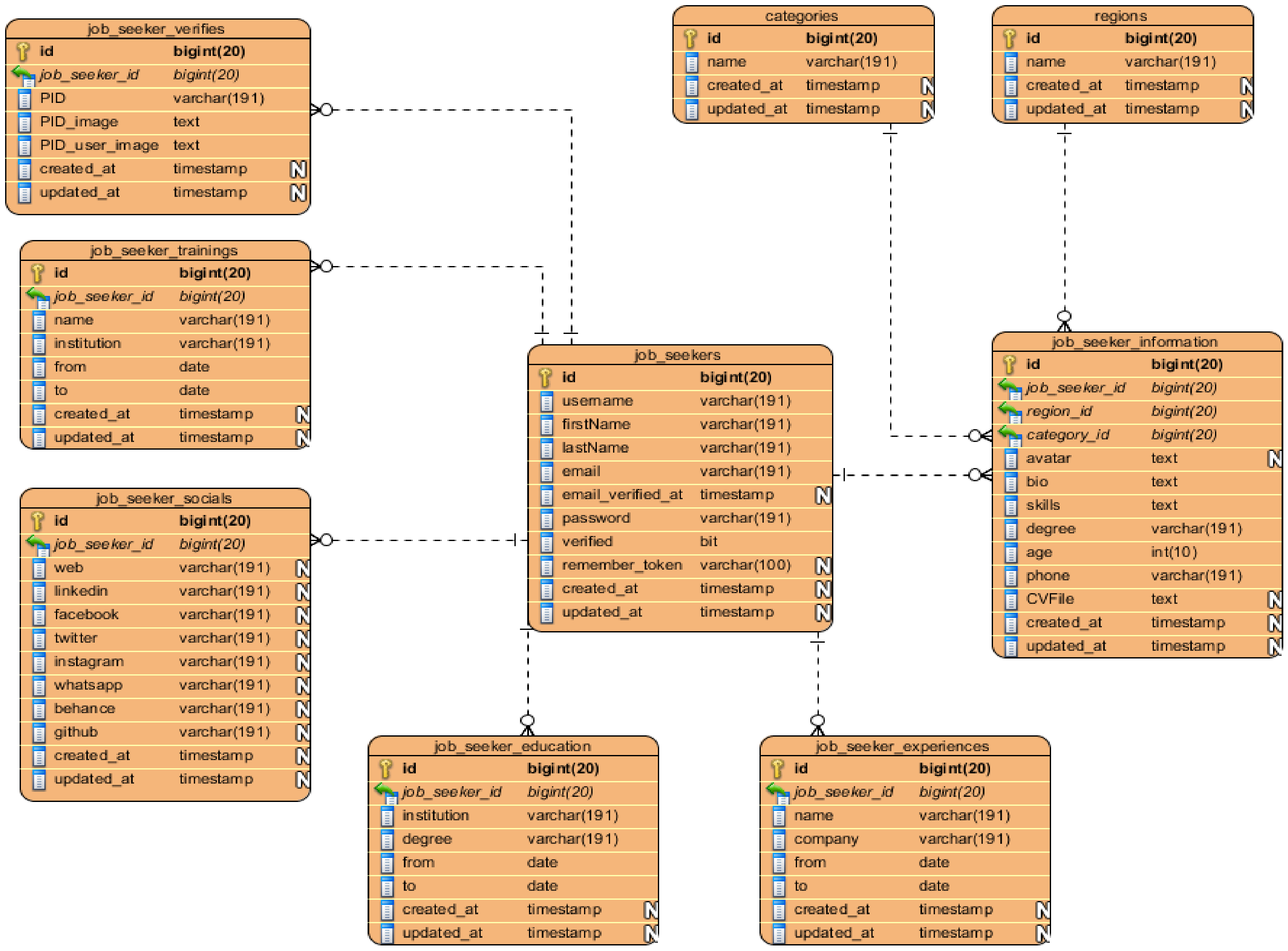


Figure 17 - JobSeeker Diagram

6.2.1.2 Employer Diagram:

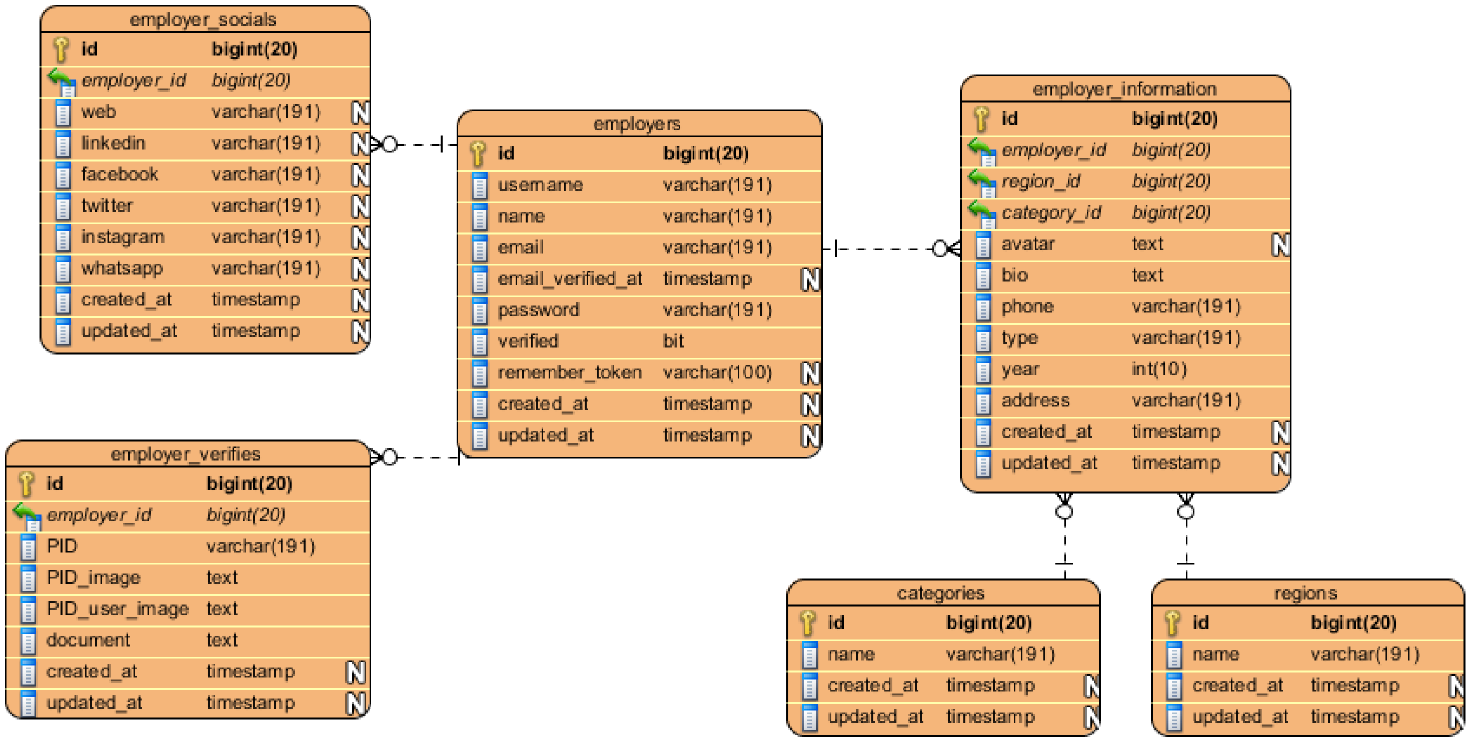


Figure 18 - Employer Diagram

6.2.1.3 ER Job and Training Diagram:

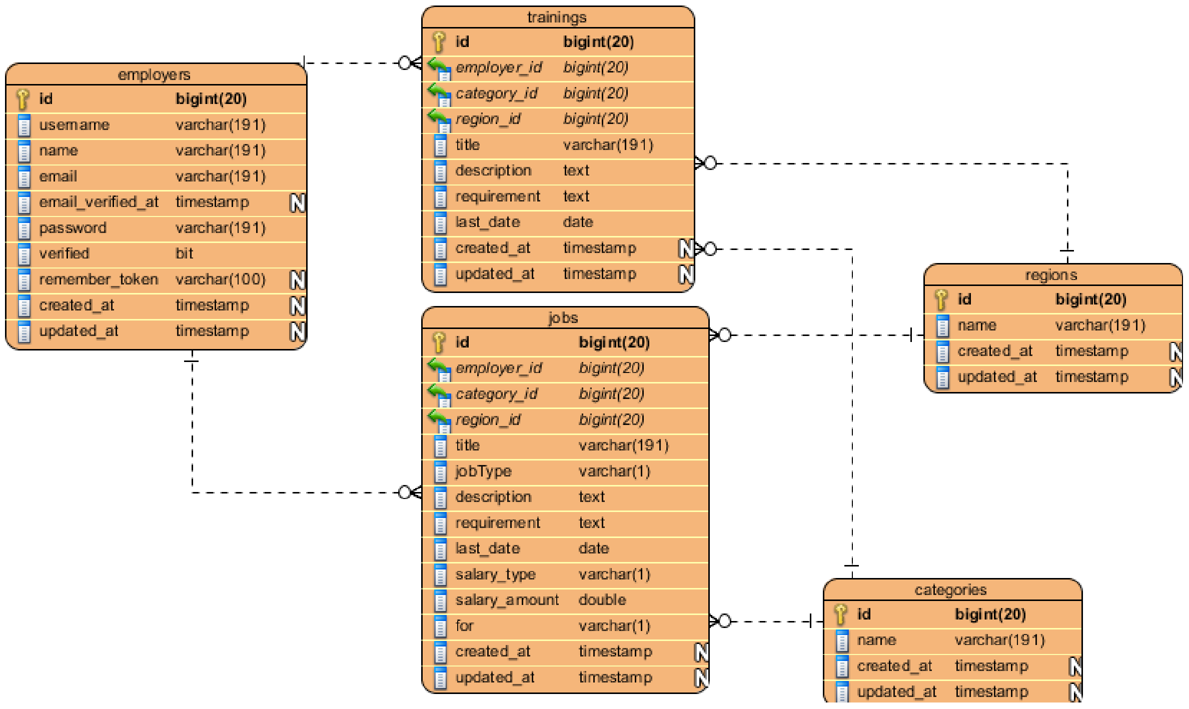


Figure 19 - Job and Training Diagram

6.2.1.4 Application and Inquire Diagram:

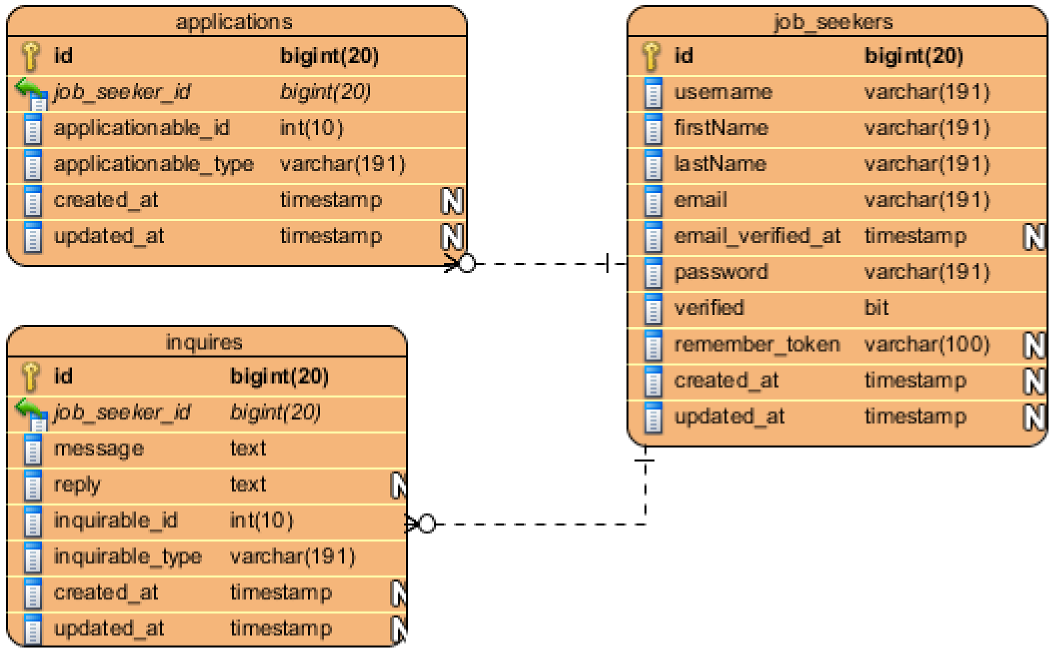


Figure 20 - Application and Inquire Diagram

6.2.1.5 Team Diagram:

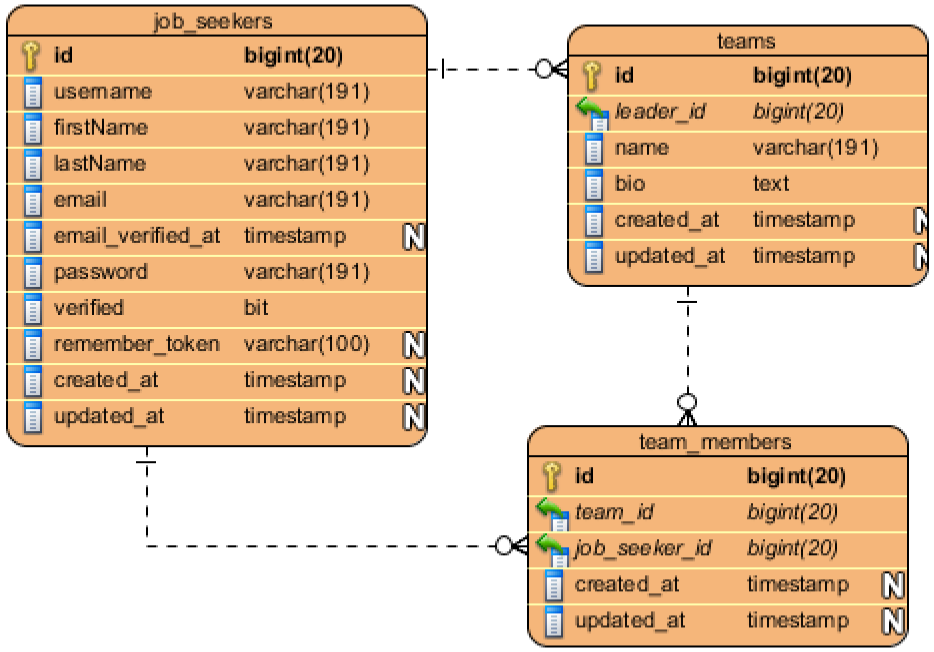


Figure 21 - Team Diagram

6.2.1.6 Roles, permissions, admins and messages Diagram:

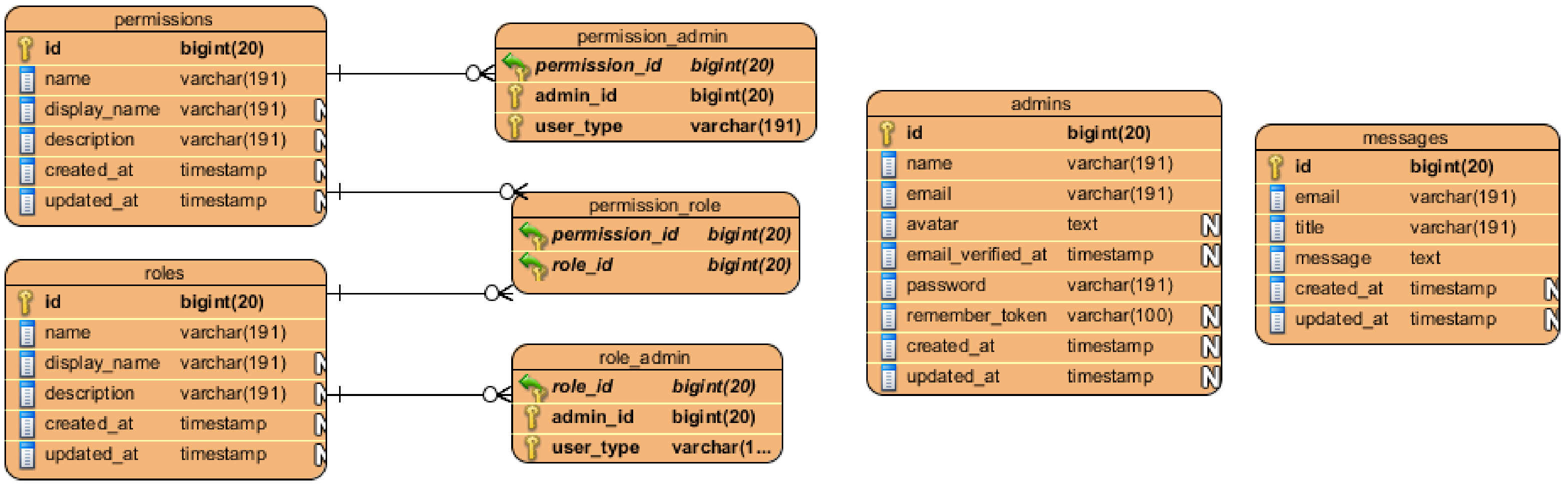


Figure 22 - Roles, permissions, admins and messages

6.2.2 Tables:

6.2.2.1 admins:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
name	varchar(191)		NO					
email	varchar(191)		NO					
avatar	text		YES	NULL				
email_verified_at	timestamp		YES	NULL				
password	varchar(191)		NO					
remember_token	varchar(100)		YES	NULL				
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 52 - admins table

