Name : Kunal Sanjay Ekare

Batch/Roll : B2-37

Subject: Software lab

Date :01-04-2025

Aim : Design a web application using Servlets and Java server pages to design a menu driven application for Company management system having two types of users **Admin** and **Worker**. Admin can add new worker and view all the tasks done in company. Worker can enter job details and view all the job done. Admin can **review and rate** worker performance based on completed tasks and Workers receive feedback and can request clarifications on tasks.

Theory:

Company Management System using Servlets and JSP – Project Overview

1. Introduction

This project focuses on designing a **menu-driven web application** using **Java Servlets and JSP (Java Server Pages)** for managing company tasks efficiently. The system has two types of users:

- **Admin**: Manages workers, reviews tasks, and provides feedback.
- **Worker**: Submits job details, views completed jobs, and receives feedback.

---

2. Technologies Used

- **Frontend**: HTML, CSS, JSP
- **Backend**: Java Servlets, JDBC (Java Database Connectivity)
- **Database**: MySQL
- **Server**: Apache Tomcat

---

3. Functionalities of the System

Admin Functionalities

1. **Add a new worker** → Admin can register new workers into the system.
2. **View all tasks done in the company** → Admin can see a list of completed jobs.
3. **Review and rate worker performance** → Admin can provide feedback and rate workers based on their completed tasks.

Worker Functionalities

1. **Enter job details** → Workers can submit job descriptions for tracking.
2. **View all jobs done** → Workers can see their submitted jobs along with their status.
3. **Receive feedback** → Workers can read reviews given by the admin and request clarifications.

---

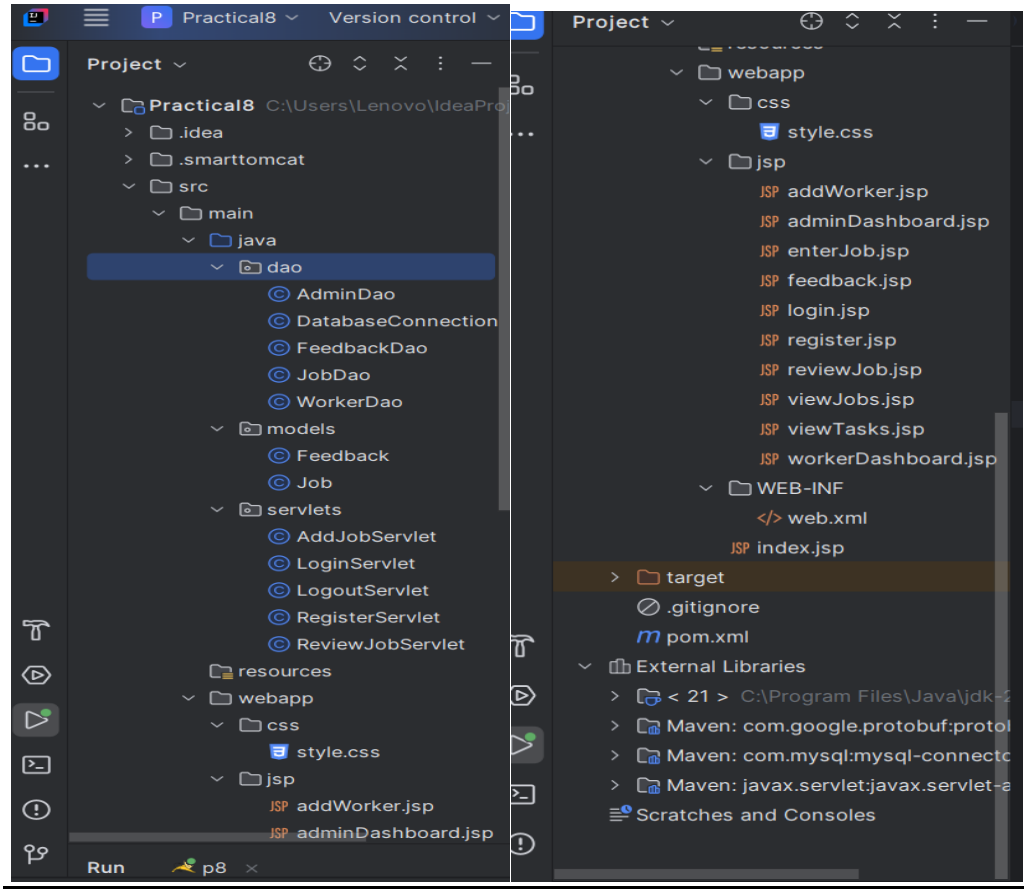
4. System Design & Workflow

Database Structure

The system consists of the following key tables:

1. **`Admin` Table** → Stores admin credentials.
2. **`Worker` Table** → Stores worker details (ID, Name, Email, etc.).
3. **`Job` Table** → Stores job details (ID, Description, Status, Assigned Worker).
4. **`Feedback` Table** → Stores feedback given by the admin for each job.

Project structure:



Code:(AdminDao.java)

```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class AdminDao {
    public boolean authenticateAdmin(String email, String password) {
        String query = "SELECT * FROM admin WHERE email = ? AND password =
?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {

            stmt.setString(1, email);
            stmt.setString(2, password);
            ResultSet rs = stmt.executeQuery();
            return rs.next();

        } catch (Exception e) {
            e.printStackTrace();
        }
        return false;
    }

    public boolean addAdmin(String name, String email, String password) {
        String query = "INSERT INTO admin (name, email, password) VALUES (?,
?, ?)";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {

            stmt.setString(1, name);
            stmt.setString(2, email);
            stmt.setString(3, password);

            int rowsInserted = stmt.executeUpdate();
            return rowsInserted > 0;

        } catch (Exception e) {
            e.printStackTrace();
        }
        return false; //
    }

    public boolean adminExists(String email) {
        String query = "SELECT * FROM admin WHERE email = ?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {

            stmt.setString(1, email);
            ResultSet rs = stmt.executeQuery();
            return rs.next();

        } catch (Exception e) {
            e.printStackTrace();
```

```
        }
        return false;
    }


    public int getAdminIdByEmail(String email) {
        String query = "SELECT id FROM admin WHERE email = ?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {

            stmt.setString(1, email);
            ResultSet rs = stmt.executeQuery();
            if (rs.next()) {
                return rs.getInt("id");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return -1; //
    }
}
```

DatabaseConnection.java

```
package dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private static final String URL =
"jdbc:mysql://localhost:3306/CompanyDB";
    private static final String USER = "root";
    private static final String PASSWORD = "pass123";

    static {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver"); // Load MySQL Driver
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    public static Connection getConnection() {
        try {
            return DriverManager.getConnection(URL, USER, PASSWORD);
        } catch (SQLException e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

FeedbackDao.java

```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
import models.Feedback;

public class FeedbackDao {

    // ☑ Add a feedback entry
    public boolean addFeedback(int jobId, int adminId, String feedbackText,
int rating) {
        String sql = "INSERT INTO Feedback (job_id, admin_id, feedback_text,
rating) VALUES (?, ?, ?, ?)";

        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setInt(1, jobId);
            stmt.setInt(2, adminId);
            stmt.setString(3, feedbackText);
            stmt.setInt(4, rating);

            int rowsAffected = stmt.executeUpdate();
            return rowsAffected > 0;

        } catch (Exception e) {
            e.printStackTrace();
        }
        return false;
    }

    // ☑ Get all feedback for a specific worker
    public List<Feedback> getAllFeedbackForWorker(int workerId) {
        List<Feedback> feedbackList = new ArrayList<>();
        String sql = "SELECT F.feedback_id, F.feedback_text, F.rating, A.name
AS admin_name " +
                "FROM Feedback F " +
                "INNER JOIN Job J ON F.job_id = J.job_id " +
                "INNER JOIN Admin A ON F.admin_id = A.admin_id " +
                "WHERE J.worker_id = ?";

        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setInt(1, workerId);
            ResultSet rs = stmt.executeQuery();

            while (rs.next()) {
                Feedback feedback = new Feedback(
                        rs.getInt("feedback_id"),
                        rs.getString("feedback_text"),
```

```
                        rs.getInt("rating"),
                        rs.getString("admin_name")
                );
                feedbackList.add(feedback);
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return feedbackList;
    }
}
```

JobDao.java

```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
import models.Job;

public class JobDao {


    public boolean addJob(int workerId, String jobDescription) {
        String sql = "INSERT INTO Job (worker_id, job_description, status)
VALUES (?, ?, 'Pending')";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setInt(1, workerId);
            stmt.setString(2, jobDescription);

            System.out.println(" DEBUG: Executing SQL: " + stmt.toString());

            int rowsAffected = stmt.executeUpdate();
            if (rowsAffected > 0) {
                System.out.println(" DEBUG: Job inserted successfully for
Worker ID: " + workerId);
                return true;
            } else {
                System.out.println(" ERROR: Job insertion failed for Worker
ID: " + workerId);
            }

        } catch (Exception e) {
            System.out.println(" ERROR in addJob(): " + e.getMessage());
        }
        return false;
    }
```

```
    public int getWorkerIdByEmail(String workerEmail) {
        String sql = "SELECT worker_id FROM Worker WHERE email = ?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setString(1, workerEmail);
            ResultSet rs = stmt.executeQuery();

            if (rs.next()) {
                int workerId = rs.getInt("worker_id");
                System.out.println(" DEBUG: Worker ID found: " + workerId + "
for Email: " + workerEmail);
                return workerId;
            } else {
                System.out.println(" ERROR: No Worker found for Email: " +
workerEmail);
            }

        } catch (Exception e) {
            System.out.println(" ERROR in getWorkerIdByEmail(): " +
e.getMessage());
        }
        return -1;  // Worker not found
    }


    public List<Job> getAllJobs() {
        List<Job> jobs = new ArrayList<>();
        String sql = "SELECT J.job_id, J.job_description, J.status,
W.worker_id, W.name AS worker_name, W.email AS worker_email " +
                "FROM Job J INNER JOIN Worker W ON J.worker_id =
W.worker_id";

        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql);
             ResultSet rs = stmt.executeQuery()) {

            while (rs.next()) {
                Job job = new Job(
                        rs.getInt("job_id"),
                        rs.getInt("worker_id"),
                        rs.getString("worker_name"),
                        rs.getString("worker_email"),
                        rs.getString("job_description"),
                        rs.getString("status")
                );
                jobs.add(job);
            }
            System.out.println("DEBUG: Retrieved " + jobs.size() + " jobs
from database.");

        } catch (Exception e) {
            System.out.println(" ERROR in getAllJobs(): " + e.getMessage());
        }
        return jobs;
    }
```

```
    public boolean updateJobStatus(int jobId, String newStatus) {
        String sql = "UPDATE Job SET status = ? WHERE job_id = ?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setString(1, newStatus);
            stmt.setInt(2, jobId);

            int rowsUpdated = stmt.executeUpdate();
            if (rowsUpdated > 0) {
                System.out.println(" DEBUG: Job ID " + jobId + " status
updated to " + newStatus);
                return true;
            } else {
                System.out.println(" ERROR: Failed to update job status for
Job ID " + jobId);
            }

        } catch (Exception e) {
            System.out.println(" ERROR in updateJobStatus(): " +
e.getMessage());
        }
        return false;
    }

    // ☑ NEW: Get Jobs by Worker Email (To allow workers to see jobs they
have done)
    public List<Job> getJobsByWorkerEmail(String workerEmail) {
        List<Job> jobs = new ArrayList<>();
        String sql = "SELECT job_id, job_description, status FROM Job WHERE
worker_id = (SELECT worker_id FROM Worker WHERE email = ?)";

        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setString(1, workerEmail);
            ResultSet rs = stmt.executeQuery();

            while (rs.next()) {
                jobs.add(new Job(
                        rs.getInt("job_id"),
                        -1,  // No need for worker ID here
                        null, // No worker name needed
                        workerEmail,
                        rs.getString("job_description"),
                        rs.getString("status")
                ));
            }
            System.out.println(" DEBUG: Retrieved " + jobs.size() + " jobs
for Worker: " + workerEmail);

        } catch (Exception e) {
            System.out.println(" ERROR in getJobsByWorkerEmail(): " +
e.getMessage());
        }
```

```
        return jobs;
    }
}
```

WorkerDao.java

```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

public class WorkerDao {


    public boolean workerExists(String email) {
        String sql = "SELECT 1 FROM Worker WHERE email = ?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setString(1, email);
            ResultSet rs = stmt.executeQuery();
            return rs.next(); // Returns true if a record is found

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }


    public boolean addWorker(String name, String email, String password) {
        if (workerExists(email)) {
            return false; // Prevent adding duplicate workers
        }

        String sql = "INSERT INTO Worker (name, email, password) VALUES (?,
?, ?)";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setString(1, name);
            stmt.setString(2, email);
            stmt.setString(3, password);
            return stmt.executeUpdate() > 0;

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
```

```java
public boolean authenticateWorker(String email, String password) {
    String sql = "SELECT * FROM Worker WHERE email = ? AND password = ?";
    try (Connection conn = DatabaseConnection.getConnection();
         PreparedStatement stmt = conn.prepareStatement(sql)) {

        stmt.setString(1, email);
        stmt.setString(2, password);
        ResultSet rs = stmt.executeQuery();
        return rs.next();

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return false;
}


public int getWorkerIdByEmail(String email) {
    String sql = "SELECT worker_id FROM Worker WHERE email = ?";
    try (Connection conn = DatabaseConnection.getConnection();
         PreparedStatement stmt = conn.prepareStatement(sql)) {

        stmt.setString(1, email);
        ResultSet rs = stmt.executeQuery();

        if (rs.next()) {
            return rs.getInt("worker_id");
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return -1;  // Worker not found
}


public List<String> getAllJobsForWorker(int workerId) {
    List<String> jobs = new ArrayList<>();
    String sql = "SELECT job_description FROM Job WHERE worker_id = ?";
    try (Connection conn = DatabaseConnection.getConnection();
         PreparedStatement stmt = conn.prepareStatement(sql)) {

        stmt.setInt(1, workerId);
        ResultSet rs = stmt.executeQuery();

        while (rs.next()) {
            jobs.add(rs.getString("job_description"));
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return jobs;
}
```

