HireSmart CRM – Project Document

Github Link Repo : https://github.com/jyotiraditya0607/HireSmart-CRM

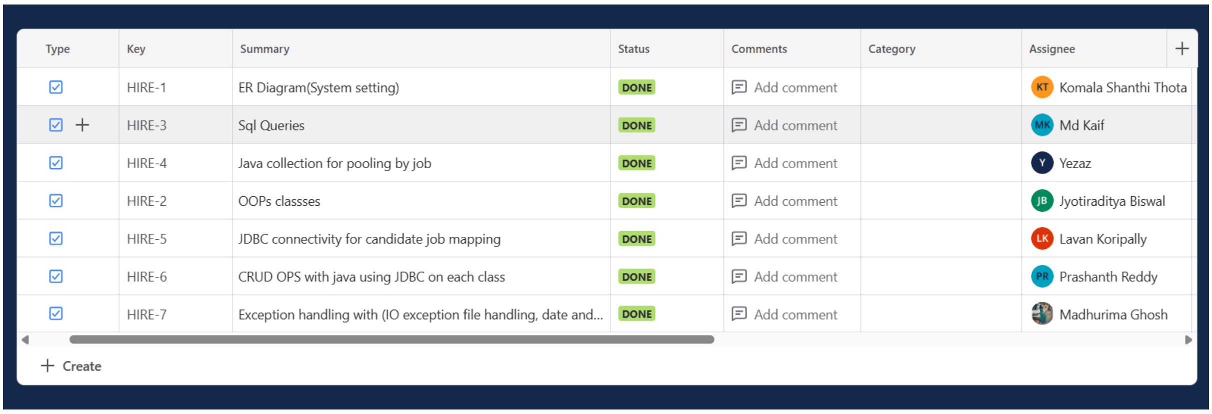
Notion Link: https://www.notion.so/215402dcc41d80478d1ffa6f8212e5f4?source=copy_link

1. Sprint Goal

In this Sprint, we focused primarily on the backend system for the recruitment platform. This involved designing the overall data model, implementing object-oriented structures, connecting with the database, and preparing for real-time application workflows like candidate-job matching, resume uploads, and interview tracking.

User Stories:

Jira Tool Burndown chart:



Type	Key	Summary	Status	Comments	Category	Assignee
☑	HIRE-1	ER Diagram(System setting)	DONE	Add comment		Komala Shanthi Thota
☑	HIRE-3	Sql Queries	DONE	Add comment		Md Kaif
☑	HIRE-4	Java collection for pooling by job	DONE	Add comment		Yezaz
☑	HIRE-2	OOPs classses	DONE	Add comment		Jyotiraditya Biswal
☑	HIRE-5	JDBC connectivity for candidate job mapping	DONE	Add comment		Lavan Koripally
☑	HIRE-6	CRUD OPS with java using JDBC on each class	DONE	Add comment		Prashanth Reddy
☑	HIRE-7	Exception handling with (IO exception file handling, date and...	DONE	Add comment		Madhurima Ghosh

Fig.1: users assign with each task

User Story 1: ER Diagram Design
As a system architect,
I want to design a normalized ER diagram for the core entities (Candidate, Job, Interview, Recruiter),
so that the database structure is clear, scalable, and supports all recruitment operations.

User Story 2: OOP Class Implementation
As a backend developer,

```
I want to implement Java classes for Candidate and Job using interfaces and inheritance,
so that the codebase is modular, reusable, and follows OOP best practices.

User Story 3: SQL & CRUD Operations
As a database engineer,
I want to write SQL scripts and implement CRUD operations for Candidate and Job entities,
so that data can be created, read, updated, and deleted efficiently in the Oracle database.

User Story 4: Applicant Pool Management
As a recruiter,
I want to use Java collections and generics to manage pools of candidates for each job,
so that I can easily view, shortlist, and manage applicants for specific job postings.

User Story 5: JDBC Integration for Candidate-Job Mapping
As a software integrator,
I want to integrate Java JDBC to map candidates to jobs and perform database operations,
so that the application can persist and retrieve candidate-job relationships reliably.

User Story 6: Exception Handling, File I/O, and Interview Scheduling
As a system maintainer,
I want to add exception handling, file I/O for resume uploads, and date/time management for interview logs,
so that the system is robust, can handle errors gracefully, and supports real-world recruitment workflows.
```