6.2.2.2 applications:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
job_seeker_id	bigint(20)	UNSIGNED	NO			-> job_seekers.id ON UPDATE RESTRICT ON DELETE CASCADE		
applicationable_id	int(11)		NO					
applicationable_type	varchar(191)		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 53 - applications table

6.2.2.3 categories:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
name	varchar(191)		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 54 - categories table

6.2.2.4 employers:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
username	varchar(191)		NO					
name	varchar(191)		NO					
email	varchar(191)		NO					
email_verified_at	timestamp		YES	NULL				
password	varchar(191)		NO					
verified	tinyint(1)		NO	0				
remember_token	varchar(100)		YES	NULL				
created_at	timestamp		YES	NULL				

Table 55 – employers table

6.2.2.5 employer_information:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
employer_id	bigint(20)	UNSIGNED	NO			-> employers.id ON UPDATE RESTRICT ON DELETE CASCADE		
region_id	bigint(20)	UNSIGNED	NO			-> regions.id ON UPDATE RESTRICT ON DELETE CASCADE		
category_id	bigint(20)	UNSIGNED	NO			-> categories.id ON UPDATE RESTRICT ON DELETE CASCADE		
avatar	text		YES	NULL				
bio	text		NO					
phone	varchar(191)		NO					
type	varchar(191)		NO					
year	int(11)		NO					
address	varchar(191)		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 56 - employer_information table

6.2.2.6 employer_socials:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
employer_id	bigint(20)	UNSIGNED	NO			-> employers.id ON UPDATE RESTRICT ON DELETE CASCADE		
web	bigint(20)		YES	NULL				
linkedin	bigint(20)		YES	NULL				
facebook	text		YES	NULL				
twitter	text		YES	NULL				
instagram	varchar(191)		YES	NULL				
wahtsapp	varchar(191)		YES	NULL				
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 57 - employer_socials table

6.2.2.7 employer_verifies:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
employer_id	bigint(20)	UNSIGNED	NO			-> employers.id ON UPDATE RESTRICT ON DELETE CASCADE		
PID	varchar(191)		NO					
PID_image	text		NO					
PID_user_image	text		NO					
document	text		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 58 - employer_verifies table

6.2.2.8 inquires:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
job_seeker_id	bigint(20)	UNSIGNED	NO			-> job_seekers.id ON UPDATE RESTRICT ON DELETE CASCADE		
message	varchar(191)		NO					
reply	text		YES	NULL				
inquirable_id	text		NO					
inquirable_type	text		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 59 - inquires table

6.2.2.9 jobs:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
employer_id	bigint(20)	UNSIGNED	NO			-> employers.id ON UPDATE RESTRICT ON DELETE CASCADE		
category_id	bigint(20)	UNSIGNED	NO			-> categories.id ON UPDATE RESTRICT ON DELETE CASCADE		
region_id	bigint(20)	UNSIGNED	NO			-> regions.id ON UPDATE RESTRICT ON DELETE CASCADE		
title	varchar(191)		NO					
jobType	enum('1', '2', '3')		NO					
description	text		NO					
requirement	text		NO					
last_date	date		NO					
salary_type	enum('1', '2')		NO					
salary_amount	double(8, 2)		NO					
for	enum('1', '2')		NO					

created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 60 - jobs table

6.2.2.10 job_seekers:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
username	varchar(191)		NO					
firstName	varchar(191)		NO					
lastName	varchar(191)		NO					
email	varchar(191)		NO					
email_verified_at	timestamp		YES	NULL				
password	varchar(191)		NO					
verified	tinyint(1)		NO	0				
remember_token	varchar(100)		YES	NULL				
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 61 - job_seekers table

6.2.2.11 job_seeker_education:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
job_seeker_id	bigint(20)	UNSIGNED	NO			-> job_seekers.id ON UPDATE RESTRICT ON DELETE CASCADE		
institution	varchar(191)		NO					
degree	varchar(191)		NO					
from	date		NO					
to	date		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 62 - job_seeker_education table

6.2.2.12 job_seeker_experiences:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
job_seeker_id	bigint(20)	UNSIGNED	NO			-> job_seekers.id ON UPDATE RESTRICT ON DELETE CASCADE		
name	varchar(191)		NO					
company	varchar(191)		NO					
from	date		NO					
to	date		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 63 - job_seeker_experiences table

6.2.2.13 Job_seeker_information:

Table 64 - job_seeker_information table

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
job_seeker_id	bigint(20)	UNSIGNED	NO			-> job_seekers.id ON UPDATE RESTRICT ON DELETE CASCADE		
region_id	bigint(20)	UNSIGNED	NO			-> regions.id ON UPDATE RESTRICT ON DELETE CASCADE		
category_id	bigint(20)	UNSIGNED	NO			-> categories.id ON UPDATE RESTRICT ON DELETE CASCADE		
avatar	text		YES	NULL				
bio	text		NO					
skills	text		NO					
degree	varchar(191)		NO					
age	int(11)		NO					
phone	varchar(191)		NO					
CVFile	text		YES	NULL				
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

6.2.2.14 Job_seeker_socials:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
job_seeker_id	bigint(20)	UNSIGNED	NO			-> job_seekers.id ON UPDATE RESTRICT ON DELETE CASCADE		
web	bigint(20)		YES	NULL				
linkedin	bigint(20)		YES	NULL				
facebook	text		YES	NULL				
twitter	text		YES	NULL				
instagram	varchar(191)		YES	NULL				
wahtsapp	varchar(191)		YES	NULL				
behance	varchar(191)		YES	NULL				
github	varchar(191)		YES	NULL				
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 65 - job_seeker_socials table

6.2.2.15 job_seeker_trainings:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
job_seeker_id	bigint(20)	UNSIGNED	NO			-> job_seekers.id ON UPDATE RESTRICT ON DELETE CASCADE		
name	varchar(191)		NO					
institution	varchar(191)		NO					
from	date		NO					
to	date		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 66 - job_seeker_trainings table

6.2.2.16 Job_seeker_verifies:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
job_seeker_id	bigint(20)	UNSIGNED	NO			-> job_seekers.id ON UPDATE RESTRICT ON DELETE CASCADE		
PID	varchar(191)		NO					
PID_image	text		NO					
PID_user_image	text		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 67 - job_seeker_verifies table

6.2.2.17 messages:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
email	varchar(191)		NO					
title	varchar(191)		NO					
message	text		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 68 - messages table

6.2.2.18 permissions:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
name	varchar(191)		NO					
display_name	varchar(191)		YES	NULL				
description	varchar(191)		YES	NULL				
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 69 - permissions table

6.2.2.19 permission_admin:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
permssion_id	bigint(20)	UNSIGNED	NO			-> permissions.id ON UPDATE CASCADE ON DELETE CASCADE		
admin_id	bigint(20)		NO					
user_type	varchar(191)		NO					

Table 70 - permission_admin table

6.2.2.20 permission_role:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
permssion_id	bigint(20)	UNSIGNED	NO			-> permissions.id ON UPDATE CASCADE ON DELETE CASCADE		
role_id	bigint(20)	UNSIGNED	NO			-> roles.id ON UPDATE CASCADE ON DELETE CASCADE		

Table 71 - permission_role table

6.2.2.21 regions:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
name	varchar(191)		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 72 - regions table

6.2.2.22 roles:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
name	varchar(191)		NO					
display_name	varchar(191)		YES	NULL				
description	varchar(191)		YES	NULL				
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 73 - roles table

6.2.2.23 role_admin:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
role_id	bigint(20)	UNSIGNED	NO			-> roles.id ON UPDATE CASCADE ON DELETE CASCADE		
admin_id	bigint(20)	UNSIGNED	NO					
user_type	varchar(191)		NO					

Table 74 - role_admin table

6.2.2.24 teams:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
leader_id	bigint(20)	UNSIGNED	NO			-> job_seekers.id ON UPDATE RESTRICT ON DELETE CASCADE		
name	varchar(191)		NO					
bio	text		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 75 - teams table

6.2.2.25 team_members:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
team_id	bigint(20)	UNSIGNED	NO			-> teams.id jobsON UPDATE RESTRICT ON DELETE CASCADE		
job_seeker_id	bigint(20)	UNSIGNED	NO			-> job_seekers.id ON UPDATE RESTRICT ON DELETE CASCADE		
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 76 - team_members table

6.2.2.26 trainings:

Column	Type	Attribute	Null	Default	Extra	Links to	Comments	MIME
id	bigint(20)	UNSIGNED	NO		auto_increment			
employer_id	bigint(20)	UNSIGNED	NO			-> employers.id ON UPDATE RESTRICT ON DELETE CASCADE		
category_id	bigint(20)	UNSIGNED	NO			-> categories.id ON UPDATE RESTRICT ON DELETE CASCADE		
region_id	bigint(20)	UNSIGNED	NO			-> regions.id ON UPDATE RESTRICT ON DELETE CASCADE		
title	varchar(191)		NO					
description	text		NO					
requirement	text		NO					
last_date	date		NO					
created_at	timestamp		YES	NULL				
updated_at	timestamp		YES	NULL				

Table 77 - trainings table

6.3 Screens:

This is the jobs screen that shows all the jobs on the site, and you can search for specific jobs using the search box on the side.

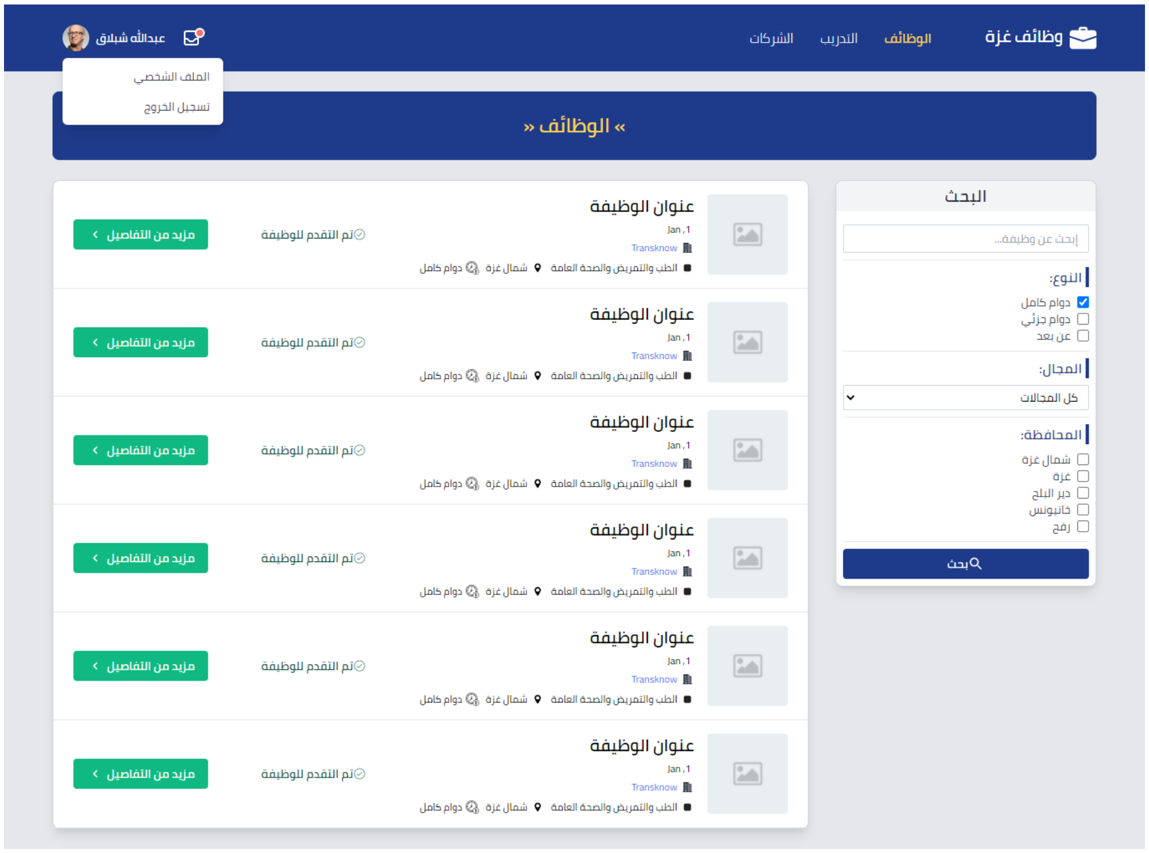
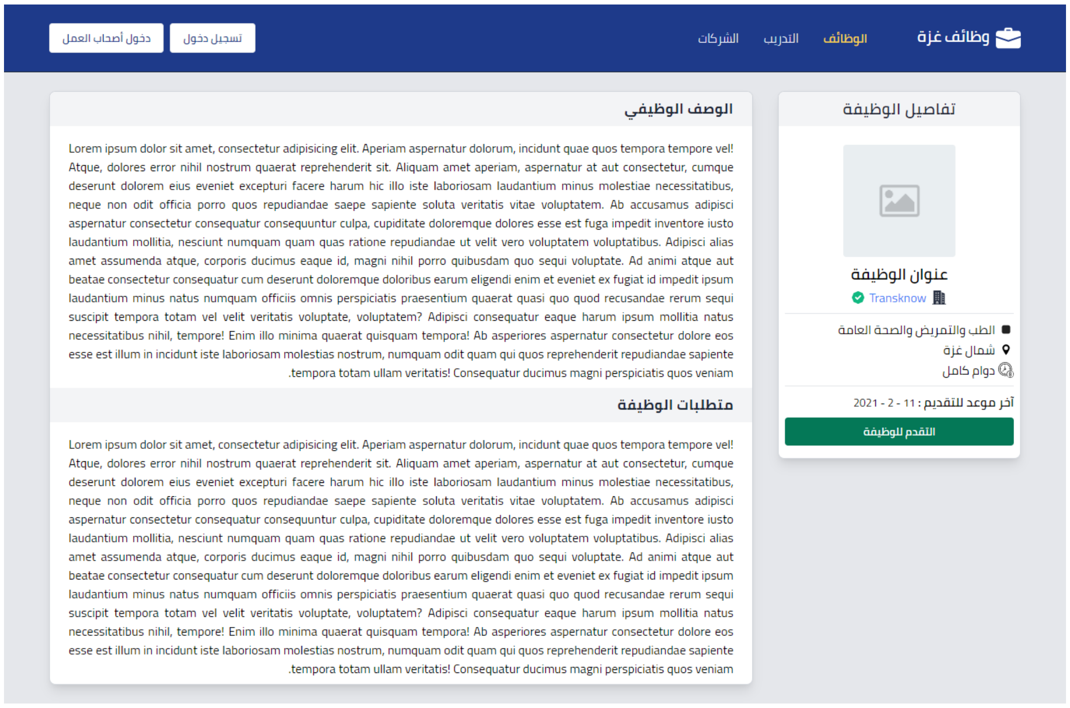


Figure 23 - Jobs page screen

This is the job details page that shows more details about the job like the description and requirements, and you can apply to the job on this page.



