T.C. ADANA ALPARSLAN TÜRKEŞ UNIVERSITY
FACULTY OF COMPUTER AND INFORMATICS
DEPARTMENT OF COMPUTER ENGİNEERING

CEN224

DATABASE MANAGEMENT SYSTEMS PROJECT

Prepared by

Eren Karadeniz

Mahmut Enes Sağlam

Ulaş Deniz Çakmazel

Miray Çalık

Instructor

Assoc. Prof. Dr. Mümine KAYA KELEŞ

June 6 , 2023

ADANA

CONTENTS

1.ABSTRACT 3

2.INTRODUCTION 3

2.1 Purpose of Student Information System Database 3

2.2 Structure of Relational Database 4

2.3 Database Users and User Interfaces 4

2.4 Software Requirements Specification 4

2.5 ER Diagram 8

2.6 Relational Database 8

3. REPORT 9

3.1 Employer 9

3.2 Job Seeker 9

3.3 Tables 9

3.4 Testing and Security 11

4. USER INTERFACE 11

5. SQL Queries 19

6. Conclusion 19

7. References 20

1. ABSTRACT

In this project, a Job Recruitment database was created. ER model, relational model and algebra, SQL language were used. We wrote in MSSQL and PHP environment.

2. INTRODUCTION

A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient. Database systems are designed tomanage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results. Because information is so important in most organizations, computer scientists have developed a large body of concepts and techniques for managing data. These concepts and techniques form the focus of this book. This chapter briefly introduces the principles of database systems.

2.1. Purpose of Job Recruitment Database

The Job Recruitment Database plays a vital role in supporting an online job recruitment system. It serves as the backbone of the system, storing and organizing essential data related to employers, job seekers, job postings, applications, cities, and districts. The main purpose of the database is to make the recruitment process easier by providing a structured and efficient storage system for all relevant information.

The database allows employers to create job posting, manage them, and handle received applications. Job seekers can create applications to active job posting. In addtion employers and jobseekers can update and delete their acount and information that created by related account. By maintaining the necessary data relationships and using appropriate table structures, the Job Recruitment Database ensures smooth integration between different parts of the system.

In simpler terms, the Job Recruitment Database is like the heart of the job recruitment system. It stores and organizes all the important data, making it easier for employers and job seekers to interact with the system and find a job. It ensures that job postings and applications are managed effectively, leading to successful job placements.

Key features and functionalities of the Job Recruitment Database include:
Storing employer information such as company details, contact information, and login credentials.

- Managing job posting, including job titles, descriptions, posting dates, status, and working types.
- Storing job seeker information, including personal details, contact information, and login credentials.
- Recording job applications, including the associated job posting, job seeker, application date, and status.
- Filtering for accurate searches .

Overall, the Job Recruitment Database plays a crucial role in supporting the job recruitment system, providing a strong and reliable foundation for the storage, retrieval, and management of data related to employers, job seekers, job postings, and applications.

2.2. Structure of Relational Databases

Cities(cityID, city_name)

Districts(districtID, district_name, cityID (FK))

Users(userID, gender, birth_date, password, email, other_adress, district, city, name, cityID(FK), districtID (FK))

Users_phone(phone, userID (FK))

Companies(companyID, company_name, website, email, city, district, other_adress)

Companies_phone(phone, companyID (FK))

Jobs(jobID, job_title, job_description, listing_date, listing_status, working_type, companyID (FK))

Applications(application_date, application_status, userID(FK), jobID(FK))

2.3. Database Users and User Interfaces

Naive users in the context of job recruitment are individuals who interact with the system through predefined user interfaces. These users typically utilize forms interfaces, where they can fill in the required fields.

For instance, let's consider a job seeker who wishes to apply for a position on an online job portal. The job seeker would connect to a web application that is hosted on a server. Upon logging in, the application provides the active jobs from the database server that related job seeker can apply to.

If jobseeker do not have an account they can create new account to access application. The application have a register form that the job seeker can enter their relevant information, such as personal details, location and email. After they fill the form correctly they can create account to apply jobs.

In summary, naive users that uses job recruitment application to interact with the system through user interfaces use forms to input their information and access information or job posting from the recruitment database.

