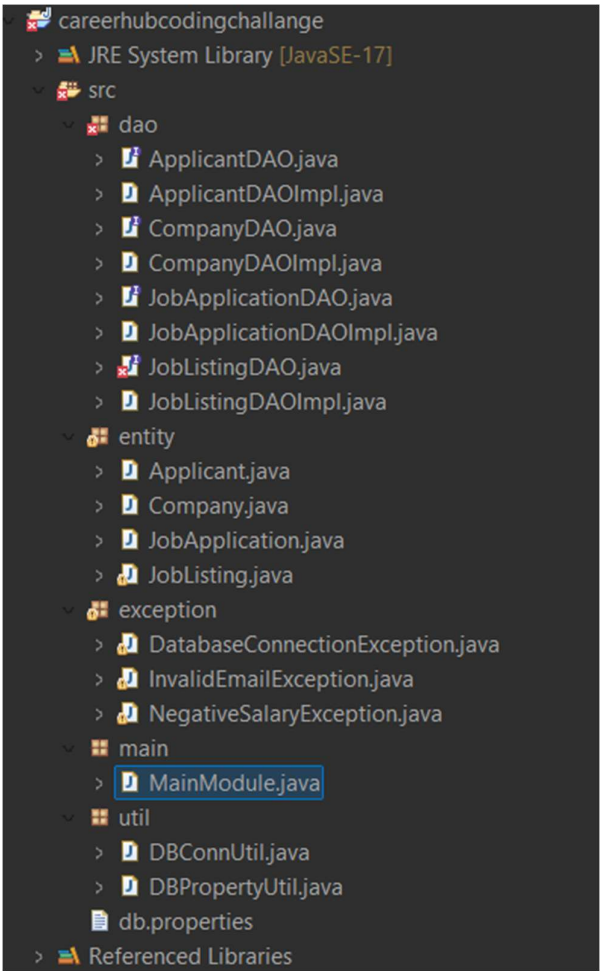


CareerHub - The Job Board

Directory structure:



Database : careerhub

Tables :

1. Applicants

ApplicantID	FirstName	LastName	Email	Phone	Resume
1	Rahul	Verma	rahul.verma@example.com	9876543210	rahul_resume.pdf
2	Sneha	Patil	sneha.patil@example.com	9123456780	sneha_resume.pdf
3	Arjun	Mehta	arjun.mehta@example.com	9988776655	arjun_resume.pdf
4	Mohan	R	mohan@gmail.com	9988789889	mohan.pdf

2. Applications

ApplicationID	JobID	ApplicantID	ApplicationDate	CoverLetter
13	8	1	2025-04-09 12:42:42	cover4.cv
14	10	3	2025-04-09 13:01:28	cv.cv

3. Companies

CompanyID	CompanyName	Location
2	Tata Consultancy Services	Mumbai
3	Infosys Limited	Bengaluru
4	Wipro Technologies	Hyderabad
5	Google	USA

4. Jobs

JobID	CompanyID	JobTitle	JobDescription	JobLocation	Salary	JobType	PostedDate
8	2	Java Developer	Build RESTful services using Spring Boot.	Pune	800000.00	Full-time	2025-04-09 12:31:36
9	3	Data Analyst	Analyze datasets to support business decisions.	Bangalore	700000.00	Full-time	2025-04-09 12:31:36
10	4	Frontend Engineer	Develop interactive UIs using React.	Hyderabad	750000.00	Contract	2025-04-09 12:31:36
12	5	Cloud Developer	Cloud Developer	Cloud Developer	60000.00	Cloud Developer	2025-04-09 13:00:03

Code :

Package : entity

Class : Applicant.java

```
package entity;

public class Applicant {
    private int applicantID;
    private String firstName;
    private String lastName;
    private String email;
    private String phone;
    private String resume;

    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;
    }

    public int getApplicantID() { return applicantID; }
    public String getFirstName() { return firstName; }
    public String getLastName() { return lastName; }
    public String getEmail() { return email; }
    public String getPhone() { return phone; }
    public String getResume() { return resume; }

}
```