الوصف الوظيفي

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aperiam aspernatur dolorum, incidunt quae quos tempora tempore vel! Atque, dolores error nihil nostrum quaerat reprehenderit sit. Aliquam amet aperiam, aspernatur at aut consectetur, cumque deserunt dolorem eius eveniet excepturi facere harum hic illo iste laboriosam laudantium minus molestiae necessitatibus, neque non odit officia porro quos repudiandae saepe sapiente soluta veritatis vitae voluptatem. Ab accusamus adipisci aspernatur consectetur consequatur consequuntur culpa, cupiditate doloremque dolores esse est fuga impedit inventore iusto laudantium mollitia, nesciunt numquam quam quas ratione repudiandae ut velit vero voluptatem voluptatibus. Adipisci alias amet assumenda atque, corporis ducimus eaque id, magni nihil porro quibusdam quo sequi voluptate. Ad animi atque aut beatae consectetur consequatur cum deserunt doloremque doloribus earum eligendi enim et eveniet ex fugiat id impedit ipsum laudantium minus natus numquam officiis omnis perspiciatis praesentium quaerat quasi quo quod recusandae rerum sequi suscipit tempora totam vel velit veritatis voluptate, voluptatem? Adipisci consequatur eaque harum ipsum mollitia natus necessitatibus nihil, tempore! Enim illo minima quaerat quisquam tempora! Ab asperiores aspernatur consectetur dolore eos esse est illum in incidunt iste laboriosam molestias nostrum, numquam odit quam qui quos reprehenderit repudiandae sapiente .tempora totam ullam veritatis! Consequatur ducimus magni perspiciatis quos veniam

متطلبات الوظيفة

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aperiam aspernatur dolorum, incidunt quae quos tempora tempore vel! Atque, dolores error nihil nostrum quaerat reprehenderit sit. Aliquam amet aperiam, aspernatur at aut consectetur, cumque deserunt dolorem eius eveniet excepturi facere harum hic illo iste laboriosam laudantium minus molestiae necessitatibus, neque non odit officia porro quos repudiandae saepe sapiente soluta veritatis vitae voluptatem. Ab accusamus adipisci aspernatur consectetur consequatur consequuntur culpa, cupiditate doloremque dolores esse est fuga impedit inventore iusto laudantium mollitia, nesciunt numquam quam quas ratione repudiandae ut velit vero voluptatem voluptatibus. Adipisci alias amet assumenda atque, corporis ducimus eaque id, magni nihil porro quibusdam quo sequi voluptate. Ad animi atque aut beatae consectetur consequatur cum deserunt doloremque doloribus earum eligendi enim et eveniet ex fugiat id impedit ipsum laudantium minus natus numquam officiis omnis perspiciatis praesentium quaerat quasi quo quod recusandae rerum sequi suscipit tempora totam vel velit veritatis voluptate, voluptatem? Adipisci consequatur eaque harum ipsum mollitia natus necessitatibus nihil, tempore! Enim illo minima quaerat quisquam tempora! Ab asperiores aspernatur consectetur dolore eos esse est illum in incidunt iste laboriosam molestias nostrum, numquam odit quam qui quos reprehenderit repudiandae sapiente .tempora totam ullam veritatis! Consequatur ducimus magni perspiciatis quos veniam

Figure 24 - Job details page screen

This is the employers screen that shows all the employers on the site, and you can search for specific employers using the search box on the side.

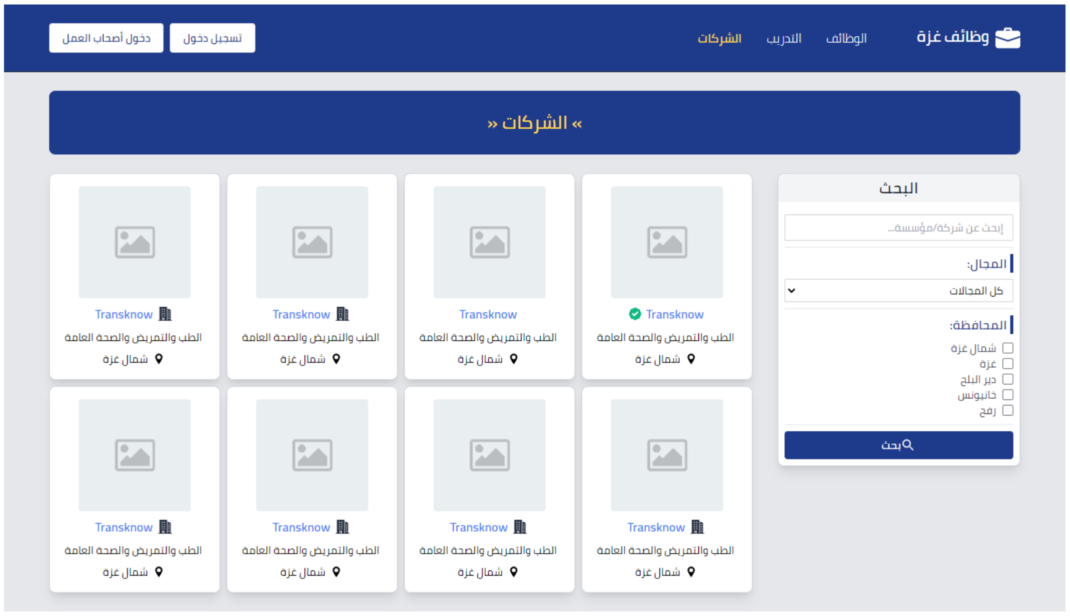
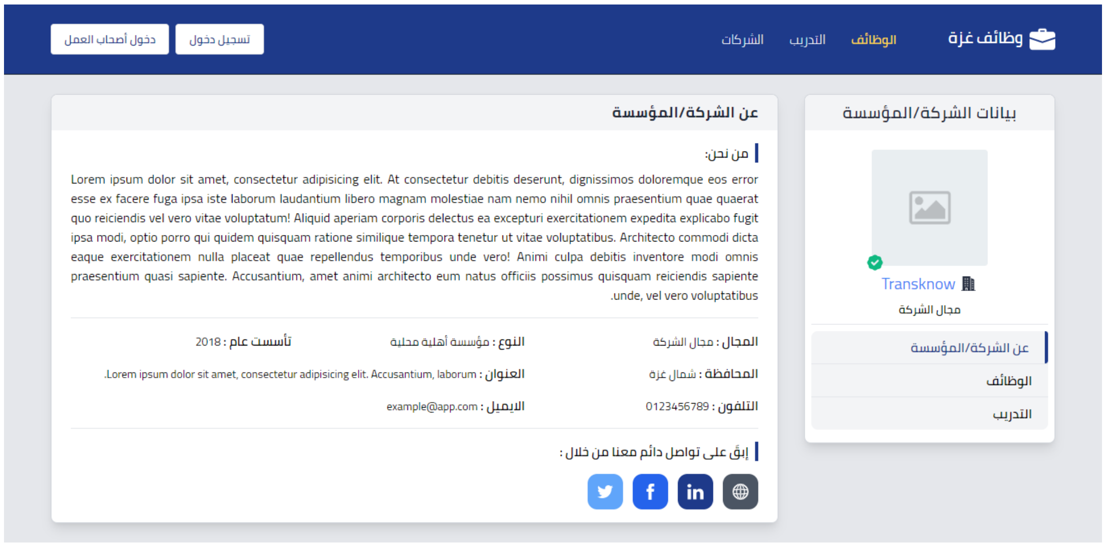


Figure 25 – Employers page screen

This is the employer profile page, that shows more info about the employer like the bio, address and email, and there a section for the employer social media links.



عن الشركة/المؤسسة

من نحن:

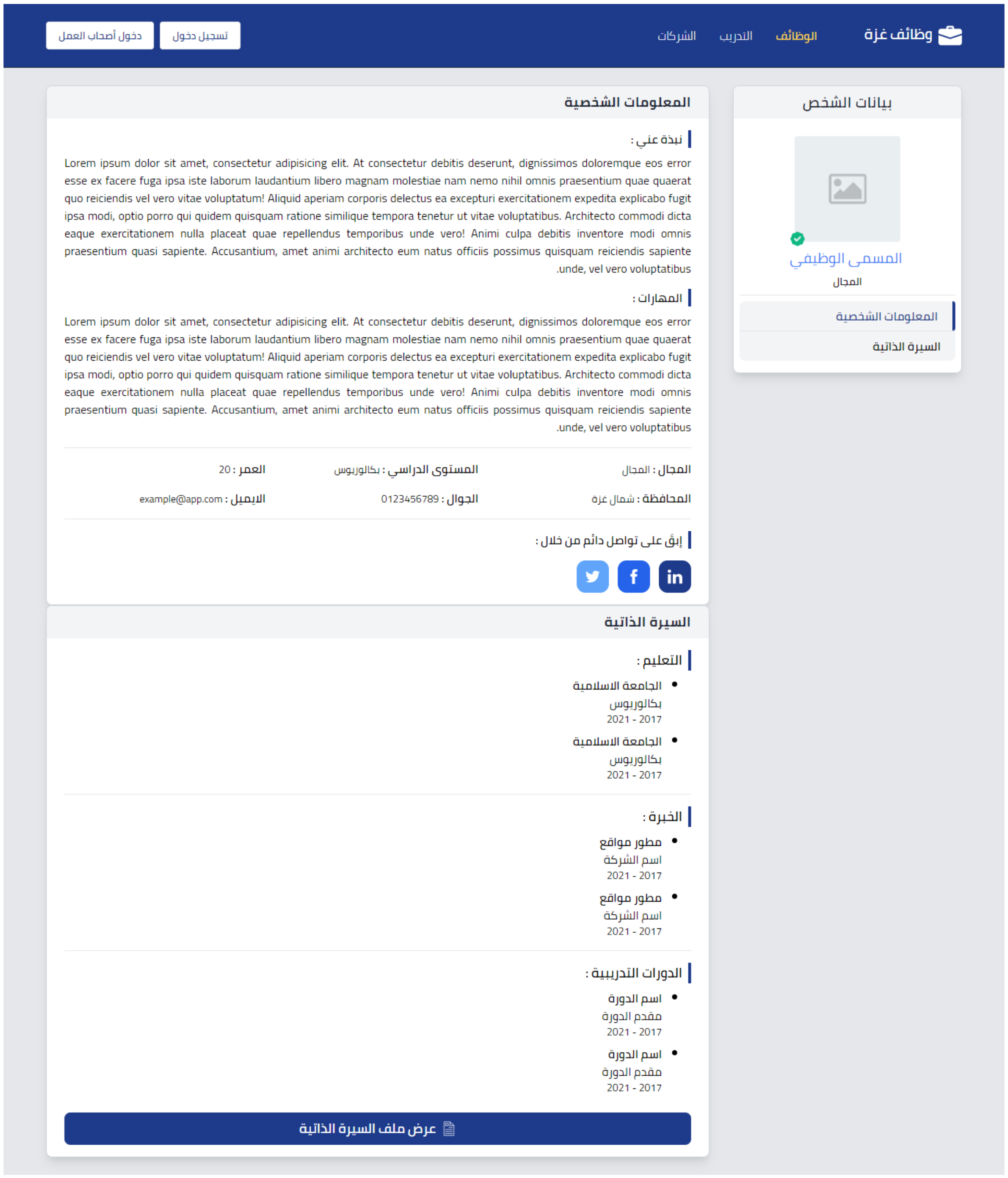
Lorem ipsum dolor sit amet, consectetur adipisicing elit. At consectetur debitis deserunt, dignissimos doloremque eos error esse ex facere fuga ipsa iste laborum laudantium libero magnam molestiae nam nemo nihil omnis praesentium quae quaerat quo reiciendis vel vero vitae voluptatum! Aliquid aperiam corporis delectus ea excepturi exercitationem expedita explicabo fugit ipsa modi, optio porro qui quidem quisquam ratione similique tempora tenetur ut vitae voluptatibus. Architecto commodi dicta eaque exercitationem nulla placeat quae repellendus temporibus unde vero! Animi culpa debitis inventore modi omnis praesentium quasi sapiente. Accusantium, amet animi architecto eum natus officiis possimus quisquam reiciendis sapiente .unde, vel vero voluptatibus

المجال : مجال الشركة	النوع : مؤسسة أهلية محلية	تأسست عام : 2018
المحافظة : شمال غزة	العنوان : .Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusantium, laborum	
التلفون : 0123456789	الايميل : example@app.com	

إبقَ على تواصل دائم معنا من خلال :

Figure 26 - Employer profile page screen

This is the job seeker profile page that shows more information about him like bio, skills, age, email and phone, and there a section for the job seeker social media links.



المعلومات الشخصية

نبذة عني :

Lorem ipsum dolor sit amet, consectetur adipisicing elit. At consectetur debitis deserunt, dignissimos doloremque eos error esse ex facere fuga ipsa iste laborum laudantium libero magnam molestiae nam nemo nihil omnis praesentium quae quaerat quo reiciendis vel vero vitae voluptatum! Aliquid aperiam corporis delectus ea excepturi exercitationem expedita explicabo fugit ipsa modi, optio porro qui quidem quisquam ratione similique tempora tenetur ut vitae voluptatibus. Architecto commodi dicta eaque exercitationem nulla placeat quae repellendus temporibus unde vero! Animi culpa debitis inventore modi omnis praesentium quasi sapiente. Accusantium, amet animi architecto eum natus officiis possimus quisquam reiciendis sapiente .unde, vel vero voluptatibus

المهارات :

Lorem ipsum dolor sit amet, consectetur adipisicing elit. At consectetur debitis deserunt, dignissimos doloremque eos error esse ex facere fuga ipsa iste laborum laudantium libero magnam molestiae nam nemo nihil omnis praesentium quae quaerat quo reiciendis vel vero vitae voluptatum! Aliquid aperiam corporis delectus ea excepturi exercitationem expedita explicabo fugit ipsa modi, optio porro qui quidem quisquam ratione similique tempora tenetur ut vitae voluptatibus. Architecto commodi dicta eaque exercitationem nulla placeat quae repellendus temporibus unde vero! Animi culpa debitis inventore modi omnis praesentium quasi sapiente. Accusantium, amet animi architecto eum natus officiis possimus quisquam reiciendis sapiente .unde, vel vero voluptatibus

المجال : المجال	المستوى الدراسي : بكالوريوس	العمر : 20
المحافظة : شمال غزة	الجوال : 0123456789	الايميل : example@app.com

إبق على تواصل دائم من خلال :

السيرة الذاتية

التعليم :

- الجامعة الاسلامية
  بكالوريوس
  2021 - 2017
- الجامعة الاسلامية
  بكالوريوس
  2021 - 2017

الخبرة :

