Aim: Building Express Application by Sending and Receiving Cookies

Requirements:

1. **Software & Tools:**
    - Node.js (latest LTS version)
    - Express.js
    - Cookie-parser middleware
    - Web browser (Chrome, Firefox, or Edge)

Theory:

Cookies are small pieces of data stored on the client's browser that help in maintaining user sessions, tracking user activity, and personalizing user experiences.

Types of Cookies:

1. **Session Cookies:** Stored temporarily and deleted when the browser is closed.
2. **Persistent Cookies:** Stored for a specific duration even after the browser is closed.
3. **Secure Cookies:** Transmitted only over HTTPS.
4. **HttpOnly Cookies:** Cannot be accessed via JavaScript for security reasons.

Working with Cookies in Express.js:

- **Setting a Cookie:** The server sends a cookie to the client using the `Set-Cookie` HTTP header.
- **Receiving a Cookie:** The client sends stored cookies to the server with each request.
- **Deleting a Cookie:** The server can clear a cookie by setting an expiration date in the past.

To handle cookies in Express, we use the `cookie-parser` middleware, which simplifies cookie management in request objects.

Code:

Step 1: Install Dependencies

```
Npm install express
npm install express cookie-parser
```

Step 2: Create `server.js` File

```
const express = require("express");
const cookieParser = require("cookie-parser");

const app = express();
const PORT = 3000;
```

```
// Middleware to parse cookies
app.use(cookieParser());

// Route to set a cookie
app.get("/set-cookie", (req, res) => {
    res.cookie("username", "student", { maxAge: 60000, httpOnly: true });
    res.send("Cookie has been set!");
});

// Route to get cookies
app.get("/get-cookie", (req, res) => {
    const cookies = req.cookies;
    res.json({ cookies });
});

// Route to delete a cookie
app.get("/delete-cookie", (req, res) => {
    res.clearCookie("username");
    res.send("Cookie has been deleted!");
});

// Start the server
app.listen(PORT, () => {
    console.log(`Server running at http://localhost:${PORT}`);
});
```

Step 3: Running the Application

```
node server.js
```

Then, open a browser and test the following routes:

- `http://localhost:3000/set-cookie` → Sets a cookie
- `http://localhost:3000/get-cookie` → Retrieves the stored cookie
- `http://localhost:3000/delete-cookie` → Deletes the cookie

Conclusion:

In this experiment, we successfully built an Express.js application that demonstrates cookie management. We used the `cookie-parser` middleware to set, retrieve, and delete cookies. This knowledge is essential for implementing user authentication, session tracking, and personalization in web applications.