CODING CHALLENGE – CAREER HUB

NAME- Tejsinh J. Harale-Bhosale

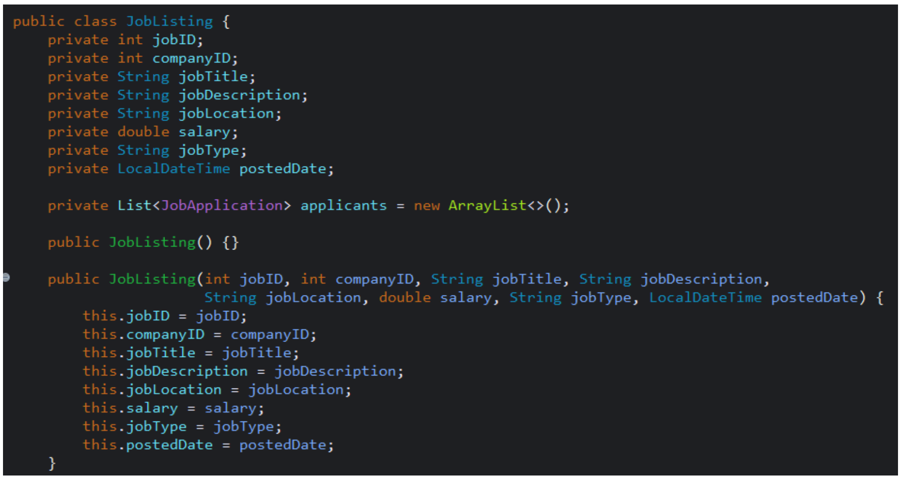
Github Repository: https://github.com/tejsinh3600/DA_foundation

1.Create and implement the mentioned class and the structure in your application.

<u>JobListing Class:</u>

Attributes:

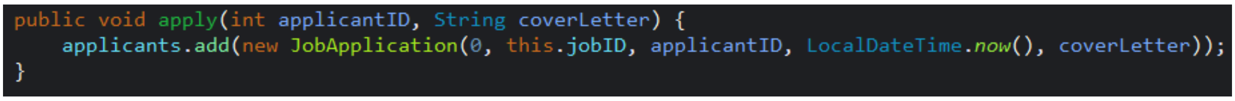
a. JobID (int): A unique identifier for each job listing.
b. CompanyID (int): A reference to the company offering the job.
c. JobTitle (string): The title of the job.
d. JobDescription (string): A detailed description of the job.
e. JobLocation (string): The location of the job.
f. Salary (decimal): The salary offered for the job.
g. JobType (string): The type of job (e.g., Full-time, Part-time, Contract).
h. PostedDate (DateTime): The date when the job was posted.



```
public class JobListing {
    private int jobID;
    private int companyID;
    private String jobTitle;
    private String jobDescription;
    private String jobLocation;
    private double salary;
    private String jobType;
    private LocalDateTime postedDate;

    private List<JobApplication> applicants = new ArrayList<>();

    public JobListing() {}

    public JobListing(int jobID, int companyID, String jobTitle, String jobDescription,
                      String jobLocation, double salary, String jobType, LocalDateTime postedDate) {
        this.jobID = jobID;
        this.companyID = companyID;
        this.jobTitle = jobTitle;
        this.jobDescription = jobDescription;
        this.jobLocation = jobLocation;
        this.salary = salary;
        this.jobType = jobType;
        this.postedDate = postedDate;
    }
```

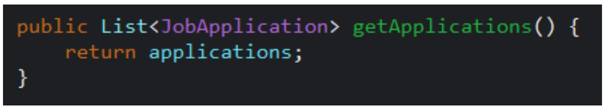
Methods:

• Apply(applicantID: int, coverLetter: string): Allows applicants to apply for the job by providing their ID and a cover letter.



```
public void apply(int applicantID, String coverLetter) {
    applicants.add(new JobApplication(0, this.jobID, applicantID, LocalDateTime.now(), coverLetter));
}
```

• GetApplicants(): List<Applicant>: Retrieves a list of applicants who have applied for the job.

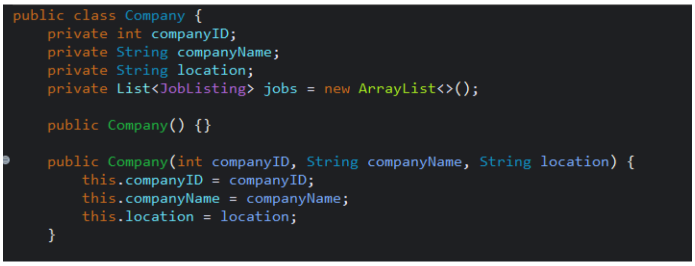


```
public List<JobApplication> getApplications() {
    return applications;
}
```

Company Class:

Attributes:

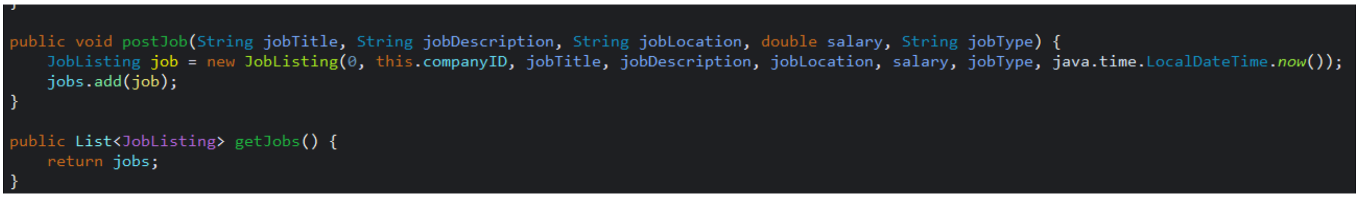
a. CompanyID (int): A unique identifier for each company.
b. CompanyName (string): The name of the hiring company.
c. Location (string): The location of the company.



```
public class Company {
    private int companyID;
    private String companyName;
    private String location;
    private List<JobListing> jobs = new ArrayList<>();

    public Company() {}

    public Company(int companyID, String companyName, String location) {
        this.companyID = companyID;
        this.companyName = companyName;
        this.location = location;
    }
```

Methods:

a. PostJob(jobTitle: string, jobDescription: string, jobLocation: string, salary: decimal, jobType:
b. string): Allows a company to post a new job listing.
c. GetJobs(): List<JobListing>: Retrieves a list of job listings posted by the company.

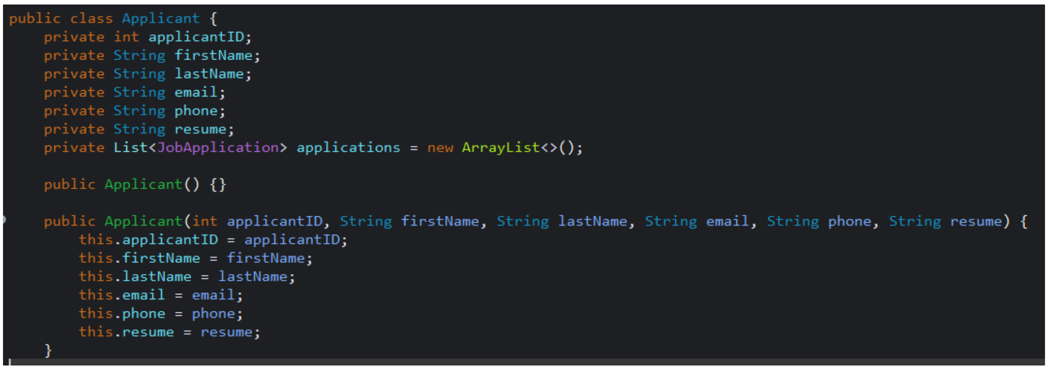


```
public void postJob(String jobTitle, String jobDescription, String jobLocation, double salary, String jobType) {
    JobListing job = new JobListing(0, this.companyID, jobTitle, jobDescription, jobLocation, salary, jobType, java.time.LocalDateTime.now());
    jobs.add(job);
}

public List<JobListing> getJobs() {
    return jobs;
}
```

Applicant Class:

Attributes:

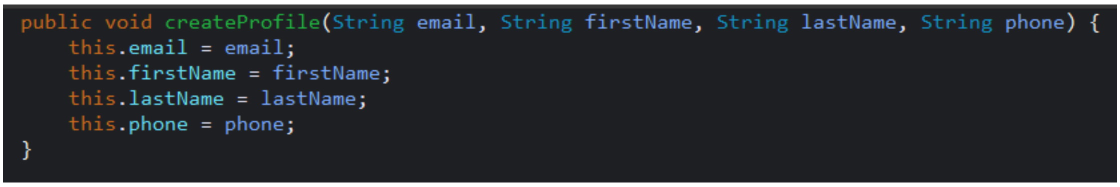
a. ApplicantID (int): A unique identifier for each applicant.
b. FirstName (string): The first name of the applicant.
c. LastName (string): The last name of the applicant.
d. Email (string): The email address of the applicant.
e. Phone (string): The phone number of the applicant.
f. Resume (string): The applicant's resume or a reference to the resume file.



```
public class Applicant {
    private int applicantID;
    private String firstName;
    private String lastName;
    private String email;
    private String phone;
    private String resume;
    private List<JobApplication> applications = new ArrayList<>();

    public Applicant() {}

    public Applicant(int applicantID, String firstName, String lastName, String email, String phone, String resume) {
        this.applicantID = applicantID;
        this.firstName = firstName;
        this.lastName = lastName;
        this.email = email;
        this.phone = phone;
        this.resume = resume;
    }
```