2.4. Software Requirements Specification

Frontend- HTML, CSS, Java Script, JQuery

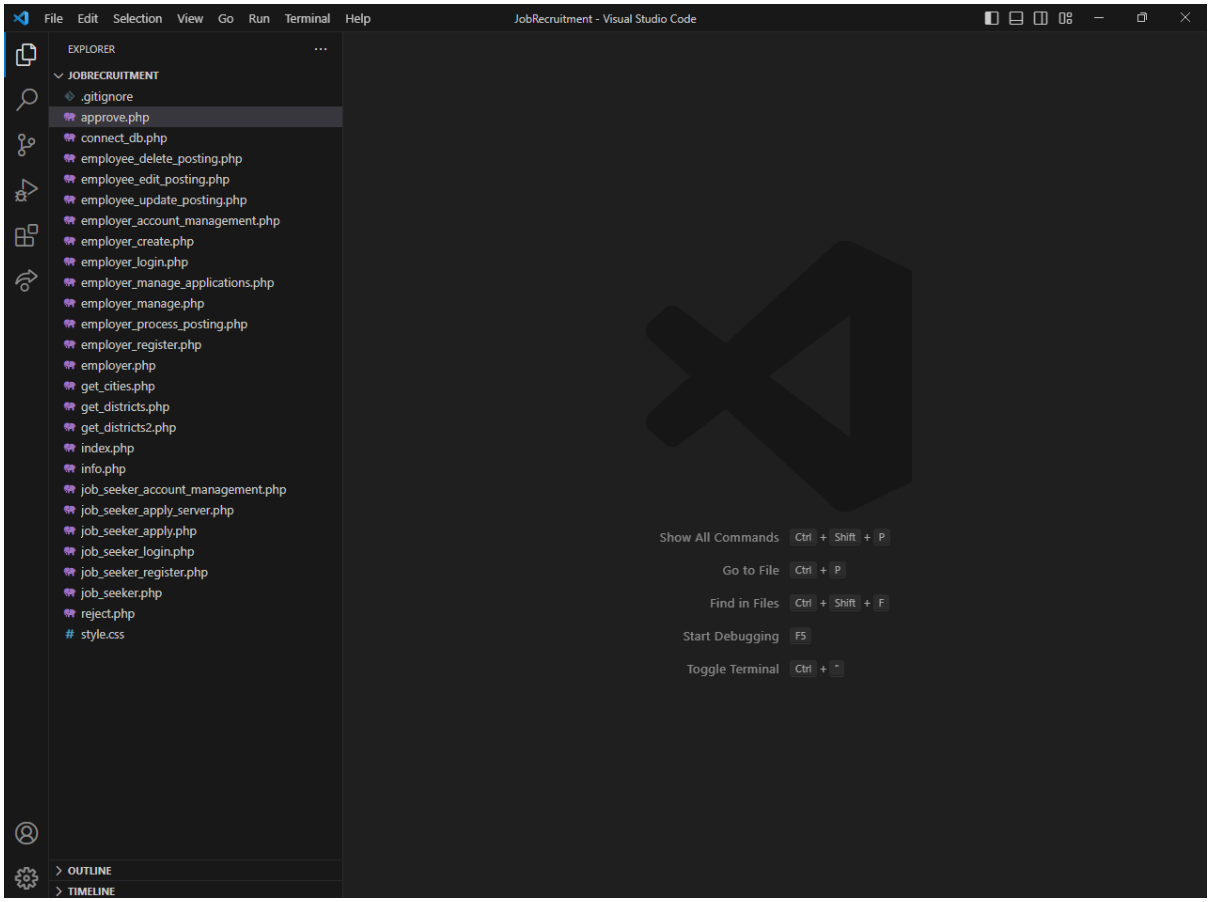
Backend- PHP

Operating System: Windows 10/11

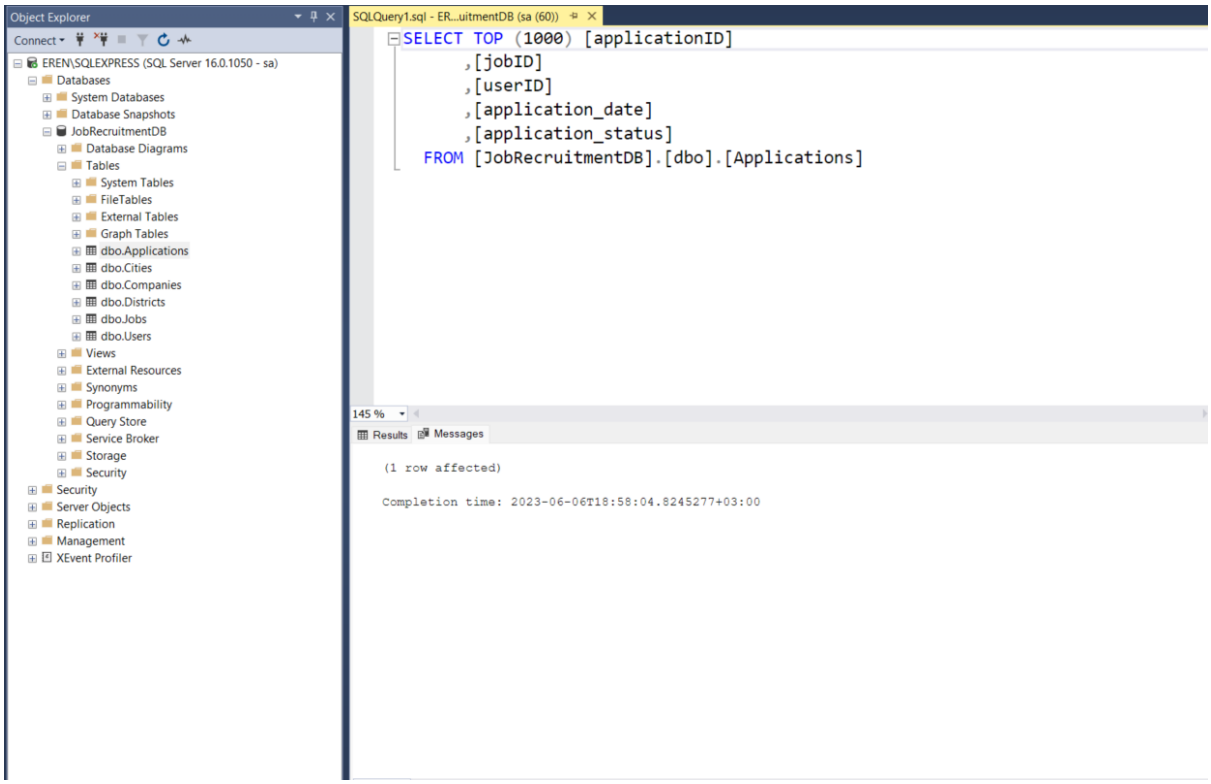
Google Chrome/Microsoft Edge

XAMPP (Version-8.2.4)

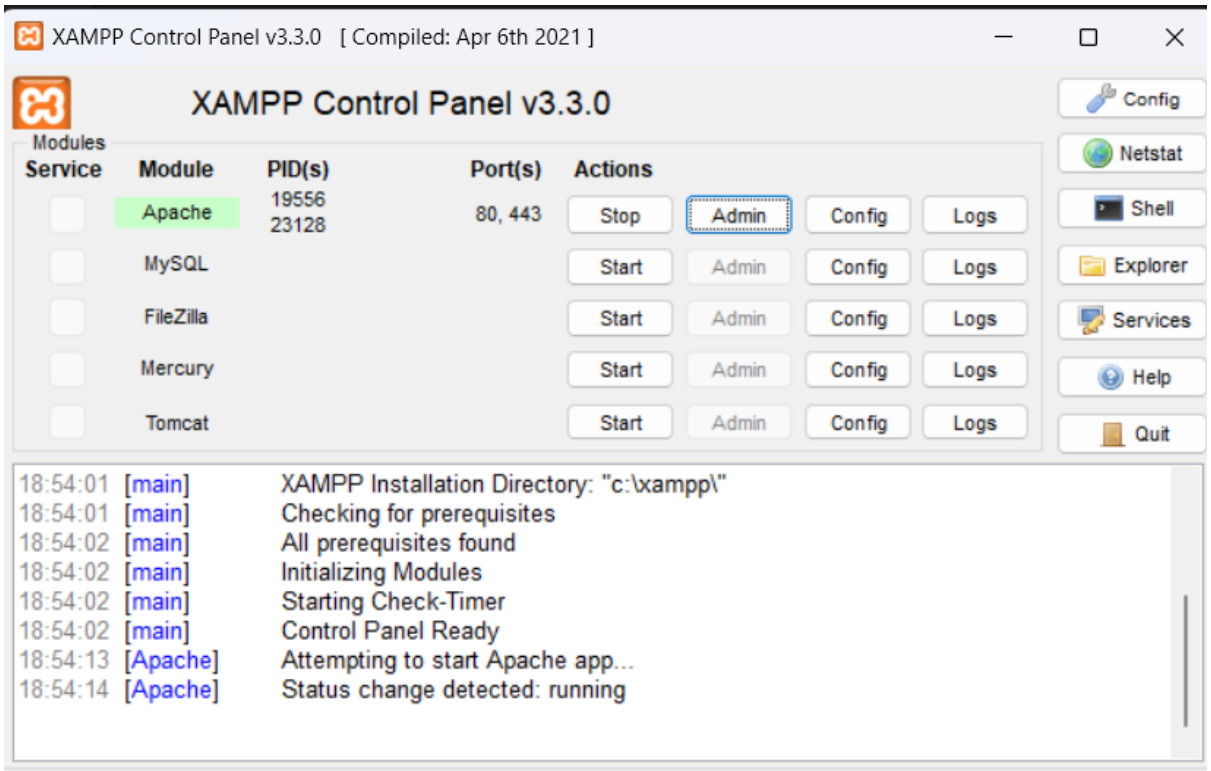
Visual Studio Code Editor (user interface)



