NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

Nitte (DU) established under Section 3 of UGC Act 1956 | Accredited with 'A+' Grade by NAAC

Department of Computer Science & Engineering

Report on Mini Project

Online Job Portal

Course Code: CS2102-1

Course Name: **Database Management System**

Semester: IV SEM Section: A

Submitted To:

Ms. Jayapadmini Kanchan

Assistant Professor Gd -II
Department of Computer Science and Engineering

Submitted By:

Suraj N (NNM22CS185)
Varshith Pawar H R (NNM22CS200)

Date of submission:
19/04/2024

Signature of Course Instructor

ABSTRACT

The project endeavors to create a user-friendly job portal designed to streamline the recruitment process for both employers and job seekers. Employers are empowered to effortlessly post job vacancies, manage applicant data, and review job applications efficiently. Similarly, job seekers can register on the platform, build comprehensive profiles, and apply for relevant job openings seamlessly. By providing a centralized hub for job listings and applications, the platform simplifies the job search journey for individuals seeking employment opportunities, while offering employers a convenient platform to connect with potential candidates. With a focus on simplicity and efficiency, this project aims to bridge the gap between job seekers and employers by providing a straightforward interface for job posting and application management. While not supporting direct acceptance or rejection functionalities, the platform serves as a crucial intermediary, facilitating communication and data collection between employers and job seekers. By harnessing technology to streamline the recruitment process, the project endeavors to enhance the overall efficiency and effectiveness of the job market, ultimately contributing to a more seamless and transparent employment ecosystem.

ACKNOWLEDGEMENT

The satisfaction that accompanies the completion of any task would be incomplete without the mention of all the people, without whom this endeavour would have been a difficult one to achieve. Their constant blessings, encouragement, guidance, and suggestions have been a constant source of inspiration.

I take this opportunity to express my gratitude and regards to my project guide, **Ms. Jayapadmini Kanchan**, Assistant Professor Gd-II, Department of Computer Science Engineering for her guidance, monitoring, and encouragement throughout this Mini project**.**

I would like to express my deepest gratitude to **Dr. Jyothi Shetty**, Head of the Department of Computer Science and Engineering, for her invaluable guidance, unwavering support, and encouragement throughout my academic journey.

My sincere thanks to our beloved principal, **Dr. Niranjan N Chiplunkar,** and vice principal **Dr. I.R.Mithanthaya** for permitting us to carry out this Mini project at our college and providing us with all needed facilities. Also, I thank my parents and friends for the immense support, honest opinion, and suggestions throughout the mini project.

Table of contents

List of figures

Chapter 1

Introduction

In an era defined by digital connectivity and rapid technological advancements, the traditional job search and recruitment processes are undergoing a significant transformation. Recognizing the need for an efficient and user-friendly solution, our project introduces a modernized job portal designed to simplify the recruitment journey for both employers and job seekers. With a focus on streamlining the process of job posting, application submission, and applicant management, our platform aims to revolutionize the way individuals and organizations engage in the hiring process.

Employers face the challenge of attracting top talent in a competitive job market, while job seekers navigate through countless job listings to find the perfect opportunity. Our project addresses these challenges by providing a centralized platform where employers can effortlessly post job openings and manage applicant data, and where job seekers can create detailed profiles, apply for jobs, and track their application status. Through innovative features and a user-friendly interface, our job portal aims to foster seamless communication and collaboration between employers and job seekers, ultimately contributing to a more efficient and transparent recruitment ecosystem.

1.1 Purpose

The purpose of this project is to revolutionize the recruitment process by providing a streamlined and user-friendly job portal. With a focus on simplicity and efficiency, the platform aims to simplify the job search journey for both employers and job seekers. For employers, the portal offers a seamless way to post job openings, manage applicant data, and communicate with potential candidates. Job seekers, on the other hand, can create detailed profiles, apply for jobs effortlessly. By serving as a centralized hub for job listings and applications, the platform aims to foster transparent communication and collaboration between employers and job seekers, ultimately enhancing the overall efficiency and effectiveness of the recruitment process.

1.2 Scope

The scope of this project encompasses the development of a limited interaction job portal focused on facilitating essential functionalities such as job posting, application submission, and data management. While the platform does not support direct acceptance/rejection or interview scheduling, it serves as a centralized hub for collecting applicant data and managing job listings. Emphasizing efficiency and simplicity, the project aims to provide a user-friendly interface for both employers and job seekers, streamlining the job search and hiring process without overwhelming users with unnecessary features.

1.3 Overview

This project aims to develop an intuitive online job portal designed to simplify the recruitment process for both employers and job seekers. Employers will have the capability to register on the platform, effortlessly post job openings, and manage applicant data efficiently. Similarly, job seekers can create detailed profiles, explore job listings, and apply for positions seamlessly. While the platform does not support direct acceptance or rejection functionalities, it serves as a centralized hub for collecting applicant data and facilitating communication between employers and job seekers. With a focus on user-friendly design and essential functionalities, the project endeavours to streamline the job search and hiring process, ultimately contributing to a more efficient and transparent employment ecosystem.

Chapter 2

Requirements Specification

2.1 Hardware specification

- Processor: AMD Ryzen 5 5600H with Radeon Graphics 3.30 GHz
- RAM: 8GB
- Hard Disk: 512GB
- Input Device: Standard keyboard and Mouse
- Output Device: Monitor

2.2 Software specification

- Database: MySQL 5.5
- Markup Language: HTML5
- Scripting Language: PHP 7.0.1
- IDE: Visual Studio Code

Chapter 3

System Design

3.1 ER diagram

The project starts with creating two strong entities company and applicant and then the remaining entities tbl_job_application, tbl_jobs the relationship

An applicant can apply n number of jobs which is shown by the relation applies and a company can post n number of jobs which is shown by the relation posts, a single company can receive n number of job application.

