

**Aim:**

To implement client-side state management using HTTP cookies in Java Servlets by creating a preference system that remembers user's theme and language choices across browser sessions.

**ALGORITHM:****1. UI Design Phase**

- Create a preferences page with:
- Theme selection (light/dark) radio buttons
- Language selection dropdown (EN/ES/FR)
- Save preferences button
- Design dynamic page that reflects current preferences

**2. Cookie Creation Phase**

- On form submission:
- Extract theme and language parameters □ Create new Cookie objects for each preference □ Set cookie properties:
- `maxAge` = 1 year (persistent cookie)
- `path` = "/" (accessible across entire app)
- Add cookies to HTTP response

**3. Preference Retrieval Phase**

- For every page request:
- Check request for existing cookies □ If cookies exist:
- Read 'theme' and 'language' values
- Apply preferences to current page □ If no cookies found:
- Use default values (dark theme, English)

**4. Client-Side Enhancement**

- Implement JavaScript to:
- Provide instant theme preview
- Show language change effects without reload
- Display save confirmation

**5. Maintenance & Security** □ Cookie validation checks □ Secure cookie attributes: `HttpOnly` flag □ SameSite policy

- Fallback mechanism for cookie-disabled browsers

## CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Neon Preferences</title>
  <style>
    :root {
      --neon-blue: #08f7fe;
      --neon-green: #09fbd3;
      --dark-bg: #0c0c15;
      --darker-bg: #07070e;
      --light-bg: #f5f5f5;
      --lighter-bg: #ffffff;
      --dark-text: #222;
      --glow: 0 0 10px;
    }
    body {
      background: var(--dark-bg);
      color: white;
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      margin: 0; padding: 20px; min-height: 100vh; display: flex;
      flex-direction: column; align-items: center; transition: all
      0.5s ease; overflow-x: hidden;
    }
    body.light-theme {
      background: var(--
      light-bg);
      color: var(--dark-text);
    }

    .container {
      max-width: 800px; width: 90%;
      background: rgba(12, 12, 21, 0.8); border-
      radius: 15px; padding: 30px; box-shadow: 0 0
      30px rgba(8, 247, 254, 0.2); border: 1px
      solid rgba(8, 247, 254, 0.3); backdrop-
      filter: blur(5px); position: relative;
      overflow: hidden; z-index: 1; animation:
      fadeIn 1s ease-out; transition: all 0.5s
      ease;
    }

    .light-theme .container {
      background:
      rgba(245, 245, 245, 0.9); border: 1px
      solid rgba(0, 0, 0, 0.1); box-shadow: 0
      0 30px rgba(0, 0, 0, 0.1);
    }

    @keyframes fadeIn {
      from { opacity: 0; transform:
      translateY(20px); }
      to { opacity: 1; transform:
      translateY(0); }
    }
```

```

h1 { color: var(--neon-blue); text-align: center; font-
      size: 2.5rem; margin-bottom: 30px; text-
      shadow: var(--glow) var(--neon-blue);
      transition: all 0.5s ease;
    }

    .light-theme h1 {
      color: #0066cc;
      text-shadow: none;
    }

    .preference-card { background: rgba(7, 7, 14,
      0.7); border-radius: 10px; padding: 20px;
      margin-bottom: 30px; border: 1px solid
      rgba(8, 247, 254, 0.2); transition: all
      0.3s ease;
    }

    .light-theme .preference-card {
      background: rgba(255, 255, 255, 0.9);
      border: 1px solid rgba(0, 0, 0, 0.1);
    }

h2 { color: var(--neon-green); margin-top: 0; font-size:
      1.5rem; text-shadow: var(--glow) var(--neon-
      green); transition: all 0.5s ease;
    }

    .light-theme h2 {
      color: #009999;
      text-shadow: none;
    }

    .form-group { margin-bottom:
      20px;
    }

label { display: block; margin-bottom:
      8px; color: var(--neon-blue);
      font-weight: bold; transition:
      all 0.5s ease;
    }

