JOB APPLICATION PORTAL

Guided by Prof. Nirosha Dinayadura

Group-2 Members:

- **DivyaLakshmi Thota**
- Vishal Rachuri
- Sai Varshith Gandu
- Sushil Banka
- Teegala Tejasvi
- Lalitha Nali

Table of Contents

Abstract:

Creative techniques that speed up the recruitment procedure are required in the modern job market. Our Job Application Portal intends to completely change the recruiting environment through integrating robust backend MYSQL server capabilities with a Python Tkinter-based graphical user interface. This portal seeks to offer an easy-to-use user interface, establish essential collaborations, and increase successful job matching and management. It provides essential knowledge and interactions to the recruiting process by adapting to the demands of both job seekers and employers. The current study covers the development and deployment of this essential job recruitment tool.

Introduction:

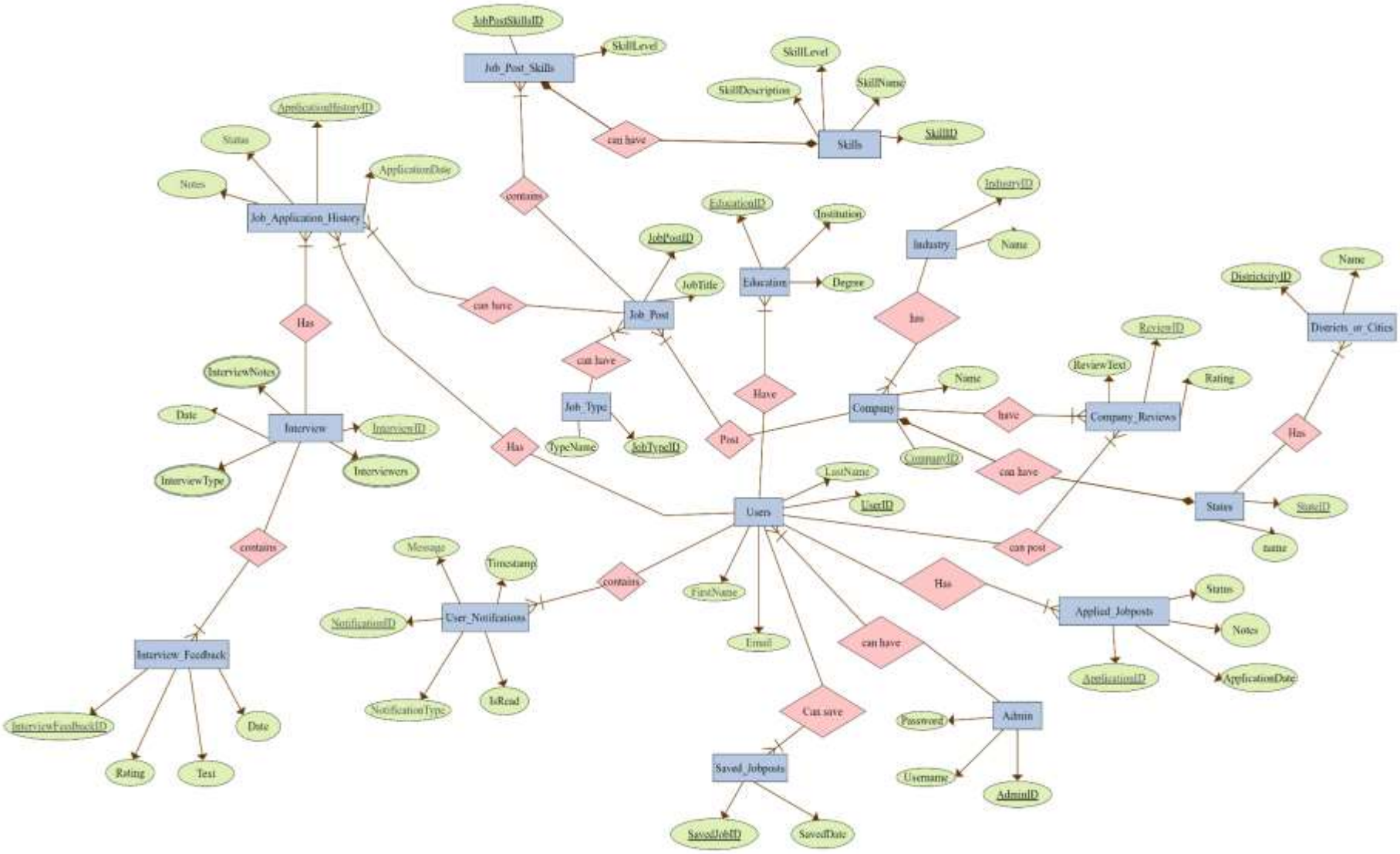
In today's job market, matching job searchers with potential employers requires an effective and efficient system. Understanding the challenges and complexities of the recruiting industry, our team decided to develop a Job Application Portal that satisfied these requirements. The primary objective of the portal is to accelerate the recruiting process through providing a user-friendly interface for both job seekers and businesses. Our project intends to offer a comprehensive solution that facilitates job matching, enables efficient communication, and offers critical insights into the recruiting process through using MYSQL server for backend data management and Python Tkinter for frontend implementation. This work describes the phases involved in planning, building, and implementing the Job Application Portal, as well as its implementation and working.

Implementation:

1. Planning and Design:
   The project initiation phase involved meticulous planning and design considerations. This encompassed identifying essential features, database structure design, and front-end layout. Clear goals were established to create a user-centric interface while ensuring efficient data handling through the MYSQL server.
2. Backend Development:
   The backend was developed using the MYSQL server, involving the creation of necessary tables in addition to the establishment of relationships between them. Throughout this phase, the database architecture was built up to securely store job-related information and user data.
3. Frontend Implementation:
   The front-end interface has been developed using Python's Tkinter framework, providing users with a simple and interactive experience. Individual scripts were linked to particular table activities, with the main menu serving as a centralized portal for accessing multiple features.
4. Integration and Testing:
   After the finalization of both the backend and frontend, thorough testing was carried out to ensure perfect integration and functionality. To provide an effective and error-free application, testing included verifying data exchanges, user inputs, and system responses to various instances.
5. End user experience:

   The Job Application Portal was introduced with successful testing, allowing users both job seekers and employers to engage with the system. Job search, submission of applications, employer profile management, and information into current openings were all available over the platform.

Updated ER diagram:



Cardinalities:

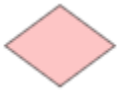
- Users have a one-to-many relationship with Applied_JobPosts, Education, User_Notifications, Company_Reviews.
- Company has a one-to-many relationship with Job_Post and a many-to-one relationship with Industry.
- Job_Post has a one-to-many relationship with Job_Application_History, Job_Post_Skills, and a many-to-one relationship with Company and Job_Type.
- Districts_or_Cities have a many-to-one relationship with States.
- Job_Application_History has a many-to-one relationship with Users and Job_Post.
- Job_Post_Skills has a many-to-one relationship with Job_Post and many-to-many with Skills.
- Interviews have a one-to-many relationship with Job_Application_History.
- Company_Reviews have a many-to-one relationship with Company and Users.
- Saved_JobPosts have a many-to-one relationship with Users.
- User_Notifications have a many-to-one relationship with Users.
- Interview_Feedback has a many-to-one relationship with Interviews.

We have made changes to ER diagram based on comments and all the entities mentioned in ER are being used.

Many-to-Many relationships are represented by a diamond shape connecting entities.

Many-to-One relationships are also represented by lines connecting entities, with a crow's foot on the "one" side and a straight line on the "many" side.

One-to-Many relationships are represented by lines connecting entities, with a crow's foot (a small line with three lines) on the "many" sides and a straight line on the "one" side.

