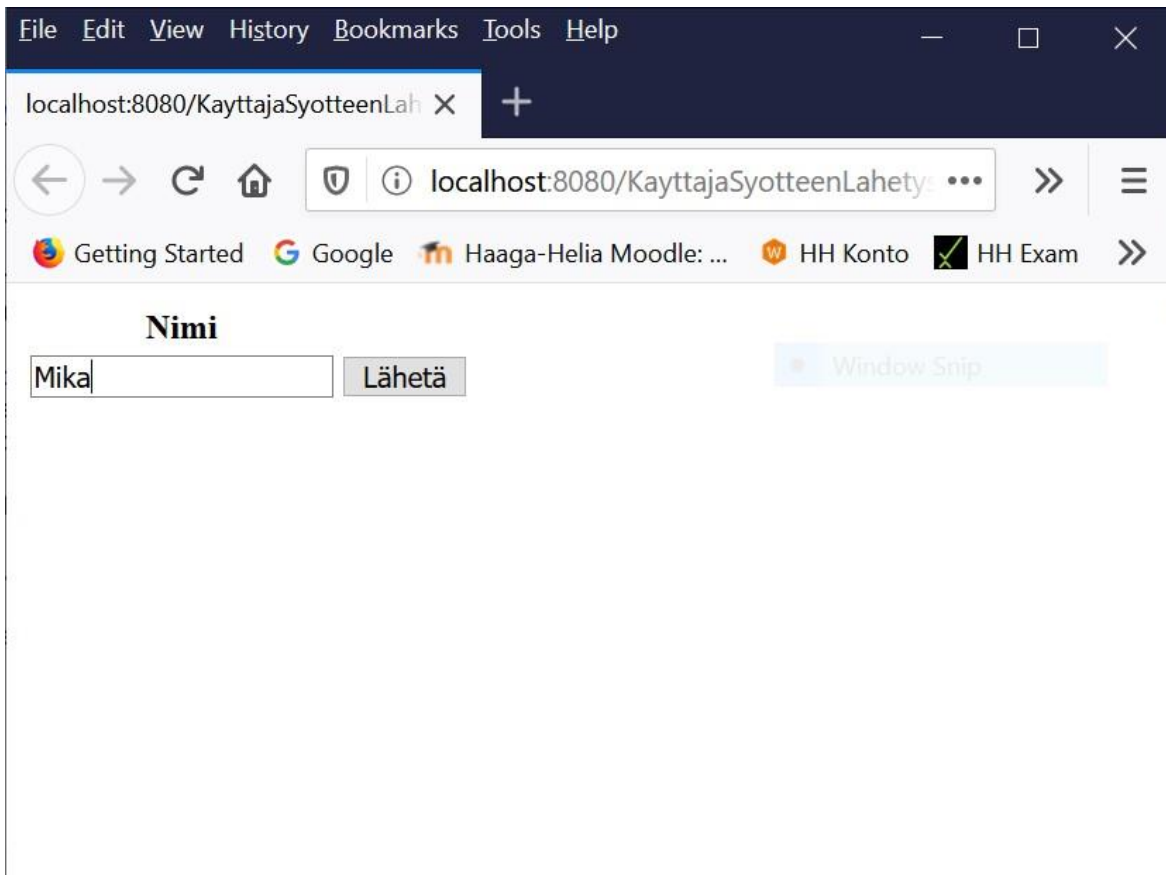


Tuntitehtävä 3b:

Tee alla olevan kuvan mukainen HTML sivu, joka lähettää POST viestin otavastaan/ nimiselle Servlet:ille:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/KayttajaSyotteenLahety...'. The browser's menu bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. The address bar also shows 'localhost:8080/KayttajaSyotteenLahety...'. Below the address bar, there are several bookmarks: 'Getting Started', 'Google', 'Haaga-Helia Moodle: ...', 'HH Konto', and 'HH Exam'. The main content area of the browser displays a form titled 'Nimi'. The form consists of a text input field containing the text 'Mika' and a button labeled 'Lähetä'. A 'Window Snip' watermark is visible in the bottom right corner of the browser window.

Käytä tässä html:n form ja table elementissä olevaa painonappulaa, joka käynnistää lahetaTiedot() JavaScript funktion:

```
<form id="tiedot">
<table>
  <thead>
    <tr>
      <th>Nimi</th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td><input type="text" name="nimi" id="nimi"></td>
      <td><input type="button" name="nappi" id="taLlenna"
value="Lähetä" onclick="lahetaTiedot()"></td>
    </tr>
  </tbody>
</table>
</form>
```

```
<span id="ilmo"></span>
```

JavaScript funktio on puolestaan seuraavanlainen:

```
function lahetaTiedot(){
    var formJsonStr=formDataToJSON(document.getElementById("tiedot"));
    //muutetaan lomakkeen tiedot json-stringiksi

    fetch("otavastaan/",{
        method: 'POST',
        body:formJsonStr
    })
    .then( function (responseJson) {
        var vastaus = responseJson.response;
        if(vastaus==0){
            document.getElementById("ilmo").innerHTML= "epaonnistui";
        }else if(vastaus==1){
            document.getElementById("ilmo").innerHTML= "onnistui";
        }
    });
}
```

Tämä lahetaTiedot() funktio käynnistää fetch() funktion, jonka parametrit ovat seuraavanlaiset:

```
fetch(käynnitettäväServlet,
käynnitettäväMetodiServletissä)
    .then(mitäTehdäänSitten).then(mitäTehdäänSenJälkeen)
```

fetch() funktio siis lähettää HTML sivulle syötetyn nimen otavastaan nimiselle Servletille ja käynnistää siellä doPost() metodin. Tähän servlet:iin ja doPost metodiin tulee seuraavat lauseet:

```
package controller;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.json.JSONObject;

@WebServlet("/otavastaan/*")
```

```

public class KayttajaSyotteenLahetys extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public KayttajaSyotteenLahetys() {
        super();
        // TODO Auto-generated constructor stub
    }

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        JSONObject jsonObj = new JsonStrToObj().convert(request);
        //Muutetaan kutsun mukana tuleva json-string json-objektiksi
        String nimi = jsonObj.getString("nimi");
        System.out.println("Annettu nimi on " + nimi);
    }
}

```

Ensimmäiseksi muutetaan HTML sivulta lähetetty käyttäjäsyöte eli tässä tapauksessa nimi JSON olioksi. Tämä tarvitsee `json-20180813.jar` Java kirjaston, jonka joudut kopioimaan kurssin Moodle sivuilta Java projektisi `WebContent/WEB-INF/lib` kansioon. Lisäksi kopioi Moodlessa tehtävän alla oleva `JSONObject.java` tiedosto samaan kansioon missä `Servlet`:isi on. Tämän jälkeen konvertointi onnistuu Java olioksi. Seuraavassa lauseessa pystytään löytämään tästä Java oliosta parametri nimeltään `nimi`, jossa käyttäjän syöttämä data on. Lopuksi tulos tulostetaan Eclipsen konsoli-ikkunaan. Alla oleva kuva näyttää miltä ajotilanne näyttää:

