

Governors State University

OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Fall 2022

Career Engine

Anudeep Gudimalla

Follow this and additional works at: <https://opus.govst.edu/capstones>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Capstone Project is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

CAREER ENGINE

By

Anudeep Gudimalla

Bachelor's Degree, Jawaharlal Nehru Technological University, 2011

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science



Governors State University
University Park, IL 60484

2022

ABSTRACT

Everyone's learning is increasing in this competitive environment, to the point that professional chances are becoming scarce. Companies desire the greatest personnel in their areas. It was difficult to find persons who were bright enough to be recruited at the time. Companies' attempts to find qualified employees are also increasing. The purpose of this application is to provide a system that allows employers and candidates to communicate with one another. The idea is to speed up the recruiting process by facilitating contact between interested parties. A career engine is a website that offers recruiters and job seekers with online information. Our application helps both job searchers and recruiters choose the finest company for their workers. For job seekers, the employment site presents a list of companies based on their educational credentials, experience, and interests. Furthermore, it offers appropriate candidates from a pool of shortages to recruiters.

In this project, we utilized MySQL to store the data, and the frontend is constructed using HTML5, CSS3, Bootstrap, JavaScript, and the backend for PHP is made with the MVC CodeIgniter framework. In our application, we use the technological stacks listed above. Our application presented three responsibilities: job searchers, recruiters, and administrators. The responsibility of the job seeker is to apply for the position and offer feedback to the business. The following recruiter's role is to publicize job vacancies, contact job seekers, and conduct interviews on their behalf. All users, including job seekers, recruiters, and job listings, must be tracked by the administrator.

Table of Content

1	<i>Project Description</i>	1
1.1	Competitive Information	1
1.2	Relationship to Other Applications/Projects	1
1.3	Assumptions and Dependencies	1
1.4	Future Enhancements.....	1
1.5	Definitions and Acronyms	2
2	<i>Project Technical Description</i>	2
2.1	Application Architecture	2
2.2	Application Information flows.....	4
2.3	Interactions with other Applications	4
2.4	Capabilities	4
2.5	Risk Assessment and Management.....	5
3	<i>Project Requirements</i>	5
3.1	Identification of Requirements	5
3.2	Operations, Administration, Maintenance and Provisioning (OAM&P)	5
3.3	Security and Fraud Prevention.....	5
3.4	Release and Transition Plan.....	5
4	<i>Project Design Description</i>	6
5	<i>Internal/external Interface Impacts and Specification</i>	12
6	<i>Design Units Impacts</i>	13
6.1	Functional Area A/Design Unit A	13
6.1.1	<i>Impacts</i>	13
6.1.2	<i>Requirements</i>	13
7	<i>Acknowledgements</i>	14
8	<i>References</i>	14
9	<i>Appendices</i>	14

1 Project Description

A career engine called HIKEPRO created to give job seekers access to an interactive job board. In its current condition, this online application should be considered a dynamic website that constantly requires updates from both users and businesses. The project's objective is to make it possible for job seekers to submit their resumes and find appropriate employment, while companies may post openings and find competent candidates. On the website, job searchers may upload their resumes, do a job search, and view listings for individual positions. A variety of organizations will be able to browse candidate resumes and post job opportunities on the website. An admin module will be accessible in addition to the jobseeker and company modules, allowing for the administration of the whole Portal.

1.1 Competitive Information

There are several job portal websites available, however for the time being, we decided to match our application with Dice and LinkedIn. We can browse at job seekers' profiles and apply for positions on Dice and LinkedIn. Similar to this, the front page of an online job site features the newest job information as well as the top categories for job search filters by category. Based on input from users or job seekers, we provide star ratings.

1.2 Relationship to Other Applications/Projects

This project provides an all-inclusive solution that may be used for any purpose. On order to search for available positions in the market, job searchers may utilize this project. In addition to communicating with one another about it, recruiters might look for job seekers who are looking for work. The hiring manager can choose to shortlist or reject a job candidate, and the candidate can choose to apply for positions that are still open.

1.3 Assumptions and Dependencies

A fundamental grasp of how to utilize the website is required for all users. Anyone else filling the role of administrator has to be knowledgeable about the process, how to handle recruiters, job searchers, job advertisements, and other administrative duties.

1.4 Future Enhancements

The goals and objectives of the system have been met, yet there is still potential for development. SMS integration for OTP is used to stop recruiters or job seekers from registering without permission. This effort's scope encompasses all information relating to jobs. Additionally, we may give recruiters access to a database of application profiles and training for prospects' self-improvement. Some of the limitations of older job search techniques have been eliminated.

1.5 Definitions and Acronyms

- XAMPP - XAMPP is a free and open-source cross-platform web server package that consists mostly of the Apache HTTP Server, MySQL database, and interpreters for PHP and Perl scripts [1]. The acronym XAMPP stands for: X: any of the several operating systems (Windows, Linux, Mac OS X), to be understood as "cross," which means "platform."
- HTML (hypertext markup language) - HTML stands for hypertext markup language. It's a hypertext markup language (HTML) that's used to make web pages. It's written with HTML components in mind.
- CSS - Cascading Style Sheets is a prominent design language with a full name of CSS. It simplifies the work of displaying webpages and is a relatively easy language to use.
- Hypertext Preprocessor (PHP) (earlier called, Personal Home Page) PHP is a server-side scripting language for web development that is incorporated in HTML.

2 Project Technical Description

This application's architecture and development tools were as follows:

- Database: MySQL; Operating System: Microsoft Windows 10
- Front end components include PHP, HTML, CSS, and JavaScript [3].
- Local host components include Apache Server (XAMPP) and Chrome.
- Any text editor is OK. We preferred Visual Studio Code.