    .light-theme label {
      color: #0066cc;
    }

select { width: 100%; padding: 12px; background:
      var(--darker-bg); border: 1px solid
      var(--neon-blue); border-radius: 5px;
      color: white; font-size: 1rem;
      transition: all 0.3s ease;
    }

```

```
.light-theme select { background:
    var(--lighter-bg); border: 1px
    solid #0066cc; color: var(--
    dark-text);
}
select:focus { outline:
    none;
```

```

    }

    border-color: var(--neon-green); box-shadow:
    0 0 15px var(--neon-green);

    .light-theme select:focus { border-color:
    #009999; box-shadow: 0 0 15px rgba(0, 153, 153,
    0.5); }

input[type="submit"] { width: 100%; padding: 12px; background: linear-
gradient(45deg, var(--neon-blue), var(--neon-
green)); color: var(--dark-bg);
font-weight: bold; cursor:
pointer; border: none;
border-radius: 5px;
margin-top: 10px; text-
transform: uppercase;
letter-spacing: 1px; font-
size: 1rem; transition:
all 0.3s ease;

}

.light-theme input[type="submit"] { background: linear-
gradient(45deg, #0066cc, #009999); color: white;
}

input[type="submit"]:hover { transform: translateY(-2px);
box-shadow: 0 5px 15px rgba(8, 247, 254, 0.4); }

.light-theme input[type="submit"]:hover { box-
shadow: 0 5px 15px rgba(0, 102, 204, 0.4); }

.current-preferences {
display: flex; gap:
20px; margin-
bottom: 30px;
}

.preference-badge { flex: 1; padding:
15px; text-align: center;
background: rgba(8, 247, 254, 0.1);
border-radius: 8px; border: 1px
solid var(--neon-blue); animation:
float 4s ease-in-out infinite;
transition: all 0.5s ease;

.light-theme .preference-badge {
background: rgba(0, 102, 204, 0.1);
border: 1px solid #0066cc;

```

```

}

}

.preference-badge:nth-child(2) {
  animation-delay: 0.5s; border-
  color: var(--neon-green);
  background: rgba(9, 251, 211, 0.1);
}

.light-theme .preference-badge:nth-child(2) {
  border-color: #009999; background:
  rgba(0, 153, 153, 0.1);
}

@keyframes float {
  0%, 100% { transform: translateY(0); }
  50% { transform: translateY(-10px); }
}

.preference-value { font-size:
  1.2rem; font-weight: bold;
  margin-top: 5px; color:
  var(--neon-green);
  transition: all 0.5s ease;
}

.light-theme .preference-value { color:
  #009999;
}

.theme-blue .preference-value { color:
  var(--neon-blue);
}

.light-theme .theme-blue .preference-value { color:
  #0066cc;
}

.home-link { display: inline-block;
  margin-top: 20px; color: var(--
  neon-blue); text-decoration: none;
  font-weight: bold; border: 1px
  solid var(--neon-blue); padding:
  10px 20px;
  border-radius: 5px; transition:
  all 0.3s ease;
}

.light-theme .home-link {
  color: #0066cc; border:
  1px solid #0066cc;

```

```

    }

    }

    .home-link:hover { background: var(--neon-
        blue); color: var(--dark-bg); text-
        shadow: none; box-shadow: 0 0 15px
        var(--neon-blue);
    }

    .light-theme .home-link:hover {
        background: #0066cc; color:
        white;
        box-shadow: 0 0 15px #0066cc;
    }

    /*      Language-specific      styles      */
    body[lang="es"] .language-example::after {
        content: "¡Hola!"; color: var(--neon-
        green);
    }
    body[lang="fr"] .language-example::after {
        content: "Bonjour!"; color: var(--neon-
        green);
    }
    body[lang="en"] .language-example::after {
        content: "Hello!"; color: var(--neon-
        green);
    }

    .light-theme body[lang] .language-example::after { color:
        #009999;
    }
</style>
</head>
<body class="${theme}-theme" lang="${language}">
    <div class="container">
        <h1>USER PREFERENCES</h1>