2. Project Purpose and Scope

Purpose:

The primary purpose of the HireSmart CRM is to streamline and manage the core activities of a recruitment agency or internal HR department. It provides functionalities to keep track of recruiters, available job positions, potential candidates, their applications, and scheduled interviews.

Scope:
The current scope of the project includes:

- Managing core entities: Recruiters, Jobs, Candidates, and Interviews.
- Performing standard CRUD (Create, Read, Update, Delete) operations for each managed entity via a command-line interface.
- Storing data persistently in an Oracle database.
- Applying basic OOP principles and the DAO pattern.
- Handling dates and basic file path storage (for resumes).

Out of Scope:

- A graphical user interface (GUI).
- Advanced features like reporting, analytics, automated candidate matching, email notifications, integration with external systems, or user authentication/authorization.
- Complex error recovery or robust validation beyond basic input types.

3. Features

The HireSmart CRM offers the following key features:

- **Recruiter Management:**
    - Add new recruiters with details like name, email, phone, expertise, and active status.
    - View a list of all recruiters.
    - Update existing recruiter details.
    - Delete recruiters.
- **Job Management:**
    - Post new job openings, linking them to a recruiter. Includes details like title, description, requirements, location, salary range, and posted date.
    - View a list of all job postings.
    - Update existing job details.

    - Delete job postings.
- **Candidate Management:**
    - Add new candidates with personal details, skills, experience, resume file path, and status.
    - View a list of all candidates.
    - Update existing candidate details.
    - Delete candidates.
- **Interview Management:**
    - Schedule interviews linking a candidate, a job, and a recruiter. Includes date, time, and status.
    - View a list of all scheduled interviews.
    - Update interview details, including status, feedback, and rating.
    - Delete scheduled interviews.
- **Application Management (Implicit/Database Layer):** While not directly exposed in the `Main` CLI menu, the database schema includes an `APPLICATION` table to track candidate applications to specific jobs, indicating the intended relationship. The provided code does not expose CRUD for Applications directly through the main menu, but the structure supports it.
- **Menu-driven Interface:** An interactive command-line menu simplifies navigation and access to different management functionalities.
- **Data Persistence:** All data is stored in an Oracle database, ensuring data integrity and persistence across application sessions.

4. Technologies Used

- **Programming Language:** Java 8+
- **Database:** Oracle Database
- **Database Connectivity:** JDBC (Java Database Connectivity)
- **Design Patterns:** Data Access Object (DAO), Object-Oriented Programming (OOP)

5. Architecture and Design

The application follows a layered architecture:

- **Presentation Layer (CLI):** Handled by the `Main` class, which provides the user interface (command-line menu), takes user input, and orchestrates calls to the service/DAO layer.
- **Service/Business Logic Layer (Implicit):** The `Main` class contains some direct logic for handling user input and calling DAOs. A dedicated service layer could be introduced in a larger application for more complex business rules.
- **Data Access Layer (DAO):** Comprises the `dao` package (`RecruiterDAO`, `JobDAO`, `CandidateDAO`, `InterviewDAO`). These classes are responsible for all database interactions (CRUD operations) for their respective entities, abstracting the database details from the business logic. `DBConnection` manages the database connection pool or single connection.
- **Model Layer:** The `model` package contains Plain Old Java Objects (POJOs) representing the database entities (`Recruiter`, `Job`, `Candidate`, `Interview`, `Application`). These classes hold the data and may include basic getter/setter methods. An `Entity` interface and `Person` abstract class demonstrate OOP principles.
- **Utility Layer:** The `util` package (`DateUtil`, `FileUtil`) provides helper functions for common tasks like date formatting/parsing or simulating file operations (like resume handling).

