

**Задатак:** Користећи шаблон клијент-сервер апликације написати апликацију која ће водити рачуна о евиденцији поруџбина хране. **Као клијентску апликацију користити веб претраживач!**

За сваку поруџбину се евидентирају:

- код
- адреса
- телефон поручиоца
- тип производа
- начин плаћања

Тип производа може имати једну од следећих вредности:

- Burger
- Pizza
- Palacinka
- Sendvic

Начини плаћања могу имати једну од следећих вредности:

- Gotovina
- Kartica

Клијент на сервер шаље параметре из форме за унос (слика 1).

## Dodaj novu porudzbину

Kod porudzbine:	<input type="text"/>
Adresa:	<input type="text"/>
Broj telefona:	<input type="text"/>
Izaberite proizvod:	<input type="text" value="Burger"/>
Nacin placanja:	<input type="text" value="Gotovina"/>
<input type="button" value="Dodaj"/>	

### Brisanje porudzbine:

Kod:	<input type="text"/>
<input type="button" value="Obrisi"/>	

**Слика 1: Почетна страница**

Ако корисник покуша да дода поруџбину са већ постојећим кодом, исписује се порука, као и линк за повратак на почетну страницу (слика 2).

## Porudzbina sa kodom 12 vec postoji!

[Povratak na pocetnu stranicu](#)

**Слика 2: Страница грешке да поруџбина већ постоји**

Након клика на дугме **Dodaj** прелази се на табелу која приказује списак свих поруџбина, као и на линк који омогућава повратак на почетну страницу (слика 3).

Porudzbine				
KOD	Adresa	Telefon	Proizvod	Tip placanja
1	Kacka ulica 2	0621234322	PIZZA	KARTICA
2	Safarikova 2	0693245324	PALACINKA	GOTOVINA

[Povratak na pocetnu stranicu](#)

**Слика 3: Страница за приказ унетих поруџбина**

Корисник има могућност на почетној страници да обрише поруџбину уносећи њен код. Након што унесе код поруџбине и кликне дугме **Obrisi**, прелази на табеларни приказ као на слици (слика 4).

Uspesno je obrisana porudzbina sa kodom: 1			
Preostale porudzbine			
Adresa	Telefon	Proizvod	Tip placanja
Novasadska 431A	061232322	PIZZA	GOTOVINA

[Povratak na pocetnu stranicu](#)

**Слика 4: Страница за приказ пронађење поруџбине**

Уколико не постоји таква поруџбина, исписује се порука (слика 5).

## Ne postoji porudzbina sa kodom 3!

[Povratak na pocetnu stranicu](#)

**Слика 5: Страница за приказ грешке**

## Решење

### Статичка страница index.html

```
<html lang="sr">
  <head>
    <title>SOV 1 - Dodavanje porudzbine</title>
  </head>
  <body>
    <h1>Dodaj novu porudzbину</h1>
    <form action="dodaj">
      <table>
        <tr>
          <td><label>Kod porudzbine: </label></td>
          <td><input type="text" name="kod" required /></td>
        </tr>
        <tr>
          <td><label>Adresa: </label></td>
          <td><input type="text" name="adresa" required /></td>
        </tr>
        <tr>
          <td><label>Broj telefona: </label></td>
          <td><input type="text" name="telefon" required /></td>
        </tr>
        <tr>
          <td><label>Izaberite proizvod: </label></td>
          <td>
            <select name="proizvod" required>
              <option value="Burger">Burger</option>
              <option value="Pizza">Pizza</option>
              <option value="Palacinka">Palacinka</option>
              <option value="Sendvic">Sendvic</option>
            </select>
          </td>
        </tr>
        <tr>
          <td><label>Nacin placanja: </label></td>
          <td>
            <select name="placanje" required>
              <option value="Gotovina">Gotovina</option>
              <option value="Kartica">Kartica</option>
            </select>
          </td>
        </tr>
        <tr>
          <td><input type="submit" value="Dodaj" /></td>
        </tr>
      </table>
    </form>

    <form action="brisanje">
      <table>
        <tr>
          <td colspan="2"><h3>Brisanje porudzbine:</h3></td>
        </tr>
      </table>
    </form>
```

```

        <tr>
            <td>Kod: </td>
            <td><input type="text" name="kod" /></td>
        </tr>
        <tr>
            <td><input type="submit" value="Obrisi" /></td>
        </tr>
    </table>
</form>
</body>
</html>

```

### Серверски C# код Porudzbina.cs

```

namespace Httpd
{
    public class Porudzbina
    {
        public int Kod { get; set; }
        public string Adresa { get; set; }
        public string Telefon { get; set; }
        public string Proizvod { get; set; }
        public string NacinPlacanja { get; set; }

        public Porudzbina(int kod, string adresa, string telefon, string p, string np)
        {
            Kod = kod;
            Adresa = adresa;
            Telefon = telefon;
            Proizvod = p;
            NacinPlacanja = np;
        }
    }
}