- مطور مواقع
  اسم الشركة
  2021 - 2017
- مطور مواقع
  اسم الشركة
  2021 - 2017

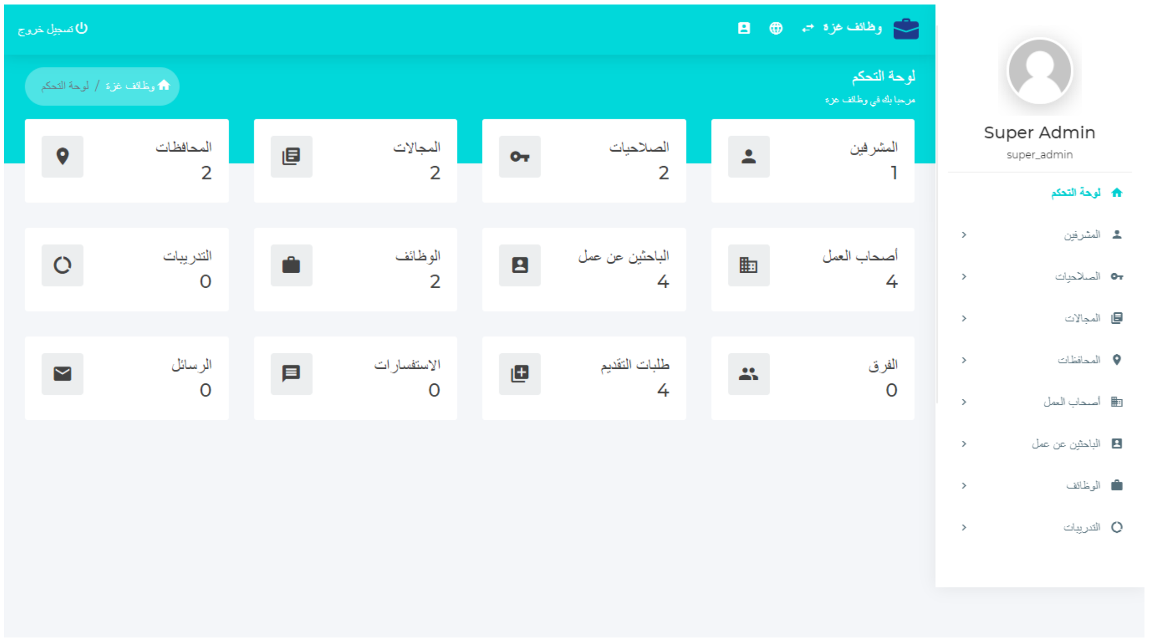
الدورات التدريبية :

- اسم الدورة
  مقدم الدورة
  2021 - 2017
- اسم الدورة
  مقدم الدورة
  2021 - 2017

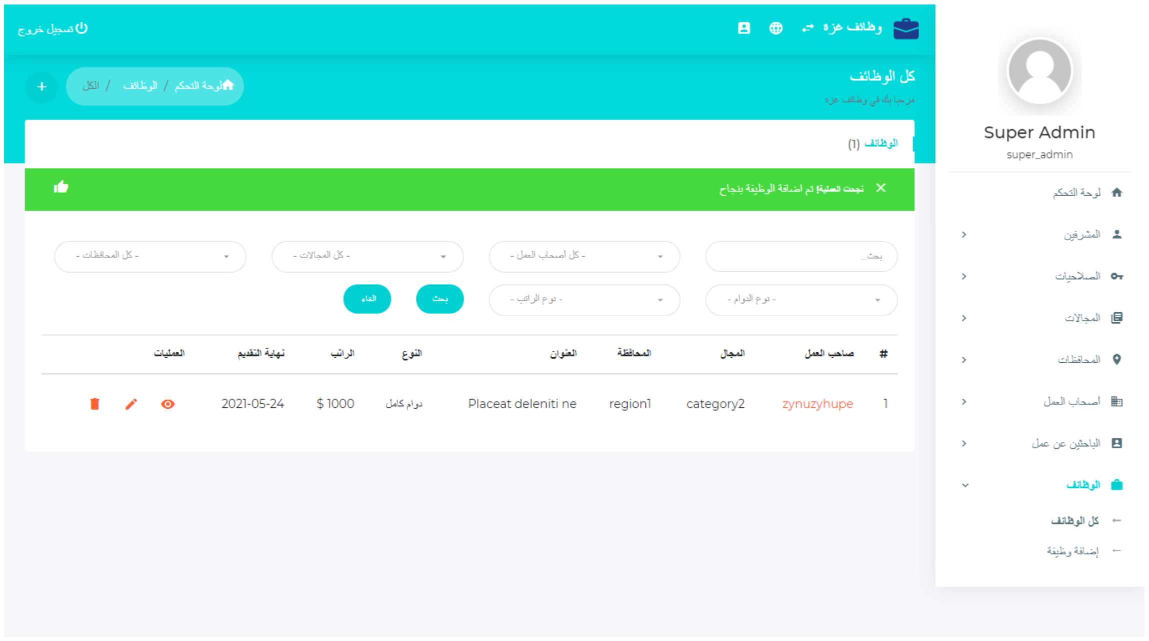
عرض ملف السيرة الذاتية

Figure 27 - Job Seeker profile page screen

This is the dashboard main screen the shows the system statistics.



This is the display jobs page that shows all jobs, and the admin can manage them.



#	صاحب العمل	المجال	المحافظة	العنوان	النوع	الراتب	نهاية التقديم	العمليات
1	zynuzyhupe	category2	region1	Placeat deleniti ne	دوام كامل	$ 1000	2021-05-24	

Figure 28 - Display jobs page screen

This is the add job page that shows the add job form.

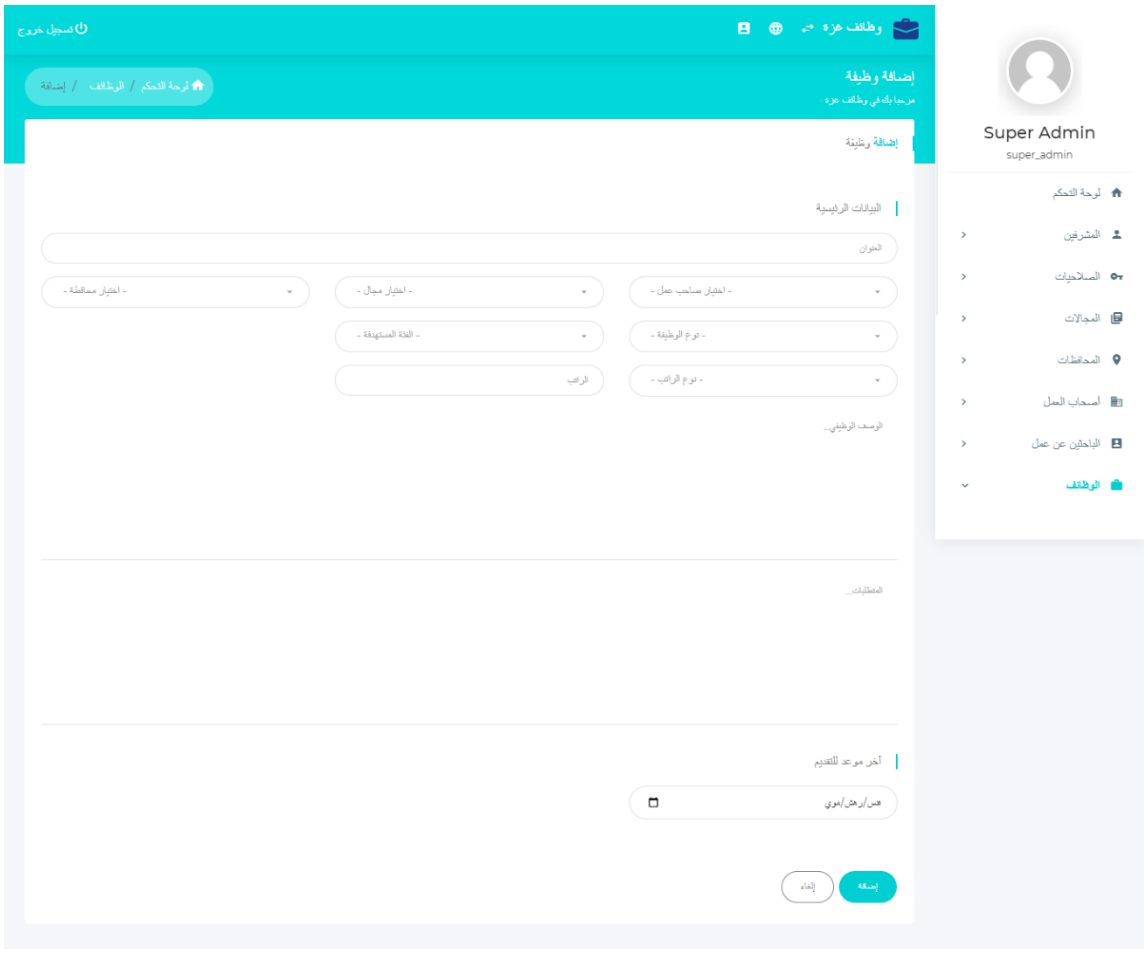


Figure 29 - Add job page screen

Chapter 7

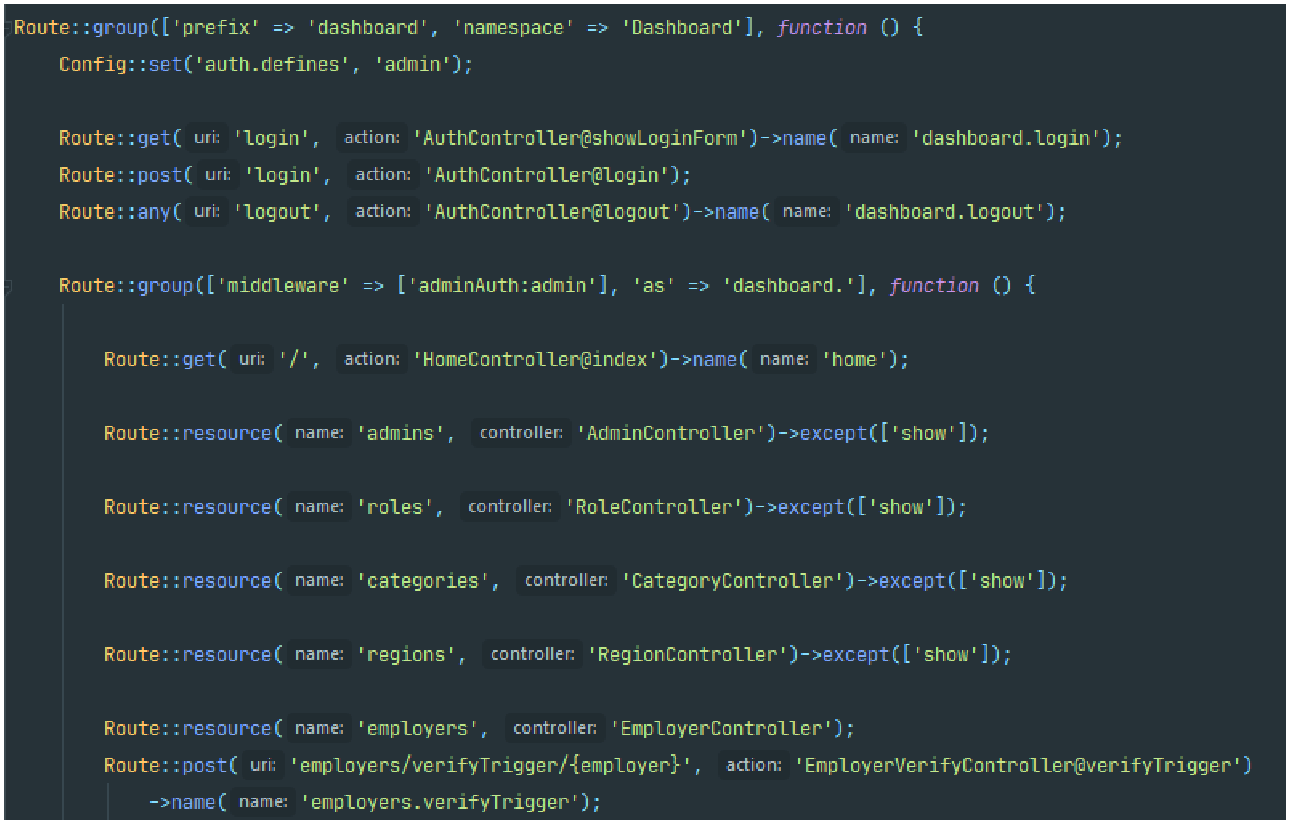
Implementation

Chapter7
Implementation

To implement this system, we use MySQL database and Laravel 7 for the back-end and for the front-end we use Bootstrap 4 in the dashboard and Tailwind CSS in the user pages, in this section we will show a snapshot from the code:

7.1 Route:

for the route we use group function to share the attributes such as “middleware” or “namespaces”, without defining these attributes on each individual route,
and also we use resource routing that assigns the typical create, read, update, and delete ("CRUD") routes to a controller with a single line of code.

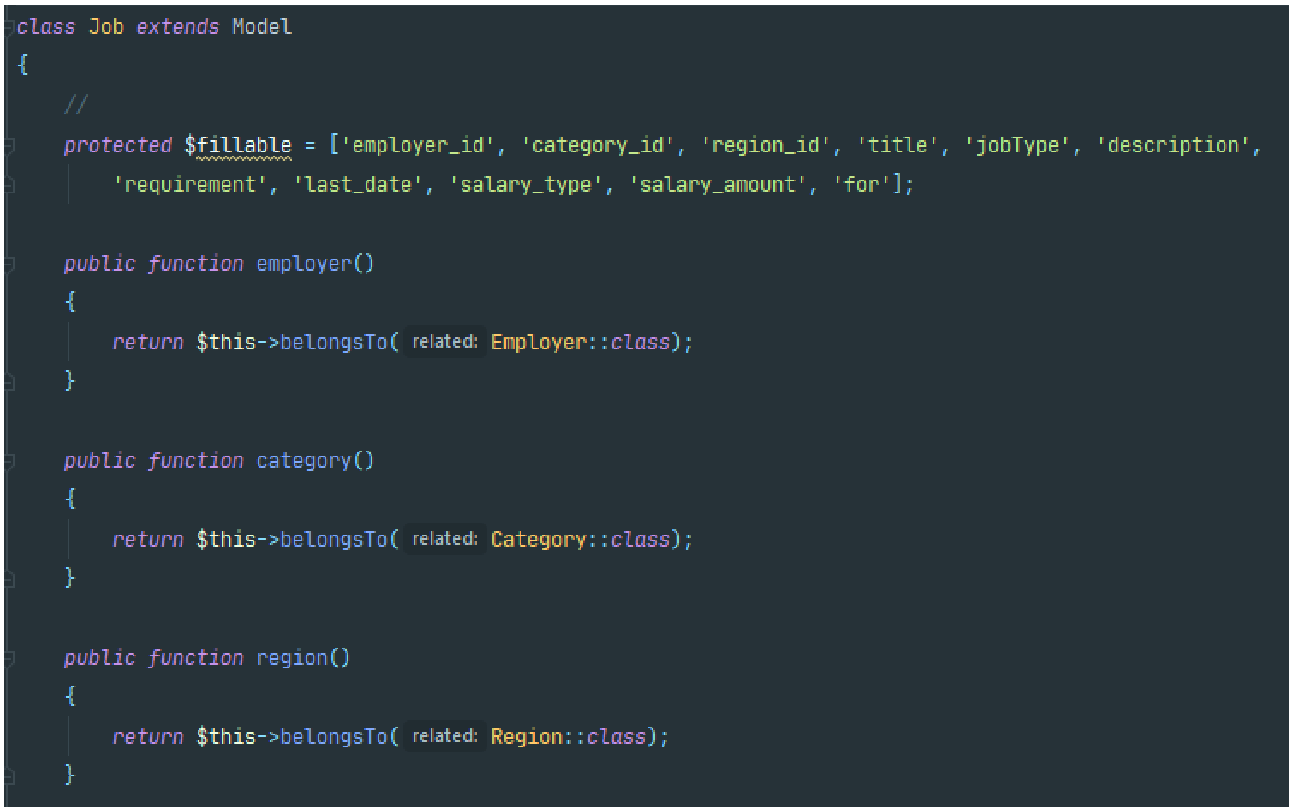


```
Route::group(['prefix' => 'dashboard', 'namespace' => 'Dashboard'], function () {
    Config::set('auth.defines', 'admin');

    Route::get( uri: 'login',  action: 'AuthController@showLoginForm')->name( name: 'dashboard.login');
    Route::post( uri: 'login',  action: 'AuthController@login');
    Route::any( uri: 'logout',  action: 'AuthController@logout')->name( name: 'dashboard.logout');

    Route::group(['middleware' => ['adminAuth:admin'], 'as' => 'dashboard.'], function () {

        Route::get( uri: '/',  action: 'HomeController@index')->name( name: 'home');

        Route::resource( name: 'admins',  controller: 'AdminController')->except(['show']);

        Route::resource( name: 'roles',  controller: 'RoleController')->except(['show']);

        Route::resource( name: 'categories',  controller: 'CategoryController')->except(['show']);

        Route::resource( name: 'regions',  controller: 'RegionController')->except(['show']);

        Route::resource( name: 'employers',  controller: 'EmployerController');
        Route::post( uri: 'employers/verifyTrigger/{employer}',  action: 'EmployerVerifyController@verifyTrigger')
            ->name( name: 'employers.verifyTrigger');
```

Figure 30 - Implementation Route

7.2 Model:

In the model we use the “$fillable” attribute that will containing all those fields of table which can be filled using mass-assignment, And also we use the “Eloquent relationships” functions that will provides powerful method chaining and querying capabilities.