The **DAO Pattern** is central to the design, separating data persistence logic from the rest of the application. The **OOP Design** uses inheritance (`Candidate` and `Recruiter` potentially extending `Person`, though `Person` seems unused in the provided code excerpt) and interfaces (`Entity`) for structure and potential future extensibility.

6. Database Design

The database schema is designed to store information about Recruiters, Jobs, Candidates, their Applications, and Interviews. The relationships are based on the Entity-Relationship Diagram (ERD).

6.1 Entity-Relationship Diagram (ERD)

```
erDiagram
    CANDIDATE ||--o{ APPLICATION : applies
    JOB ||--o{ APPLICATION : has
    RECRUITER ||--o{ JOB : posts
    CANDIDATE ||--o{ INTERVIEW : attends
    JOB ||--o{ INTERVIEW : for
    RECRUITER ||--o{ INTERVIEW : conducts
    CANDIDATE {
        int candidate_id PK
        string name
        string email
        string phone
        string skills
        int experience_years
        string resume_file_path
        string status
        date created_date
    }
    JOB {
        int job_id PK
        int recruiter_id FK
        string title
        string description
        string requirements
        string location
        double salary_min
        double salary_max
        string feedback
        date posted_date
    }
    RECRUITER {
        int recruiter_id PK
        string name
        string email
        string expertise
        string phone
        string active_status
    }
    APPLICATION {
        int application_id PK
        int candidate_id FK
        int job_id FK
        date application_date
        string status
    }
    INTERVIEW {
        int interview_id PK
        int candidate_id FK
        int job_id FK
        int recruiter_id FK
        date interview_date
        string interview_time
        string status
        string feedback
        int rating
    }
```

6.2 Schema Details

The database consists of the following tables:

RECRUITER

Column Name	Type	Constraints
`recruiter_id`	NUMBER	**PK**
`name`	VARCHAR2	
`email`	VARCHAR2	**UNIQUE**
`expertise`	VARCHAR2	
`phone`	VARCHAR2	
`active_status`	VARCHAR2	**CHECK** ('ACTIVE', 'INACTIVE')

CANDIDATE

Column Name	Type	Constraints
`candidate_id`	NUMBER	**PK**

`name`	VARCHAR2	
`email`	VARCHAR2	**UNIQUE**
`phone`	VARCHAR2	
`skills`	VARCHAR2	
`experience_years`	NUMBER	
`resume_file_path`	VARCHAR2	
`status`	VARCHAR2	**CHECK** ('ACTIVE', 'INACTIVE', 'HIRED')
`created_date`	DATE	**DEFAULT** `SYSDATE`

JOB

Column Name	Type	Constraints
`job_id`	NUMBER	**PK**
`recruiter_id`	NUMBER	**FK → RECRUITER(recruiter_id)**
`title`	VARCHAR2	
`description`	CLOB	
`requirements`	VARCHAR2	
`location`	VARCHAR2	
`salary_min`	NUMBER	
`salary_max`	NUMBER	
`feedback`	VARCHAR2	
`posted_date`	DATE	**DEFAULT** `SYSDATE`

APPLICATION

Column Name	Type	Constraints
`application_id`	NUMBER	**PK**
`candidate_id`	NUMBER	**FK → CANDIDATE(candidate_id), UNIQUE with job_id**
`job_id`	NUMBER	**FK → JOB(job_id), UNIQUE with candidate_id**
`application_date`	DATE	**DEFAULT** `SYSDATE`
`status`	VARCHAR2	**CHECK** ('APPLIED', 'SHORTLISTED', 'REJECTED', 'HIRED')

INTERVIEW

Column Name	Type	Constraints
`interview_id`	NUMBER	**PK**
`candidate_id`	NUMBER	**FK → CANDIDATE(candidate_id)**
`job_id`	NUMBER	**FK → JOB(job_id)**
`recruiter_id`	NUMBER	**FK → RECRUITER(recruiter_id)**
`interview_date`	DATE	
`interview_time`	VARCHAR2	
`status`	VARCHAR2	**CHECK** ('SCHEDULED', 'COMPLETED', 'CANCELLED', 'RESCHEDULED')
`feedback`	VARCHAR2	
`rating`	NUMBER	**CHECK** BETWEEN 1 AND 10

Sequences: Oracle sequences (recruiter_seq, candidate_seq, job_seq, application_seq, interview_seq) are used to generate unique primary key values for each table.