```

### Серверски C# код Program.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Net;
using System.Net.Sockets;
using System.Threading.Tasks;

namespace Httpd
{
    class Program
    {
        public static List<Porudzbina> porudzbine = new List<Porudzbina>();

        public static void StartListening()
        {
            IPAddress ipAddress = IPAddress.Loopback;
            IPEndPoint localEndPoint = new IPEndPoint(ipAddress, 8080);

            Socket serverSocket = new Socket(AddressFamily.InterNetwork,

```

```

        SocketType.Stream, ProtocolType.Tcp);

    try
    {
        serverSocket.Bind(localEndPoint);
        serverSocket.Listen(10);

        while (true)
        {
            Console.WriteLine("Waiting for a connection...");
            Socket socket = serverSocket.Accept();

            Task t = Task.Factory.StartNew(() => Run(socket));
        }
    }
    catch (Exception e)
    {
        Console.WriteLine(e.ToString());
    }

    Console.WriteLine("\nPress ENTER to continue...");
    Console.Read();
}

private static void Run(Socket socket)
{
    NetworkStream stream = new NetworkStream(socket);
    StreamReader sr = new StreamReader(stream);
    StreamWriter sw = new StreamWriter(stream){NewLine="\r\n",AutoFlush=true };

    string resource = GetResource(sr);

    if (resource != null)
    {
        if (resource.Equals(""))
            resource = "index.html";

        Console.WriteLine("Request from " + socket.RemoteEndPoint + ": " +
            resource + "\n");

        if (resource.StartsWith("dodaj?"))
        {
            string[] porudzbina = resource.Split(new string[]
            { "kod=", "adresa=", "telefon=", "proizvod=", "placanje=" },
            StringSplitOptions.None);
            sw.Write("HTTP/1.0 200 OK\r\n\r\n");

            string kod_string = GetPropertyValue(porudzbina[1]);
            string adresa = GetPropertyValue(porudzbina[2]);
            string telefon = GetPropertyValue(porudzbina[3]);
            string proizvod = GetPropertyValue(porudzbina[4]);
            string placanje = GetPropertyValue(porudzbina[5]);

            if (!int.TryParse(kod_string, out int kod))
            {
                sw.Write("<html><body><h2 style='color: crimson;'>Niste uneli
                    broj za kod porudzbine!</h2>" + "<br/><a href='/'>

```

```

        Povratak na pocetnu stranicu</a></body></html>");
    }
    else
    {
        // ako postoji porudzbina ispisati gresku
        foreach (Porudzbina p in porudzbine)
        {
            if (p.Kod == kod)
            {
                sw.Write("<html><body><h2>Porudzbina sa kodom " +
                    kod_string + " vec postoji!</h2>" +
                    "<a href='/'>Povratak na pocetnu
                    stranicu</a></body></html>");
                return;
            }
        }

        Porudzbina nova = new Porudzbina(kod, adresa, telefon,
                                           proizvod, placanje);
        porudzbine.Add(nova);

        // ispis tabele sa svim porudzbinama nakon uspesnog dodavanja
        sw.Write("<html><head><style>th { padding: 0px 30px; }
            td { padding: 0px 50px 0px 0px; }</style></head><body>");
        sw.Write(IspisTabelePorudzbina());
        sw.Write("</body></html>");
    }
}
else if (resource.StartsWith("brisanje?"))
{
    string[] porudzbina = resource.Split(new string[]
        { "kod=" }, StringSplitOptions.None);
    string kod_string = GetPropertyValue(porudzbina[1]);
    sw.Write("HTTP/1.0 200 OK\r\n\r\n");

    if (!int.TryParse(kod_string, out int kod))
    {
        sw.Write("<html><body><h2 style='color: crimson;'>
            Niste uneli broj za kod porudzbine!</h2>" +
            "<br/><a href='/'>Povratak na pocetnu
            stranicu</a></body></html>");
    }
    else
    {
        foreach (Porudzbina p in porudzbine)
        {
            if (p.Kod == kod)
            {
                porudzbine.Remove(p);

                sw.Write("<html><head><style>
                    th { padding: 0px 30px; }
                    td { padding: 0px 50px 0px 0px; }
                    </style></head><body>");
                sw.Write(IspisPorudzbine(p));
                sw.Write("</body></html>");