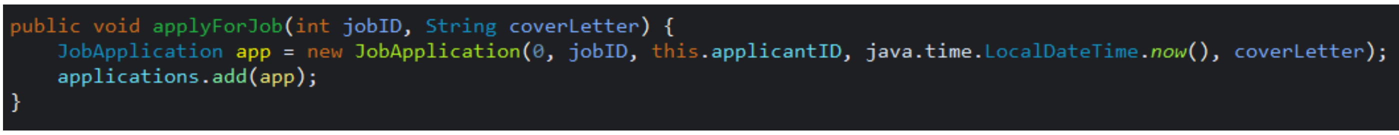
Methods:

- CreateProfile(email: string, firstName: string, lastName: string, phone: string): Allows applicants to create a profile with their contact information.



```java
public void createProfile(String email, String firstName, String lastName, String phone) {
    this.email = email;
    this.firstName = firstName;
    this.lastName = lastName;
    this.phone = phone;
}
```

- ApplyForJob(jobID: int, coverLetter: string): Enables applicants to apply for a specific job listing.

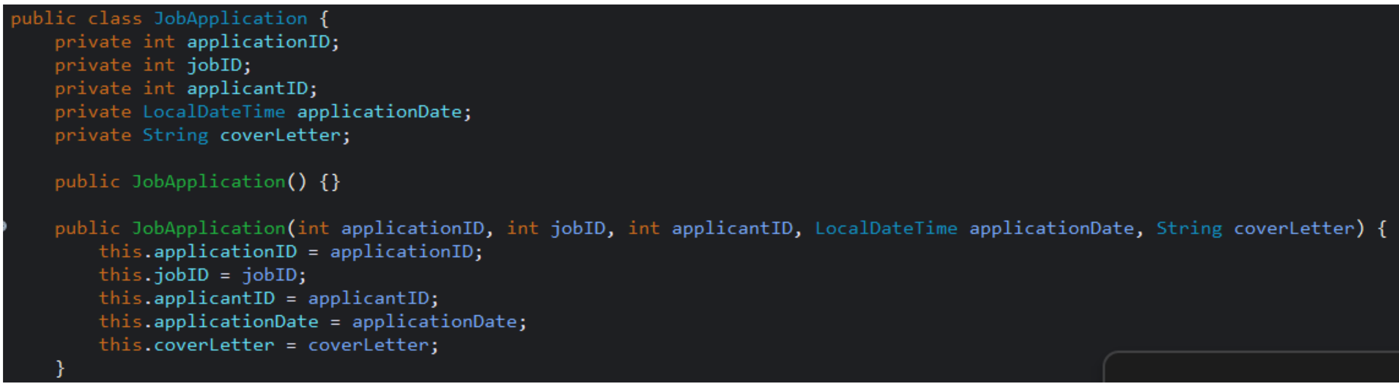


```java
public void applyForJob(int jobID, String coverLetter) {
    JobApplication app = new JobApplication(0, jobID, this.applicantID, java.time.LocalDateTime.now(), coverLetter);
    applications.add(app);
}
```

<u>JobApplication Class:</u>

Attributes:

a. ApplicationID (int): A unique identifier for each job application.
b. JobID (int): A reference to the job listing.
c. ApplicantID (int): A reference to the applicant.
d. ApplicationDate (DateTime): The date and time when the application was submitted.
e. CoverLetter (string): The cover letter submitted with the application.

