

MOBILE SPECIFICATIONS WEBSITE

V. SAI ABHINAV REDDY : CB.EN.U4CSE15358
C. SANJAY : CB.EN.U4CSE15311
ANIL KUMAR : CB.EN.U4CSE15336

ABSTRACT:

The project is a web application which has the details of various mobile brands and different mobile models in each brand. It shows detailed specifications of each mobile model. A brand can be chosen from the list and it displays all the models of that brand. On selecting a model from the list, it displays the detailed specifications of that model. User can compare the specifications of two different mobile models. A search box will be present in which the user can search for a particular brand or a particular model to view the details. User can register in the webpage either through Google mail or directly and upon logging in user can post comments and reviews of any mobile model and rate it on the scale of 5. User will have his dedicated page where he/she can update or modify his profile, view all his comments and reviews.

INTRODUCTION:

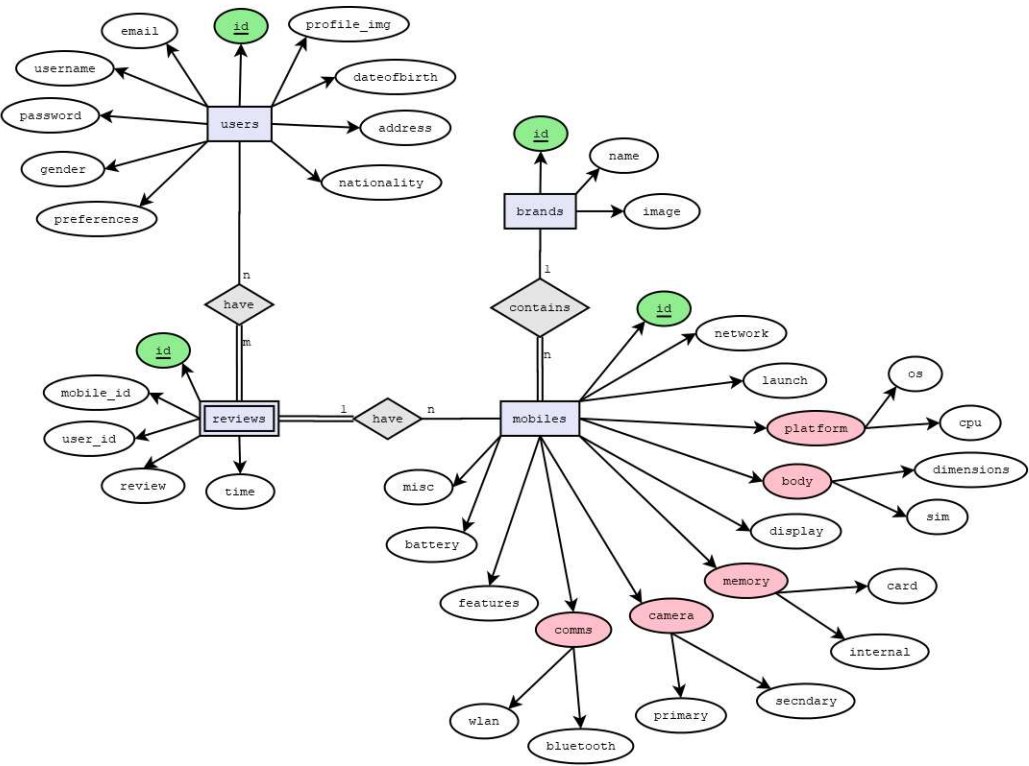
There is a huge boost in the smart phones market in the recent years and with increasing number of people using smart phones, everyone wants to know the specifications, pros and cons, comments and reviews, compare with another mobile before buying the product.

With a web application showing all the detailed specifications of various mobile brands and models, it can be made easy for the users to decide what mobile model they can buy.

