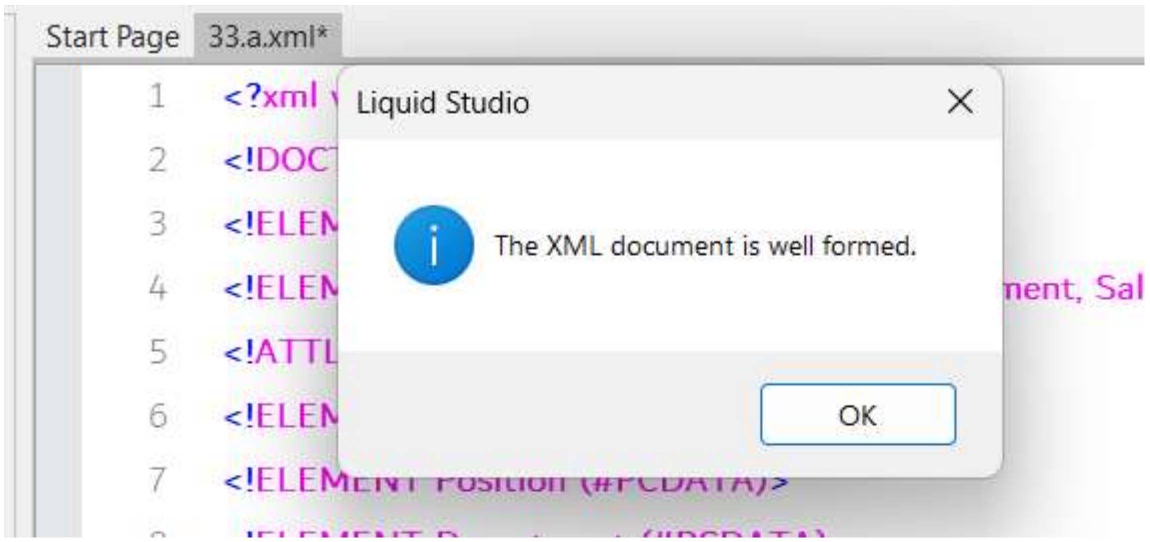
33.a

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Employees [
<!ELEMENT Employees (Employee+)>
<!ELEMENT Employee (Name, Position, Department, Salary, ContactDetails)>
<!ATTLIST Employee id CDATA #REQUIRED>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Position (#PCDATA)>
<!ELEMENT Department (#PCDATA)>
<!ELEMENT Salary (#PCDATA)>
<!ATTLIST Salary currency CDATA #IMPLIED>
<!ELEMENT ContactDetails (Email, Phone)>
<!ELEMENT Email (#PCDATA)>
<!ELEMENT Phone (#PCDATA)>
]>

<Employees>
    <Employee id="100">
        <Name>John Doe</Name>
        <Position>Accountant</Position>
        <Department>Finance</Department>
        <Salary currency="USD">60000</Salary>
        <ContactDetails>
            <Email>john@gmail.com</Email>
            <Phone>+9779800000000000</Phone>
        </ContactDetails>
    </Employee>
    <Employee id="101">
        <Name>Jane Smith</Name>
        <Position>Software Engineer</Position>
        <Department>IT</Department>
        <Salary>80000</Salary>
        <ContactDetails>
            <Email></Email>
            <Phone></Phone>
        </ContactDetails>
    </Employee>
</Employees>
```



Start Page | 33.a.xml*

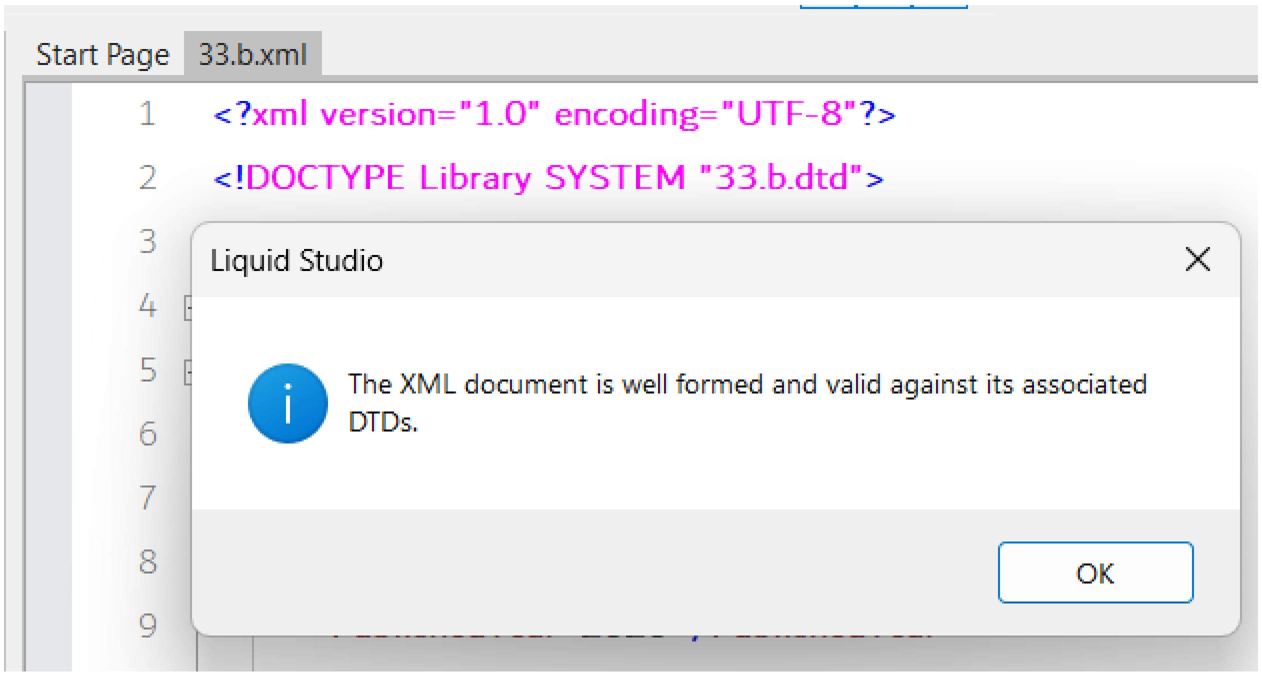
```
<?xml v
<!DOCT
<!ELEM
<!ELEM                                                    ment, Sal
<!ATTL
<!ELEM
<!ELEMENT Position (#PCDATA)>
```

Liquid Studio

The XML document is well formed.

33.b.dtd

```
<!ELEMENT Library (Book+)>
<!ELEMENT Book (Title, Author+, Genre, PublishedYear, Price, Copies)>
<!ATTLIST Book isbn CDATA #REQUIRED>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Author (#PCDATA)>
<!ELEMENT Genre (#PCDATA)>
<!ELEMENT PublishedYear (#PCDATA)>
<!ELEMENT Price (#PCDATA)>
<!ATTLIST Price currency CDATA #IMPLIED>
<!ELEMENT Copies (Copy+)>
<!ELEMENT Copy (Availability)>
<!ATTLIST Copy copyID ID #REQUIRED>
<!ELEMENT Availability EMPTY>
<!ATTLIST Availability status (Available | Checked_Out) #REQUIRED>
```



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Library SYSTEM "33.b.dtd">
```

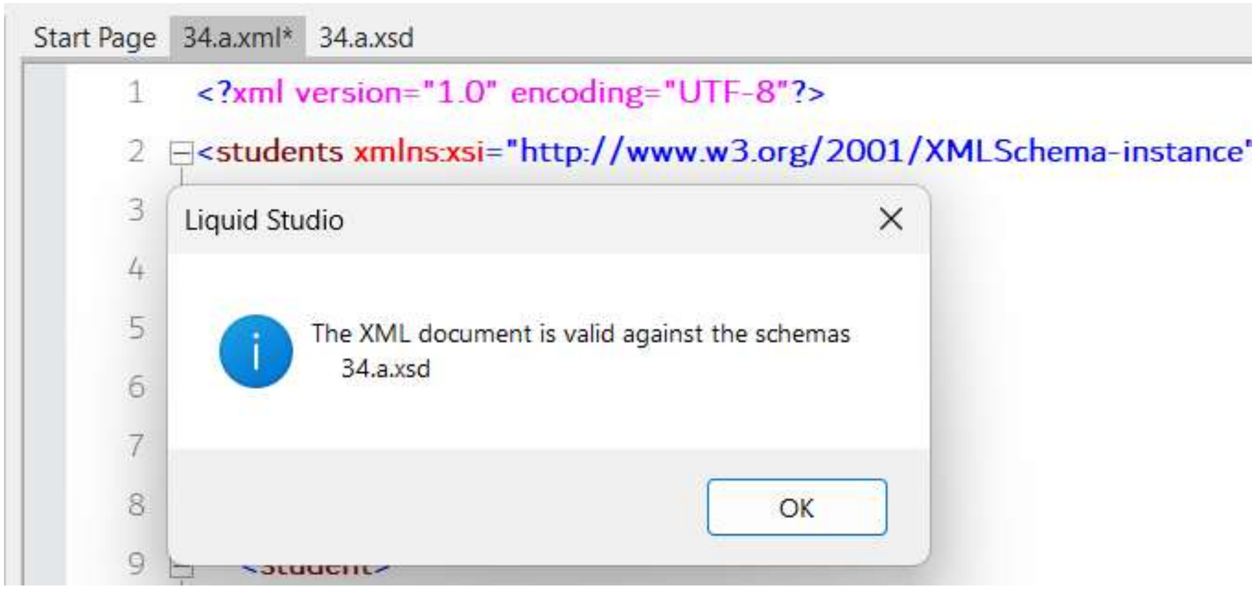
The XML document is well formed and valid against its associated DTDs.

33.b.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Library SYSTEM "33.b.dtd">

<Library>
    <Book isbn="978-3-16-148410-0">
        <Title>The Great Library</Title>
        <Author>John Doe</Author>
        <Genre>Fiction</Genre>
        <PublishedYear>2020</PublishedYear>
        <Price currency="USD">25.99</Price>
        <Copies>
            <Copy copyID="v1">
                <Availability status="Available"/>
            </Copy>
            <Copy copyID="v2">
                <Availability status="Checked_Out"/>
            </Copy>
        </Copies>
    </Book>
    <Book isbn="978-1-40-289462-6">
        <Title>Data Structures</Title>
        <Author>Jane Smith</Author>
        <Genre>Education</Genre>
        <PublishedYear>2018</PublishedYear>
        <Price>45.00</Price>
        <Copies>
            <Copy copyID="v3">
                <Availability status="Available"/>
            </Copy>
            <Copy copyID="v4">
                <Availability status="Available"/>
            </Copy>
        </Copies>
    </Book>
</Library>
```

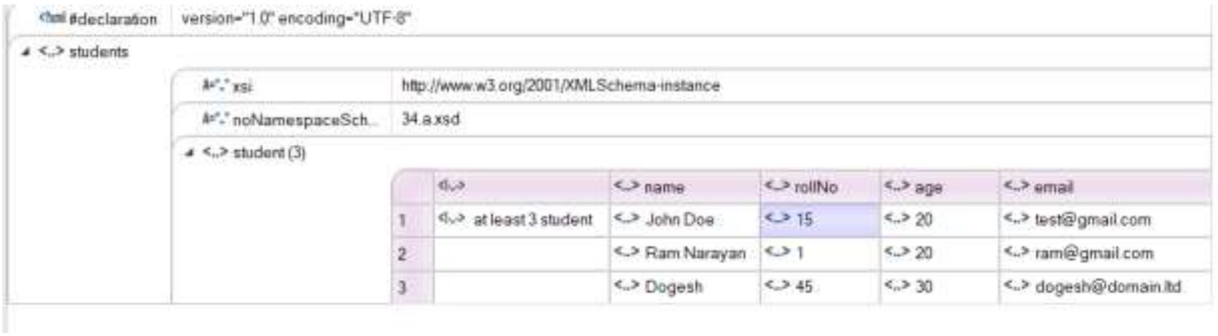
34.a.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<students xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="34.a.xsd">
   <student>  <!-- at least 3 student -->
      <name>John Doe</name>
      <rollNo>15</rollNo>
      <age>20</age>
      <email>test@gmail.com</email>
   </student>
   <student>
      <name>Ram Narayan</name>
      <rollNo>1</rollNo>
      <age>20</age>
      <email>ram@gmail.com</email>
   </student>
   <student>
      <name>Dogesh</name>
      <rollNo>45</rollNo>
      <age>30</age>
      <email>dogesh@domain.ltd</email>
   </student>
</students>
```



```
<?xml version="1.0" encoding="UTF-8"?>
<students xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

The XML document is valid against the schemas
34.a.xsd



#declaration version="1.0" encoding="UTF-8"

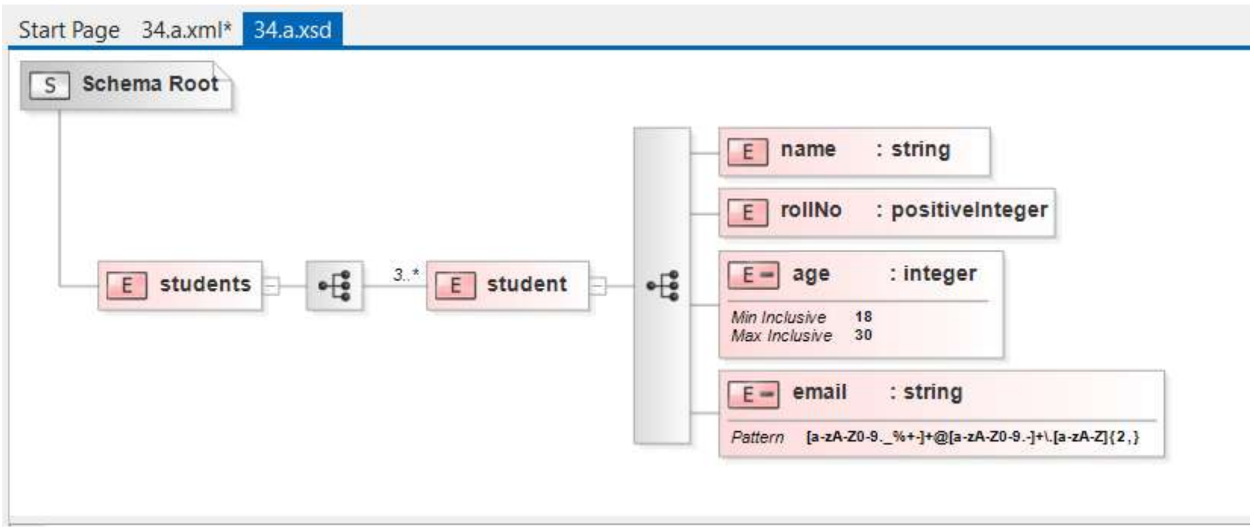
students

xsi http://www.w3.org/2001/XMLSchema-instance

noNamespaceSch... 34.a.xsd

student (3)

		name	rollNo	age	email
1	at least 3 student	John Doe	15	20	test@gmail.com
2		Ram Narayan	1	20	ram@gmail.com
3		Dogesh	45	30	dogesh@domain.ltd



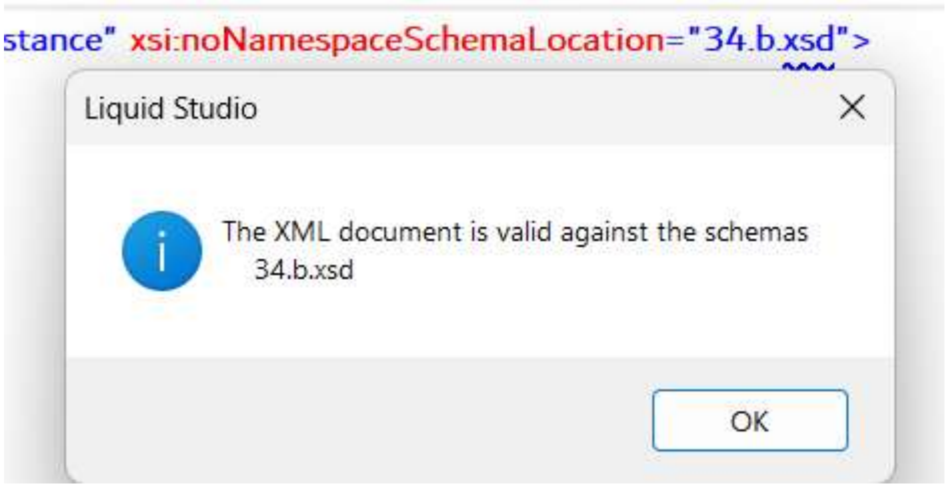
34.a.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs ="http://www.w3.org/2001/XMLSchema">
    <xs:element name="students">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="student" maxOccurs="unbounded" minOccurs="3">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="name" type="xs:string"/>
                            <xs:element name="rollNo" type="xs:positiveInteger"/>
                            <xs:element name="age">
                                <xs:simpleType>
                                    <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="18"/>
                                        <xs:maxInclusive value="30"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="email">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:pattern value="[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-
Z]{2,}"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>

                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

34.b.xml