The CodeIgniter Framework's MVC architecture [4] is used by us. Since this is a one-time operation, we decided to utilize the auto-load database library instead of loading it each time you query the database. We employed session variables for the authentication of administrators, recruiters, and seekers. For Notification, we utilized the jQuery basic toast capability. PHP was chosen because it is flexible, affordable, easy to set up, safe, quick, and dependable, and it can interface with a variety of relational database management systems. One of the most widely used open source database management systems worldwide is MySQL. Users may create databases using this free and open source database, as well as carry out various data transformations and database management operations. Because of its advanced features, including data security, on-demand scalability, fast speed, and comprehensive workflow management, MySQL was chosen for this project.

2.1 Application Architecture

Administrator Functionalities

Login - A legitimate username and password are entered into the system by the administrator to login.

Dashboard - The administrator may view the system's total number of candidates, companies, job postings, and registered applications.

Active / Inactive - Job seekers' and employers' accounts are activated by the administrator, who also adds accounts for both parties.

Category - To filter job posts by category, the administrator must first establish the category.

Profile - Administrators have the ability to edit their profile information and password.

Logout - The administrator logs out of the system after each activity.

Employer Functionalities

Employers Registration - A company registers as an employer and generates a new account.

Employer Login - Using a legitimate email account and password, the company logs into the system.

Post a Job - An employer post the jobs without any restrictions.

Profile update - An employer makes changes to their profile.

Job List Details – Employers locate the positions they listed.

Candidate List – The employer may browse all the applicants, arrange interviews, and pick or reject individuals.

View resume – Look through the resumes of the candidates who applied.

Chat - An employer will communicate with a registered job seeker to discuss the interview and other matters.

Logout - After the action is over, the user logs out of the system.

Job Seekers Functionalities

Registration - A applicant must create a new account to register.

Login - The candidate logs in to the system with a legitimate email address and password.

Find Job. - A job seeker searches for job postings to find a job.

Edit Profile - Changes are made to a seeker's profile using the editing feature.

Upload Resume - An applicant enters their résumé into the system by uploading it.

Job Application - An applicant who is interested in a position submits an application for it.

Search company profile - Candidates can use the system to look up firms by name and location by searching the company profile.

Send Feedback - Sending feedback to the recruiter is an option for candidates.

View applied jobs - A candidate has access to a list of all the positions for which they have applied.

Logout - When the candidate has finished the activity, he or she logs out of the system.

2.2 Application Information flows

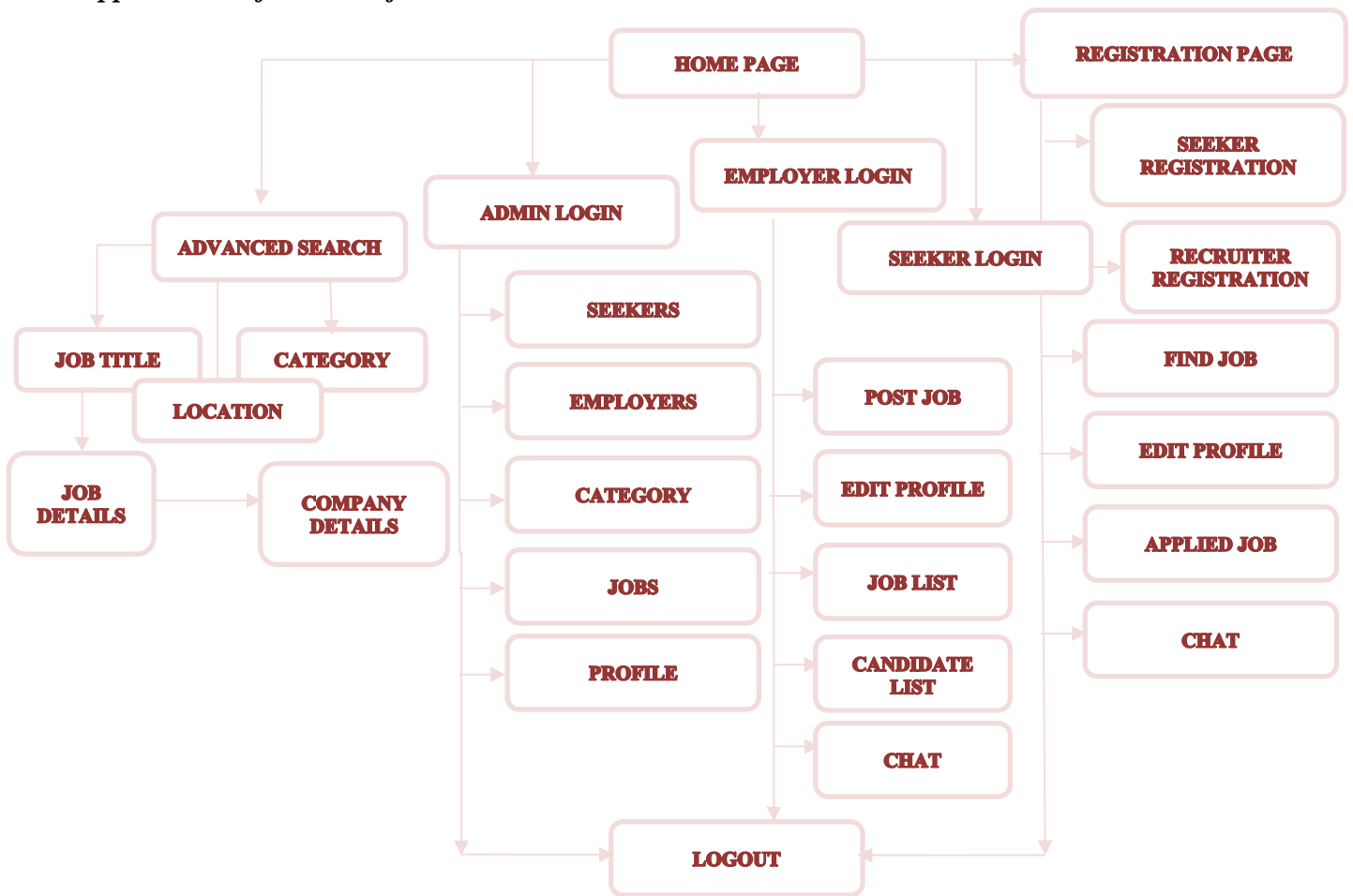


Figure 1: Application workflow

2.3 Interactions with other Applications

We provide job specifications, seeker data, and company details on this platform. We can also offer training in technology and personality. That is determined by how well they performed in the interview, which is better for job seekers in terms of extending their knowledge and for recruiters in terms of discovering more competent applicants for their vacant positions.