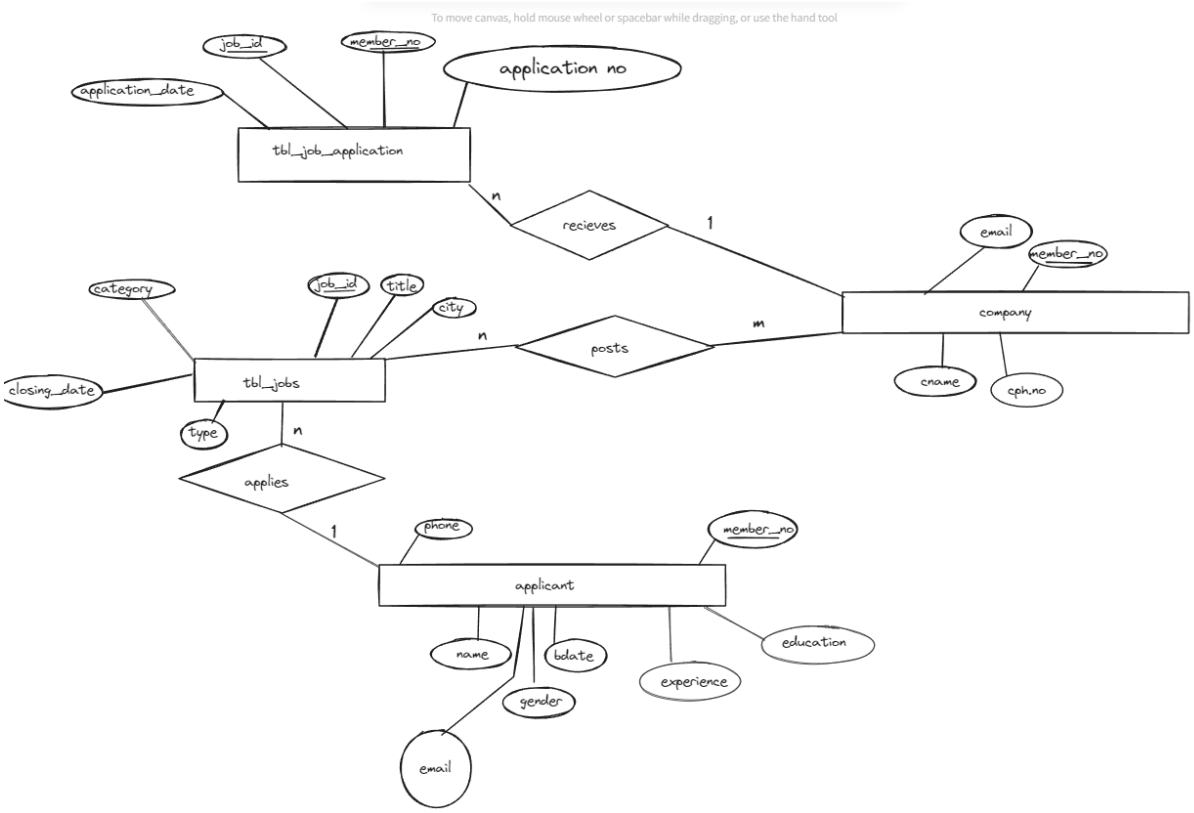

Figure 3.1: ER Diagram of Online Job Portal

3.2 Mapping from ER diagram to schema diagram

1. **Mapping of Regular Entities**: This step involves mapping all the regular entity types to tabular format by identifying their primary keys.
2. **Mapping of 1:1 Relation:** In this step foreign keys are assigned using foreign key approach. The primary key of the participating relation R or S is added as primary keyto second entity types by looking at the participating constraints.
3. **Mapping of 1: N Relation:** Foreign key approach is used to add one sided primarykey to the n sided entity at foreign key.
4. **Mapping of M: N Relation:** Here we use the cross-reference approach where the relationship is converted to a new relation within attributes on primary keys of both participating relations.
5. **Mapping of Weak Entity:** When mapping weak entity types along with other attributes the partial key and primary key of parent entity together will form their primary key of the new relation.
6. **Mapping of N-ary Relation:** For mapping N array relationship we create a new relation with a relationship name in its attribute and primary keys of all participating entity types.
7. **Mapping of Multivalued Relation:** For multivalued attributes a separate relation must be created along with primary key of parent relation.

This database has the following mappings:

1.Mapping of Regular Entities:

Company, applicant, tbl_jobs, tbl_job_application

2.Mapping of 1:1 relationship

3.Mapping of 1: n relationship

4.Mapping of m: n relationship

3.3 Assumptions

The ER diagram reveals several assumptions about the system. Firstly, it conclusively indicates that only a single application can be submitted for a specific job opening, which implies that applicants are unable to apply for multiple jobs with one application. The inclusion of a separate "applicant" table with a one-to-one relationship with "member" suggests a deliberate design choice rather than redundancy. Similarly, the presence of the "posts" attribute in the "tbl_jobs" table strongly indicates the necessity for a separate table dedicated to managing job postings, even though it is not explicitly shown. Details about companies within the "tbl_jobs" table provide clear evidence of a requirement for a separate table to store more comprehensive company information. While basic member information is included, the absence of additional details such as resumes or skills confirms that this data is stored elsewhere in the system. Lastly, the presence of some redundancy in the diagram suggests that it reflects a denormalized design for performance reasons, indicating deliberate decisions made during the schema design process.

3.4 Schema Diagram

A Schema is a pictorial representation of the relationship between the database tables in the database that is created. The database schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term" schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a database schema is a set of formulas (sentences) called integrity constraints imposed on a database. These integrity constraints ensure compatibility between partsof the schema. All constraints are expressible in the same language. A database can be considered a structure in realization of the database language. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modelled in the database.

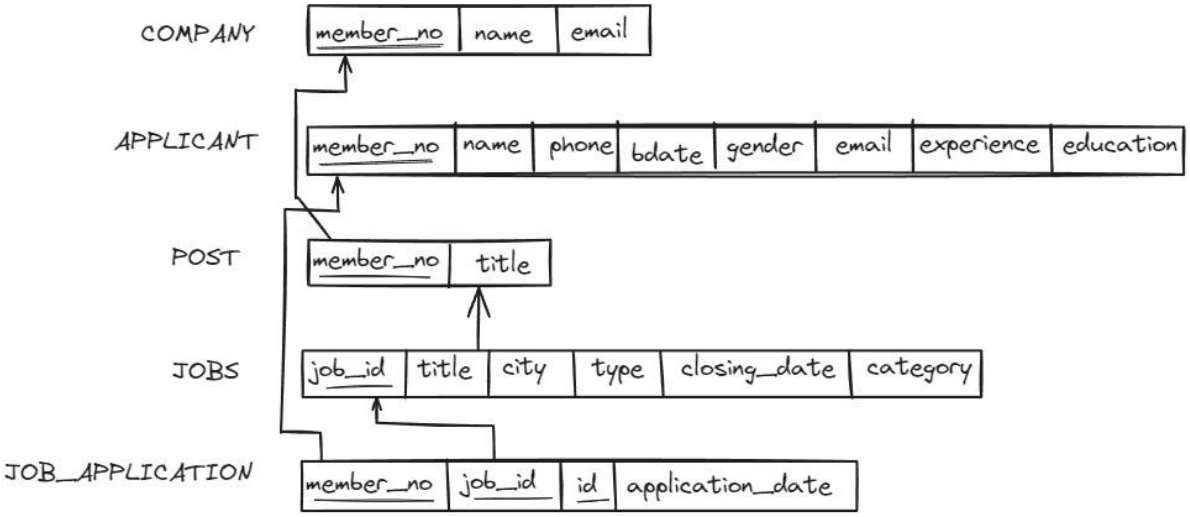

Figure 3.2: Schema Diagram for Online Job Portal

Chapter 4

Implementation

4.1 Pseudocode for Online Job Portal

Pseudocode to connection of database:

To store or access the data inside a MySQL database, you first need to connect to the MySQL database server. In PHP you can easily do this using the MySQL connect () function. All communication between PHP and the MySQL database server takes place through this connection. The hostname parameter in the above syntax specifies the host name (e.g. localhost), whereas the username and password parameters specify the credentials to access MySQL server, and the database parameter, if provided will specify the default MySQL database to be used when performing queries. The default usernamefor MySQL database server is root and there is no password.

```php
<?php
require 'constants/db_config.php';
if (isset($_GET['r'])) {
$error_code = $_GET['r'];

ogin.php
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);


    $stmt = $conn->prepare("SELECT * FROM tbl_alerts WHERE code = :errorcode");
    $stmt->bindParam(':errorcode', $error_code);
    $stmt->execute();
    $result = $stmt->fetchAll();

    foreach($result as $row)
    {

     $description = $row['description'];
     $type = $row['type'];
     print '
     <div class="alert alert-'.$type.'">
     '.$description.'
     </div>
     ';
    }
```

Figure 4.1: Pseudocode for connection to database

Pseudocode to insert values into database:

Insert statement is a DML (Data modification language) statement which is used to insert data in the MySQL table. PHP_ $ POST is a PHP super global variable which is used to collect form data after submitting an HTML form with method=" post." The PHP MySQL query () function accepts a string value representing a query as one of the parameters and, executes/performs the given query on the database.

```
include 'constants/check_reply.php';
?>
                    <form name="frm" action="app/auth.php" method="POST" autocomplete="off">
                    <div class="login-box-wrapper">

                    <div class="modal-header">
                    <h4 class="modal-title text-center">Access your account</h4>
                    </div>

                    <div class="modal-body">

                    <div class="row gap-20">


                    <div class="col-sm-12 col-md-12">

                    <div class="form-group">
                    <label>Email Address</label>
                    <input class="form-control" placeholder="Enter your email address"
                    name="email" required type="text">
                    </div>

                     </div>
```

Figure 4.2: Pseudocode for Inserting values into the database

Pseudocode for updating:

The UPDATE statement is used to modify the existing records in a table. The WHEREclause specifies which record(s) that should be updated.

```
update.php M
update.php
<?php
session_start();
include("connect/config.php");
$pid = $_GET["id"];
$sql = "SELECT * FROM plants WHERE P_id = $pid";
$result = mysqli_query($conn,$sql);
while($row = mysqli_fetch_array($result)){
    $pname = $row['P_name'];
    $ptype = $row['P_type'];
    $quantity = $row['Quantity'];
    $price = $row['Price'];
}
$db_fields = array('P_name','P_type','Quantity', 'Price');
$sqlfield = '';
if ($_SERVER['REQUEST_METHOD'] == 'POST' )
{
    foreach($db_fields as $fieldname) {
        if (!empty($_POST[$fieldname])) {
            $sqlfield .= "$fieldname = '{$_POST[$fieldname]}',";
        }
    }
    $sqlfield = rtrim($sqlfield,',');
    $sql = "UPDATE plants SET $sqlfield WHERE P_id = $pid";
    if (mysqli_query($conn, $sql)) {
        echo "<script type='text/javascript'>alert('Details updates successfully');
        window.location='adminplants.php';</script>";
        die;
    } else{
        echo "<script type='text/javascript'>alert('Oops! Something went wrong. Please try again later.');
        window.location='adminplants.php';</script>";
        die;
    }
    mysqli_close($conn);
    }
?>
```

Figure 4.3: Pseudocode for Updating

4.2 Tables used

User table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	first_name	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
2	last_name	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
3	gender	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
4	bdate	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
5	bmonth	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
6	byear	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
7	email	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
8	education	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
9	title	varchar(255)	latin1_swedish_ci		No	Your professional			Change Drop More
10	city	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
11	street	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
12	zip	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
13	country	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
14	phone	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
15	about	longtext	latin1_swedish_ci		Yes	NULL			Change Drop More
16	avatar	longblob			Yes	NULL			Change Drop More
17	services	longtext	latin1_swedish_ci		Yes	NULL			Change Drop More
18	expertise	longtext	latin1_swedish_ci		Yes	NULL			Change Drop More
19	people	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
20	last_login	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
21	role	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
22	website	varchar(255)	latin1_swedish_ci		No	-			Change Drop More
23	login	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
24	member_no	varchar(255)	latin1_swedish_ci		No	None			Change Drop More

Figure 4.4: Structure of user table

Company table:

Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
member_no	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
first_name	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
last_name	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
email	varchar(255)	latin1_swedish_ci		No	None			Change Drop More

Figure 4.5: Structure of company table

Jobs table:

Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
job_id	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
title	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
city	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
country	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
category	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
type	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
experience	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
description	longtext	latin1_swedish_ci		No	None			Change Drop More
responsibility	longtext	latin1_swedish_ci		No	None			Change Drop More
requirements	longtext	latin1_swedish_ci		No	None			Change Drop More
company	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
date_posted	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
closing_date	varchar(255)	latin1_swedish_ci		No	None			Change Drop More

Figure 4.6: Structure of Jobs table

Job_application table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(255)			No	None		AUTO_INCREMENT	Change Drop More
2	member_no	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
3	job_id	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
4	application_date	varchar(255)	latin1_swedish_ci		No	None			Change Drop More

Figure 4.7: Structure of Job application table

Chapter 5

Results and discussion

Landing page:

This is the common landing page for both company and applicants.