        <div class="current-preferences">
            <div class="preference-badge theme-blue">
                <div>Current Theme</div>
                <div class="preference-value">${theme}</div>
            </div>
            <div class="preference-badge">

```

```

        <div>Current Language</div>
        <div class="preference-value">${language}</div>
    </div>
</div>

<div class="language-example" style="text-align: center; margin: 20px 0; font-size: 1.2rem;">
    Example greeting: <span></span>
</div>

<div class="preference-card">
    <h2>SET YOUR PREFERENCES</h2>
    <form method="post" action="preferences">
        <div class="form-group">
            <label for="theme">Theme</label>
            <select name="theme" id="theme" onchange="previewTheme()">
                <option value="dark" ${"dark" == theme ? "selected" : ""}>Neon Dark Theme</option>
                <option value="light" ${"light" == theme ? "selected" : ""}>Light Theme</option>
            </select>
        </div>

        <div class="form-group">
            <label for="language">Language</label>
            <select name="language" id="language"
onchange="previewLanguage()">
                <option value="en" ${"en" == language ? "selected" : ""}>English</option>
                <option value="es" ${"es" == language ? "selected" : ""}>Spanish</option>
                <option value="fr" ${"fr" == language ? "selected" : ""}>French</option>
            </select>
        </div>

        <input type="submit" value="Save Preferences">
    </form>
</div>

<a href="index.html" class="home-link">Back to Home</a>
</div>

<script>
    // Theme preview function function previewTheme() { const
    themeSelect = document.getElementById('theme');
    document.body.classList.remove('dark-theme', 'light-theme');
    document.body.classList.add(themeSelect.value + '-theme'); }

```



```

// Language preview function function
previewLanguage() { const langSelect
=
document.getElementById('language');
document.body.setAttribute('lang',
langSelect.value); }

// Initialize with current theme if already set
document.addEventListener('DOMContentLoaded', function() { const
themeSelect = document.getElementById('theme'); if (themeSelect) {
document.body.classList.add(themeSelect.value + '-theme'); }
const langSelect = document.getElementById('language'); if
(langSelect) { document.body.setAttribute('lang',
langSelect.value); } });
</script>
</body>
</html>

```

### PreferenceServlet.java:

```

package com.example.cookies;
import java.io.*; import
javax.servlet.*; import
javax.servlet.http.*;
public class PreferenceServlet extends HttpServlet
{
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    // Read cookies
    String theme = "dark"; // default to dark theme for neon effect String
    language = "en"; // default

    Cookie[] cookies = request.getCookies(); if
    (cookies != null) { for (Cookie cookie :
    cookies) { if
    (cookie.getName().equals("theme")) { theme =
    cookie.getValue();
        } else if (cookie.getName().equals("language")) { language
        = cookie.getValue();
        }
    }
    }

    // Generate HTML with neon styling
    out.println("<!DOCTYPE html>");
    out.println("<html lang=\"" + language + "\">");
    out.println("<head>"); out.println("<meta
charset=\"UTF-8\">"); out.println("<meta
name=\"viewport\" content=\"width=device-width,

```