7. Code Structure

```
HireSmart-CRM/
├── README.md        (Project description and setup instructions)
├── sql/
│   └── schema.sql     (Database schema creation and sample data)
└── src/
    └── com/
        └── hiresmart/
            ├── dao/        (Data Access Objects - interact with the database)
            │   ├── CandidateDAO.java
            │   ├── DBConnection.java  (Handles database connection)
            │   ├── InterviewDAO.java
            │   ├── JobDAO.java
            │   └── RecruiterDAO.java
            ├── Main.java   (Main application class, CLI interface)
            ├── model/     (Model classes - represent database entities)
            │   ├── Application.java
            │   ├── Candidate.java
            │   ├── Interview.java
            │   ├── Job.java
            │   ├── Person.java      (Abstract class, potential parent for Cand date/Recruiter)
            │   ├── Recruiter.java
            │   └── Entity.java      (Interface for model entities)
            └── util/      (Utility classes)
                ├── DateUtil.java     (Date formatting and parsing)
                └── FileUtil.java     (Potential file handling utilities)
```

8. Setup and Installation

To set up and run the HireSmart CRM, follow these steps:

1. **Database Setup:**
    - Ensure you have access to an Oracle database instance.
    - Connect to your Oracle database using a database client (e.g., SQL Developer, SQL*Plus).
    - Execute the SQL script located at sql/schema.sql. This script will create the necessary tables, sequences, and insert sample data.
2. **Java Development Kit (JDK):**
    - Make sure you have Java 8 or a higher version installed on your system.
3. **Oracle JDBC Driver:**
    - Download the appropriate Oracle JDBC driver JAR file (e.g., ojdbc8.jar for Java 8). Place this JAR file in a convenient location.
4. **Configure Database Connection:**
    - Open the src/com/hiresmart/dao/DBConnection.java file.
    - Update the database connection URL, username, and password variables within this file to match your Oracle database configuration.
5. **Compilation:**
    - Open a terminal or command prompt.

- Navigate to the root directory of the project (HireSmart-CRM/).
- Compile the Java source files, including the JDBC driver in the classpath. Replace ojdbc8.jar with the actual name and path of your driver file if necessary.

```
javac -cp ".:path/to/your/ojdbc8.jar" src/com/hiresmart/**/*.java
# On Windows, use a semicolon (;) instead of a colon (:)
# javac -cp ".;path\to\your\ojdbc8.jar" src/com/hiresmart/**/*.java
```

6. Execution:
    - After successful compilation, run the main application class, again including the JDBC driver and the compiled classes (or source directory) in the classpath.

    ```
    java -cp ".:path/to/your/ojdbc8.jar:src" com.hiresmart.Main
    # On Windows
    # java -cp ".;path\to\your\ojdbc8.jar;src" com.hiresmart.Main
    ```