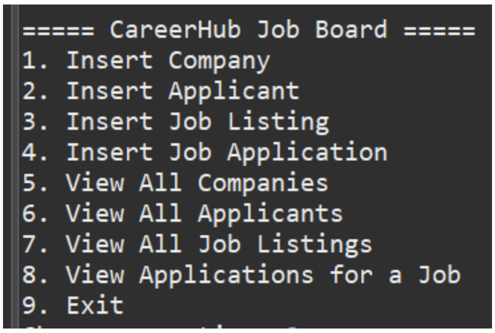


```java
public class JobApplication {
    private int applicationID;
    private int jobID;
    private int applicantID;
    private LocalDateTime applicationDate;
    private String coverLetter;

    public JobApplication() {}

    public JobApplication(int applicationID, int jobID, int applicantID, LocalDateTime applicationDate, String coverLetter) {
        this.applicationID = applicationID;
        this.jobID = jobID;
        this.applicantID = applicantID;
        this.applicationDate = applicationDate;
        this.coverLetter = coverLetter;
    }
```

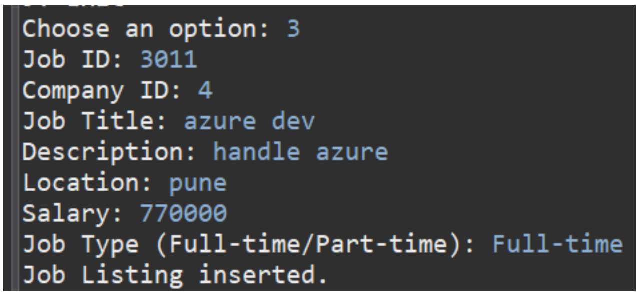
2.DatabaseManager Class:

- **InitializeDatabase(): Initializes the database schema and tables.**

- **InsertJobListing(job: JobListing): Inserts a new job listing into the "Jobs" table.**

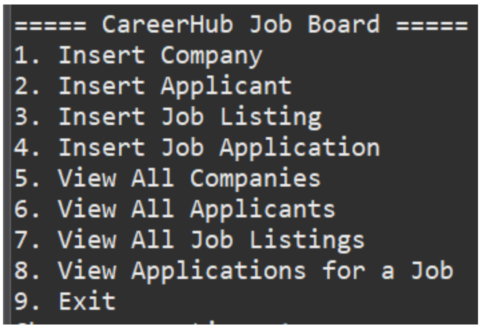


```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
```

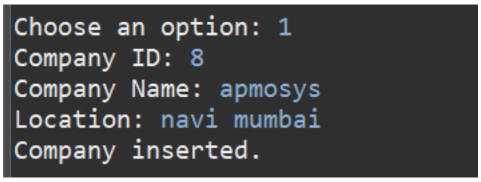


```
Choose an option: 3
Job ID: 3011
Company ID: 4
Job Title: azure dev
Description: handle azure
Location: pune
Salary: 770000
Job Type (Full-time/Part-time): Full-time
Job Listing inserted.
```

• InsertCompany(company: Company): Inserts a new company into the "Companies" table.

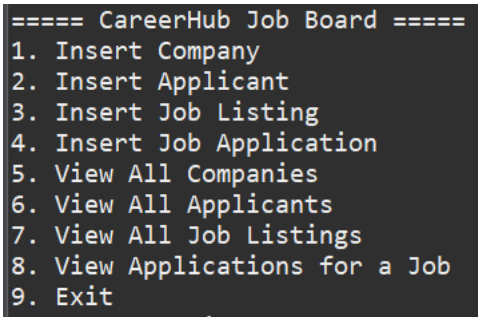


```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
```

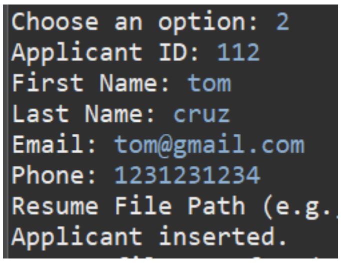


```
Choose an option: 1
Company ID: 8
Company Name: apmosys
Location: navi mumbai
Company inserted.
```

• InsertApplicant: Inserts a new applicant into the "Applicants" table.

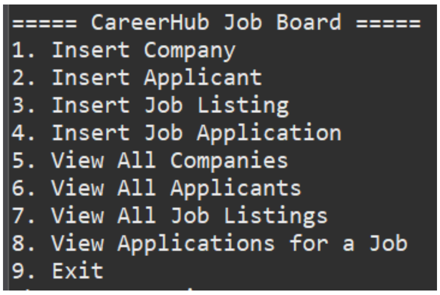


```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
```

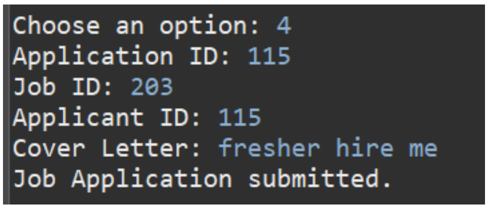


```
Choose an option: 2
Applicant ID: 112
First Name: tom
Last Name: cruz
Email: tom@gmail.com
Phone: 1231231234
Resume File Path (e.g.
Applicant inserted.
```

• InsertJobApplication: Inserts a new job application into the "Applications" table.