2.4 Capabilities

- It maintains a database of recruiters and job opportunities and accepts resume submissions from job seekers [2].
- Job applicants' personal information, educational level, skills, and experience are among the data it saves.
- Employers and job seekers can exchange messages. It includes feedback mechanisms.

2.5 *Risk Assessment and Management*

The right risk management procedures were followed, and backup plans were established, such as task management process error termination, to make the system easier to use in the event of a mistake.

3 *Project Requirements*

3.1 *Identification of Requirements*

<HikePro_CareerEngine_2022-1 User-capability-“CE1”>

Responsive Design - On whatever device, users can anticipate the finest user experience.

< HikePro_Career_Engine_2022-2 User-capability-“CE2”>

Advanced Search: Job searchers may look for openings based on a variety of parameters, including job location, job type, and job title.

< HikePro_Career_Engine_2022-3 User-capability-“CE3”>

Registration: The registration process is voluntary for recruiters and job searchers, but it is a prerequisite to carrying out the required action on their portal.

3.2 *Operations, Administration, Maintenance and Provisioning (OAM&P)*

There is no requirement for user registration or login in order to use the website. You have access to the main pages as a guest. The data acquired through the search option ought to be kept in the company's or job seeker's database. The admin may view all of the data in the dashboard for easy access. Pages load more quickly for complex web apps like this when large file sizes are reduced. This program ensures that a regular backup of the entire website and data is produced in order to protect the integrity of the website and data. This program verifies that the settings are accurate. The capacity to monitor user progress and place access restrictions.

3.3 *Security and Fraud Prevention*

The method for establishing security and fraud detection may be established using security planning and planning for fraud prevention. The following phases make up the fraud prevention planning process. Regular updates should be made to the application. As a consequence of the required registration process, all user information is saved.

3.4 *Release and Transition Plan*

The website must be properly finished and tested prior to being pushed to the live server. The website shouldn't be made available for job consultation until the problems are fixed. The data storage procedure will start after the computer has booted, and the maintenance process will be used to maintain the application's continued operation.

4 Project Design Description

The MVC framework, which is the most modern and effective, speedier than any other platform, open source, and independent of any environment, was used to develop all the features in the Career engine project. The MVC design pattern, which offers greater space and simpler code, is used in project development. Model, View, and Controller are the three main divisions used by MVC to organize all code segments. Depending on the controller function, it retrieves data from the model, utilizes it, and shows it in the view. This controller is used to redirect user requests. In this project, we also used to ajax for notification alerts, and we designed the sites so they wouldn't need to be reloaded. As a result, we used a single instance to develop toast notification for the alert messaging service. The administrator has the ability to create the category. The first step is for the administrator to create a category since in order for a recruiter to post a position after registering they must first select a category. A recruiter's profile can be updated. Everyone wants to visit the website except for the recruiter who advertised the position once the administrator activated it. The applicant can submit an application after the administrator is engaged in the post. If the recruiter likes the applicant's profile, they select them from the shortlist; if they don't think the applicant is a good fit for the position, they reject the application. In the case of an emergency, recruiters will use the applicant they are holding onto while searching for a better fit. The chat feature enables communication between the recruiter and the candidate. Among the advantages of this job application are the following: availability, learnability, user-friendliness, and simplicity of usage Access is available to authorized users around-the-clock. Any computer or portable device with internet connectivity can use this system. Efficiency, speed, and maintainability are all taken into consideration, as well as the validation of data in fields like names, phone numbers, and email addresses.