```
<!-- <?xml version="1.0" encoding="UTF-8"?> -->
<books xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="34.b.xsd">
   <book>
      <title>The Great Gatsby</title>
      <author>F. Scott Fitzgerald</author>
      <ISBN>9780743273565</ISBN>
      <price>10.99</price>
      <publisher>
         <name>Scribner</name>
         <yearEstablished>1998</yearEstablished>
      </publisher>
   </book>
   <book>
      <title>1984</title>
      <author>George Orwell</author>
      <ISBN>9780451524935</ISBN>
      <price>8.99</price>
      <publisher>
         <name>Plume</name>
         <yearEstablished>2000</yearEstablished>
      </publisher>
   </book>
   <book>
      <title>To Kill a Mockingbird</title>
      <author>Harper Lee</author>
      <ISBN>9780060935467</ISBN>
      <price>7.99</price>
      <publisher>
         <name>Harper Perennial</name>
         <yearEstablished>2005</yearEstablished>
      </publisher>
   </book>
   <book>
      <title>Pride and Prejudice</title>
      <author>Jane Austen</author>
      <ISBN>9780141439518</ISBN>
      <price>9.99</price>
```

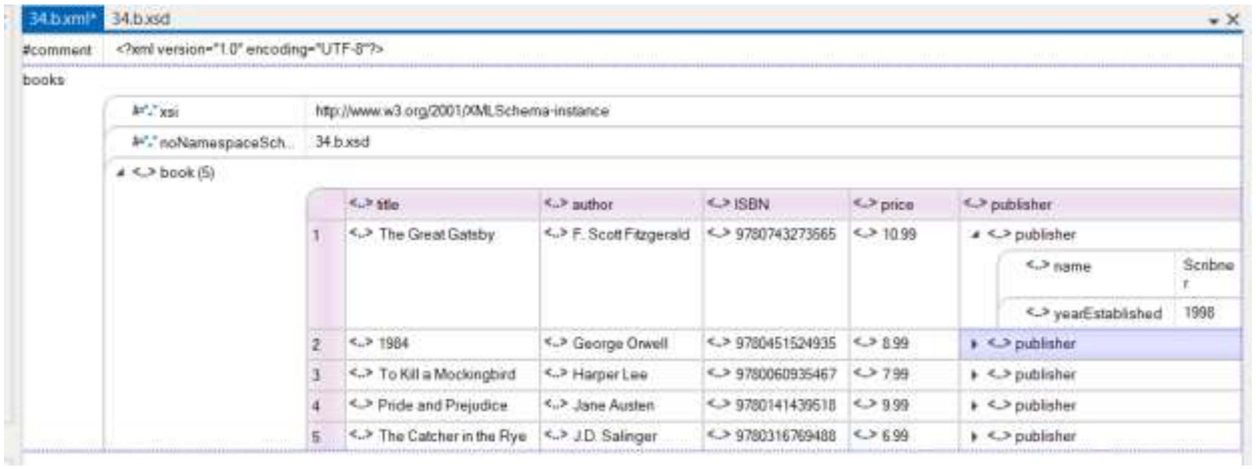
```
        <publisher>
            <name>Penguin Classics</name>
            <yearEstablished>2010</yearEstablished>
        </publisher>
    </book>
    <book>
        <title>The Catcher in the Rye</title>
        <author>J.D. Salinger</author>
        <ISBN>9780316769488</ISBN>
        <price>6.99</price>
        <publisher>
            <name>Little, Brown and Company</name>
            <yearEstablished>2001</yearEstablished>
        </publisher>
    </book>
</books>
```



```
stance" xsi:noNamespaceSchemaLocation="34.b.xsd">
```

Liquid Studio

The XML document is valid against the schemas
34.b.xsd



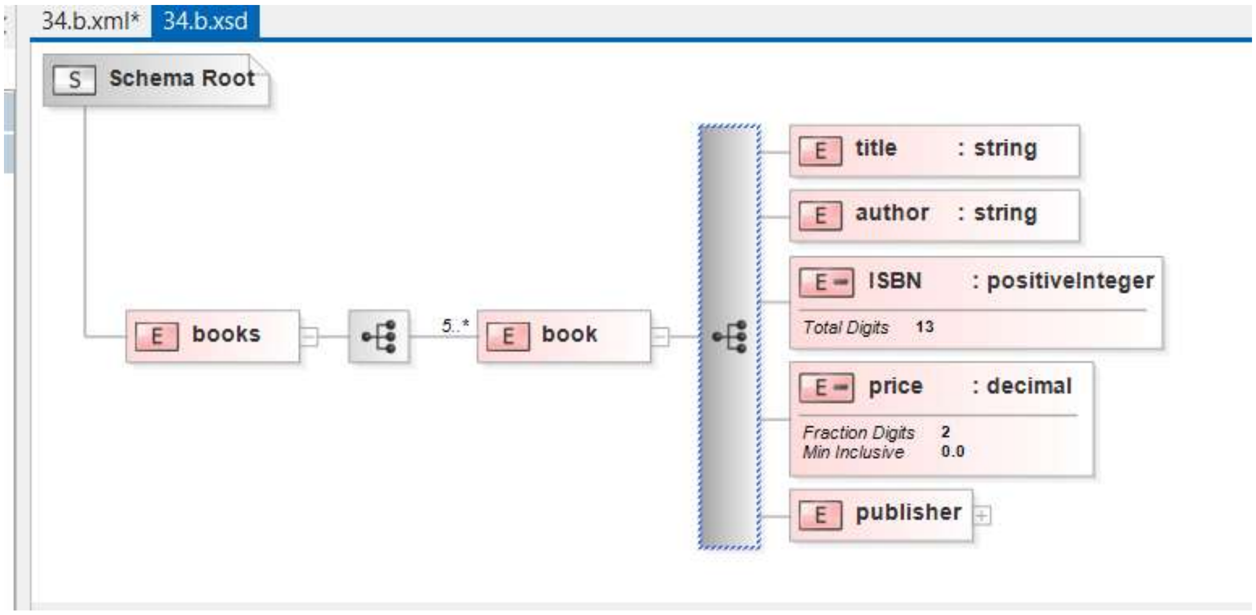
#comment	<?xml version="1.0" encoding="UTF-8"?>
books	
xsi	http://www.w3.org/2001/XMLSchema-instance
noNamespaceSch...	34.b.xsd
book (5)	

	title	author	ISBN	price	publisher
1	The Great Gatsby	F. Scott Fitzgerald	9780743273565	10.99	publisher: name Scribne r; yearEstablished 1998
2	1984	George Orwell	9780451524935	8.99	publisher
3	To Kill a Mockingbird	Harper Lee	9780060935467	7.99	publisher
4	Pride and Prejudice	Jane Austen	9780141439518	9.99	publisher
5	The Catcher in the Rye	J.D. Salinger	9780316769488	6.99	publisher

34.b.xsd

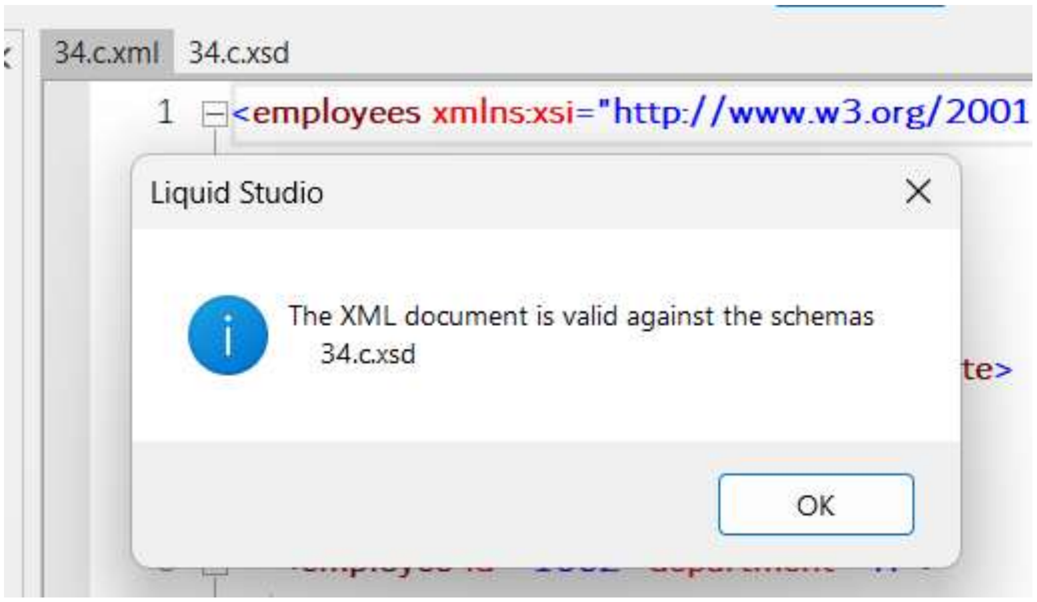
```
<!-- <?xml version="1.0" encoding="UTF-8"?>   -->
<xs:schema xmlns:xs= "http://www.w3.org/2001/XMLSchema">
    <xs:element name="books">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="book" minOccurs="5" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="title" type="xs:string"/>
                            <xs:element name="author" type="xs:string"/>
                            <xs:element name="ISBN">
                                <xs:simpleType>
                                    <xs:restriction base="xs:positiveInteger">
                                        <xs:totalDigits value="13"/> <!-- exactly 13 digits long--
>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="price">
                                <xs:simpleType>
                                    <xs:restriction base="xs:decimal">
                                        <xs:minInclusive value="0.0"/> <!-- price must be a
positive number -->
                                        <xs:fractionDigits value="2"></xs:fractionDigits>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="publisher">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="name" type="xs:string"/>
                                        <xs:element name="yearEstablished">
                                            <xs:simpleType>
                                                <xs:restriction base="xs:gYear">
                                                    <xs:minInclusive value="1990"/> <!-- year must
be 1990 or later -->
                                                    <xs:maxInclusive value="2025"/> <!-- year must
be 2025 or earlier -->
```

```
                                    </xs:restriction>
                                  </xs:simpleType>
                                </xs:element>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:schema>
```


34.b.xml*
34.b.xsd
S Schema Root
E books
5..*
E book
E title : string
E author : string
E ISBN : positiveInteger
Total Digits 13
E price : decimal
Fraction Digits 2
Min Inclusive 0.0
E publisher

34.c.xml

```
<employees xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="34.c.xsd">
   <employee id="1001" department="HR">
      <name>Alice Johnson</name>
      <salary>55000</salary>
      <joiningDate>2018-04-15</joiningDate>
   </employee>

   <employee id="1002" department="IT">
      <name>Bob Smith</name>
      <salary>72000</salary>
      <joiningDate>2019-09-01</joiningDate>
   </employee>