9. Usage

```
--- HireSmart CRM ---
1. Manage Recruiters
2. Manage Jobs
3. Manage Candidates
4. Manage Interviews
5. Exit
Choose an option: 1

--- Manage Recruiters ---
1. Add Recruiter
2. View All Recruiters
3. Update Recruiter
4. Delete Recruiter
5. Back to Main Menu
Choose an option: 1
Enter Name: John Smith
```

```
Enter Email: john.smith@hiresmart.com
Enter Phone: 9876543210
Enter Expertise: IT Recruitment
Recruiter added successfully!

--- Manage Recruiters ---
1. Add Recruiter
2. View All Recruiters
3. Update Recruiter
4. Delete Recruiter
5. Back to Main Menu
Choose an option: 2

--- All Recruiters ---
Recruiter{recruiterId=1, name='John Smith', email='john.smith@hiresmart.com', phone='9876543210', expertise='IT Recruitment', activeStatus='ACTIVE'}

--- Manage Recruiters ---
1. Add Recruiter
2. View All Recruiters
3. Update Recruiter
4. Delete Recruiter
5. Back to Main Menu
Choose an option: 5

--- HireSmart CRM ---
1. Manage Recruiters
2. Manage Jobs
3. Manage Candidates
4. Manage Interviews
5. Exit
Choose an option: 2

--- Manage Jobs ---
1. Add Job
2. View All Jobs
3. Update Job
```

```
4. Delete Job
5. Back to Main Menu
Choose an option: 1
Enter Title: Java Developer
Enter Description: Develop enterprise applications using Java
Enter Requirements: Java, Spring Boot, Oracle DB
Enter Location: Bangalore
Enter Minimum Salary: 600000
Enter Maximum Salary: 900000
Enter Recruiter ID: 1
Job added successfully!

--- Manage Jobs ---
1. Add Job
2. View All Jobs
3. Update Job
4. Delete Job
5. Back to Main Menu
Choose an option: 2

--- All Jobs ---
Job{jobId=1, recruiterId=1, title='Java Developer', description='Develop enterprise applications using Java', requirements='Java, Spring Boot, Oracle DB', location='Bangalore', salaryMin=600000.0, salaryMax=900000.0, feedback='null', postedDate=2024-07-01}

--- Manage Jobs ---
1. Add Job
2. View All Jobs
3. Update Job
4. Delete Job
5. Back to Main Menu
Choose an option: 5

--- HireSmart CRM ---
1. Manage Recruiters
2. Manage Jobs
3. Manage Candidates
```

```
4. Manage Interviews
5. Exit
Choose an option: 3

--- Manage Candidates ---
1. Add Candidate
2. View All Candidates
3. Update Candidate
4. Delete Candidate
5. Back to Main Menu
Choose an option: 1
Enter Name: Alice Wilson
Enter Email: alice.wilson@email.com
Enter Phone: 9123456789
Enter Skills: Java, Spring, Oracle
Enter Experience Years: 3
Enter Resume File Path: /path/to/alice_resume.pdf
Candidate added successfully!

--- Manage Candidates ---
1. Add Candidate
2. View All Candidates
3. Update Candidate
4. Delete Candidate
5. Back to Main Menu
Choose an option: 2

--- All Candidates ---
Candidate{candidateId=1, name='Alice Wilson', email='alice.wilson@email.com', phone='9123456789', skills='Java, Spring, Oracle', experienceYears=3, resumeFilePath='/path/to/alice_resume.pdf', status='ACTIVE', createdDate=2024-07-01}

--- Manage Candidates ---
1. Add Candidate
2. View All Candidates
3. Update Candidate
4. Delete Candidate
```

```
5. Back to Main Menu
Choose an option: 5

--- HireSmart CRM ---
1. Manage Recruiters
2. Manage Jobs
3. Manage Candidates
4. Manage Interviews
5. Exit
Choose an option: 4

--- Manage Interviews ---
1. Schedule Interview
2. View All Interviews
3. Update Interview
4. Delete Interview
5. Back to Main Menu
Choose an option: 1
Enter Candidate ID: 1
Enter Job ID: 1
Enter Recruiter ID: 1
Enter Interview Date (YYYY-MM-DD): 2024-07-10
Enter Interview Time: 10:00 AM
Interview scheduled successfully!

--- Manage Interviews ---
1. Schedule Interview
2. View All Interviews
3. Update Interview
4. Delete Interview
5. Back to Main Menu
Choose an option: 2

--- All Interviews ---
Interview{interviewId=1, candidateId=1, jobId=1, recruiterId=1, interviewDate=2024-07-10, interviewTime='10:00 AM', status='SCHEDULED', feedback='null', rating=0}
```

```
--- Manage Interviews ---
1. Schedule Interview
2. View All Interviews
3. Update Interview
4. Delete Interview
5. Back to Main Menu
Choose an option: 3
Enter Interview ID to update: 1
Enter new Interview Date (current: 2024-07-10): 2024-07-12
Enter new Interview Time (current: 10:00 AM): 11:00 AM
Enter new Status (current: SCHEDULED): COMPLETED
Enter Feedback: Excellent technical skills, recommended for hire.
Enter Rating (1-10): 9
Interview updated successfully!

--- Manage Interviews ---
1. Schedule Interview
2. View All Interviews
3. Update Interview
4. Delete Interview
5. Back to Main Menu
Choose an option: 2

--- All Interviews ---
Interview{interviewId=1, candidateId=1, jobId=1, recruiterId=1, interviewDate=2024-07-12, interviewTime='11:00 AM', status='COMPLETED', feedback='Excellent technical skills, recommended for hire.', rating=9}

--- Manage Interviews ---
1. Schedule Interview
2. View All Interviews
3. Update Interview
4. Delete Interview
5. Back to Main Menu
Choose an option: 5

--- HireSmart CRM ---
1. Manage Recruiters
```

````
2. Manage Jobs
3. Manage Candidates
4. Manage Interviews
5. Exit
Choose an option: 5

Exiting...
````