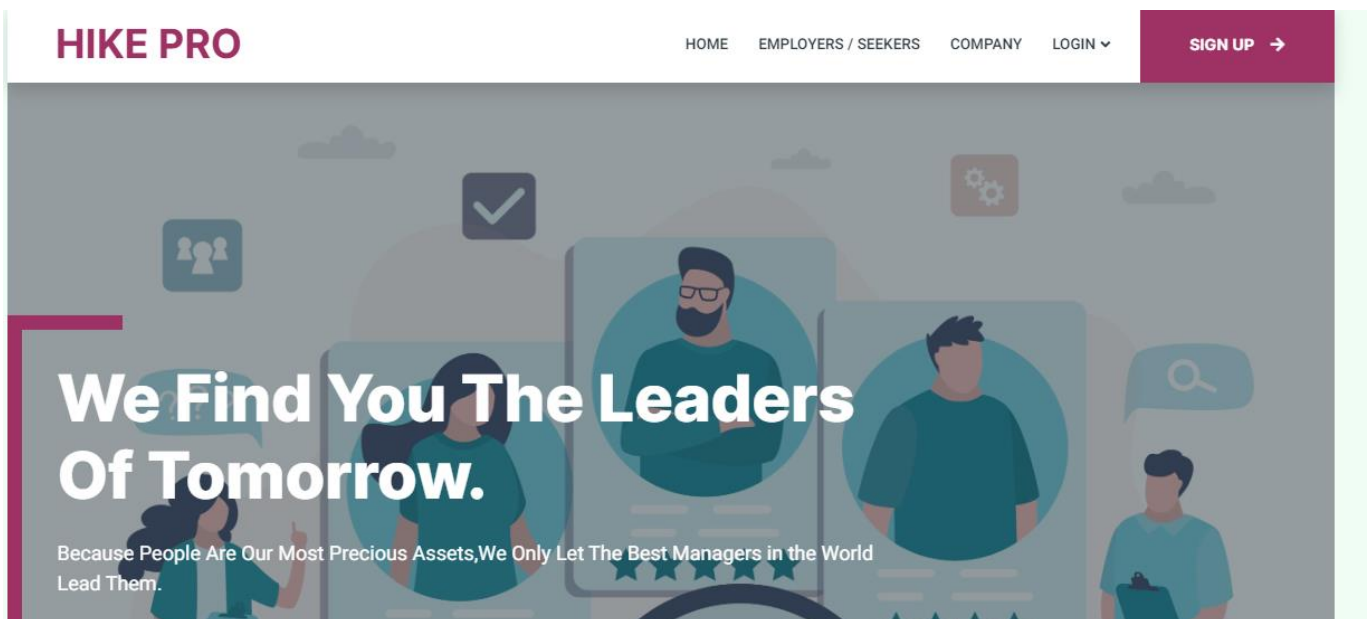


Figure 2: Guest Page

Enter Job Keyword to search Category Location **Search**

FEATURED BY CATEGORY

Frontend Developer Angular Developer 10 Vacancy	Frontend Developer React JS 10 Vacancy	Frontend Developer Web Designers 8 Vacancy	Software Testing Not Yet Assigned
Human Resource Project Manager 5 Vacancy	Oracle Backend Backend JS 10 Vacancy		

Figure 3: Feature Category

JOB LISTING

[All Featured](#) [Experienced](#) [Freshers](#)

Web Designers Los Angeles Fresher 96000 Date created: November, Fri, 2022	Apply Now
Backend JS California Fresher 25000 Date created: November, Fri, 2022	Apply Now
Project Manager CA,Arkansas Fresher 15000 Date created: November, Fri, 2022	Apply Now

Figure 4: Job Listing

EMPLOYERS

Brentwood CountryMart brentwoodtech@yahoo.com Contact :charles 3103943245 224 25th Street,Santa Monica CA 9042	Frederick NewsAgencies frederivkit@gmail.com Contact :santamonica 1233943245 935,San vincent,Los Angels CA 90049
---	---

Figure 5: Company List

ADMIN LOGIN

User Name
Admin

Password

Submit

Figure 6: Admin Login View

HIKE PRO USERS LOGIN

EMPLOYERS LOGIN **JOB SEARCHER LOGIN**

EMPLOYER LOGIN

User Name Admin	Password *****
--------------------	-------------------

Admin Login **Register New Employer** **Submit**

Figure 7: Employer / Job Searcher Login

EMPLOYERS **JOB SEARCHER**

EMPLOYERS REGISTRATION

Company Name	Official Email
Company Mobile	Contact Person
User Name Admin	Password *****
Confirm Password	Company Location

User Login **Reset** **Submit**

Figure 8: Employer / Job Searcher Registration

Category List Count 4 Job Seeker List Count 3 Employers List Count 2 Jobs List Count 5

Q JOB SEEKER

EMPLOYERS

★ CATEGORY

★ JOBS

+ ADD CATEGORY

JOB SEEKERS DETAILS LIST

Show entries Search:

S.No	Name	Email	Mobile	Dob	Status	Change Status
1	Hailey	haileybajars@gmail.com	4156789870	1990-11-10	Active	<input checked="" type="checkbox"/>
2	Carleton	carletonwidget@gmail.com	4348903266	2001-05-06	Active	<input checked="" type="checkbox"/>
3	Marvaarob	marvaarobsvb@gmail.com	2224316756	1995-20-02	Active	<input checked="" type="checkbox"/>

Showing 1 to 3 of 3 entries Previous Next

Figure 9: Admin Layout

HIKE PRO

Employer Dashboard

[DASHBOARD](#) [WEB SITE](#) [COMPANY](#)

Logout →

Dashboard

+ Post Job

✓ Edit Profile

👤 Job List Details

👤 Candidate List

Details

👤 Chat with Seeker

Brentwood CountryMart COMPANY PROFILE

ADD JOB

Job Title

Job Category

Job Type

Job Company Name
Brentwood CountryMart

Job Featured

Job Status

Job Location

Job Vacancy

Job Description

Notice Period

Expected Salary

Figure 10: Employer Layout

- Dashboard
- Find Job
- Edit Profile
- Applied Job Details
- Chat with Company

Carleton COMPANY PROFILE

SEARCH JOB

Angular Developer
CA, Los Angeles

React JS
US, Arizona

Project Manager
CA, Arkansas

Backend JS
California

Web Designers
Los Angeles

Figure 11: Job Searcher Layout

- Dashboard
- Find Job
- Edit Profile
- Applied Job Details
- Chat with Company

Carleton PROFILE

S.No	Job Title	Company Name	Applied Date	Applied Status
1	Backend JS	Frederick NewsAgencies	November,25, 2022	Pending

Interview Date

S.No	Company Name	Date
1	Frederick NewsAgencies	September,04, 2022

Figure 22: Applied Jobs

Apply For The Job

About Yourself

Job summary

- > Published On: November, 2022
- > Vacancy: 10
- > Job Nature: Fresher
- > Salary: 78000
- > Location: US,Arizona

React JS

📍 US,Arizona 🕒 Fresher 💰 78000

Job description

Company Detail

> Company Name: Frederick NewsAgencies

Figure 33: Specific Job Information

5 Internal/external Interface Impacts and Specification

The software requirements specification includes a complete description of the system's behavior. This lists the hardware and software requirements that the system must have in order to function optimally. Before they can extract system requirements, developers must have a thorough understanding of the products or systems being created. This is achieved by staying in close contact with system users. The architecture of a system, which specifies its structure, behavior, and other perspectives and analyses, is the main emphasis of the architectural design of that system. Three-tier architectural design was used to create this system. The user interacts with the software through the client tier, which is often an interface like a web browser. Web browsers include those like Internet Explorer, Mozilla Firefox, Google Chrome, Microsoft Edge, Opera, and others. Application logic that transmits data or information between the first and second levels makes up the second/middle tier. The database management system, which is in charge of managing databases, is part of the database tier.