```
    public List<String> getAllJobs() {
        List<String> jobs = new ArrayList<>();
        String sql = "SELECT W.name, J.job_description FROM Job J INNER JOIN
Worker W ON J.worker_id = W.worker_id";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql);
             ResultSet rs = stmt.executeQuery()) {

            while (rs.next()) {
                String jobEntry = "Worker: " + rs.getString("name") + " |
Job: " + rs.getString("job_description");
                jobs.add(jobEntry);
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return jobs;
    }

        public boolean requestClarification(int workerId, String
clarificationText) {
        String sql = "INSERT INTO ClarificationRequests (worker_id,
request_text) VALUES (?, ?)";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setInt(1, workerId);
            stmt.setString(2, clarificationText);
            return stmt.executeUpdate() > 0;

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }


    public List<String> getWorkerFeedback(int workerId) {
        List<String> feedbackList = new ArrayList<>();
        String sql = "SELECT feedback, rating FROM Feedback WHERE worker_id =
?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setInt(1, workerId);
            ResultSet rs = stmt.executeQuery();

            while (rs.next()) {
                String feedbackEntry = "Rating: " + rs.getInt("rating") + " |
Feedback: " + rs.getString("feedback");
                feedbackList.add(feedbackEntry);
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }
```

```
        return feedbackList;
    }
}
```

feedback.java

```
package models;

public class Feedback {
    private int feedbackId;
    private String feedbackText;
    private int rating;
    private String adminName; // To show which admin gave the feedback

    public Feedback(int feedbackId, String feedbackText, int rating, String
adminName) {
        this.feedbackId = feedbackId;
        this.feedbackText = feedbackText;
        this.rating = rating;
        this.adminName = adminName;
    }

    public int getFeedbackId() {
        return feedbackId;
    }

    public String getFeedbackText() {
        return feedbackText;
    }

    public int getRating() {
        return rating;
    }

    public String getAdminName() {
        return adminName;
    }
}
```

Job.java

```
package models;

public class Job {
    private int jobId;
    private int workerId;
    private String workerName;
    private String workerEmail;
    private String jobDescription;
    private String status;


    public Job(int jobId, int workerId, String workerName, String
```

```
workerEmail, String jobDescription, String status) {
        this.jobId = jobId;
        this.workerId = workerId;
        this.workerName = workerName;
        this.workerEmail = workerEmail;
        this.jobDescription = jobDescription;
        this.status = status;
    }


    public Job() {}

    public int getJobId() { return jobId; }
    public int getWorkerId() { return workerId; }
    public String getWorkerName() { return workerName; }
    public String getWorkerEmail() { return workerEmail; }
    public String getJobDescription() { return jobDescription; }
    public String getStatus() { return status; }


    public void setStatus(String status) {
        this.status = status;
    }
}
```

AddJobServlet.java

```
package servlets;

import dao.JobDao;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
@WebServlet("/AddJobServlet")
public class AddJobServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        String workerEmail = (String) session.getAttribute("email");

        if (workerEmail == null) {
            System.out.println(" ERROR: Worker Email is NULL! Redirecting to
login.");
            response.sendRedirect("jsp/login.jsp");
            return;
        }


        Integer workerIdObj = (Integer) session.getAttribute("workerId");
        int workerId = (workerIdObj != null) ? workerIdObj : 0;
```

```
        System.out.println(" DEBUG: Worker ID from session: " + workerId);

        if (workerId == 0) {

            JobDao jobDao = new JobDao();
            workerId = jobDao.getWorkerIdByEmail(workerEmail);
            if (workerId != -1) {
                session.setAttribute("workerId", workerId); // Store in
session
                System.out.println(" DEBUG: Worker ID found from DB: " +
workerId);
            } else {
                System.out.println(" ERROR: Worker ID not found for Email: "
+ workerEmail);

response.sendRedirect("jsp/enterJob.jsp?error=workerNotFound");
                return;
            }
        }

        String jobDescription = request.getParameter("jobDescription");
        if (jobDescription == null || jobDescription.trim().isEmpty()) {
            System.out.println(" ERROR: Job Description is empty!");
            response.sendRedirect("jsp/enterJob.jsp?error=emptyJob");
            return;
        }

        JobDao jobDao = new JobDao();
        boolean jobAdded = jobDao.addJob(workerId, jobDescription);

        if (jobAdded) {
            System.out.println(" DEBUG: Job successfully added for Worker ID:
" + workerId);

response.sendRedirect("jsp/workerDashboard.jsp?success=jobAdded");
        } else {
            System.out.println(" ERROR: Job insertion failed for Worker ID: "
+ workerId);
            response.sendRedirect("jsp/enterJob.jsp?error=failed");
        }
    }
}
```

LoginServlet.java

```
package servlets;

import dao.WorkerDao;
import dao.AdminDao;
import java.io.IOException;
import java.util.regex.Pattern;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {

        String email = request.getParameter("email");
        String password = request.getParameter("password");
        String userType = request.getParameter("userType");

        System.out.println(" ◇  Login Attempt: Email=" + email + ", UserType="
+ userType);

        if (userType == null || email == null || password == null ||
                email.trim().isEmpty() || password.trim().isEmpty()) {
            response.sendRedirect("jsp/login.jsp?error=missingFields");
            return;
        }

        String emailRegex = "^[A-Za-z0-9+_.-]+@[A-Za-z0-9.-]+$";
        if (!Pattern.matches(emailRegex, email)) {
            response.sendRedirect("jsp/login.jsp?error=invalidEmailFormat");
            return;
        }

        if (password.length() < 6) {
            response.sendRedirect("jsp/login.jsp?error=shortPassword");
            return;
        }

        HttpSession session = request.getSession(false);

        if (session != null) {
            session.invalidate();
        }

        session = request.getSession(true);

        if ("admin".equalsIgnoreCase(userType)) {
            AdminDao adminDao = new AdminDao();
            int adminId = adminDao.getAdminIdByEmail(email);

            if (adminId > 0 && adminDao.authenticateAdmin(email, password)) {
                session.setAttribute("adminId", adminId);
                session.setAttribute("adminEmail", email);
                session.setAttribute("role", "admin");
                System.out.println(" Admin login successful: " + email);
                response.sendRedirect("jsp/adminDashboard.jsp");
            } else {
                System.out.println(" Admin login failed: " + email);
                response.sendRedirect("jsp/login.jsp?error=invalidAdmin");
            }
        } else if ("worker".equalsIgnoreCase(userType)) {
            WorkerDao workerDao = new WorkerDao();
```

```
            if (workerDao.authenticateWorker(email, password)) {
                int workerId = workerDao.getWorkerIdByEmail(email);
                if (workerId > 0) {
                    session.setAttribute("workerEmail", email);
                    session.setAttribute("workerId", workerId);
                    session.setAttribute("role", "worker");
                    System.out.println(" Worker login successful: " + email);
                    response.sendRedirect("jsp/workerDashboard.jsp");
                } else {
                    System.out.println(" Worker not found: " + email);

response.sendRedirect("jsp/login.jsp?error=workerNotFound");
                }
            } else {
                System.out.println(" Invalid Worker credentials: " + email);
                response.sendRedirect("jsp/login.jsp?error=invalidWorker");
            }
        } else {
            System.out.println(" Invalid user type: " + userType);
            response.sendRedirect("jsp/login.jsp?error=invalidUserType");
        }
    }
}
```

LogoutServelet.java

```
package servlets;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/LogoutServlet")
public class LogoutServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        HttpSession session = request.getSession(false); // Prevents creating
a new session if none exists
        if (session != null) {
            session.invalidate();  // Destroy session
        }
        response.sendRedirect("jsp/login.jsp?message=loggedOut");
    }
}
```

RegisterServlet.java

```
package servlets;
```

```java
import dao.WorkerDao;
import dao.AdminDao;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/RegisterServlet")
public class RegisterServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String name = request.getParameter("name");
        String email = request.getParameter("email");
        String password = request.getParameter("password");
        String role = request.getParameter("role");

        WorkerDao workerDao = new WorkerDao();
        AdminDao adminDao = new AdminDao();


        if (workerDao.workerExists(email) || adminDao.adminExists(email)) {
            response.sendRedirect("jsp/register.jsp?error=userExists");
            return;
        }

        boolean isRegistered = false;


        if ("worker".equals(role)) {
            isRegistered = workerDao.addWorker(name, email, password);
        } else if ("admin".equals(role)) {
            isRegistered = adminDao.addAdmin(name, email, password);
        }

        if (isRegistered) {
            response.sendRedirect("jsp/login.jsp?success=registered");
        } else {
            response.sendRedirect("jsp/register.jsp?error=failed");
        }
    }
}
```

ReviewJobServlet.java

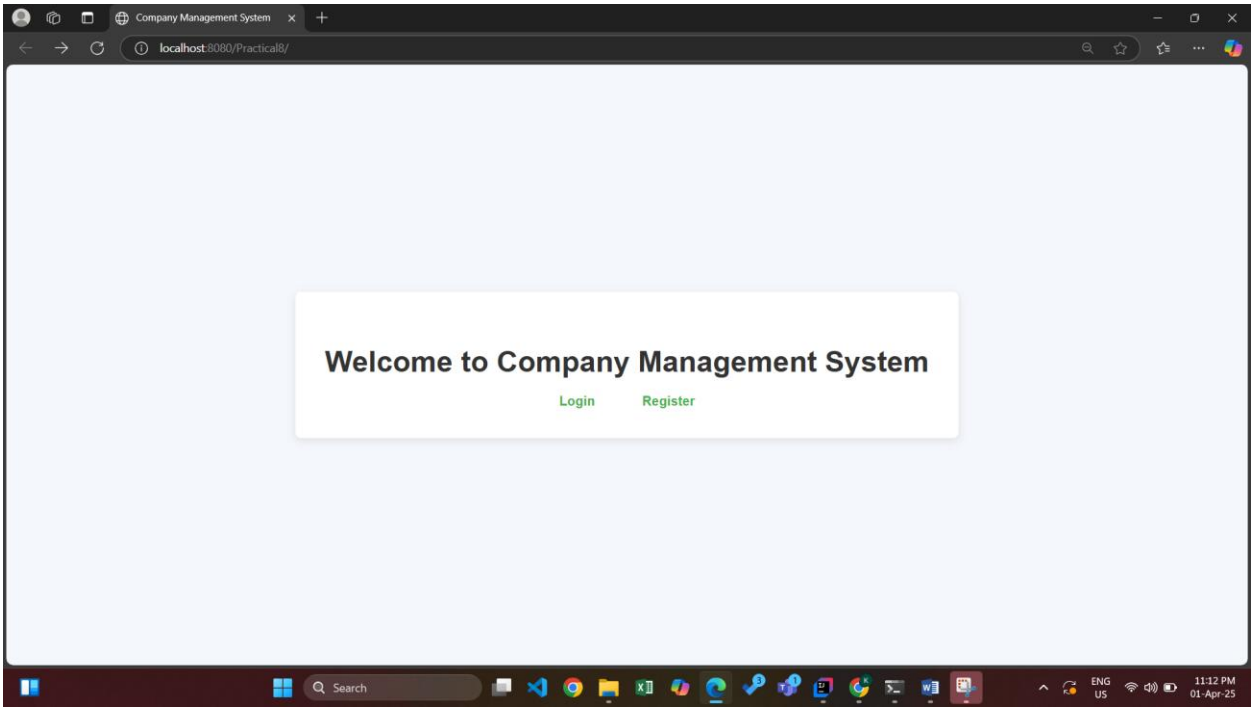
```java
package servlets;

import dao.FeedbackDao;
import dao.JobDao;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

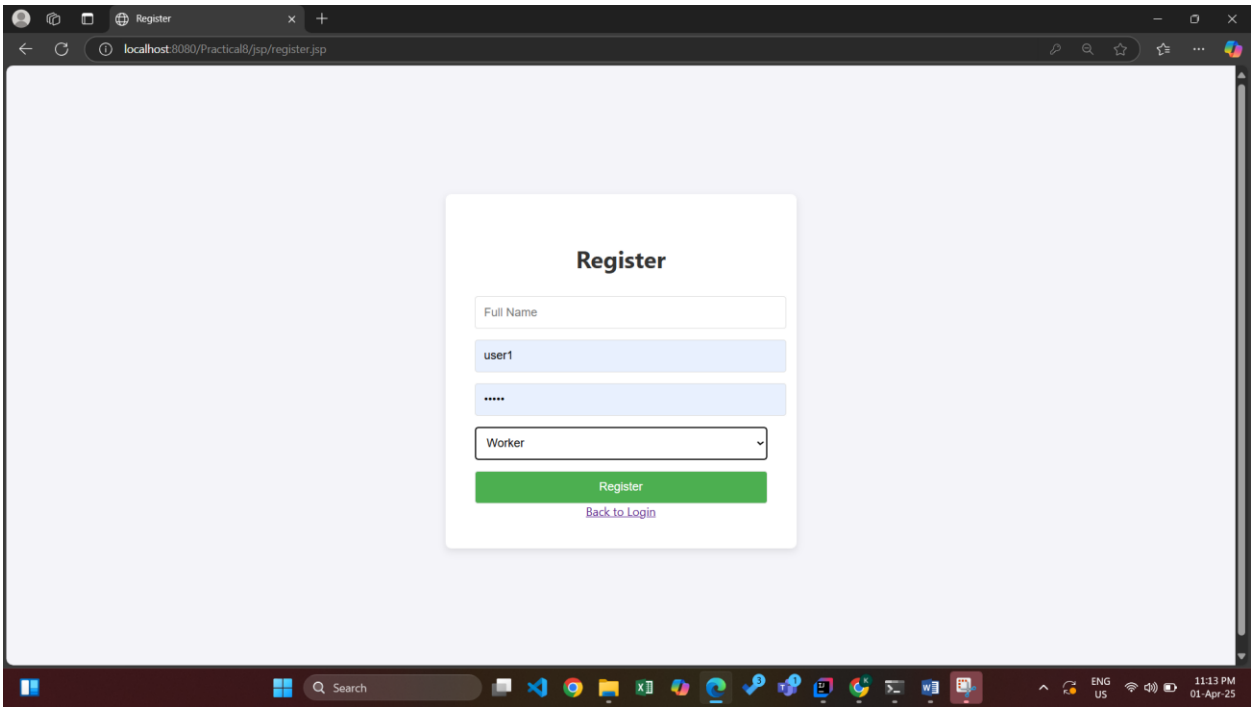
```
import java.io.IOException;

@WebServlet("/ReviewJobServlet")
public class ReviewJobServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String jobIdStr = request.getParameter("jobId");
        String adminIdStr = request.getParameter("adminId");  // Admin ID
from session or form
        String feedbackText = request.getParameter("feedback");
        String ratingStr = request.getParameter("rating");
        String status = request.getParameter("status");


        if (jobIdStr == null || jobIdStr.trim().isEmpty()) {
            System.out.println(" ERROR: jobId is missing or empty!");
            response.sendRedirect("jsp/reviewJob.jsp?error=invalidJobId");
            return;
        }


        if (adminIdStr == null || adminIdStr.trim().isEmpty()) {
            System.out.println(" ERROR: adminId is missing!");
            response.sendRedirect("jsp/reviewJob.jsp?error=missingAdminId");
            return;
        }

        int jobId, adminId, rating;
        try {
            jobId = Integer.parseInt(jobIdStr);
            adminId = Integer.parseInt(adminIdStr);
        } catch (NumberFormatException e) {
            System.out.println(" ERROR: Invalid jobId or adminId format!");

response.sendRedirect("jsp/reviewJob.jsp?error=invalidJobIdOrAdminId");
            return;
        }


        if (ratingStr == null || ratingStr.trim().isEmpty()) {
            rating = 0;  // Default rating
        } else {
            try {
                rating = Integer.parseInt(ratingStr);
            } catch (NumberFormatException e) {
                System.out.println(" ERROR: Invalid rating format!");

response.sendRedirect("jsp/reviewJob.jsp?error=invalidRating");
                return;
            }
        }


        JobDao jobDao = new JobDao();
        boolean jobUpdated = jobDao.updateJobStatus(jobId, status);
```