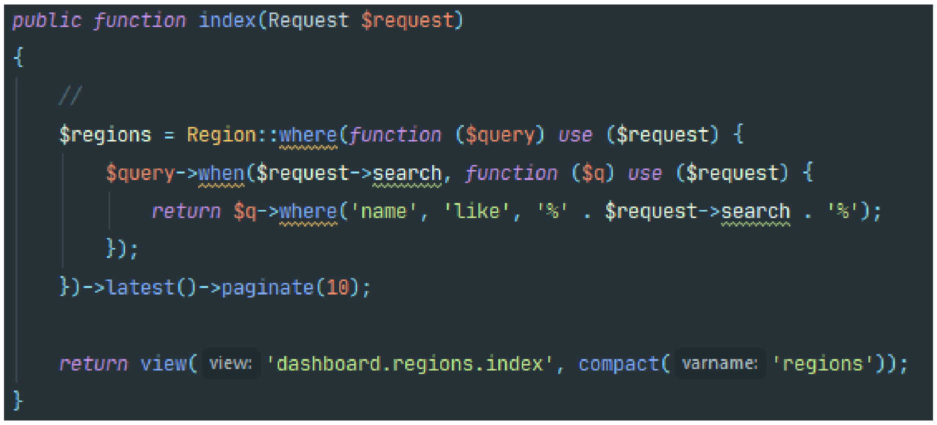
```
class Job extends Model
{
    //
    protected $fillable = ['employer_id', 'category_id', 'region_id', 'title', 'jobType', 'description',
        'requirement', 'last_date', 'salary_type', 'salary_amount', 'for'];

    public function employer()
    {
        return $this->belongsTo( related: Employer::class);
    }

    public function category()
    {
        return $this->belongsTo( related: Category::class);
    }

    public function region()
    {
        return $this->belongsTo( related: Region::class);
    }
```

Figure 31 - Implementation Model

7.3 Controller:

7.3.1 Index method:

In the index method we using “Laravel Eloquent” to retrieve the data from the database, And for filtering the data we use “where” and “when” function that will search about the data that will match the specific condition.

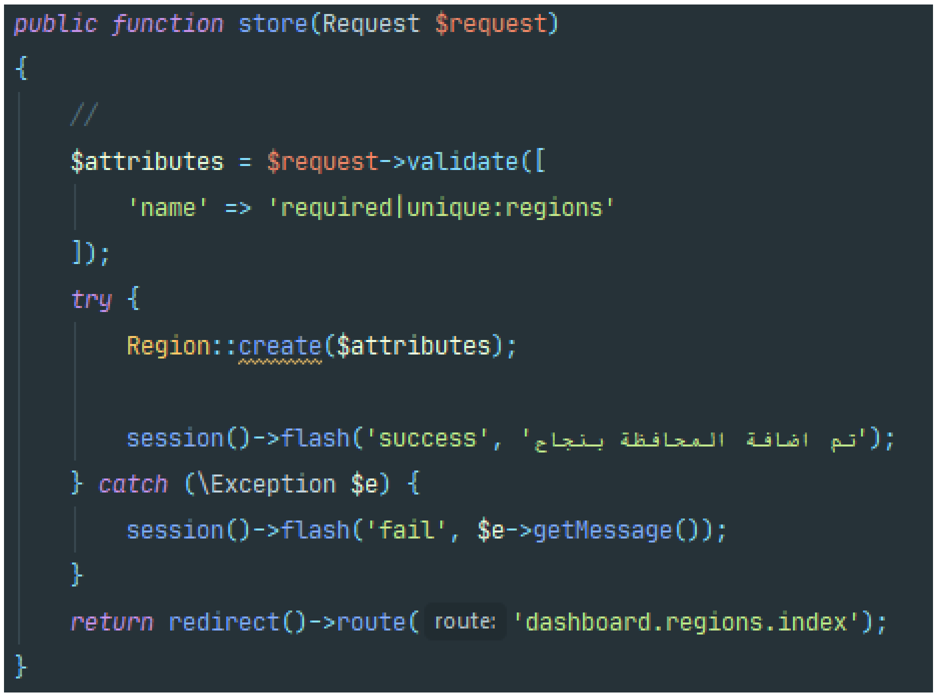


```php
public function index(Request $request)
{
    //
    $regions = Region::where(function ($query) use ($request) {
        $query->when($request->search, function ($q) use ($request) {
            return $q->where('name', 'like', '%' . $request->search . '%');
        });
    })->latest()->paginate(10);

    return view( view: 'dashboard.regions.index', compact( varname: 'regions'));
}
```

Figure 32 - Implementation Controller Index Method

7.3.2 Store method:

In the store method we first validate the request data and after that, we will store this data in the database and return success message if it succeed and fail messages if it failed.

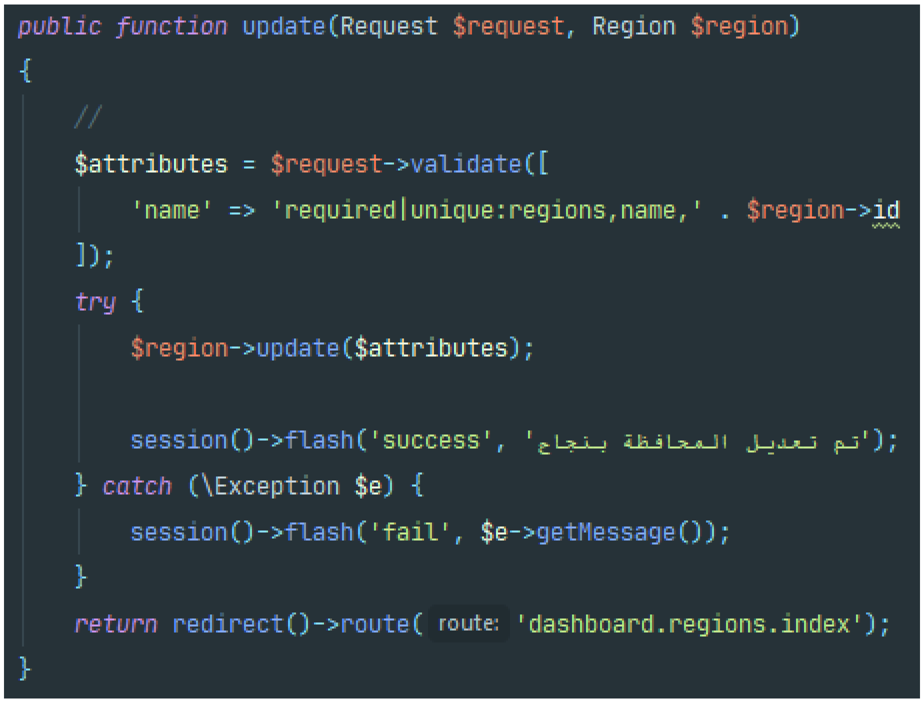


```php
public function store(Request $request)
{
    //
    $attributes = $request->validate([
        'name' => 'required|unique:regions'
    ]);
    try {
        Region::create($attributes);

        session()->flash('success', 'تم اضافة المحافظة بنجاح');
    } catch (\Exception $e) {
        session()->flash('fail', $e->getMessage());
    }
    return redirect()->route( route: 'dashboard.regions.index');
}
```

Figure 33 - Implementaion Controller Store Method

7.3.3 Update method:

In the update method we first validate the request data, and after that we will get the model instance from the “route binding” and update the data in the database.

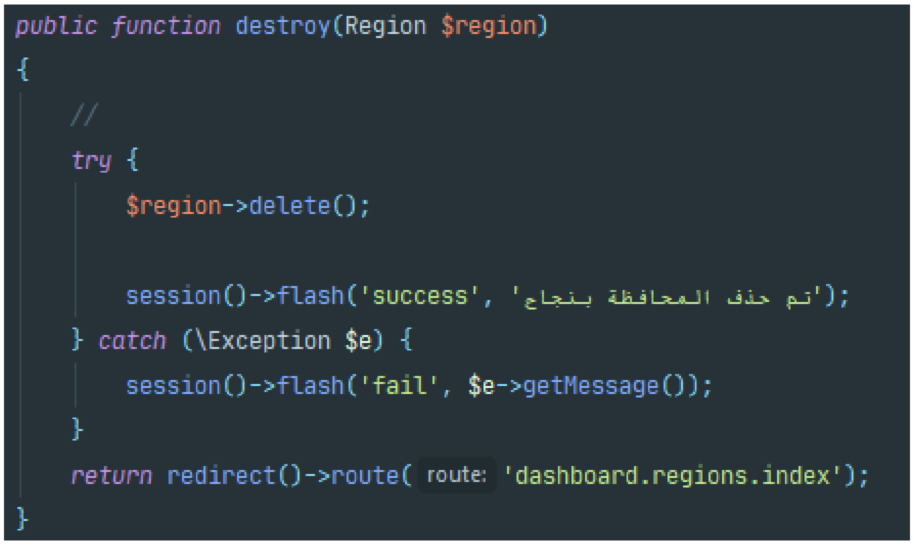


```
public function update(Request $request, Region $region)
{
    //
    $attributes = $request->validate([
        'name' => 'required|unique:regions,name,' . $region->id
    ]);
    try {
        $region->update($attributes);

        session()->flash('success', 'تم تعديل المحافظة بنجاح');
    } catch (\Exception $e) {
        session()->flash('fail', $e->getMessage());
    }
    return redirect()->route( route: 'dashboard.regions.index');
}
```

Figure 34 - Implementation Controller Update Method

7.3.4 Delete method:

In the delete method we will get the model instance from the “route binding” and delete the data from the database.



```
public function destroy(Region $region)
{
    //
    try {
        $region->delete();

        session()->flash('success', 'تم حذف المحافظة بنجاح');
    } catch (\Exception $e) {
        session()->flash('fail', $e->getMessage());
    }
    return redirect()->route( route: 'dashboard.regions.index');
}
```

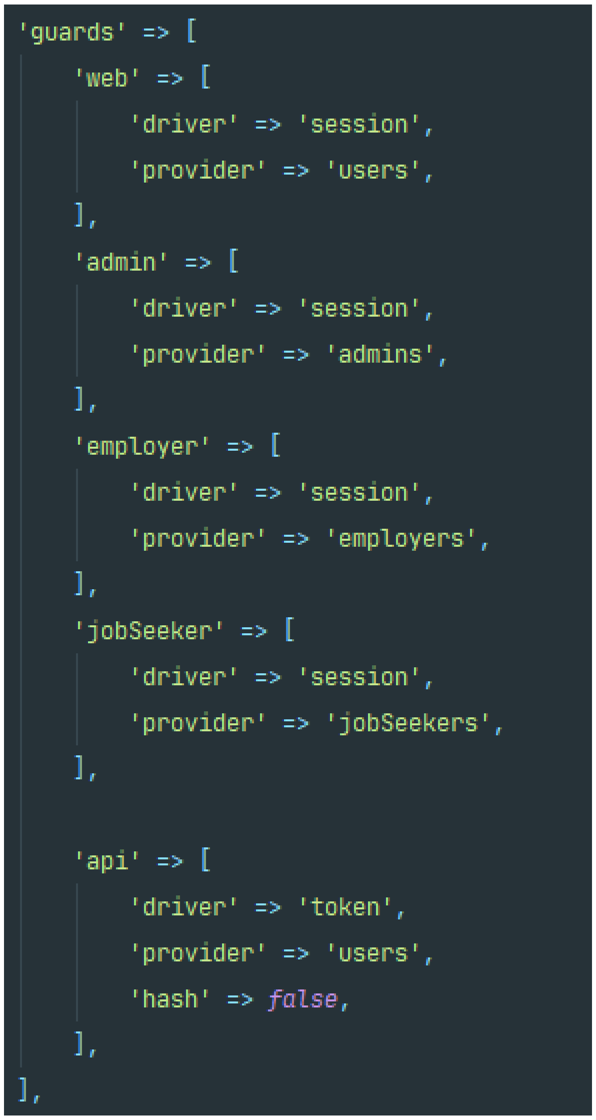
Figure 35 - Implementation Controller Destroy Method

7.4 Authentication:

7.4.1 Multi Guards:

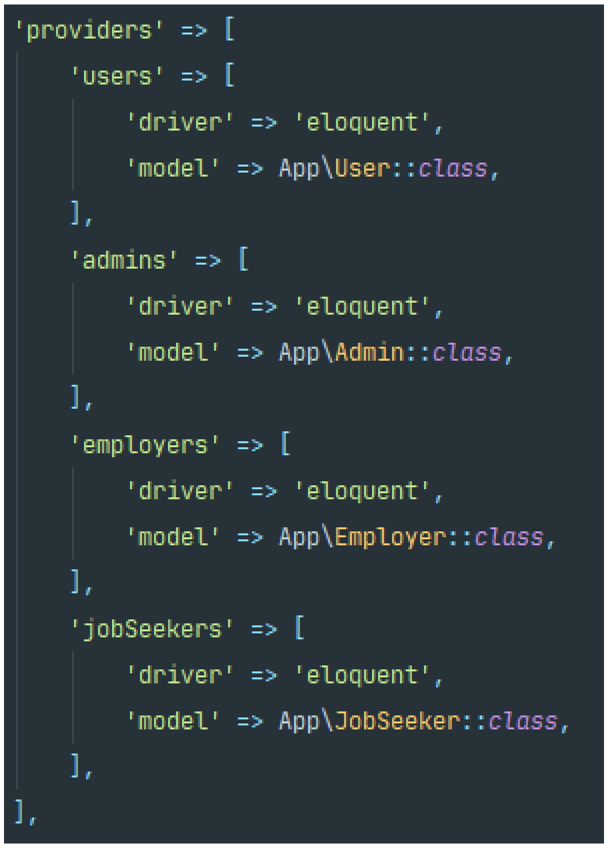
Laravel's authentication facilities are made up of “guards” and “providers”.
Guards define how users are authenticated for each request.
For example, Laravel ships with a session guard which maintains state using session storage and cookies. Providers define how users are retrieved from your persistent storage.