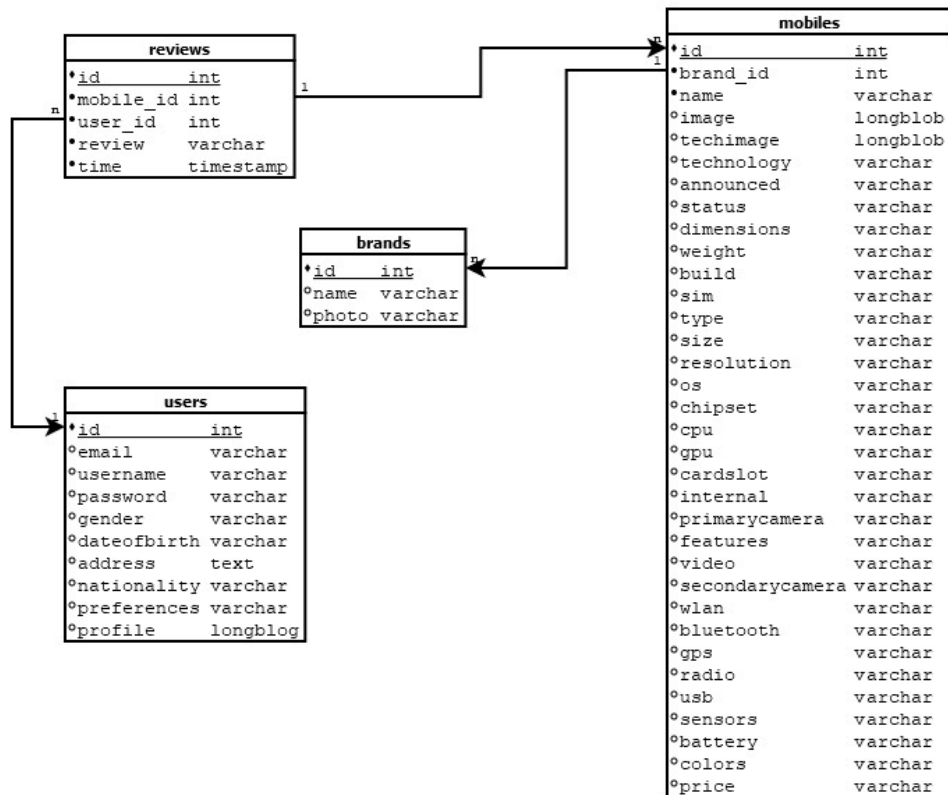
This project will have the features like commenting on different mobile brands and models, compare between different mobiles. All these features can be used on creating an account and logging in into the web application.

There will be a search box where user can enter a brand name or mobile name and view the specifications of that mobile directly.

ER-DIAGRAM:



SCHEMA:



Users Table:

Contains the username, passwords, emails, etc., details about the registered users.

- **Id (int):** Unique identification for a user. (primary key, auto generated)
- **Email (varchar):** Valid email address of the user.
- **Username (varchar):** Used to login into his/her account.
- **Preferences (varchar):** User can select some mobile brands as their preferences.

Brands Table:

Contains the names of all the available brands and their images.

- **Id (int):** Unique identification number of the brand. (primary key, auto generated)

Mobiles Table:

Contains the specifications like sim type, camera, price, colors available, etc., of a mobile model.

- **Id (int):** Unique identification number of the mobile model. (primary key, auto generated)
- **Brand_id (int):** Brand id for the corresponding brand. (foreign key to the brands table)
- **Other specifications (varchar & text):** Specifications of the mobile model.

Reviews Table:

Contains the reviews on a mobile model made by the user.

- **Id (int):** Unique identification number of the review. (primary key, auto generated)
- **Mobile_id (int):** Mobile id of the corresponding mobile model. (foreign key to the mobiles table)
- **User_id (int):** User id of the corresponding user. (foreign key to the users table)
- **Review (varchar):** Review for the mobile given by the user.
- **Time (timestamp):** Timestamp for when the review is given. (auto generated)

TECHNOLOGIES:

XML:

- Store and validate data in xml using dtd or schema.
- Display xml data in html page using javascript.
- Dynamic generation of xml page for a registered user and validate it using schema and upload it to database.
- Generate xml page with the details of the logged in user.

JAVA SERVELETS:

- To generate dynamic xml pages with the corresponding data.
- Upload the data from xml page to the database.

SOURCE CODES:

REVIEWS PAGE

MOBI SPEC ⓘ List Details				
<div><div><div>Users </div><div>Users </div></div><div><div>Brands </div><div>Brands </div></div><div><div>Reviews </div><div>Reviews </div></div></div>				
Id	Reviewed On	Reviewed By	Review	Time
1	iPhone X	abhinav	testing review	2018-07-30 09:43:43
2	iPhone 8	chaitu	another testing review	2018-07-30 09:46:48
4	Galaxy On 6	abhinav	lol	2018-08-04 15:21:31
5	iPhone X	abhinav		2018-08-04 15:54:38
6	iPhone X	abhinav	afasd	2018-08-04 15:54:44
7	Galaxy On 6	chaitu	this is a very cool mobile in less budget	2018-08-04 16:05:50
8	Galaxy On 6	sanjay	chaitu you are true	2018-08-04 16:11:27
9	Galaxy On 6	sanjay	but there is a catch to it!!!	2018-08-04 16:12:06
10	Galaxy On 6	arvind	Wow ! c>3	2018-08-04 16:32:33
11	Galaxy On 6	chaitu	its worst phone and hanging too much!	2018-08-04 23:08:40
12	Galaxy S9 Plus	chaitu	excellent phone	2018-08-04 23:10:52

REVIEWS PAGE: (XML)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<!DOCTYPE reviews SYSTEM "reviewsdata.dtd">
```

```
<reviews>
```

```
  <review>
```

```
    <id>1</id>
```

```
    <mobile>iPhone X</mobile>
```

```
    <user>abhinav</user>
```

```
    <text>testing review</text>
```

```
    <time>2018-07-30 09:43:43</time>
```

```
  </review>
```

```
  <review>
```

```
    <id>2</id>
```

```
    <mobile>iPhone 8</mobile>
```

```
    <user>chaitu</user>
```

```
    <text>another testing review</text>
```

```
    <time>2018-07-30 09:46:48</time>
```

```
  </review>
```

```
  <review>
```

```
    <id>4</id>
```

```
    <mobile>Galaxy On 6</mobile>
```

```
    <user>abhinav</user>
```

```
    <text>lol</text>
```


<time>2018-08-04 15:21:31</time>

</review>

</reviews>

REVIEWS PAGE (DTD):

<?xml version='1.0' encoding='UTF-8'?>

<!ELEMENT reviews (review)*>

<!ELEMENT review (id,mobile,user,text,time)*>

<!ELEMENT id (#PCDATA)>

<!ELEMENT mobile (#PCDATA)>

<!ELEMENT user (#PCDATA)>

<!ELEMENT text (#PCDATA)>

<!ELEMENT time (#PCDATA)>

REVIEWS PAGE (XSL):

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:output method="html"/>

<xsl:template match="/">

<html>

<body>

<table class="table table-hover" border="0">

<thead class="bg-dark text-white">

<tr>

<th>Id</th>

<th>Reviewed On</th>

<th>Reviewed By</th>

<th>Review</th>

<th>Time</th>

</tr>

</thead>

<tbody>

<xsl:for-each select="reviews/review">

<tr>

<td>

<xsl:value-of select="id"/>

</td>

<td>

<xsl:value-of select="mobile"/>

</td>

<td>

<xsl:value-of select="user"/>

</td>

<td>

<xsl:value-of select="text"/>

</td>

<td>

<xsl:value-of select="time"/>

```

        </td>

    </tr>

</xsl:for-each>

</tbody>

</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>
```

USERS PAGE:

MOBI SPEC

List Details

Search

Users

Users

Brands

Brands

Reviews

Reviews

Email	Username	Gender	DOB	Address	Preferences
saiabhinavvasepalli@gmail.com	abhinav	male	1997-09-28	D.No: 7-5-92, Opp. Nirmal Concept School, Nirmal Nagar	SAMSUNG,LENOVO,APPLE,ZTE,GOOGLE
dkchaitanyavarma06@gmail.com	chaitu	male	2018-07-13	2-20, 1st floor, Lakshmipuram colony, Poranki, Tadigadapa, Krishna Dist	APPLE,ONEPLUS
raviteja9597@gmail.com	raviteja	male	2018-07-11	D.No: 7-5-92, Opp. Nirmal Concept School, Nirmal-Nagar	SAMSUNG,APPLE,ONEPLUS
sriharsha@gmail.com	sriharsha	male	2018-07-21	2-20, 1st floor, Lakshmipuram colony, Poranki, Tadigadapa, Krishna Dist	APPLE,ONEPLUS
sanjay99@gmail.com	sanjay	male	2018-07-18	D.No: 7-5-92, Opp. Nirmal Concept School, Nirmal Nagar	APPLE,ONEPLUS
arvindsai8198@gmail.com	arvind	male	2018-07-02	2-20, 1st floor, Lakshmipuram colony, Poranki, Tadigadapa, Krishna Dist	APPLE,ONEPLUS
pandurangareddy@gmail.com	pandu	male	2018-	lol city	SAMSUNG,LENOVO,ALCATEL,APPLE,MOTORO

USERS PAGE (XML):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<users                                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="registeruserschema.xsd">

  <user>

    <email>aruna@gmail.com</email>

    <username>aruna</username>

    <password>aruna</password>

    <gender>female</gender>

    <dob>2018-08-30</dob>

    <address>D.No: 7-5-92, Opp. Nirmal Concept School, Nirmal Nagar</address>

    <preferences>APPLE,XIAOMI</preferences>

    <nation>india</nation>

  </user>

</users>
```