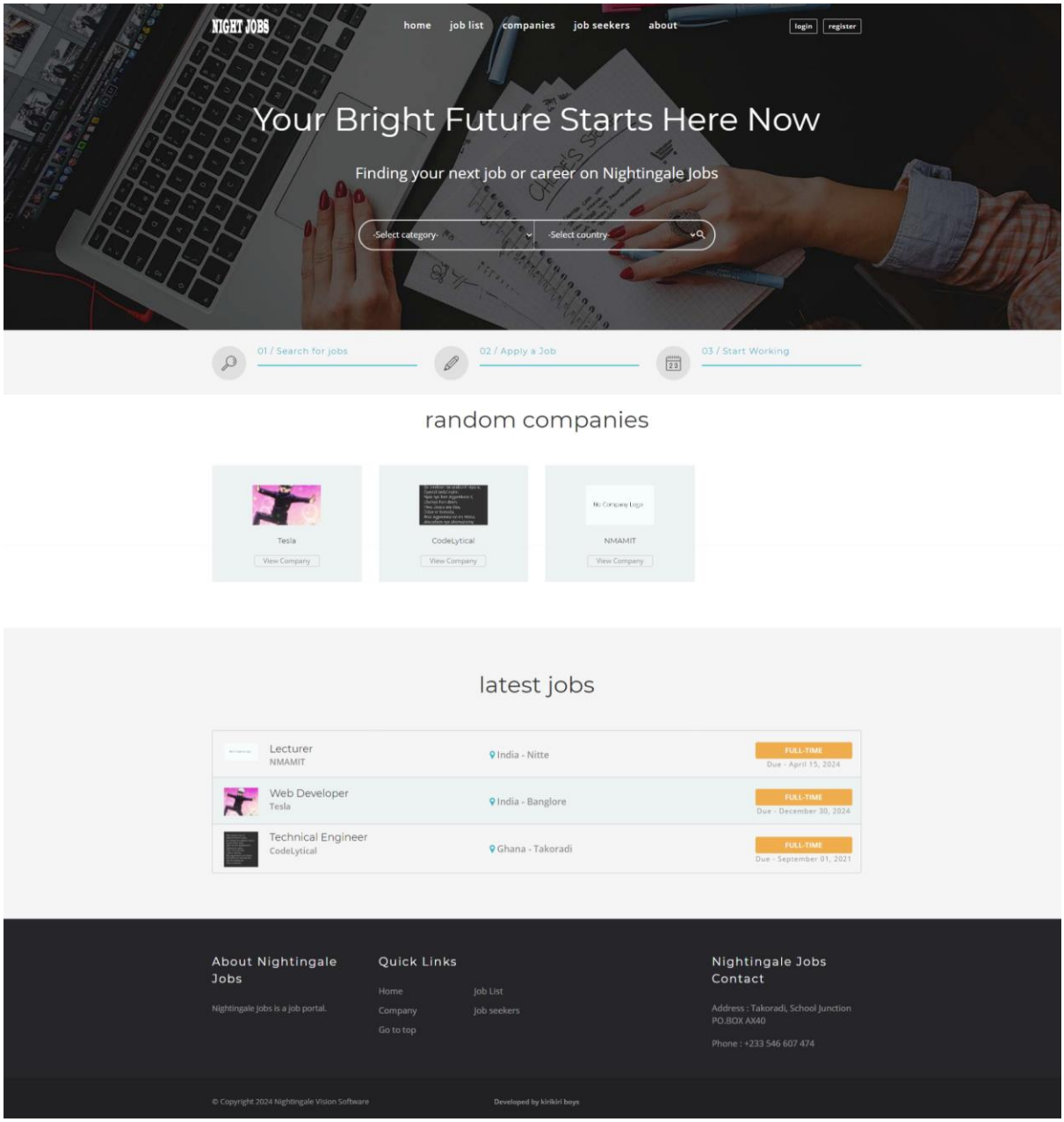

Figure 5.1: Landing Page

Register page:

This is where users can register themselves as company or applicant

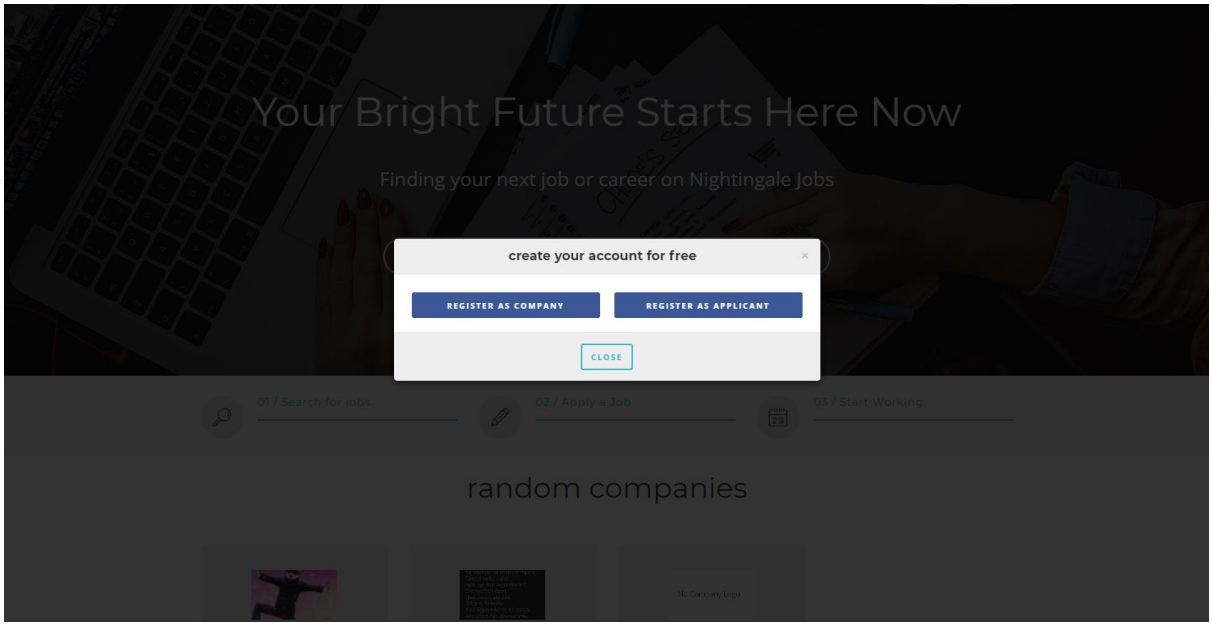

Figure 5.2: Register Page

Register forms:

Form to register in the portal.

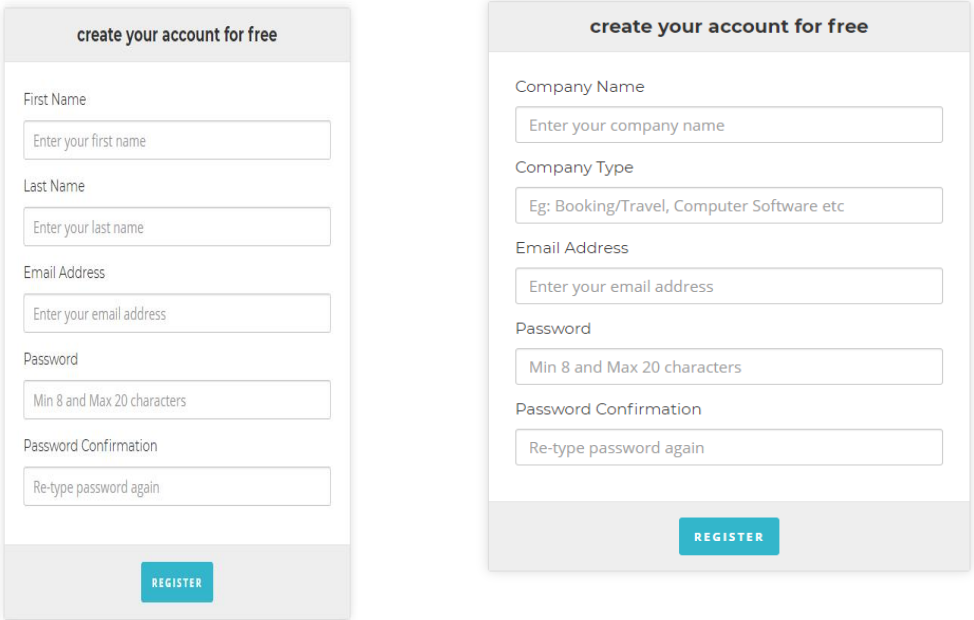

Figure 5.3: Register forms for both company and applicant

Job listed page:

Here all the jobs are listed the applicant can view the job and apply as per his/her need

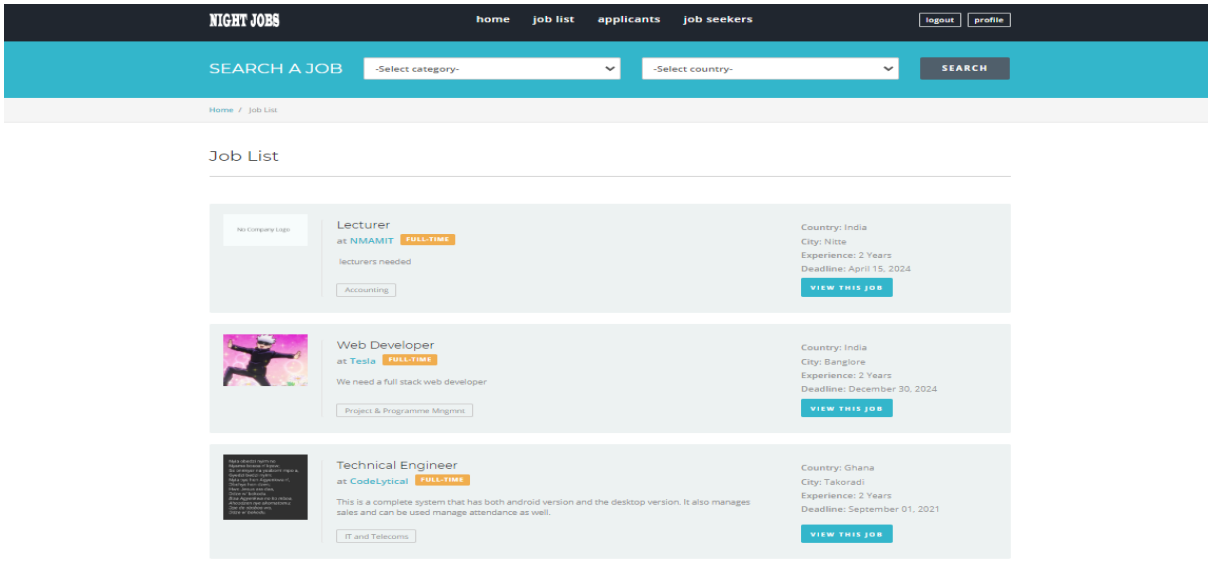

Figure 5.4: Job list page

Employers list page:

Here you can see all the companies that are listed

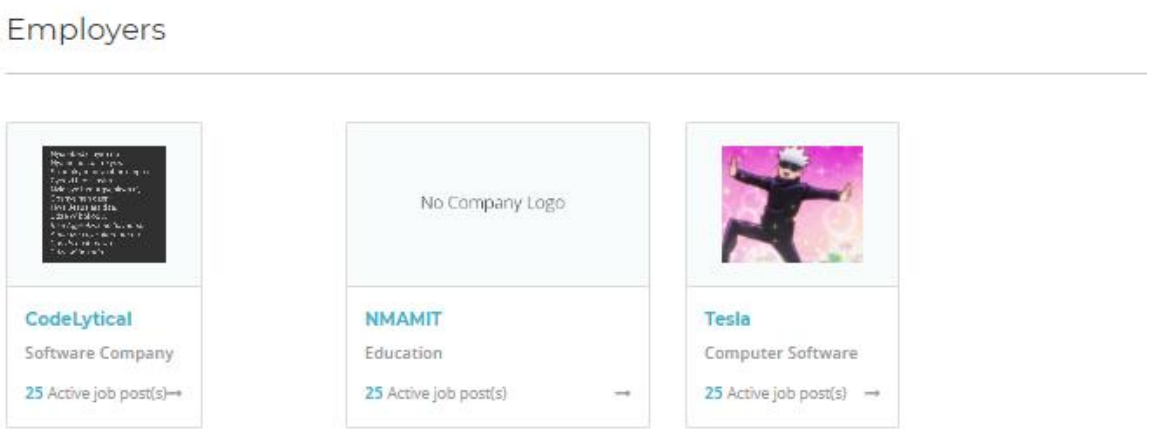

Figure 5.5: employer list page

Profile page of Applicant:

Profile page of applicant where he/she can enter his/her complete details

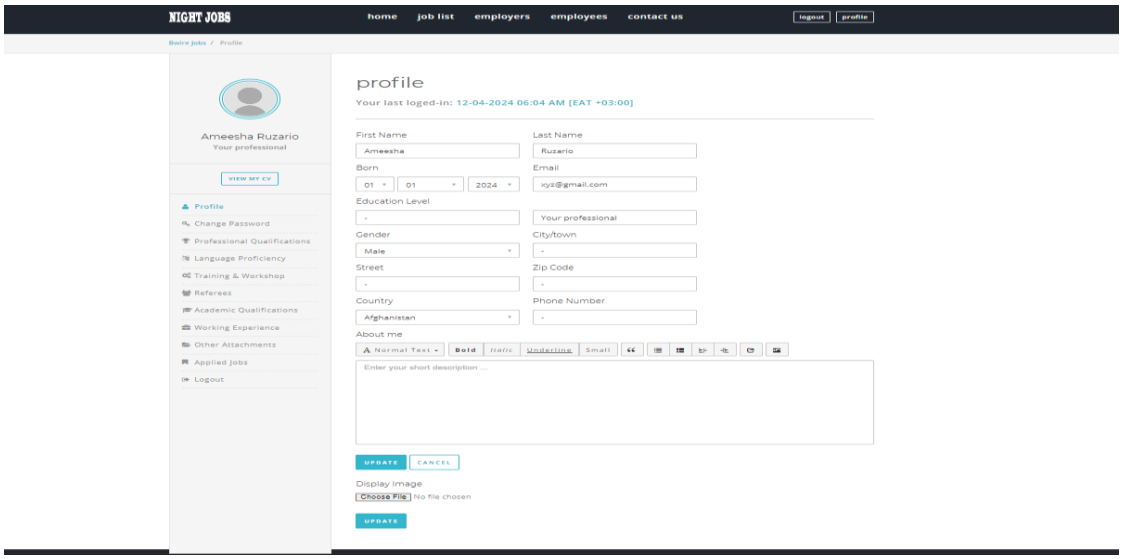

Figure 5.6: profile page of applicant

Job application page:

Here the applicant can see the responsibility and click on apply and after clicking on Apply will receive a message that job has been applied successfully

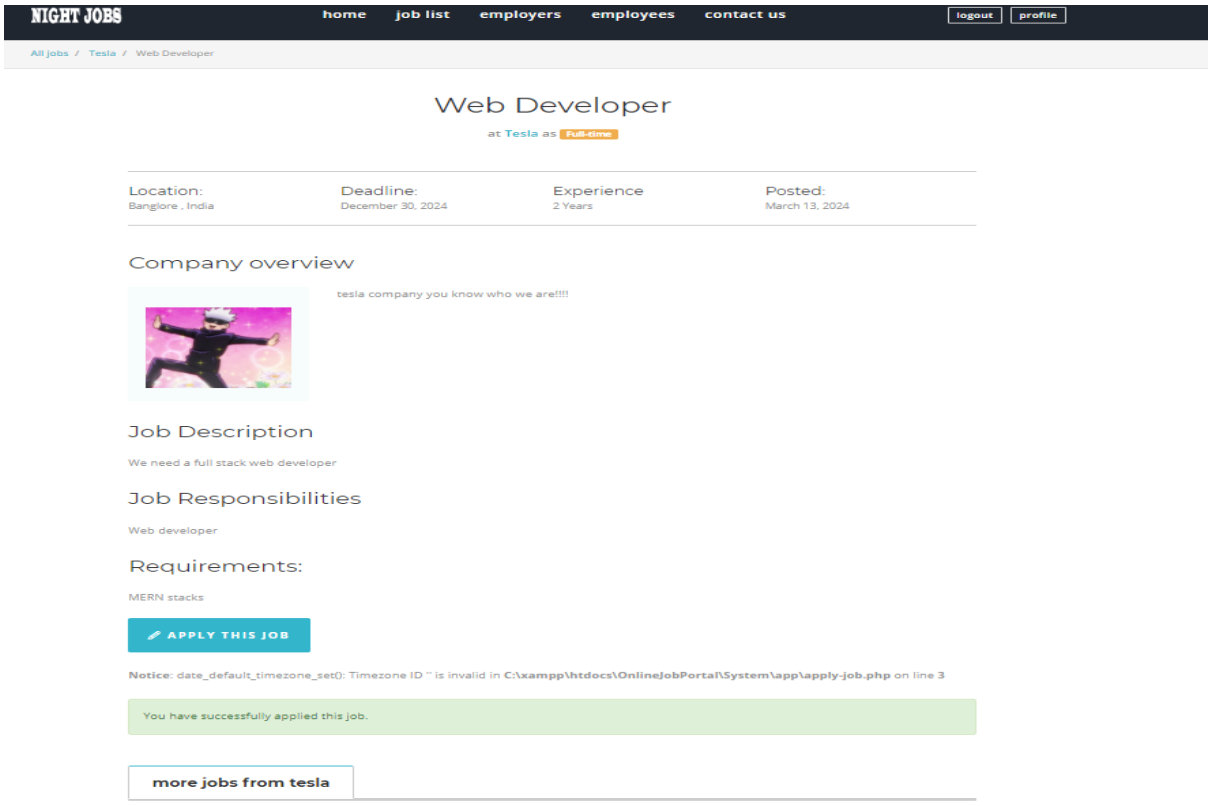

Figure 5.7: Job application page

Company profile:

Here is the profile of a company with the last login information.

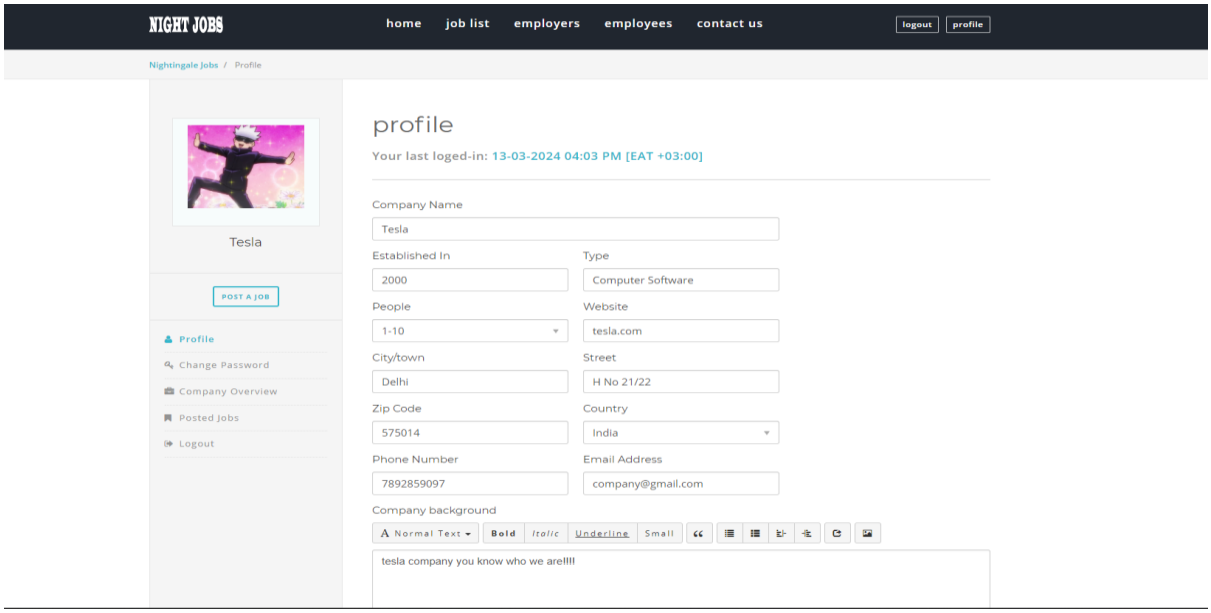

Figure 5.8: company profile page

Post a job form:

Form for posting a job, company can post the job.