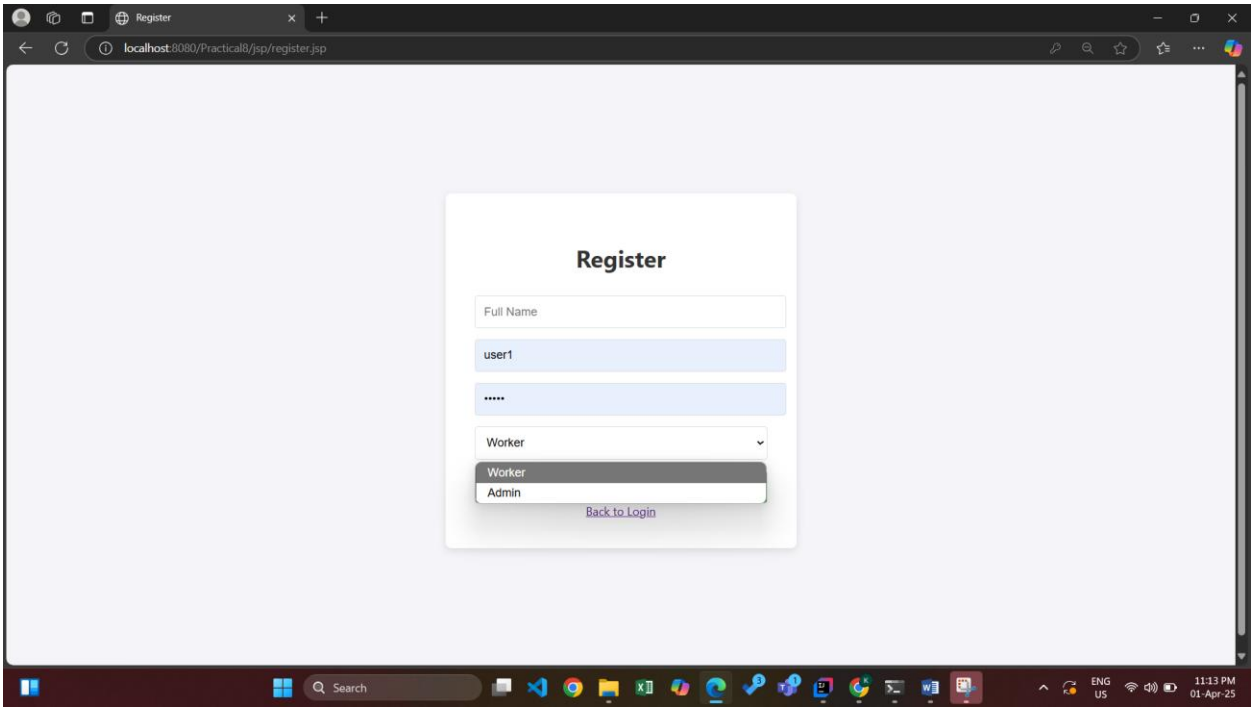
```
        boolean feedbackSuccess = true;
        if (feedbackText != null && !feedbackText.trim().isEmpty()) {
            FeedbackDao feedbackDao = new FeedbackDao();
            feedbackSuccess = feedbackDao.addFeedback(jobId, adminId,
feedbackText, rating);
        }

        if (jobUpdated && feedbackSuccess) {
            System.out.println(" Job ID " + jobId + " updated with status: "
+ status);

response.sendRedirect("jsp/adminDashboard.jsp?success=jobReviewed");
        } else {
            System.out.println(" ERROR: Failed to update job or add
feedback!");
            response.sendRedirect("jsp/reviewJob.jsp?error=updateFailed");
        }
    }
}
```

(Note :all jsp.file and rest of file in RAR folder)

Output:

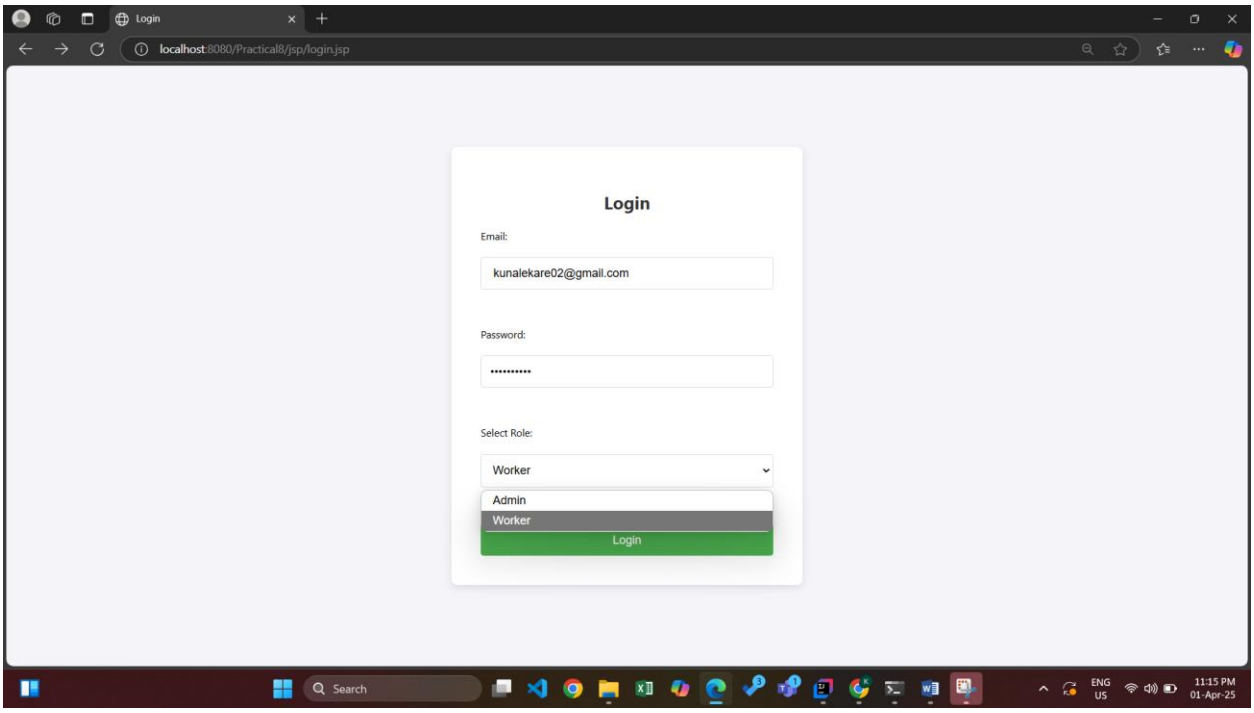


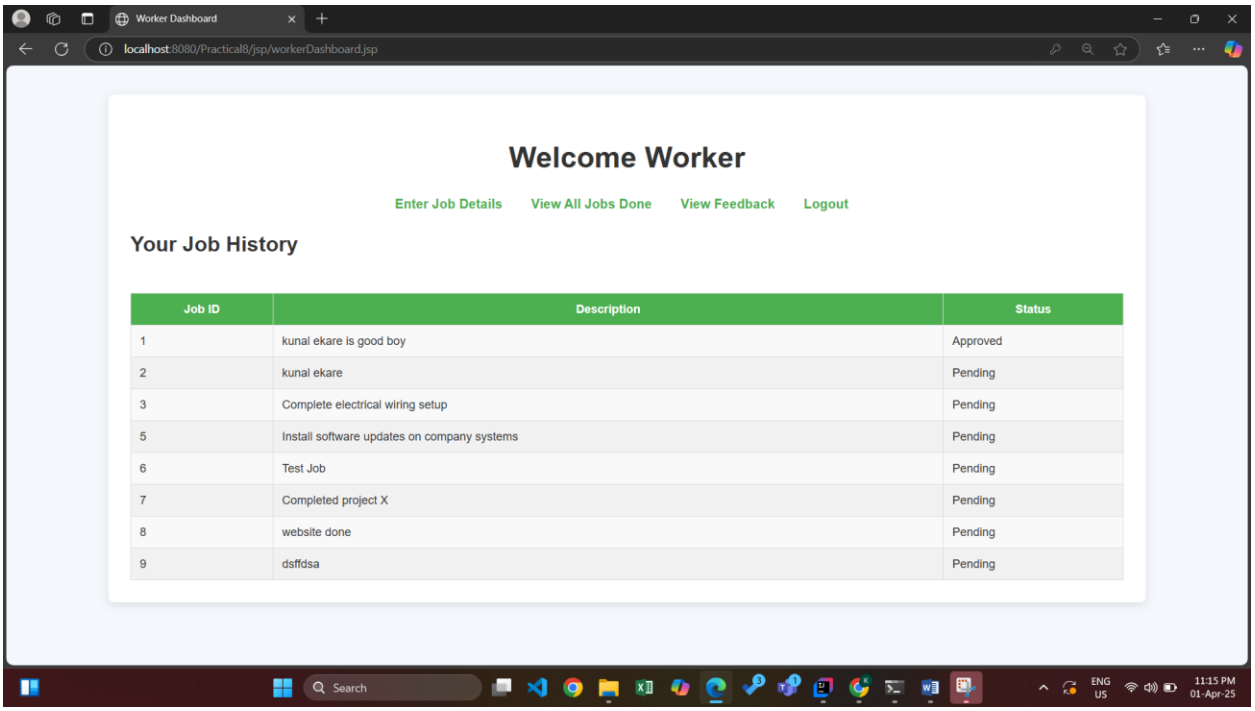


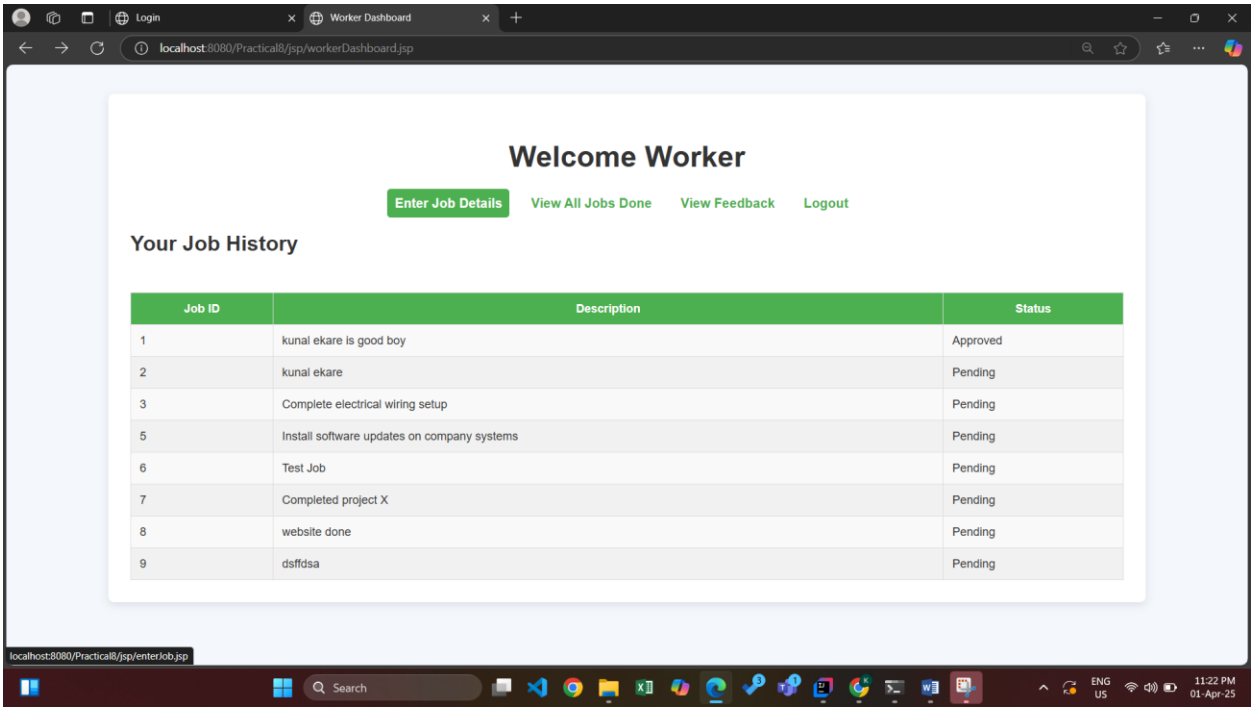


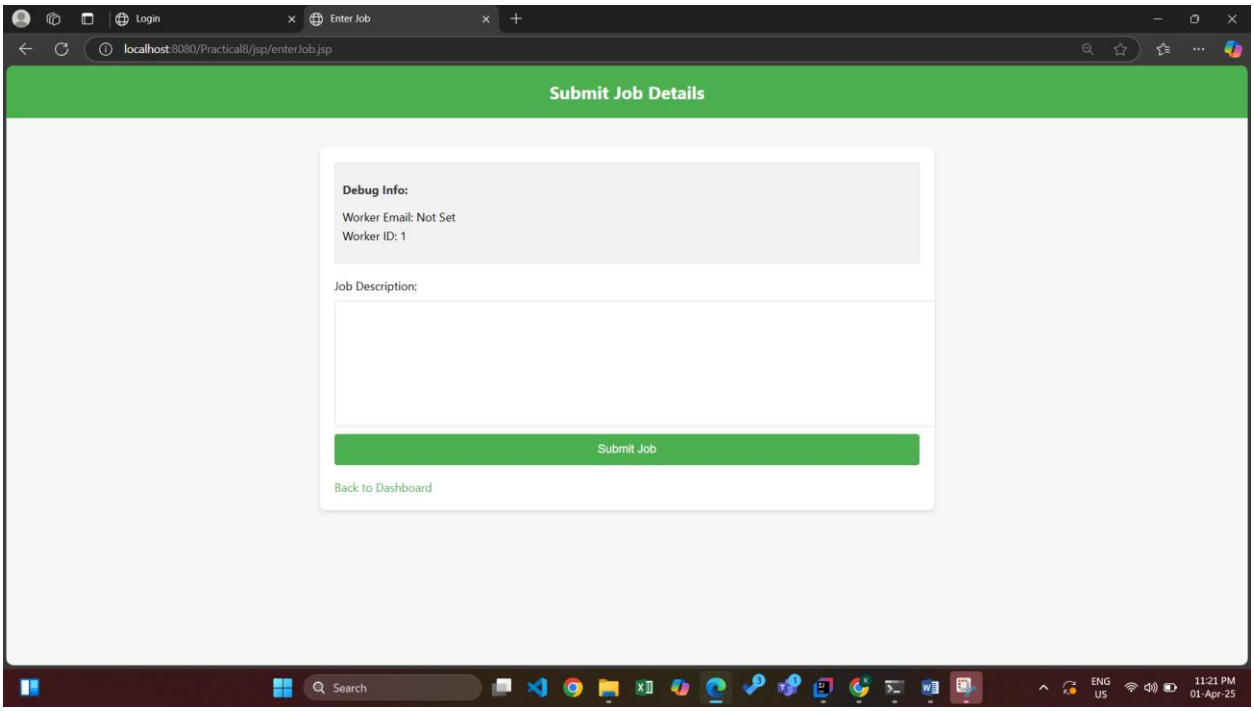
Database changes..

```
mysql> select * from admin;
+----------+-------------+-----------------------+-----------+
| admin_id | name        | email                 | password  |
+----------+-------------+-----------------------+-----------+
|        1 | NULL        | admin@example.com     | admin123  |
|        2 | Kunal Ekare | kunalekare02@gmail.com | pass123   |
|        3 | Rahul Sharma | kunalekare08@gmail.com | Pass@1234 |
+----------+-------------+-----------------------+-----------+
3 rows in set (0.00 sec)

mysql> select * from worker;
+-----------+-----------------------------+-----------------------+-------------+--------+
| worker_id | name                        | email                 | password    | rating |
+-----------+-----------------------------+-----------------------+-------------+--------+
|         1 | Kunal sanjay ekare          | kunalekare02@gmail.com | Pass@1234p  |      0 |
|         2 | Kunal sanjay ekare          | kunalekare03@gmail.com | PAss@1234p  |      0 |
|         3 | Kunal sanjay ekare          | kunalekare05@gmial.com | Pass@34     |      0 |
|         4 | Kunal sanjay ekare          | ekareks@rknec.edu     | Pass@1234   |      0 |
|         5 | Kunal sanjay ekare rcoemjhb | kunalekare04@gmial.com | Pass@345    |      0 |
|         6 | Kunal sanjay ekare          | Kunal03@gmail.com     | Pass@1234   |      0 |
|         7 | John Doe                    | john@example.com      | password123 |      0 |
|         8 | mahendra bahuali            | wanimj@rknec.edu      | Pass@123    |      0 |
+-----------+-----------------------------+-----------------------+-------------+--------+
8 rows in set (0.01 sec)

mysql>
```

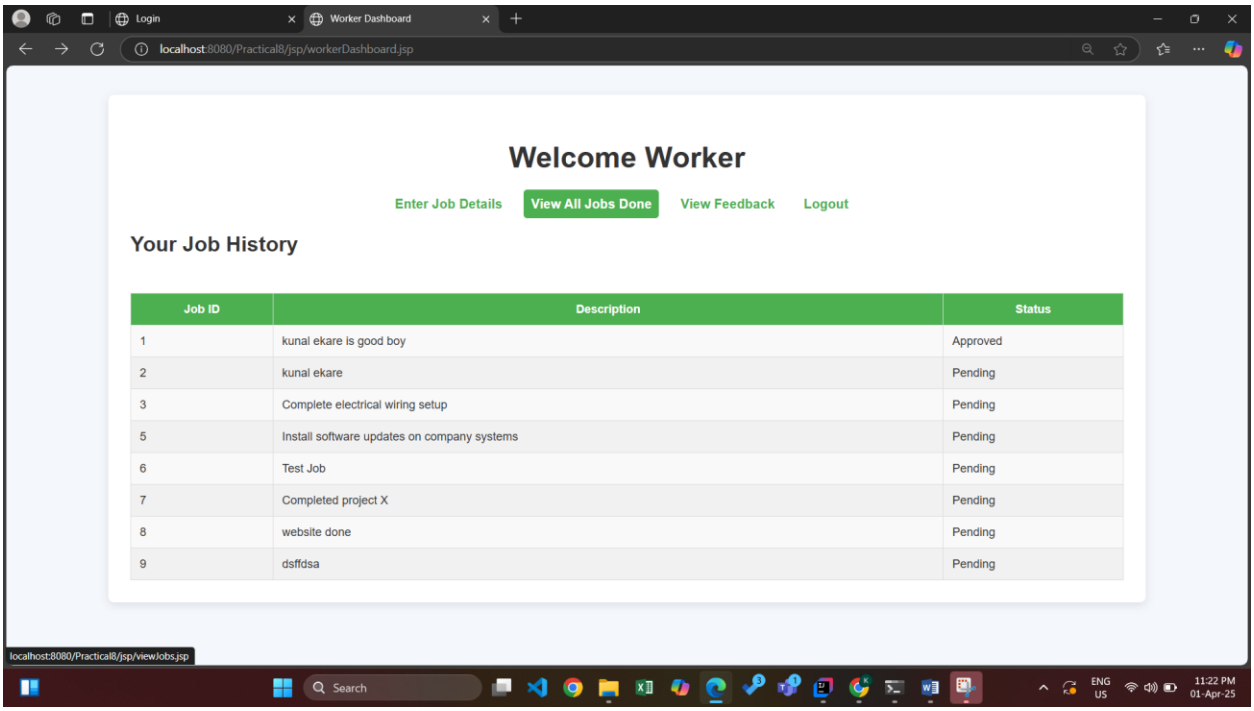


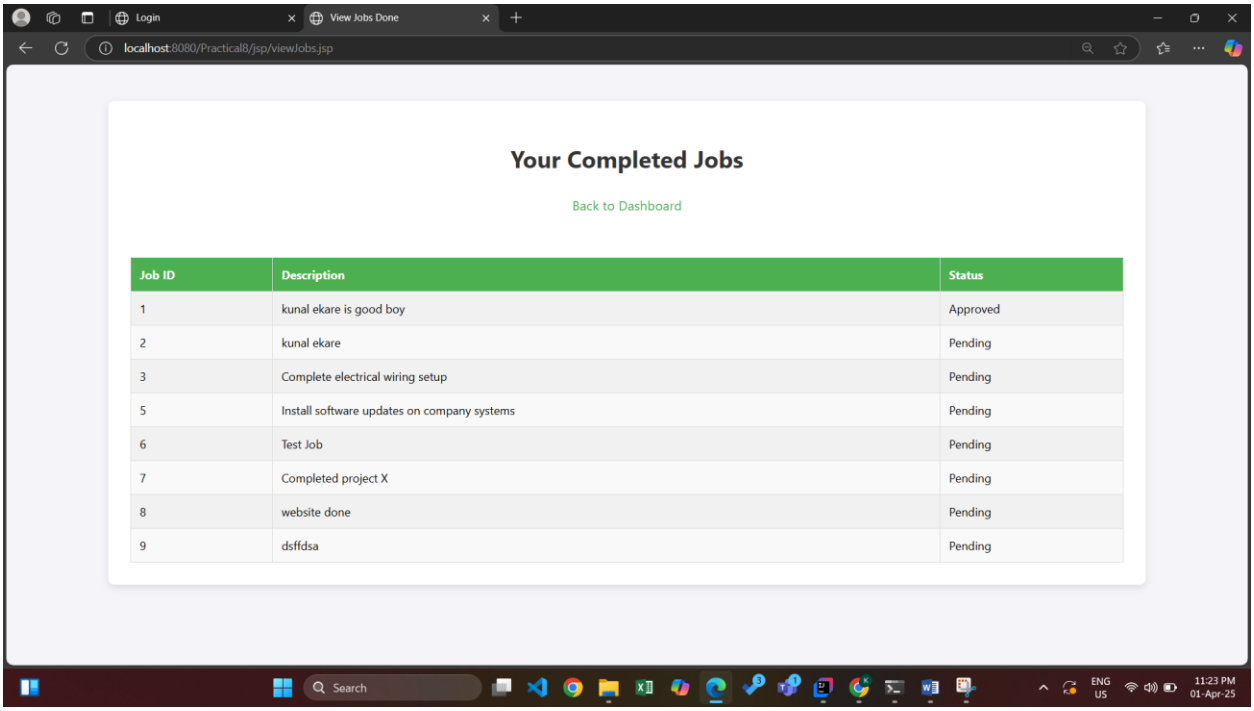

Worker Dashboard
localhost:8080/Practical8/jsp/workerDashboard.jsp
Welcome Worker
Enter Job Details
View All Jobs Done
View Feedback
Logout
Your Job History
Job ID
Description
Status
1
kunal ekare is good boy
Approved
2
kunal ekare
Pending
3
Complete electrical wiring setup
Pending
5
Install software updates on company systems
Pending
6
Test Job
Pending
7
Completed project X
Pending
8
website done
Pending
9
dsffdsa
Pending
Search
ENG
US
11:15 PM
01-Apr-25

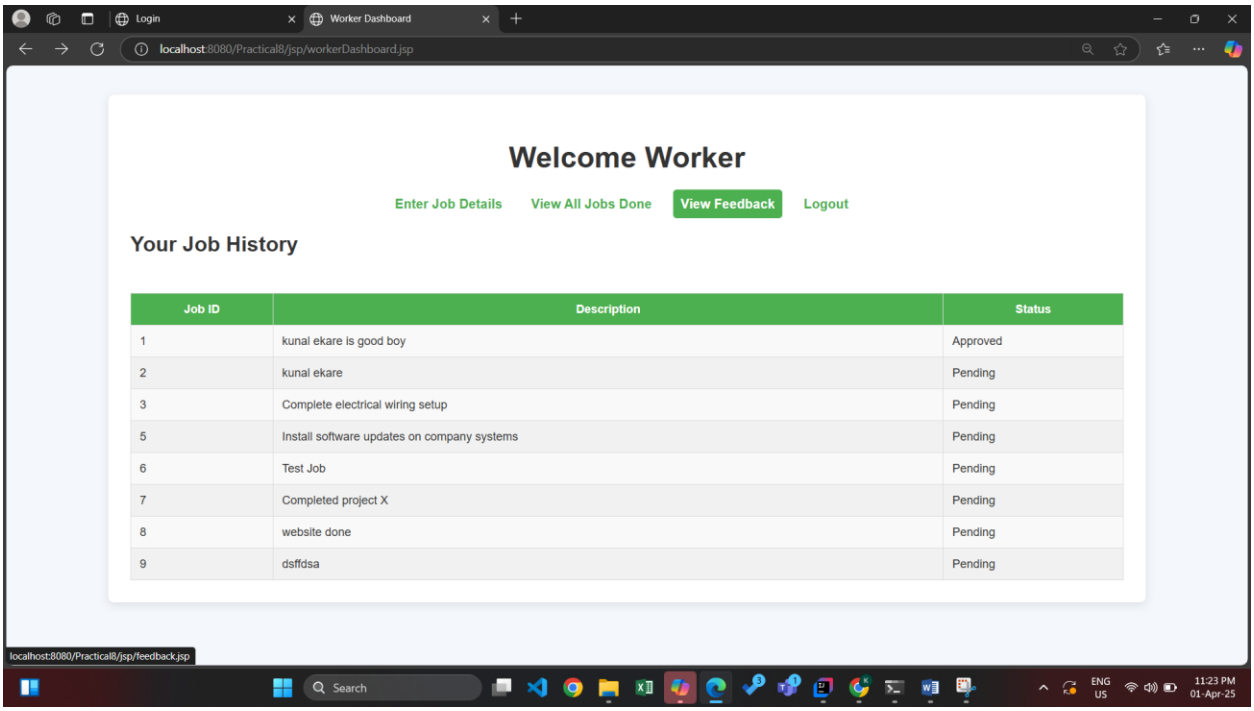

Login
Worker Dashboard
localhost:8080/Practical8/jsp/workerDashboard.jsp
Welcome Worker
Enter Job Details
View All Jobs Done
View Feedback
Logout
Your Job History
Job ID
Description
Status
1
kunal ekare is good boy
Approved
2
kunal ekare
Pending
3
Complete electrical wiring setup
Pending
5
Install software updates on company systems
Pending
6
Test Job
Pending
7
Completed project X
Pending
8
website done
Pending
9
dsffdsa
Pending
localhost:8080/Practical8/jsp/enterJob.jsp
Search
ENG
US
11:22 PM
01-Apr-25