   <employee id="1003" department="Finance">
      <name>Clara Williams</name>
      <salary>68000</salary>
      <joiningDate>2020-01-20</joiningDate>
   </employee>
</employees>
```

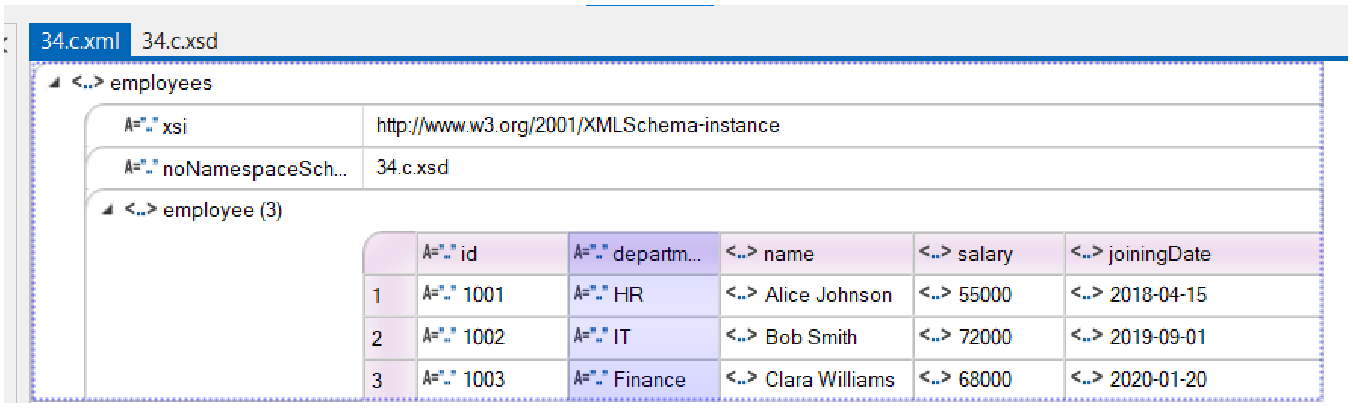


```
<employees xmlns:xsi="http://www.w3.org/2001
```

The XML document is valid against the schemas
34.c.xsd

34.c.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="employees">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="employee" minOccurs="3" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="name" type="xs:string"/>
              <xs:element name="salary">
                <xs:simpleType>
                  <xs:restriction base="xs:decimal">
                    <xs:minInclusive value="0.0"/>  <!--positive decimal no-->
                    <xs:fractionDigits value="2"/>  <!-- upto 2 decimal digits-->
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="joiningDate">
                <xs:simpleType>
                  <xs:restriction base="xs:date"/>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="id" type="xs:positiveInteger" use="required"/>
            <xs:attribute name="department">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="HR"/>
                  <xs:enumeration value="IT"/>
                  <xs:enumeration value="Finance"/>
                  <xs:enumeration value="Sales"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



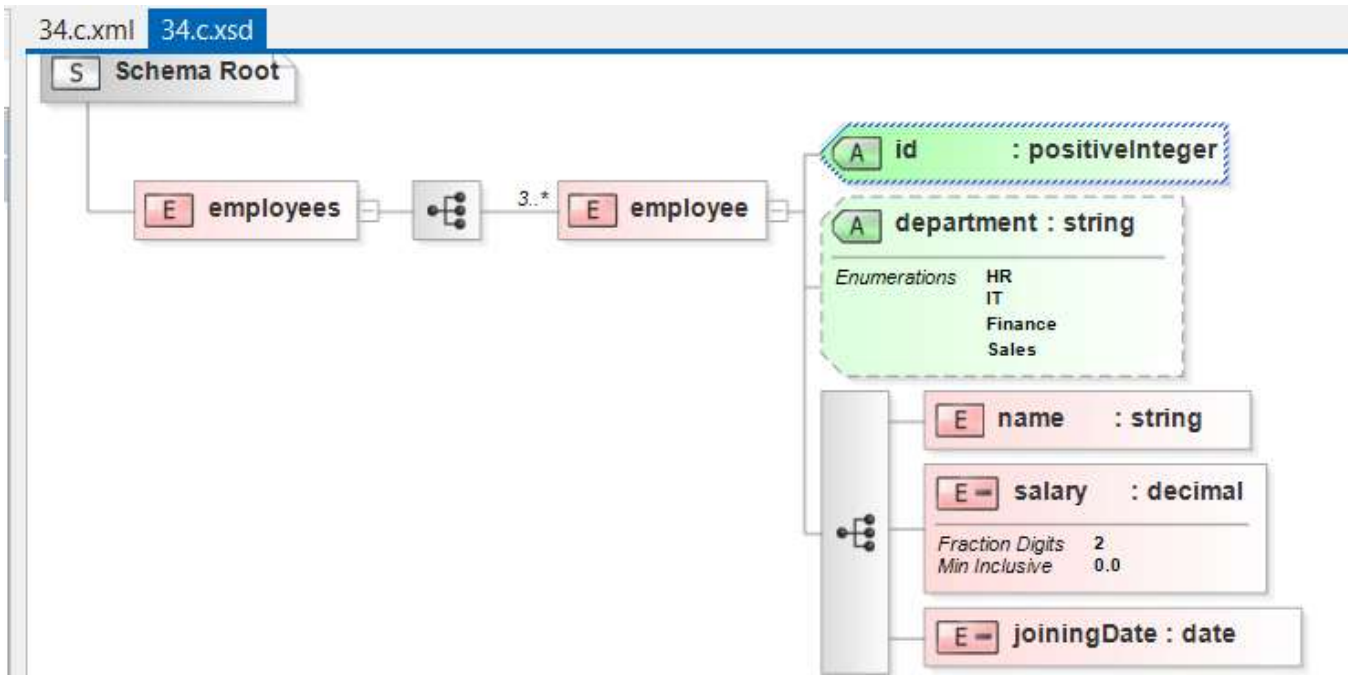
34.c.xml 34.c.xsd

<..> employees

A="." xsi	http://www.w3.org/2001/XMLSchema-instance
A="." noNamespaceSch...	34.c.xsd

<..> employee (3)

	A="." id	A="." departm...	<..> name	<..> salary	<..> joiningDate
1	A="." 1001	A="." HR	<..> Alice Johnson	<..> 55000	<..> 2018-04-15
2	A="." 1002	A="." IT	<..> Bob Smith	<..> 72000	<..> 2019-09-01
3	A="." 1003	A="." Finance	<..> Clara Williams	<..> 68000	<..> 2020-01-20



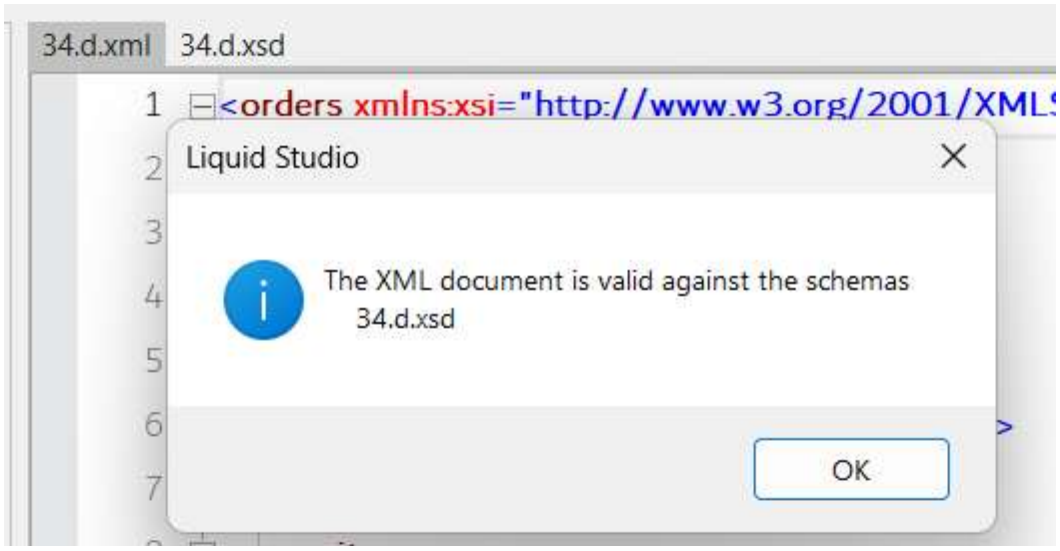
34.d.xml

```
<orders xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="34.d.xsd">
  <order>
    <customer>
      <name>John Doe</name>
      <phone>9800000012</phone>
      <email>john.doe@example.com</email>
    </customer>
    <items>
      <item>
        <itemName>Laptop</itemName>
        <quantity>1</quantity>
        <price>1200.50</price>
      </item>
      <item>
        <itemName>Mouse</itemName>
        <quantity>2</quantity>
        <price>25.75</price>
      </item>
    </items>
  </order>

  <order>
    <customer>
      <name>Jane Smith</name>
      <phone>9876543210</phone>
      <email>jane.smith@example.com</email>
    </customer>
    <items>
      <item>
        <itemName>Smartphone</itemName>
        <quantity>1</quantity>
        <price>799.99</price>
      </item>
      <item>
        <itemName>Headphones</itemName>
        <quantity>1</quantity>
        <price>89.90</price>
      </item>
    </items>   </order> </orders>
```

34.d.xsd

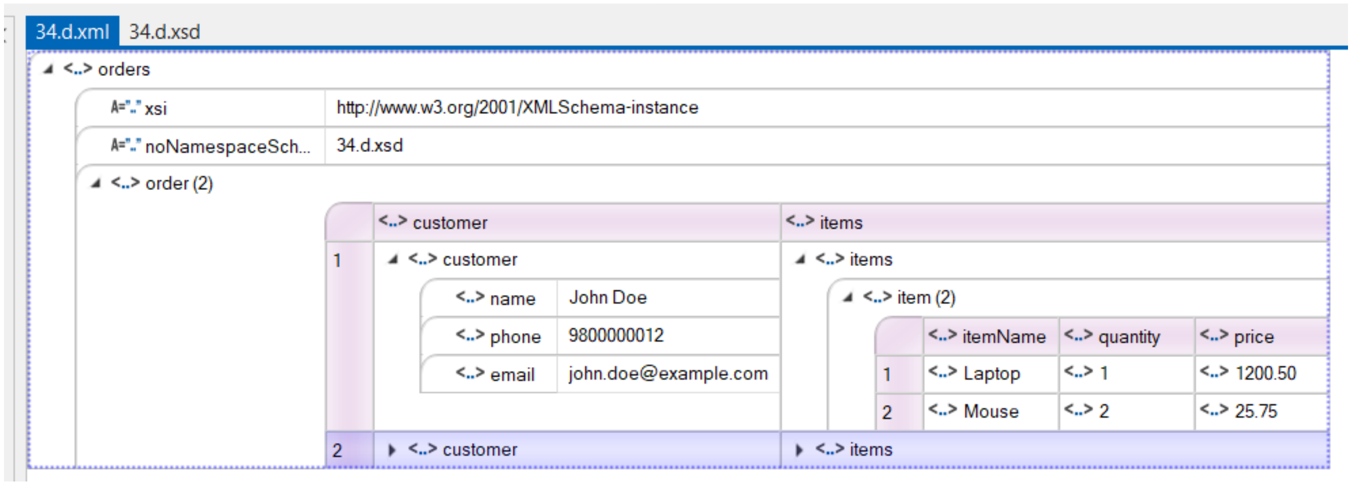
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="orders">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="order" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="customer">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="name" type="xs:string"/>
                    <xs:element name="phone">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:pattern value="9[7-8]\d{8}"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element name="email">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:pattern value="[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-
]+\.[a-zA-Z]{2,}"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="items">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="item" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="itemName" type="xs:string"/>
                          <xs:element name="quantity"
type="xs:positiveInteger"/>
                          <xs:element name="price">
                            <xs:simpleType>
```

```
                                                <xs:restriction base="xs:decimal">
                                                    <xs:minExclusive value="0.0"/>
                                                    <xs:fractionDigits value="2"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```



```
<orders xmlns:xsi="http://www.w3.org/2001/XML
```

The XML document is valid against the schemas
34.d.xsd



34.d.xml 34.d.xsd

◢ <..> orders

A="..." xsi	http://www.w3.org/2001/XMLSchema-instance
A="..." noNamespaceSch...	34.d.xsd

◢ <..> order (2)

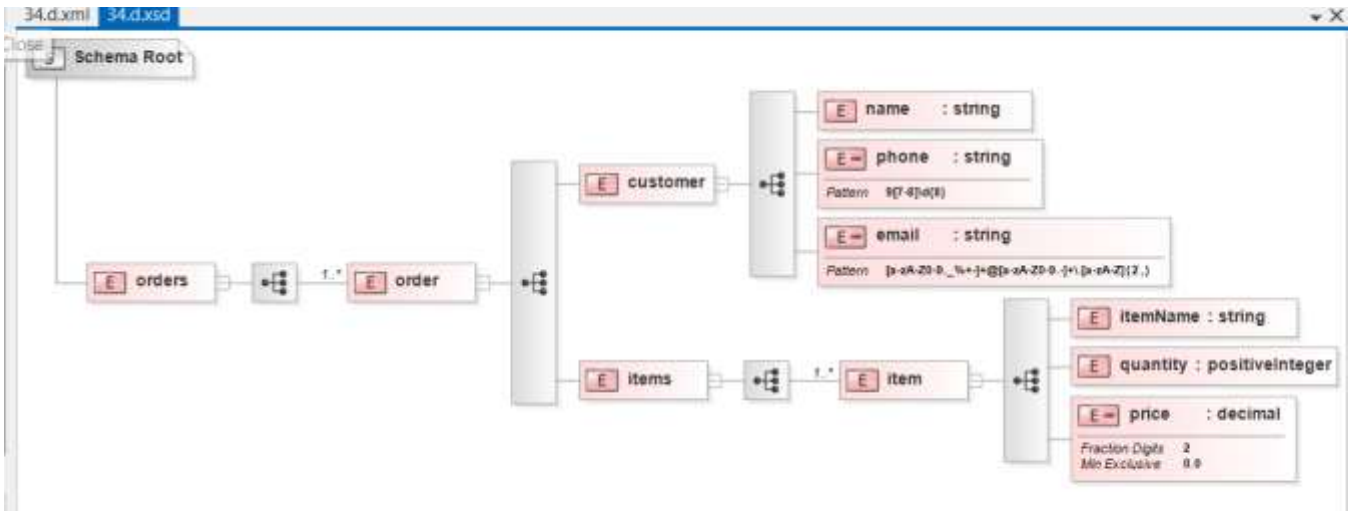
	<..> customer	<..> items
1	◢ <..> customer	◢ <..> items
2	▸ <..> customer	▸ <..> items

Order 1, customer:

<..> name	John Doe
<..> phone	9800000012
<..> email	john.doe@example.com

Order 1, items: ◢ <..> item (2)

	<..> itemName	<..> quantity	<..> price
1	<..> Laptop	<..> 1	<..> 1200.50
2	<..> Mouse	<..> 2	<..> 25.75

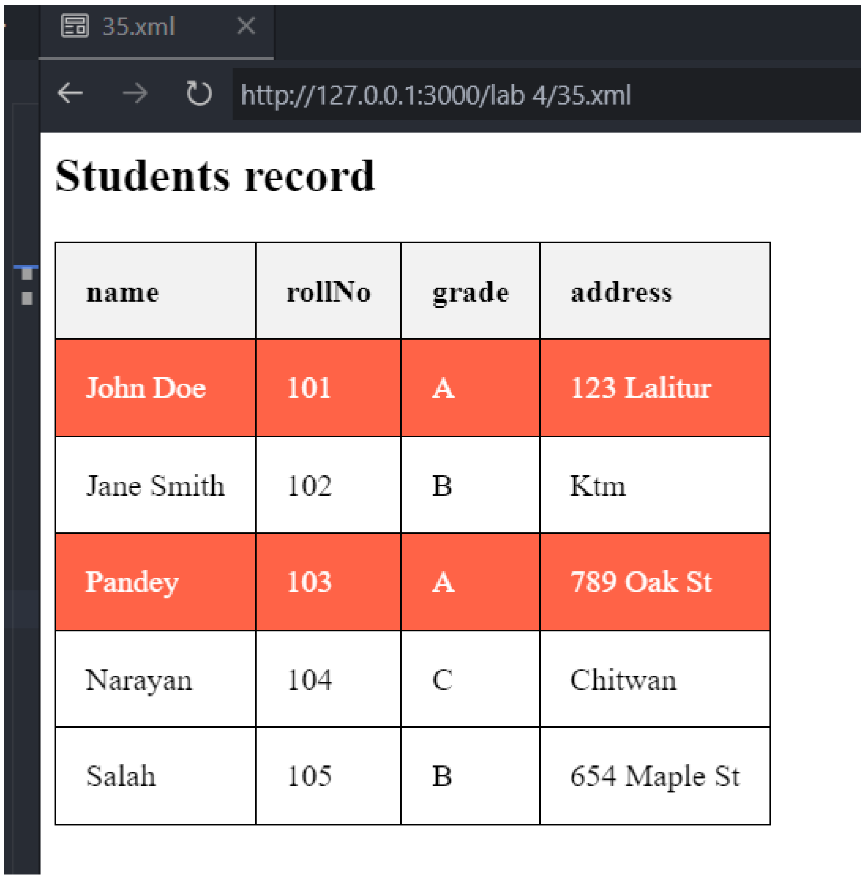


35.xml