USERS PAGE (XSD):

```
<?xml version="1.0"?>

<xs:schema          version="1.0"          xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">

  <xs:element name="users">

    <xs:complexType>

      <xs:sequence>

        <xs:element ref="user" minOccurs="1" maxOccurs="unbounded"/>

      </xs:sequence>

    </xs:complexType>

  </xs:element>

</xs:schema>
```

```

    </xs:sequence>

  </xs:complexType>

</xs:element>

<xs:element name="user">

  <xs:complexType>

    <xs:sequence>

      <xs:element name="email" minOccurs='1' maxOccurs='1'>

        <xs:simpleType>

          <xs:restriction base='xs:string'>

            <xs:pattern value="([a-zA-Z0-9][\w.]*@(gmail|yahoo|outlook)\.com)"/>

          </xs:restriction>

        </xs:simpleType>

      </xs:element>

      <xs:element name="username" minOccurs='1' maxOccurs='1'>

        <xs:simpleType>

          <xs:restriction base='xs:string'>

            <xs:pattern value="([a-zA-Z0-9])+"/>

          </xs:restriction>

        </xs:simpleType>

      </xs:element>

      <xs:element name="password" minOccurs='1' maxOccurs='1'>

        <xs:simpleType>

```

```

        <xs:restriction base='xs:string'>

            <xs:minLength          value="((?=[A-Z])(?=[!@#$$&*])(?=[0-9])(?=[a-
z]).{8,})"/>

        </xs:restriction>

    </xs:simpleType>

</xs:element>

<xs:element name="gender" minOccurs='1' maxOccurs='1'>

    <xs:simpleType>

        <xs:restriction base="xs:string">

            <xs:enumeration value="male" />

            <xs:enumeration value="female" />

        </xs:restriction>

    </xs:simpleType>

</xs:element>

<xs:element name="dob" minOccurs='1' maxOccurs='1'>

    <xs:simpleType>

        <xs:restriction base="xs:string">

            <xs:pattern          value="((((19|[2-9]\d)\d{2})\-(0[13578]|1[02])\-(0[1-
9]|[12]\d|3[01]))|(((19|[2-9]\d)\d{2})\-(0[13456789]|1[012])\-(0[1-9]|[12]\d|30))|(((19|[2-
9]\d)\d{2})\-(02\-(0[1-9]|1\d|2[0-8]))|(((1[6-9]|[2-
9]\d)(0[48]|[2468][048]|[13579][26])|((16|[2468][048]|[3579][26])00))\-(02\-(29)))" />

        </xs:restriction>

    </xs:simpleType>

</xs:element>

```

```
<xs:element name="address" minOccurs="1" maxOccurs="1" type="xs:string" />
```

```
<xs:element name="preferences" minOccurs="1" maxOccurs="1">
```

```
  <xs:simpleType>
```

```
    <xs:restriction base="xs:string">
```

```
      <xs:enumeration value="SAMSUNG"/>
```

```
      <xs:enumeration value="LENOVO"/>
```

```
      <xs:enumeration value="ALCATEL"/>
```

```
      <xs:enumeration value="WIKO"/>
```

```
      <xs:enumeration value="APPLE"/>
```

```
      <xs:enumeration value="XIAOMI"/>
```

```
      <xs:enumeration value="ZTE"/>
```

```
      <xs:enumeration value="VIVO"/>
```

```
      <xs:enumeration value="NOKIA"/>
```

```
      <xs:enumeration value="GOOGLE"/>
```

```
      <xs:enumeration value="TOSHIBA"/>
```

```
      <xs:enumeration value="LEECO"/>
```

```
      <xs:enumeration value="SONY"/>
```

```
      <xs:enumeration value="ACER"/>
```

```
      <xs:enumeration value="VODAFONE"/>
```

```
      <xs:enumeration value="PANASONIC"/>
```

```
      <xs:enumeration value="LG"/>
```

```
      <xs:enumeration value="ASUS"/>
```

```
      <xs:enumeration value="ENERGIZER"/>
```

```
<xs:enumeration value="HP"/>

<xs:enumeration value="HTC"/>

<xs:enumeration value="OPPO"/>

<xs:enumeration value="XOLO"/>

<xs:enumeration value="YU"/>

<xs:enumeration value="MOTOROLA"/>

<xs:enumeration value="ONEPLUS"/>

<xs:enumeration value="LAVA"/>

<xs:enumeration value="VERYKOOL"/>

<xs:enumeration value="HUAWEI"/>

<xs:enumeration value="MEIZU"/>

<xs:enumeration value="MICROMAX"/>

<xs:enumeration value="MAXWEST"/>

<xs:enumeration value="MICROSOFT"/>

<xs:enumeration value="BLACKBERRY"/>

<xs:enumeration value="BLU"/>

<xs:enumeration value="PLUM"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="nation" minOccurs='1' maxOccurs='1'>

  <xs:simpleType>

    <xs:restriction base="xs:string">
```



```

        <xs:enumeration value="india" />

        <xs:enumeration value="other" />

    </xs:restriction>

</xs:simpleType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