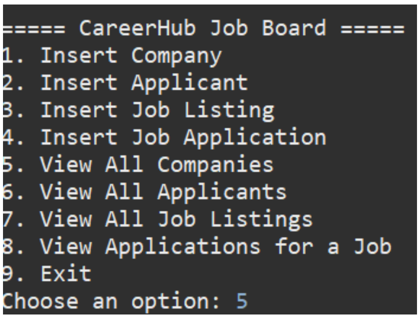


```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
```

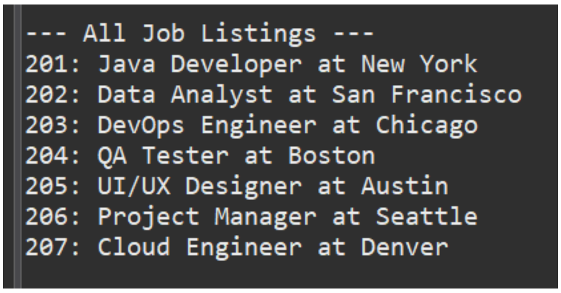


```
Choose an option: 4
Application ID: 115
Job ID: 203
Applicant ID: 115
Cover Letter: fresher hire me
Job Application submitted.
```

• GetJobListings(): List<JobListing>: Retrieves a list of all job listings.

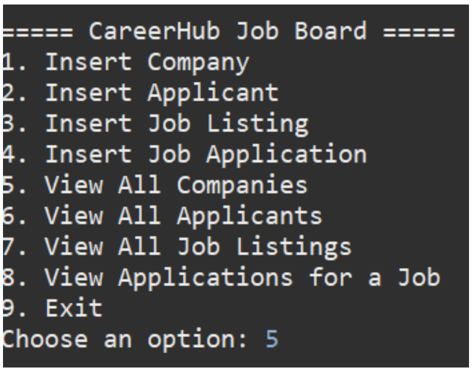


```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
Choose an option: 5
```

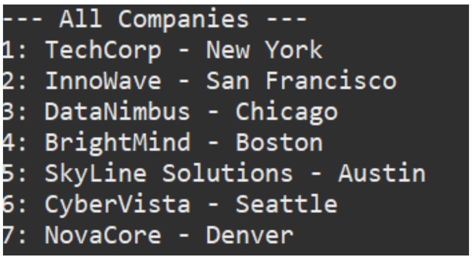


```
--- All Job Listings ---
201: Java Developer at New York
202: Data Analyst at San Francisco
203: DevOps Engineer at Chicago
204: QA Tester at Boston
205: UI/UX Designer at Austin
206: Project Manager at Seattle
207: Cloud Engineer at Denver
```

- **GetCompanies(): List<Company>: Retrieves a list of all companies.**

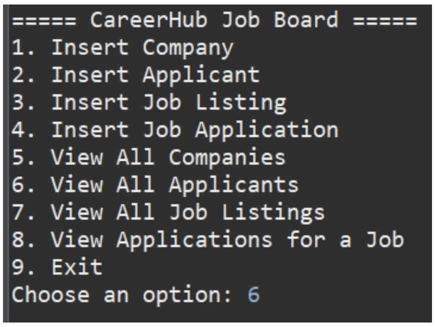


```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
Choose an option: 5
```

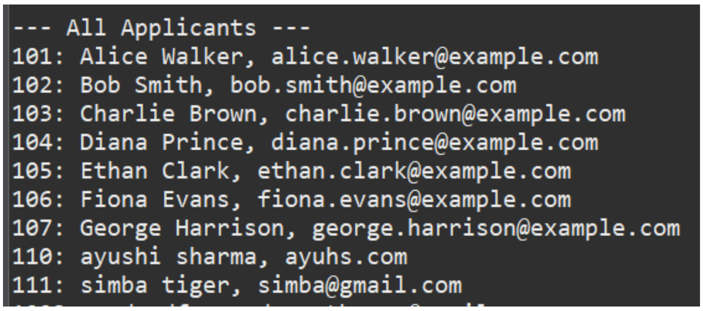


```
--- All Companies ---
1: TechCorp - New York
2: InnoWave - San Francisco
3: DataNimbus - Chicago
4: BrightMind - Boston
5: SkyLine Solutions - Austin
6: CyberVista - Seattle
7: NovaCore - Denver
```