Microsoft Visual Studio Code Editor



Microsoft Sql Server Management Studio

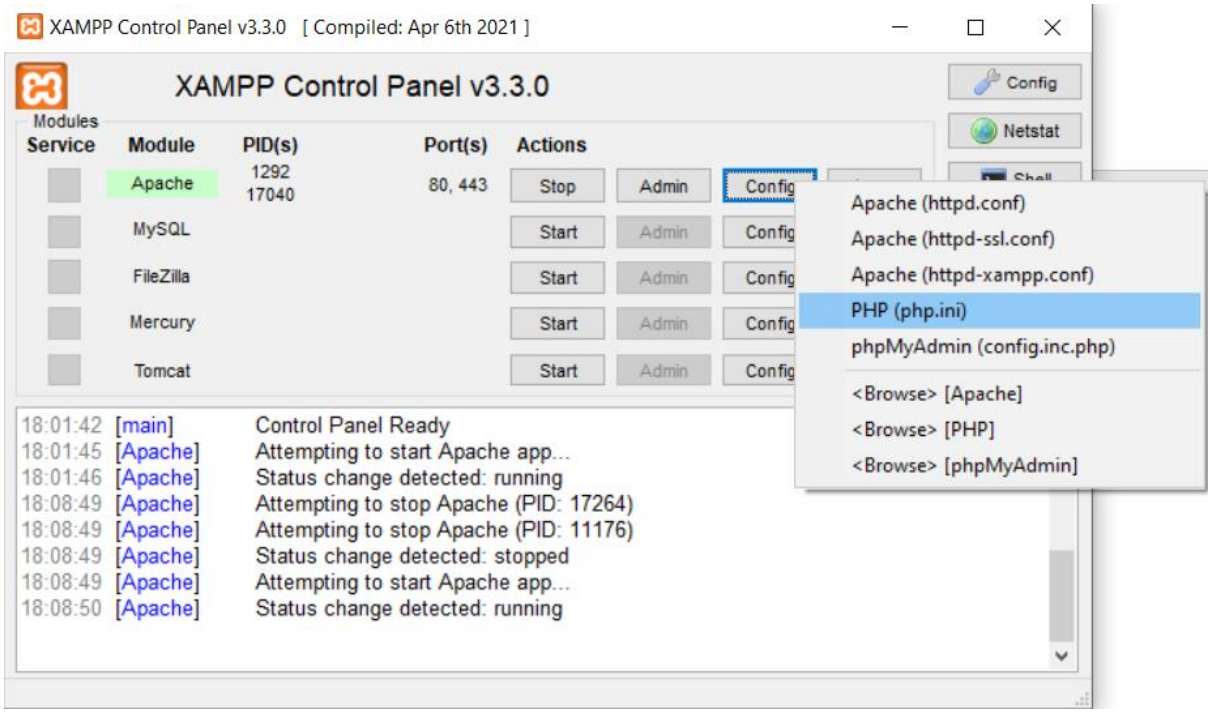


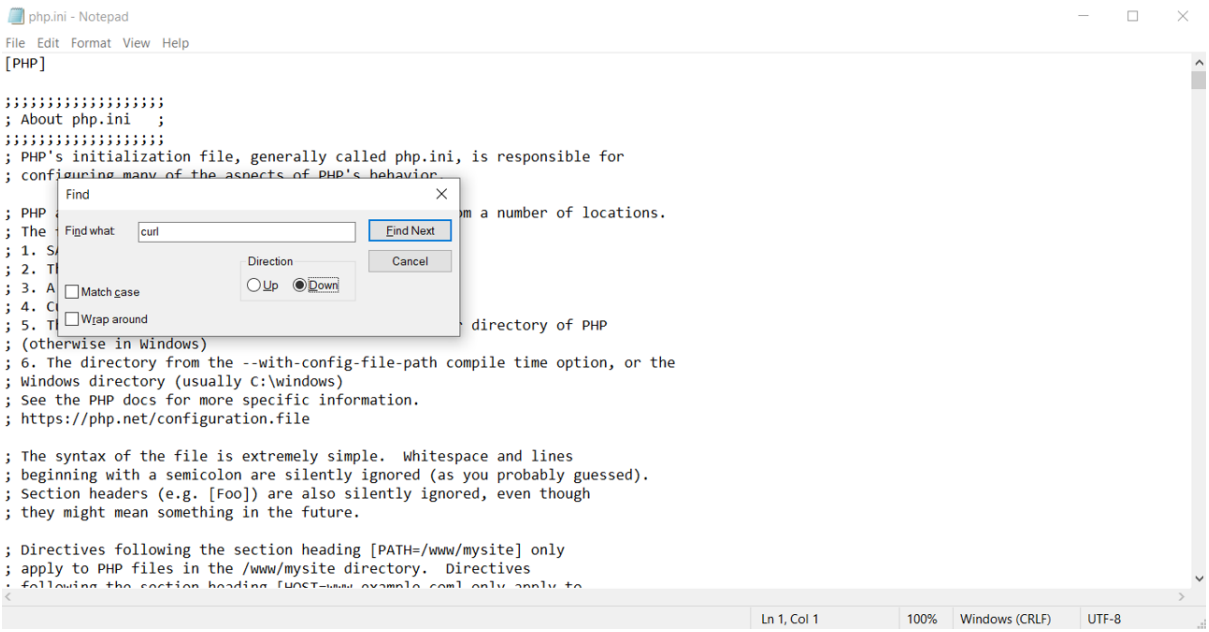
XAMPP (Version-8.2.4)

NOTE: In order to work with XAMMP, we should add these files to xampp -> php -> ext folder:

php_sqlsrv_82_ts_x64.dll

php_sqlsrv_82_nts_x64.dll



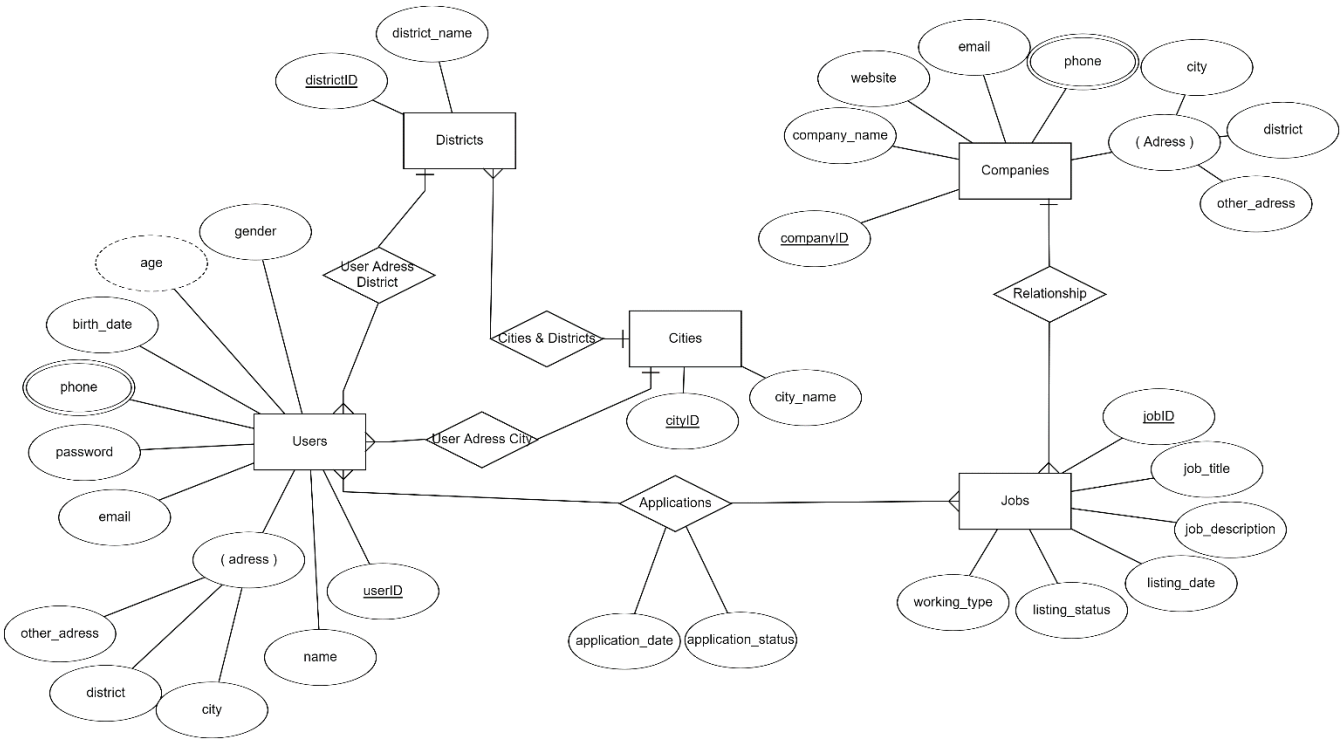


We should also add these texts to this part:

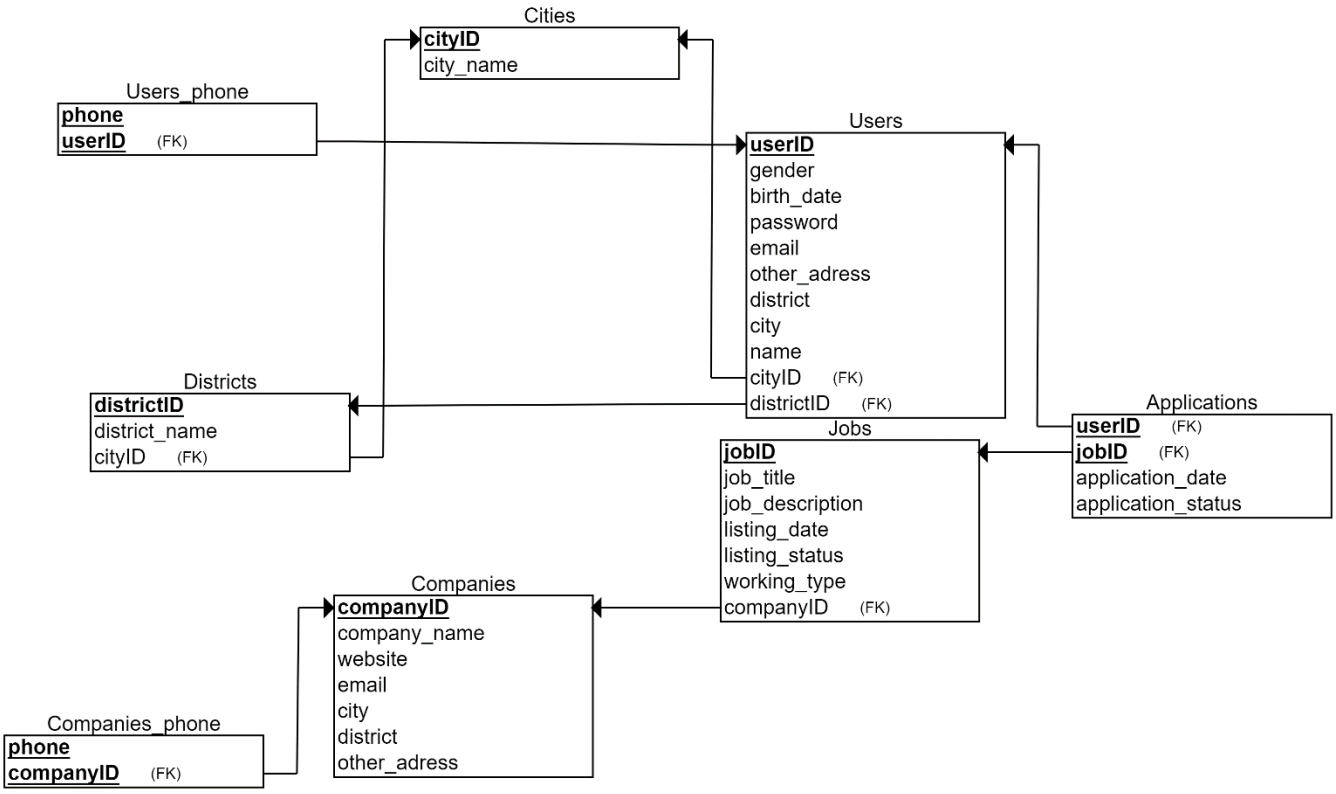
extension=sqlsrv_82_nts_x64

extension=sqlsrv_82_ts_x64

2.5. ER Diagram



2.6. Relational Tables



3. REPORT

Job Recruitment System

Introduction The Job Recruitment System is a web-based application developed using PHP and Microsoft SQL Server (MSSQL). It provides a platform for employers and job seekers to connect and facilitate the recruitment process. This technical report outlines the menu structure and data structure of the project.

Menu Structure The menu structure is designed to provide a user-friendly interface for both employers and job seekers. The main menu consists of two sections: Employer Login and Job Seeker Login.

3.1. Employer

Login (Successful): Allows employers to log in to their accounts successfully.

Create Job Posting: Enables employers to create new job postings.

Manage Job Posting: Allows employers to edit existing job postings.

Manage Applications: Provides employers with the ability to manage received job applications.

Manage Account: Allows employers to manage their account settings.

Logout: Logs out the employer and returns to the main menu.

3.2. Job Seeker

Login (Successful): Allows job seekers to log in to their accounts successfully.

Create Application: Enables job seekers to create and submit job applications.

Manage Account: Allows job seekers to manage their account settings.

Logout: Logs out the job seeker and returns to the main menu.

3.3. Tables

Data Structure and Tables The project utilizes a Microsoft SQL Server database named "JobRecruitmentDB." The database consists of several tables that store the necessary information for the application.

Cities Table

cityID (int): A unique identifier for each city.

city_name (nvarchar(50)): Stores the name of the city.

Districts Table

districtID (int): A unique identifier for each district.

cityID (int): Foreign key referencing the city in which the district is located.

district_name (varchar(100)): Stores the name of the district.