Class: Company

```
package entity;

public class Company {
```

```
    private int companyID;
    private String companyName;
    private String location;

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

    public int getCompanyID() { return companyID; }
    public String getCompanyName() { return companyName; }
    public String getLocation() { return location; }

}
```

Class : JobApplications

```
package entity;

import java.time.LocalDateTime;

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

    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;
    }

    public int getApplicationID() { return applicationID; }
```

```
public int getJobID() { return jobID; }  
public int getApplicantID() { return applicantID; }  
public LocalDateTime getApplicationDate() { return applicationDate; }  
public String getCoverLetter() { return coverLetter; }  
  
}
```

Class : Job Listing

```
package entity;  
import java.util.*;  
import java.time.LocalDateTime;  
  
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<Applicant> applicants = new ArrayList<>();  
  
    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;  
    }  
  
    public void apply(Applicant applicant, String coverLetter) {
```

```

        applicants.add(applicant);
        System.out.println("Applicant " + applicant.getFirstName() + " applied
successfully.");
    }

    public List<Applicant> getApplicants() {
        return applicants;
    }

    public int getJobID() { return jobID; }
    public String getJobTitle() { return jobTitle; }
    public double getSalary() { return salary; }
    public int getCompanyID() { return companyID; }

}

```

Package : dao

Interface : Applicantdao

```

package dao;
import java.util.List;

import entity.Applicant;

public interface ApplicantDAO {
    void insertApplicant(Applicant applicant);
    List<Applicant> getAllApplicants();
}

```

Interface : Companydao

```

package dao;
import java.util.List;

import entity.Company;

public interface CompanyDAO {
    void insertCompany(Company company);
}

```

```
List<Company> getAllCompanies();  
  
}
```

Interface : JobApplicationdao

```
package dao;  
  
import java.util.List;  
  
import entity.JobApplication;  
  
public interface JobApplicationDAO {  
    void insertJobApplication(JobApplication application);  
    List<JobApplication> getApplicationsByJobID(int jobID);  
  
}
```

Interface : JobListingdao

```
package dao;  
  
import java.util.List;  
  
import entity.JobListing;  
  
public interface JobListingDAO {  
    void insertJobListing(JobListing job);  
    List<JobListing> getAllJobListings();  
    List<JobListing> getJobListingsBySalaryRange(double minSalary, double  
maxSalary);  
}  
  
}
```

Class : ApplicantdaoImpl.java

```
package dao;
```

```

import java.sql.*;
import java.util.*;

import entity.Applicant;
import exception.InvalidEmailException;

public class ApplicantDAOImpl implements ApplicantDAO {
    private Connection conn;

    public ApplicantDAOImpl(Connection conn) {
        this.conn = conn;
    }

    public void insertApplicant(Applicant applicant) {
        try {
            if (!applicant.getEmail().contains("@")) {
                throw new InvalidEmailException("Email format is invalid: " +
applicant.getEmail());
            }
            PreparedStatement ps = conn.prepareStatement("INSERT INTO
Applicants (FirstName, LastName, Email, Phone, Resume) VALUES
(?, ?, ?, ?, ?)");
            ps.setString(1, applicant.getFirstName());
            ps.setString(2, applicant.getLastName());
            ps.setString(3, applicant.getEmail());
            ps.setString(4, applicant.getPhone());
            ps.setString(5, applicant.getResume());
            ps.executeUpdate();
        } catch (InvalidEmailException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (SQLException e) {
            System.out.println("Database error while inserting applicant.");
            e.printStackTrace();
        }
    }

    public List<Applicant> getAllApplicants() {
        List<Applicant> applicants = new ArrayList<>();
    }

```



```

try {
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery("SELECT * FROM Applicants");
    while (rs.next()) {
        applicants.add(new Applicant(
            rs.getInt("ApplicantID"),
            rs.getString("FirstName"),
            rs.getString("LastName"),
            rs.getString("Email"),
            rs.getString("Phone"),
            rs.getString("Resume")
        ));
    }
} catch (SQLException e) {
    System.out.println("Database error while fetching applicants.");
    e.printStackTrace();
}
return applicants;
}
}

