Develop a JAX-RS client that consumes RESTful service developed in Program-11. Utilize the client in UI layer (JSP pages).

CurrencyConvert.jsp

```
<%@ page import="in.ga.client.CurrencyConverterClient" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
  <title>Currency Converter</title>
</head>
<body>
  <h2>Currency Converter</h2>
  <form method="post">
    From Currency: <input type="text" name="from" required> (e.g., USD, EUR)<br>
    To Currency: <input type="text" name="to" required> (e.g., INR)<br>
    Amount: <input type="number" name="amount" step="0.01" required><br>
    <input type="submit" value="Convert">
  </form>

  <%
    String from = request.getParameter("from");
    String to = request.getParameter("to");
    String amountStr = request.getParameter("amount");

    if (from != null && to != null && amountStr != null) {
      try {
        double amount = Double.parseDouble(amountStr);
        String result = CurrencyConverterClient.convertCurrency(from, to, amount);
        out.println("<h3>" + result + "</h3>");
```

```
        } catch (Exception e) {
            out.println("<h3>Error: Invalid input.</h3>");
        }
    }
%>
</body>
</html>
```

CurrencyConvertorService.java

```
package in.ga.Services;

import jakarta.ws.rs.*;
import jakarta.ws.rs.core.MediaType;
import jakarta.ws.rs.core.Response;
import java.util.HashMap;
import java.util.Map;

@Path("/currency")
public class CurrencyConverterService {

    private static final Map<String, Double> exchangeRates = new HashMap<>();

    static {
        exchangeRates.put("USD_TO_INR", 83.0);
        exchangeRates.put("EUR_TO_INR", 90.0);
    }

    @GET
```

```
    @Path("/convert")
    @Produces(MediaType.APPLICATION_JSON)
    public Response convertCurrency(@QueryParam("from") String from,
                                    @QueryParam("to") String to,
                                    @QueryParam("amount") double amount) {
        String key = from.toUpperCase() + "_TO_" + to.toUpperCase();

        if (!exchangeRates.containsKey(key)) {
            return Response.status(Response.Status.BAD_REQUEST)
                    .entity("Exchange rate for " + key + " not available.")
                    .build();
        }

        double rate = exchangeRates.get(key);
        double convertedAmount = amount * rate;

        Map<String, Object> response = new HashMap<>();
        response.put("from", from);
        response.put("to", to);
        response.put("amount", amount);
        response.put("convertedAmount", convertedAmount);

        return Response.ok(response).build();
    }
}
```

CurrencyConvertorClient.java

```
package in.ga.client;
```

```
import jakarta.ws.rs.client.Client;
import jakarta.ws.rs.client.ClientBuilder;
import jakarta.ws.rs.client.WebTarget;
import jakarta.ws.rs.core.MediaType;
import jakarta.ws.rs.core.Response;
import org.json.JSONObject;

public class CurrencyConverterClient {

    private static final String BASE_URL =
"http://localhost:8080/JerseryClientDemo/ws/currency";

    public static String convertCurrency(String from, String to, double amount) {
        Client client = ClientBuilder.newClient();
        WebTarget target = client.target(BASE_URL)
                .path("convert")
                .queryParam("from", from)
                .queryParam("to", to)
                .queryParam("amount", amount);

        Response response = target.request(MediaType.APPLICATION_JSON).get();

        if (response.getStatus() == Response.Status.OK.getStatusCode()) {
            String jsonResponse = response.readEntity(String.class);
            JSONObject json = new JSONObject(jsonResponse);
            return "Converted Amount: " + json.getDouble("convertedAmount") + " " + to;
        } else {
            return "Error: " + response.readEntity(String.class);
```

```
    }
  }
}
```

Pom.xml

```
<dependency>
        <groupId>org.json</groupId>
        <artifactId>json</artifactId>
        <version>20231013</version>
    </dependency>
```