Let's walk through a **basic servlet example** that is similar to what you'll need for your assignment. This example will demonstrate how to create servlets for handling user interactions in your Online Library Management System.

Example: Handling Book Borrowing Using Servlets

Question:

"Create a servlet-based system for an Online Library Management System that allows users to view a list of books and borrow a selected book. The system should include two servlets: one to display the available books and another to handle the borrowing process. Implement this using Java Servlets in NetBeans IDE, and ensure the system provides feedback to the user on the book they have borrowed."

Deliverables:

1. ViewBooksServlet: Displays a list of books with a "Borrow" link next to each book.
2. BorrowBookServlet: Processes the user's request to borrow a book and provides confirmation.

In this example, we will create two servlets:

1. **ViewBooksServlet** – This servlet will display a list of available books.
2. **BorrowBookServlet** – This servlet will allow users to "borrow" a book by interacting with the list.

Step-by-Step Guide

1. Create ViewBooksServlet

This servlet will display a list of available books to the user.

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/viewbooks")
public class ViewBooksServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
```

```
        // Set the response type to HTML
        response.setContentType("text/html");

        // Simulate book data (in a real application, you might retrieve this from a database)
        String[] books = {"The Alchemist", "1984", "To Kill a Mockingbird", "The Great Gatsby"};

        // Write the response (HTML)
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>Available Books</h1>");
        out.println("<ul>");

        // Display each book with a 'Borrow' link
        for (String book : books) {
            out.println("<li>" + book + " - <a href='borrowbook?book=" + book +
"'>Borrow</a></li>");
        }

        out.println("</ul>");
        out.println("</body></html>");
    }
}
```

2. Create BorrowBookServlet

This servlet will handle the user's request to borrow a book.

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/borrowbook")
public class BorrowBookServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        // Get the 'book' parameter from the URL
        String book = request.getParameter("book");

        // Set the response type to HTML
        response.setContentType("text/html");
```

```
        // Write the response (HTML)
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>Borrowing Book</h1>");

        // Check if the book parameter is present
        if (book != null) {
            out.println("<p>You have borrowed: " + book + "</p>");
        } else {
            out.println("<p>No book selected to borrow.</p>");
        }
        out.println("<a href='viewbooks'>Back to book list</a>");
        out.println("</body></html>");
    }
}
```

How it Works:

1. **ViewBooksServlet** generates an HTML page with a list of books. Each book has a "Borrow" link next to it. The link points to the **BorrowBookServlet** and passes the book name as a URL parameter (e.g., borrowbook?book=The Alchemist).
2. When the user clicks the "Borrow" link, the **BorrowBookServlet** processes the request, retrieves the book name from the URL, and displays a message confirming the book has been borrowed.

Steps to Test the Example:

1. Run your project in **NetBeans** using **Apache Tomcat**.
2. Access the ViewBooksServlet by visiting the URL http://localhost:8080/YourProjectName/viewbooks.
3. Click on the "Borrow" link next to a book. This will take you to BorrowBookServlet, and you'll see a confirmation message showing the book you borrowed.

How This Fits into Your Project:

- In your **Online Library Management System**, you can modify these servlets to handle real book data from a database and manage session tracking to keep track of the books the user has borrowed.
- You will also implement **session tracking** so that a user cannot borrow the same book twice.

Let me know if you need more help or examples!