EXPERIMENT 8 : LIBRARY MANAGEMENT SYSTEM

AIM :

To develop a JavaScript program that will validate the controls in the forms you have created for the application.

ALGORITHM :

1. **Load Form**
   Display fields for Title, Author, Year, and Genre. Hide success message.
2. **Submit Button Click**
   On submit, prevent default form submission.
3. **Validate Inputs**
   - Check if fields are filled (Title, Author, Year, Genre).
   - Ensure Year is valid (between 1000 and current year).
   - Set `isValid = false` if any field fails validation.
4. **Display Success**
   If `isValid == true`, reset form and show success message.
5. **Keep UX Simple**
   After one submission, allow continuous input without page refresh.

CODE:

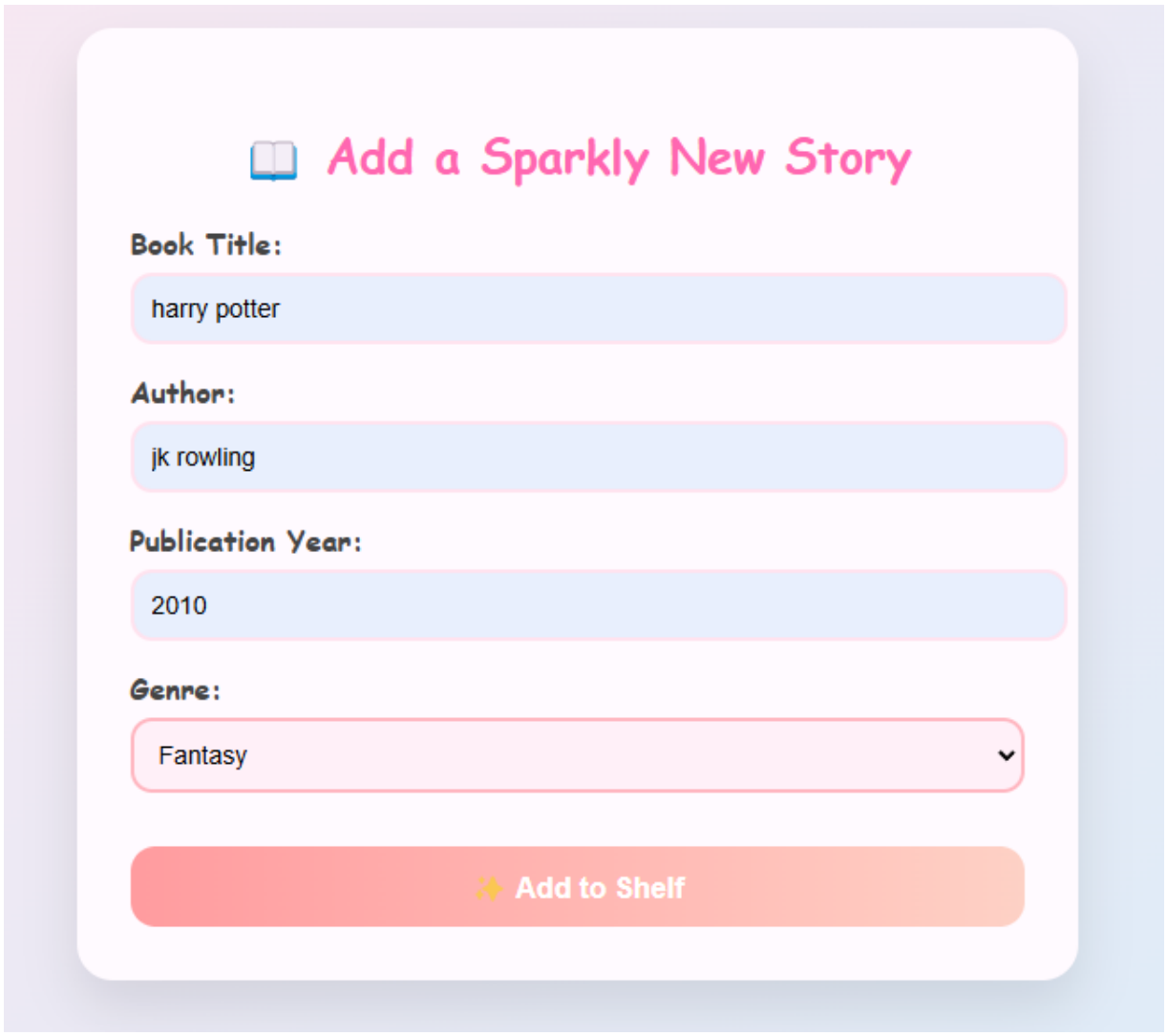
```
<!DOCTYPE html>

<html>

<head>

  <title>Library Book Issue</title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <div class="container">

    <h2>Library Book Issue</h2>

    <form action="BookIssueServlet" method="post">

      <input type="text" name="fullname" placeholder="Full Name" required>

      <input type="email" name="email" placeholder="Email" required>

      <input type="text" name="book" placeholder="Book Title" required>

      <input type="date" name="date" required>

      <button type="submit">Submit</button>
```