Filters

Containing the word:

Table ▲	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> hp_admin	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> hp_category	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> hp_chat	★ Browse Structure Search Insert Empty Drop	6	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> hp_employers	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> hp_interview	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> hp_jobapply	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> hp_jobs	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> hp_review	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> hp_searcher	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci	16.0 KiB	-
<input type="checkbox"/> migrations	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	16.0 KiB	-
10 tables	Sum	30	InnoDB	utf8mb4_general_ci	160.0 KiB	0 B

Figure 44: Database Information

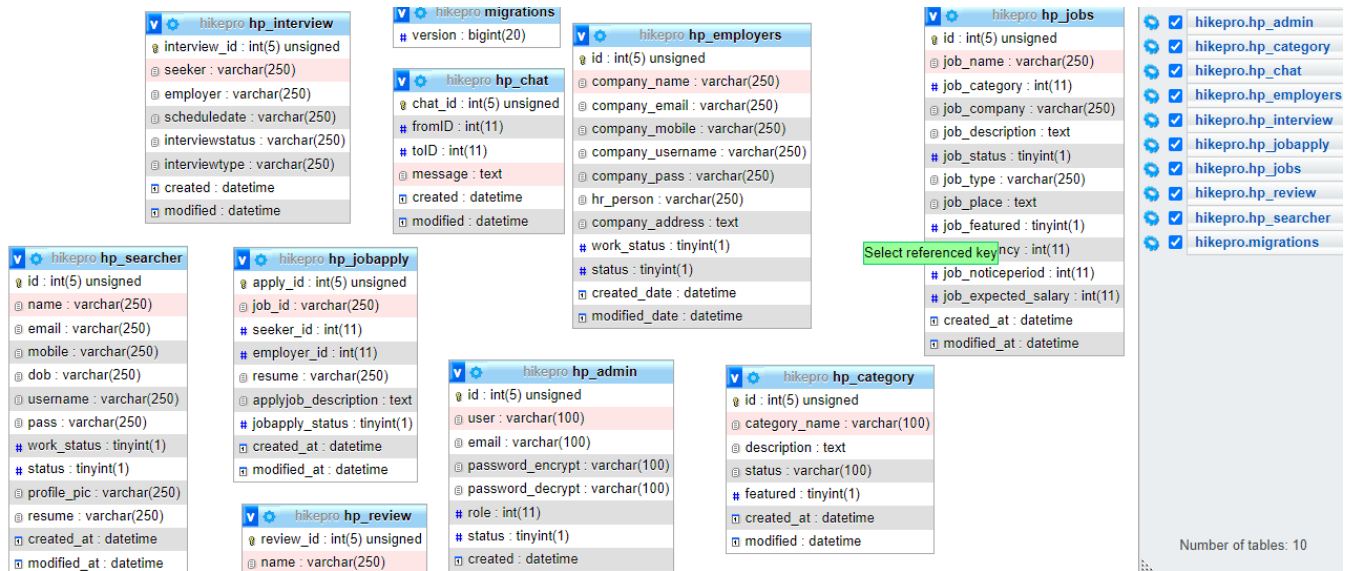


Figure 55: Database Diagram

6 Design Units Impacts

Applications are developed to carry out certain tasks, therefore satisfying the demands of the user. According to a theoretical application development process, the designer gathers all the specifications, develops the system, performs a verification test to make sure his creation adheres to the guidelines, and finally sends it to production. In the majority of product development processes, one of the most challenging challenges is identifying all demands. It could improve your productivity, efficiency, and the worth of your goods and services. It can also reduce costs and risks.

6.1 Functional Area A/Design Unit A

6.1.1 Impacts

The workflow method and design unit for the seeker, administrator, and recruiter controller are adapted to the operational and client needs. The work process includes authentication, updating data and user information, backing up and recovering data, and administering processes.

6.1.2 Requirements

Users are authenticated by the administrator. For access to this web application, neither registration nor login are required. Both the employer and the candidate's progress should be tracked by the admin. Interfaces between systems must be properly designed and dependable on both the inside and outside. A data backup strategy should be implemented for the protection of customer data.