```

initial-scale=1.0\>");    out.println("<title>Neon    Preferences</title>");
    out.println("<style>"); out.println(":root {"); out.println("  --neon-
blue:    #08f7fe;");    out.println("    --neon-green:    #09fbd3;");
    out.println("  --dark-bg:    #0c0c15;");    out.println("  --darker-bg:
#07070e;"); out.println("  --light-bg:    #f5f5f5;"); out.println("  --
lighter-bg:    #ffffff;");    out.println("    --dark-text:    #222;");
    out.println("  --glow: 0 0 10px;"); out.println("}"); out.println("");
    out.println("body {"); out.println("  background: var(--dark-bg);");
    out.println("  color: white;"); out.println("  font-family: 'Segoe UI',
Tahoma, Geneva, Verdana, sans-
serif;");
        out.println("  margin: 0;");
        out.println("    padding:        20px;");
        out.println("    min-height:    100vh;");
        out.println("    display:        flex;");
        out.println("    flex-direction:    column;");
        out.println("    align-items:    center;");
        out.println("  transition: all 0.5s ease;");
        out.println("    overflow-x:    hidden;");
        out.println("  background-image: ");
        out.println("    radial-gradient(circle at 20% 30%, rgba(8, 247, 254,
0.1) 0%, transparent 25%),");
        out.println("    radial-gradient(circle at 80% 70%, rgba(9, 251, 211,
0.1) 0%, transparent 25%);");
        out.println("}");
        out.println(""); out.println(".container {"); out.println("
max-width:    800px;");    out.println("  width:    90%;");
        out.println("  background:    rgba(12, 12, 21, 0.8);");
        out.println("  border-radius: 15px;"); out.println("  padding:
30px;"); out.println("  box-shadow: 0 0 30px rgba(8, 247, 254,
0.2);"); out.println("  border: 1px solid rgba(8, 247, 254,
0.3);");    out.println("  backdrop-filter:    blur(5px);");
        out.println("  position: relative;"); out.println("  overflow:
hidden;");    out.println("  z-index:    1;");    out.println("
animation:    fadeIn 1s ease-out;");    out.println("}");
        out.println(""); out.println("@keyframes fadeIn {");
        out.println("  from { opacity: 0; transform: translateY(20px); }");
        out.println("  to { opacity: 1; transform: translateY(0); }");
        out.println("}"); out.println(""); out.println("h1 {"); out.println("
color:    var(--neon-blue);");    out.println("  text-align:    center;");
        out.println("  font-size: 2.5rem;");    out.println("  margin-bottom:
30px;"); out.println("  text-shadow: var(--glow) var(--neon-blue);");
        out.println("  animation:    pulse 3s infinite alternate;");
        out.println("}"); out.println(""); out.println("@keyframes pulse {");
        out.println("  0% { text-shadow: 0 0 10px var(--neon-blue); }");
        out.println("  100% { text-shadow: 0 0 20px var(--neon-blue), 0 0 30px
var(--neon-green); }"); out.println("}"); out.println("");
        out.println(".preference-card {"); out.println("
background:    rgba(7, 7, 14, 0.7);");    out.println("  border-radius: 10px;");
        out.println("  padding:        20px;"); out.println("  margin-bottom: 30px;");
        out.println("  border: 1px solid rgba(8, 247, 254, 0.2);");
        out.println("  transition: all 0.3s ease;"); out.println("}");
        out.println(""); out.println("h2 {"); out.println("  color:

```

```

var(--neon-green));"); out.println("    margin-top:    0;");
out.println("    font-size: 1.5rem;"); out.println("    text-
shadow: var(--glow) var(--neon-green);"); out.println("}");
out.println(""); out.println(".form-group {"); out.println("
margin-bottom: 20px;"); out.println("}"); out.println("");
out.println("label {"); out.println("    display: block;");
out.println("    margin-bottom: 8px;"); out.println("    color:
var(--neon-blue);"); out.println("    font-weight: bold;");
out.println("}"); out.println("");
out.println("select, input[type=\"submit\"] {");
out.println("    width: 100%;"); out.println("    padding:
12px;"); out.println("    background: var(--darker-
bg);"); out.println("    border: 1px solid var(--neon-
blue);"); out.println("    border-radius: 5px;");
out.println("    color: white;");
out.println("    font-size: 1rem;"); out.println("    transition: all 0.3s
ease;"); out.println("}"); out.println("");
out.println("select:focus {"); out.println("    outline: none;");
out.println("    border-color: var(--neon-green);"); out.println("
box-shadow: 0 0 15px var(--neon-green);"); out.println("}");
out.println(""); out.println("input[type=\"submit\"] {");
out.println("    background: linear-gradient(45deg, var(--neon-blue),
var(--neon-green));"); out.println("    color:    var(--dark-bg);");
out.println("    font-weight: bold;"); out.println("    cursor:
pointer;"); out.println("    border: none;"); out.println("    margin-
top: 10px;"); out.println("    text-transform: uppercase;");
out.println("    letter-spacing: 1px;"); out.println("}");
out.println(""); out.println("input[type=\"submit\"]:hover {");
out.println("    transform: translateY(-2px);"); out.println("    box-
shadow: 0 5px 15px rgba(8, 247, 254, 0.4);"); out.println("}");
out.println(""); out.println(".current-preferences {");
out.println("    display: flex;"); out.println("    gap: 20px;");
out.println("    margin-bottom: 30px;"); out.println("}");
out.println("");
out.println(".preference-badge {"); out.println("    flex:
1;"); out.println("    padding: 15px;"); out.println("    text-
align: center;"); out.println("    background: rgba(8, 247,
254, 0.1);"); out.println("    border-radius: 8px;");
out.println("    border: 1px solid var(--neon-blue);");
out.println("    animation: float 4s ease-in-out infinite;");
out.println("}"); out.println("");
out.println(".preference-badge:nth-child(2) {");
out.println("    animation-delay: 0.5s;"); out.println("
border-color:    var(--neon-green);"); out.println("
background: rgba(9, 251, 211, 0.1);"); out.println("}");
out.println(""); out.println("@keyframes float {");
out.println("    0%, 100% { transform: translateY(0); }");
out.println("    50% { transform: translateY(-10px); }");
out.println("}");
out.println(""); out.println(".preference-value {"); out.println("
font-size: 1.2rem;"); out.println("    font-weight: bold;"); out.println("
margin-top: 5px;"); out.println("    color:    var(--neon-green);");
out.println("}"); out.println(""); out.println(".home-link {"); out.println("
display: inline-block;"); out.println("    margin-top: 20px;"); out.println("

```