Users Table

userID (int): A unique identifier for each user.

cityID (int): Foreign key referencing the city associated with the user's address.

districtID (int): Foreign key referencing the district associated with the user's address.

name (varchar(100)): Stores the user's first name.

surname (varchar(100)): Stores the user's last name.

password (varchar(100)): Stores the user's password.

email (varchar(100)): Stores the user's email address.

phone (varchar(100)): Stores the user's phone number.

address (text): Stores the user's address.

birth_date (date): Stores the user's birth date.

gender (varchar(10)): Stores the user's gender.

Companies Table

companyID (int): A unique identifier for each company.

cityID (int): Foreign key referencing the city where the company is located.

districtID (int): Foreign key referencing the district where the company is located.

company_name (varchar(100)): Stores the name of the company.

website (varchar(150)): Stores the company's website URL.

email (varchar(100)): Stores the company's email address.

password (int): Stores the company's password.

phone (varchar(100)): Stores the company's phone number.

address (text): Stores the company's address.

Jobs Table

jobID (int): A unique identifier for each job posting.

companyID (int): Foreign key referencing the company that posted the job.

cityID (int): Foreign key referencing the city where the job is located.

districtID (int): Foreign key referencing the district where the job is located.

job_title (varchar(100)): Stores the title of the job.

job_description (text): Stores the description of the job.

listing_date (date): Stores the date when the job posting was created.

listing_status (varchar(10)): Stores the status of the job posting.

working_type (varchar(20)): Stores the type of working arrangement for the job.

Applications Table

applicationID (int): A unique identifier for each job application.

jobID (int): Foreign key referencing the job to which the application is submitted.

userID (int): Foreign key referencing the user who submitted the application.

application_date (int): Stores the date when the application was submitted.

application_status (varchar(50)): Stores the status of the application.

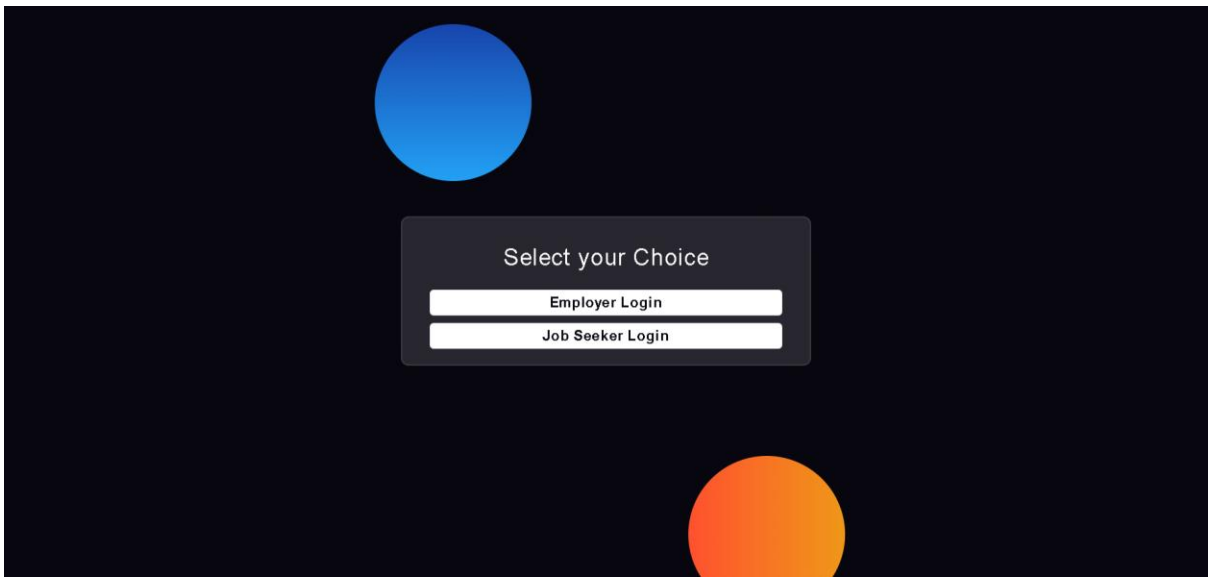
3.4. Testing and Security

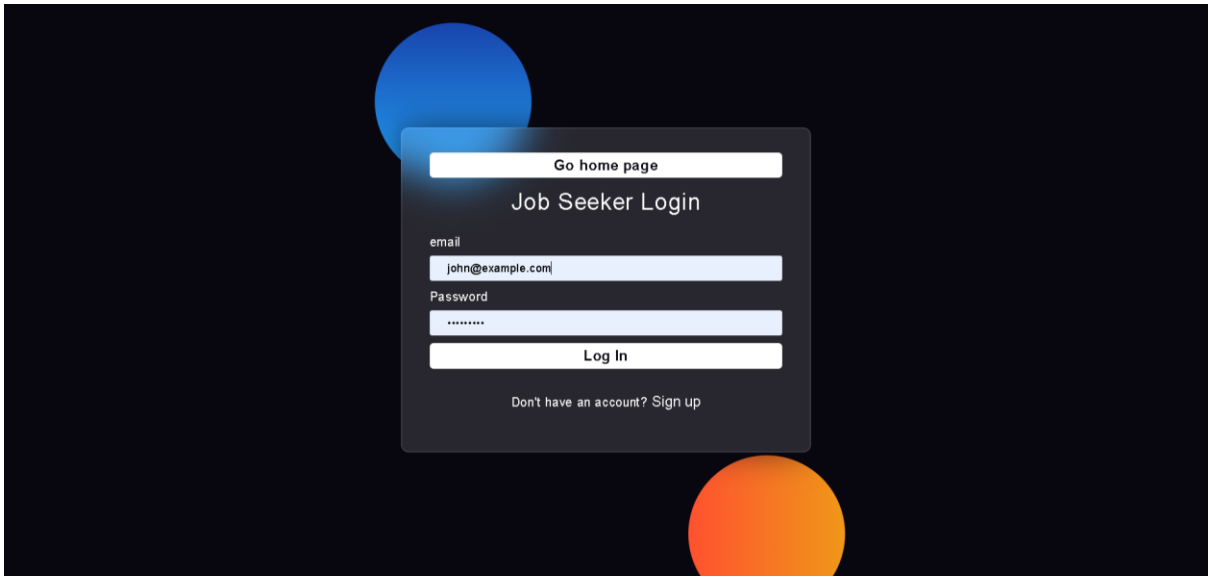
For security, the security.php page has been created. This page contains two securities.

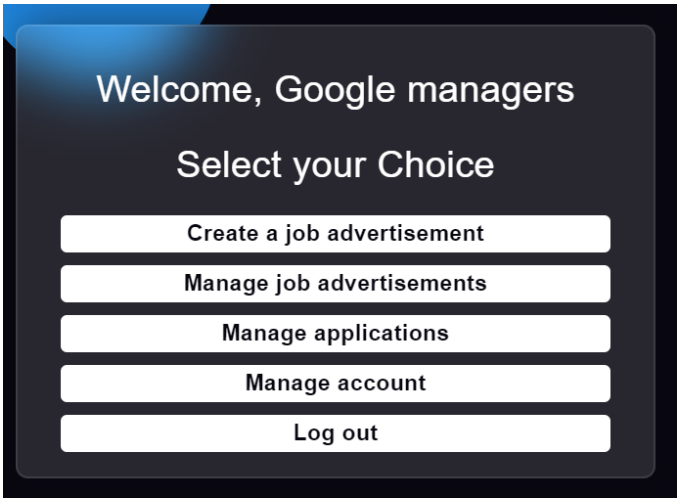
1) It encrypts the password with an encryption algorithm we wrote and sends it to the database.