```xml
<!-- <?xml version="1.0" encoding="UTF-8"?> -->
<?xml-stylesheet href="35.xsl" type="text/xsl"?>

<students xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="35.xsd">
   <student>
      <name>John Doe</name>
      <rollNo>101</rollNo>
      <grade>A</grade>
      <address>123 Lalitur</address>
   </student>
   <student>
      <name>Jane Smith</name>
      <rollNo>102</rollNo>
      <grade>B</grade>
      <address>Ktm</address>
   </student>
   <student>
      <name>Pandey</name>
      <rollNo>103</rollNo>
      <grade>A</grade>
      <address>789 Oak St</address>
   </student>
   <student>
      <name>Narayan</name>
      <rollNo>104</rollNo>
      <grade>C</grade>
      <address>Chitwan</address>
   </student>
   <student>
      <name>Salah</name>
      <rollNo>105</rollNo>
      <grade>B</grade>
      <address>654 Maple St</address>
   </student>
</students>
```

35.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="students">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="student" maxOccurs="unbounded" minOccurs="5">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="name" type="xs:string"/>
              <xs:element name="rollNo" type="xs:positiveInteger"/>
              <xs:element name="grade" type="xs:string"/>
              <xs:element name="address" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

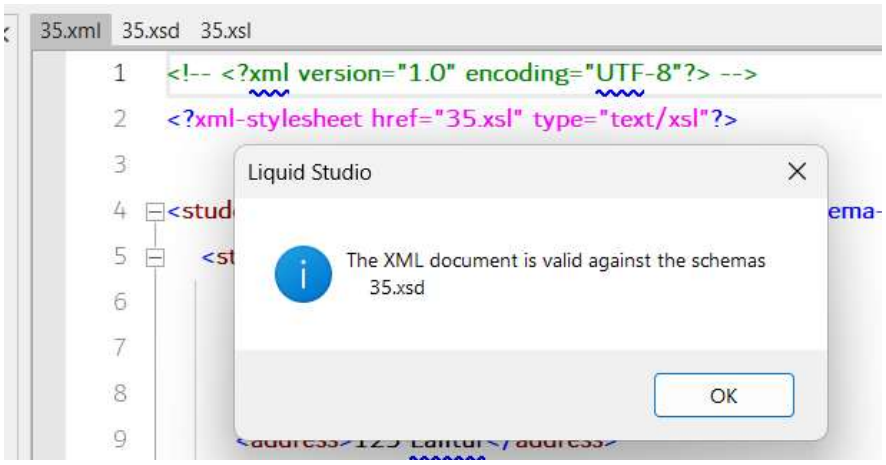


Students record

name	rollNo	grade	address
John Doe	101	A	123 Lalitur
Jane Smith	102	B	Ktm
Pandey	103	A	789 Oak St
Narayan	104	C	Chitwan
Salah	105	B	654 Maple St

35.xsl

```
<!-- <?xml version="1.0" encoding="UTF-8"?> -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <html>
            <style>
                table, th, td {
                    border: 1px solid black;
                    border-collapse: collapse;
                }
                th, td {
                    padding: 15px;
                    text-align: left;
                }
                th {
                    background-color: #f2f2f2;
                }
            </style>
            <body>
                <h2>Students record</h2>
                <table>
                    <thead>
                        <tr>
                            <th>name</th>
                            <th>rollNo</th>
                            <th>grade</th>
                            <th>address</th>
                        </tr>
                    </thead>
                    <tbody>
                        <xsl:for-each select="students/student">
                            <xsl:sort select='rollNo' data-type='number'
order='ascending'/>
                            <tr>
                                <!--conditionally apply background color if grade is A-->
                                <xsl:if test="grade='A'">
                                    <xsl:attribute name='style'>background-
color:tomato;color:white</xsl:attribute>
                                </xsl:if>
                                <td><xsl:value-of select='name'/></td>
                                <td><xsl:value-of select='rollNo'/></td>
                        <td><xsl:value-of select='grade'/></td>
                        <td><xsl:value-of select='address'/></td>
                      </tr>
                    </xsl:for-each>
                  </tbody>
                </table>
              </body>
            </html>
          </xsl:template>
        </xsl:stylesheet>
```



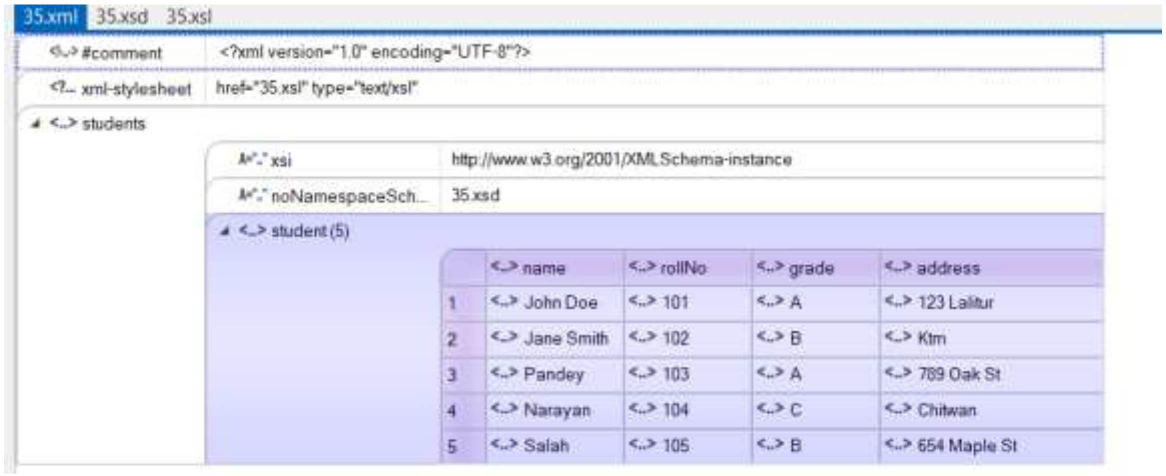
35.xml 35.xsd 35.xsl

```
<!-- <?xml version="1.0" encoding="UTF-8"?> -->
<?xml-stylesheet href="35.xsl" type="text/xsl"?>
```

Liquid Studio

The XML document is valid against the schemas
35.xsd

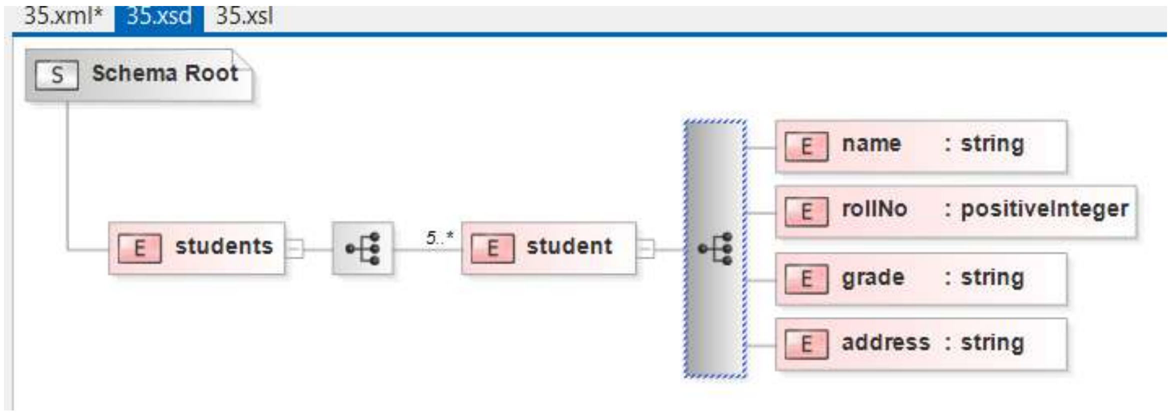
OK



35.xml 35.xsd 35.xsl

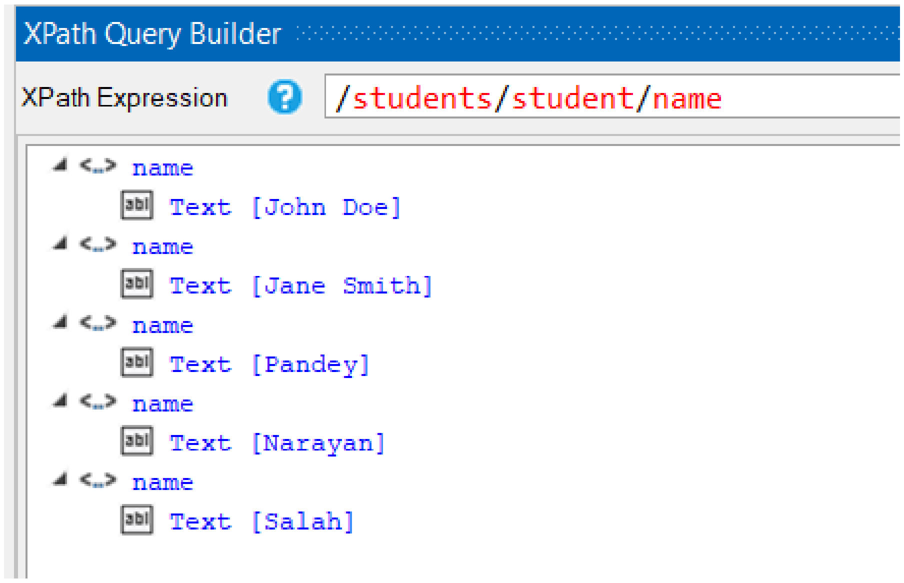
#comment	<?xml version="1.0" encoding="UTF-8"?>
xml-stylesheet	href="35.xsl" type="text/xsl"
students	
xsi	http://www.w3.org/2001/XMLSchema-instance
noNamespaceSch...	35.xsd
student (5)	

	name	rollNo	grade	address
1	John Doe	101	A	123 Lalitur
2	Jane Smith	102	B	Ktm
3	Pandey	103	A	789 Oak St
4	Narayan	104	C	Chitwan
5	Salah	105	B	654 Maple St



36.xsl (Xpath)

1. All student names.
   a. /students/student/name
   b. //student/name

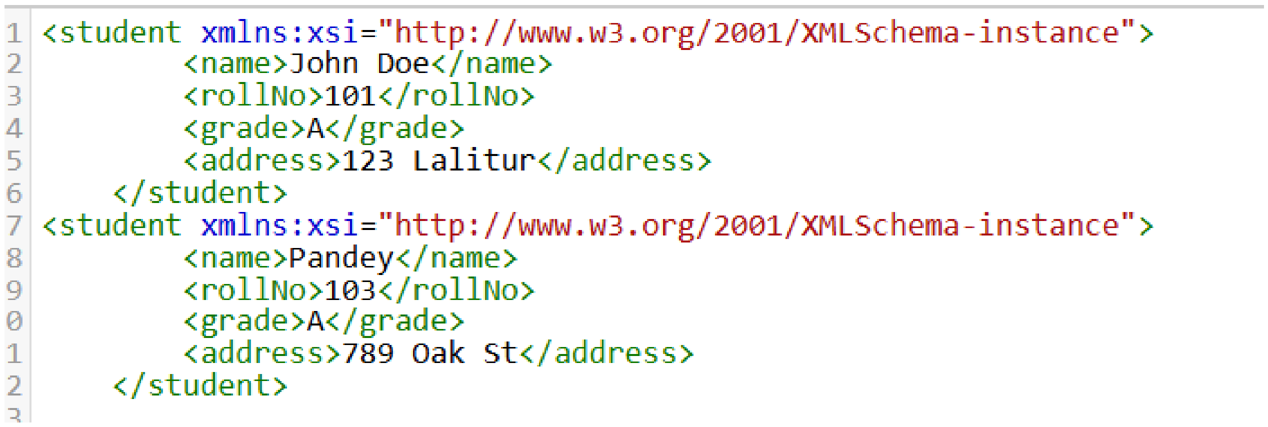


XPath Query Builder

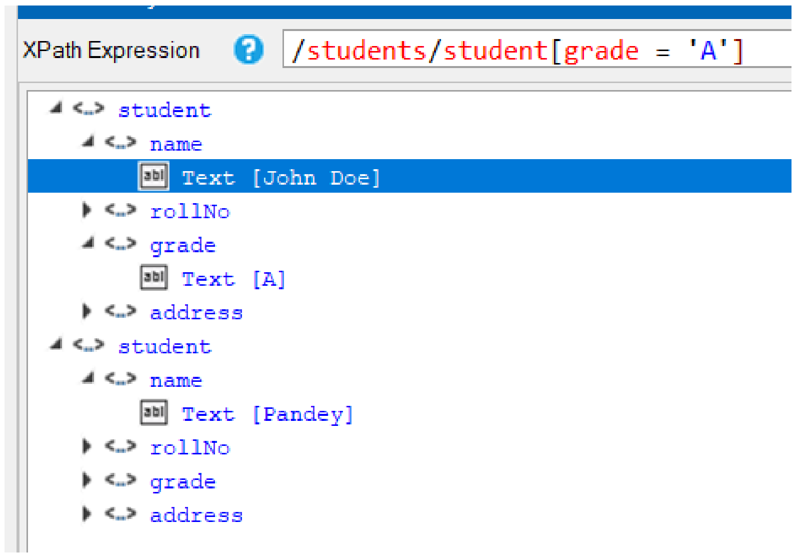
XPath Expression `/students/student/name`