```

Class : CompanydaoImpl.java

```

package dao;

import java.sql.*;
import java.util.*;

import entity.Company;

public class CompanyDAOImpl implements CompanyDAO {
    private Connection conn;

    public CompanyDAOImpl(Connection conn) {
        this.conn = conn;
    }

    public void insertCompany(Company company) {
        try {

```

```

        PreparedStatement ps = conn.prepareStatement("INSERT INTO
Companies (CompanyName, Location) VALUES (?, ?)");
        ps.setString(1, company.getCompanyName());
        ps.setString(2, company.getLocation());
        ps.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public List<Company> getAllCompanies() {
    List<Company> companies = new ArrayList<>();
    try {
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM Companies");
        while (rs.next()) {
            companies.add(new Company(
                rs.getInt("CompanyID"),
                rs.getString("CompanyName"),
                rs.getString("Location")
            ));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return companies;
}
}

```

Class : JobApplicationdaoimpl.java

```

package dao;

import java.sql.*;
import java.util.*;

import entity.JobApplication;

public class JobApplicationDAOImpl implements JobApplicationDAO {
    private Connection conn;

```

```

public JobApplicationDAOImpl(Connection conn) {
    this.conn = conn;
}

public void insertJobApplication(JobApplication application) {
    try {
        PreparedStatement ps = conn.prepareStatement("INSERT INTO
Applications (JobID, ApplicantID, ApplicationDate, CoverLetter) VALUES
(?, ?, ?, ?)");
        ps.setInt(1, application.getJobID());
        ps.setInt(2, application.getApplicantID());
        ps.setTimestamp(3,
Timestamp.valueOf(application.getApplicationDate()));
        ps.setString(4, application.getCoverLetter());
        ps.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public List<JobApplication> getApplicationsByJobID(int jobID) {
    List<JobApplication> apps = new ArrayList<>();
    try {
        PreparedStatement ps = conn.prepareStatement("SELECT * FROM
Applications WHERE JobID = ?");
        ps.setInt(1, jobID);
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            apps.add(new JobApplication(
                rs.getInt("ApplicationID"),
                rs.getInt("JobID"),
                rs.getInt("ApplicantID"),
                rs.getTimestamp("ApplicationDate").toLocalDateTime(),
                rs.getString("CoverLetter")
            ));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```
        return apps;
    }
}
```

Class : JobListingdaoImpl.java

```
package dao;

import java.sql.*;
import java.util.*;

import entity.JobListing;

public class JobListingDAOImpl implements JobListingDAO {
    private Connection conn;

    public JobListingDAOImpl(Connection conn) {
        this.conn = conn;
    }

    public void insertJobListing(JobListing job) {
        try {
            PreparedStatement ps = conn.prepareStatement("INSERT INTO Jobs  
(CompanyID, JobTitle, JobDescription, JobLocation, Salary, JobType,  
PostedDate) VALUES (?, ?, ?, ?, ?, ?, ?)");
            ps.setInt(1, job.getCompanyID());
            ps.setString(2, job.getJobTitle());
            ps.setString(3, job.getJobTitle());
            ps.setString(4, job.getJobTitle());
            ps.setDouble(5, job.getSalary());
            ps.setString(6, job.getJobTitle());
            ps.setTimestamp(7,  
Timestamp.valueOf(java.time.LocalDateTime.now()));
            ps.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

```

public List<JobListing> getAllJobListings() {
    List<JobListing> jobs = new ArrayList<>();
    try {
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM Jobs");
        while (rs.next()) {
            jobs.add(new JobListing(
                rs.getInt("JobID"),
                rs.getInt("CompanyID"),
                rs.getString("JobTitle"),
                rs.getString("JobDescription"),
                rs.getString("JobLocation"),
                rs.getDouble("Salary"),
                rs.getString("JobType"),
                rs.getTimestamp("PostedDate").toLocalDateTime()
            ));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return jobs;
}

```

```

public List<JobListing> getJobListingsBySalaryRange(double minSalary,
double maxSalary) {
    List<JobListing> jobs = new ArrayList<>();
    try {
        PreparedStatement ps = conn.prepareStatement("SELECT * FROM Jobs
WHERE Salary BETWEEN ? AND ?");
        ps.setDouble(1, minSalary);
        ps.setDouble(2, maxSalary);
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            jobs.add(new JobListing(
                rs.getInt("JobID"),
                rs.getInt("CompanyID"),
                rs.getString("JobTitle"),
                rs.getString("JobDescription"),
                rs.getString("JobLocation"),
                rs.getDouble("Salary"),

```