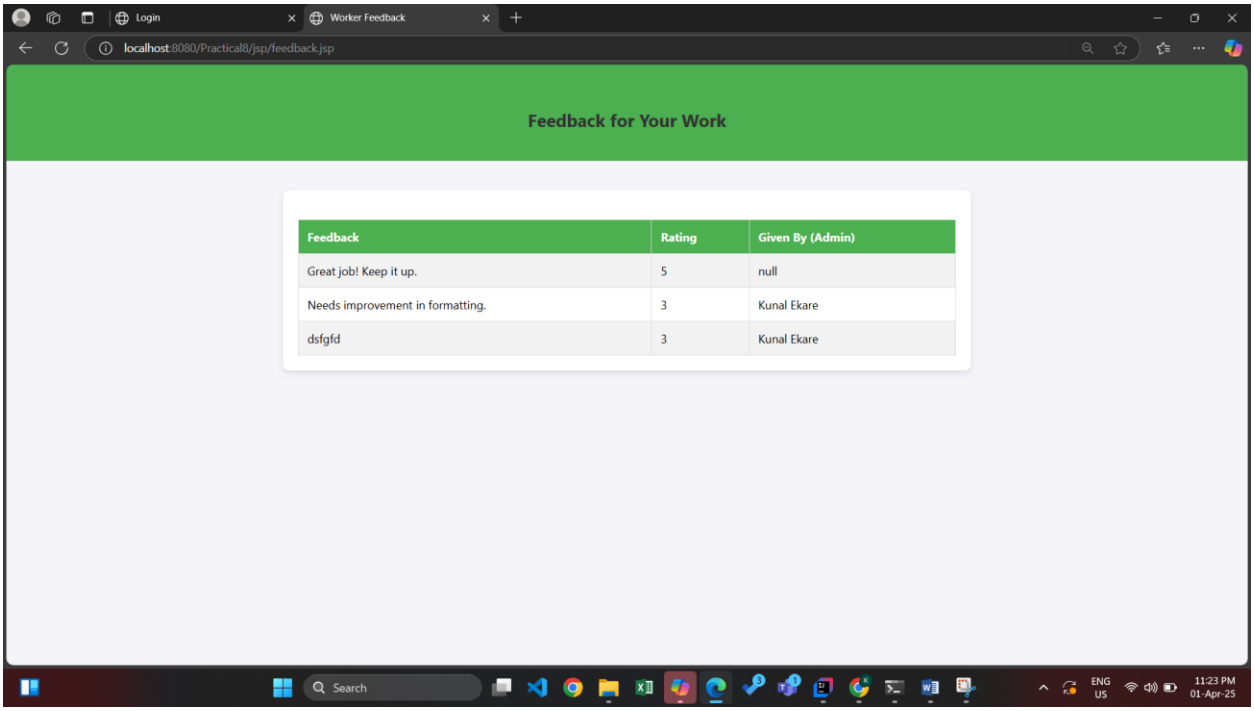


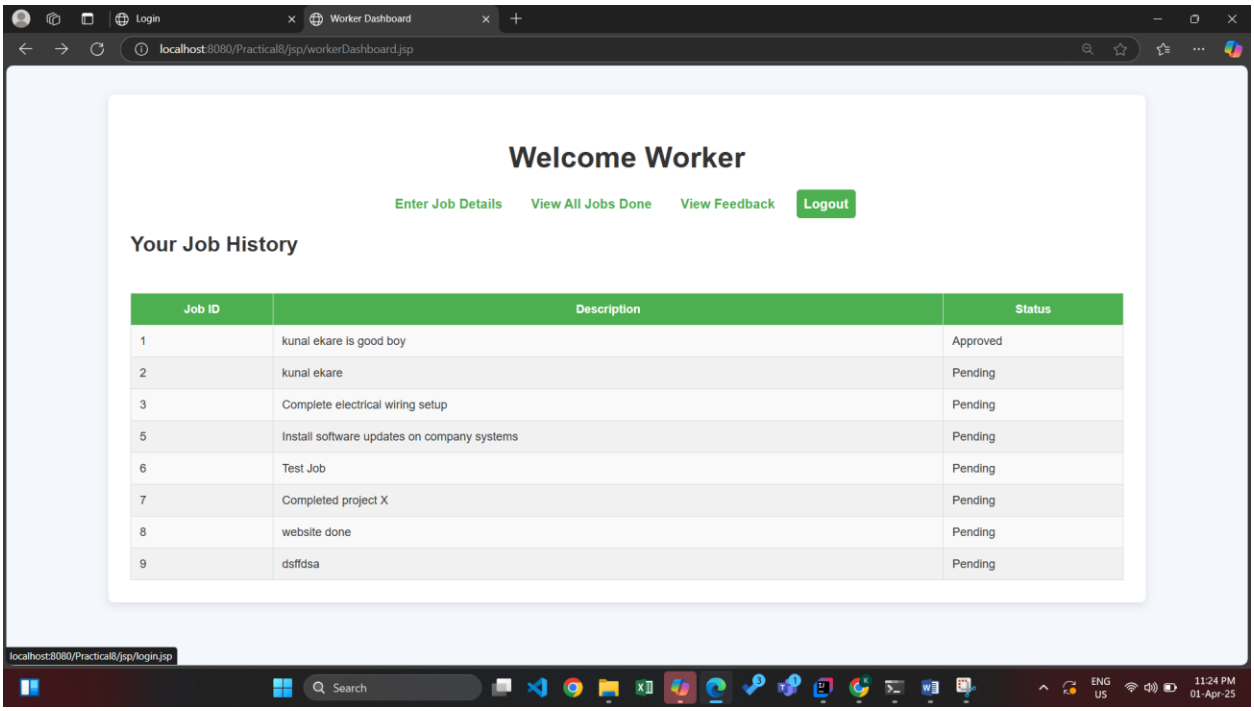

```
mysql> select * from Job;
+--------+-----------+-----------------------------------------+----------+
| job_id | worker_id | job_description                         | status   |
+--------+-----------+-----------------------------------------+----------+
|      1 |         1 | kunal ekare is good boy                 | Approved |
|      2 |         1 | kunal ekare                             | Pending  |
|      3 |         1 | Complete electrical wiring setup        | Pending  |
|      4 |         2 | Fix plumbing issues in the new building | Pending  |
|      5 |         1 | Install software updates on company systems | Pending  |
|      6 |         1 | Test Job                                | Pending  |
|      7 |         1 | Completed project X                     | Pending  |
|      8 |         1 | website done                            | Pending  |
|      9 |         1 | dsffdsa                                 | Pending  |
+--------+-----------+-----------------------------------------+----------+
9 rows in set (0.00 sec)

mysql>
```

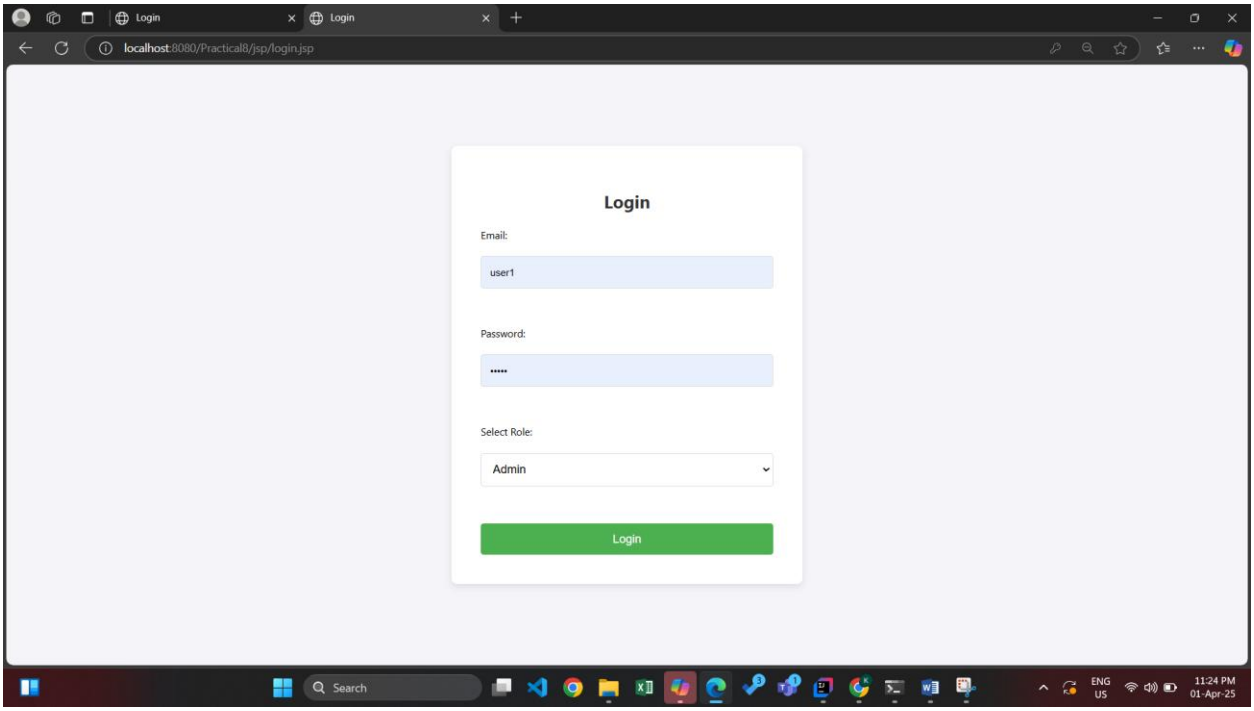

Login
Worker Dashboard
localhost:8080/Practical8/jsp/workerDashboard.jsp
Welcome Worker
Enter Job Details
View All Jobs Done
View Feedback
Logout
Your Job History
Job ID
Description
Status
1
kunal ekare is good boy
Approved
2
kunal ekare
Pending
3
Complete electrical wiring setup
Pending
5
Install software updates on company systems
Pending
6
Test Job
Pending
7
Completed project X
Pending
8
website done
Pending
9
dsffdsa
Pending
localhost:8080/Practical8/jsp/viewJobs.jsp
Search
ENG
US
11:22 PM
01-Apr-25

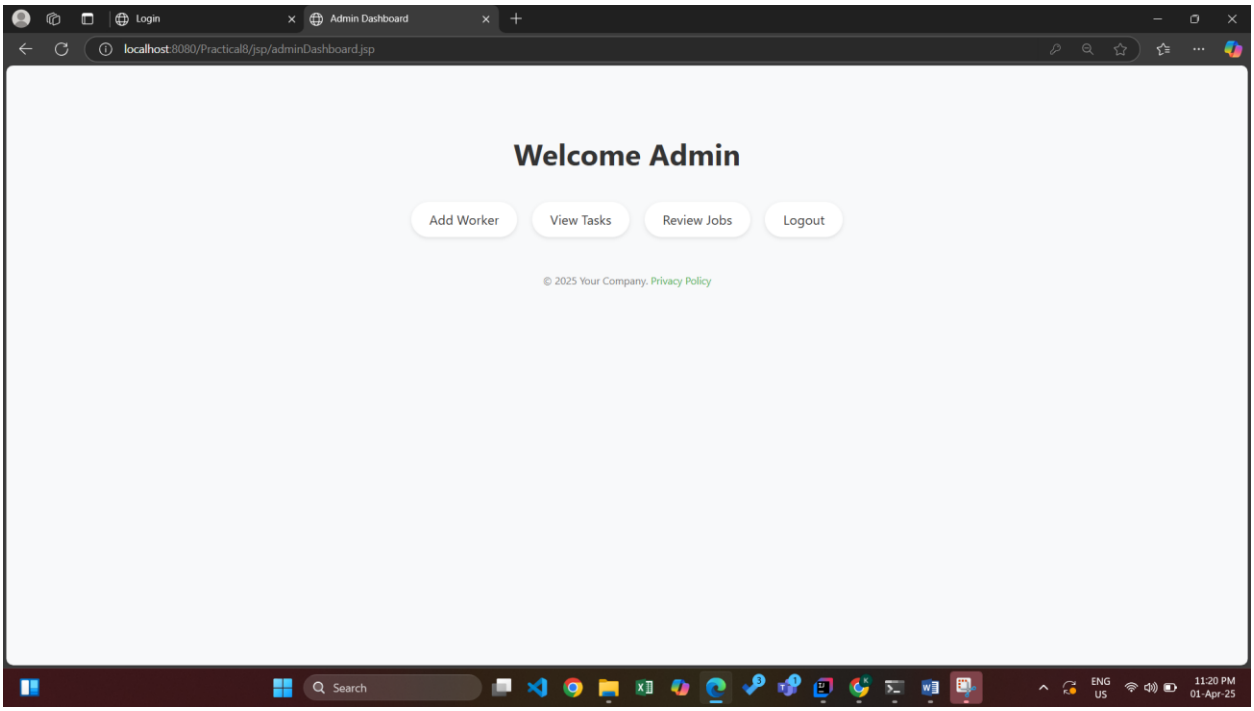

Login
View Jobs Done
localhost:8080/Practical8/jsp/viewJobs.jsp
Your Completed Jobs
Back to Dashboard
Job ID
Description
Status
1
kunal ekare is good boy
Approved
2
kunal ekare
Pending
3
Complete electrical wiring setup
Pending
5
Install software updates on company systems
Pending
6
Test Job
Pending
7
Completed project X
Pending
8
website done
Pending
9
dsffdsa
Pending
Search
ENG
US
11:23 PM
01-Apr-25