```
◢ <..> name
      Text [John Doe]
◢ <..> name
      Text [Jane Smith]
◢ <..> name
      Text [Pandey]
◢ <..> name
      Text [Narayan]
◢ <..> name
      Text [Salah]
```

2. Students who have a grade 'A'.



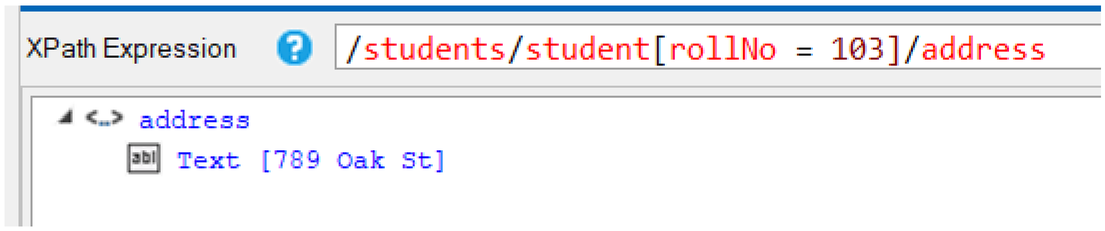
```
<student xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <name>John Doe</name>
        <rollNo>101</rollNo>
        <grade>A</grade>
        <address>123 Lalitur</address>
    </student>
<student xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <name>Pandey</name>
        <rollNo>103</rollNo>
        <grade>A</grade>
        <address>789 Oak St</address>
    </student>
```



XPath Expression `/students/student[grade = 'A']`

```
◢ <..> student
   ◢ <..> name
         Text [John Doe]
   ▸ <..> rollNo
   ◢ <..> grade
         Text [A]
   ▸ <..> address
◢ <..> student
   ◢ <..> name
         Text [Pandey]
   ▸ <..> rollNo
   ▸ <..> grade
   ▸ <..> address
```

3. The address of the student whose roll number is '103'.



XPath Expression

```
/students/student[rollNo = 103]/address
```

```
address
    Text [789 Oak St]
```

37. (XQuery) (37.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<books>
  <book>
      <title>Learning XML</title>
      <author>Erik T. Ray</author>
      <price>39.95</price>
      <publisher>O'Reilly</publisher>
      <year>2003</year>
  </book>
  <book>
      <title>Programming Python</title>
      <author>Mark Lutz</author>
      <price>59.99</price>
      <publisher>O'Reilly</publisher>
      <year>2016</year>
  </book>
  <book>
      <title>Clean Code</title>
      <author>Robert C. Martin</author>
      <price>42.99</price>
      <publisher>Prentice Hall</publisher>
      <year>2008</year>
  </book>
  <book>
      <title>Fluent Python</title>
      <author>Luciano Ramalho</author>
      <price>49.99</price>
      <publisher>O'Reilly</publisher>
      <year>2015</year>
  </book>
  <book>
      <title>Introduction to Algorithms</title>
      <author>Thomas H. Cormen</author>
      <price>84.75</price>
      <publisher>MIT Press</publisher>
      <year>2022</year>
  </book>
  <book>
      <title>Artificial Intelligence: A Modern Approach</title>
      <author>Stuart Russell</author>
      <price>115.00</price>
      <publisher>Pearson</publisher>
      <year>2021</year>
  </book>
</books>
```

37.a.xquery

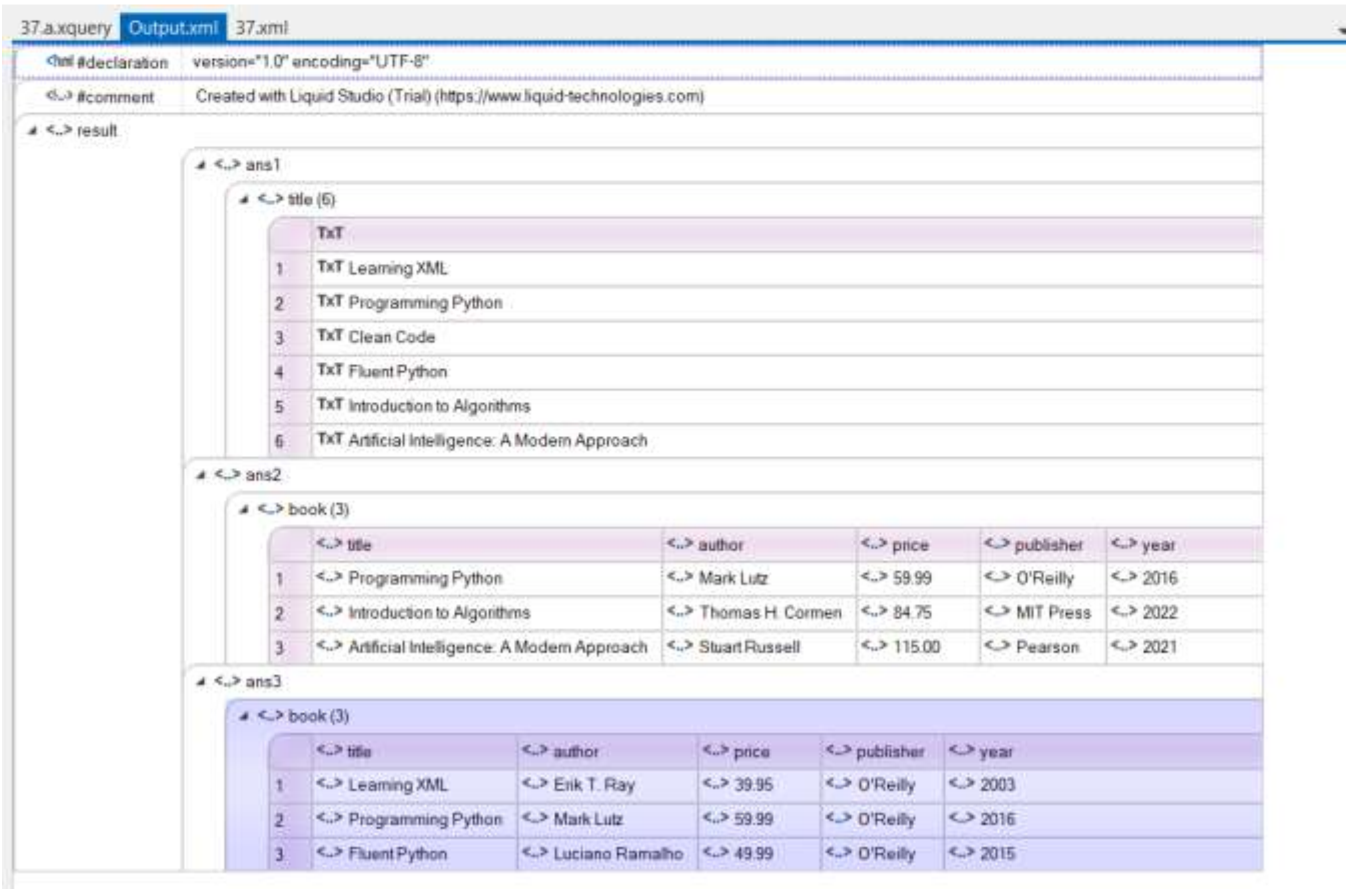
```
let $xml := doc('37.xml')

return
<result>
   <ans1>
      {
         $xml/books/book/title
      }
   </ans1>

   <ans2>
      {
         $xml/books/book[year>2015]
      }
   </ans2>
   <ans3>
      {
         $xml/books/book[publisher = "O'Reilly"]
      }
   </ans3>
</result>
```

Output.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- Created with Liquid Studio (Trial) (https://www.liquid-technologies.com) -->
<result>
   <ans1>
      <title>Learning XML</title>
      <title>Programming Python</title>
      <title>Clean Code</title>
      <title>Fluent Python</title>
      <title>Introduction to Algorithms</title>
      <title>Artificial Intelligence: A Modern Approach</title>
   </ans1>
   <ans2>
      <book>
         <title>Programming Python</title>
         <author>Mark Lutz</author>
         <price>59.99</price>
         <publisher>O'Reilly</publisher>
         <year>2016</year>
      </book>
      <book>
         <title>Introduction to Algorithms</title>
         <author>Thomas H. Cormen</author>
         <price>84.75</price>
         <publisher>MIT Press</publisher>
         <year>2022</year>
      </book>
      <book>
         <title>Artificial Intelligence: A Modern Approach</title>
         <author>Stuart Russell</author>
         <price>115.00</price>
         <publisher>Pearson</publisher>
         <year>2021</year>
      </book>
   </ans2>
   <ans3>
      <book>
         <title>Learning XML</title>
         <author>Erik T. Ray</author>
         <price>39.95</price>
```

