Candidate Management System Code & Output:

```
using System;
using System.Collections.Generic;
using System.Linq;

public class Candidate
{
    public string FullName { get; set; }
    public string Skills { get; set; }
    public bool IsQualified { get; set; }
}

public class InterviewUtility
{
    public bool Register(string fullName, string skills)
    {
        if (Program.CandidatesSet.Any(c => c.FullName.Equals(fullName,
StringComparison.OrdinalIgnoreCase)))
        {
            return false;
        }

        Program.CandidatesSet.Add(new Candidate { FullName = fullName, Skills
= skills, IsQualified = false });
        return true;
    }

    public bool UpdateCandidateSkills(string fullName, string newSkills)
    {
        var candidate = Program.CandidatesSet.FirstOrDefault(c =>
c.FullName.Equals(fullName, StringComparison.OrdinalIgnoreCase));

        if (candidate != null)
        {
            candidate.Skills = newSkills;
            return true;
        }
        return false;
    }

    public List<string> MarkCandidateAsQualified(string requiredSkills)
{
    List<string> qualifiedCandidates = new List<string>();
```

```
    foreach (var candidate in Program.CandidatesSet)
    {
        var candidateSkills = candidate.Skills.Split(',')
            .Select(s => s.Trim().ToLower()).ToList();

        var requiredSkillList = requiredSkills.Split(',')
            .Select(s => s.Trim().ToLower()).ToList();

        if (requiredSkillList.Any(skill => candidateSkills.Contains(skill)))
        {
            candidate.IsQualified = true;
            qualifiedCandidates.Add(candidate.FullName);
        }
    }

    return qualifiedCandidates;
}

}

public class Program
{
    public static HashSet<Candidate> CandidatesSet { get; set; } = new
HashSet<Candidate>();

    public static void Main(string[] args)
    {
        InterviewUtility utility = new InterviewUtility();

        while (true)
        {
            Console.WriteLine("\nTech Recruit System");
            Console.WriteLine("1. Register Candidate");
            Console.WriteLine("2. Update Candidate Skills");
            Console.WriteLine("3. Mark Candidates as Qualified");
            Console.WriteLine("4. Exit");
            Console.Write("Enter your choice: ");

            string choice = Console.ReadLine();

            switch (choice)
            {
                case "1":
                    Console.Write("Enter full name: ");
                    string fullName = Console.ReadLine();
                    Console.Write("Enter skills (use ' , ' if more than 1 ):
");
```

```
                    string skills = Console.ReadLine();
                    if (utility.Register(fullName, skills))
                        Console.WriteLine($"Candidate '{fullName}' registered
successfully.");
                    else
                        Console.WriteLine($"Candidate '{fullName}' already
registered.");
                    break;

                case "2":
                    Console.Write("Enter full name of the candidate to update:
");
                    string nameToUpdate = Console.ReadLine();
                    Console.Write("Enter new skills (use ' , '): ");
                    string newSkills = Console.ReadLine();
                    if (utility.UpdateCandidateSkills(nameToUpdate,
newSkills))
                        Console.WriteLine($"Skills of candidate
'{nameToUpdate}' updated successfully.");
                    else
                        Console.WriteLine($"Candidate '{nameToUpdate}' not
found.");
                    break;

                case "3":
                    Console.Write("Enter required skills for the position: ");
                    string requiredSkills = Console.ReadLine();
                    List<string> qualifiedCandidates =
utility.MarkCandidateAsQualified(requiredSkills);
                    if (qualifiedCandidates.Any())
                    {
                        Console.WriteLine($"Qualified candidates for the
position requires '{requiredSkills}':");
                        qualifiedCandidates.ForEach(Console.WriteLine);
                    }
                    else
                        Console.WriteLine("No qualified candidates found for
the position.");
                    break;

                case "4":
                    Console.WriteLine("Exiting Tech Recruit System");
                    return;

                default:
                    Console.WriteLine("Invalid choice,,enter a valid
option.");
                    break;
```

```
            }
        }
    }
}
```

Output:

```
Tech Recruit System
1. Register Candidate
2. Update Candidate Skills
3. Mark Candidates as Qualified
4. Exit
Enter your choice: 1
Enter full name: Harsha Kapu
Enter skills (use ' , ' if more than 1 ): Java, C, C++
Candidate 'Harsha Kapu' registered successfully.

Tech Recruit System
1. Register Candidate
2. Update Candidate Skills
3. Mark Candidates as Qualified
4. Exit
Enter your choice: 1
Enter full name: Avinash Kapu
Enter skills (use ' , ' if more than 1 ): C, C++
Candidate 'Avinash Kapu' registered successfully.

Tech Recruit System
1. Register Candidate
2. Update Candidate Skills
3. Mark Candidates as Qualified
4. Exit
Enter your choice: 3
Enter required skills for the position: C
Qualified candidates for the position requires 'C':
Harsha Kapu
Avinash Kapu

Tech Recruit System
1. Register Candidate
2. Update Candidate Skills
3. Mark Candidates as Qualified
4. Exit
Enter your choice: E:\CAPGEMINI PREP\CandidateManagementSystem\CandidateManagementSystem.csproj
```

----------Thank You----------