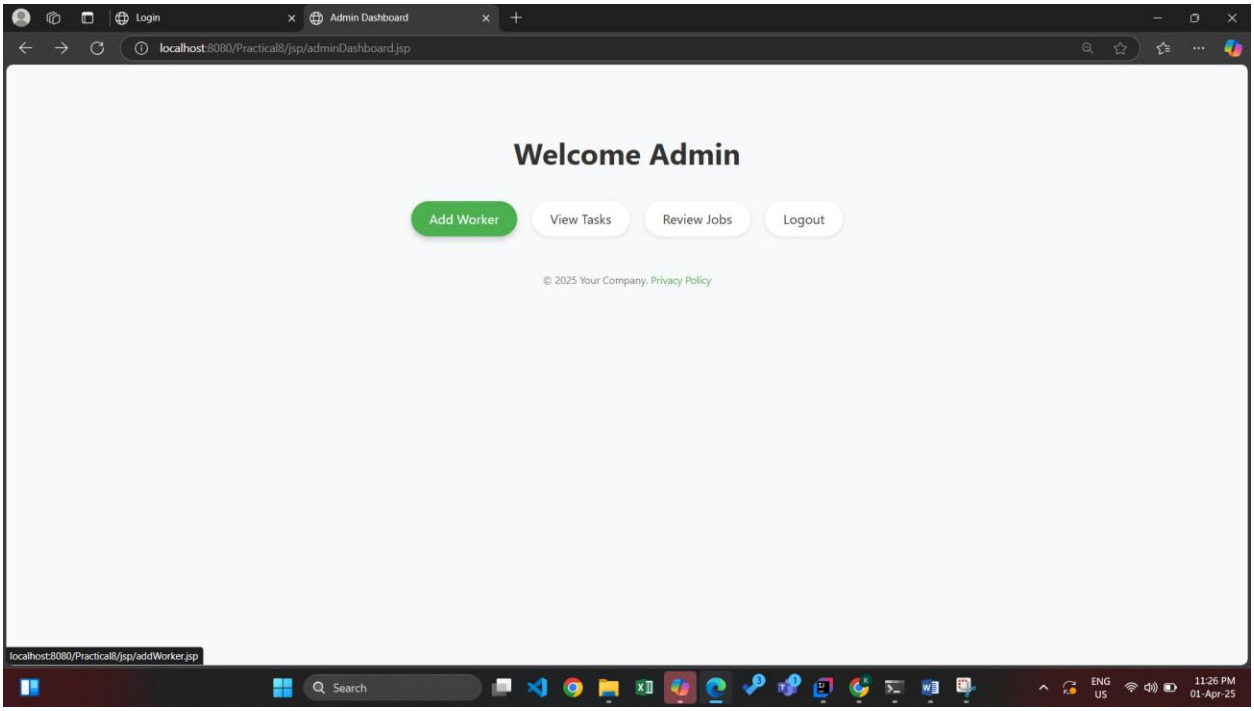

Login
Worker Dashboard
localhost:8080/Practical8/jsp/workerDashboard.jsp
Welcome Worker
Enter Job Details
View All Jobs Done
View Feedback
Logout
Your Job History
Job ID
Description
Status
1
kunal ekare is good boy
Approved
2
kunal ekare
Pending
3
Complete electrical wiring setup
Pending
5
Install software updates on company systems
Pending
6
Test Job
Pending
7
Completed project X
Pending
8
website done
Pending
9
dsffdsa
Pending
localhost:8080/Practical8/jsp/feedback.jsp
Search
ENG
US
11:23 PM
01-Apr-25

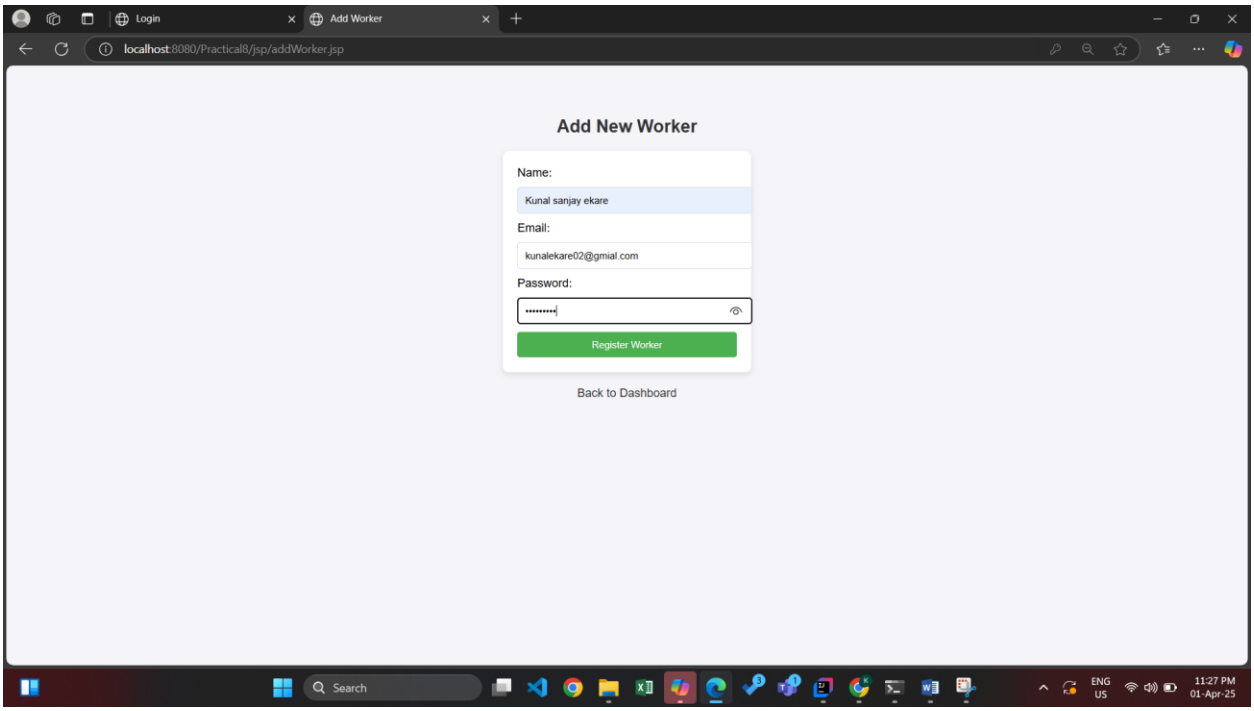

Login
Worker Feedback
localhost:8080/Practical8/jsp/feedback.jsp
Feedback for Your Work
Feedback
Rating
Given By (Admin)
Great job! Keep it up.
5
null
Needs improvement in formatting.
3
Kunal Ekare
dsfgfd
3
Kunal Ekare
Search
ENG
US
11:23 PM
01-Apr-25


Login
Worker Dashboard
localhost:8080/Practical8/jsp/workerDashboard.jsp
Welcome Worker
Enter Job Details
View All Jobs Done
View Feedback
Logout
Your Job History
Job ID
Description
Status
1
kunal ekare is good boy
Approved
2
kunal ekare
Pending
3
Complete electrical wiring setup
Pending
5
Install software updates on company systems
Pending
6
Test Job
Pending
7
Completed project X
Pending
8
website done
Pending
9
dsffdsa
Pending
localhost:8080/Practical8/jsp/login.jsp
Search
ENG
US
11:24 PM
01-Apr-25

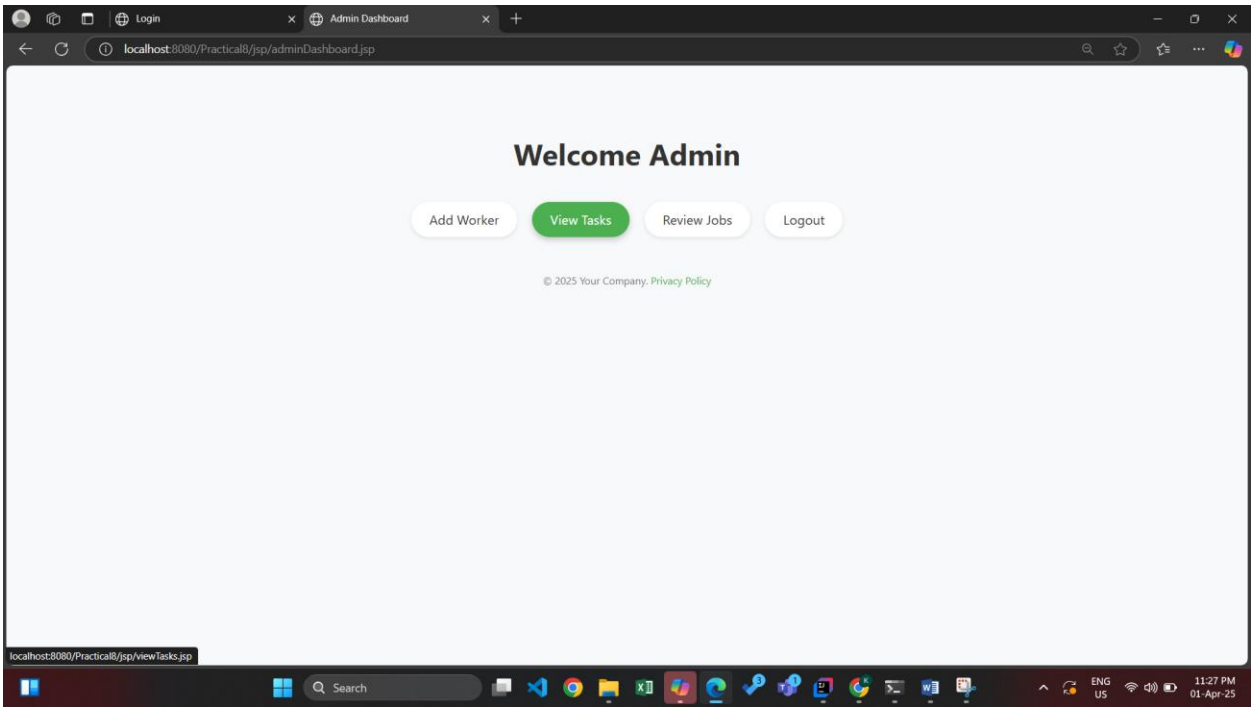

Login
Login
localhost:8080/Practical8/jsp/login.jsp
Login
Email:
user1
Password:
Select Role:
Admin
Login
Search
ENG
US
11:24 PM
01-Apr-25

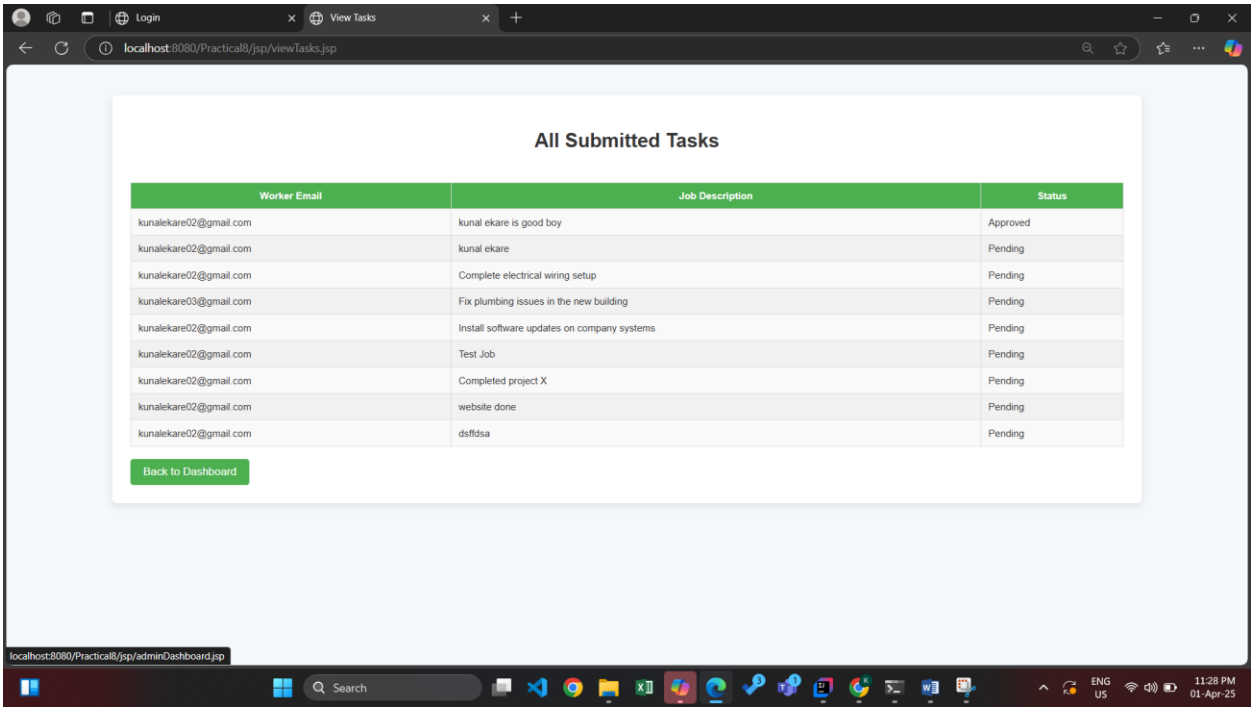

Login
Admin Dashboard
localhost:8080/Practical8/jsp/adminDashboard.jsp
Welcome Admin
Add Worker
View Tasks
Review Jobs
Logout
© 2025 Your Company. Privacy Policy
Search
ENG
US
11:20 PM
01-Apr-25