```
            <publisher>O'Reilly</publisher>
            <year>2003</year>
        </book>
        <book>
            <title>Programming Python</title>
            <author>Mark Lutz</author>
            <price>59.99</price>
            <publisher>O'Reilly</publisher>
            <year>2016</year>
        </book>
        <book>
            <title>Fluent Python</title>
            <author>Luciano Ramalho</author>
            <price>49.99</price>
            <publisher>O'Reilly</publisher>
            <year>2015</year>
        </book>
    </ans3>
</result>
```

Output: design



37.a.xquery Output.xml 37.xml

#declaration	version="1.0" encoding="UTF-8"
#comment	Created with Liquid Studio (Trial) (https://www.liquid-technologies.com)

result

ans1

title (6)

	TxT
1	Learning XML
2	Programming Python
3	Clean Code
4	Fluent Python
5	Introduction to Algorithms
6	Artificial Intelligence: A Modern Approach

ans2

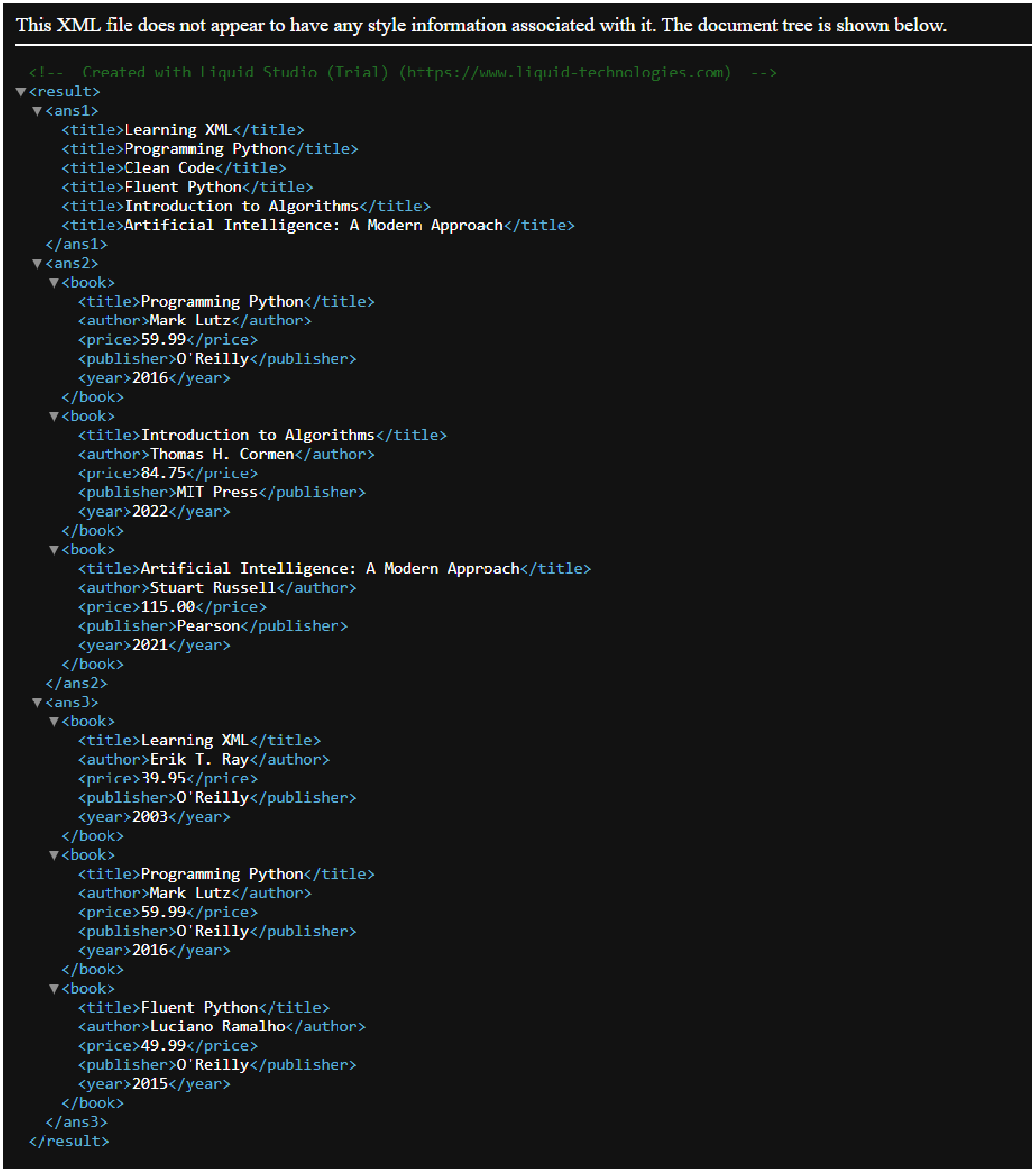
book (3)

	title	author	price	publisher	year
1	Programming Python	Mark Lutz	59.99	O'Reilly	2016
2	Introduction to Algorithms	Thomas H. Cormen	84.75	MIT Press	2022
3	Artificial Intelligence: A Modern Approach	Stuart Russell	115.00	Pearson	2021

ans3

book (3)

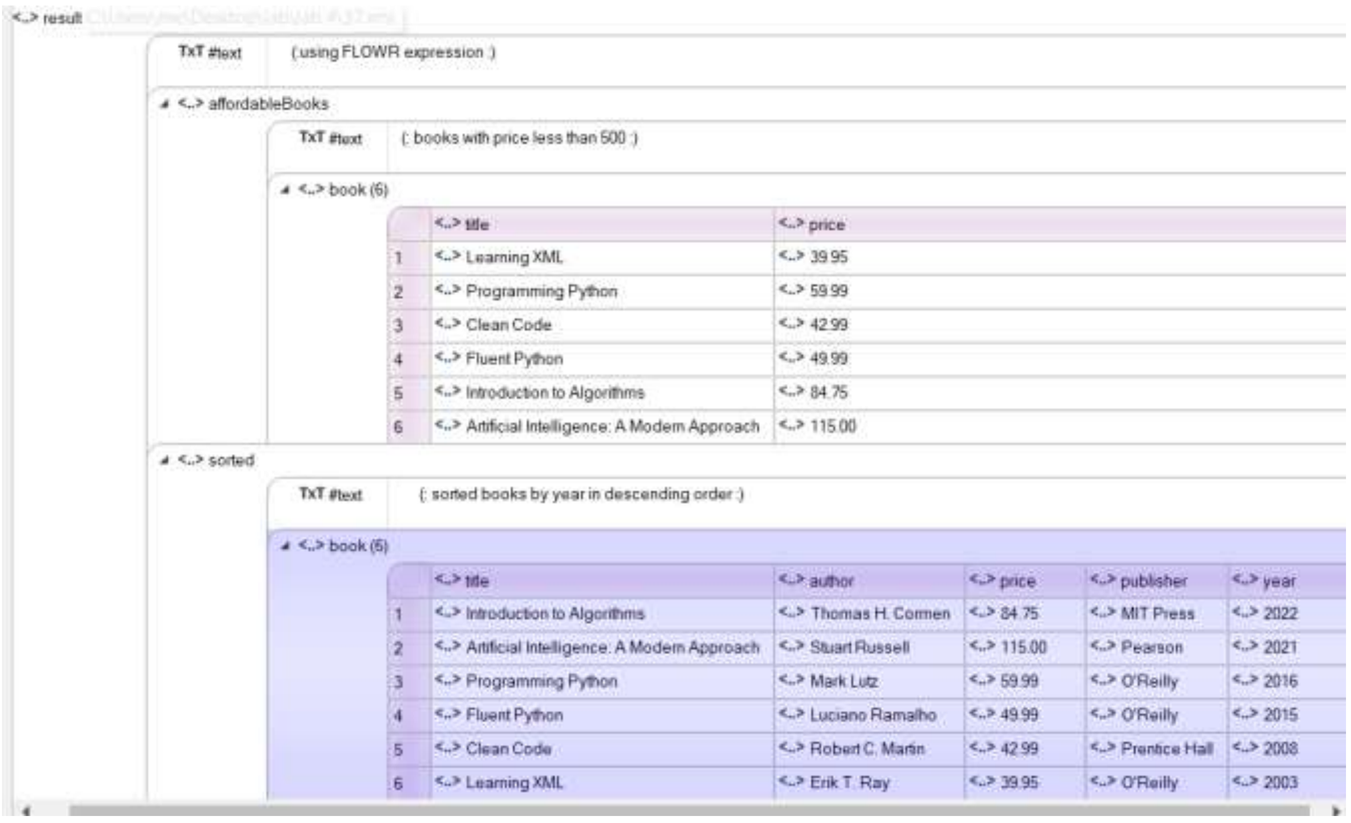
	title	author	price	publisher	year
1	Learning XML	Erik T. Ray	39.95	O'Reilly	2003
2	Programming Python	Mark Lutz	59.99	O'Reilly	2016
3	Fluent Python	Luciano Ramalho	49.99	O'Reilly	2015


This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--  Created with Liquid Studio (Trial) (https://www.liquid-technologies.com)  -->
▼<result>
  ▼<ans1>
     <title>Learning XML</title>
     <title>Programming Python</title>
     <title>Clean Code</title>
     <title>Fluent Python</title>
     <title>Introduction to Algorithms</title>
     <title>Artificial Intelligence: A Modern Approach</title>
   </ans1>
  ▼<ans2>
    ▼<book>
       <title>Programming Python</title>
       <author>Mark Lutz</author>
       <price>59.99</price>
       <publisher>O'Reilly</publisher>
       <year>2016</year>
     </book>
    ▼<book>
       <title>Introduction to Algorithms</title>
       <author>Thomas H. Cormen</author>
       <price>84.75</price>
       <publisher>MIT Press</publisher>
       <year>2022</year>
     </book>
    ▼<book>
       <title>Artificial Intelligence: A Modern Approach</title>
       <author>Stuart Russell</author>
       <price>115.00</price>
       <publisher>Pearson</publisher>
       <year>2021</year>
     </book>
   </ans2>
  ▼<ans3>
    ▼<book>
       <title>Learning XML</title>
       <author>Erik T. Ray</author>
       <price>39.95</price>
       <publisher>O'Reilly</publisher>
       <year>2003</year>
     </book>
    ▼<book>
       <title>Programming Python</title>
       <author>Mark Lutz</author>
       <price>59.99</price>
       <publisher>O'Reilly</publisher>
       <year>2016</year>
     </book>
    ▼<book>
       <title>Fluent Python</title>
       <author>Luciano Ramalho</author>
       <price>49.99</price>
       <publisher>O'Reilly</publisher>
       <year>2015</year>
     </book>
   </ans3>
 </result>
```

37.b.xquery

```
let $xml := doc('37.xml')

return
<result>    (:using FLOWR expression :)
   <affordableBooks> (: books with price less than 500 :)
      {
         for $book in $xml/books/book
         where $book/price < 500
         return <book>
         {$book/title}
         {$book/price}
         </book>
      }
   </affordableBooks>
   <sorted>      (: sorted books by year in descending order :)
      {
         for $book in $xml/books/book
         order by $book/year descending
         return $book
      }
   </sorted>
</result>
```



<..> result

TxT #text (:using FLOWR expression :)

<..> affordableBooks

TxT #text (: books with price less than 500 :)

<..> book (6)

	<..> title	<..> price
1	<..> Learning XML	<..> 39.95
2	<..> Programming Python	<..> 59.99
3	<..> Clean Code	<..> 42.99
4	<..> Fluent Python	<..> 49.99
5	<..> Introduction to Algorithms	<..> 84.75
6	<..> Artificial Intelligence: A Modern Approach	<..> 115.00

<..> sorted

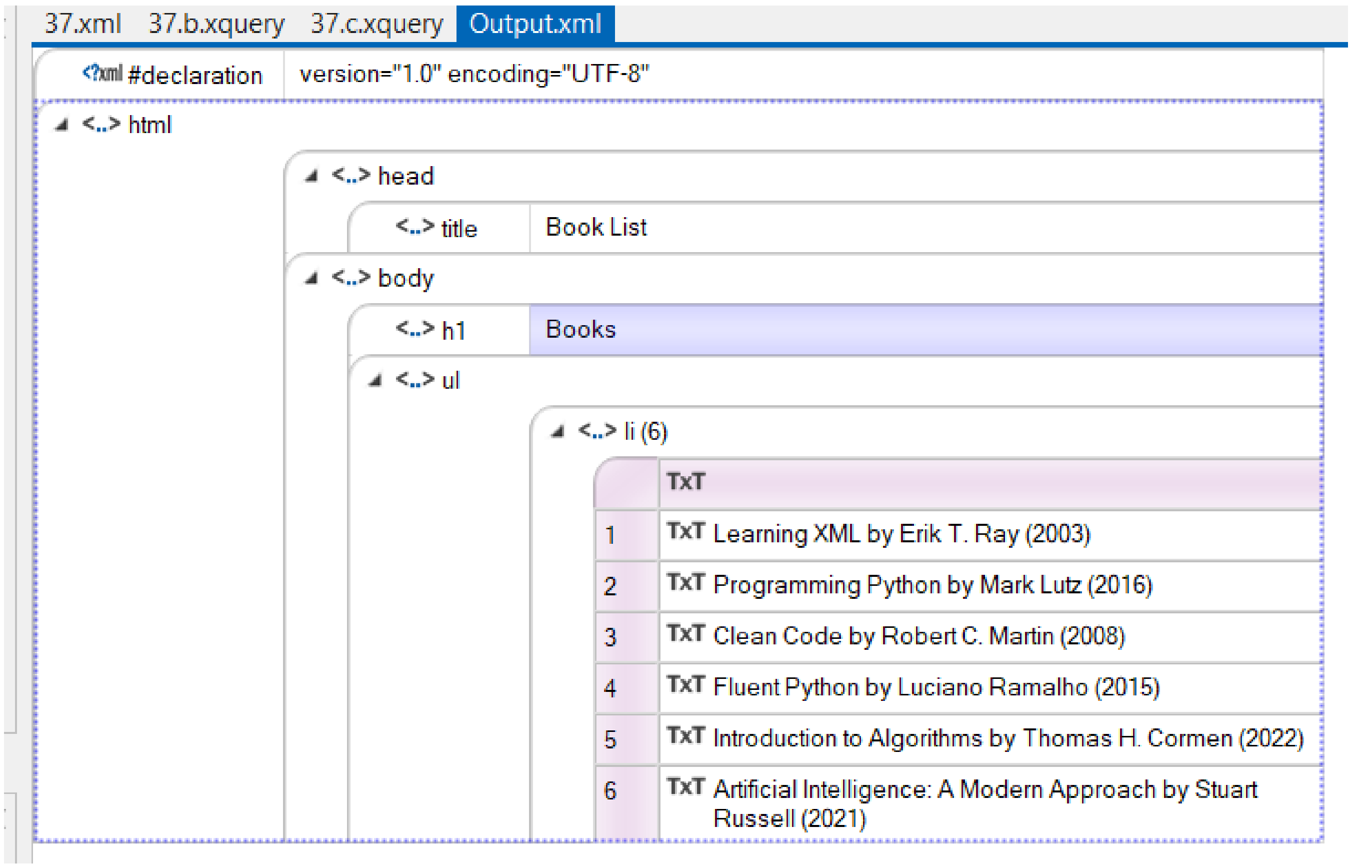
TxT #text (: sorted books by year in descending order :)

<..> book (6)

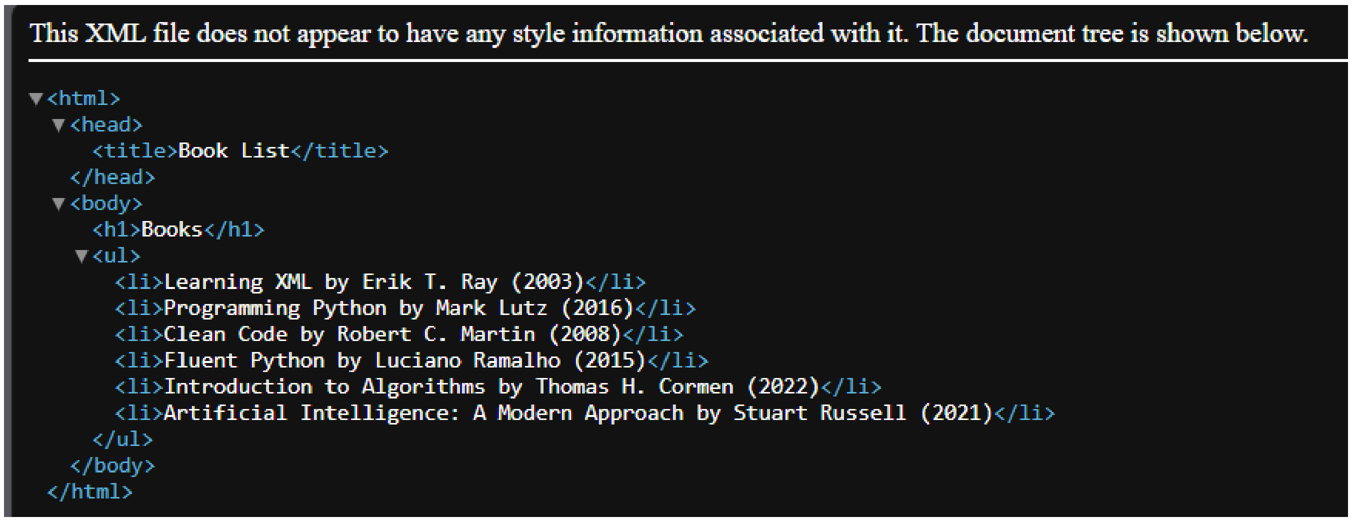
	<..> title	<..> author	<..> price	<..> publisher	<..> year
1	<..> Introduction to Algorithms	<..> Thomas H. Cormen	<..> 84.75	<..> MIT Press	<..> 2022
2	<..> Artificial Intelligence: A Modern Approach	<..> Stuart Russell	<..> 115.00	<..> Pearson	<..> 2021
3	<..> Programming Python	<..> Mark Lutz	<..> 59.99	<..> O'Reilly	<..> 2016
4	<..> Fluent Python	<..> Luciano Ramalho	<..> 49.99	<..> O'Reilly	<..> 2015
5	<..> Clean Code	<..> Robert C. Martin	<..> 42.99	<..> Prentice Hall	<..> 2008
6	<..> Learning XML	<..> Erik T. Ray	<..> 39.95	<..> O'Reilly	<..> 2003

37.c.xquery

```
let $xml := doc('37.xml')

return
<html>
   <head>
      <title>Book List</title>
   </head>
   <body>
      <h1>Books</h1>
      <ul>
      {
         for $book in $xml/books/book
         return <li>{$book/title/text()} by {$book/author/text()}
({$book/year/text()})</li>
      }
      </ul>
   </body>
</html>
```



```
<?xml #declaration   version="1.0" encoding="UTF-8"
<..> html
   <..> head
      <..> title   Book List
   <..> body
      <..> h1   Books
      <..> ul
         <..> li (6)
               TxT
            1  TxT Learning XML by Erik T. Ray (2003)
            2  TxT Programming Python by Mark Lutz (2016)
            3  TxT Clean Code by Robert C. Martin (2008)
            4  TxT Fluent Python by Luciano Ramalho (2015)
            5  TxT Introduction to Algorithms by Thomas H. Cormen (2022)
            6  TxT Artificial Intelligence: A Modern Approach by Stuart
                   Russell (2021)
```