                return;
            }
        }
    }
}

```

```

    }

    // porudzbina ne postoji, ispisati gresku
    sw.Write("<html><body><h2>Ne postoji porudzbina sa kodom " +
        kod_string + "!</h2>" +
        "<a href='/'>Povratak na pocetnu"
        "stranicu</a></body></html>");
    }
}
else if (resource.StartsWith("pregled"))
{
    sw.Write("HTTP/1.0 200 OK\r\n\r\n");
    sw.Write("<html><head><style>th { padding: 0px 30px; }"
        "td { padding: 0px 50px 0px 0px; }</style></head><body>");
    sw.Write(IspisTabelePorudzbina());
    sw.Write("</body></html>");
}
else
{
    SendResponse(resource, socket, sw);
}
}

sr.Close();
sw.Close();
stream.Close();

socket.Shutdown(SocketShutdown.Both);
socket.Close();
}

private static string GetPropertyValue(string field)
{
    string new_field = field.Split('&')[0];
    new_field = Uri.UnescapeDataString(new_field);
    new_field = new_field.Replace("+", " ");

    return new_field;
}

private static string IspisTabelePorudzbina()
{
    string ispis = "";

    ispis += "<table border='1'>" +
        "<tr>" +
        "<th colspan='5' style='text-align: center;'>Porudzbine</th>" +
        "</tr>" +
        "<tr>" +
        "<th>KOD</th>" +
        "<th>Adresa</th>" +
        "<th>Telefon</th>" +
        "<th>Proizvod</th>" +
        "<th>Tip placanja</th>" +
        "</tr>";

    if (porudzbine.Count == 0)
    {
        ispis += "<tr><td colspan='5'>Nema porudzbina</td></tr>";
    }
}

```

```

    }
    else
    {
        foreach (Porudzbina p in porudzbine)
        {
            ispis += "<tr>" +
                "<td>" + p.Kod + "</td>" +
                "<td>" + p.Adresa + "</td>" +
                "<td>" + p.Telefon + "</td>" +
                "<td>" + p.Proizvod.ToUpper() + "</td>" +
                "<td>" + p.NacinPlacanja.ToUpper() + "</td>" +
                "</tr>";
        }
    }

    ispis += "</table>\r\n";
    ispis += "<br/><a href='/'>Povratak na pocetnu stranicu</a>";

    return ispis;
}

public static string IspisPorudzbine(Porudzbina p)
{
    string ispis = "";

    ispis += "<table border='1'>" +
        "<tr>" +
        "<th colspan='5' style='text-align: center;'>Uspesno je obrisana" +
        "porudzbina sa kodom: " + p.Kod + "</th>" +
        "</tr>" +
        "<tr>" +
        "<th colspan='5' style='text-align: center;'>Preostale porudzbine</th>" +
        " +
        "</tr>" +
        "<tr>" +
        "<th>Adresa</th>" +
        "<th>Telefon</th>" +
        "<th>Proizvod</th>" +
        "<th>Tip placanja</th>" +
        "</tr>";

    foreach (Porudzbina pn in porudzbine)
    {
        ispis += "<tr>" +
            "<td>" + pn.Adresa + "</td>" +
            "<td>" + pn.Telefon + "</td>" +
            "<td>" + pn.Proizvod.ToUpper() + "</td>" +
            "<td>" + pn.NacinPlacanja.ToUpper() + "</td>" +
            "</tr>";
    }

    ispis += "</table>\r\n";
    ispis += "<br/><a href='/'>Povratak na pocetnu stranicu</a>";

    return ispis;
}

```



```
private static string GetResource(StreamReader sr)
{
    string line = sr.ReadLine();
    if (line == null)
        return null;
    String[] tokens = line.Split(' ');

    string method = tokens[0];
    if (!method.Equals("GET"))
    {
        return null;
    }

    string rsrc = tokens[1];

    rsrc = rsrc.Substring(1);

    string s1;
    while (!(s1 = sr.ReadLine()).Equals(""))
        Console.WriteLine(s1);
    Console.WriteLine("Request: " + line);
    return rsrc;
}

private static void SendResponse(string resource, Socket socket, StreamWriter sw)
{
    resource = "../.../" + resource;
    FileInfo fi = new FileInfo(resource);

    string responseText;
    if (!fi.Exists)
    {
        responseText = "HTTP/1.0 404 File not found\r\n"
            + "Content-type: text/html; charset=UTF-8\r\n\r\n"
            + "<b>404 Stranica nije pronadjena!:"
            + fi.Name + "</b>";
        sw.Write(responseText);
        Console.WriteLine("Could not find resource: " + fi.Name);
        return;
    }

    responseText = "HTTP/1.0 200 OK\r\n\r\n";
    sw.Write(responseText);
    socket.SendFile(resource);
}

public static int Main(String[] args)
{
    StartListening();
    return 0;
}
}
```