

## EXP 7 Servlet-based project is to store user preferences

**AIM:**

The goal of this Servlet-based project is to store user preferences (such as theme and language) using cookies and retrieve and display them on subsequent visits.

### ALGORITHM:

Step 1: Load Preferences Page → **Browser checks for existing cookies.**

**Step 2 Retrieve Preferences (GET Request) → Servlet reads cookies and applies saved settings.**

**Step 3: User Selects Preferences → User picks theme and language.**

Step 4: Save Preferences (POST Request) → **JavaScript** sends data to **Servlet**.

**Step 5: Servlet Stores Preferences → Cookies are updated and sent back to the browser.**

Step 6: Preferences Applied on Next Visit → **Browser loads saved settings automatically.**

**PROGRAM:**

## index.html (Frontend - Combined HTML, CSS & JavaScript)

```
html
CopyEdit
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Preferences using Cookies</title>
  <style>
    body {
      font-family: 'Poppins', sans-serif;
      background: linear-gradient(to right, #ffafbd, #ffc3a0);
      text-align: center;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
      transition: background 0.5s ease-in-out;
    }

    .container {
      background: white;
      padding: 20px;
      border-radius: 15px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
```

```

        width: 320px;
    }

    h2 {
        color: #ff6f61;
    }

    select, button {
        width: 90%;
        padding: 10px;
        margin: 10px 0;
        border: 1px solid #ff6f61;
        border-radius: 8px;
        text-align: center;
    }

    button {
        background: #ff6f61;
        color: white;
        border: none;
        cursor: pointer;
        transition: 0.3s;
    }

    button:hover {
        background: #e6514a;
    }

    .pref-box {
        margin-top: 20px;
        padding: 10px;
        background: rgba(255, 255, 255, 0.8);
        color: black;
        border-radius: 10px;
        font-weight: bold;
    }

    .hidden {
        display: none;
    }

    .dark-mode {
        background: linear-gradient(to right, #333333, #1e1e1e);
        color: white;
    }

    .dark-mode .container {
        background: rgba(255, 255, 255, 0.1);
        backdrop-filter: blur(10px);
    }

    .dark-mode .pref-box {
        background: rgba(255, 255, 255, 0.2);
        color: white;
    }
}
</style>
</head>
<body>
    <div class="container">
        <h2>Set Your Preferences</h2>
        <div id="form-section">

```

```

        <label for="theme">Choose Theme:</label>
        <select id="theme">
            <option value="light">Light</option>
            <option value="dark">Dark</option>
        </select>

        <label for="language">Choose Language:</label>
        <select id="language">
            <option value="English">English</option>
            <option value="French">French</option>
        </select>

        <button onclick="savePreferences()">Save Preferences</button>
    </div>

    <div id="view-section" class="hidden">
        <h2>Your Preferences</h2>
        <div class="pref-box">
            <p>★ Theme: <span id="saved-theme">-</span></p>
            <p>🌐 Language: <span id="saved-language">-</span></p>
        </div>
        <button onclick="resetPreferences()">Change
Preferences</button>
    </div>
</div>

<script>
    function savePreferences() {
        let theme = document.getElementById("theme").value;
        let language = document.getElementById("language").value;

        fetch("PreferenceServlet", {
            method: "POST",
            headers: { "Content-Type": "application/x-www-form-
urlencoded" },
            body: `theme=${theme}&language=${language}`
        }).then(() => updateView());
    }

    function updateView() {
        fetch("PreferenceServlet")
            .then(response => response.json())
            .then(data => {
                document.getElementById("saved-theme").innerText =
data.theme;
                document.getElementById("saved-language").innerText =
data.language;
                document.body.classList.toggle("dark-mode", data.theme
=== "dark");

                document.getElementById("form-
section").classList.add("hidden");
                document.getElementById("view-
section").classList.remove("hidden");
            });
    }

    function resetPreferences() {
        fetch("PreferenceServlet", { method: "DELETE" }).then(() => {
            document.getElementById("form-
section").classList.remove("hidden");

```

```

        document.getElementById("view-
section").classList.add("hidden");
    });
}

window.onload = updateView;
</script>
</body>
</html>

```

---

## PreferenceServlet.java (Backend - Java Servlet)

```

java
CopyEdit
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.json.JSONObject;

@WebServlet("/PreferenceServlet")
public class PreferenceServlet extends HttpServlet {
    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 * 7); // 1 Week
        languageCookie.setMaxAge(60 * 60 * 24 * 7);

        response.addCookie(themeCookie);
        response.addCookie(languageCookie);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String theme = "light";
        String language = "English";

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

        JSONObject json = new JSONObject();
        json.put("theme", theme);
        json.put("language", language);
    }
}

```

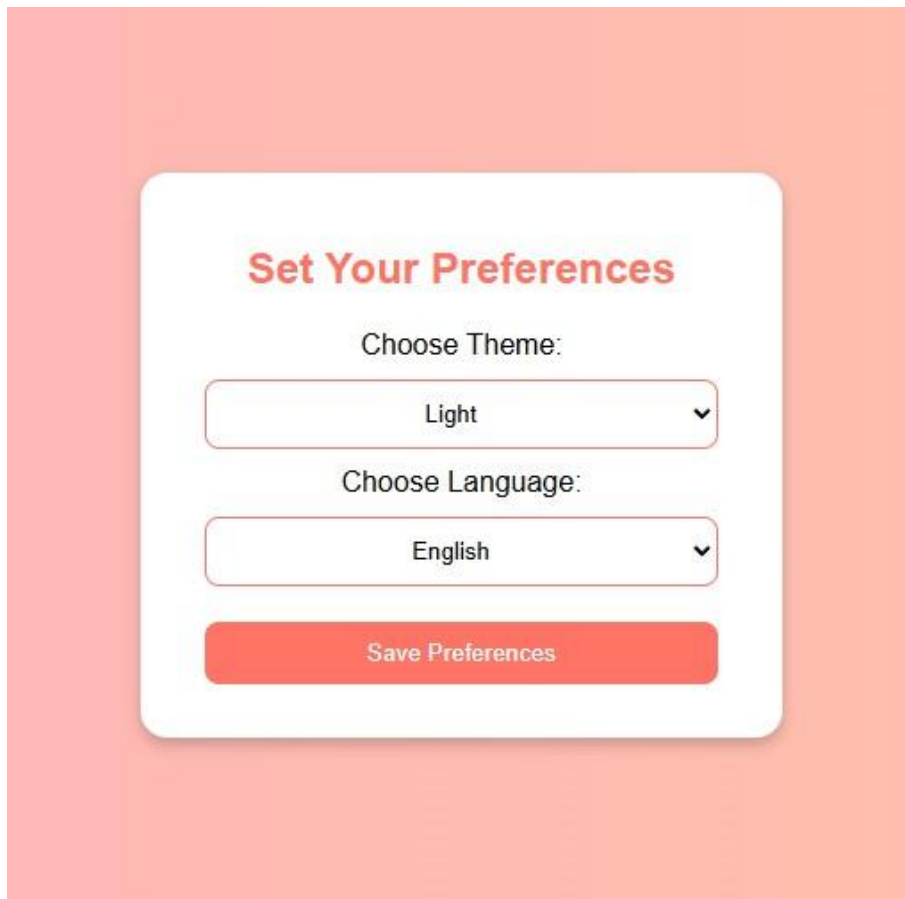
```
        response.setContentType("application/json");
        response.getWriter().write(json.toString());
    }

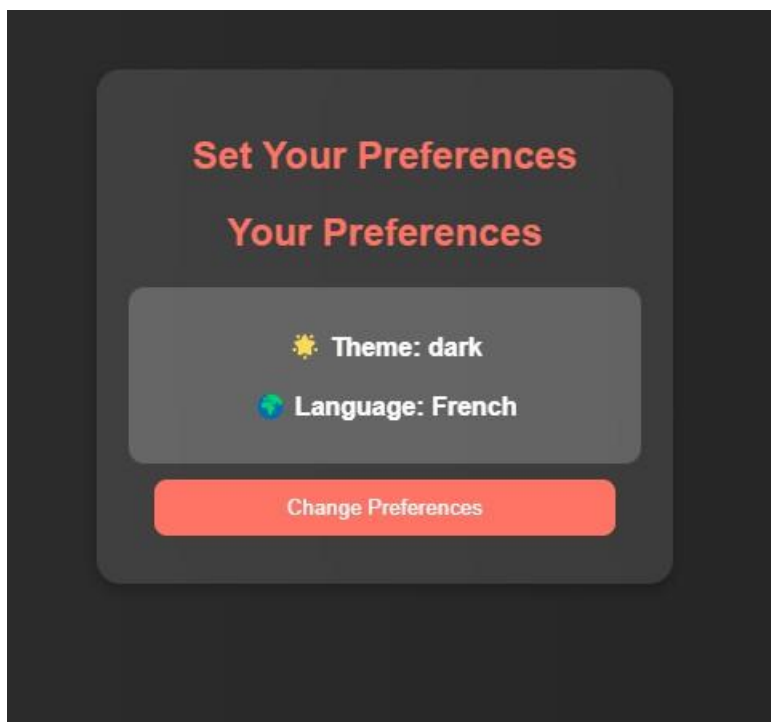
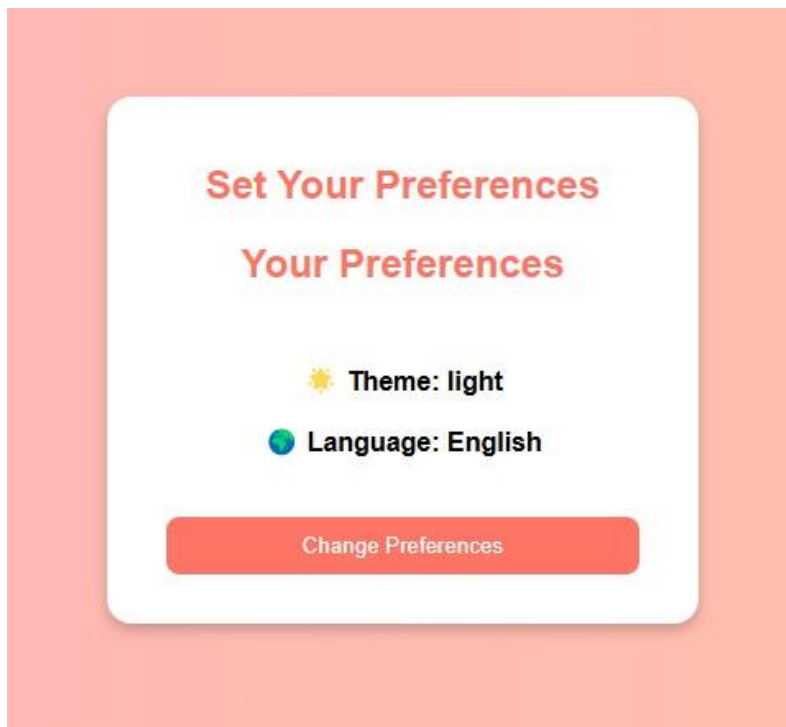
    protected void doDelete(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        Cookie themeCookie = new Cookie("theme", "");
        Cookie languageCookie = new Cookie("language", "");

        themeCookie.setMaxAge(0);
        languageCookie.setMaxAge(0);

        response.addCookie(themeCookie);
        response.addCookie(languageCookie);
    }
}
```

## OUTPUT:





**RESULT:** The Servlet-based project to store user preferences using cookies were executed successfully.