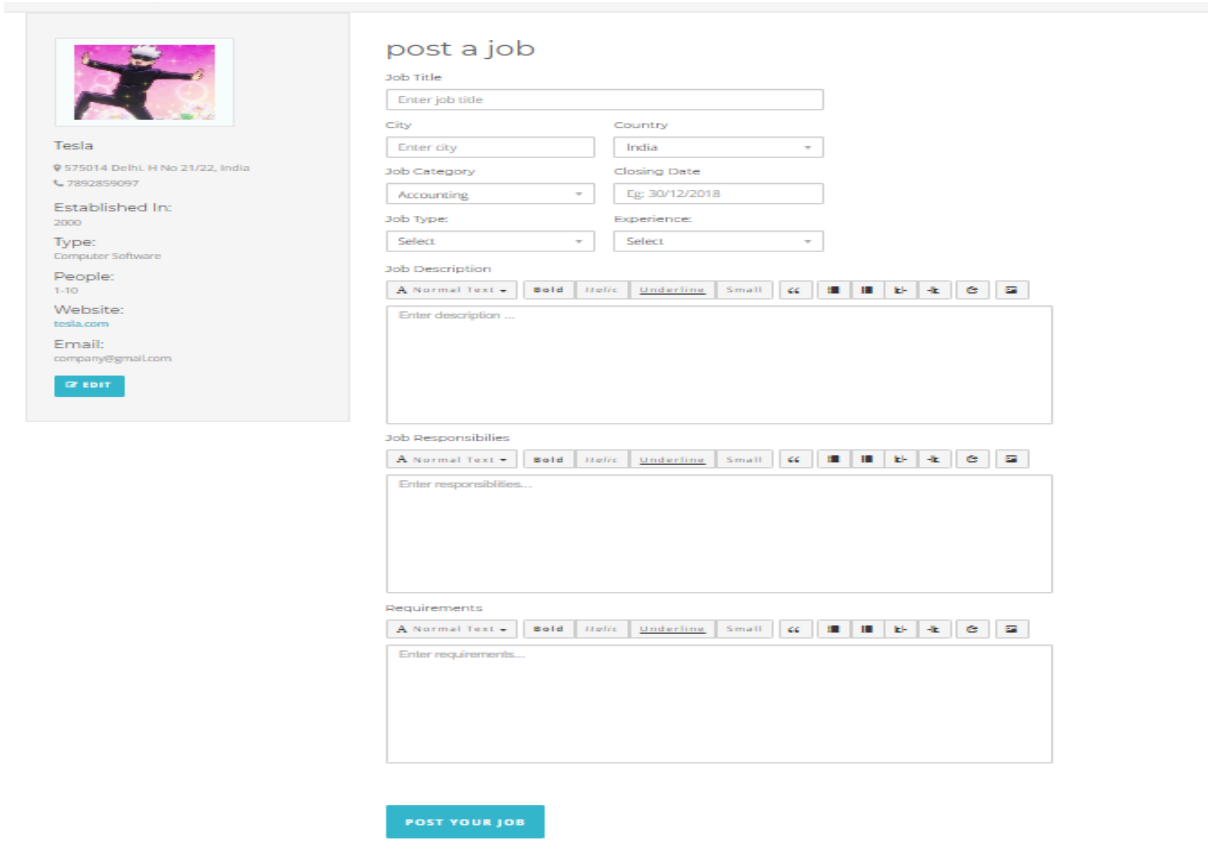

Figure 5.9: post a job form

Message for successful posting of job:

Message pop up for posting a job.

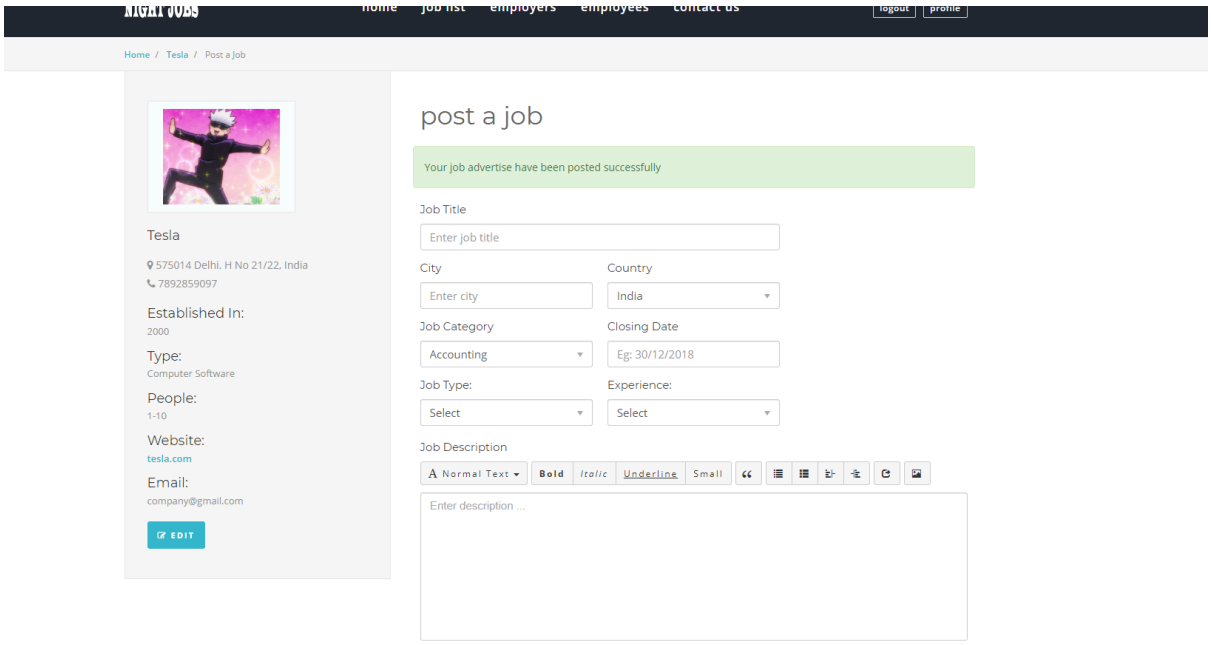

Figure 5.10: message pop up

Login page:

A single login page for both the companies and applicants

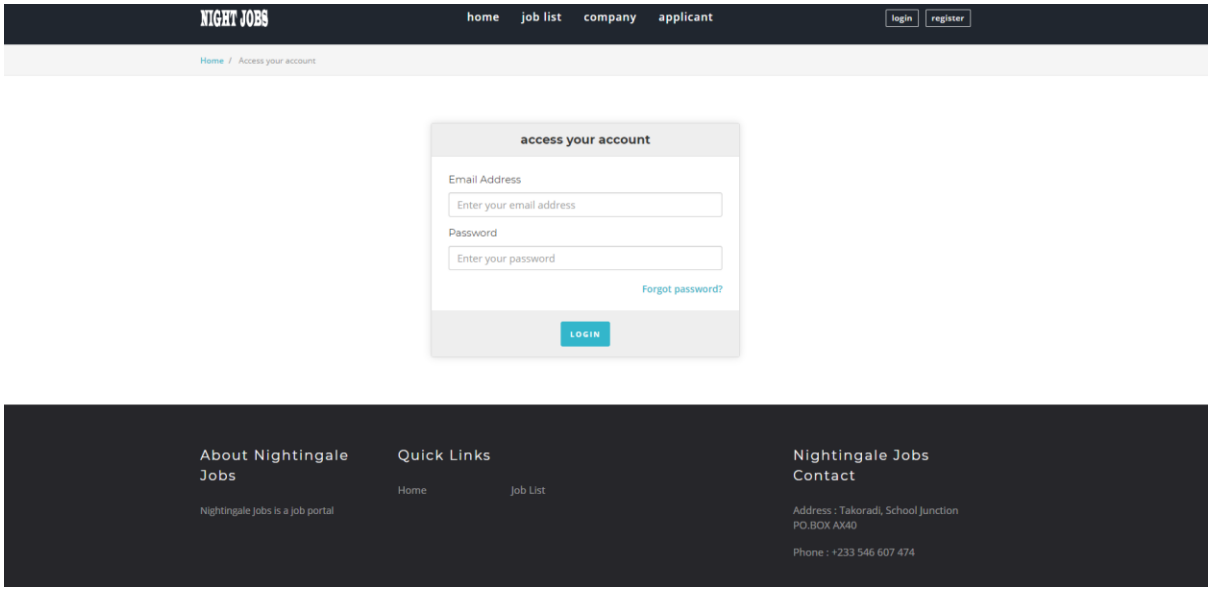

Figure 5.11: Login page

About page:

Complete description of the project is given in this page.

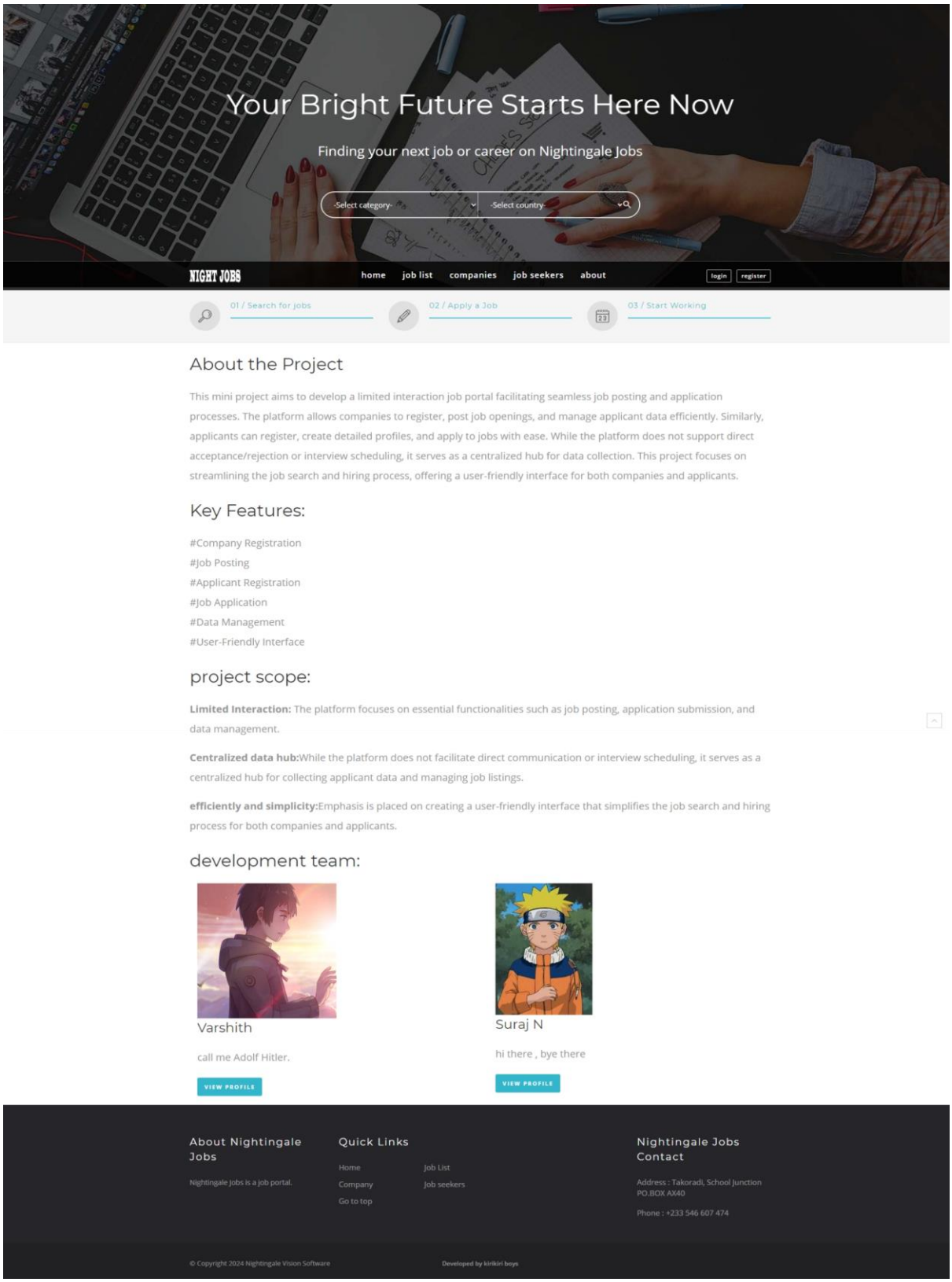

Figure 5.12: about page

Chapter 6

Conclusion and Future works

In summary, the project has aimed to develop a comprehensive job portal solution that streamlines the recruitment process for both employers and job seekers. By providing intuitive interfaces for job posting, application submission, and applicant management, the portal enhances efficiency and transparency in the job market. The ER diagram analysis has illuminated various aspects of the system's design, including assumptions about application submissions and the organization of data. Moving forward, the project seeks to address these assumptions and capitalize on additional features to further optimize system performance and user experience.

In the future, the project aims to further enhance its capabilities by integrating triggers within the database management system to automate tasks and improve data integrity. Additionally, there is a focus on enriching user profiles with additional details such as resumes and skills, implementing advanced search and matching algorithms for more accurate job recommendations, integrating communication tools to facilitate seamless interaction between employers and job seekers, and incorporating analytics and reporting features to gain valuable insights into system usage and user behaviour. These enhancements will contribute to a more efficient and user-centric job portal, ultimately providing a better experience for both employers and job seekers alike.

References

[1] Database systems Models, Languages, Design and Application Programming, RamezElmasri and Shamkant B. Navathe, 7th Edition, Pearson.

[2] Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, Mc-Graw Hill.

[3] Silberschatz Korth and Sudharshan: Database System Concepts, 6th Edition, Mc-Graw Hill, 2013.

[4] Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementnation and Management, Cengage Learning 2012.