In the auth config file we define each guard in the system in the “guards” and “providers” array.



```
'guards' => [
    'web' => [
        'driver' => 'session',
        'provider' => 'users',
    ],
    'admin' => [
        'driver' => 'session',
        'provider' => 'admins',
    ],
    'employer' => [
        'driver' => 'session',
        'provider' => 'employers',
    ],
    'jobSeeker' => [
        'driver' => 'session',
        'provider' => 'jobSeekers',
    ],

    'api' => [
        'driver' => 'token',
        'provider' => 'users',
        'hash' => false,
    ],
],
```

Figure 37 - Implementation Authentication Guard



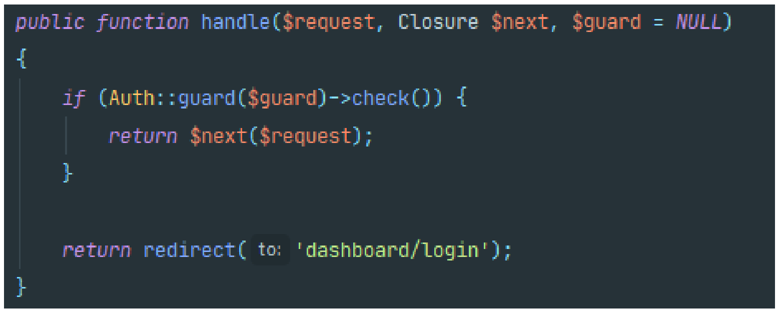
```
'providers' => [
    'users' => [
        'driver' => 'eloquent',
        'model' => App\User::class,
    ],
    'admins' => [
        'driver' => 'eloquent',
        'model' => App\Admin::class,
    ],
    'employers' => [
        'driver' => 'eloquent',
        'model' => App\Employer::class,
    ],
    'jobSeekers' => [
        'driver' => 'eloquent',
        'model' => App\JobSeeker::class,
    ],
],
```

Figure 36 - Implementation Authentication Providers

7.4.2 Auth Middleware:

Laravel includes a middleware that verifies the user of your application is authenticated. If the user is not authenticated, the middleware will redirect the user to the login screen. However, if the user is authenticated, the middleware will allow the request to proceed further.

This is the Auth middleware for the Admin.

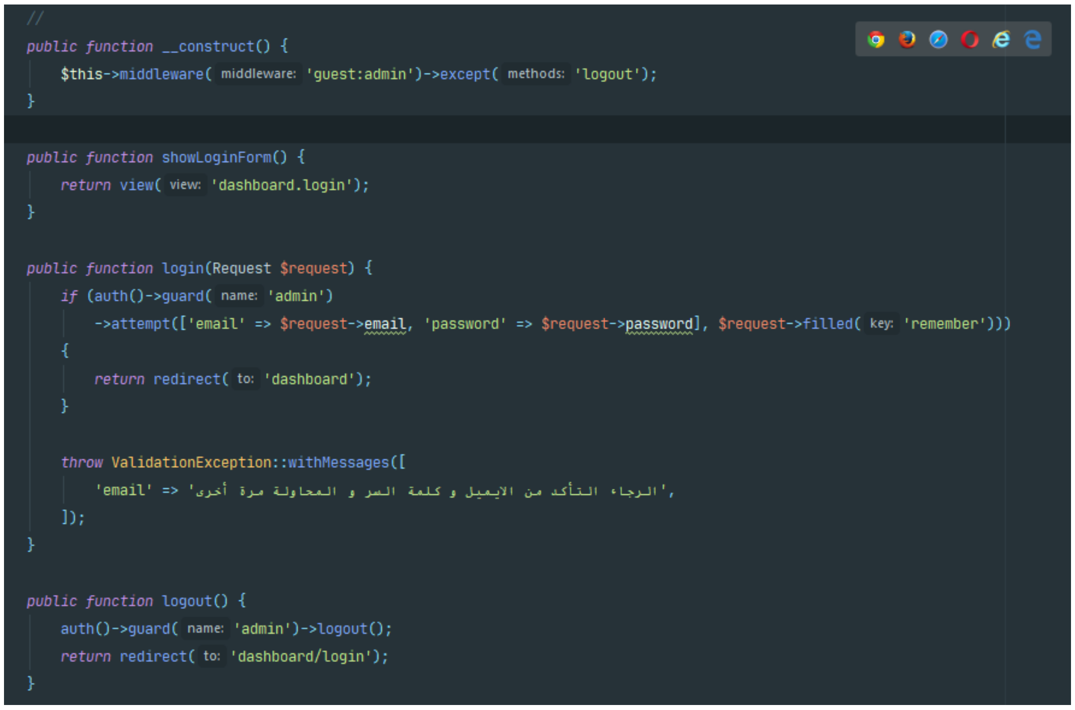


```
public function handle($request, Closure $next, $guard = NULL)
{
    if (Auth::guard($guard)->check()) {
        return $next($request);
    }

    return redirect( to: 'dashboard/login');
}
```

Figure 38 - Implementation Authentication Middleware

7.4.3 Auth Controller:

In the controller we set a “construct” function that will run a middleware that will check if the admin is not authenticate in all methods except logout method, And in login method we get the auth from the admin guard then check the credentials and if it match he will login and if not an “ValidationException” will be throw, and in logout method it will get the auth from the guard admin then logout him.



```
//
public function __construct() {
    $this->middleware( middleware: 'guest:admin')->except( methods: 'logout');
}

public function showLoginForm() {
    return view( view: 'dashboard.login');
}

public function login(Request $request) {
    if (auth()->guard( name: 'admin')
        ->attempt(['email' => $request->email, 'password' => $request->password], $request->filled( key: 'remember')))
    {
        return redirect( to: 'dashboard');
    }

    throw ValidationException::withMessages([
        'email' => 'الرجاء التأكد من الايميل و كلمة السر و المحاولة مرة أخرى',
    ]);
}

public function logout() {
    auth()->guard( name: 'admin')->logout();
    return redirect( to: 'dashboard/login');
}
```

Figure 39 - Implementation Authentication Controller

7.5 Tools and equipment's:

- **Front-end Development**
    - Bootstrap [10]
    - Tailwind CSS [11]
    - jQuery [12]
    - Alpine.js [13]

- **Back-end Development**
    - PHP Storm [14]
    - XAMPP [15]
    - Laravel v7. * [16]

- **Design**
    - Visual paradigm for UML 10.0 [17]
    - SequenceDiagram.org [18]
    - Laravel Schema Designer [19]

Chapter 8
Testing

Chapter8
Testing

8.1 Unit Test:

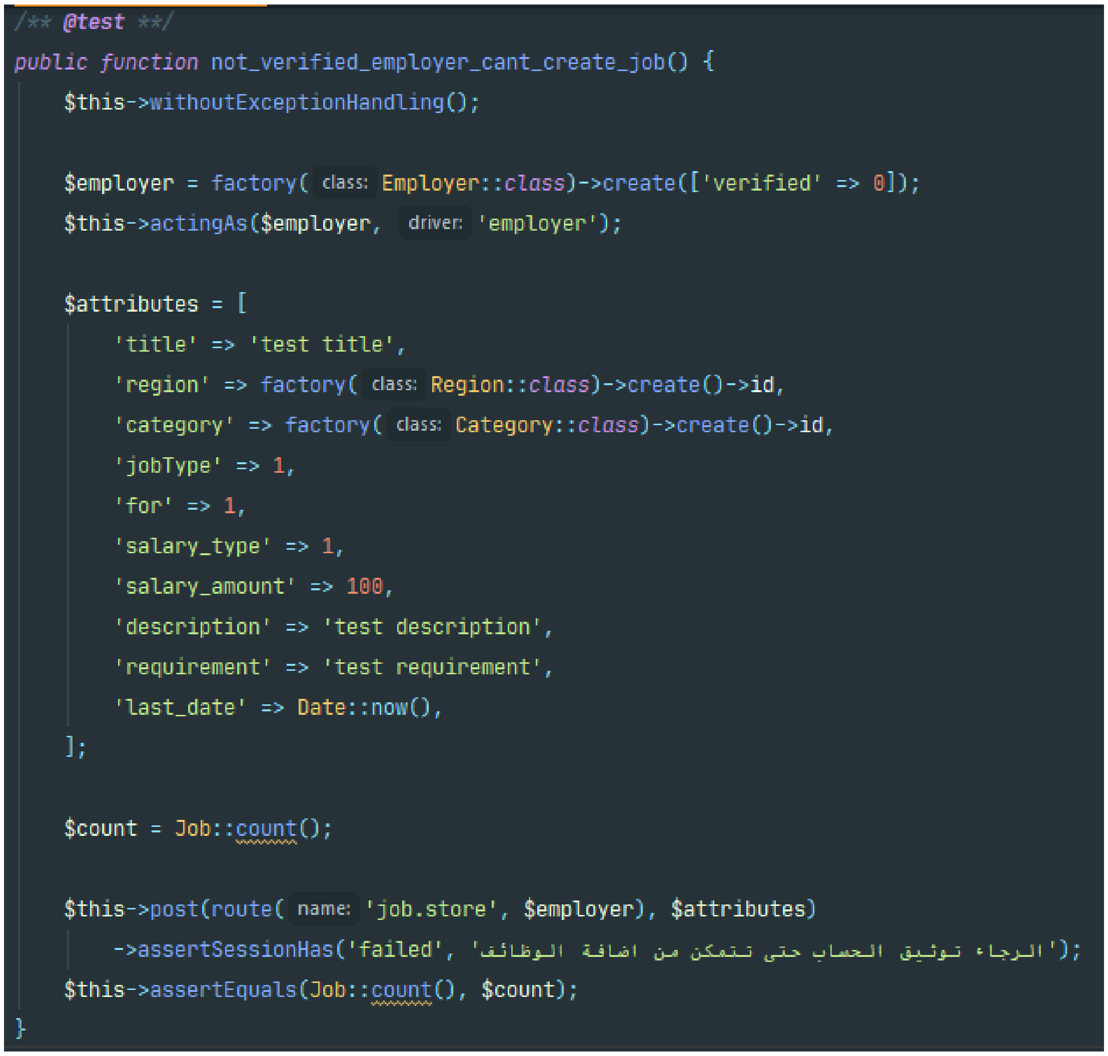
Unit testing is a software testing method in which a programmer tests if individual units of source code are fit for use.

Unit tests are conducted right after units are completed to ensure that all units work well. Using PHPUnit, we run a set of test cases for some units in the system.

Here are some of unit testing examples:

8.1.1 not_verified_employer_cant_create_job:

This test was conducted to determine that a not verified employer can't post a job, and it will return a failed message.

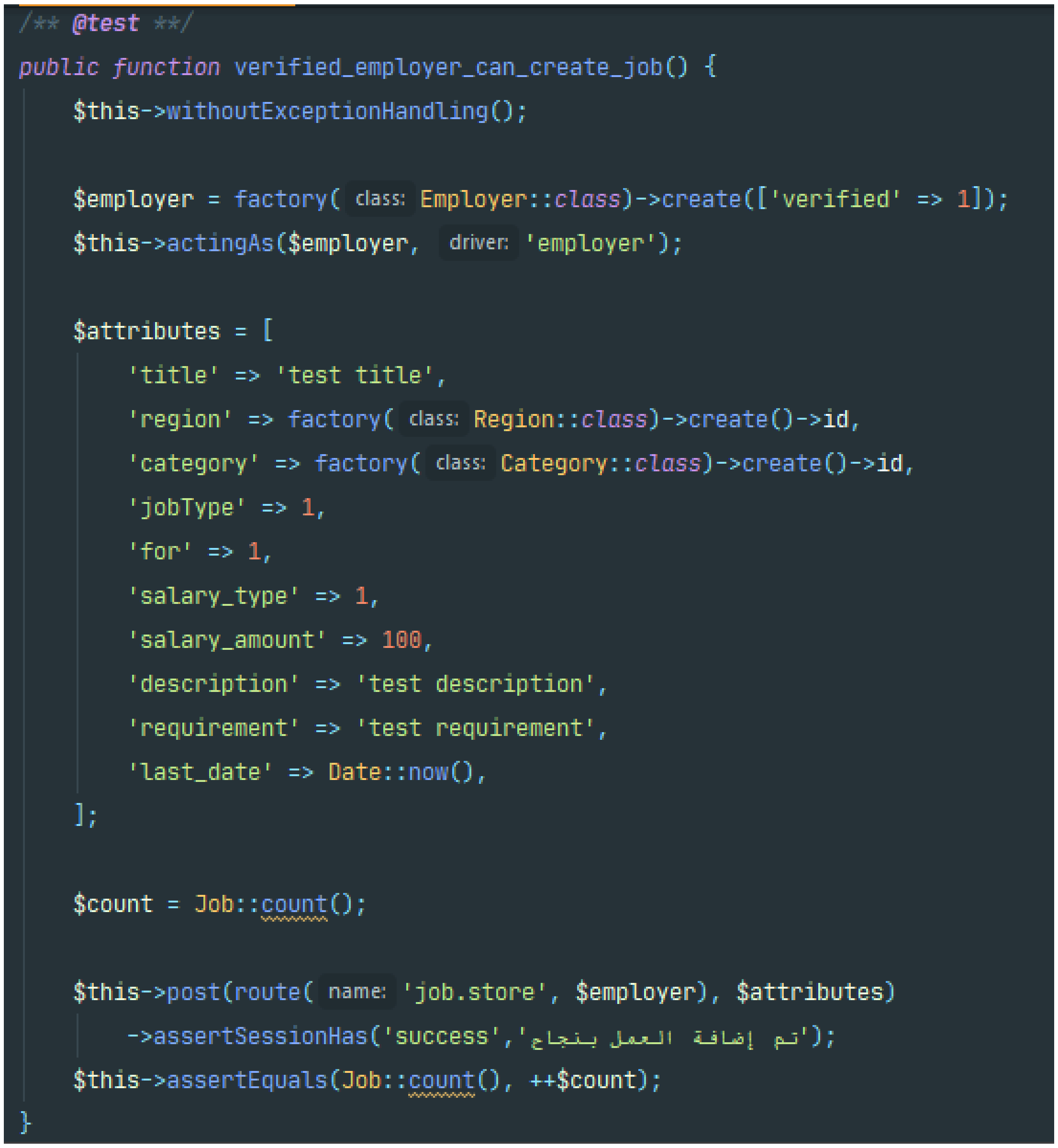


```
/** @test **/
public function not_verified_employer_cant_create_job() {
    $this->withoutExceptionHandling();

    $employer = factory( class: Employer::class)->create(['verified' => 0]);
    $this->actingAs($employer,  driver: 'employer');

    $attributes = [
        'title' => 'test title',
        'region' => factory( class: Region::class)->create()->id,
        'category' => factory( class: Category::class)->create()->id,
        'jobType' => 1,
        'for' => 1,
        'salary_type' => 1,
        'salary_amount' => 100,
        'description' => 'test description',
        'requirement' => 'test requirement',
        'last_date' => Date::now(),
    ];

    $count = Job::count();

    $this->post(route( name: 'job.store', $employer), $attributes)
        ->assertSessionHas('failed', 'الرجاء توثيق الحساب حتى تتمكن من اضافة الوظائف');
    $this->assertEquals(Job::count(), $count);
}
```

Figure 40 - not_verified_employer_cant_create_job

8.1.2 verified_employer_can_create_job:

This test was conducted to determine that a verified employer can post a job and return a success message.

