Developed by : SRIHARIHARAN D

Applicant Tracking System (ATS) Documentation

Project Overview

Project Name:

Applicant Tracking System (ATS)

Purpose:

The ATS is designed to automatically match candidate resumes against job descriptions (JDs) provided by employers. If a resume matches the job description with a score above a certain threshold (80 in this case), the system prompts for the candidate's name and email and stores the information along with the resume and its calculated match score in a database.

Key Features:

- Input for both the Job Description and Resume.
- Scoring system that calculates the match percentage between the JD and Resume.
- A threshold score (80) to determine whether the candidate's resume qualifies for the job.
- Database integration to store shortlisted resumes.
- Console-based user interface (can be extended to a GUI in the future).

Database Schema

Database Name: ats_db

Table: shortlisted_resumes

Table Structure:

Column Name	Data Type	Constraints
id	INT	AUTO_INCREMENT, PRIMARY KEY
candidate_name	VARCHAR(100)	NOT NULL
resume_text	TEXT	NOT NULL
score	DECIMAL(5, 2)	NOT NULL

Description:

This table stores the shortlisted resumes. When a candidate's resume passes the match threshold (score ≥ 80), their name, resume text, and score are inserted into this table.

System Flow

1. **Input:**
   - The system accepts input for both the Job Description and Resume in the form of text.
   - If the resume score meets or exceeds 80, the system prompts the user for the candidate's name and email.
2. **Processing:**
   - The system calculates the match score between the Job Description and Resume based on keyword matching, skill matching, and relevant experience.
   - A scoring algorithm evaluates how well the resume matches the JD.
3. **Output:**
   - If the resume score is 80 or above, the system:
     - Displays the match score.
     - Prompts the user to enter the candidate’s name and email.
     - Stores the name, resume text, and score in the shortlisted_resumes table in the database.

Example Use Case Scenarios

Use Case 1: Successful Match

- **Input:**
  - JD: "Looking for a Java Developer with experience in Spring Boot, RESTful APIs, and MySQL."
  - Resume: "Experienced Java Developer with Spring Boot, RESTful APIs, and MySQL skills."

- **Expected Result:**
  - Match Score: 85
  - Prompt for candidate name and email.
  - Insert into the shortlisted_resumes table with a score of 85.

Use Case 2: Low Match Score

- **Input:**
  - JD: "Looking for a Python Developer with Flask and PostgreSQL experience."
  - Resume: "Experienced Java Developer with Spring Boot, RESTful APIs, and MySQL skills."
- **Expected Result:**
  - Match Score: 45
  - No candidate name or email prompt.
  - No data inserted into the database.

Test Cases

Test Case 1: High Match (Score ≥ 80)

Job Description

Copy code

Position: Software Engineer

Responsibilities:

- Develop scalable web applications using Java and Spring Boot.

- Collaborate with teams to design new features.

- Write clean, maintainable, and efficient code.

Requirements:

- Bachelor's degree in Computer Science or related field.

- Strong knowledge of Java and RESTful APIs.

- Experience with databases (e.g., MySQL, PostgreSQL).

Resume:

Name: John Doe

Skills: Java, Spring Boot, RESTful APIs, MySQL, PostgreSQL

Experience: 2 years as a Software Engineer developing scalable applications using Java and Spring Boot.

Education: Bachelor’s degree in Computer Science.

Expected Result:

- **Match Score:** 85
- **Outcome:** Candidate name and email prompted, details stored in the database.

Test Case 2: Low Match (Score < 80)

Job Description

Position: Python Developer

Responsibilities:

- Develop web applications using Flask.

- Write and optimize database queries (PostgreSQL).

- Experience with REST APIs.

Requirements:

- Bachelor's degree in Computer Science.

- Knowledge of Python, Flask, PostgreSQL.

- Familiarity with JavaScript.

Resume

Name: Jane Smith

Skills: Java, Spring Boot, MySQL, REST APIs

Experience: 3 years as a Software Developer using Java and Spring Boot.

Education: Bachelor’s degree in Information Technology.

Expected Result:

- **Match Score:** 60

- **Outcome:** No prompt for name and email. Resume not stored in the database.

