EX : 8 Consider a Library Management System. Develop a JavaScript program that will validate the controls in the forms you have created for the application. State the assumptions you make (business logic you are taking into consideration). Note: Your application must access a database using Servlet/JSP.

AIM:

To develop a web-based Library Management System using HTML, CSS, JavaScript for frontend form validation and JSP/Servlet for backend processing. The application will validate form fields before submission and store data in a database.

ALGORITHM:

- Design a form to input book details.
- Use JavaScript to validate input fields:
    - Check for required values.
    - Validate formats.
    - Ensure correct number inputs.
- If validation passes, submit the form to a Java Servlet (`AddBookServlet`).
- The servlet inserts data into the MySQL database.
- Show a success/failure response to the user.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Add Book | Library System</title>
  <style>
    body {
      font-family: 'Segoe UI', sans-serif;
      background: #e0f7fa;
      display: flex;
      justify-content: center;
      align-items: center;
```

```
  height: 100vh;
}

form {
  background: #ffffff;
  padding: 30px;
  border-radius: 15px;
  box-shadow: 0 0 10px rgba(0,0,0,0.2);
  width: 350px;
}

h2 {
  text-align: center;
  color: #00796b;
}

label {
  margin-top: 10px;
  display: block;
  font-weight: 600;
}

input, select {
  width: 100%;
  padding: 8px;
  margin-top: 5px;
  border-radius: 8px;
  border: 1px solid #ccc;
}
```

```
  button {
    margin-top: 20px;
    width: 100%;
    padding: 10px;
    border: none;
    border-radius: 10px;
    background-color: #00796b;
    color: white;
    font-size: 16px;
    cursor: pointer;
  }

  button:hover {
    background-color: #004d40;
  }
</style>

<script>
  function validateForm() {
    const bookId = document.getElementById("bookId").value.trim();
    const title = document.getElementById("title").value.trim();
    const author = document.getElementById("author").value.trim();
    const quantity = document.getElementById("quantity").value.trim();

    if (!/^[a-zA-Z0-9]+$/.test(bookId)) {
      alert("Book ID must be alphanumeric.");
      return false;
    }

    if (title.length < 3) {
```

```
        alert("Title must be at least 3 characters.");
        return false;
      }

      if (author === "") {
        alert("Author is required.");
        return false;
      }

      if (isNaN(quantity) || Number(quantity) <= 0) {
        alert("Quantity must be a positive number.");
        return false;
      }

      return true;
    }
  </script>
</head>

<body>
  <form method="post" action="AddBookServlet" onsubmit="return validateForm()">
    <h2>Add New Book 📚</h2>

    <label for="bookId">Book ID:</label>
    <input type="text" id="bookId" name="bookId" required>

    <label for="title">Title:</label>
    <input type="text" id="title" name="title" required>

    <label for="author">Author:</label>
```

```
    <input type="text" id="author" name="author" required>

    <label for="category">Category:</label>
    <select id="category" name="category" required>
      <option value="">--Select--</option>
      <option value="Fiction">Fiction</option>
      <option value="Science">Science</option>
      <option value="Technology">Technology</option>
      <option value="History">History</option>
    </select>

    <label for="quantity">Quantity:</label>
    <input type="number" id="quantity" name="quantity" required>

    <button type="submit">Add Book</button>
  </form>
</body>
</html>
```

Servlet Code (`AddBookServlet.java`)

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class AddBookServlet extends HttpServlet {
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    response.setContentType("text/html");
```

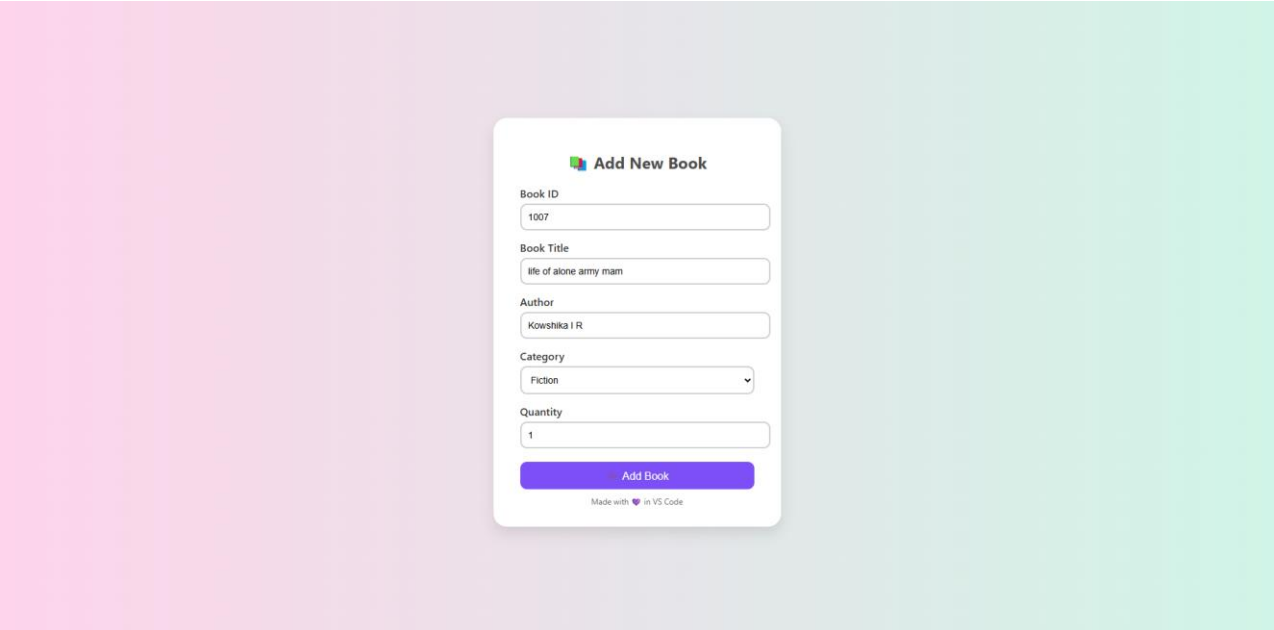
```
    PrintWriter out = response.getWriter();

    String bookId = request.getParameter("bookId");
    String title = request.getParameter("title");
    String author = request.getParameter("author");
    String category = request.getParameter("category");
    int quantity = Integer.parseInt(request.getParameter("quantity"));

    try {
      Class.forName("com.mysql.jdbc.Driver");
      Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/library",
"root", "password");

      PreparedStatement ps = con.prepareStatement("INSERT INTO books VALUES (?, ?, ?,
?, ?)");
      ps.setString(1, bookId);
      ps.setString(2, title);
      ps.setString(3, author);
      ps.setString(4, category);
      ps.setInt(5, quantity);

      int i = ps.executeUpdate();

      if (i > 0) {
        out.println("<h2>Book Added Successfully!</h2>");
      } else {
```

```
        out.println("<h2>Failed to Add Book</h2>");
      }

      con.close();
    } catch (Exception e) {
      out.println("<h2>Error: " + e.getMessage() + "</h2>");
    }
  }
}
```

MySQL Table Structure

```
CREATE DATABASE library;
USE library;

CREATE TABLE books (
  bookId VARCHAR(20) PRIMARY KEY,
  title VARCHAR(100),
  author VARCHAR(100),
  category VARCHAR(50),
  quantity INT
);
```

OUTPUT:



RESULT:

Thus the program is executed successfully.