**Lab 10**



**Submitted by:**

Samiyya Aftab

**Submitted to:**

Ma’am Shazwa

**Department of Software Engineering**

**Riphah International University, Islamabad**

**Output:**

1. Add Job

2. View All Jobs

3. Mark Job as Completed

4. Exit

Enter choice: 1

Enter job title: sd

Enter job description: ff

Job added.

1. Add Job

2. View All Jobs

3. Mark Job as Completed

4. Exit

Enter choice: 2

Job ID: 1, Title: sd, Description: ff, Completed: false

1. Add Job

2. View All Jobs

3. Mark Job as Completed

4. Exit

Enter choice: 3

Enter job ID to mark as completed: 1

Job 1 marked as completed.

1. Add Job

2. View All Jobs

3. Mark Job as Completed

4. Exit

Enter choice: 2

Job ID: 1, Title: sd, Description: ff, Completed: true

1. Add Job

2. View All Jobs

3. Mark Job as Completed

4. Exit

Enter choice: 4

Exiting...

BUILD SUCCESSFUL (total time: 22 seconds)

**Task:**

package task;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

JobView view = new JobView();

JobController controller = new JobController(view);

Scanner scanner = new Scanner(System.in);

System.out.println("Job Management Application");

while (true) {

System.out.println("\n1. Add Job");

System.out.println("2. View All Jobs");

System.out.println("3. Mark Job as Completed");

System.out.println("4. Exit");

System.out.print("Enter choice: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

System.out.print("Enter job title: ");

String title = scanner.nextLine();

System.out.print("Enter job description: ");

String description = scanner.nextLine();

controller.addJob(title, description);

System.out.println("Job added.");

break;

case 2:

controller.displayAllJobs();

break;

case 3:

System.out.print("Enter job ID to mark as completed: ");

int jobId = scanner.nextInt();

controller.markJobAsCompleted(jobId);

break;

case 4:

System.out.println("Exiting...");

scanner.close();

return;

default:

System.out.println("Invalid choice. Try again.");

}

}

}

}

package task;

public class Job {

private int id;

private String title;

private String description;

private boolean isCompleted;

// Constructor

public Job(int id, String title, String description) {

this.id = id;

this.title = title;

this.description = description;

this.isCompleted = false; // Default to not completed

}

// Getters

public int getId() {

return id;

}

public String getTitle() {

return title;

}

public String getDescription() {

return description;

}

public boolean isCompleted() {

return isCompleted;

}

// Mark the job as completed

public void markAsCompleted() {

this.isCompleted = true;

}

@Override

public String toString() {

return "Job ID: " + id + ", Title: " + title + ", Description: " + description + ", Completed: " + isCompleted;

}

}

package task;

import java.util.List;

public class JobView {

public void displayJobs(List<Job> jobs) {

if (jobs.isEmpty()) {

System.out.println("No jobs available.");

} else {

for (Job job : jobs) {

System.out.println(job);

}

}

}

}

package task;

import java.util.ArrayList;

import java.util.List;

public class JobController {

private List<Job> jobs;

private JobView view;

private int nextId;

public JobController(JobView view) {

this.jobs = new ArrayList<>();

this.view = view;

this.nextId = 1; // Start job IDs at 1

}

public void addJob(String title, String description) {

Job job = new Job(nextId++, title, description);

jobs.add(job);

}

public void displayAllJobs() {

view.displayJobs(jobs);

}

public void markJobAsCompleted(int jobId) {

for (Job job : jobs) {

if (job.getId() == jobId) {

job.markAsCompleted();

System.out.println("Job " + jobId + " marked as completed.");

return;

}

}

System.out.println("Job ID " + jobId + " not found.");

}

}