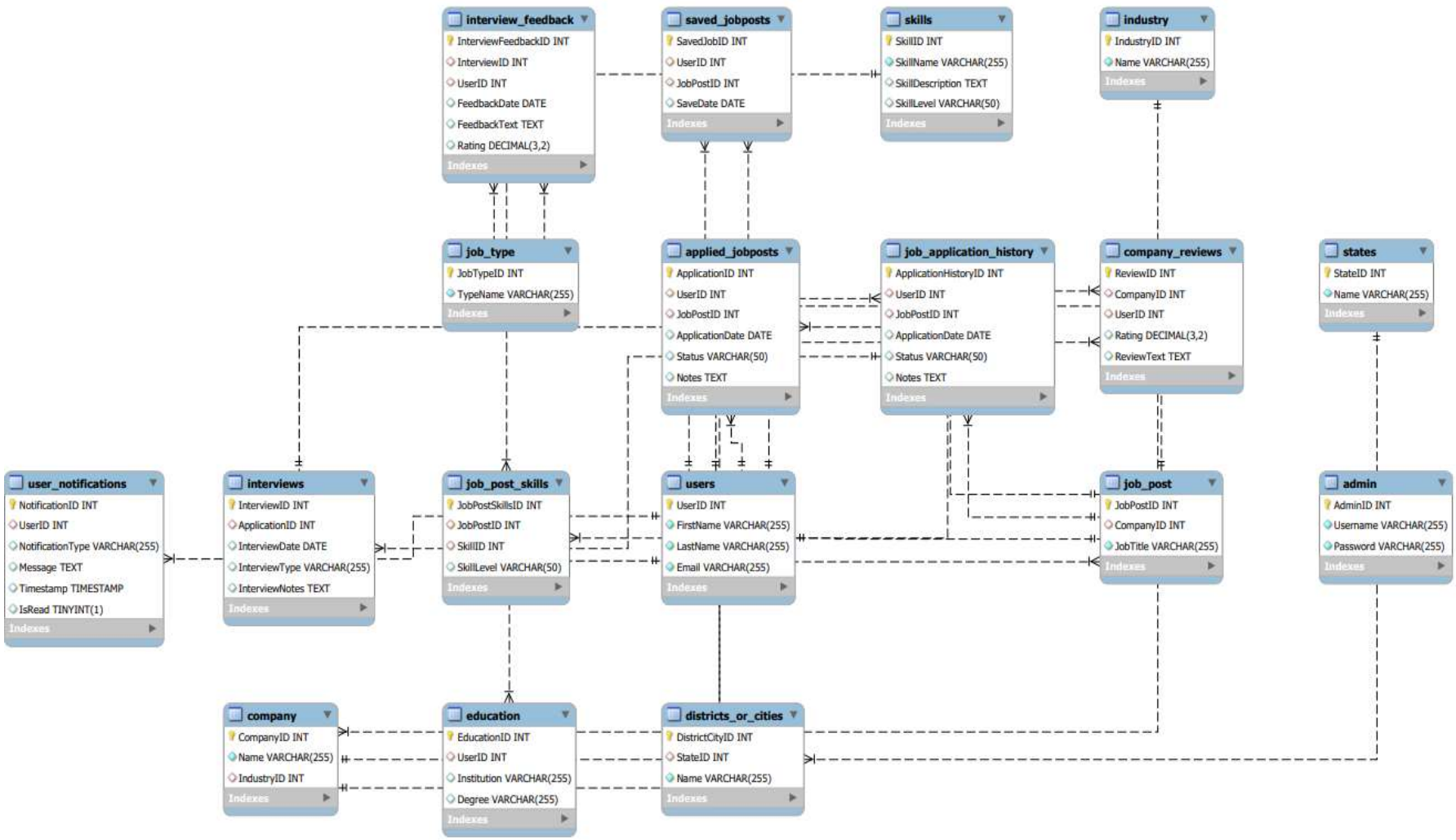


Relationship

Entity

Attributes

Relational Schema:



SQL Statements:

Creating tables:

```
Create database jobApplicationPortal;
use jobApplicationPortal;
-- Admin Table
CREATE TABLE Admin (
    AdminID INT PRIMARY KEY,
    Username VARCHAR(255) NOT NULL,
    Password VARCHAR(255) NOT NULL
);
-- Users Table
CREATE TABLE Users (
    UserID INT PRIMARY KEY,
    FirstName VARCHAR(255) NOT NULL,
    LastName VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL
);
-- Industry Table
CREATE TABLE Industry (
    IndustryID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL
);
-- Company Table
CREATE TABLE Company (
    CompanyID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    IndustryID INT,
    FOREIGN KEY (IndustryID) REFERENCES Industry(IndustryID)
);
-- Job_Post Table
CREATE TABLE Job_Post (
    JobPostID INT PRIMARY KEY,
    CompanyID INT,
    JobTitle VARCHAR(255) NOT NULL,
    FOREIGN KEY (CompanyID) REFERENCES Company(CompanyID)
);
-- Applied JobPosts Table
CREATE TABLE Applied_JobPosts (
    ApplicationID INT PRIMARY KEY,
    UserID INT,
    JobPostID INT,
    ApplicationDate DATE,
    Status VARCHAR(50),
    Notes TEXT,
    FOREIGN KEY (UserID) REFERENCES Users(UserID),
    FOREIGN KEY (JobPostID) REFERENCES Job_Post(JobPostID)
);
-- Company Reviews Table
```

```
CREATE TABLE Company_Reviews (
    ReviewID INT PRIMARY KEY,
    CompanyID INT,
    UserID INT,
    Rating DECIMAL(3, 2),
    ReviewText TEXT,
    FOREIGN KEY (CompanyID) REFERENCES Company(CompanyID),
    FOREIGN KEY (UserID) REFERENCES Users(UserID)
);

-- States Table
CREATE TABLE States (
    StateID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL
);
-- Districts or Cities Table
CREATE TABLE Districts_or_Cities (
    DistrictCityID INT PRIMARY KEY,
    StateID INT,
    Name VARCHAR(255) NOT NULL,
    FOREIGN KEY (StateID) REFERENCES States(StateID)
);
-- Education Table
CREATE TABLE Education (
    EducationID INT PRIMARY KEY,
    UserID INT,
    Institution VARCHAR(255),
    Degree VARCHAR(255),
    FOREIGN KEY (UserID) REFERENCES Users(UserID)
);
-- Job_Type Table
CREATE TABLE Job_Type (
    JobTypeID INT PRIMARY KEY,
    TypeName VARCHAR(255) NOT NULL
);

-- Saved JobPosts Table
CREATE TABLE Saved_JobPosts (
    SavedJobID INT PRIMARY KEY,
    UserID INT,
    JobPostID INT,
    SaveDate DATE,
    FOREIGN KEY (UserID) REFERENCES Users(UserID),
    FOREIGN KEY (JobPostID) REFERENCES Job_Post(JobPostID)
);
-- Job Application History Table
CREATE TABLE Job_Application_History (
    ApplicationHistoryID INT PRIMARY KEY,
    UserID INT,
```

```
    JobPostID INT,
    ApplicationDate DATE,
    Status VARCHAR(50),
    Notes TEXT,
    FOREIGN KEY (UserID) REFERENCES Users(UserID),
    FOREIGN KEY (JobPostID) REFERENCES Job_Post(JobPostID)
);
-- Job Post Skills Table
CREATE TABLE Job_Post_Skills (
    JobPostSkillsID INT PRIMARY KEY,
    JobPostID INT,
    SkillID INT,
    SkillLevel VARCHAR(50),
    FOREIGN KEY (JobPostID) REFERENCES Job_Post(JobPostID),
    FOREIGN KEY (SkillID) REFERENCES Skills(SkillID)
);
--Interviews Table
CREATE TABLE Interviews (
    InterviewID INT PRIMARY KEY,
    ApplicationID INT,
    InterviewDate DATE,
    InterviewType VARCHAR(255),
    InterviewNotes TEXT,
    FOREIGN KEY (ApplicationID) REFERENCES Job_Application_History(ApplicationHistoryID)
);
-- Interview Feedback Table
CREATE TABLE Interview_Feedback (
    InterviewFeedbackID INT PRIMARY KEY,
    InterviewID INT,
    UserID INT,
    FeedbackDate DATE,
    FeedbackText TEXT,
    Rating DECIMAL(3, 2),
    FOREIGN KEY (InterviewID) REFERENCES Interviews(InterviewID),
    FOREIGN KEY (UserID) REFERENCES Users(UserID)
);
-- User Notifications Table
CREATE TABLE User_Notifications (
    NotificationID INT PRIMARY KEY,
    UserID INT,
    NotificationType VARCHAR(255),
    Message TEXT,
    Timestamp TIMESTAMP,
    IsRead BOOLEAN,
    FOREIGN KEY (UserID) REFERENCES Users(UserID)
);
-- Skills Table
CREATE TABLE Skills (
    SkillID INT PRIMARY KEY,
```

```
    SkillName VARCHAR(255) NOT NULL,
    SkillDescription TEXT,
    SkillLevel VARCHAR(50)
);
```

Inserting into tables:

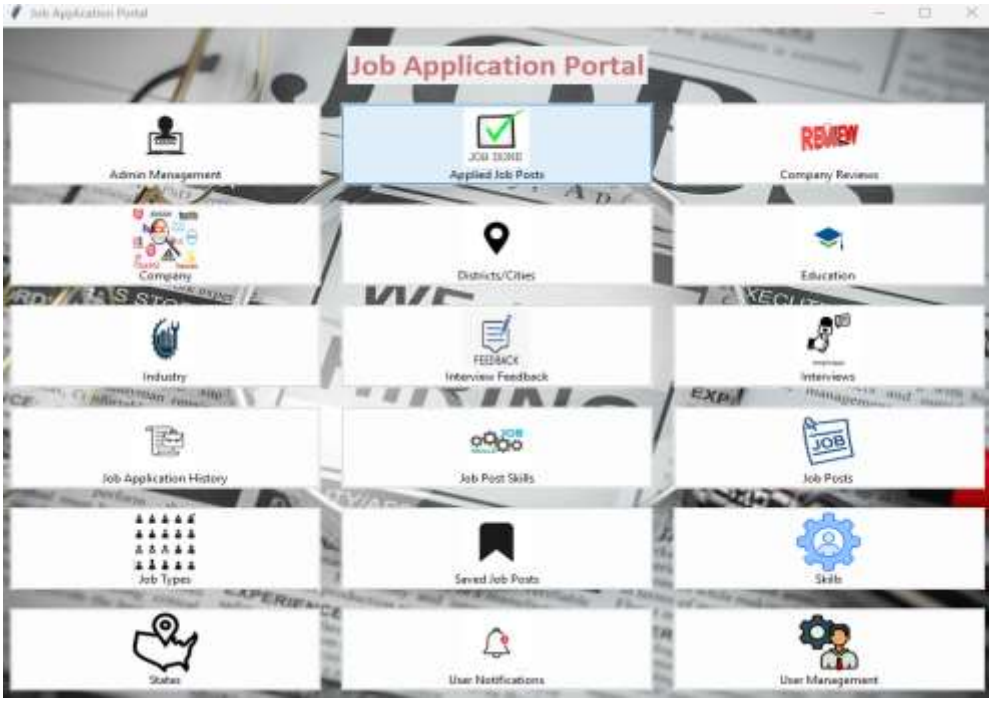
```
INSERT INTO Admin (AdminID, Username, Password) VALUES
    (1, 'admin1', 'password1'),
    (2, 'admin2', 'password2'),
    (3, 'admin3', 'password3'),
    (4, 'admin4', 'password4'),
    (5, 'admin5', 'password5'),
    (6, 'admin6', 'password6');

INSERT INTO Users (UserID, FirstName, LastName, Email) VALUES
    (1, 'John', 'Doe', 'john.doe@example.com'),
    (2, 'Jane', 'Smith', 'jane.smith@example.com'),
    (3, 'Michael', 'Johnson', 'michael.johnson@example.com'),
    (4, 'Emily', 'Brown', 'emily.brown@example.com'),
    (5, 'David', 'Lee', 'david.lee@example.com'),
    (6, 'Sarah', 'Wilson', 'sarah.wilson@example.com');

INSERT INTO States (StateID, Name) VALUES
    (1, 'California'),
    (2, 'New York'),
    (3, 'Texas');

INSERT INTO Industry (IndustryID, Name) VALUES
    (1, 'Technology'),
    (2, 'Finance'),
    (3, 'Healthcare');

INSERT INTO Company (CompanyID, Name, IndustryID) VALUES
    (1, 'Tech Innovators Inc.', 1),
    (2, 'Finance Experts Ltd.', 2),
    (3, 'HealthCare Solutions LLC', 3),
    (4, 'Data Insights Co.', 1),
    (5, 'Marketing Producers Corp.', 2),
    (6, 'HR Solutions Group', 3);

INSERT INTO Job_Post (JobPostID, CompanyID, JobTitle) VALUES
    (1, 1, 'Software Engineer'),
    (2, 2, 'Financial Analyst'),
    (3, 3, 'Registered Nurse'),
    (4, 4, 'Data Scientist'),
    (5, 5, 'Marketing Manager'),
    (6, 6, 'HR Coordinator');

INSERT INTO Applied_JobPosts (ApplicationID, UserID, JobPostID, ApplicationDate, Status, Notes) VALUES
```

```
    (1, 1, 1, '2023-01-15', 'Pending', 'Application under review.'),
    (2, 2, 2, '2023-02-20', 'Accepted', 'Scheduled for an interview.'),
    (3, 3, 3, '2023-03-10', 'Rejected', 'Position filled.'),
    (4, 4, 4, '2023-04-05', 'Pending', 'Awaiting HR response.'),
    (5, 5, 5, '2023-05-12', 'Accepted', 'Interview completed.'),
    (6, 6, 6, '2023-06-25', 'Pending', 'Application under review.');

INSERT INTO Company_Reviews (ReviewID, CompanyID, UserID, Rating, ReviewText) VALUES
    (1, 1, 1, 4.5, 'Great company to work for.'),
    (2, 2, 2, 3.8, 'Decent workplace with room for improvement.'),
    (3, 3, 3, 4.2, 'Excellent culture and management.'),
    (4, 4, 4, 3.5, 'Average experience, but good benefits.'),
    (5, 5, 5, 4.7, 'Top-notch company with competitive pay.'),
    (6, 6, 6, 3.0, 'Needs improvement in work-life balance.');

INSERT INTO Districts_or_Cities (DistrictCityID, StateID, Name) VALUES
    (1, 1, 'San Francisco'),
    (2, 2, 'New York City'),
    (3, 3, 'Houston'),
    (4, 1, 'Los Angeles'),
    (5, 2, 'Chicago'),
    (6, 3, 'Dallas');

INSERT INTO Education (EducationID, UserID, Institution, Degree) VALUES
    (1, 1, 'Stanford University', 'Bachelor of Science'),
    (2, 2, 'Harvard Business School', 'Master of Business Administration'),
    (3, 3, 'University of Texas', 'Bachelor of Arts'),
    (4, 4, 'High School XYZ', 'High School Diploma'),
    (5, 5, 'Massachusetts Institute of Technology', 'Master of Science'),
    (6, 6, 'University of Chicago', 'Bachelor of Business Administration');

INSERT INTO Job_Type (JobTypeID, TypeName) VALUES
    (1, 'Full-Time'),
    (2, 'Part-Time'),
    (3, 'Contract'),
    (4, 'Internship');

INSERT INTO Saved_JobPosts (SavedJobID, UserID, JobPostID, SaveDate) VALUES
    (1, 1, 1, '2023-01-20'),
    (2, 2, 2, '2023-02-25'),
    (3, 3, 3, '2023-03-15'),
    (4, 4, 4, '2023-04-10'),
    (5, 5, 5, '2023-05-17'),
    (6, 6, 6, '2023-06-30');

INSERT INTO Job_Application_History (ApplicationHistoryID, UserID, JobPostID, ApplicationDate, Status, Notes)
VALUES
    (1, 1, 1, '2023-01-15', 'Pending', 'Application under review.'),
    (2, 2, 2, '2023-02-20', 'Accepted', 'Scheduled for an interview.'),
```

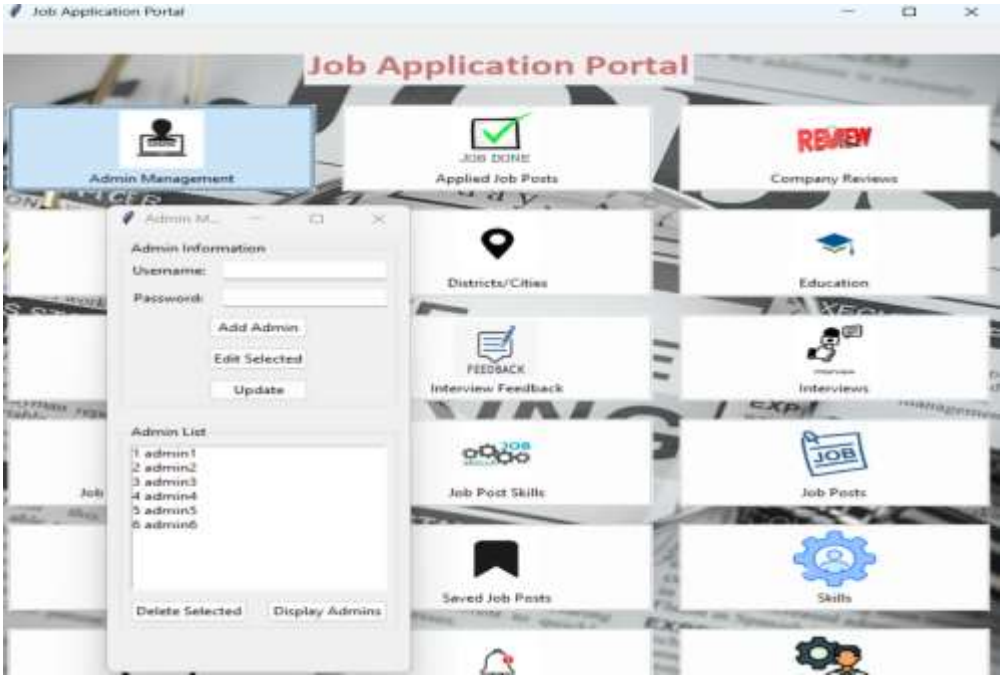
```
    (3, 3, 3, '2023-03-10', 'Rejected', 'Position filled.'),
    (4, 4, 4, '2023-04-05', 'Pending', 'Awaiting HR response.'),
    (5, 5, 5, '2023-05-12', 'Accepted', 'Interview completed.'),
    (6, 6, 6, '2023-06-25', 'Pending', 'Application under review.');

INSERT INTO Skills (SkillID, SkillName, SkillDescription, SkillLevel) VALUES
    (1, 'Java Programming', 'Object-oriented programming language', 'Intermediate'),
    (2, 'Financial Analysis', 'Analyzing financial data', 'Advanced'),
    (3, 'Nursing Skills', 'Medical care skills', 'Expert'),
    (4, 'Data Analysis', 'Analyzing data sets', 'Intermediate'),
    (5, 'Marketing Strategy', 'Strategic marketing planning', 'Advanced'),
    (6, 'HR Management', 'Human resource management', 'Expert');

INSERT INTO Job_Post_Skills (JobPostSkillsID, JobPostID, SkillID, SkillLevel) VALUES
    (1, 1, 1, 'Intermediate'),
    (2, 2, 2, 'Advanced'),
    (3, 3, 3, 'Expert'),
    (4, 4, 4, 'Intermediate'),
    (5, 5, 5, 'Advanced'),
    (6, 6, 6, 'Expert');

INSERT INTO Interviews (InterviewID, ApplicationID, InterviewDate, InterviewType, InterviewNotes) VALUES
    (1, 1, '2023-01-30', 'Phone Interview', 'Candidate performed well.'),
    (2, 2, '2023-02-25', 'In-Person Interview', 'Detailed discussion on qualifications.'),
    (3, 3, '2023-03-15', 'Video Interview', 'Positive interaction with the applicant.'),
    (4, 4, '2023-04-20', 'Technical Interview', 'Average technical skills.'),
    (5, 5, '2023-05-27', 'On-Site Interview', 'Strong presentation skills.'),
    (6, 6, '2023-06-10', 'Phone Interview', 'Needs improvement in communication.');


INSERT INTO Interview_Feedback (InterviewFeedbackID, InterviewID, UserID, FeedbackDate, FeedbackText,
Rating) VALUES
    (1, 1, 1, '2023-01-25', 'Impressed with the candidate.', 4.8),
    (2, 2, 2, '2023-02-28', 'Good communication skills.', 4.0),
    (3, 3, 3, '2023-03-20', 'Positive interaction with the applicant.', 4.5),
    (4, 4, 4, '2023-04-15', 'Average performance in the interview.', 3.2),
    (5, 5, 5, '2023-05-22', 'Strong technical skills.', 4.7),
    (6, 6, 6, '2023-06-05', 'Needs improvement in problem-solving.', 3.3);

INSERT INTO User_Notifications (NotificationID, UserID, NotificationType, Message, Timestamp, IsRead)
VALUES
    (1, 1, 'Job Alert', 'New job opportunity available.', '2023-01-10 08:30:00', false),
    (2, 2, 'Interview Invitation', 'You have been invited for an interview.', '2023-02-15 14:20:00', false),
    (3, 3, 'Application Update', 'Your job application status has changed.', '2023-03-05 10:45:00', false),
    (4, 4, 'Job Offer', 'Congratulations! You have received a job offer.', '2023-04-08 16:10:00', false),
    (5, 5, 'Job Alert', 'New job opportunity available.', '2023-05-18 09:55:00', false),
    (6, 6, 'Interview Invitation', 'You have been invited for an interview.', '2023-06-28 11:30:00', false);
```