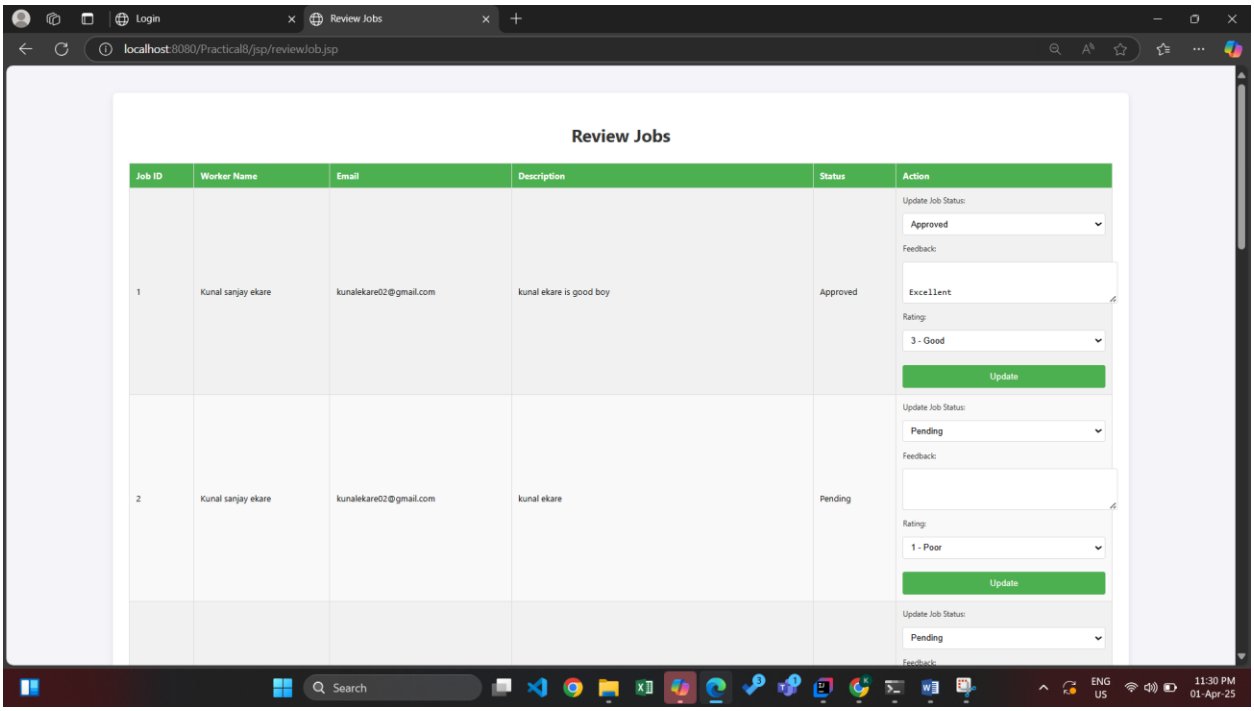

Login
Admin Dashboard
localhost:8080/Practical8/jsp/adminDashboard.jsp
Welcome Admin
Add Worker
View Tasks
Review Jobs
Logout
© 2025 Your Company. Privacy Policy
localhost:8080/Practical8/jsp/addWorker.jsp
Search
ENG
US
11:26 PM
01-Apr-25

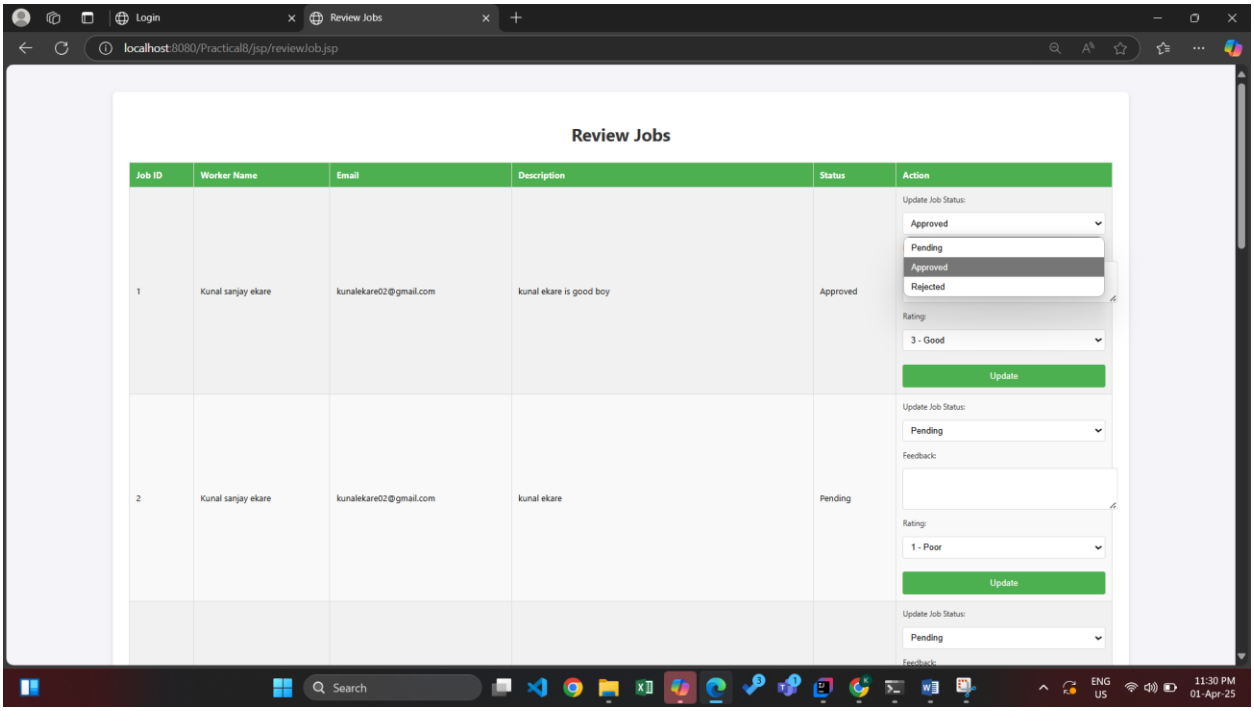


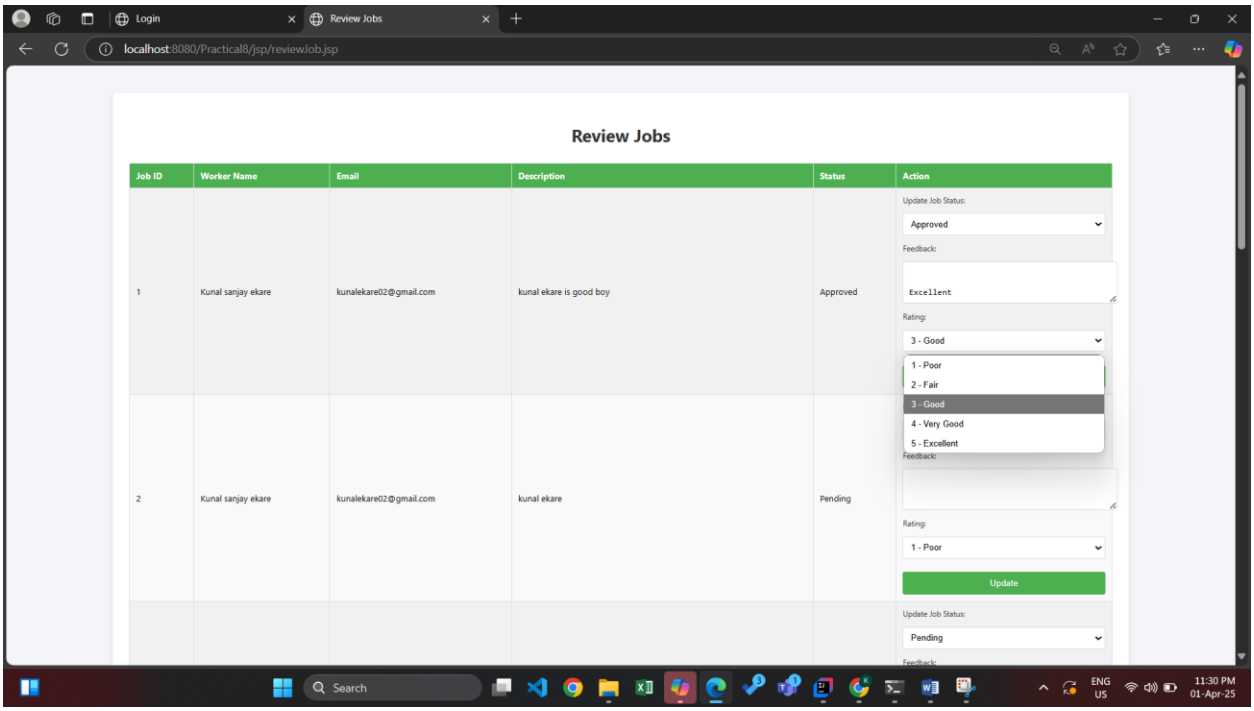

```
mysql> select * from worker;
+-----------+---------------------------+-----------------------+-------------+--------+
| worker_id | name                      | email                 | password    | rating |
+-----------+---------------------------+-----------------------+-------------+--------+
|         1 | Kunal sanjay ekare        | kunalekare02@gmail.com | Pass@1234p  |      0 |
|         2 | Kunal sanjay ekare        | kunalekare03@gmail.com | PAss@1234p  |      0 |
|         3 | Kunal sanjay ekare        | kunalekare05@gmial.com | Pass@34     |      0 |
|         4 | Kunal sanjay ekare        | ekareks@rknec.edu      | Pass@1234   |      0 |
|         5 | Kunal sanjay ekare rcoemjhb | kunalekare04@gmial.com | Pass@345    |      0 |
|         6 | Kunal sanjay ekare        | Kunal03@gmail.com      | Pass@1234   |      0 |
|         7 | John Doe                  | john@example.com       | password123 |      0 |
|         8 | mahendra bahuali          | wanimj@rknec.edu       | Pass@123    |      0 |
+-----------+---------------------------+-----------------------+-------------+--------+
8 rows in set (0.01 sec)
```

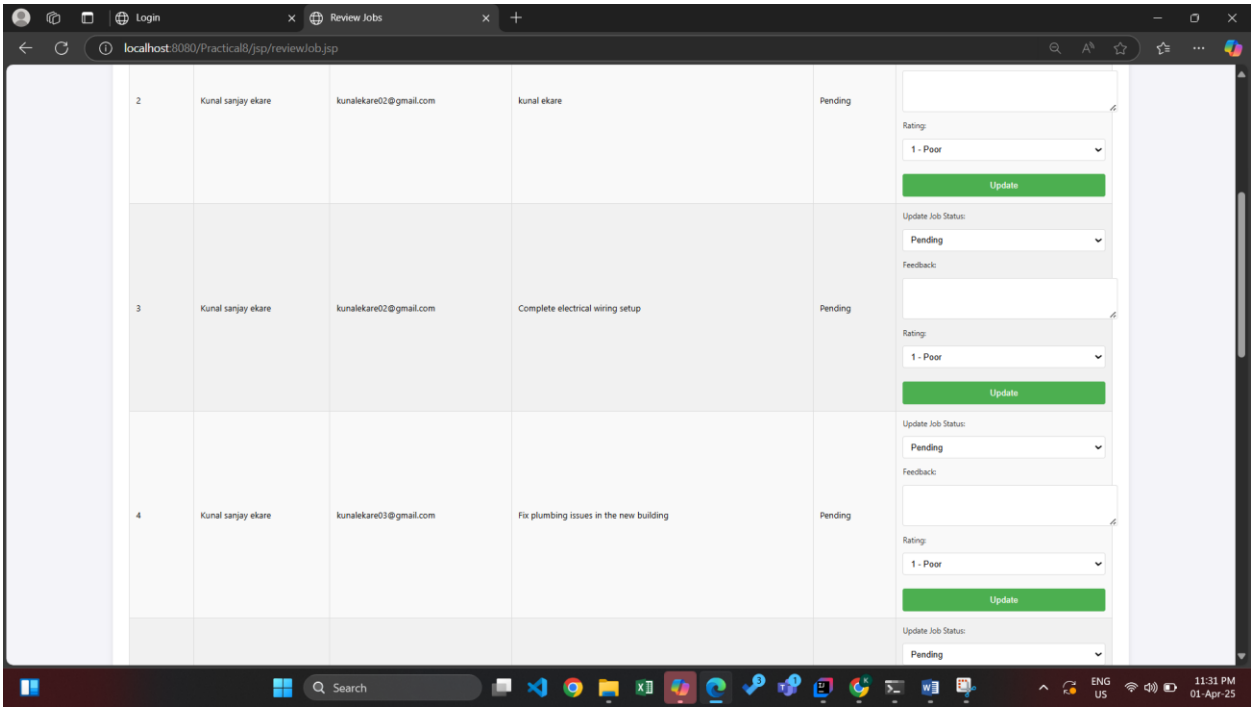

Login
Admin Dashboard
localhost:8080/Practical8/jsp/adminDashboard.jsp
Welcome Admin
Add Worker
View Tasks
Review Jobs
Logout
© 2025 Your Company. Privacy Policy
localhost:8080/Practical8/jsp/viewTasks.jsp
Search
ENG
US
11:27 PM
01-Apr-25


Login
View Tasks
localhost:8080/Practical8/jsp/viewTasks.jsp
All Submitted Tasks
Worker Email
Job Description
Status
kunalekare02@gmail.com
kunal ekare is good boy
Approved
kunalekare02@gmail.com
kunal ekare
Pending
kunalekare02@gmail.com
Complete electrical wiring setup
Pending
kunalekare03@gmail.com
Fix plumbing issues in the new building
Pending
kunalekare02@gmail.com
Install software updates on company systems
Pending
kunalekare02@gmail.com
Test Job
Pending
kunalekare02@gmail.com
Completed project X
Pending
kunalekare02@gmail.com
website done
Pending
kunalekare02@gmail.com
dsffdsa
Pending
Back to Dashboard
localhost:8080/Practical8/jsp/adminDashboard.jsp
Search
ENG
US
11:28 PM
01-Apr-25


Login
Review Jobs
localhost:8080/Practical8/jsp/reviewJob.jsp
Review Jobs
Job ID
Worker Name
Email
Description
Status
Action
1
Kunal sanjay ekare
kunalekare02@gmail.com
kunal ekare is good boy
Approved
Update Job Status:
Approved
Feedback:
Excellent
Rating:
3 - Good
Update
2
Kunal sanjay ekare
kunalekare02@gmail.com
kunal ekare
Pending
Update Job Status:
Pending
Feedback:
Rating:
1 - Poor
Update
Update Job Status:
Pending
Feedback:
Search
ENG
US
11:30 PM
01-Apr-25