```

        rs.getString("JobType"),
        rs.getTimestamp("PostedDate").toLocalDateTime()
    ));
    }
} catch (SQLException e) {
    e.printStackTrace();
}
return jobs;
}
}

```

Package : Exceptions

Class : DatabaseConnectioException

```

package exception;

public class DatabaseConnectionException extends Exception {
    public DatabaseConnectionException(String message) {
        super(message);
    }
}

```

Class : InvalidEmailException

```

package exception;

public class InvalidEmailException extends Exception {
    public InvalidEmailException(String message) {
        super(message);
    }
}

```

Class : NegativeSalary

```

package exception;

public class NegativeSalaryException extends Exception {

```

```
public NegativeSalaryException(String message) {  
    super(message);  
}  
}
```

Package : util

Class : DBConnutil

```
package util;  
  
import java.sql.*;  
  
public class DBConnUtil {  
    public static Connection getConnection(String fileName) {  
        try {  
            String connString = DBPropertyUtil.getConnectionString(fileName);  
            return DriverManager.getConnection(connString);  
        } catch (SQLException e) {  
            e.printStackTrace();  
            return null;  
        }  
    }  
}
```

Class: DBPropertyutil

```
package util;  
  
import java.io.*;  
import java.util.Properties;  
  
public class DBPropertyUtil {  
    public static String getConnectionString(String fileName) {  
        try (InputStream input = new FileInputStream(fileName)) {  
            Properties prop = new Properties();  
            prop.load(input);  
        }  
    }  
}
```

```

        return prop.getProperty("db.url") + "?user=" +
prop.getProperty("db.user") + "&password=" +
prop.getProperty("db.password");
    } catch (IOException ex) {
        ex.printStackTrace();
        return null;
    }
}
}
}

```

Package : main

Class : MainMeathod

```

package main;

import java.io.IOException;
import java.sql.Connection;
import java.time.LocalDateTime;
import java.util.List;
import java.util.Scanner;

import entity.Applicant;
import entity.Company;
import entity.JobApplication;
import entity.JobListing;
import util.DBConnUtil;

import dao.CompanyDAO;
import dao.JobListingDAO;
import dao.ApplicantDAO;
import dao.JobApplicationDAO;
import dao.CompanyDAOImpl;
import dao.JobListingDAOImpl;
import dao.ApplicantDAOImpl;
import dao.JobApplicationDAOImpl;

public class MainModule {
    public static void main(String[] args) throws IOException {
        Scanner scanner = new Scanner(System.in);
    }
}

```



```
Connection conn = DBConnUtil.getConnection("src/db.properties");

CompanyDAO companyDAO = new CompanyDAOImpl(conn);
JobListingDAO jobListingDAO = new JobListingDAOImpl(conn);
ApplicantDAO applicantDAO = new ApplicantDAOImpl(conn);
JobApplicationDAO jobApplicationDAO = new
JobApplicationDAOImpl(conn);

while (true) {
    System.out.println(" CareerHub Job Board\n");
    System.out.println("1. Register Company ");
    System.out.println("2. Company Job Posting:");
    System.out.println("3. Applicant Profile Creation");
    System.out.println("4. Job Application Submission");
    System.out.println("5. Job Listing Retrieval");
    System.out.println("6. Exit");
    System.out.print("Enter your choice: ");

    int choice = scanner.nextInt();
    scanner.nextLine();

    switch (choice) {
        case 1:
            System.out.print("Company Name: ");
            String cname = scanner.nextLine();
            System.out.print("Location: ");
            String loc = scanner.nextLine();
            Company company = new Company(0, cname, loc);
            companyDAO.insertCompany(company);
            System.out.println("Company Registered.");
            break;
        case 2:
            System.out.print("Company ID: ");
            int cid = scanner.nextInt(); scanner.nextLine();
            System.out.print("Job Title: ");
            String jtitle = scanner.nextLine();
            System.out.print("Job Desc: ");
            String jdesc = scanner.nextLine();
            System.out.print("Job Location: ");
            String jloc = scanner.nextLine();
```