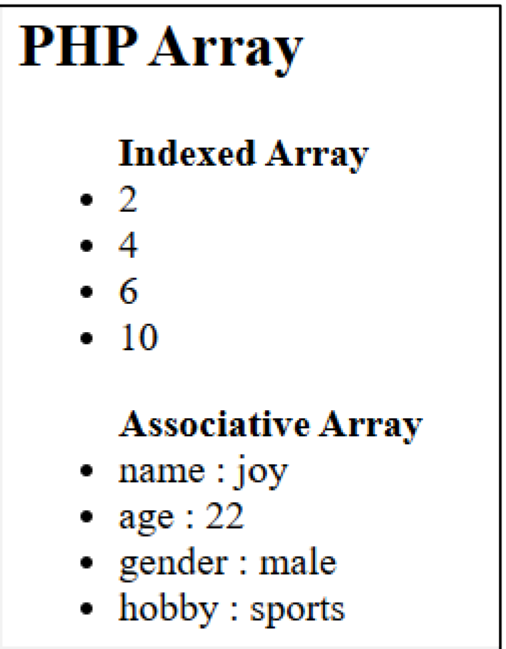
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<html>
  ▼<head>
     <title>Book List</title>
   </head>
  ▼<body>
     <h1>Books</h1>
    ▼<ul>
       <li>Learning XML by Erik T. Ray (2003)</li>
       <li>Programming Python by Mark Lutz (2016)</li>
       <li>Clean Code by Robert C. Martin (2008)</li>
       <li>Fluent Python by Luciano Ramalho (2015)</li>
       <li>Introduction to Algorithms by Thomas H. Cormen (2022)</li>
       <li>Artificial Intelligence: A Modern Approach by Stuart Russell (2021)</li>
     </ul>
   </body>
 </html>
```

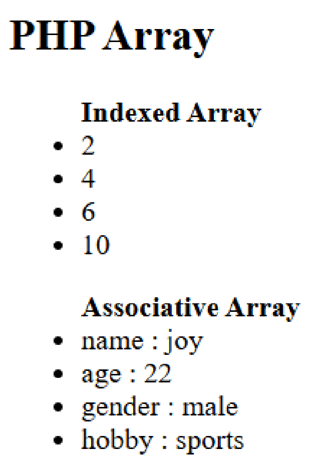
38. Server Side Scripting

38.a.php

```
<!DOCTYPE html>
<html>
<body>
    <h2>PHP Array</h2>
    <ul> <strong>Indexed Array</strong>
        <?php
            // indexed array
            $arr1= [2,4,6,10];
            // displaying the array
            foreach ($arr1 as $value) {
                echo "<li>$value </li>";
            }

        ?>
    </ul>
    <ul><strong>Associative Array</strong>
        <?php
            // associative array
            $arr2= ['name' => 'joy', 'age'=> 22, 'gender' => 'male', 'hobby' => 'sports'];

            foreach ($arr2 as $key => $value) {
                echo "<li>$key : $value </li>";
            }
        ?>
    </ul>

</body>
</html>
```



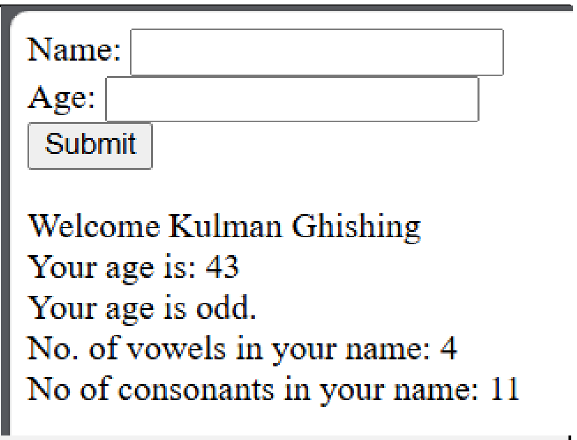
```
PHP Array

    Indexed Array
  • 2
  • 4
  • 6
  • 10

    Associative Array
  • name : joy
  • age : 22
  • gender : male
  • hobby : sports
```



```
PHP Array

    Indexed Array
  • 2
  • 4
  • 6
  • 10

    Associative Array
  • name : joy
  • age : 22
  • gender : male
  • hobby : sports
```

38.b.php

```
<html>
<body>

<form action="38.b.php" method= "POST">
   Name: <input type="text" name="name"><br>
   Age: <input type="number" name="age"><br>
   <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
   $name = $_POST['name'];
   $name = htmlspecialchars($name);  // Converts special characters to HTML
entities to prevent XSS attacks.
   $age = $_POST['age'];

   echo "Welcome ". $name. "<br>";

   if(! empty($age)) {
      echo "Your age is: " . $age . "<br>";
      if ($age % 2 == 0) {
         echo "Your age is even.<br> Multiplication table of your age.:<br>";
         foreach (range(1, 10) as $i){
            echo "{$age} * {$i} = " . $age * $i . "<br>";
         }
      } else {
         // age is odd -> display vowels and consonants counts
         echo "Your age is odd.<br>";
         $vowels = preg_match_all('/[aeiou]/i', $name);
         echo "No. of vowels in your name: " . $vowels . "<br>" . "No of consonants in
your name: " . (strlen($name) - $vowels) "<br>";
      }
   } else
      echo "Age is required!.<br>";
}
?>
</body></html>
```

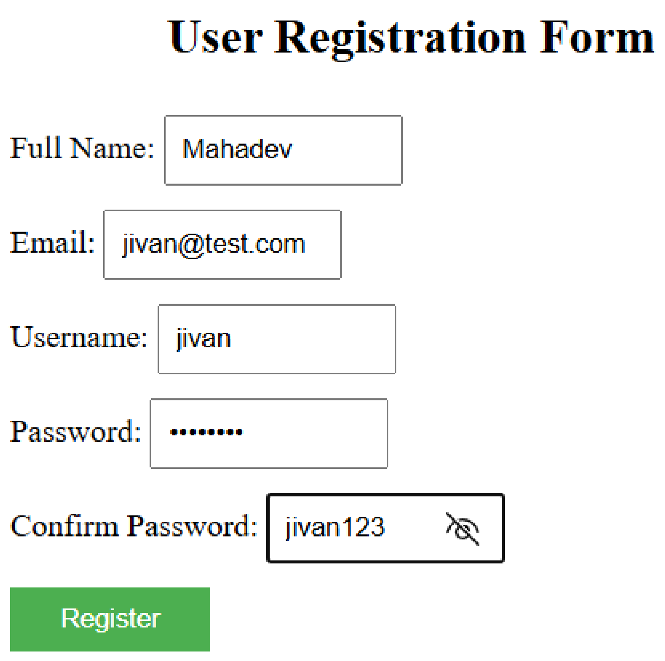


```
Welcome Kulman Ghishing
Your age is: 43
Your age is odd.
No. of vowels in your name: 4
No of consonants in your name: 11
```

38.c.php

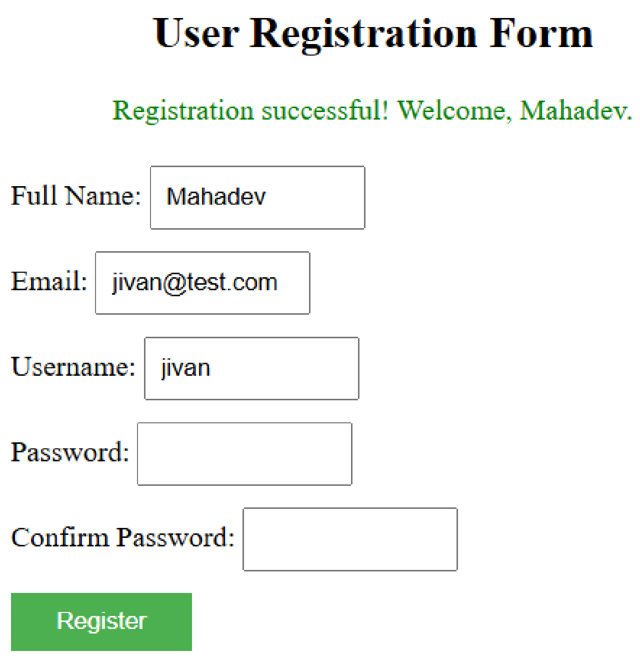
```
<?php
$name = $email = $username = $password = $confirm_password = "";
$nameErr = $emailErr = $usernameErr = $passwordErr = $confirm_passwordErr = "";
$successMsg = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
   $isValid = true; // Flag to check if all fields are valid

   // FULL NAME
   if (empty($_POST['name'])){
      $nameErr = "Full Name is required";
      $isValid = false;
   }else
      $name = htmlspecialchars(trim($_POST['name'])); // Converts special
characters to HTML entities to prevent XSS attacks.']))

   // email
      if (empty($_POST['email'])){
      $emailErr = "Email is required";
      $isValid = false;
   }elseif (!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
      $$emailErr = "Invalid email format";
      $isValid = false;
   } else
      $email = htmlspecialchars(trim($_POST['email']));

   // username (regex)
   if (empty($_POST['username'])){
      $usernameErr = "Username is required";
      $isValid = false;
   } elseif (!preg_match('/^[a-zA-Z0-9_]+$/', $_POST['username'])) {
      $usernameErr= 'Username can only contain letters, numbers, and underscores';
      $isValid = false;
   }else
      $username = htmlspecialchars(trim($_POST['username'])); // Converts special
characters to HTML entities to prevent XSS attacks.']))

   // password
   if(empty($_POST['password'])) {
      $passwordErr= "Password is required";
```

```
        $isValid = false;
    }elseif (strlen($_POST['password'] < 8)) {
        $passwordErr = "Password must be at least 8 characters long";
        $isValid = false;
    }else
        $password = htmlspecialchars(trim($_POST['password']));

    // confirm password
    if(empty($_POST['confirm_password'])) {
        $confirm_passwordErr= 'Please confirm your password';
        $isValid = false;
    }elseif ($_POST['confirm_password'] !== $_POST['password']) {
        $confirm_passwordErr = 'Passwords do not match';
        $isValid = false;
    }else
        $confirm_password = htmlspecialchars(trim($_POST['confirm_password']));

    // If all fields are valid, process the registration
    if($isValid) {
        $successMsg = "Registration successful! Welcome, " . $name . ".<br>";
        // here data can be inserted into a database or processed further
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration Form</title>
    <style>
        .error {color: red;}
        .success {color: green;}
        form {max-width: 400px; margin:auto;}
        form {max-width: 400px; margin:auto}
        input[type="text"], input[type="email"], input[type="password"]
         {
            width: 100px; padding: 8px; margin: 6px 0;
         }
         input[type="submit"] {
            width: 100px; padding: 8px; margin: 6px 0;
            background-color: #4CAF50; color: white; border: none;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <h2 style='text-align:center;'>User Registration Form</h2>
    <p class='success' style='text-align: center;'> <?php echo $successMsg ?> </p>
    <form method='post' action='<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>'>
        <label>Full Name:
            <input type="text" name="name" value= '<?php echo $name; ?>'>
        </label>
        <span class='error'><?php echo $nameErr; ?></span><br>

        <label>Email:
            <input type="email" name="email" value= '<?php echo $email; ?>'>
        </label>
        <span class='error'><?php echo $emailErr; ?></span><br>

        <label>Username:
            <input type="text" name="username" value= '<?php echo $username; ?>'>
        </label>
        <span class='error'><?php echo $usernameErr; ?></span><br>

        <label>Password:
            <input type="password" name="password">
        </label>
        <span class='error'><?php echo $passwordErr; ?></span><br>

        <label>Confirm Password:
            <input type="password" name="confirm_password">
        </label>
        <span class='error'><?php echo $confirm_passwordErr; ?></span><br>

        <input type="submit" value="Register">
    </form>

</body>
</html>
```



User Registration Form

Full Name:	Mahadev
Email:	jivan@test.com
Username:	jivan
Password:	••••••••
Confirm Password:	jivan123



User Registration Form

Registration successful! Welcome, Mahadev.

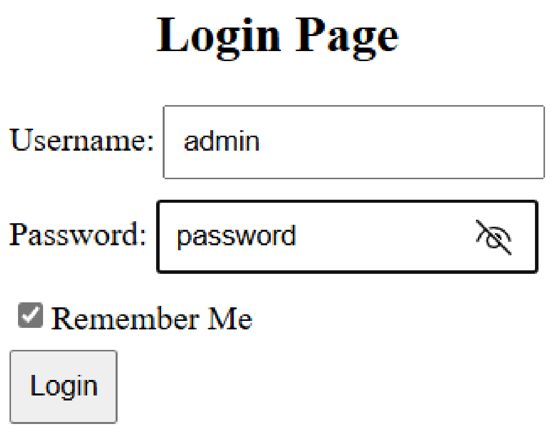
Full Name:	Mahadev
Email:	jivan@test.com
Username:	jivan
Password:	
Confirm Password:	

38.d

login.php

```
<?php
session_start();

// Dummy ceredentials
$correct_username = "admin";
$correct_password = "password";

// auto-login using cookie
if (!isset($_SESSION['username']) && isset($_SESSION['remember_username']))
{
   $_SESSION['username'] = $_COOKIE['remember_username'];
   header("Location: welcome.php");
   exit;
}

$error = '';
if( $_SERVER['REQUEST_METHOD'] == 'POST') {
   $username= $_POST['username'] ?? '';
   $password = $_POST['password'] ?? '';
   $remember = $_POST['remember'] ?? false;

   // Validate credentials
   if ($username === $correct_username && $password === $correct_password) {
      $_SESSION['username'] = $username;

      if ($remember) {
         // Set a cookie for auto-login
         setcookie('remember_username', $username, time() + (86400 *7), '/'); // set
cookie for 7 days
      }else {         // Clear the cookie if "Remember Me" is not checked
         setcookie('remember_username', '', time() - 3600, '/');
      }
      header("Location: welcome.php");
      exit;
   }else {
      $error = "Invalid username or password.";
   }
}
?>
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>login</title>
    <style>
        .error {color: red;}
        form { max-width: 500px; margin:auto; justify-items: center; }
        input margin-bottom: 10px; padding: 8px; }
    </style>
</head>
<body>
    <h2 style='text-align: center;'>Login Page</h2>
    <form method='post' action =''>
        <label>Username:
            <input type='text' name='username' required>
        </label></br>
        <label>Password:
            <input type='password' name='password' required>
        </label><br>