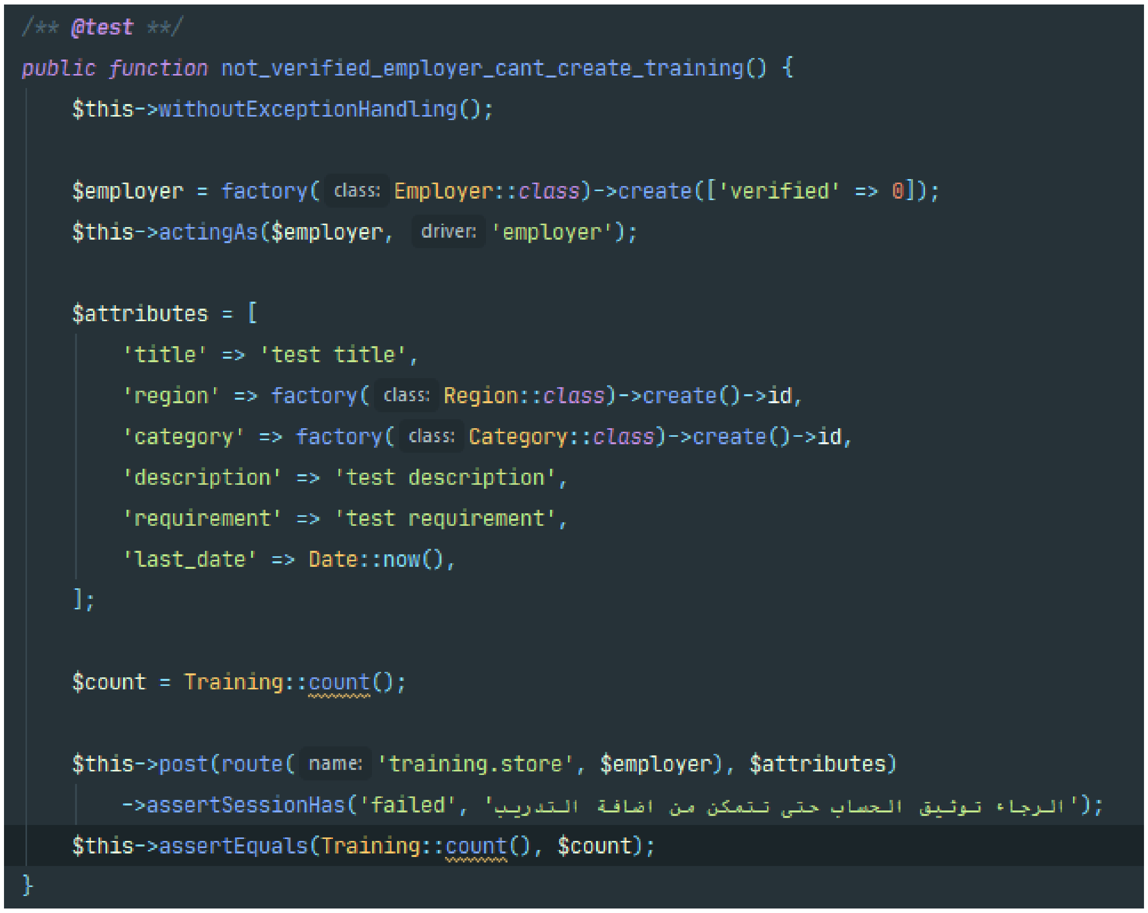


```
/** @test **/
public function verified_employer_can_create_job() {
    $this->withoutExceptionHandling();

    $employer = factory( class: Employer::class)->create(['verified' => 1]);
    $this->actingAs($employer,  driver: 'employer');

    $attributes = [
        'title' => 'test title',
        'region' => factory( class: Region::class)->create()->id,
        'category' => factory( class: Category::class)->create()->id,
        'jobType' => 1,
        'for' => 1,
        'salary_type' => 1,
        'salary_amount' => 100,
        'description' => 'test description',
        'requirement' => 'test requirement',
        'last_date' => Date::now(),
    ];

    $count = Job::count();

    $this->post(route( name: 'job.store', $employer), $attributes)
        ->assertSessionHas('success','تم إضافة العمل بنجاح');
    $this->assertEquals(Job::count(), ++$count);
}
```

Figure 41 - verified_employer_can_create_job

8.1.3 not_verified_employer_cant_create_training:

This test was conducted to determine that a not verified employer can't post a training, and it will return a failed message.

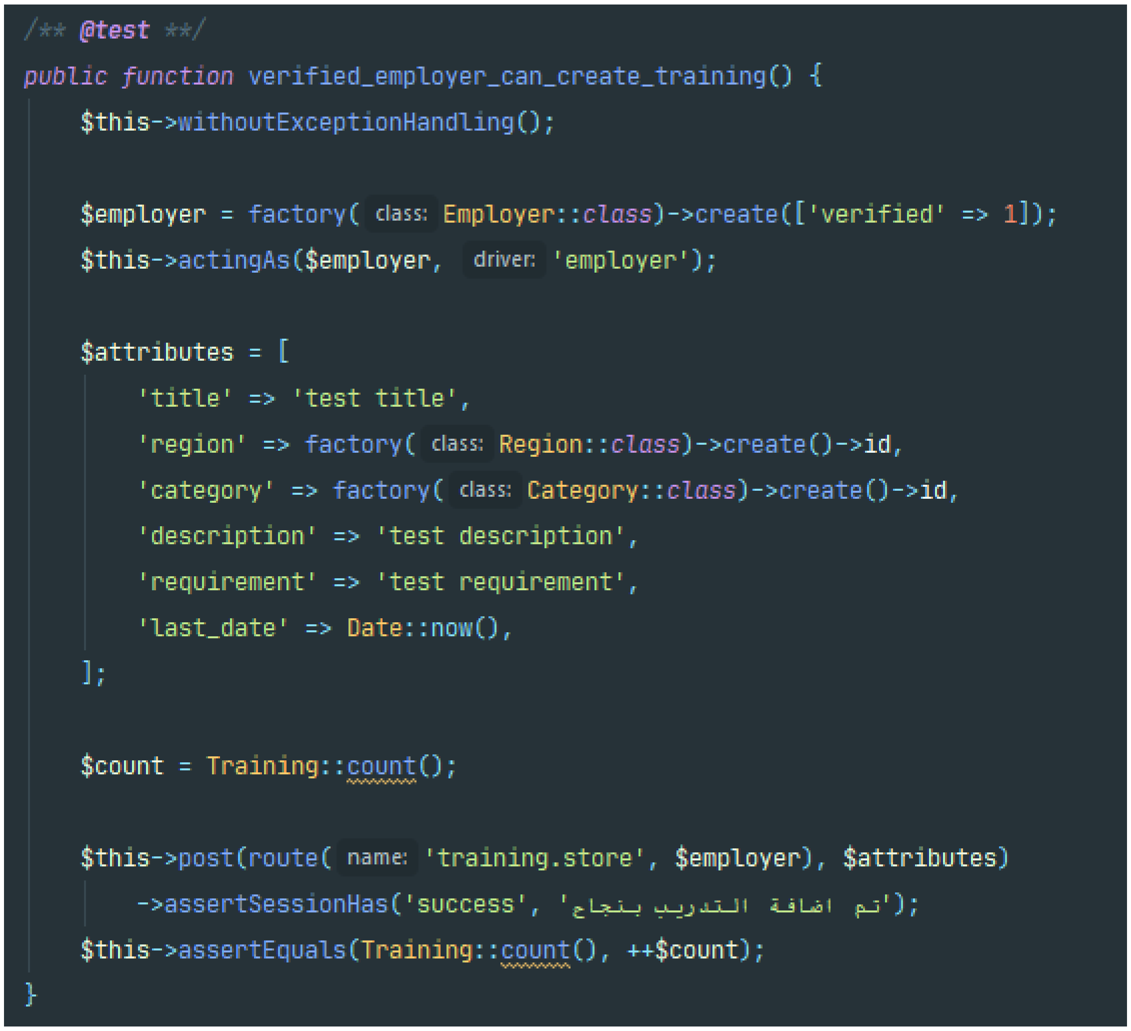


```php
/** @test **/
public function not_verified_employer_cant_create_training() {
    $this->withoutExceptionHandling();

    $employer = factory( class: Employer::class)->create(['verified' => 0]);
    $this->actingAs($employer,  driver: 'employer');

    $attributes = [
        'title' => 'test title',
        'region' => factory( class: Region::class)->create()->id,
        'category' => factory( class: Category::class)->create()->id,
        'description' => 'test description',
        'requirement' => 'test requirement',
        'last_date' => Date::now(),
    ];

    $count = Training::count();

    $this->post(route( name: 'training.store', $employer), $attributes)
        ->assertSessionHas('failed', 'الرجاء توثيق الحساب حتى تتمكن من اضافة التدريب');
    $this->assertEquals(Training::count(), $count);
}
```

Figure 42 - not_verified_employer_cant_create_training

8.1.4 verified_employer_can_create_training:

This test was conducted to determine that a verified employer can post a training, and it will return a success message.

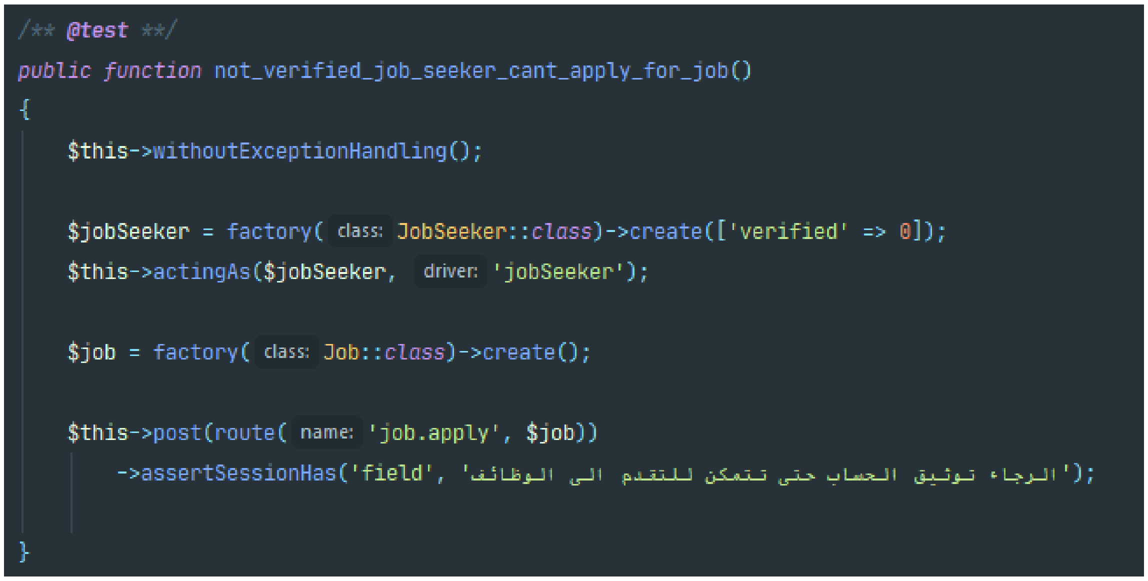


```
/** @test **/
public function verified_employer_can_create_training() {
    $this->withoutExceptionHandling();

    $employer = factory( class: Employer::class)->create(['verified' => 1]);
    $this->actingAs($employer,  driver: 'employer');

    $attributes = [
        'title' => 'test title',
        'region' => factory( class: Region::class)->create()->id,
        'category' => factory( class: Category::class)->create()->id,
        'description' => 'test description',
        'requirement' => 'test requirement',
        'last_date' => Date::now(),
    ];

    $count = Training::count();

    $this->post(route( name: 'training.store', $employer), $attributes)
        ->assertSessionHas('success', 'تم اضافة التدريب بنجاح');
    $this->assertEquals(Training::count(), ++$count);
}
```

Figure 43 - verified_employer_can_create_training

8.1.5 not_verified_job_seeker_cant_apply_for_job:

This test was conducted to determine that a not verified job seeker can’t apply to a job, and it will return a failed message.

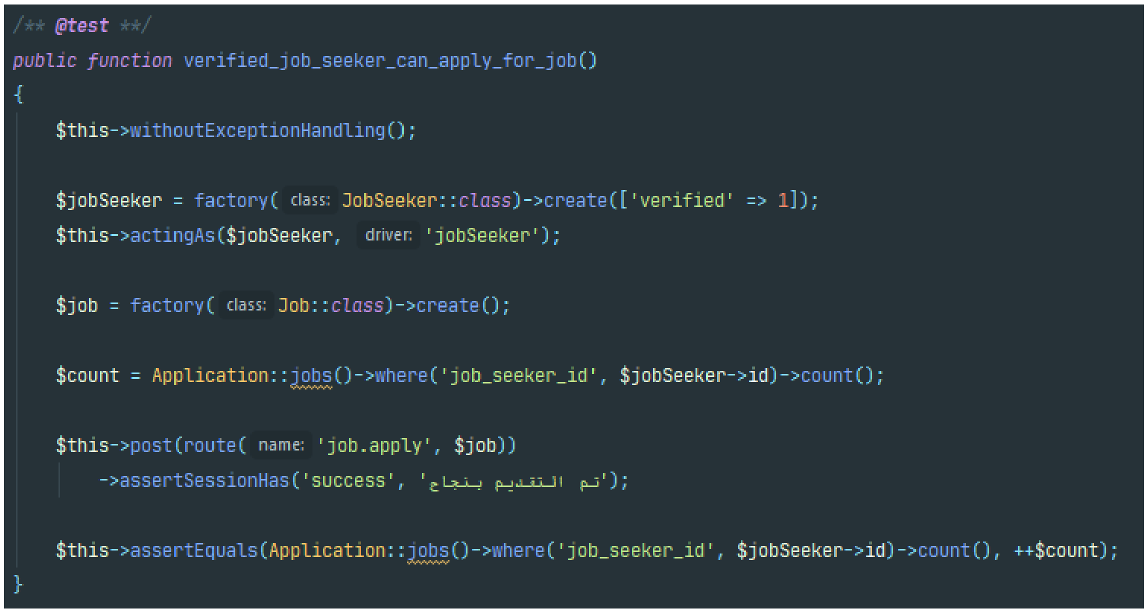


```php
/** @test **/
public function not_verified_job_seeker_cant_apply_for_job()
{
    $this->withoutExceptionHandling();

    $jobSeeker = factory( class: JobSeeker::class)->create(['verified' => 0]);
    $this->actingAs($jobSeeker,  driver: 'jobSeeker');

    $job = factory( class: Job::class)->create();

    $this->post(route( name: 'job.apply', $job))
        ->assertSessionHas('field', 'الرجاء توثيق الحساب حتى تتمكن للتقدم الى الوظائف');

}
```

Figure 44 - not_verified_job_seeker_cant_apply_for_job

8.1.6 verified_job_seeker_can_apply_for_job:

This test was conducted to determine that a verified job seeker can apply to a job, and it will return a success message.

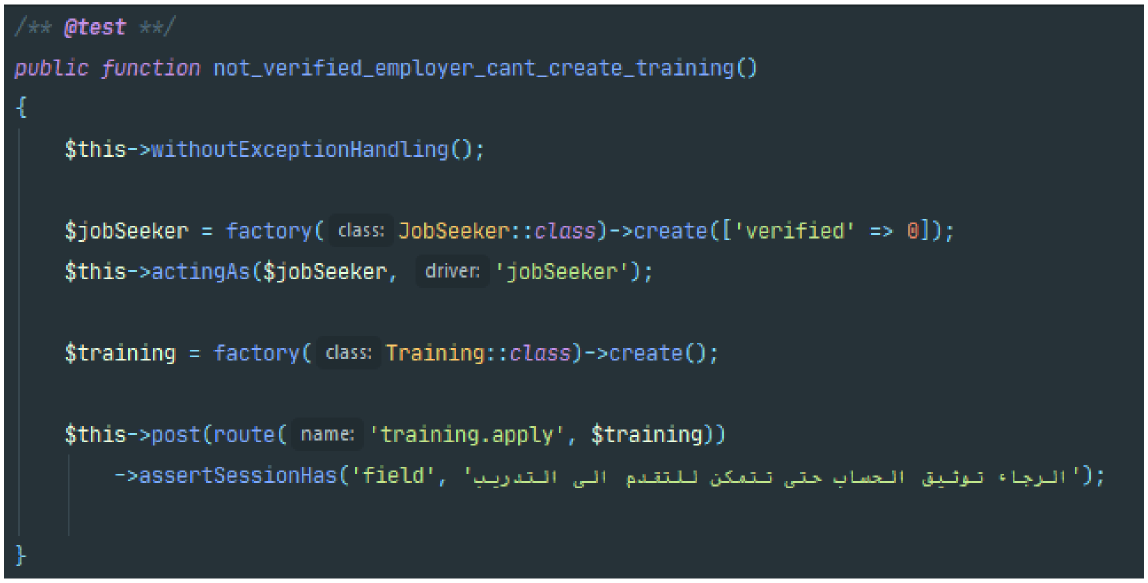


```php
/** @test **/
public function verified_job_seeker_can_apply_for_job()
{
    $this->withoutExceptionHandling();

    $jobSeeker = factory( class: JobSeeker::class)->create(['verified' => 1]);
    $this->actingAs($jobSeeker,  driver: 'jobSeeker');

    $job = factory( class: Job::class)->create();

    $count = Application::jobs()->where('job_seeker_id', $jobSeeker->id)->count();

    $this->post(route( name: 'job.apply', $job))
        ->assertSessionHas('success', 'تم التقديم بنجاح');

    $this->assertEquals(Application::jobs()->where('job_seeker_id', $jobSeeker->id)->count(), ++$count);
}
```

Figure 45 - verified_job_seeker_can_apply_for_job

8.1.7 not_verified_employer_cant_create_training:

This test was conducted to determine that a not verified job seeker can't apply to a training, and it will return a failed message.