```

USERS PAGE (XSL):

```

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:output method="html"/>

    <xsl:template match="/">

        <html>

            <body>

                <table class="table table-hover" border="0">

                    <thead class="bg-dark text-white">

                        <tr>

                            <th>Email</th>

                            <th>Username</th>

                            <th>Gender</th>

                            <th>DOB</th>

```

<th>Address</th>

<th>Preferences</th>

<th>Nationality</th>

</tr>

</thead>

<tbody>

<xsl:for-each select="users/user">

<tr>

<td>

<xsl:value-of select="email"/>

</td>

<td>

<xsl:value-of select="username"/>

</td>

<td>

<xsl:value-of select="gender"/>

</td>

<td>

<xsl:value-of select="dob"/>

</td>

<td>

<xsl:value-of select="address"/>

</td>

```
        <td>

            <xsl:value-of select="preferences"/>

        </td>

        <td>

            <xsl:value-of select="nation"/>

        </td>

    </tr>

</xsl:for-each>

</tbody>

</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>
```

USERS PAGE (HTML):

```
<html>

<head>

<style>

    ::-webkit-scrollbar {

        display: none;

    }

</style>

<script>
```

```
function loadXMLDoc(filename) { if (window.ActiveXObject) { xhttp = new
ActiveXObject("Msxml2.XMLHTTP"); } else { xhttp = new XMLHttpRequest(); } xhttp.open("GET",
filename, false); try { xhttp.responseType = "msxml-document" } catch (err) { } // Helping IE11
xhttp.send(""); return xhttp.responseXML; }
```

```
function displayResult() { xml = loadXMLDoc("userdata.xml"); xsl = loadXMLDoc("userdata.xsl"); //
code for IE if (window.ActiveXObject || xhttp.responseType == "msxml-document") { ex =
xml.transformNode(xsl); document.getElementById("example").innerHTML = ex; } // code for
Chrome, Firefox, Opera, etc. else if (document.implementation &&
document.implementation.createDocument) { xsltProcessor = new XSLTProcessor();
xsltProcessor.importStylesheet(xsl); resultDocument = xsltProcessor.transformToFragment(xml,
document); document.getElementById("example").appendChild(resultDocument); }
```

```
</script>
```

```
</head>
```

```
<body onload="displayResult()">
```

```
    <div id="example"></div>
```

```
</body>
```

```
</html>
```

USERS PAGE (SERVELET):

```
/*
```

```
 * To change this license header, choose License Headers in Project Properties.
```

```
 * To change this template file, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
package xmlqueries;
```

```
import java.io.File;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.*;

import java.sql.DriverManager;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.transform.Transformer;

import javax.xml.transform.TransformerFactory;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.Document;

import org.w3c.dom.Element;
```

```
/**
```

```
*
```

```
* @author saiab
```

```
*/
```

```

public class getUsers extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */

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

        /*response.setContentType("text/html;charset=UTF-8");

        try (PrintWriter out = response.getWriter()) {

            out.println("<!DOCTYPE html>");

            out.println("<html>");

            out.println("<head>");

            out.println("<title>Servlet getUsers</title>");

            out.println("</head>");

            out.println("<body>");

            out.println("<h1>Servlet getUsers at " + request.getContextPath() + "</h1>");

            out.println("</body>");