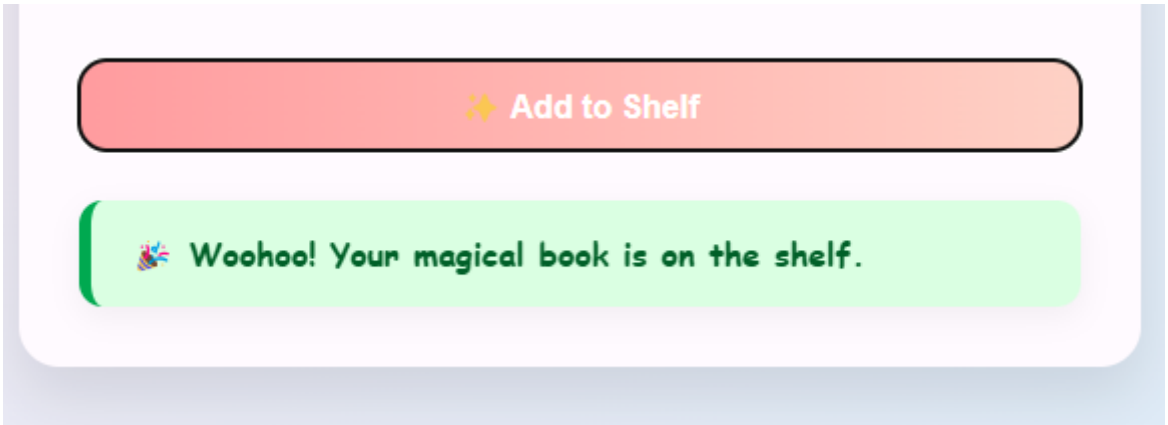
```
    </form>
  </div>
</body>
</html>
body {
  background: linear-gradient(to bottom right, #0f2027, #203a43, #2c5364);
  color: white;
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  background: rgba(0,0,0,0.6);
  padding: 30px;
  border-radius: 15px;
  box-shadow: 0 0 10px cyan;
  text-align: center;
  width: 350px;
}

input, button {
  display: block;
  width: 100%;
  margin: 10px 0;
  padding: 10px;
  border: none;
  border-radius: 8px;
}
```

```
button {
  background: linear-gradient(to right, #ff416c, #ff4b2b);
  color: white;
  font-weight: bold;
  cursor: pointer;
}
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class BookIssueServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String name = request.getParameter("fullname");
        String email = request.getParameter("email");
        String book = request.getParameter("book");
        String date = request.getParameter("date");

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><head><title>Submission</title></head><body style='background-
color:#0f2027; color:white; font-family:Arial;'>");
        out.println("<div style='margin:50px auto; width:400px; background:rgba(0,0,0,0.6);
padding:20px; border-radius:10px;'>");
        out.println("<h2>Submitted Details:</h2>");
        out.println("<p><strong>Name:</strong> " + name + "</p>");
        out.println("<p><strong>Email:</strong> " + email + "</p>");
        out.println("<p><strong>Book:</strong> " + book + "</p>");
        out.println("<p><strong>Date:</strong> " + date + "</p>");
        out.println("</div></body></html>");
```

```
    }
}
<servlet>
    <servlet-name>BookIssueServlet</servlet-name>
    <servlet-class>BookIssueServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>BookIssueServlet</servlet-name>
    <url-pattern>/BookIssueServlet</url-pattern>
</servlet-mapping>
```

OUTPUT:


📖 Add a Sparkly New Story
Book Title:
harry potter
Author:
jk rowling
Publication Year:
2010
Genre:
Fantasy
✨ Add to Shelf


✨ Add to Shelf
🎉 Woohoo! Your magical book is on the shelf.

RESULT :

Thus developed an application for library management system.