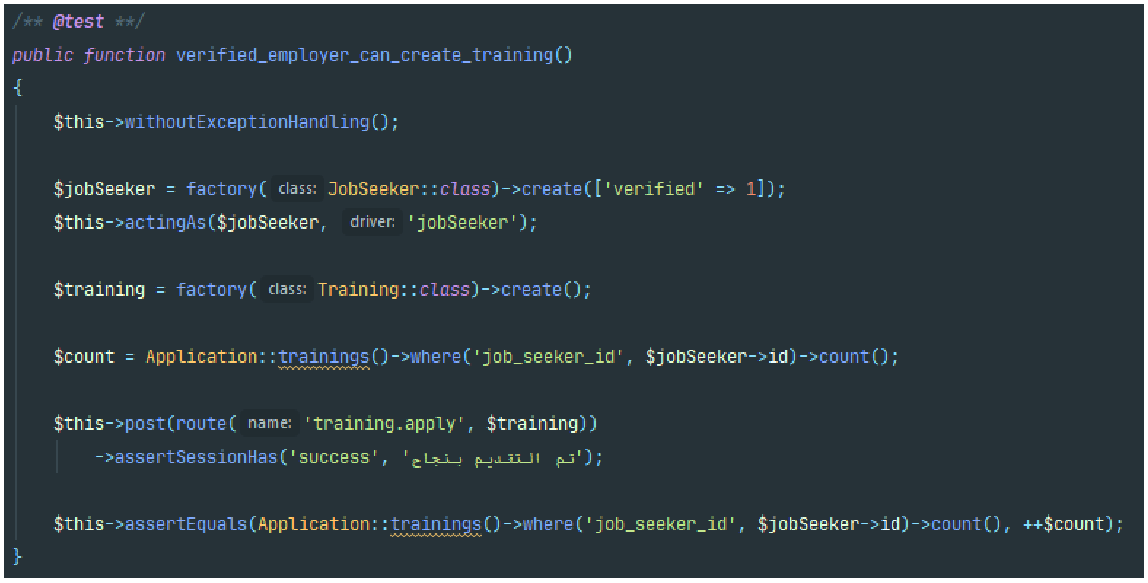


```
/** @test **/
public function not_verified_employer_cant_create_training()
{
    $this->withoutExceptionHandling();

    $jobSeeker = factory( class: JobSeeker::class)->create(['verified' => 0]);
    $this->actingAs($jobSeeker,  driver: 'jobSeeker');

    $training = factory( class: Training::class)->create();

    $this->post(route( name: 'training.apply', $training))
        ->assertSessionHas('field', 'الرجاء توثيق الحساب حتى تتمكن للتقدم الى التدريب');

}
```

Figure 46 - not_verified_employer_cant_create_training

8.1.8 verified_employer_can_create_training:

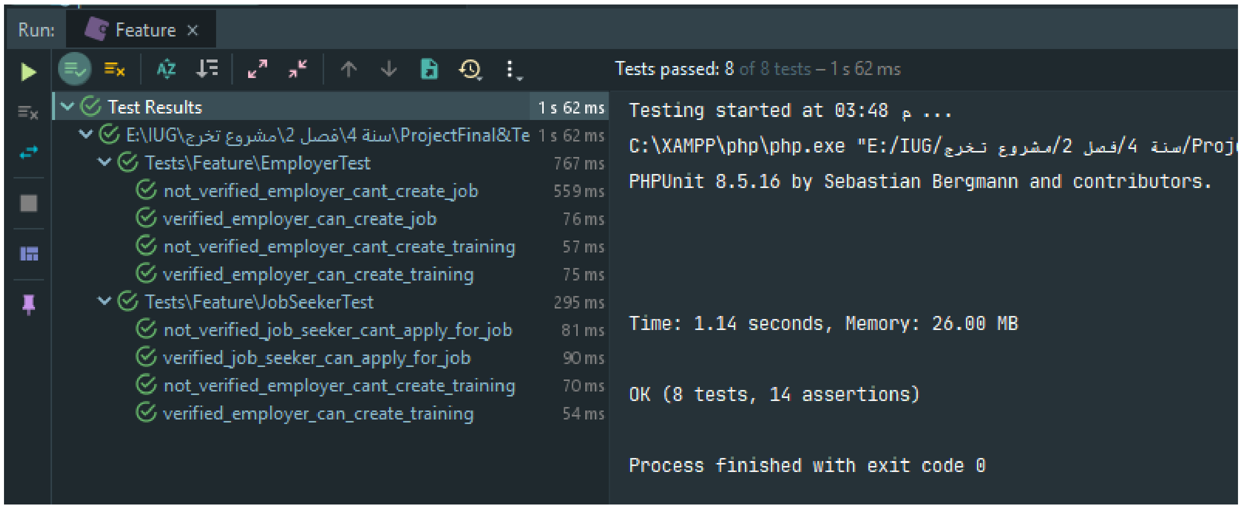
This test was conducted to determine that a verified job seeker can apply to a training, and it will return a success message.



```
/** @test **/
public function verified_employer_can_create_training()
{
    $this->withoutExceptionHandling();

    $jobSeeker = factory( class: JobSeeker::class)->create(['verified' => 1]);
    $this->actingAs($jobSeeker,  driver: 'jobSeeker');

    $training = factory( class: Training::class)->create();

    $count = Application::trainings()->where('job_seeker_id', $jobSeeker->id)->count();

    $this->post(route( name: 'training.apply', $training))
        ->assertSessionHas('success', 'تم التقديم بنجاح');

    $this->assertEquals(Application::trainings()->where('job_seeker_id', $jobSeeker->id)->count(), ++$count);
}
```

Figure 47 - verified_employer_can_create_training

8.1.9 Unit Test Results:



```
Tests passed: 8 of 8 tests – 1 s 62 ms

Test Results                                          1 s 62 ms
  E:\IUG\مشروع تخرج\فصل 2\سنة 4\ProjectFinal&Te       1 s 62 ms
    Tests\Feature\EmployerTest                          767 ms
      not_verified_employer_cant_create_job             559 ms
      verified_employer_can_create_job                   76 ms
      not_verified_employer_cant_create_training         57 ms
      verified_employer_can_create_training              75 ms
    Tests\Feature\JobSeekerTest                         295 ms
      not_verified_job_seeker_cant_apply_for_job         81 ms
      verified_job_seeker_can_apply_for_job              90 ms
      not_verified_employer_cant_create_training         70 ms
      verified_employer_can_create_training              54 ms
```

```
Testing started at 03:48 م ...
C:\XAMPP\php\php.exe "E:/IUG/مشروع تخرج/فصل 2/سنة 4/Proj
PHPUnit 8.5.16 by Sebastian Bergmann and contributors.



Time: 1.14 seconds, Memory: 26.00 MB

OK (8 tests, 14 assertions)

Process finished with exit code 0
```

Figure 48 - Unit Test Results

8.2 Usability Test:

After we are shown this project to some students, we got this feedback from them.

- I found the job search filter very convenient.
- I think I would definitely use this website in the purpose of finding a job.
- As someone who is looking for training, I think there should be more communication with the companies offering training.
- As a student looking for a training chance, I found this website very helpful and it would help me and every student looking for a training chance.
- I would imagine that most people would learn to use this website very quickly.

Chapter 9
Conclusions and Future Work

Chapter 9
Conclusions and Future Work

9.1 Conclusions:

We proposed Gaza Jobs which is a web-based application that allows students, graduates or people to find a job or training in a fast way, and also know more about the companies in their local.

We developed Gaza Jobs to be able to provide a platform that will help them to find the jobs and training, using PHP Laravel.

The system currently covers the Gaza Strip, and we hope that the system will be expanded to cover All Palestine and other countries.

9.2 Future Work:

In the future work we will add more features and we will continue improving to the current system, Like:

- Online Chat System.
- Real-Time Notification.
- Support Multi-languages.
- Making Contracts.
- Online Payment.
- Exams System for Jobs and Training Applications.

References

References

[1] Indeed Editorial Team, “7 Reasons Why it’s So Hard to Get a Job After College (With Helpful Tips),” 2021. https://www.indeed.com/career-advice/finding-a-job/why-is-it-so-hard-to-get-a-job-after-college (accessed Dec. 28, 2020).

[2] “Jobs.ps,” 2008. https://www.jobs.ps/ (accessed Oct. 28, 2020).

[3] “Indeed.com.” https://www.indeed.com/ (accessed Oct. 28, 2020).

[4] “linkedin.” https://www.linkedin.com/ (accessed Oct. 28, 2020).

[5] Smartsheet, “The Agile Software Development Lifecycle Explained,” *SmartSheet*, 2017. https://www.smartsheet.com/understanding-agile-software-development-lifecycle-and-process-workflow (accessed Jun. 21, 2021).

[6] W3Adda, “Laravel Models.” https://www.w3adda.com/laravel-tutorial/laravel-models (accessed Apr. 15, 2021).

[7] W3Adda, “Laravel Views.” https://www.w3adda.com/laravel-tutorial/laravel-views (accessed Apr. 15, 2021).

[8] W3Adda, “Laravel Controllers.” https://www.w3adda.com/laravel-tutorial/laravel-controllers (accessed Apr. 15, 2021).

[9] W3Adda, “Laravel Routing.” https://www.w3adda.com/laravel-tutorial/laravel-routing (accessed Apr. 15, 2021).

[10] “Bootstrap.” https://getbootstrap.com/ (accessed May 10, 2021).

[11] A. Wathan, “Tailwindcss.” https://tailwindcss.com/ (accessed May 10, 2021).

[12] “jQuery.” https://jquery.com (accessed Jun. 07, 2021).

[13] calebporzio, “Alpinejs,” 2019. https://github.com/alpinejs/alpine (accessed Jun. 07, 2021).

[14] JetBrains, “PhpStorm,” 2000. https://www.jetbrains.com/phpstorm/ (accessed May 10, 2021).

[15] “XAMPP.” https://www.apachefriends.org/index.html (accessed May 10, 2021).

[16] Taylor Otwell, “Laravel7,” 2021. https://laravel.com/docs/7.x/releases (accessed May 10, 2021).

[17] “Visual Paradigm 10.0,” 2012. https://www.visual-paradigm.com/tw/aboutus/newsreleases/vpuml100.jsp (accessed May 10, 2021).

[18] “Sequence Diagram.” https://sequencediagram.org/ (accessed May 10, 2021).

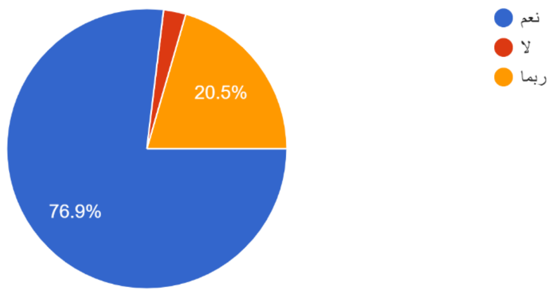
[19] “Laravel Schema Designer,” 2013. https://laravelsd.com/ (accessed Apr. 10, 2021).

Appendix 1: Survey Results

- **This question shows the percentage of job seekers that faced problems.**

هل تواجه مشاكل او عقبات في البحث عن عمل او تدريب ؟

39 responses



- **This question shows the problems that face the job Seekers.**

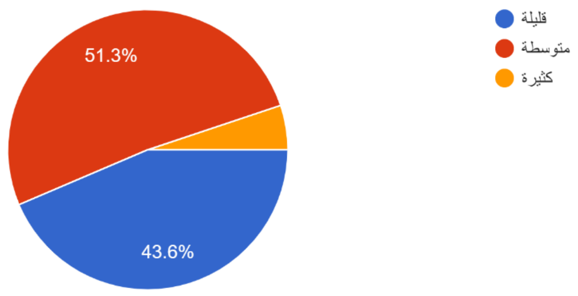
ما هي المشاكل التي تواجهها في عمليه البحث عن (عمل / تدريب)

39 responses

- **This question shows how much knowledge the job Seekers have about the companies that work in their field.**

ما مدى معرفتك بالشركات او المؤسسات المحلية التي تعمل في مجالك ؟

39 responses



- **This question shows the features needed by the job seekers that will help them.**

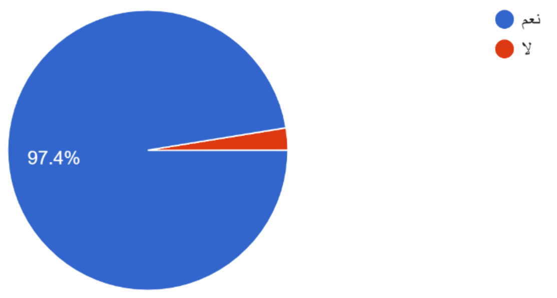
ما هي الخصائص والميزات التي تظن انها ستساعدك اذا وجدت في هذا الموقع ؟

39 responses

- **This question shows the percentage of job Seekers that see that this website will help them.**

لو كان هناك موقع مختص في نشر الوظائف التدريب هل تجده أمر سيساعدك ؟

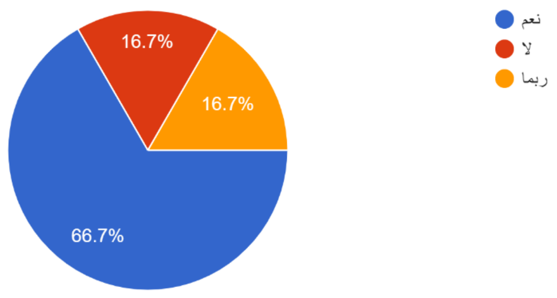
39 responses



- **This question shows the percentage of employers that faced problems when they post a job or training**

هل تواجه مشاكل او عقبات في نشر وظيفة او تدريب ؟

6 responses

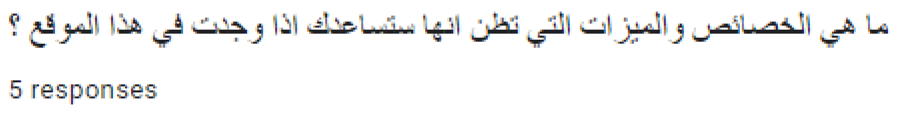


- **This question shows the reason of problems that faced employers when they post a job or training.**

ما هي الأسباب وراء هذه المشكلة ؟

5 responses

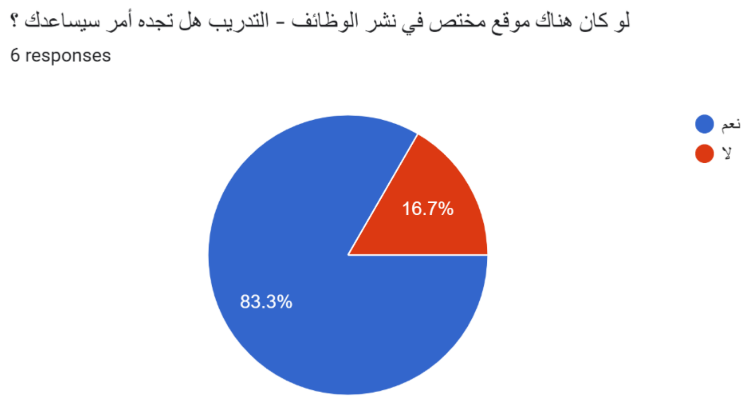
- **This question shows the features needed by the employers that will help them.**



ما هي الخصائص والميزات التي تظن انها ستساعدك اذا وجدت في هذا الموقع ؟

5 responses

- **This question shows the percentage of employers that see that this website will help them.**



لو كان هناك موقع مختص في نشر الوظائف - التدريب هل تجده أمر سيساعدك ؟

6 responses