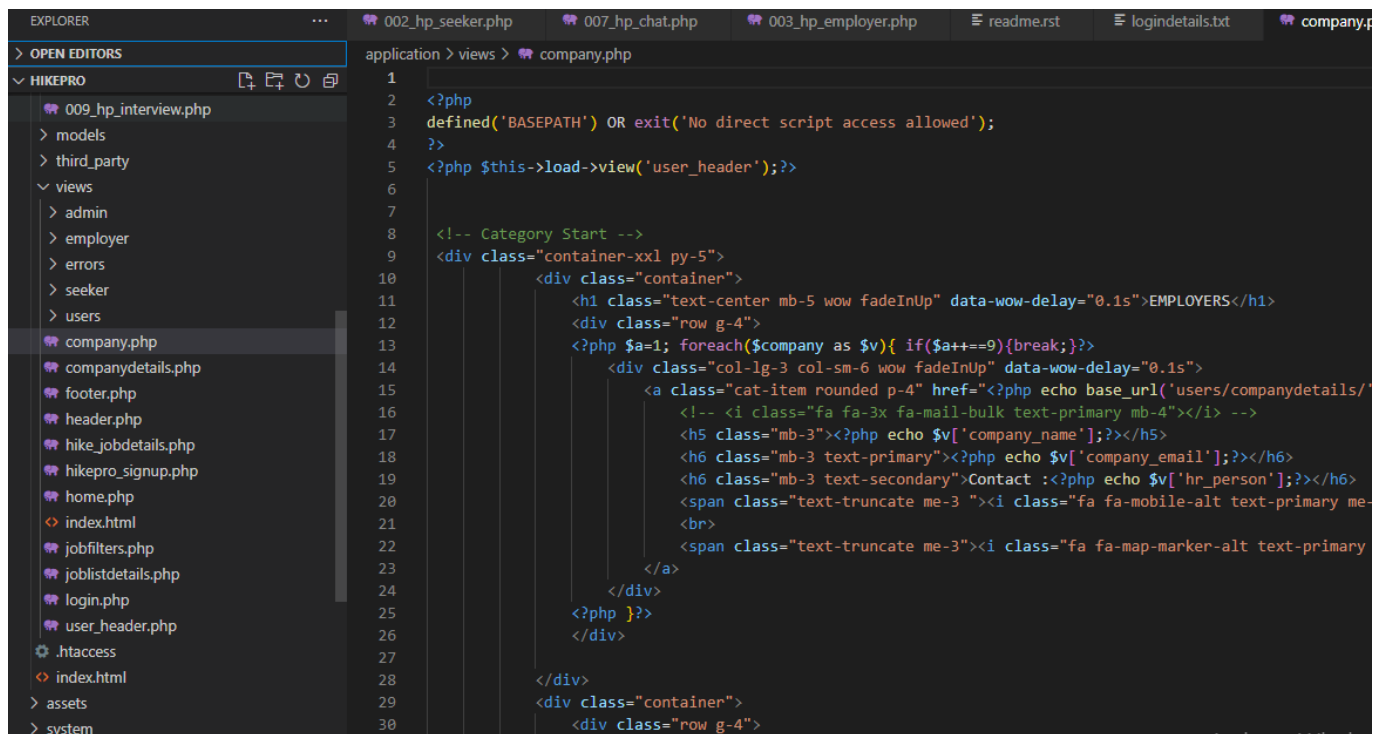
7 Acknowledgements

God Almighty deserves all the praise for the mercies, unwavering goodness, and abounding love He has shown me. I sincerely appreciate Dr. Dae Wook (Wooky) Kim advice and assistance during the development of the project. I'd want to thank my teammates for their assistance in getting the assignment completed.

8 References

- [1] Chiang, C. J. (2012). Enabling successful Collaboration 2.0: A REST-based Web Service and Web 2.0 technology oriented information platform for collaborative product development.
- [2] Firth, R. (n.d.). In *High Powered CVs*.
- [3] Saikia, R. D. (2016). Comparison of Procedural PHP with Codeigniter and Laravel Framework.
- [4] Utomo, H. a. (2014). Implementing Code Igniter Framework in Open Source Mobile Learning Application.

9 Appendices



```
1
2 <?php
3 defined('BASEPATH') OR exit('No direct script access allowed');
4 ?>
5 <?php $this->load->view('user_header');?>
6
7
8 <!-- Category Start -->
9 <div class="container-xxl py-5">
10     <div class="container">
11         <h1 class="text-center mb-5 wow fadeInUp" data-wow-delay="0.1s">EMPLOYERS</h1>
12     <div class="row g-4">
13         <?php $a=1; foreach($company as $v){ if($a++==9){break;}?>
14         <div class="col-lg-3 col-sm-6 wow fadeInUp" data-wow-delay="0.1s">
15             <a class="cat-item rounded p-4" href="<?php echo base_url('users/companydetails/'>
16                 <!-- <i class="fa fa-3x fa-mail-bulk text-primary mb-4"></i> -->
17                 <h5 class="mb-3"><?php echo $v['company_name'];?></h5>
18                 <h6 class="mb-3 text-primary"><?php echo $v['company_email'];?></h6>
19                 <h6 class="mb-3 text-secondary">Contact :<?php echo $v['hr_person'];?></h6>
20                 <span class="text-truncate me-3 "><i class="fa fa-mobile-alt text-primary me-
21                 <br>
22                 <span class="text-truncate me-3"><i class="fa fa-map-marker-alt text-primary
23             </a>
24         </div>
25         <?php }?>
26     </div>
27 </div>
28 </div>
29 <div class="container">
30     <div class="row g-4">
```

Figure 66: Company View

```

EXPLORER
...
007_hp_chat.php 003_hp_employer.php readme.rst logindetails.txt company.php hike_jobdetail

OPEN EDITORS
application > views > hike_jobdetails.php

HIKEPRO
> third_party
  > views
    > admin
    > employer
    > errors
    > seeker
    > users
  company.php
  companydetails.php
  footer.php
  header.php
  hike_jobdetails.php
  hikepro_signup.php
  home.php
  <> index.html
  jobfilters.php
  joblistdetails.php
  login.php
  user_header.php
  .htaccess
  <> index.html
  > assets
  > system
  > uploads
  > user_guide

12 <!-- Jobs Start -->
13 <div class="container-xxl py-5">
14   <div class="container">
15     <h3 class="text-center mb-5 wow fadeInUp" data-wow-delay="0.1s">JOB LIST AND DETAILS INFO
16     <div class="tab-class text-center wow fadeInUp" data-wow-delay="0.3s">
17       <ul class="nav nav-pills d-inline-flex justify-content-center border-bottom mb-5">
18         <li class="nav-item">
19           <a class="d-flex align-items-center text-start mx-3 ms-0 pb-3 active" data-bs-
20             <h6 class="mt-n1 mb-0">All Featured</h6>
21           </a>
22         </li>
23         <li class="nav-item">
24           <a class="d-flex align-items-center text-start mx-3 pb-3" data-bs-toggle="pil
25             <h6 class="mt-n1 mb-0">Experienced</h6>
26           </a>
27         </li>
28         <li class="nav-item">
29           <a class="d-flex align-items-center text-start mx-3 me-0 pb-3" data-bs-toggle
30             <h6 class="mt-n1 mb-0">Freshers</h6>
31           </a>
32         </li>
33       </ul>
34       <div class="tab-content">
35         <div id="tab-1" class="tab-pane fade show p-0 active">
36           <?php foreach($categorydetails['jobs'] as $j){?>
37             <div class="job-item p-4 mb-4">
38               <div class="row g-4">
39                 <div class="col-sm-12 col-md-8 d-flex align-items-center">
40                   <div class="text-start ps-4">