- **GetApplicants(): List<Applicant>: Retrieves a list of all applicants.**

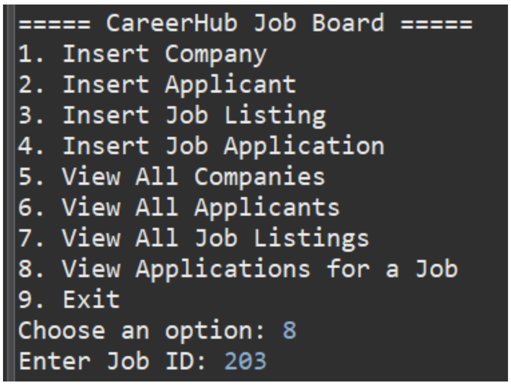


```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
Choose an option: 6
```

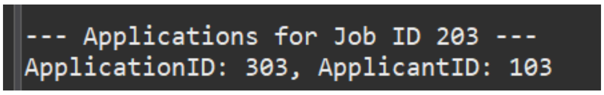


```
--- All Applicants ---
101: Alice Walker, alice.walker@example.com
102: Bob Smith, bob.smith@example.com
103: Charlie Brown, charlie.brown@example.com
104: Diana Prince, diana.prince@example.com
105: Ethan Clark, ethan.clark@example.com
106: Fiona Evans, fiona.evans@example.com
107: George Harrison, george.harrison@example.com
110: ayushi sharma, ayuhs.com
111: simba tiger, simba@gmail.com
```

- **GetApplicationsForJob(jobID: int): List<JobApplication>: Retrieves a list of job applications for a specific job listing.**



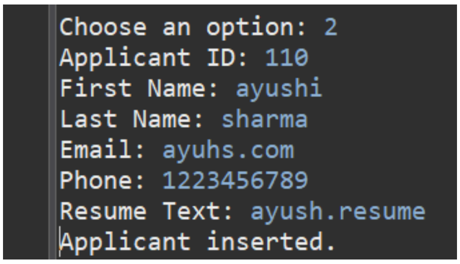
```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
Choose an option: 8
Enter Job ID: 203
```



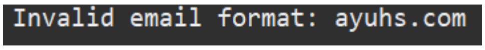
```
--- Applications for Job ID 203 ---
ApplicationID: 303, ApplicantID: 103
```

3.Exceptions handling

- **Invalid Email Format Handling**:

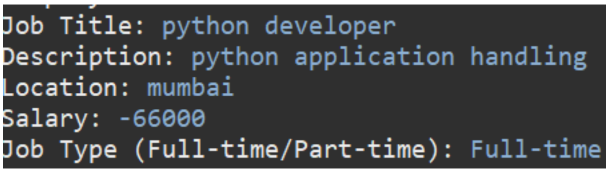


```
Choose an option: 2
Applicant ID: 110
First Name: ayushi
Last Name: sharma
Email: ayuhs.com
Phone: 1223456789
Resume Text: ayush.resume
Applicant inserted.
```

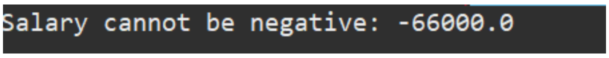


```
Invalid email format: ayuhs.com
```

- **Salary Calculation Handling:**

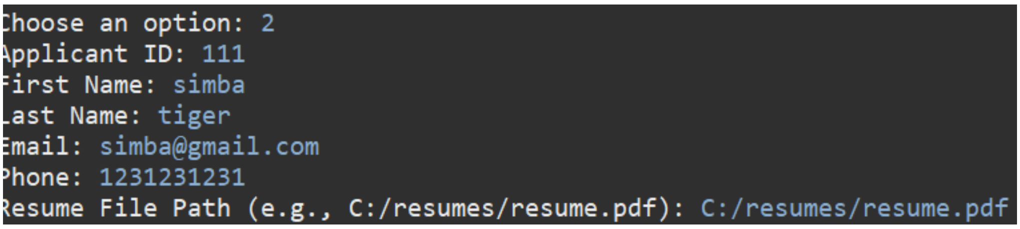


```
Job Title: python developer
Description: python application handling
Location: mumbai
Salary: -66000
Job Type (Full-time/Part-time): Full-time
```

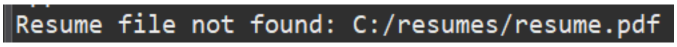


```
Salary cannot be negative: -66000.0
```