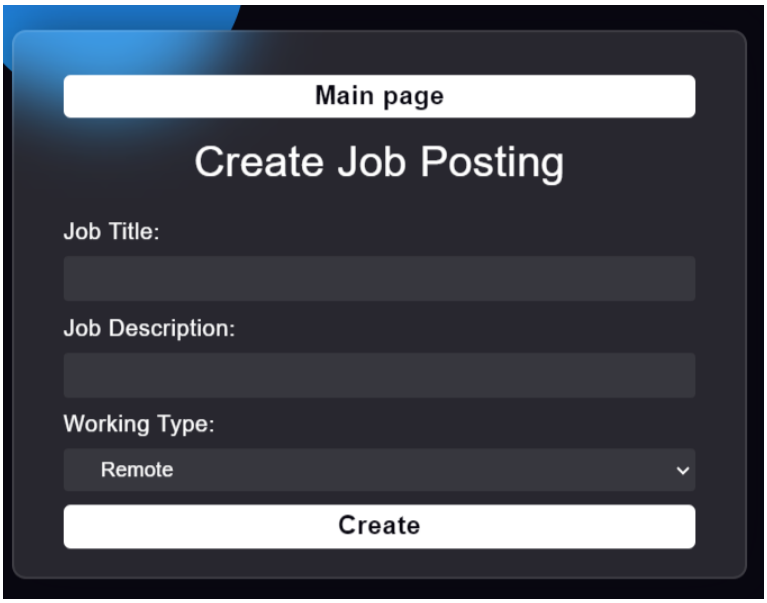
2) It prevents page access via URL to the pages accessed by members without login.

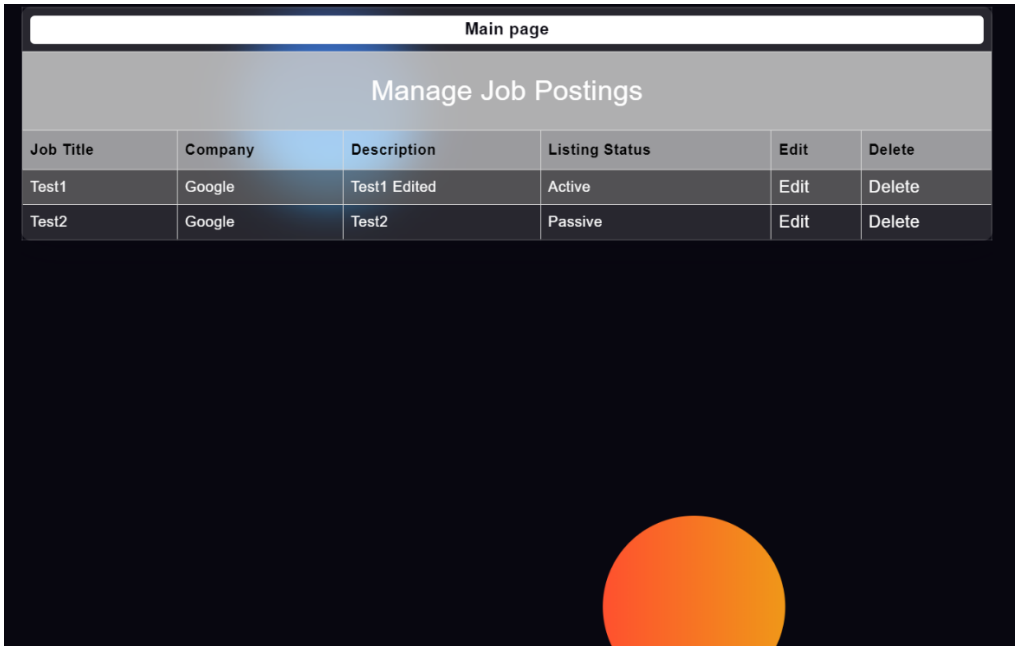
4. USER INTERFACE

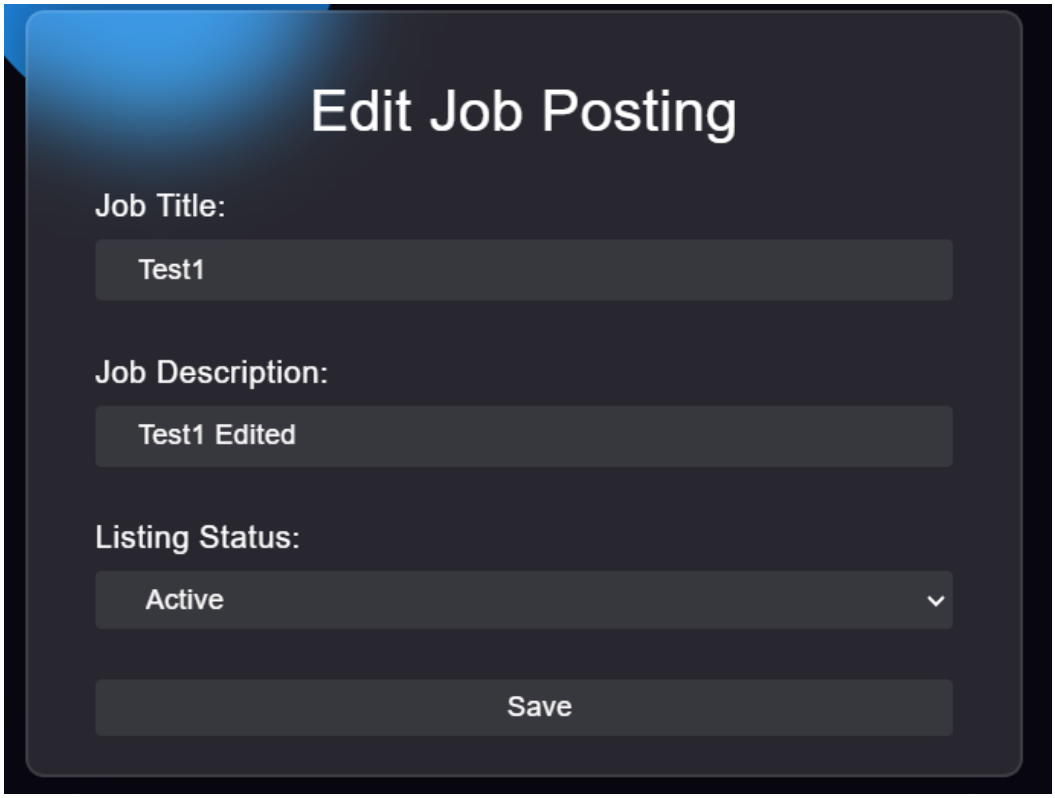





Welcome, Google managers
Select your Choice
Create a job advertisement
Manage job advertisements
Manage applications
Manage account
Log out


Main page
Create Job Posting
Job Title:
Job Description:
Working Type:
Remote
Create