- **update** and **delete** operations

~~~
---

```
--- HireSmart CRM ---
1. Manage Recruiters
2. Manage Jobs
3. Manage Candidates
4. Manage Interviews
5. Exit
Choose an option: 1

--- Manage Recruiters ---
1. Add Recruiter
2. View All Recruiters
3. Update Recruiter
4. Delete Recruiter
5. Back to Main Menu
Choose an option: 3
Enter Recruiter ID to update: 1
Enter new Name (current: John Smith): John S. Smith
Enter new Email (current: john.smith@hiresmart.com): john.smith@hiresmart.c
Enter new Phone (current: 9876543210): 9876543211
Enter new Expertise (current: IT Recruitment): IT & Tech Recruitment
Enter new Status (current: ACTIVE): ACTIVE
Recruiter updated successfully!

--- Manage Recruiters ---
1. Add Recruiter
~~~

```
2. View All Recruiters
3. Update Recruiter
4. Delete Recruiter
5. Back to Main Menu
Choose an option: 2

--- All Recruiters ---
Recruiter{recruiterId=1, name='John S. Smith', email='john.smith@hiresmart.c

--- Manage Recruiters ---
1. Add Recruiter
2. View All Recruiters
3. Update Recruiter
4. Delete Recruiter
5. Back to Main Menu
Choose an option: 4
Enter Recruiter ID to delete: 1
Recruiter deleted successfully!

--- Manage Recruiters ---
1. Add Recruiter
2. View All Recruiters
3. Update Recruiter
4. Delete Recruiter
5. Back to Main Menu
Choose an option: 2

--- All Recruiters ---
(no output, list is empty if you deleted the only recruiter)

--- HireSmart CRM ---
1. Manage Recruiters
2. Manage Jobs
3. Manage Candidates
4. Manage Interviews
5. Exit
Choose an option: 2
```

```
--- Manage Jobs ---
1. Add Job
2. View All Jobs
3. Update Job
4. Delete Job
5. Back to Main Menu
Choose an option: 3
Enter Job ID to update: 1
Enter new Title (current: Java Developer): Senior Java Developer
Enter new Description (current: Develop enterprise applications using Java): L
Enter new Requirements (current: Java, Spring Boot, Oracle DB): Java, Spring
Enter new Location (current: Bangalore): Bangalore
Enter new Minimum Salary (current: 600000.0): 800000
Enter new Maximum Salary (current: 900000.0): 1200000
Enter new Recruiter ID (current: 1): 1
Job updated successfully!