- **File Upload Exception Handling:**



```
Choose an option: 2
Applicant ID: 111
First Name: simba
Last Name: tiger
Email: simba@gmail.com
Phone: 1231231231
Resume File Path (e.g., C:/resumes/resume.pdf): C:/resumes/resume.pdf
```

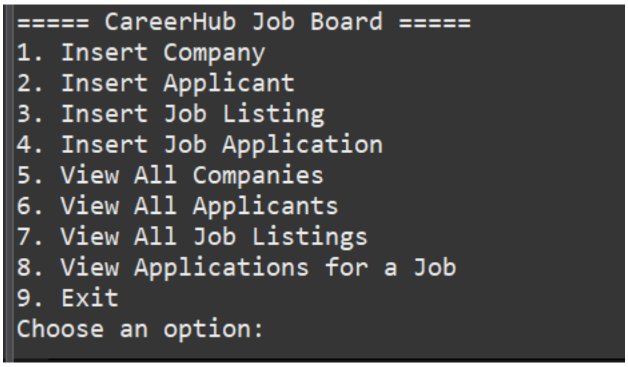


```
Resume file not found: C:/resumes/resume.pdf
```

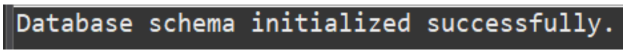
- **Application Deadline Handling:**

o Develop a program that checks whether a job application is submitted before the application deadline. Implement exception handling to catch situations where an applicant tries to submit an application after the deadline has passed. Display a message indicating that the application is no longer accepted.

- **Database Connection Handling:**



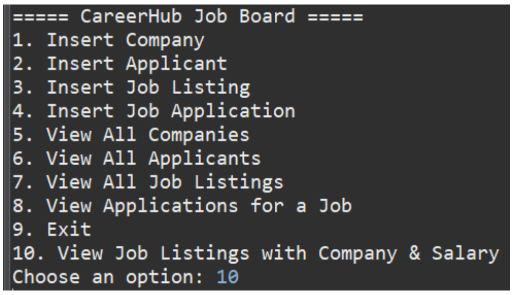
```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
Choose an option:
```



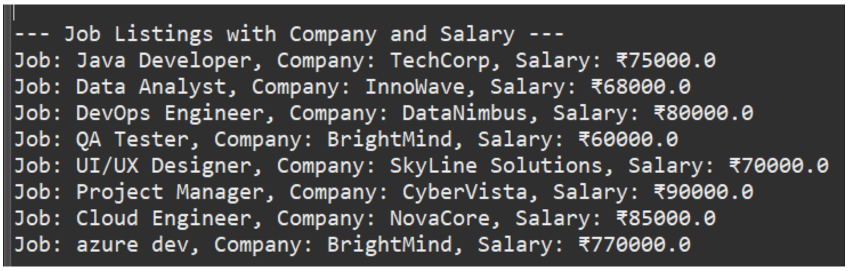
```
Database schema initialized successfully.
```

4.Database Connectivity

• Job Listing Retrieval:

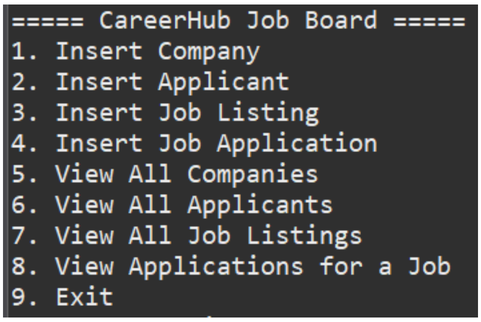


```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
10. View Job Listings with Company & Salary
Choose an option: 10
```

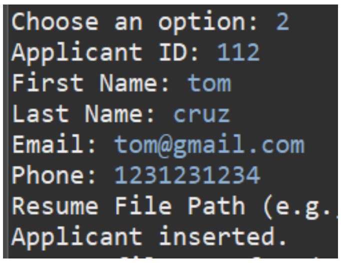


```
--- Job Listings with Company and Salary ---
Job: Java Developer, Company: TechCorp, Salary: ₹75000.0
Job: Data Analyst, Company: InnoWave, Salary: ₹68000.0
Job: DevOps Engineer, Company: DataNimbus, Salary: ₹80000.0
Job: QA Tester, Company: BrightMind, Salary: ₹60000.0
Job: UI/UX Designer, Company: SkyLine Solutions, Salary: ₹70000.0
Job: Project Manager, Company: CyberVista, Salary: ₹90000.0
Job: Cloud Engineer, Company: NovaCore, Salary: ₹85000.0
Job: azure dev, Company: BrightMind, Salary: ₹770000.0
```

