## Name: **BHV Satyanarayana**

**Index.html:**

<!DOCTYPE html>

<html>

<head>

<title>JSP</title>

</head>

<body bgcolor=*"lightblue"*>

<h1 align=*"center"*>Digitalization of the Hospitality Process</h1>

<hr color=*"black"*><hr color=*"black"*><br>

<style>

**button**{

color: *black*;

background-color: *white*;

padding: *6px*;

cursor:*pointer*;

border-radius: *30px*;

}

**button***:hover* {

background-color:*pink*;

color:*darkblack*;

}

</style>

<button onclick="window.location.href='index.html'">Home</button>

<button onclick="window.location.href='takecsv.html'"> Perform operations</button>

<br><br>

</body>

</html>

**Takecsv.hml:**

<!DOCTYPE html>

<html lang=*"en"*>

<head>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>

<title>Upload CSV Files</title>

<style>

**pre** {

white-space: *pre-wrap*;

background-color: *#f0f0f0*;

padding: *10px*;

border-radius: *5px*;

}

</style>

</head>

<body>

<h1>Upload CSV Files</h1>

<form id=*"uploadForm"* enctype=*"multipart/form-data"*>

<label for=*"studentsFile"*>Students CSV File:</label>

<input type=*"file"* id=*"studentsFile"* name=*"studentsFile"* accept=*".csv"*><br><br>

<label for=*"hostelFile"*>Hostel CSV File:</label>

<input type=*"file"* id=*"hostelFile"* name=*"hostelFile"* accept=*".csv"*><br><br>

<button type=*"button"* onclick="uploadFiles()">Upload Files</button>

</form>

<div id=*"fileContents"*>

<!-- Content will be displayed here -->

</div>

<script>

**function** uploadFiles() {

**var** form = document.getElementById("uploadForm");

**var** formData = **new** FormData(form);

fetch('/upload.jsp', {

method: 'POST',

body: formData

})

.then(response => response.text())

.then(data => {

document.getElementById('fileContents').innerHTML = data;

})

.**catch**(error => {

console.error('Error uploading files:', error);

});

}

</script>

</body>

</html>

**Upload.jsp :**

<%@ page contentType=*"text/html;charset=UTF-8"* %>

<%@ page language=*"java"* %>

<%@ page import=*"java.io.\*, java.util.\*, javax.servlet.\*, javax.servlet.annotation.\*, javax.servlet.http.\*, com.opencsv.\*"* %>

<%@ page import=*"your.package.Group, your.package.HostelRoom"* %>

<%

response.setContentType("text/html;charset=UTF-8");

**try** {

// Get the uploaded files

Part studentsFilePart = request.getPart("studentsFile");

Part hostelFilePart = request.getPart("hostelFile");

// Check if both files are uploaded

**if** (studentsFilePart != **null** && hostelFilePart != **null**) {

// Read contents of studentsFilePart

InputStream studentsFileStream = studentsFilePart.getInputStream();

CSVReader studentsReader = **new** CSVReader(**new** InputStreamReader(studentsFileStream));

// Read contents of hostelFilePart

InputStream hostelFileStream = hostelFilePart.getInputStream();

CSVReader hostelReader = **new** CSVReader(**new** InputStreamReader(hostelFileStream));

// Prepare HTML content to return

StringBuilder htmlResponse = **new** StringBuilder();

htmlResponse.append("<h2>Contents of Students CSV File:</h2>");

htmlResponse.append("<pre>");

String[] nextLine;

**while** ((nextLine = studentsReader.readNext()) != **null**) {

htmlResponse.append(Arrays.toString(nextLine)).append("<br>");

}

htmlResponse.append("</pre>");

htmlResponse.append("<h2>Contents of Hostel CSV File:</h2>");

htmlResponse.append("<pre>");

**while** ((nextLine = hostelReader.readNext()) != **null**) {

htmlResponse.append(Arrays.toString(nextLine)).append("<br>");

}

htmlResponse.append("</pre>");

// Send the HTML response back to the client

out.println(htmlResponse.toString());

} **else** {

out.println("<p>File upload failed. Please upload both CSV files.</p>");

}

} **catch** (Exception e) {

// Handle any exceptions

e.printStackTrace();

out.println("<p>Error processing files: " + e.getMessage() + "</p>");

}

%>

**Logic:**

import java.util.\*;

class GroupInfo {

int groupId;

int members;

String gender;

public GroupInfo(int groupId, int members, String gender) {

this.groupId = groupId;

this.members = members;

this.gender = gender;

}

}

class HostelRoom {

String hostelName;

int roomNumber;

int capacity;

String genderAccommodation;

public HostelRoom(String hostelName, int roomNumber, int capacity, String genderAccommodation) {

this.hostelName = hostelName;

this.roomNumber = roomNumber;

this.capacity = capacity;

this.genderAccommodation = genderAccommodation;

}

}

public class RoomAllocator {

public static void main(String[] args) {

List<GroupInfo> groups = loadGroupInfoFromCSV("group\_info.csv");

List<HostelRoom> rooms = loadHostelRoomsFromCSV("hostel\_info.csv");

Map<Integer, List<HostelRoom>> allocatedRooms = new HashMap<>();

for (GroupInfo group : groups) {

List<HostelRoom> suitableRooms = findSuitableRooms(group, rooms);

allocateToRooms(group, suitableRooms, allocatedRooms);

}

displayAllocation(allocatedRooms);

exportToCSV(allocatedRooms, "allocation\_details.csv");

}

private static List<GroupInfo> loadGroupInfoFromCSV(String filename) {

return new ArrayList<>();

}

private static List<HostelRoom> loadHostelRoomsFromCSV(String filename) {

return new ArrayList<>();

}

private static List<HostelRoom> findSuitableRooms(GroupInfo group, List<HostelRoom> rooms) {

List<HostelRoom> suitableRooms = new ArrayList<>();

for (HostelRoom room : rooms) {

if (room.genderAccommodation.equalsIgnoreCase(group.gender) &&

room.capacity >= group.members) {

suitableRooms.add(room);

}

}

return suitableRooms;

}

private static void allocateToRooms(GroupInfo group, List<HostelRoom> suitableRooms, Map<Integer, List<HostelRoom>> allocatedRooms) {

// Implement allocation logic

}

private static void displayAllocation(Map<Integer, List<HostelRoom>> allocatedRooms) {

// Implement display logic

}

private static void exportToCSV(Map<Integer, List<HostelRoom>> allocatedRooms, String filename) {

// Implement CSV export logic

}

}

**Necessary instructions to run the application:**

Open in Red Hat code Ready Studio

Run project on Server