Login
Review Jobs
localhost:8080/Practical8/jsp/reviewJob.jsp
Review Jobs
Job ID
Worker Name
Email
Description
Status
Action
1
Kunal sanjay ekare
kunalekare02@gmail.com
kunal ekare is good boy
Approved
Update Job Status:
Approved
Pending
Approved
Rejected
Rating:
3 - Good
Update
2
Kunal sanjay ekare
kunalekare02@gmail.com
kunal ekare
Pending
Update Job Status:
Pending
Feedback:
Rating:
1 - Poor
Update
Update Job Status:
Pending
Feedback:
Search
ENG
US
11:30 PM
01-Apr-25

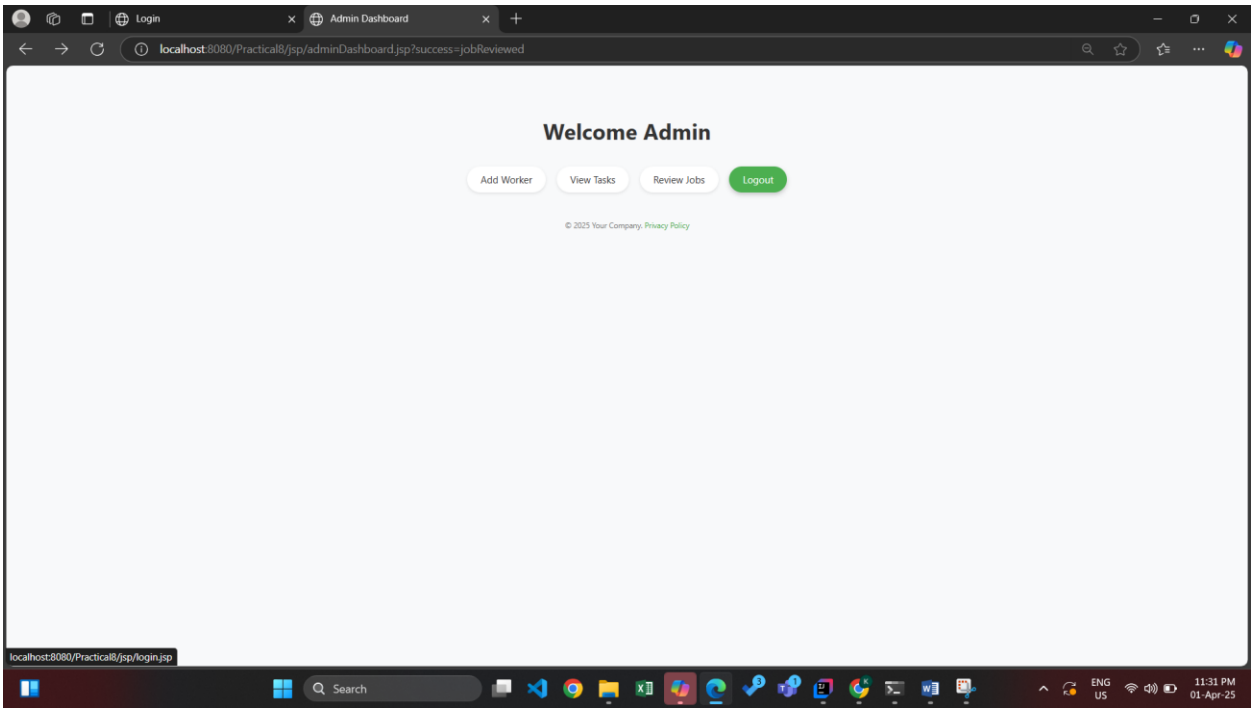


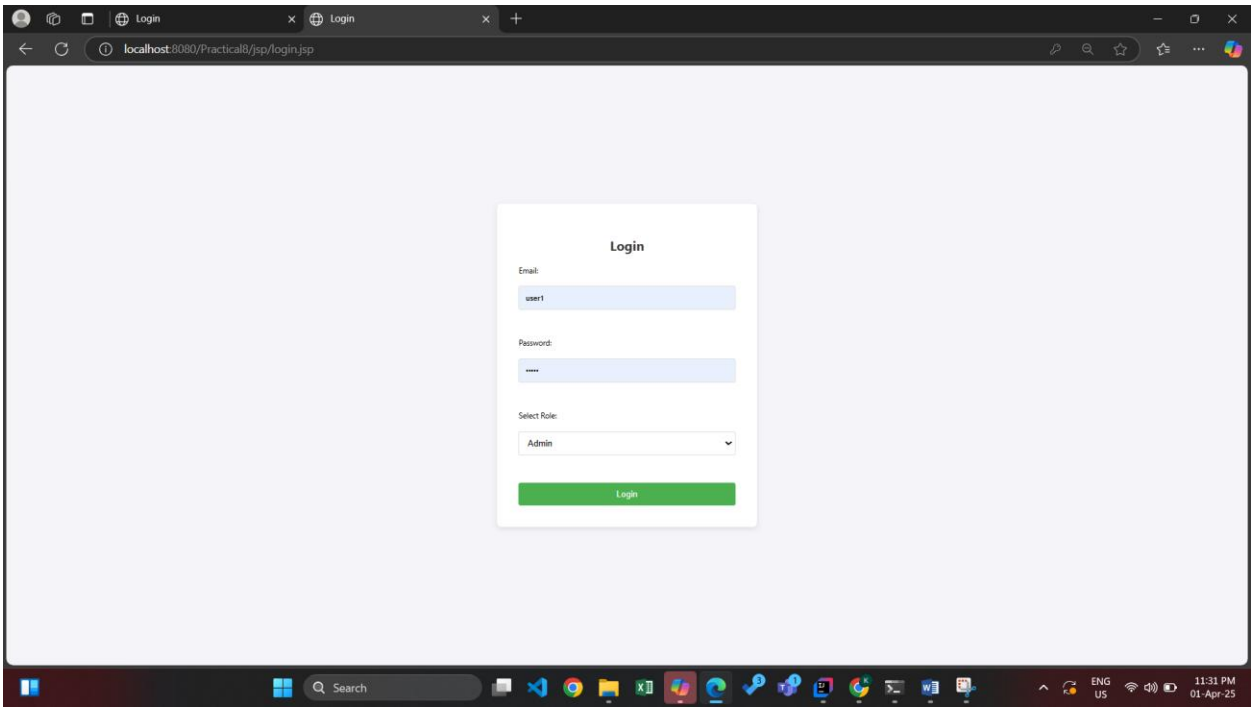


***It works and also (update) the table and in web.**

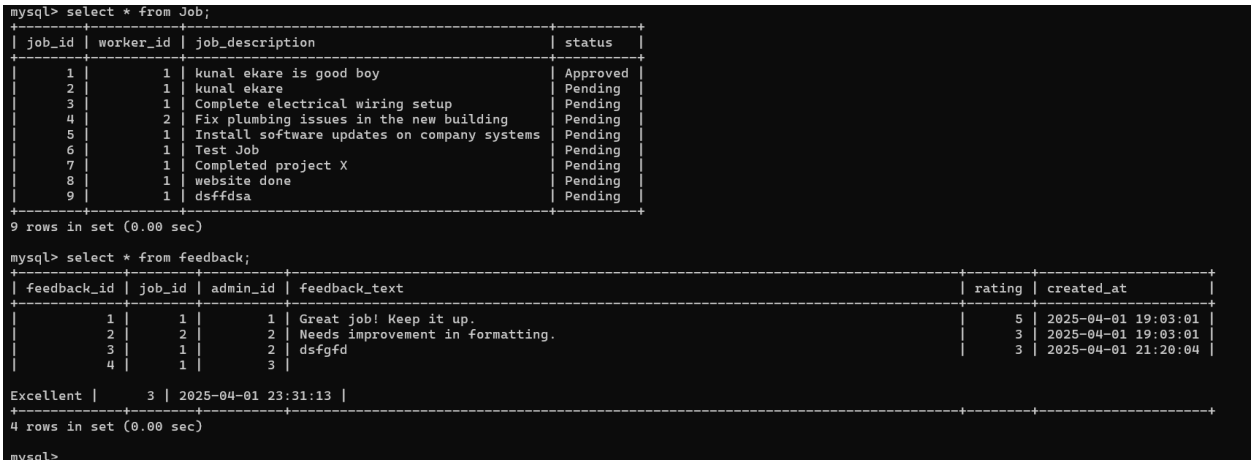
```
mysql> select * from Job;
+--------+-----------+-----------------------------------------------+-----------+
| job_id | worker_id | job_description                               | status    |
+--------+-----------+-----------------------------------------------+-----------+
|      1 |         1 | kunal ekare is good boy                       | Approved  |
|      2 |         1 | kunal ekare                                   | Pending   |
|      3 |         1 | Complete electrical wiring setup              | Pending   |
|      4 |         2 | Fix plumbing issues in the new building       | Pending   |
|      5 |         1 | Install software updates on company systems   | Pending   |
|      6 |         1 | Test Job                                      | Pending   |
|      7 |         1 | Completed project X                           | Pending   |
|      8 |         1 | website done                                  | Pending   |
|      9 |         1 | dsffdsa                                       | Pending   |
+--------+-----------+-----------------------------------------------+-----------+
9 rows in set (0.00 sec)
```

```
Database changed
mysql> select * from feedback;
+-------------+--------+----------+---------------------------------------------------------------------------------------------------------------+--------+---------------------+
| feedback_id | job_id | admin_id | feedback_text                                                                                                 | rating | created_at          |
+-------------+--------+----------+---------------------------------------------------------------------------------------------------------------+--------+---------------------+
|           1 |      1 |        1 | Great job! Keep it up.                                                                                        |      5 | 2025-04-01 19:03:01 |
|           2 |      2 |        2 | Needs improvement in formatting.                                                                              |      3 | 2025-04-01 19:03:01 |
|           3 |      1 |        2 | dsfgfd                                                                                                        |      3 | 2025-04-01 21:20:04 |
|           4 |      1 |        3 |
Excellent |      3 | 2025-04-01 23:31:13 |
+-------------+--------+----------+---------------------------------------------------------------------------------------------------------------+--------+---------------------+
4 rows in set (0.00 sec)

mysql>
```






```
mysql> show tables;
+---------------------+
| Tables_in_companydb |
+---------------------+
| admin               |
| feedback            |
| job                 |
| worker              |
+---------------------+
4 rows in set (0.01 sec)

mysql> select * from admin;
+----------+-------------+----------------------+-----------+
| admin_id | name        | email                | password  |
+----------+-------------+----------------------+-----------+
|        1 | NULL        | admin@example.com    | admin123  |
|        2 | Kunal Ekare | kunalekare02@gmail.com | pass123   |
|        3 | Rahul Sharma | kunalekare08@gmail.com | Pass@1234 |
+----------+-------------+----------------------+-----------+
3 rows in set (0.00 sec)

mysql> select * from worker;
+-----------+---------------------------+----------------------+-------------+--------+
| worker_id | name                      | email                | password    | rating |
+-----------+---------------------------+----------------------+-------------+--------+
|         1 | Kunal sanjay ekare        | kunalekare02@gmail.com | Pass@1234p  |      0 |
|         2 | Kunal sanjay ekare        | kunalekare03@gmail.com | PAss@1234p  |      0 |
|         3 | Kunal sanjay ekare        | kunalekare05@gmial.com | Pass@34     |      0 |
|         4 | Kunal sanjay ekare        | ekareks@rknec.edu    | Pass@1234   |      0 |
|         5 | Kunal sanjay ekare rcoemjhb | kunalekare04@gmial.com | Pass@345    |      0 |
|         6 | Kunal sanjay ekare        | Kunal03@gmail.com    | Pass@1234   |      0 |
|         7 | John Doe                  | john@example.com     | password123 |      0 |
|         8 | mahendra bahuali          | wanimj@rknec.edu     | Pass@123    |      0 |
+-----------+---------------------------+----------------------+-------------+--------+
8 rows in set (0.00 sec)
```



```
mysql> select * from Job;
+--------+-----------+-----------------------------------------+----------+
| job_id | worker_id | job_description                         | status   |
+--------+-----------+-----------------------------------------+----------+
|      1 |         1 | kunal ekare is good boy                 | Approved |
|      2 |         1 | kunal ekare                             | Pending  |
|      3 |         1 | Complete electrical wiring setup        | Pending  |
|      4 |         2 | Fix plumbing issues in the new building | Pending  |
|      5 |         1 | Install software updates on company systems | Pending  |
|      6 |         1 | Test Job                                | Pending  |
|      7 |         1 | Completed project X                     | Pending  |
|      8 |         1 | website done                            | Pending  |
|      9 |         1 | dsffdsa                                 | Pending  |
+--------+-----------+-----------------------------------------+----------+
9 rows in set (0.00 sec)

mysql> select * from feedback;
+-------------+--------+----------+-----------------------------------------------------------------------------------------------------------+--------+---------------------+
| feedback_id | job_id | admin_id | feedback_text                                                                                             | rating | created_at          |
+-------------+--------+----------+-----------------------------------------------------------------------------------------------------------+--------+---------------------+
|           1 |      1 |        1 | Great job! Keep it up.                                                                                    |      5 | 2025-04-01 19:03:01 |
|           2 |      2 |        2 | Needs improvement in formatting.                                                                          |      3 | 2025-04-01 19:03:01 |
|           3 |      1 |        2 | dsfgfd                                                                                                    |      3 | 2025-04-01 21:20:04 |
|           4 |      1 |        3 |

Excellent |      3 | 2025-04-01 23:31:13 |
+-------------+--------+----------+-----------------------------------------------------------------------------------------------------------+--------+---------------------+
4 rows in set (0.00 sec)

mysql>
```

Conclusion: we have successfully execute the program, web application using Servlets and Java server pages to design a menu driven application for Company management system.

Note: All file attached in RAR file in Google classroom.