This Python project is a Flask web application that facilitates the allocation of hostel rooms to groups based on specified criteria. The application takes two CSV files as input: one for group details and one for hostel room details. The output is displayed on a webpage and can be downloaded as a CSV file. Below is a detailed breakdown of the logic and usage of the project:

**Backend Logic**

1. **Flask Setup and Routes:**
   * The Flask app is initialized, and two main routes are defined:
     + The root route '/' which handles both GET and POST requests.
     + The /download\_csv route for downloading the allocation results as a CSV file.
2. **File Upload Handling:**
   * The '/' route's POST method processes the uploaded CSV files for groups and hostels.
   * The files are passed to helper functions parse\_group\_file and parse\_hostel\_file to extract and structure the data.
3. **CSV Parsing Functions:**
   * parse\_group\_file(file):
     + Reads the group CSV file and extracts details like group ID, number of members, and gender.
   * parse\_hostel\_file(file):
     + Reads the hostel CSV file and extracts details like hostel name, room number, capacity, gender, and available capacity.
4. **Room Allocation Logic:**
   * The allocate\_rooms(groups, hostels) function allocates rooms to groups based on matching gender and available capacity.
   * The allocation results are structured in a list of dictionaries, each containing group ID, hostel name, room number, and number of members allocated.
5. **CSV Download Function:**
   * The /download\_csv route generates a CSV file from the allocation results and sends it as a downloadable file.

**Frontend Usage**

1. **Main Page (mainpage.html):**
   * Provides a form for users to upload the group and hostel CSV files.
   * The form uses POST method to submit the files to the root route '/'.
2. **Result Page (result.html):**
   * Displays the room allocation results in a table format.
   * Provides a link to download the results as a CSV file.

Code : ( python )

from flask import Flask, render\_template, request, send\_file

import csv

import io

app = Flask(\_\_name\_\_)

@app.route('/', methods=['GET', 'POST'])

def index():

if request.method == 'POST':

group\_file = request.files['group\_file']

hostel\_file = request.files['hostel\_file']

groups = parse\_group\_file(group\_file)

hostels = parse\_hostel\_file(hostel\_file)

allocations = allocate\_rooms(groups, hostels)

return render\_template('result.html', allocations=allocations)

return render\_template('mainpage.html')

@app.route('/download\_csv')

def download\_csv():

# Generate CSV file from allocations

# This is a placeholder and needs to be implemented

output = io.StringIO()

writer = csv.writer(output)

writer.writerow(['Group ID', 'Hostel Name', 'Room Number', 'Members Allocated'])

# Add allocation data here

output.seek(0)

return send\_file(output, mimetype='text/csv', as\_attachment=True, attachment\_filename='allocations.csv')

def parse\_group\_file(file):

groups = []

csv\_reader = csv.reader(file.stream.read().decode("UTF-8").splitlines())

next(csv\_reader) # Skip header

for row in csv\_reader:

group\_id, members, gender = row

groups.append({

'id': group\_id,

'members': int(members),

'gender': gender

})

return groups

def parse\_hostel\_file(file):

hostels = []

csv\_reader = csv.reader(file.stream.read().decode("UTF-8").splitlines())

next(csv\_reader) # Skip header

for row in csv\_reader:

hostel\_name, room\_number, capacity, gender = row

hostels.append({

'name': hostel\_name,

'room': room\_number,

'capacity': int(capacity),

'gender': gender,

'available': int(capacity)

})

return hostels

def allocate\_rooms(groups, hostels):

allocations = []

for group in groups:

allocated\_members = 0

for hostel in hostels:

if hostel['gender'] == group['gender'] and hostel['available'] > 0:

allocated = min(group['members'] - allocated\_members, hostel['available'])

allocations.append({

'group\_id': group['id'],

'hostel\_name': hostel['name'],

'room\_number': hostel['room'],

'members\_allocated': allocated

})

hostel['available'] -= allocated

allocated\_members += allocated

if allocated\_members == group['members']:

break

return allocations

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

code : (mainpage.html)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Room Allocation System</title>

</head>

<body>

<h1>Room Allocation System</h1>

<form action="/" method="post" enctype="multipart/form-data">

<label for="group\_file">Group CSV File:</label>

<input type="file" id="group\_file" name="group\_file" accept=".csv" required><br><br>

<label for="hostel\_file">Hostel CSV File:</label>

<input type="file" id="hostel\_file" name="hostel\_file" accept=".csv" required><br><br>

<input type="submit" value="Allocate Rooms">

</form>

</body>

</html>

Code : result.html   
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Room Allocation Results</title>

</head>

<body>

<h1>Room Allocation Results</h1>

<table border="1">

<tr>

<th>Group ID</th>

<th>Hostel Name</th>

<th>Room Number</th>

<th>Members Allocated</th>

</tr>

{% for allocation in allocations %}

<tr>

<td>{{ allocation.group\_id }}</td>

<td>{{ allocation.hostel\_name }}</td>

<td>{{ allocation.room\_number }}</td>

<td>{{ allocation.members\_allocated }}</td>

</tr>

{% endfor %}

</table>

<br>

<a href="{{ url\_for('download\_csv') }}">Download CSV</a>

</body>

</html>

**Usage**

1. **Launch the Flask application.**
2. **Access the main page** via the browser where you can upload the group and hostel CSV files.
3. **Submit the form** to get the room allocation results displayed on the results page.
4. **Download the results** as a CSV file by clicking the "Download CSV" link.