```

color: var(--neon-blue);"); out.println(" text-decoration: none;");
out.println(" font-weight: bold;"); out.println(" border: 1px solid var(--
neon-blue);"); out.println(" padding: 10px 20px;"); out.println(" border-
radius: 5px;"); out.println(" transition: all 0.3s ease;"); out.println("}");
out.println(""); out.println(".home-link:hover {"); out.println(" background:
var(--neon-blue);"); out.println(" color: var(--dark-bg);"); out.println("
text-shadow: none;"); out.println(" box-shadow: 0 0 15px var(--neon-blue);");
out.println("}"); out.println(""); out.println("body[lang=\"es\"] .language-
example::after {"); out.println(" content: \"¡Hola!\";"); out.println(" color:
var(--neon-green);"); out.println("}"); out.println("");
out.println("body[lang=\"fr\"] .language-example::after {"); out.println("
content: \"Bonjour!\";"); out.println(" color: var(--neon-green);");
out.println("}"); out.println(""); out.println("body[lang=\"en\"] .language-
example::after {"); out.println(" content: \"Hello!\";"); out.println(" color:
var(--neon-green);"); out.println("}"); out.println("</style>");
out.println("</head>"); out.println("<body class=\"" + ("dark".equals(theme)
? "dark-theme" : "light-theme") + "\" lang=\"" + language + "\">");
out.println("<div class=\"container\">");
    out.println("<h1>USER
    PREFERENCES</h1>");
out.println("<div class=\"current-preferences\">"); out.println("<div
class=\"preference-badge theme-blue\">");
    out.println("<div>Current Theme</div>"); out.println("<div
class=\"preference-value\">" + theme + "</div>");
    out.println("</div>"); out.println("<div class=\"preference-
badge\">"); out.println("<div>Current Language</div>");
    out.println("<div class=\"preference-value\">" + language +
"</div>"); out.println("</div>"); out.println("</div>");
out.println("<div class=\"language-example\" style=\"text-align:
center; margin: 20px 0; font-size: 1.2rem;\">");
    out.println("Example greeting: <span></span>");
    out.println("</div>");
out.println("<div class=\"preference-card\">"); out.println("<h2>SET
YOUR PREFERENCES</h2>"); out.println("<form method=\"post\"
action=\"preferences\">"); out.println("<div class=\"form-
group\">"); out.println("<label for=\"theme\">Theme</label>");
    out.println("<select name=\"theme\" id=\"theme\">");
    out.println("<option value=\"dark\"" + ("dark".equals(theme) ? "
selected" : "") + ">Neon Dark Theme</option>"); out.println("<option
value=\"light\"" + ("light".equals(theme) ? "
selected" : "") + ">Light Theme</option>"); out.println("</select>");
    out.println("</div>"); out.println("<div class=\"form-
group\">"); out.println("<label
for=\"language\">Language</label>"); out.println("<select
name=\"language\" id=\"language\">"); out.println("<option
value=\"en\"" + ("en".equals(language) ? "
selected" : "") + ">English</option>"); out.println("<option
value=\"es\"" + ("es".equals(language) ? "
selected" : "") + ">Spanish</option>"); out.println("<option
value=\"fr\"" + ("fr".equals(language) ? "
selected" : "") + ">French</option>");
    out.println("</select>");
    out.println("</div>");

```

