**BerkeleyServer.java**

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class BerkeleyServer {

private static final int PORT = 5000;

private static List<int[]> clientTimes = new ArrayList<>();

private static List<Socket> clientSockets = new ArrayList<>();

private static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

try (ServerSocket serverSocket = new ServerSocket(PORT)) {

System.out.println("Server started. Waiting for 3 clients...");

// Accept exactly 3 clients

while (clientSockets.size() < 3) {

Socket clientSocket = serverSocket.accept();

clientSockets.add(clientSocket);

System.out.println("Client connected: " + clientSocket.getInetAddress());

}

// Step 1: Get server time input

System.out.print("Enter server time (HH MM): ");

int serverHours = scanner.nextInt();

int serverMinutes = scanner.nextInt();

int serverTotalMinutes = serverHours \* 60 + serverMinutes;

System.out.println("Server time: " + formatTime(serverHours, serverMinutes));

// Step 2: Receive times from clients

for (Socket socket : clientSockets) {

try {

DataInputStream in = new DataInputStream(socket.getInputStream());

int hours = in.readInt();

int minutes = in.readInt();

clientTimes.add(new int[]{hours, minutes});

System.out.println("Received time from client: " + formatTime(hours, minutes));

} catch (EOFException e) {

System.out.println("Client disconnected before sending data.");

}

}

// Step 3: Compute the average time

int totalMinutes = serverTotalMinutes;

for (int[] time : clientTimes) {

totalMinutes += time[0] \* 60 + time[1];

}

int averageMinutes = totalMinutes / (clientTimes.size() + 1); // +1 for server

int averageHours = (averageMinutes / 60) % 24;

int averageMins = averageMinutes % 60;

System.out.println("Average Adjusted Time: " + formatTime(averageHours, averageMins));

// Step 4: Compute adjustments and send to each client

for (int i = 0; i < clientSockets.size(); i++) {

Socket socket = clientSockets.get(i);

try {

DataOutputStream out = new DataOutputStream(socket.getOutputStream());

int[] clientTime = clientTimes.get(i);

int clientTotalMinutes = clientTime[0] \* 60 + clientTime[1];

int adjustment = averageMinutes - clientTotalMinutes;

int adjustHours = adjustment / 60;

int adjustMins = adjustment % 60;

int[] adjustedTime = adjustTime(clientTime, adjustHours, adjustMins);

out.writeInt(adjustHours);

out.writeInt(adjustMins);

out.writeInt(adjustedTime[0]);

out.writeInt(adjustedTime[1]);

out.flush();

System.out.println("Sent adjustment (" + formatTime(adjustHours, adjustMins) +

") to client. Adjusted time: " + formatTime(adjustedTime[0], adjustedTime[1]));

} catch (IOException e) {

System.out.println("Error sending adjustment to client.");

}

}

// Step 5: Close connections

for (Socket socket : clientSockets) {

socket.close();

}

} catch (IOException e) {

e.printStackTrace();

}

}

private static int[] adjustTime(int[] time, int hourOffset, int minuteOffset) {

int totalMinutes = (time[0] + hourOffset) \* 60 + (time[1] + minuteOffset);

return new int[]{(totalMinutes / 60) % 24, totalMinutes % 60}; // Keep time within 24-hour format

}

private static String formatTime(int hours, int minutes) {

return String.format("%02d:%02d", hours, minutes);

}

}

**BerkeleyClient.java**

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class BerkeleyClient {

private static final String SERVER\_IP = "127.0.0.1"; // Change if needed

private static final int PORT = 5000;

private static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

try (Socket socket = new Socket(SERVER\_IP, PORT)) {

DataOutputStream out = new DataOutputStream(socket.getOutputStream());

DataInputStream in = new DataInputStream(socket.getInputStream());

// Step 1: Get user input for local time

System.out.print("Enter client's current time (HH MM): ");

int hours = scanner.nextInt();

int minutes = scanner.nextInt();

int[] localTime = {hours, minutes};

System.out.println("Client's current time: " + formatTime(localTime[0], localTime[1]));

out.writeInt(localTime[0]); // Send hours

out.writeInt(localTime[1]); // Send minutes

out.flush(); // Ensure data is sent before reading

// Step 2: Receive time adjustment from server

int hourAdjustment = in.readInt();

int minuteAdjustment = in.readInt();

int adjustedHours = in.readInt();

int adjustedMinutes = in.readInt();

System.out.println("Time adjustment received: " + formatTime(hourAdjustment, minuteAdjustment));

System.out.println("Client's adjusted time: " + formatTime(adjustedHours, adjustedMinutes));

} catch (EOFException e) {

System.out.println("Server closed the connection unexpectedly.");

} catch (IOException e) {

System.out.println("Error connecting to server.");

e.printStackTrace();

}

}

private static String formatTime(int hours, int minutes) {

return String.format("%02d:%02d", hours, minutes);

}

}

**Output**