UI components:

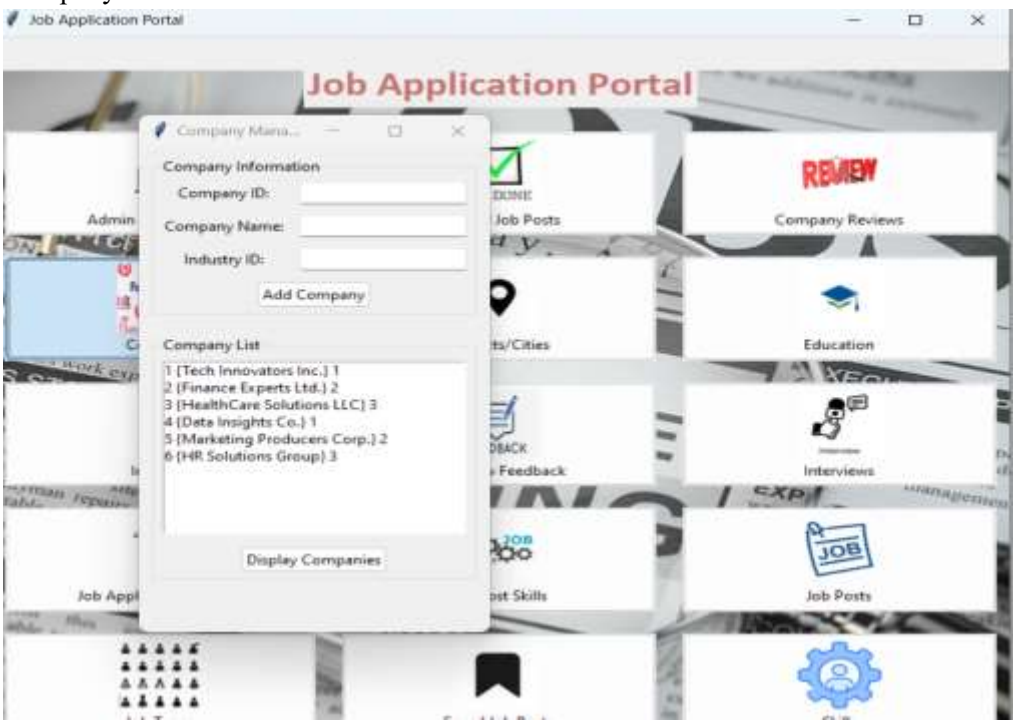
Main Menu:



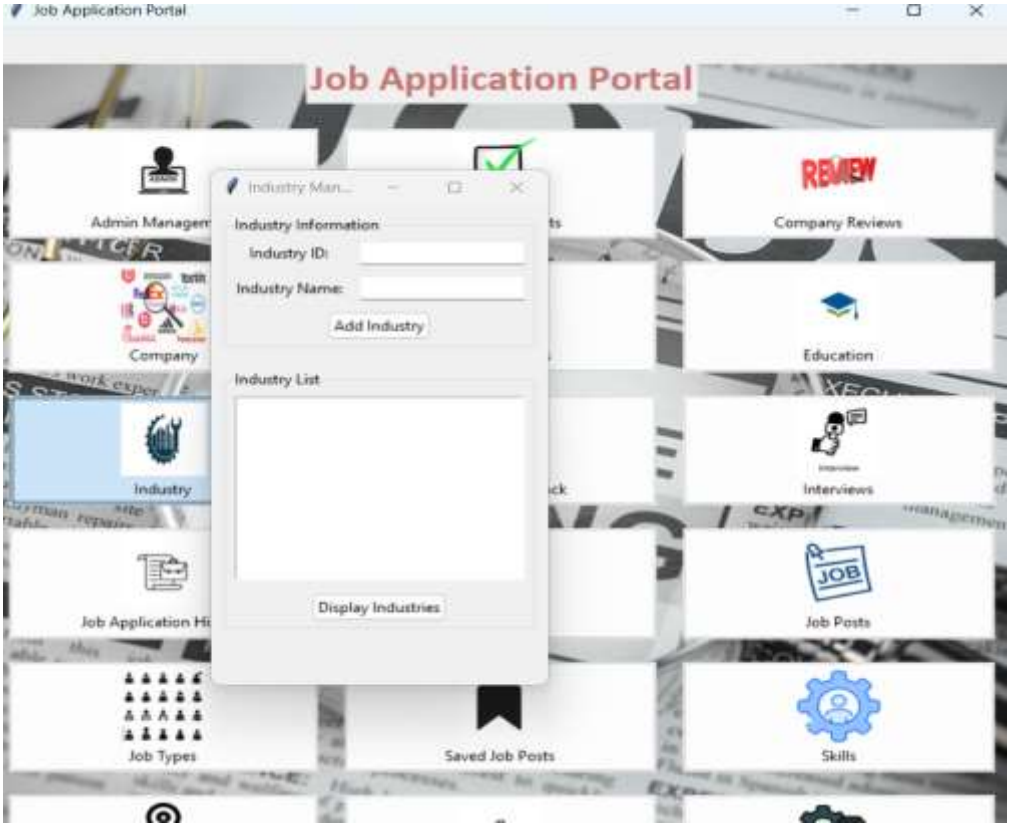
Admin_Management:



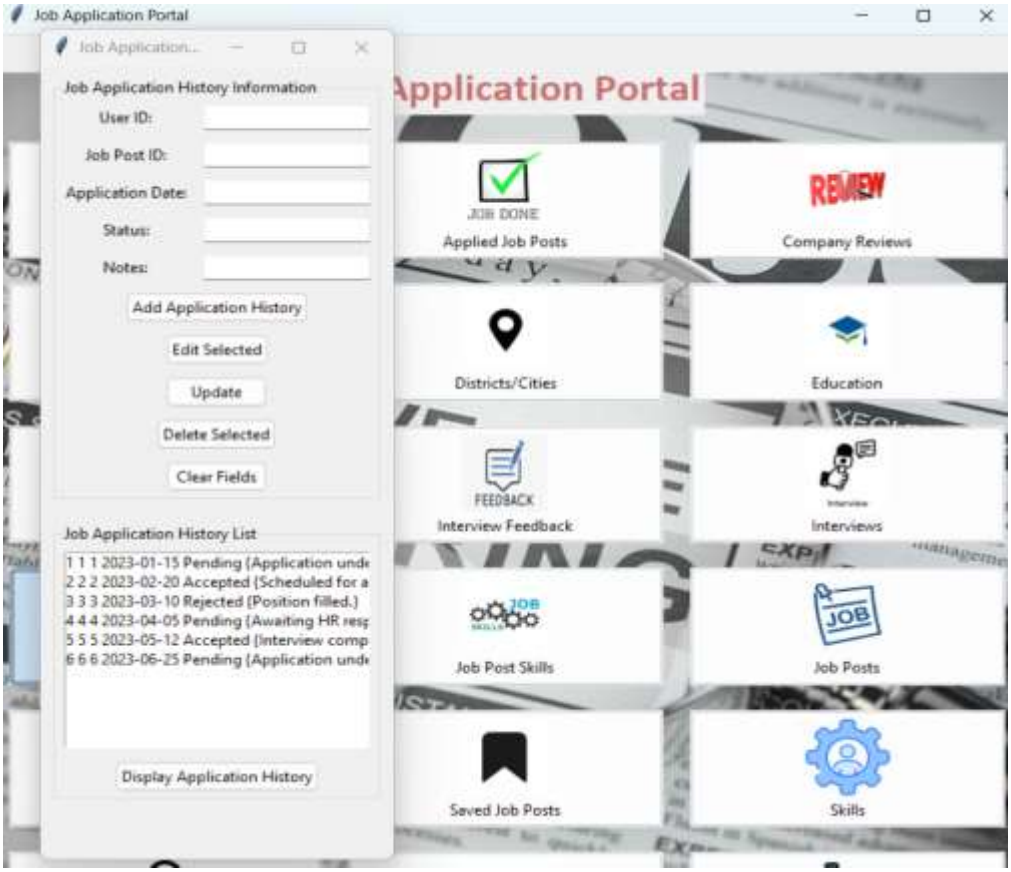
Company:



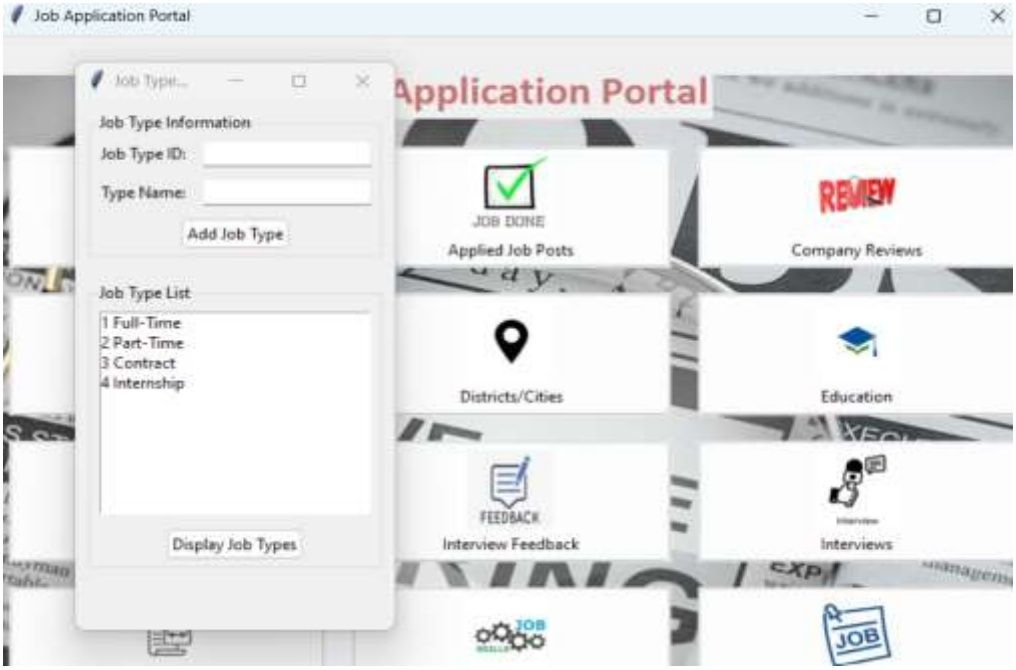
Industry:



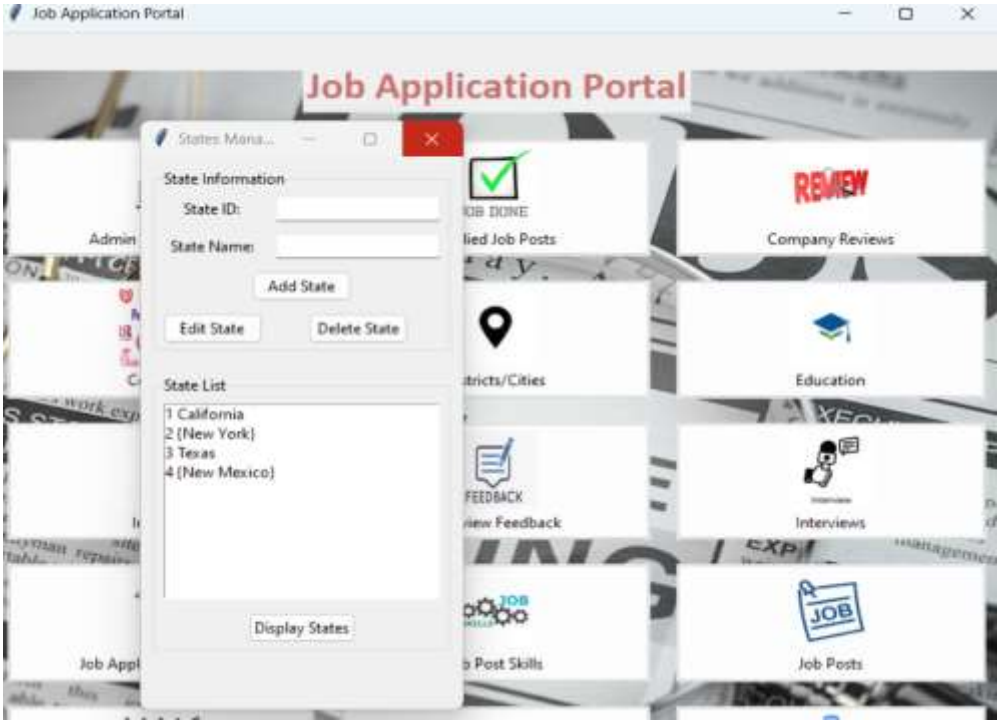
Checking application history:



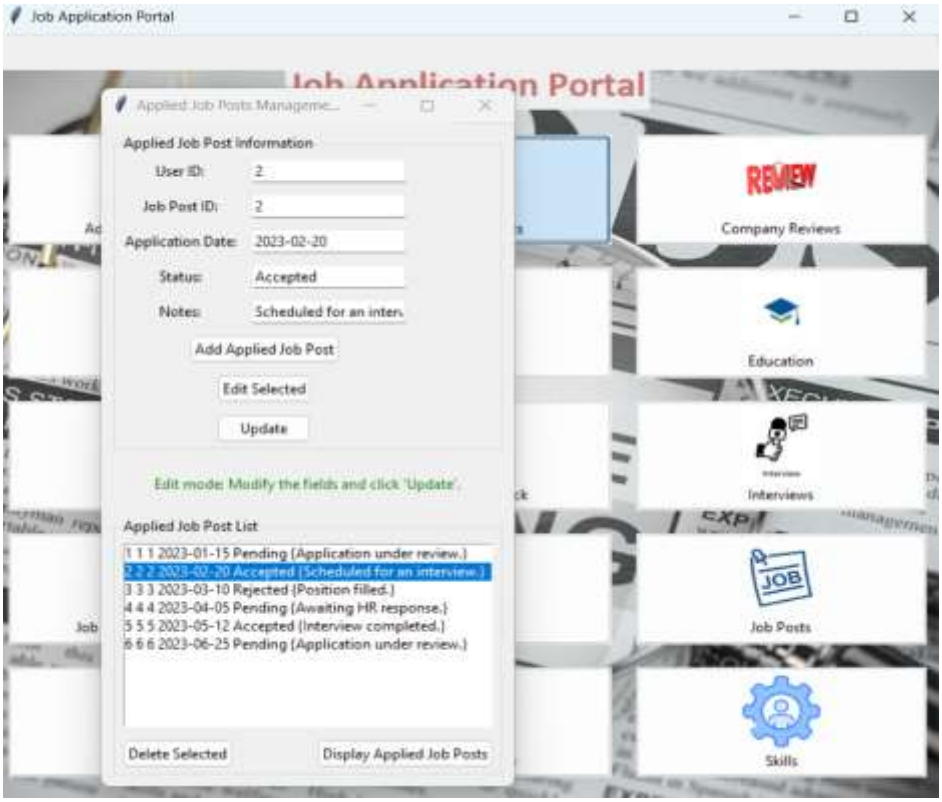
Job types:



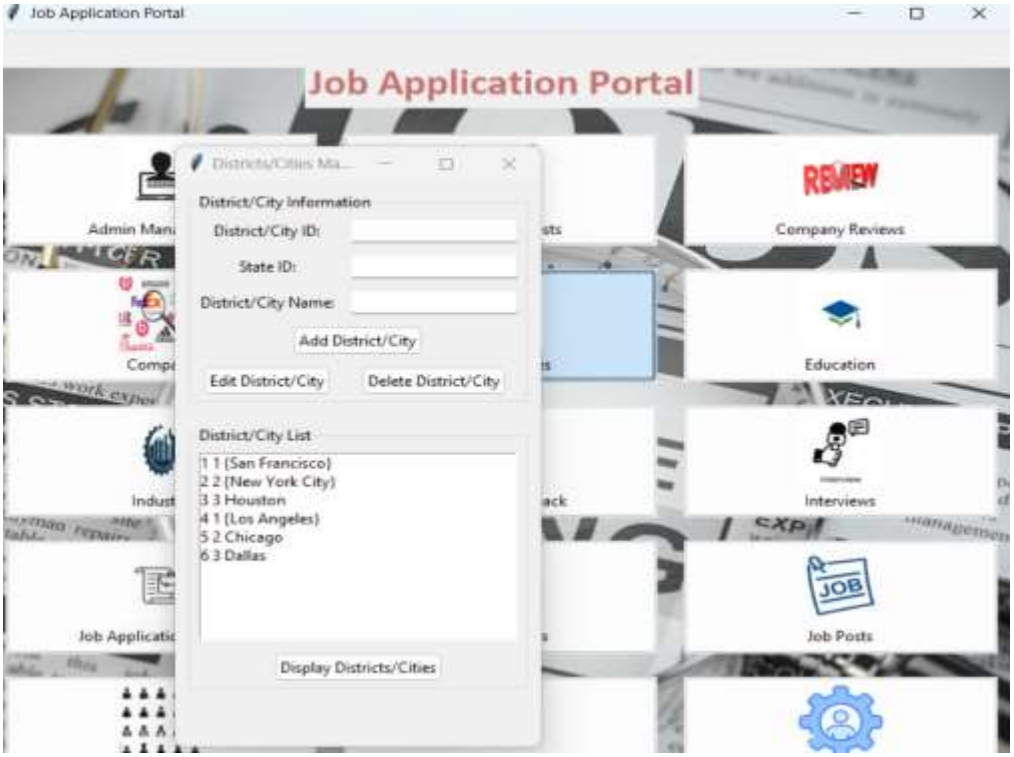
States:



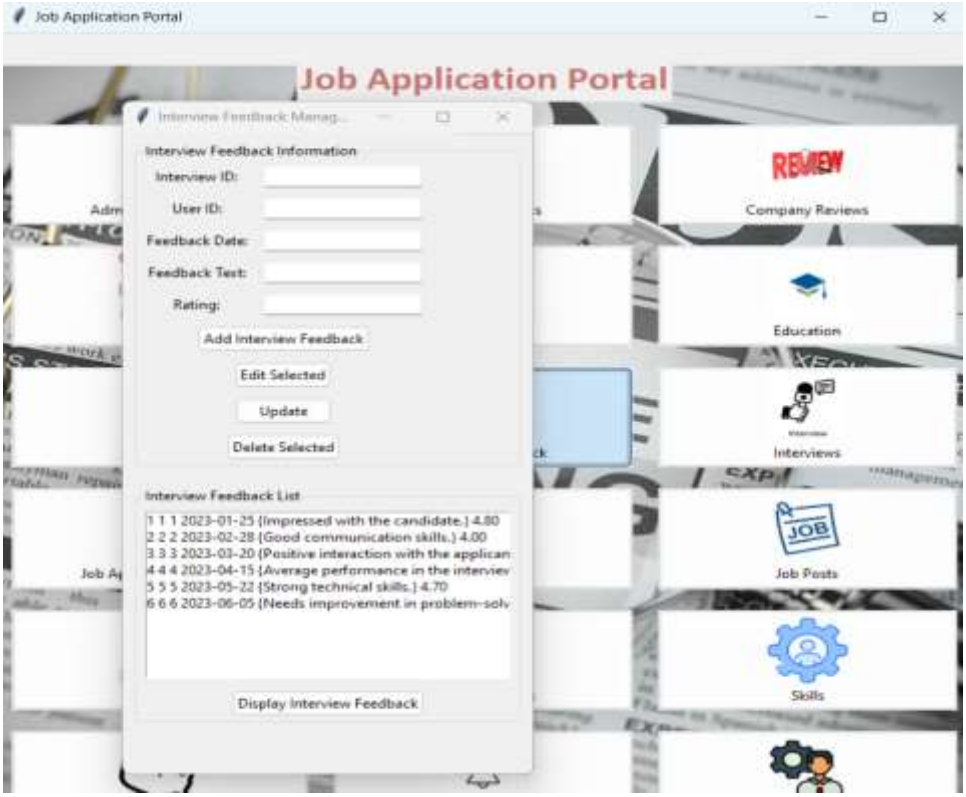
Applied Job Posts:



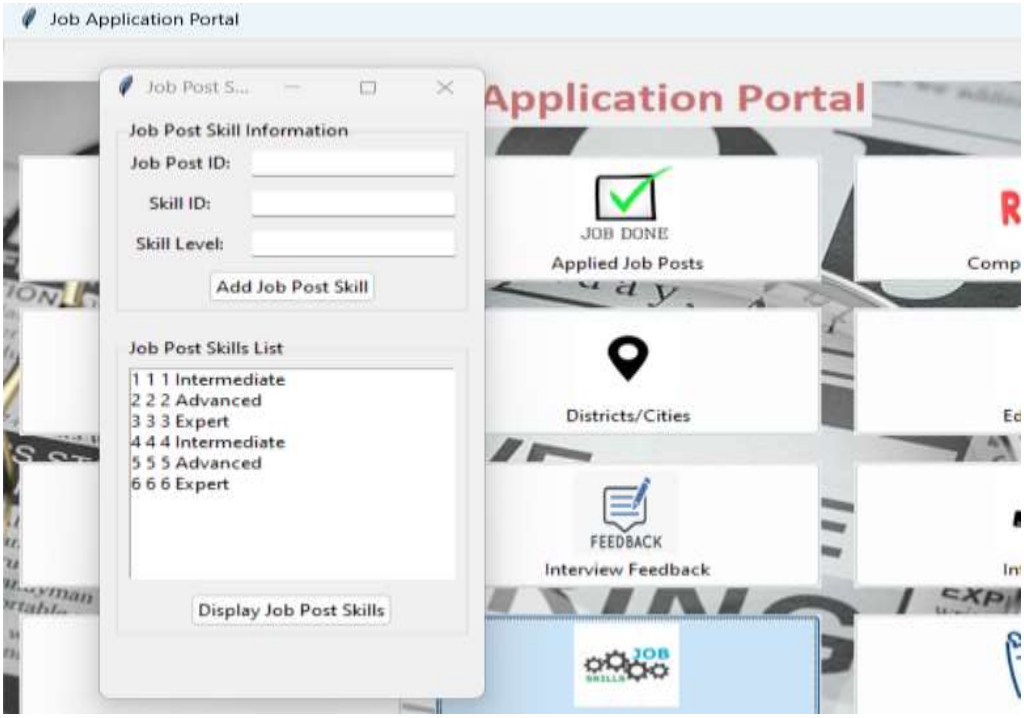
Districts/Cities:



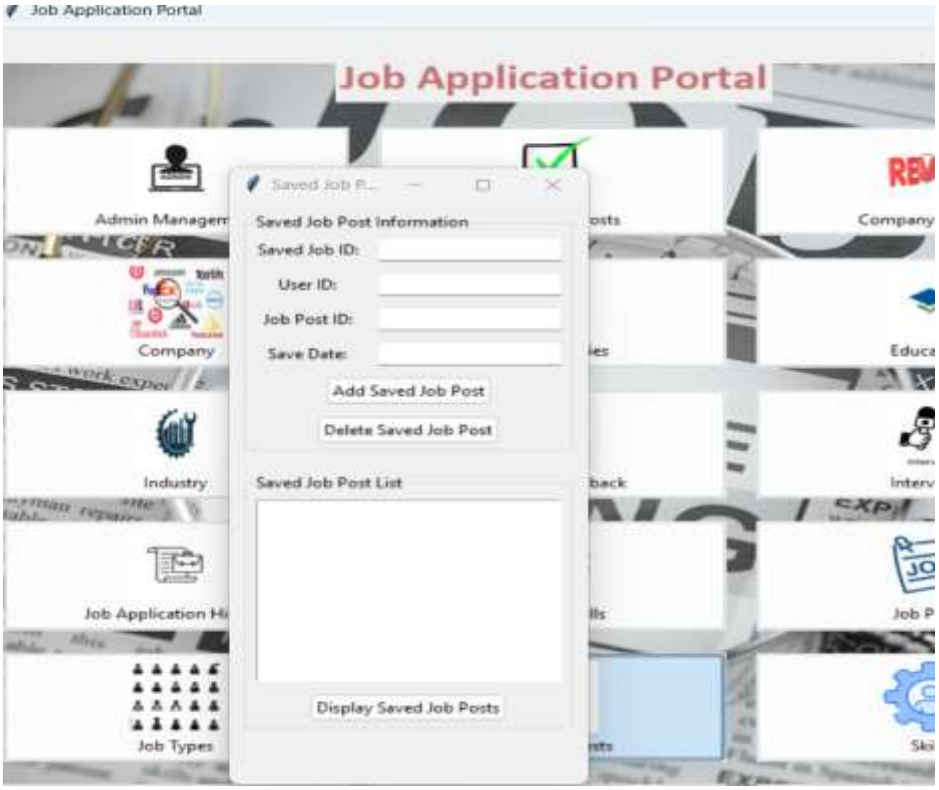
Interview Feedback:



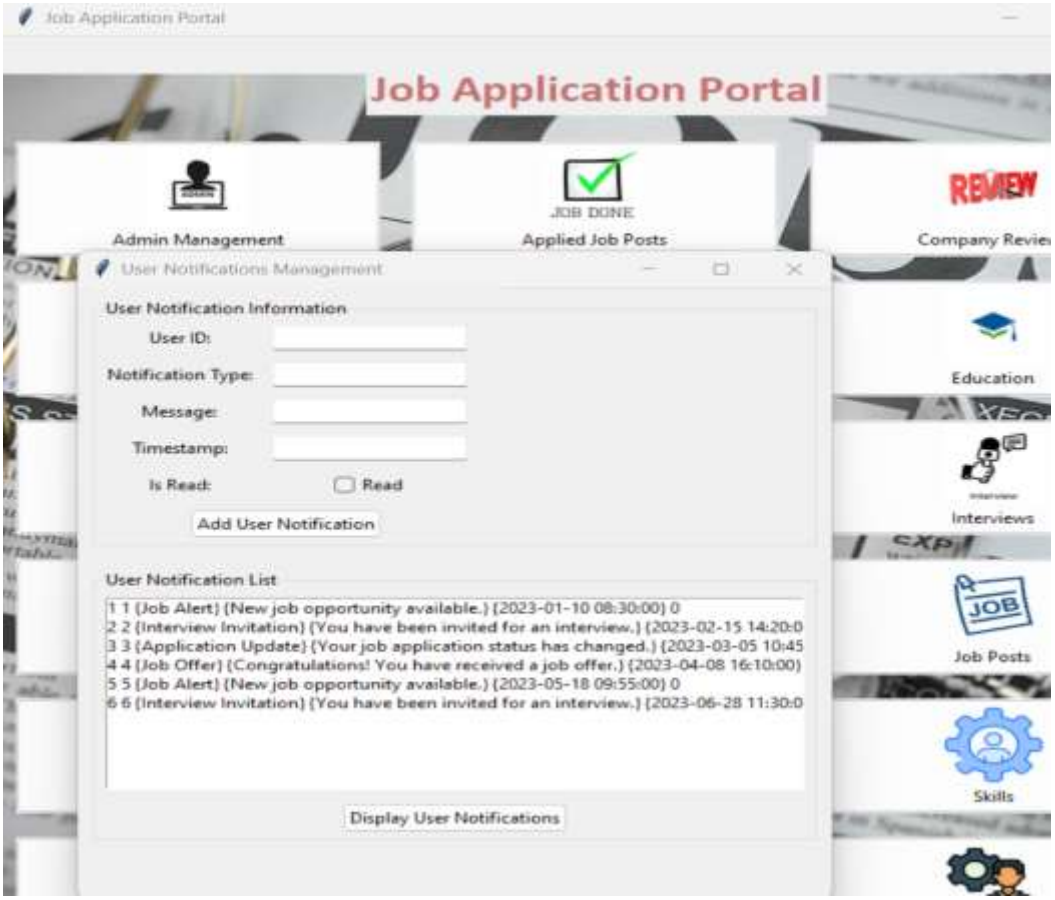
Job Post Skills:



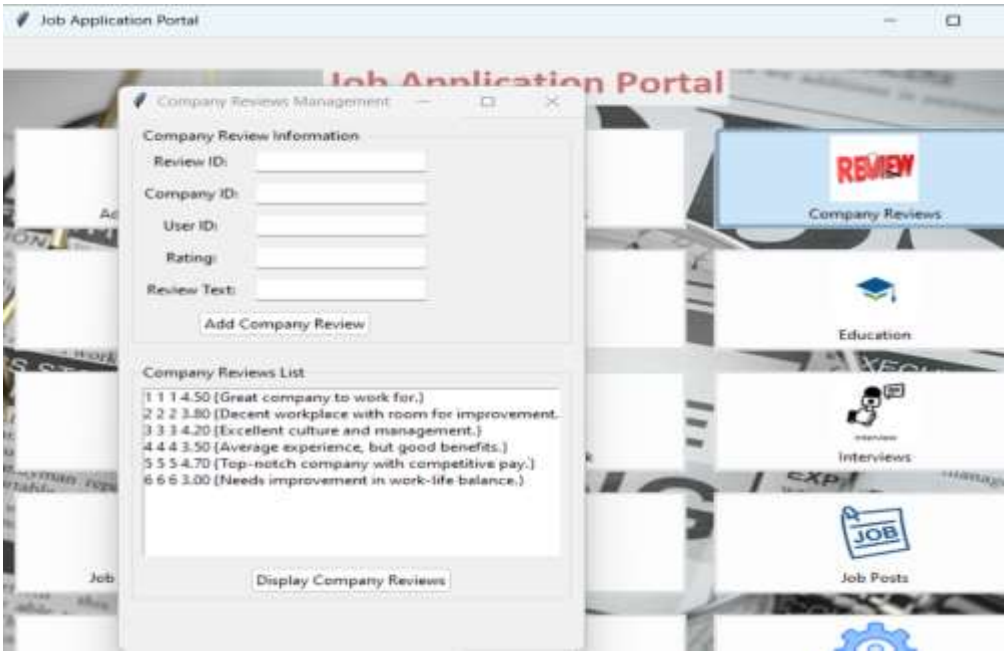
Saved Job Posts:



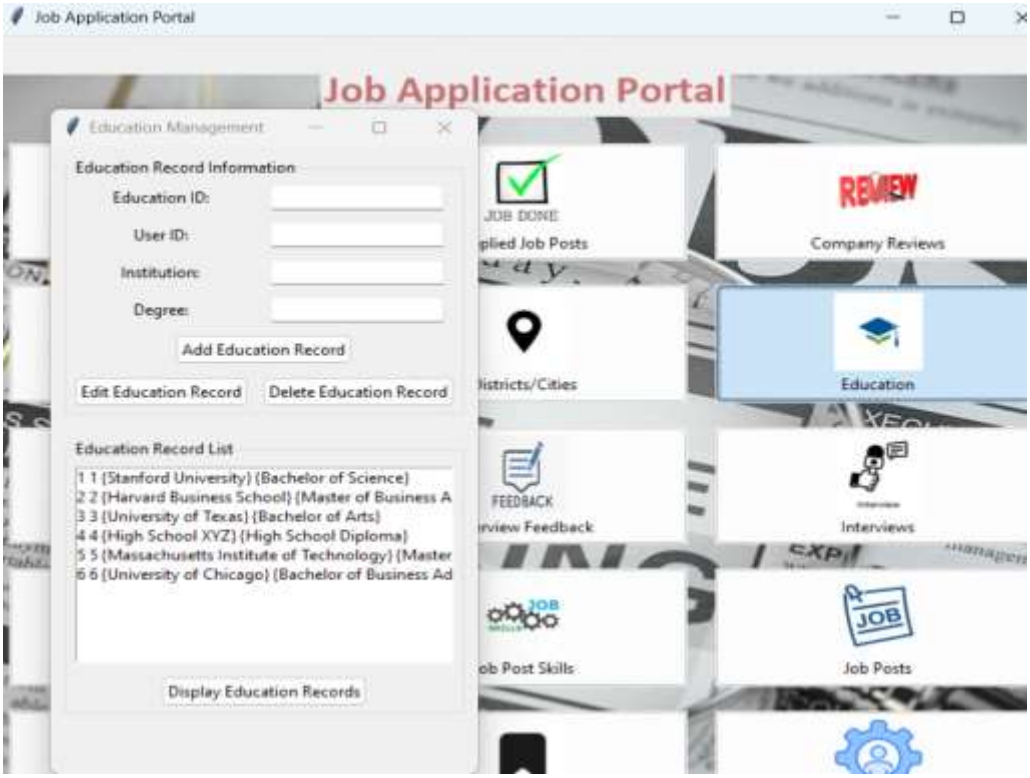
User Notifications:



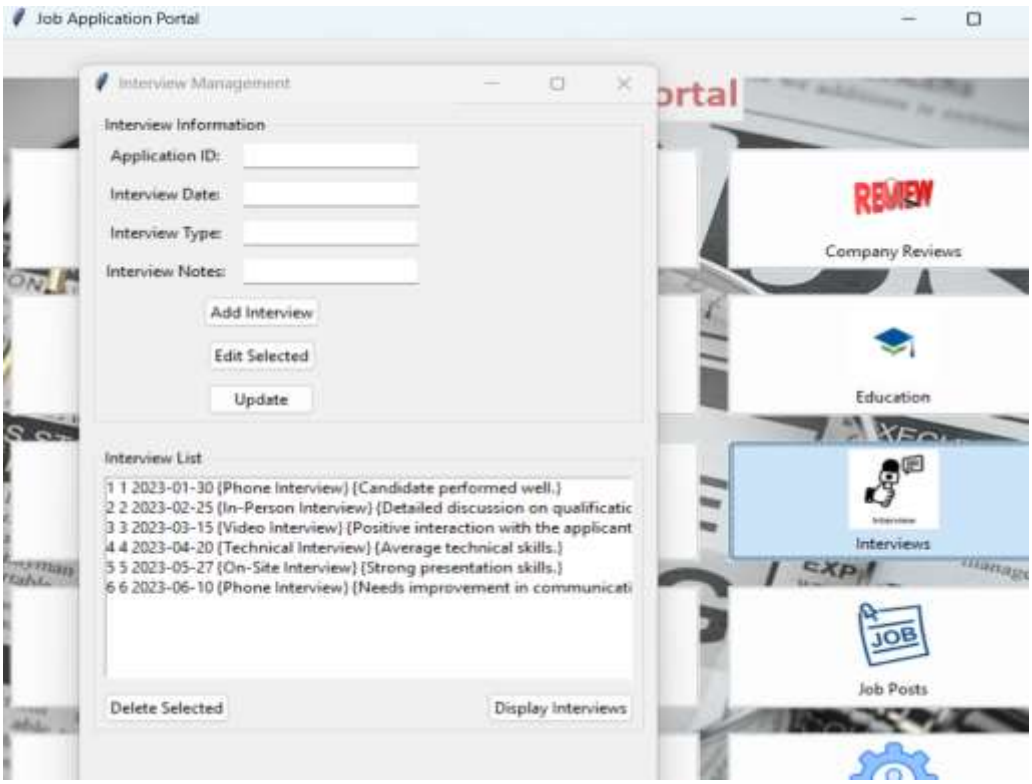
Company Reviews:



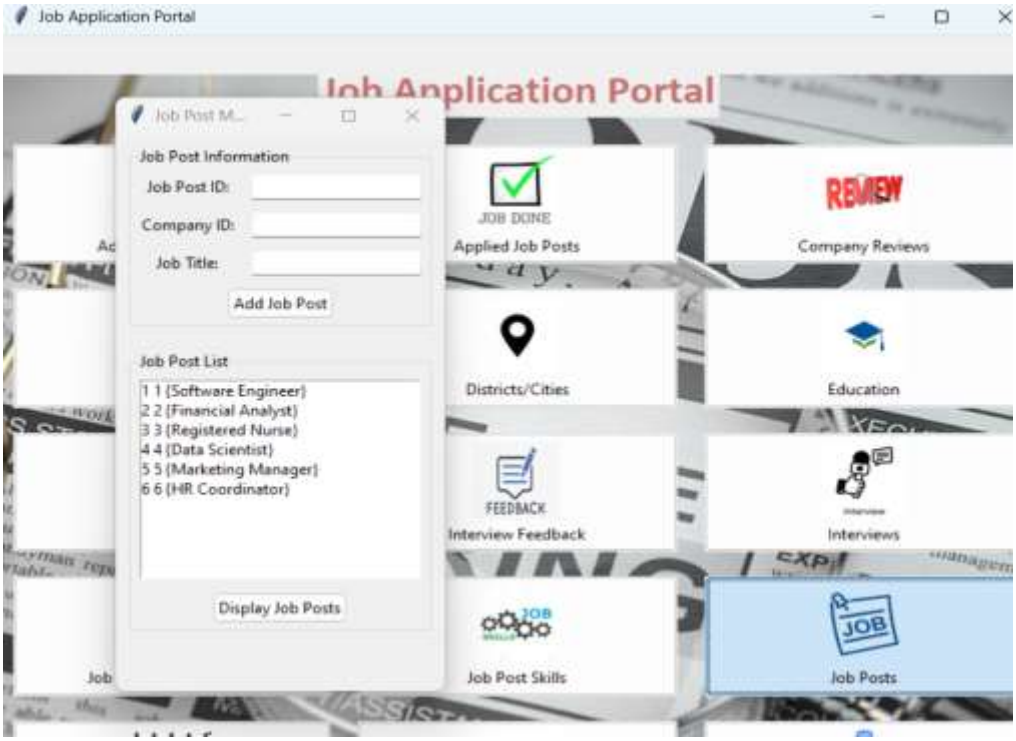
Education:



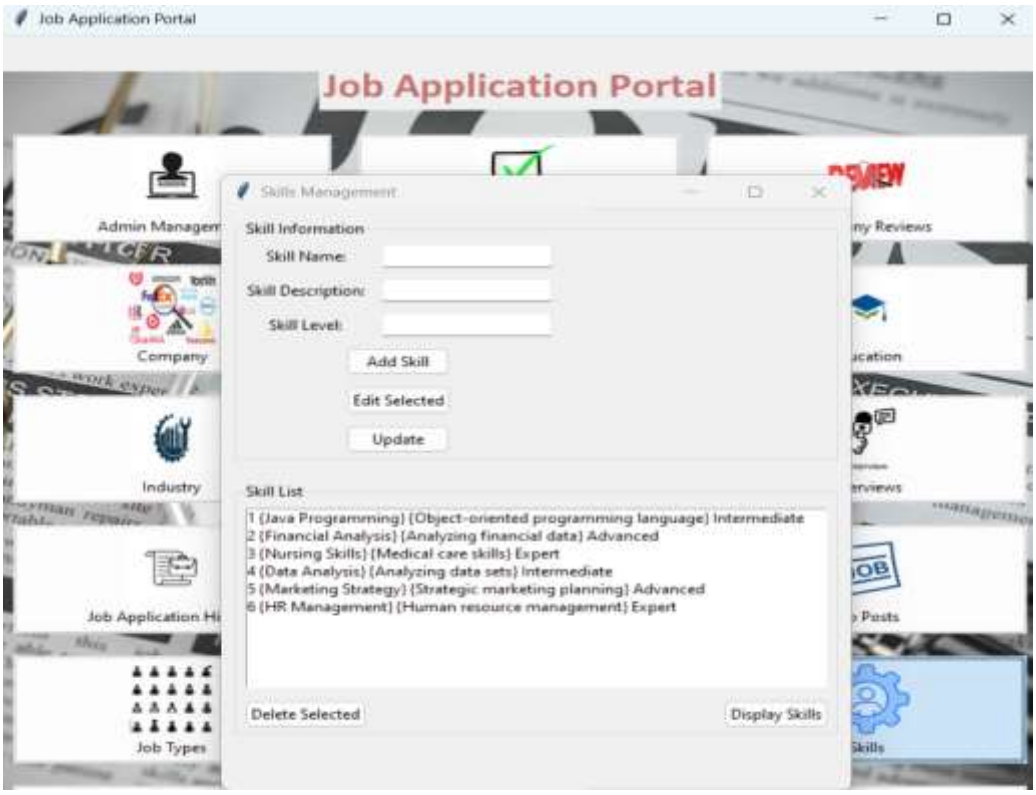
Interviews:



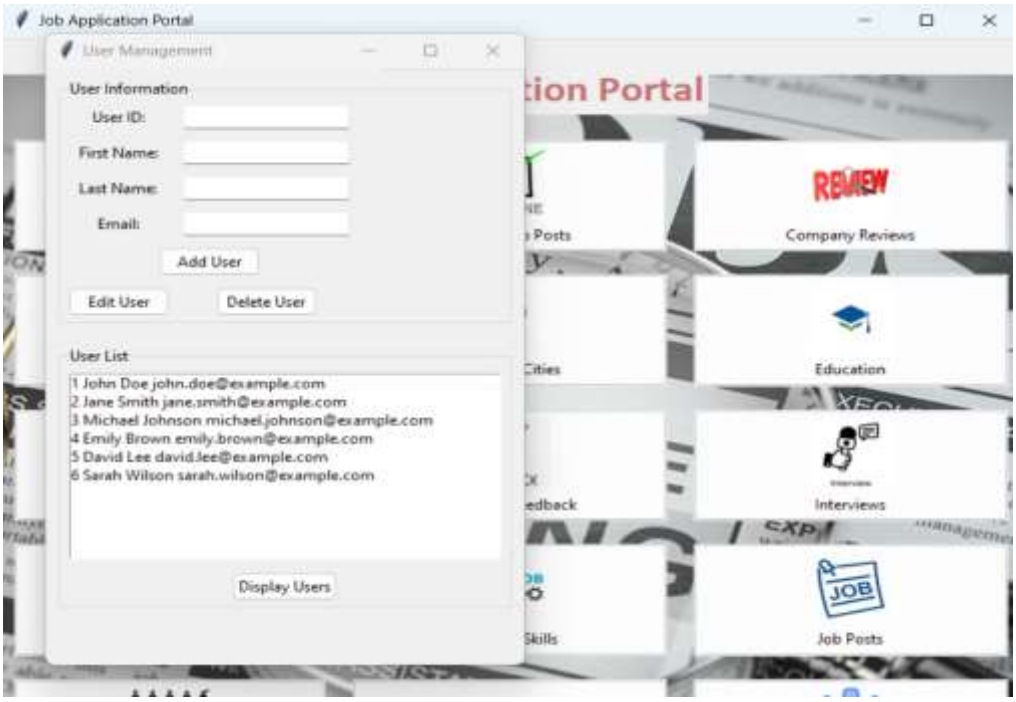
Job Posts:



Skills:



User Management:



Requirements:

Windows 7 or later Version.
Graphical tool for creating Entity-Relationship Diagram (We used YED Live).
Mysql Installed for database related operations like creating tables and inserting into tables.
IDE or Jupyter notebook supporting python to run the UI.
Zip folder that contains python scripts and icons.

Conclusion:

The completion of this project marks a significant milestone in enhancing the efficiency of the recruitment process. The Job Application Portal's successful implementation amalgamates backend robustness with frontend intuitiveness, contributing to a streamlined and user-friendly platform for job seekers and employers.

References:

[1] https://realpython.com/python-gui-tkinter/

[2] https://www.geeksforgeeks.org/create-first-gui-application-using-python-tkinter/

[3] https://www.indeed.com/career-advice/career-development/recruitment-portals

[4] https://knovator.com/blog/importance-of-creating-job-portals/

[5] Grace, M., Ventura, G., & Bringula, R.P. (2013). Effectiveness of Online Job Recruitment System: Evidence from the University of the East.