```

Figure 77: Job Details View

```

HIKEPRO
> third_party
  > views
    > admin
    > employer
    > errors
    > seeker
    > users
  company.php
  companydetails.php
  footer.php
  header.php
  hike_jobdetails.php
  hikepro_signup.php
  home.php
  <> index.html
  jobfilters.php
  joblistdetails.php
  login.php
  user_header.php
  .htaccess
  <> index.html
  > assets
  > system
  > uploads
  > user_guide

17     color:#104d85 !important;
18   }
19 </style>
20
21 <div class="container-fluid mb-5 p-5">
22   <div class="row">
23     <aside class="col-md-3">
24       <h4 class="text-uppercase text-blue mb-3 wow fadeInUp" data-wow-delay="0.1s">Apply To Search</h4>
25     </aside>
26     <div class="card">
27       <form role="form" method="post" action="<?php echo base_url('main/search'); ?>">
28         <article class="filter-group">
29           <header class="card-header">
30             <a href="#" data-toggle="collapse" data-target="#collapse_1" aria-expanded="true" class="dis
31               <i class="icon-control fa fa-chevron-down mr-1"></i>
32               <h6 class="title">Category</h6>
33             </a>
34           </header>
35           <div class="filter-content collapse show" id="collapse_1" style="">
36             <div class="card-body">
37               <select class="form-select border-1" name="category" >
38                 <option value="">Category</option>
39                 <?php foreach($getAllTablelistdetails['hp_category'] as $c){?>
40                 <option <?php echo $s=($searcharray['category']==$c['id']?'selected':'');?> value="<?php
41                 <?php ?>
42               </select>
43             </div>
44           </div>
45         </article>

```

Figure 78: Job Filter View

```

EXPLORER
OPEN EDITORS
HIKEPRO
  > 009_hp_interview.php
  > models
  > third_party
  > views
    > admin
      dashboard.php
      header.php
      profile.php
      update_category.php
      user_login.php
    > employer
    > errors
    > seeker
    > users
  company.php
  companydetails.php
  footer.php
  header.php
  hike_jobdetails.php
  hikepro_signup.php
  home.php
  index.html
  jobfilters.php
  joblistdetails.php

application > views > admin > update_category.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?>
4 <?php $this->load->view('admin/header');?>
5 <?php
6     if($categoryDetails[0]['status']=='Active'){
7         $s='selected';
8         $a='';
9     }else if($categoryDetails[0]['status']=='inactive'){
10        $s='';
11        $a='selected';
12    }
13
14    if($categoryDetails[0]['featured']==1){
15        $f='selected';
16        $g='';
17    }else if($categoryDetails[0]['featured']==0){
18        $f='';
19        $g='selected';
20    }
21
22 ?>
23 <div class="container-xxl py-5">
24     <div class="container">
25         <h1 class="text-center mb-5 wow fadeInUp" data-wow-delay="0.1s"><span style="color:#9f3264">UPDATE<
26         <div class="row">
27             <div class="col-12">
28             <div class="row gy-5">
29

```

Figure 99: Update Category View

```

EXPLORER
OPEN EDITORS
HIKEPRO
  > third_party
  > views
    > admin
      dashboard.php
      header.php
      profile.php
      update_category.php
      user_login.php
    > employer
      candidate_joblist.php
      edit_joblist.php
      employer_dashboard.php
      employer_home.php
      employer_joblist.php
      employer_profile.php
    > errors
    > seeker
    > users
  company.php
  companydetails.php
  footer.php
  header.php
  hike_jobdetails.php
  hikepro_signup.php

application > views > employer > employer_joblist.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?>
4 <?php
5 $this->load->view('user_header');
6
7 $user_session_data = $this->session->userdata('UserE');
8 ?>
9 <link rel="stylesheet" href="<?php echo base_url('assets/css/jquery.dataTables.min.css');?>">
10 <link rel="stylesheet" href="<?php echo base_url('assets/css/jquery.dataTables.css');?>">
11
12 <div class="container-fluid mt-1">
13     <div class="row flex-nowrap">
14         <input type="hidden" name="editid" id="editid" value="<?php echo $user_session_data['user_id'];?>">
15         <div class="col-auto col-md-3 col-xl-2 px-sm-2 px-0 bg-dark">
16             <div class="d-flex flex-column align-items-center align-items-sm-start px-3 pt-2 text-white min-
17                 <a href="<?php echo base_url('users/employee_home/'.$user_session_data['user_id'].'');?>">
18                     <span class="fs-5 d-none d-sm-inline text-white"> Dashboard</span>
19                 </a>
20                 <ul class="nav nav-pills flex-column mb-sm-auto mb-0 align-items-center align-items-sm-start
21                     <li class="nav-item">
22                         <a href="<?php echo base_url('users/employee_dashboard');?>" class="nav-link align-
23                         <i class="fa fa-plus text-info mb-3"></i> <span class="ms-1 d-none d-sm-inline text-
24                         </a>
25                     </li>
26                     <li class="nav-item">
27                         <a href="<?php echo base_url('users/profile/e/'.$user_session_data['user_id'].'');?>
28                         <i class="fa fa-edit text-info mb-3"></i> <span class="ms-1 d-none d-sm-inline text-
29