```

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

    }*/

}
```

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

```
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
```

```

* @param response servlet response

* @throws ServletException if a servlet-specific error occurs

* @throws IOException if an I/O error occurs

*/

```

```
@Override
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
    throws ServletException, IOException {
```

```
    processRequest(request, response);
```

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

```
    try {
```

```
        Class.forName("com.mysql.jdbc.Driver");
```

```
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mobispec?"
"user=root&password=");
```

```
        Statement st = con.createStatement();
```

```
        String user = request.getParameter("ttr");
```

```
        ResultSet rs = null;
```

```
        if(user.equals("users")) {
```

```
            rs = st.executeQuery("SELECT * FROM users");
```

```
        }
```

```
    // creating document
```

```
    DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
```



```
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

Document doc = dBuilder.newDocument();


// root element

Element rootElement = doc.createElement("users");

doc.appendChild(rootElement);


while(rs.next()) {

    // user element

    Element userElement = doc.createElement("user");

    rootElement.appendChild(userElement);

    // email element

    Element emailElement = doc.createElement("email");

    emailElement.appendChild(doc.createTextNode(rs.getString("email")));

    userElement.appendChild(emailElement);

    // username element

    Element usernameElement = doc.createElement("username");

    usernameElement.appendChild(doc.createTextNode(rs.getString("username")));

    userElement.appendChild(usernameElement);

    // password element

    Element passwordElement = doc.createElement("password");

    passwordElement.appendChild(doc.createTextNode(rs.getString("password")));

    userElement.appendChild(passwordElement);

}
```

```
// gender element

Element genderElement = doc.createElement("gender");

genderElement.appendChild(doc.createTextNode(rs.getString("gender")));

userElement.appendChild(genderElement);

// dateofbirth element

Element dobElement = doc.createElement("dob");

dobElement.appendChild(doc.createTextNode(rs.getString("dateofbirth")));

userElement.appendChild(dobElement);

// address element

Element addressElement = doc.createElement("address");

addressElement.appendChild(doc.createTextNode(rs.getString("address")));

userElement.appendChild(addressElement);

// preferences element

Element preferencesElement = doc.createElement("preferences");

preferencesElement.appendChild(doc.createTextNode(rs.getString("preferences")));

userElement.appendChild(preferencesElement);

// nation element

Element nationElement = doc.createElement("nation");

nationElement.appendChild(doc.createTextNode(rs.getString("nationality")));

userElement.appendChild(nationElement);

}

// writing into xml file
```

```

        TransformerFactory tFactory = TransformerFactory.newInstance();

        Transformer tFormer = tFactory.newTransformer();

        DOMSource source = new DOMSource(doc);

        StreamResult      result      =      new      StreamResult(new
File("F:\\NetCentric\\mobi_spec\\web\\admin\\users\\userdata.xml"));

        tFormer.transform(source, result);

    }catch(Exception e) {

        out.println(e);

    }

}

/**

 * Returns a short description of the servlet.

 *

 * @return a String containing servlet description

 */

@Override

public String getServletInfo() {

    return "Short description";

}

}

```

REGISTRAION PAGE XSD:

```
<?xml version="1.0"?>

<xs:schema      version="1.0"      xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">

  <xs:element name="users">

    <xs:complexType>

      <xs:sequence>

        <xs:element ref="user" minOccurs='1' maxOccurs='unbounded' />

      </xs:sequence>

    </xs:complexType>

  </xs:element>

  <xs:element name="user">

    <xs:complexType>

      <xs:sequence>

        <xs:element name="email" minOccurs='1' maxOccurs='1'>

          <xs:simpleType>

            <xs:restriction base='xs:string'>

              <xs:pattern value="([a-zA-Z0-9][\w.]*@(gmail|yahoo|outlook)\.com)" />

            </xs:restriction>

          </xs:simpleType>

        </xs:element>

        <xs:element name="username" minOccurs='1' maxOccurs='1'>
```

```

<xs:simpleType>

  <xs:restriction base='xs:string'>

    <xs:pattern value="([a-zA-Z0-9])+"/>

  </xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="password" minOccurs='1' maxOccurs='1'>