How to Run the Applicant Tracking System (ATS)

Pre-requisites

1. **Java Development Kit (JDK) 11 or later** installed.
2. **MySQL Server** installed and running.
3. **MySQL Connector JAR** file downloaded and placed in the project lib folder.
4. **JavaFX SDK** downloaded and properly configured in the project.
5. **Database setup:** You should have already created the ats_db database using the provided SQL schema.

Setting Up JavaFX

1. **Download JavaFX SDK:**
   - Visit the JavaFX website and download the appropriate SDK for your operating system.
2. **Unpack the SDK:**
   - Extract the JavaFX SDK to a location on your system, e.g., C:\javafx-sdk.
3. **Include JavaFX Libraries:**
   - Place the necessary JavaFX JAR files into the libdirectory in your project. The essential files include:
     - javafx.base.jar
     - javafx.controls.jar
     - javafx.fxml.jar
     - javafx.graphics.jar

Modify Your JavaFX Run Command:

- To run the project, use the following command (modify according to your SDK location):

java --module-path "path/to/your/javafx-sdk/lib" --add-modules javafx.controls,javafx.fxml -cp "bin;lib/*" Main

java --module-path "C:\javafx-sdk\lib" --add-modules javafx.controls,javafx.fxml -cp "bin;lib/*" Main

Here's a step-by-step guide on how to run the **Applicant Tracking System (ATS)** project, assuming you are using a Java-based console application that includes JavaFX and MySQL integration, without Maven.

How to Run the Applicant Tracking System (ATS)

Pre-requisites

1. **Java Development Kit (JDK) 11 or later** installed.
2. **MySQL Server** installed and running.
3. **MySQL Connector JAR** file downloaded and placed in the project lib folder.
4. **JavaFX SDK** downloaded and properly configured in the project.
5. **Database setup:** You should have already created the ats_db database using the provided SQL schema.

Project Structure

Ensure that your project structure is as follows:

bash

Copy code

/ApplicantTrackingSystem

/bin # Compiled Java class files

/lib # Required JAR files (JavaFX, MySQL Connector)

javafx.base.jar

javafx.controls.jar

javafx.fxml.jar

mysql-connector-java.jar

/src # Source code files

Main.java

DatabaseHandler.java

ScoreCalculator.java

FileChooserApp.java

/resources # Any additional resources if required (optional)

Setting Up JavaFX

1. **Download JavaFX SDK:**
   - Visit the JavaFX website and download the appropriate SDK for your operating system.
2. **Unpack the SDK:**
   - Extract the JavaFX SDK to a location on your system, e.g., C:\javafx-sdk.
3. **Include JavaFX Libraries:**
   - Place the necessary JavaFX JAR files into the libdirectory in your project. The essential files include:
     - javafx.base.jar
     - javafx.controls.jar
     - javafx.fxml.jar
     - javafx.graphics.jar
4. **Modify Your JavaFX Run Command:**
   - To run the project, use the following command (modify according to your SDK location):
   - bash
   - Copy code
   - java --module-path "path/to/your/javafx-sdk/lib" --add-modules javafx.controls,javafx.fxml -cp "bin;lib/*" Main
   - For example, if JavaFX is located at C:\javafx-sdk, you can run:
   - bash
   - Copy code
   - java --module-path "C:\javafx-sdk\lib" --add-modules javafx.controls,javafx.fxml -cp "bin;lib/*" Main

Database Setup

```
CREATE DATABASE ats_db;

USE ats_db;



CREATE TABLE shortlisted_resumes (

    id INT AUTO_INCREMENT PRIMARY KEY,

    candidate_name VARCHAR(100),

    resume_text TEXT,

    score DECIMAL(5, 2)

);
```

Compiling the Project

1. **Open a terminal/command prompt** in the root directory of your project (ApplicantTrackingSystem).
2. **Compile the Java files** using the following command:

javac --module-path "path/to/your/javafx-sdk/lib" --add-modules javafx.controls,javafx.fxml -cp "lib/*" -d bin src/*.java

Running the Project

After successfully compiling the project, run the program using the following command:

java --module-path "path/to/your/javafx-sdk/lib" --add-modules javafx.controls,javafx.fxml