```

        System.out.print("Salary: ");
        double sal = scanner.nextDouble(); scanner.nextLine();
        System.out.print("Job Type: ");
        String jtype = scanner.nextLine();
        JobListing job = new JobListing(0, cid, jtitle, jdesc, jloc, sal, jtype,
LocalDateTime.now());
        jobListingDAO.insertJobListing(job);
        System.out.println("Job Posted.");
        break;
    case 3:
        System.out.print("First Name: ");
        String fname = scanner.nextLine();
        System.out.print("Last Name: ");
        String lname = scanner.nextLine();
        System.out.print("Email: ");
        String email = scanner.nextLine();
        if (!email.contains("@")) {
            System.out.println("Invalid email format.");
            break;
        }
        System.out.print("Phone: ");
        String phone = scanner.nextLine();
        System.out.print("Resume file name: ");
        String resume = scanner.nextLine();
        Applicant app = new Applicant(0, fname, lname, email, phone,
resume);
        applicantDAO.insertApplicant(app);
        System.out.println("Applicant Registered.");
        break;
    case 4:
        System.out.print("Applicant ID: ");
        int aid = scanner.nextInt(); scanner.nextLine();
        System.out.print("Job ID: ");
        int jid = scanner.nextInt(); scanner.nextLine();
        System.out.print("Cover Letter: ");
        String cover = scanner.nextLine();
        JobApplication appn = new JobApplication(0, jid, aid,
LocalDateTime.now(), cover);
        jobApplicationDAO.insertJobApplication(appn);
        System.out.println("Application Submitted.");

```

```

        break;
    case 5:
        List<JobListing> jobs = jobListingDAO.getAllJobListings();
        for (JobListing j : jobs) {
            System.out.println "[" + j.getJobID() + " ] " + j.getJobTitle() + " -
Rs. " + j.getSalary());
        }
        break;
    case 6:
        System.out.println("Thank you for using CareerHub!");
        scanner.close();
        System.exit(0);
    }
}
}
}
}

```

Output:

```

CareerHub Job Board

1. Register Company
2. Company Job Posting:
3. Applicant Profile Creat
4. Job Application Submiss
5. Job Listing Retrieval
6. Exit
Enter your choice:

```

Case 1:

```
1. Register Company
2. Company Job Posting:
3. Applicant Profile Creation
4. Job Application Submission
5. Job Listing Retrieval
6. Exit
Enter your choice: 1
Company Name: Facebook
Location: Indore
Company Registered.
```

Case 2 :

```
CareerHub Job Board

1. Register Company
2. Company Job Posting:
3. Applicant Profile Creation
4. Job Application Submission
5. Job Listing Retrieval
6. Exit
Enter your choice: 2
Company ID: 6
Job Title: SQL Developer
Job Desc: Database
Job Location: Pune
Salary: 40000
Job Type: Full-time
Job Posted.
```

Case 3:

```
1. Register Company
2. Company Job Posting:
3. Applicant Profile Creation
4. Job Application Submission
5. Job Listing Retrieval
6. Exit
Enter your choice: 3
First Name: Om
Last Name: D
Email: om@gmail.com
Phone: 8877665544
Resume file name: omresume.pdf
Applicant Registered.
```

Case 4 :

```
1. Register Company
2. Company Job Posting:
3. Applicant Profile Creation
4. Job Application Submission
5. Job Listing Retrieval
6. Exit
Enter your choice: 4
Applicant ID: 2
Job ID: 9
Cover Letter: cover.cv
Application Submitted.
```

Case 5:

```
1. Register Company
2. Company Job Posting:
3. Applicant Profile Creation
4. Job Application Submission
5. Job Listing Retrieval
6. Exit
Enter your choice: 5
[8] Java Developer - Rs. 800000.0
[9] Data Analyst - Rs. 700000.0
[10] Frontend Engineer - Rs. 750000.0
[12] Cloud Developer - Rs. 60000.0
[13] SQL Developer - Rs. 40000.0
```