```

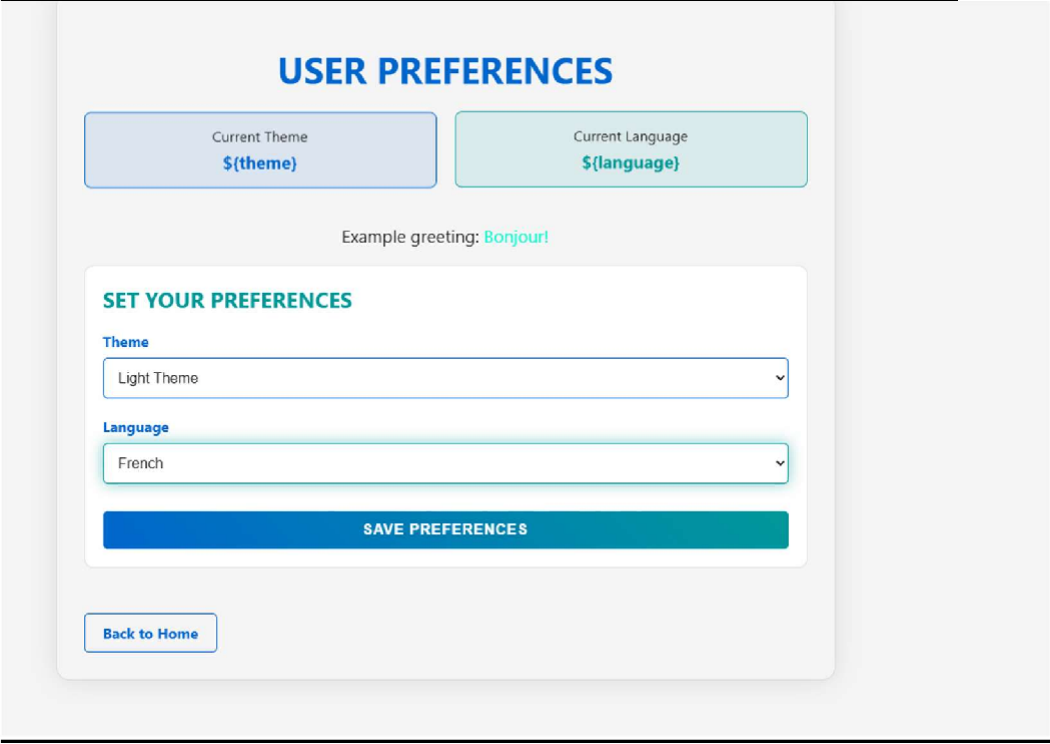
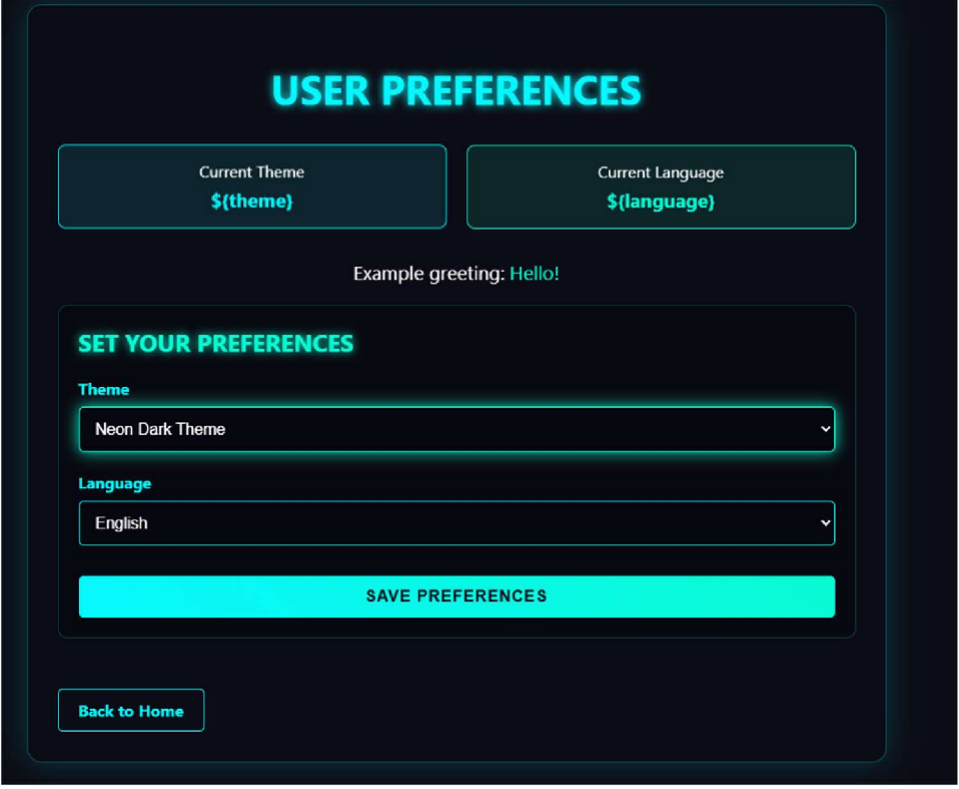
out.println("<input type=\"submit\" value=\"Save Preferences\">");
    out.println("</form>"); out.println("</div>")
    out.println("<a href=\"index.html\" class=\"home-link\">Back to
Home</a>");
    out.println("</div>");
    out.println("<script>");
    out.println("// Theme preview function");
    out.println("function previewTheme() {");
    out.println("    const themeSelect =
document.getElementById('theme');"); out.println("
    document.body.classList.remove('dark-theme', 'light-
theme');"); out.println("    document.body.classList.add(themeSelect.value +
'_
theme');");
    out.println("}");
    out.println("");
    out.println("//
Language preview
function");
    out.println("functio
n previewLanguage()
{"); out.println("
    const langSelect =
document.getElementById('language');"); out.println("
    document.body.setAttribute('lang',
langSelect.value);"); out.println("}"); out.println(""); out.println("//
Initialize with current theme if already set");
    out.println("document.addEventListener('DOMContentLoaded',
function()
{"); out.println("    const themeSelect =
document.getElementById('theme');"); out.println("    if (themeSelect) {");
    out.println("        document.body.classList.add(themeSelect.value + '-
theme');"); out.println("    }");
    out.println("    "); out.println("
    const langSelect =
document.getElementById('language');"); out.println("    if
(langSelect) {"); out.println("
        document.body.setAttribute('lang',
langSelect.value);");
    out.println("    }");
    out.println("}");");
    out.println("</script>");
    out.println("</body>");
    out.println("</html>");
}

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    String theme = request.getParameter("theme");
    String language = request.getParameter("language");
    Cookie themeCookie = new Cookie("theme", theme); Cookie
languageCookie = new Cookie("language", language);
    themeCookie.setMaxAge(60 * 60 * 24 * 365);
    languageCookie.setMaxAge(60 * 60 * 24 * 365);

```

```
        response.addCookie(themeCookie);  
        response.addCookie(languageCookie);  
response.sendRedirect("preferences"); }  
}
```

**OUTPUT:**



**RESULT:**

The experiment successfully demonstrated HTTP cookies as an effective mechanism for maintaining persistent user preferences.