  <xs:simpleType>

    <xs:restriction base='xs:string'>

      <xs:minLength value="((?=.*[A-Z])(?=.*[!@#$%*])(?=.*[0-9])(?=.*[a-
z]).{8,})"/>

    </xs:restriction>

  </xs:simpleType>

</xs:element>

<xs:element name="gender" minOccurs='1' maxOccurs='1'>

  <xs:simpleType>

    <xs:restriction base="xs:string">

      <xs:enumeration value="male" />

      <xs:enumeration value="female" />

    </xs:restriction>

  </xs:simpleType>

</xs:element>

<xs:element name="dob" minOccurs='1' maxOccurs='1'>

  <xs:simpleType>

```

```

<xs:restriction base="xs:string">

    <xs:pattern          value="((((19|[2-9]\d)\d{2})\-(0[13578]|1[02])\-(0[1-
9]|[12]\d|3[01]))|((((19|[2-9]\d)\d{2})\-(0[13456789]|1[012])\-(0[1-9]|[12]\d|30))|((((19|[2-
9]\d)\d{2})\-(02\-(0[1-9]|1\d|2[0-8]))|(((1[6-9]|[2-
9]\d)(0[48]|[2468][048]|[13579][26])|((16|[2468][048]|[3579][26])00))\-(02\-(29)))" />

</xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="address" minOccurs="1" maxOccurs="1" type="xs:string" />

<xs:element name="preferences" minOccurs="1" maxOccurs="1">

    <xs:simpleType>

        <xs:restriction base="xs:string">

            <xs:enumeration value="SAMSUNG"/>

            <xs:enumeration value="LENOVO"/>

            <xs:enumeration value="ALCATEL"/>

            <xs:enumeration value="WIKO"/>

            <xs:enumeration value="APPLE"/>

            <xs:enumeration value="XIAOMI"/>

            <xs:enumeration value="ZTE"/>

            <xs:enumeration value="VIVO"/>

            <xs:enumeration value="NOKIA"/>

            <xs:enumeration value="GOOGLE"/>

            <xs:enumeration value="TOSHIBA"/>

            <xs:enumeration value="LEECO"/>

```

<xs:enumeration value="SONY"/>
<xs:enumeration value="ACER"/>
<xs:enumeration value="VODAFONE"/>
<xs:enumeration value="PANASONIC"/>
<xs:enumeration value="LG"/>
<xs:enumeration value="ASUS"/>
<xs:enumeration value="ENERGIZER"/>
<xs:enumeration value="HP"/>
<xs:enumeration value="HTC"/>
<xs:enumeration value="OPPO"/>
<xs:enumeration value="XOLO"/>
<xs:enumeration value="YU"/>
<xs:enumeration value="MOTOROLA"/>
<xs:enumeration value="ONEPLUS"/>
<xs:enumeration value="LAVA"/>
<xs:enumeration value="VERYKOOL"/>
<xs:enumeration value="HUAWEI"/>
<xs:enumeration value="MEIZU"/>
<xs:enumeration value="MICROMAX"/>
<xs:enumeration value="MAXWEST"/>
<xs:enumeration value="MICROSOFT"/>
<xs:enumeration value="BLACKBERRY"/>
<xs:enumeration value="BLU"/>

```
        <xs:enumeration value="PLUM"/>

    </xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="nation" minOccurs="1" maxOccurs="1">

    <xs:simpleType>

        <xs:restriction base="xs:string">

            <xs:enumeration value="india" />

            <xs:enumeration value="other" />

        </xs:restriction>

    </xs:simpleType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>
```


TEST CASES FOR THE FORM INPUT FOR NEW USER REGISTRATION:

VALID:

Email	:	saiabhinavvasepalli@gmail.com
Username	:	Abhinav
Password	:	Abhi@1234
Gender	:	male female
Date of Birth	:	Until todays date
Address	:	D.No: 7-5-92, Opp. Nirmal Concept School, Nirmal Nagar
Preferences	:	From the dropdown
Nationality	:	India Other

INVALID:

Email	:	saiabhinavvasepalli@test.com
Username	:	@#\$sdafsd
Password	:	Abhinav
Gender	:	Anything other than male female
Date of Birth	:	Any future dates
Address	:	Nil
Preferences	:	Anything besides from the dropdown
Nationality	:	Anything besides from the dropdown

TEST CASES FOR THE FORM INPUT IN REVIEWS:

VALID:

Review	:	Awesome mobile! @lol
--------	---	----------------------

INVALID:

Review	:	Awesome mobile! <>%%
--------	---	----------------------