

```

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.*;

public class AdminPanel extends JPanel {
    public AdminPanel(JobPortalFrame app) {
        setLayout(new BorderLayout());

        JLabel title = new JLabel("Admin Dashboard", SwingConstants.CENTER);
        title.setFont(new Font("Arial", Font.BOLD, 20));
        add(title, BorderLayout.NORTH);

        JTabbedPane tabs = new JTabbedPane();

        // Users Table
        tabs.addTab("All Users", createTablePanel(
            new String[]{"ID", "Name", "Role"},
            Store.allUsers().stream()
                .map(u -> new Object[]{u.getId(), u.getName(), u.getRole()})
                .toArray(Object[][]::new)
        ));

        // Jobs Table
        tabs.addTab("All Jobs", createTablePanel(
            new String[]{"ID", "Title", "Company", "Description"},
            Store.allJobs().stream()
                .map(j -> new Object[]{j.getId(), j.getTitle(), j.getCompany(), j.getDesc()})
                .toArray(Object[][]::new)
        ));

        add(tabs, BorderLayout.CENTER);
    }

    private JPanel createTablePanel(String[] columns, Object[][] data) {
        JTable table = new JTable(new DefaultTableModel(data, columns));
        return new JPanel(new BorderLayout()) {{
            add(new JScrollPane(table), BorderLayout.CENTER);
        }};
    }
}

public class RecruiterPanel extends JPanel {

```

```

private DefaultTableModel jobModel;

public RecruiterPanel(JobPortalFrame app) {
    setLayout(new BorderLayout());

    JLabel title = new JLabel("Recruiter Dashboard", SwingConstants.CENTER);
    title.setFont(new Font("Arial", Font.BOLD, 20));
    add(title, BorderLayout.NORTH);

    // === Form to Add New Job ===
    JPanel form = new JPanel(new GridLayout(4, 2, 10, 10));
    JTextField titleField = new JTextField();
    JTextField companyField = new JTextField();
    JTextArea descArea = new JTextArea(3, 20);
    JButton addBtn = new JButton("Post Job");

    form.setBorder(BorderFactory.createTitledBorder("Post New Job"));
    form.add(new JLabel("Job Title:")); form.add(titleField);
    form.add(new JLabel("Company:")); form.add(companyField);
    form.add(new JLabel("Description:")); form.add(new JScrollPane(descArea));
    form.add(new JLabel()); form.add(addBtn);
    add(form, BorderLayout.NORTH);

    // === Job Table ===
    String[] cols = {"ID", "Title", "Company", "Description"};
    jobModel = new DefaultTableModel(cols, 0);
    JTable jobTable = new JTable(jobModel);
    add(new JScrollPane(jobTable), BorderLayout.CENTER);
    refreshJobTable();

    // === Button Action ===
    addBtn.addActionListener(e -> {
        String title = titleField.getText().trim();
        String company = companyField.getText().trim();
        String desc = descArea.getText().trim();

        if (title.isEmpty() || company.isEmpty() || desc.isEmpty()) {
            JOptionPane.showMessageDialog(app, "All fields are required!");
            return;
        }

        int newId = Store.allJobs().size() + 1;
        Store.addJob(new Job(newId, title, company, desc));
        JOptionPane.showMessageDialog(app, "Job posted successfully!");
    });
}

```

```
        titleField.setText("");
        companyField.setText("");
        descArea.setText("");
        refreshJobTable();
    });
}

private void refreshJobTable() {
    jobModel.setRowCount(0);
    for (Job j : Store.allJobs()) {
        jobModel.addRow(new Object[]{j.getId(), j.getTitle(), j.getCompany(), j.getDesc()});
    }
}
}
```