Main page
Manage Job Postings
Job Title
Company
Description
Listing Status
Edit
Delete
Test1
Google
Test1 Edited
Active
Edit
Delete
Test2
Google
Test2
Passive
Edit
Delete


Edit Job Posting
Job Title:
Test1
Job Description:
Test1 Edited
Listing Status:
Active
Save

Main page

Manage Job Applications

Filter Applications: All | Approved | Rejected | Pending

Applicant Name	Job Title	Date Applied	Action	Application Status
Eren Karadeniz	Test2	2023-06-08	Approve | Reject	Rejected
Eren Karadeniz	Test1	2023-06-08	Approve | Reject	Pending
Deniz Çakmazel	Test2	2023-06-08	Approve | Reject	Approved
Deniz Çakmazel	Test1	2023-06-08	Approve | Reject	Rejected

Main page

Manage Job Applications

Filter Applications: All | Approved | Rejected | Pending

Applicant Name	Job Title	Date Applied	Action	Application Status
Eren Karadeniz	Test2	2023-06-08	Approve | Reject	Rejected
Deniz Çakmazel	Test1	2023-06-08	Approve | Reject	Rejected

Account Management

Company Name

Google

Website

google.com

Email

company@google.com

Password

••••••••••

Phone

5511379055

City ADANA District SEYHAN

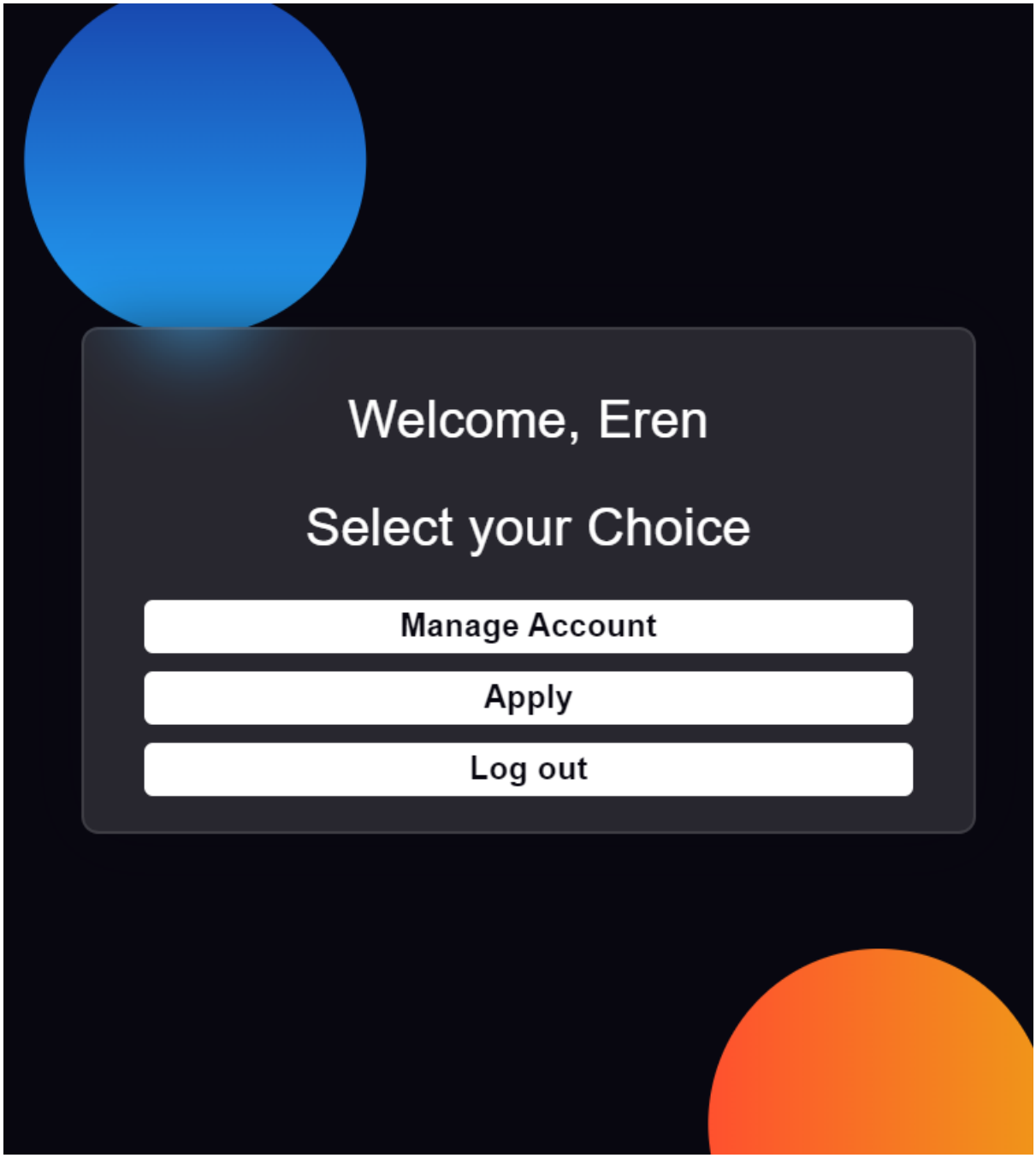
Address:

Bahçelievler

Save

Main page

Delete Account


Welcome, Eren
Select your Choice
Manage Account
Apply
Log out

Account Management

Name Eren Surname Karadeniz

Email

eren@hotmail.com

Password

••••••••

Gender : Male Female

Phone

05545454544

Birth Date

08/06/2023

City ANTALYA District KEMER

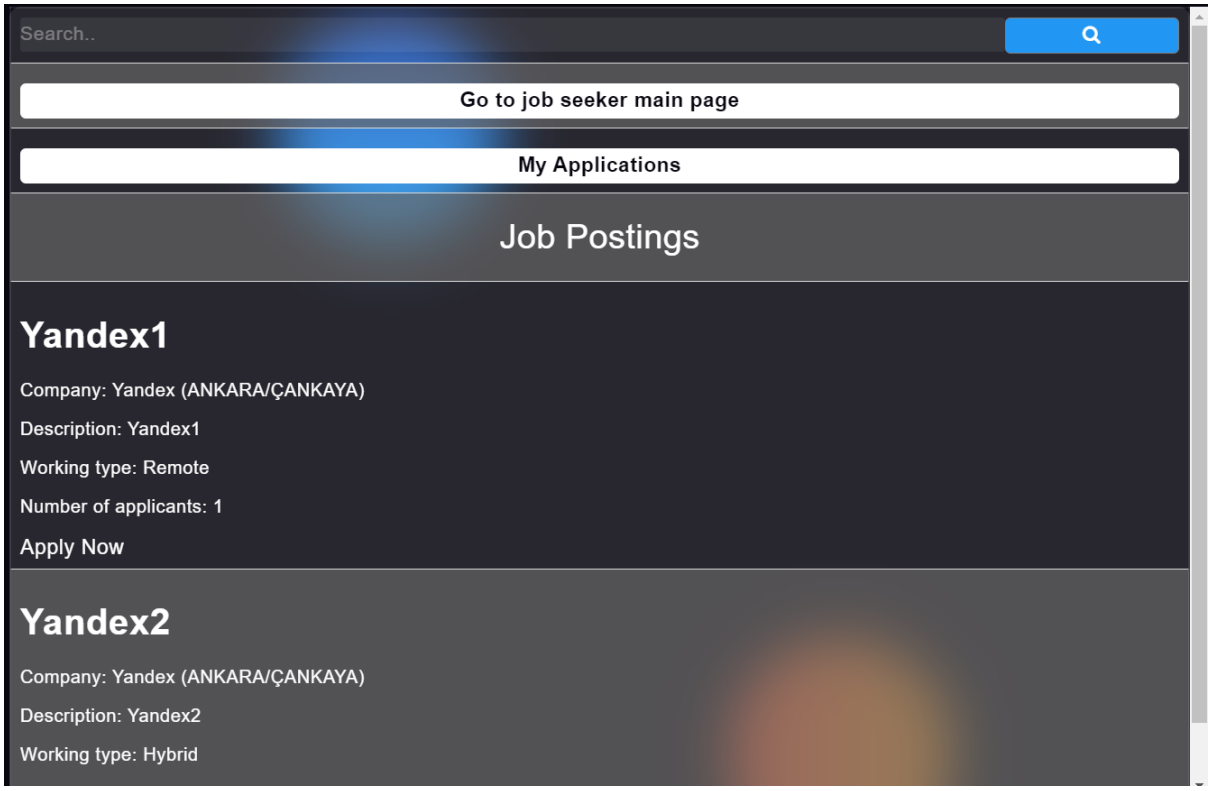
Address:

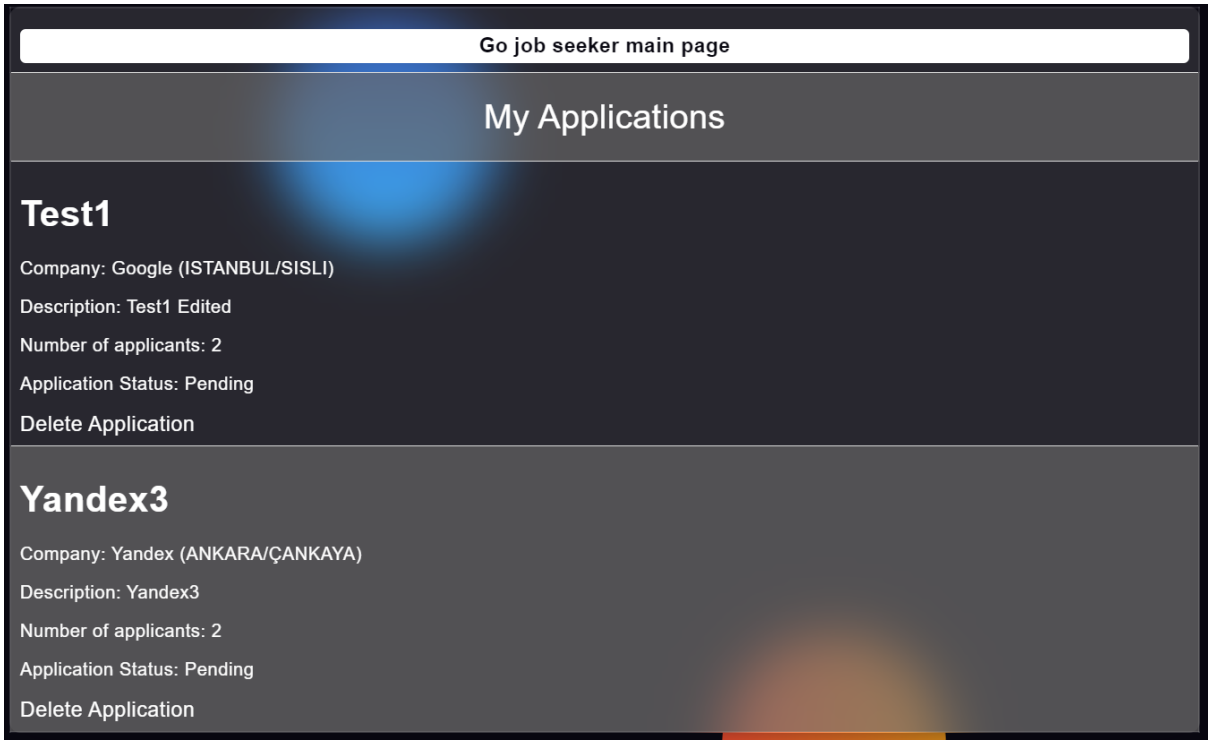
Bahçelievler

Save

Main page

Delete Account


Search..
Go to job seeker main page
My Applications
Job Postings
Yandex1
Company: Yandex (ANKARA/ÇANKAYA)
Description: Yandex1
Working type: Remote
Number of applicants: 1
Apply Now
Yandex2
Company: Yandex (ANKARA/ÇANKAYA)
Description: Yandex2
Working type: Hybrid


Go job seeker main page
My Applications
Test1
Company: Google (ISTANBUL/SISLI)
Description: Test1 Edited
Number of applicants: 2
Application Status: Pending
Delete Application
Yandex3
Company: Yandex (ANKARA/ÇANKAYA)
Description: Yandex3
Number of applicants: 2
Application Status: Pending
Delete Application

5. SQL Queries

Select Example

```
SELECT c.companyID, c.cityID, c.districtID, c.company_name, c.website,
c.email, c.password, c.phone, c.address, t.city_name, d.district_name
            FROM Companies c
            JOIN Cities t ON c.cityID = t.cityID
            JOIN Districts d ON c.districtID = d.districtID WHERE c.companyID
= $companyID
```

Inserting Example

```
INSERT INTO applications (jobID, userID, application_date, application_status)
            VALUES ('$jobID', '$userID', '$applicationDate',
'$applicationStatus')
```

Update Example

```
UPDATE Companies SET company_name='$up_company_name', website='$up_website',
email='$up_email',password='$safe_password',
    phone='$up_phone', cityID=$up_cityId, districtID=$up_districtId
    WHERE companyID=$companyID
```

Deleting Example

```
DELETE FROM Applications WHERE userID = $userID AND jobID = $jobID
```

Filtering

```
SELECT A.applicationID, U.name, U.surname, J.job_title, A.application_date,
A.application_status
      FROM Applications A
      JOIN Jobs J ON A.jobID = J.jobID
      JOIN Users U ON A.userID = U.userID
      JOIN Companies C ON J.companyID = C.companyID
      WHERE C.companyID = $companyID $filter
```

6. CONCLUSION

The Job Recruitment System provides an efficient platform for employers and job seekers to interact during the recruitment process. The menu structure offers a clear and intuitive navigation experience, while the data structure and tables ensure the proper organization and storage of essential information. With the implementation of PHP and Microsoft SQL Server, the system is equipped to handle the functionalities required for successful job recruitment.

7. REFERENCES

1) Thomas M. Connolly, Caroline E. Begg - Database Systems_ A Practical Approach to Design, Implementation, and Management-Pearson

2) Course Slides

3) https://www.w3schools.com/php/

4) https://codepen.io/