• Applicant Profile Creation:

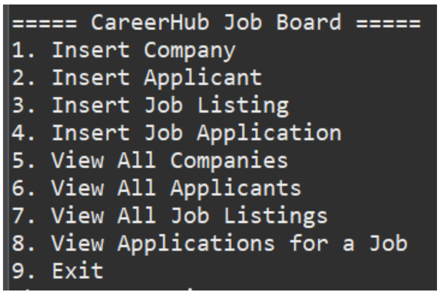


```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
```

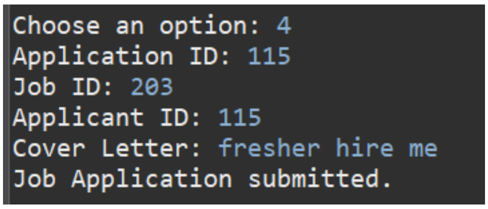


```
Choose an option: 2
Applicant ID: 112
First Name: tom
Last Name: cruz
Email: tom@gmail.com
Phone: 1231231234
Resume File Path (e.g.
Applicant inserted.
```

- **Job Application Submission:**

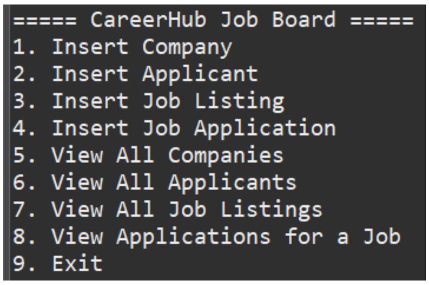


```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
```

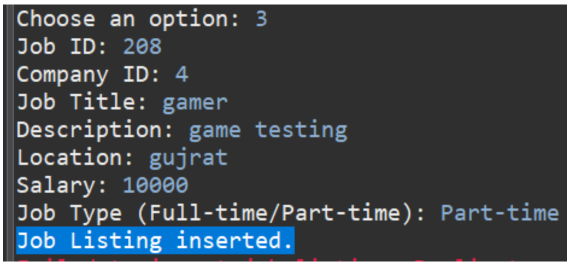


```
Choose an option: 4
Application ID: 115
Job ID: 203
Applicant ID: 115
Cover Letter: fresher hire me
Job Application submitted.
```

- **Company Job Posting:**

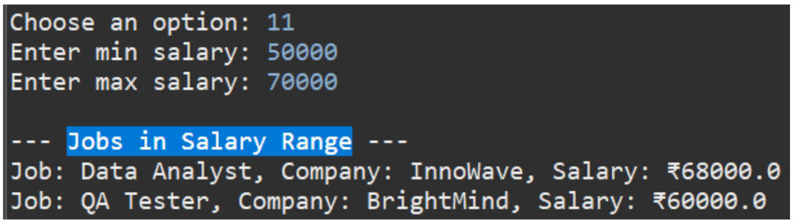


```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
```

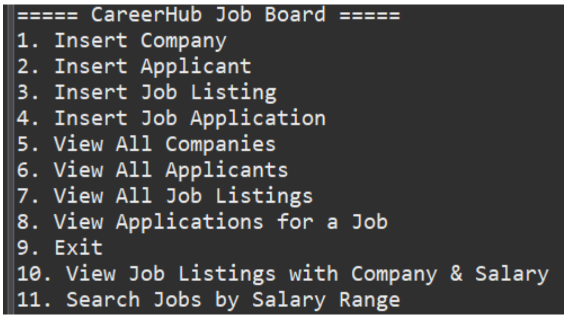


```
Choose an option: 3
Job ID: 208
Company ID: 4
Job Title: gamer
Description: game testing
Location: gujrat
Salary: 10000
Job Type (Full-time/Part-time): Part-time
Job Listing inserted.
```

- **Salary Range Query:**



```
Choose an option: 11
Enter min salary: 50000
Enter max salary: 70000

--- Jobs in Salary Range ---
Job: Data Analyst, Company: InnoWave, Salary: ₹68000.0
Job: QA Tester, Company: BrightMind, Salary: ₹60000.0
```



```
===== CareerHub Job Board =====
1. Insert Company
2. Insert Applicant
3. Insert Job Listing
4. Insert Job Application
5. View All Companies
6. View All Applicants
7. View All Job Listings
8. View Applications for a Job
9. Exit
10. View Job Listings with Company & Salary
11. Search Jobs by Salary Range
```