**RetriveAndParseXml.java:**

package internship;

import java.io.ByteArrayInputStream;

import java.io.ByteArrayOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;

import org.w3c.dom.NodeList;

import org.w3c.dom.Element;

public class RetriveAndParseXml {

private static final String DB\_URL = "jdbc:mysql://localhost:3306/xml\_database";

private static final String DB\_USER = "root";

private static final String DB\_PASSWORD = "Prince@20";

public static void main(String[] args) {

int fileId = 1; // The ID of the XML file to retrieve and parse

try {

String xmlContent = getXMLFile(fileId);

if (xmlContent != null) {

parseXML(xmlContent);

} else {

System.out.println("No XML file found with the provided ID.");

}

} catch (SQLException | IOException e) {

System.err.println("Error retrieving XML file: " + e.getMessage());

e.printStackTrace();

} catch (Exception e) {

System.err.println("Unexpected error: " + e.getMessage());

e.printStackTrace();

}

}

public static String getXMLFile(int fileId) throws SQLException, IOException {

Connection conn = null;

PreparedStatement pstmt = null;

ResultSet rs = null;

ByteArrayOutputStream baos = null;

try {

conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

System.out.println("Database connection established.");

String sql = "SELECT file\_data FROM xml\_files WHERE id = ?";

pstmt = conn.prepareStatement(sql);

pstmt.setInt(1, fileId);

rs = pstmt.executeQuery();

if (rs.next()) {

System.out.println("Data found for the provided ID.");

InputStream is = rs.getBinaryStream("file\_data");

baos = new ByteArrayOutputStream();

byte[] buffer = new byte[1024];

int bytesRead;

while ((bytesRead = is.read(buffer)) != -1) {

baos.write(buffer, 0, bytesRead);

}

String xmlContent = baos.toString("UTF-8");

System.out.println("Retrieved XML Content: " + xmlContent); // Debug statement

return xmlContent;

} else {

System.out.println("No data found for the provided ID."); // Debug statement

}

} catch (SQLException e) {

System.err.println("SQL Error: " + e.getMessage());

throw e;

} catch (IOException e) {

System.err.println("IO Error: " + e.getMessage());

throw e;

} finally {

if (rs != null) rs.close();

if (pstmt != null) pstmt.close();

if (conn != null) conn.close();

if (baos != null) baos.close();

System.out.println("Database connection closed.");

}

return null;

}

public static void parseXML(String xmlContent) {

try {

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder = factory.newDocumentBuilder();

ByteArrayInputStream input = new ByteArrayInputStream(xmlContent.getBytes("UTF-8"));

Document doc = builder.parse(input);

doc.getDocumentElement().normalize();

System.out.println("Root element: " + doc.getDocumentElement().getNodeName()); // Debug statement

NodeList nodeList = doc.getElementsByTagName("element");

if (nodeList.getLength() == 0) {

System.out.println("No elements found with the tag name 'element'."); // Debug statement

}

for (int i = 0; i < nodeList.getLength(); i++) {

Element element = (Element) nodeList.item(i);

System.out.println("Element Value: " + element.getTextContent());

}

} catch (Exception e) {

System.err.println("XML Parsing Error: " + e.getMessage());

e.printStackTrace();

}

}

}