--- Manage Jobs ---
1. Add Job
2. View All Jobs
3. Update Job
4. Delete Job
5. Back to Main Menu
Choose an option: 2

--- All Jobs ---
Job{jobId=1, recruiterId=1, title='Senior Java Developer', description='Lead Ja

--- Manage Jobs ---
1. Add Job
2. View All Jobs
3. Update Job
4. Delete Job
5. Back to Main Menu
Choose an option: 4
Enter Job ID to delete: 1
Job deleted successfully!
```

```
--- Manage Jobs ---
1. Add Job
2. View All Jobs
3. Update Job
4. Delete Job
5. Back to Main Menu
Choose an option: 2

--- All Jobs ---
(no output, list is empty if you deleted the only job)

--- HireSmart CRM ---
1. Manage Recruiters
2. Manage Jobs
3. Manage Candidates
4. Manage Interviews
5. Exit
Choose an option: 3

--- Manage Candidates ---
1. Add Candidate
2. View All Candidates
3. Update Candidate
4. Delete Candidate
5. Back to Main Menu
Choose an option: 3
Enter Candidate ID to update: 1
Enter new Name (current: Alice Wilson): Alice M. Wilson
Enter new Email (current: alice.wilson@email.com): alice.wilson@email.com
Enter new Phone (current: 9123456789): 9123456790
Enter new Skills (current: Java, Spring, Oracle): Java, Spring, Oracle, Microser
Enter new Experience Years (current: 3): 4
Enter new Resume File Path (current: /path/to/alice_resume.pdf): /path/to/alic
Enter new Status (current: ACTIVE): HIRED
Candidate updated successfully!

--- Manage Candidates ---
1. Add Candidate
```

```
2. View All Candidates
3. Update Candidate
4. Delete Candidate
5. Back to Main Menu
Choose an option: 2

--- All Candidates ---
Candidate{candidateId=1, name='Alice M. Wilson', email='alice.wilson@email.

--- Manage Candidates ---
1. Add Candidate
2. View All Candidates
3. Update Candidate
4. Delete Candidate
5. Back to Main Menu
Choose an option: 4
Enter Candidate ID to delete: 1
Candidate deleted successfully!

--- Manage Candidates ---
1. Add Candidate
2. View All Candidates
3. Update Candidate
4. Delete Candidate
5. Back to Main Menu
Choose an option: 2

--- All Candidates ---
(no output, list is empty if you deleted the only candidate)

--- HireSmart CRM ---
1. Manage Recruiters
2. Manage Jobs
3. Manage Candidates
4. Manage Interviews
5. Exit
Choose an option: 4
```

```
--- Manage Interviews ---
1. Schedule Interview
2. View All Interviews
3. Update Interview
4. Delete Interview
5. Back to Main Menu
Choose an option: 3
Enter Interview ID to update: 1
Enter new Interview Date (current: 2024-07-12): 2024-07-15
Enter new Interview Time (current: 11:00 AM): 2:00 PM
Enter new Status (current: COMPLETED): CANCELLED
Enter Feedback: Candidate unavailable, interview cancelled.
Enter Rating (1-10): 0
Interview updated successfully!

--- Manage Interviews ---
1. Schedule Interview
2. View All Interviews
3. Update Interview
4. Delete Interview
5. Back to Main Menu
Choose an option: 2

--- All Interviews ---
Interview{interviewId=1, candidateId=1, jobId=1, recruiterId=1, interviewDate=

--- Manage Interviews ---
1. Schedule Interview
2. View All Interviews
3. Update Interview
4. Delete Interview
5. Back to Main Menu
Choose an option: 4
Enter Interview ID to delete: 1
Interview deleted successfully!

--- Manage Interviews ---
1. Schedule Interview
```

~~~
2. View All Interviews
3. Update Interview
4. Delete Interview
5. Back to Main Menu
Choose an option: 2

--- All Interviews ---
(no output, list is empty if you deleted the only interview)

--- HireSmart CRM ---
1. Manage Recruiters
2. Manage Jobs
3. Manage Candidates
4. Manage Interviews
5. Exit
Choose an option: 5

Exiting...
```


---
~~~