```

Figure 20: Employer Job List View

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?>
4 <?php
5 $this->load->view('user_header');
6 $user_session_data = $this->session->userdata('UserJ');
7 ?>
8
9
10 <div class="container-fluid mt-1">
11     <div class="row flex-nowrap">
12
13         <div class="col-auto col-md-3 col-xl-2 px-sm-2 px-0 bg-dark">
14             <div class="d-flex flex-column align-items-center align-items-sm-start px-3 pt-2 text-white min-
15                 <a href="<?php echo base_url('users/seeker_home/'.$user_session_data['user_id'].''); ?>" cla
16                 <span class="fs-5 d-none d-sm-inline text-white"> Dashboard</span>
17             </a>
18             <ul class="nav nav-pills flex-column mb-sm-auto mb-0 align-items-center align-items-sm-start
19                 <li class="nav-item">
20                     <a href="<?php echo base_url('users/dashboard'); ?>" class="nav-link align-middle p
21                     <i class="fa fa-search text-info mb-3"></i> <span class="ms-1 d-none d-sm-inline tex
22                     </a>
23                 </li>
24                 <li class="nav-item">
25                     <a href="<?php echo base_url('users/profile/j/'.$user_session_data['user_id'].''); ?
26                     <i class="fa fa-edit text-info mb-3"></i> <span class="ms-1 d-none d-sm-inline tex-
27                         Edit Profile</span>
28                     </a>
29                 </li>
30                 <li class="nav-item">

```

Figure 101: Seeker Home View

```

application > migrations > 004_hp_category.php
1 <?php defined('BASEPATH') OR exit('No direct script access allowed');
2 class Migration_Hp_category extends CI_Migration {
3     public function up() {
4
5         $this->dbforge->add_field(array(
6             'id' => array(
7                 'type' => 'INT',
8                 'constraint' => 5,
9                 'unsigned' => TRUE,
10                'auto_increment' => TRUE
11            ),
12            'category_name' => array(
13                'type' => 'VARCHAR',
14                'constraint' => '100'
15            ),
16            'description' => array(
17                'type' => 'TEXT',
18            ),
19            'status' => array(
20                'type' => 'VARCHAR',
21                'constraint' => '100',
22                'default' => 'Active'
23            ),
24            'featured' => array(
25                'type' => 'TINYINT',
26                'constraint' => '1',
27                'default' => 1
28            ),
29            'created_at DATETIME DEFAULT current_timestamp',
30            'modified DATETIME DEFAULT current_timestamp on update current_timestamp',

```

Figure 22: Migrations

```

EXPLORER
... jobfilters.php update_category.php employer_joblist.php seeker_home.php 004_hp_category.php Users.php
OPEN EDITORS
application > controllers > Users.php
HIKEPRO
  application
  > cache
  > config
  > controllers
    Admin.php
    index.html
    Main.php
    Migration.php
    Users.php
  > core
  > helpers
  > hooks
  > language
  > libraries
  > logs
  > migrations
    001_hp_admin.php
    002_hp_seeker.php
    003_hp_employer.php
    004_hp_category.php
    005_hp_jobs.php
    006_hp_jobapply.php
    007_hp_chat.php
    008_hp_review.php
    009_hp_interview.php
  > migrations

4 class Users extends CI_Controller {
5
6     public function __construct(){
7         parent::__construct();
8         $this->load->helper('url');
9         $this->load->library('form_validation');
10        $this->load->model('admin/admin_model');
11        $this->load->model('main_model');
12    }
13
14    //TO GET THE LOGIN CREDENTIALS AND CHECK VALID USER
15    public function login(){
16
17        $userJ = $this->session->userdata('UserJ');
18        $userE = $this->session->userdata('UserE');
19        if($userJ){
20            redirect('users/dashboard');
21        }elseif($userE){
22            redirect('users/employee_dashboard');
23        }else{
24            $data['islogin']= 0;
25            $this->load->view("login",$data);
26        }
27    }
28
29
30
31    //TO SHOW THE EMPLOYEE HOME PAGE INFO
32    public function employee_home($id){
33        if($this->session->userdata('UserE')){
34            $data['islogin']= 1;
35            $sessiondata = array(
36                'user_id' => $this->session->userdata('UserE')['id'],
37                'username' => $this->session->userdata('UserE')['username'],
38                'user_email' => $this->session->userdata('UserE')['email'],
39                'logged_in' => TRUE,
40                'login_type' => 'Employee'
41            );
42            $this->session->set_userdata('User', $sessiondata);
43            redirect(base_url('users/employee_home/'.$id));
44        }else{
45            $data['islogin']= 0;
46            $this->load->view("employee_home",$data);
47        }
48    }
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Figure 113: Users Controller

```

EXPLORER
... update_category.php employer_joblist.php seeker_home.php 004_hp_category.php Users.php
OPEN EDITORS
application > controllers > Admin.php
HIKEPRO
  application
  > cache
  > config
  > controllers
    Admin.php
    index.html
    Main.php
    Migration.php
    Users.php
  > core
  > helpers
  > hooks
  > language
  > libraries
  > logs
  > migrations
    001_hp_admin.php
    002_hp_seeker.php
    003_hp_employer.php
    004_hp_category.php
  > migrations

19 public function login(){
20
21     $user = $this->input->post('user_name');
22     $pass = $this->input->post('user_password');
23
24     $res = $this->admin_model->login_user($user,$pass);
25     $data['res_val'] = $res;
26
27     if($data['res_val']){
28         $data['islogin']= 1;
29         $sessiondata = array(
30             'user_id' => $data['res_val'][0]['id'],
31             'username' => $data['res_val'][0]['user'],
32             'user_email' => $data['res_val'][0]['email'],
33             'logged_in' => TRUE,
34             'login_type' => 'Admin'
35         );
36         $this->session->set_userdata('User', $sessiondata);
37         redirect(base_url('admin/dashboard'));
38     }else{
39         $data['islogin']= 0;
40         $this->load->view("admin/user_login");
41     }
42 }
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Figure 124: Main Controller