        <label>
            <input type='checkbox' name ='remember'>Remember Me
        </label><br>
        <input type='submit' value='Login'>
        <p class='error'><?php echo $error; ?></p>
    </form>
</body>
</html>
```



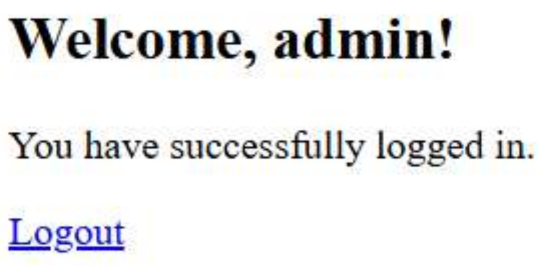
Login Page

Username: admin

Password: password

✓ Remember Me

Login



Welcome, admin!

You have successfully logged in.

Logout

welcome.php

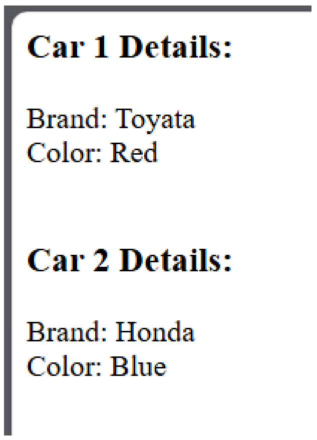
```
<?php
session_start();
if (!isset($_SESSION['username'])) {
    header("Location: login.php");
    exit;
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome</title>
</head>
<body>
    <h2>Welcome, <?php echo $_SESSION['username']; ?>! </h2>
    <p>You have successfully logged in.</p>
    <p><a href='logout.php'>Logout</a></p>
</body>
</html>
```

logout.php

```
<?php
session_start();
session_unset(); // Unset all session variables
session_destroy(); // Destroy the session

// delete cookie
setcookie('remember_username', '', time() -3600, '/'); // Delete the cookie by
setting its expiration time in the past

header("Location: login.php"); // Redirect to the login page
exit; // Ensure no further code is executed
?>
```

38.e.php

```
<?php
error_reporting(E_ALL);
ini_set("display_errors", 1);
class Car {
    public $brand;
    public $color;

    // Constructor to initialize the properties
    public function __construct($brand, $color) {
        $this->brand = $brand;
        $this->color = $color;
    }

    // method to display car details
    public function displayInfo() {
        echo "Brand: " . $this->brand . "<br>";
        echo "Color: " . $this->color . "<br><br>";
    }
}

// Create an instance of the Car class
$car1 = new Car("Toyata", "Red");
$car2 = new Car("Honda", "Blue");

// call methods on the obhects
echo "<h3>Car 1 Details:</h3>";
$car1->displayInfo();
echo "<h3>Car 2 Details:</h3>";
$car2->displayInfo();

?>
```

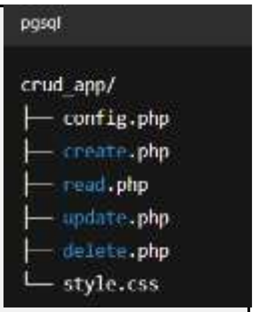


```
Car 1 Details:

Brand: Toyata
Color: Red


Car 2 Details:

Brand: Honda
Color: Blue
```

38.f (crud app)

sql

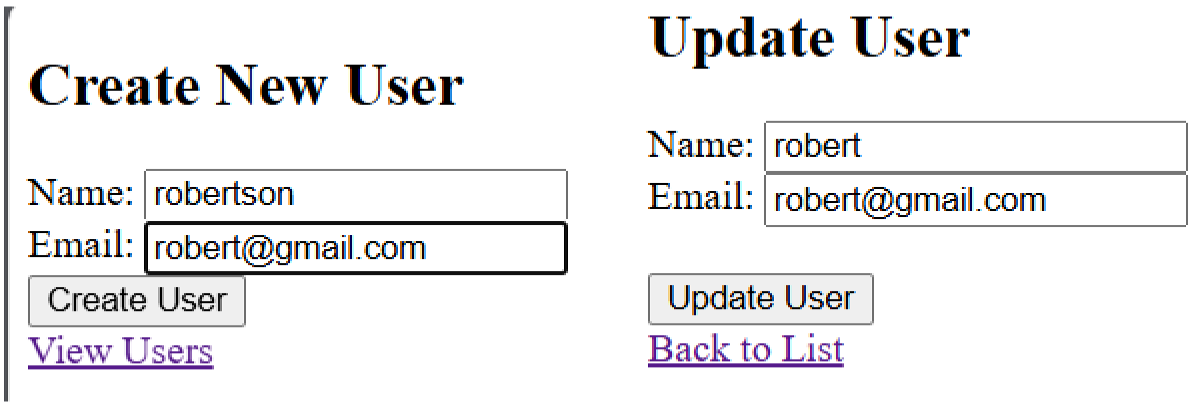
```
CREATE DATABASE crud_php;

USE crud_php;

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE
);
```



```
pgsql

crud_app/
├── config.php
├── create.php
├── read.php
├── update.php
├── delete.php
└── style.css
```

create.php

```
<!--  insert new user -->
<?php include 'config.php'?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Create User</title>
</head>
<body>
    <h2>Create New User</h2>
    <form method='post'>
        Name: <input type='text' name='name' required><br>
        Email: <input type='email' name='email' required><br>
        <button type='submit' name ='submit'>Create User</button>
    </form>
    <a href = 'read.php'>View Users</a>

<?php
if (isset($_POST['submit'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];

    // $sql = "INSERT INTO users (name, email) VALUES ('$name', '$email')"; plain
sql vulnerable to SQL injection
    // $result = mysqli_query($conn, $sql);
    /* Prepared statement to prevent SQL injection
    Sends the query to MySQL for pre-compilation (without values).
    Creates a statement object ($stmt) that can later be safely bound to variables.
    Protects against SQL injection because data is sent separately from the query.
    */

    $stmt = $conn->prepare("INSERT INTO users (name, email) values (?,?)");
    $stmt->bind_param('ss', $name, $email); // Binds the variables to the prepared
statement as parameters. 'ss' indicates that both parameters are strings.

    if ($stmt->execute())
        echo "<p>User created successfully!</p>";
    else
        echo "<p>Error creating user: " . $stmt->error . "</p>";
}
?>
</body>
</html>
```

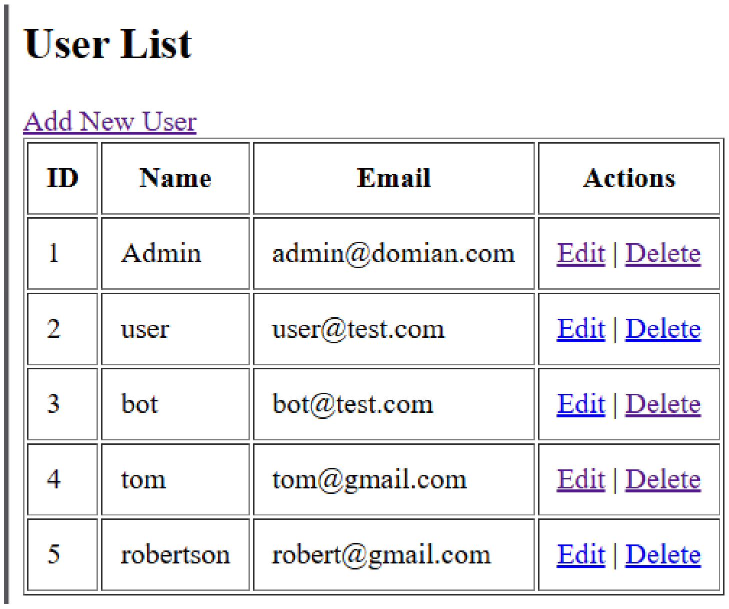


Create New User

Name: robertson
Email: robert@gmail.com

Update User

Name: robert
Email: robert@gmail.com



User List

ID	Name	Email	Actions
1	Admin	admin@domian.com	Edit | Delete
2	user	user@test.com	Edit | Delete
3	bot	bot@test.com	Edit | Delete
4	tom	tom@gmail.com	Edit | Delete
5	robertson	robert@gmail.com	Edit | Delete

read.php

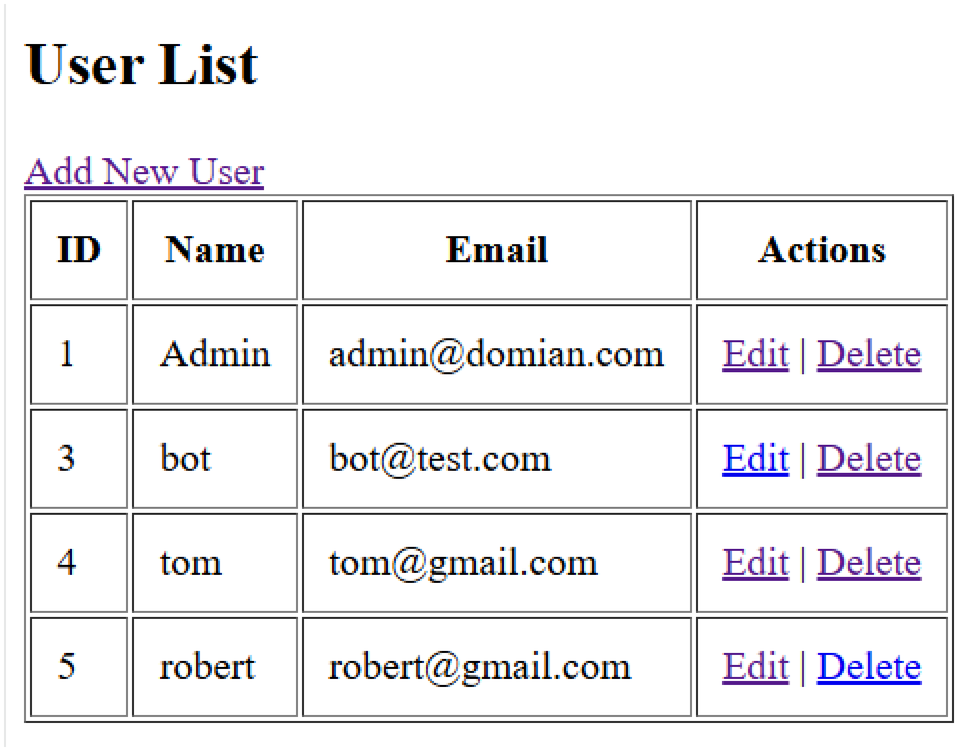
```
<!-- display all users -->
<?php include 'config.php'; ?>
<!DOCTYPE html>
<head><title>User List</title></head>
<body>
  <h2>User List</h2>
<a href ='create.php'>Add New User</a>
  <table border='1' cellpadding='10'>
    <tr><th>ID</th><th>Name</th><th>Email</th><th>Actions</th></tr>
    <?php
    $result = $conn->query("SELECT * FROM users");
    while ($row = $result->fetch_assoc()) {
      echo "<tr>
        <td>{$row['id']}</td>
        <td>{$row['name']}</td>
        <td>{$row['email']}</td>
        <td>
          <a href='update.php?id={$row['id']}'>Edit</a> |
          <a href='delete.php?id={$row['id']}' onclick ='return confirm(\"Delete
this user?\")'>Delete</a>
        </td>
      </tr>";
    }
    ?>
  </table>
</body>
</html>
```

update.php

```
<!-- edit user info -->
 <?php include 'config.php'; ?>
<!DOCTYPE html>
<html>
<head><title>Update User</title></head>
<body>
    <h2>Update User</h2>

<?php
$id= $_GET['id'];
$result = $conn->query("SELECT * FROM users WHERE id = $id");
$row = $result->fetch_assoc();
?>

<form method='post'>
    Name: <input type='text' name='name' value='<?= $row['name'] ?>' required><br>
    Email: <input type='text' name='email' value='<?= $row['email'] ?>'
required><br><br>
    <button type='submit' name='update'>Update User</button>
</form>
<a href='read.php'>Back to List</a>

<?php
if (isset($_POST['update'])) {
    $name= $_POST['name'];
    $email= $_POST['email'];
    $stmt = $conn->prepare("UPDATE users SET name = ?, email = ? WHERE id = ?");
    $stmt->bind_param('ssi', $name, $email, $id); // 'ssi' indicates two strings and
one integer

    if ($stmt->execute())
        echo "<p>User updated successfully!</p>";
    else
        echo "<p>Error updating user: " . $stmt->error . "</p>";
}
?>
</body>
</html>
```

delete.php

```
<?php
 include 'config.php';

$id= $_GET['id'];
$stmt= $conn->prepare("DELETE FROM users WHERE id = ?");
$stmt->bind_param('i', $id); // 'i' indicates that the parameter is an integer
$stmt->execute();

header("Location: read.php"); // Redirect to the user list after deletion
exit; // Ensure no further code is executed after the redirect

?>
```



User List

Add New User

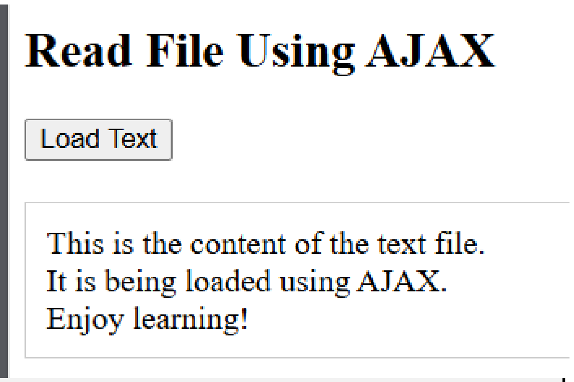
ID	Name	Email	Actions
1	Admin	admin@domian.com	Edit | Delete
3	bot	bot@test.com	Edit | Delete
4	tom	tom@gmail.com	Edit | Delete
5	robert	robert@gmail.com	Edit | Delete

39. AJAX

39.a (file.txt; index.html; fetch-text.js)

```
This is the content of the text file.
It is being loaded using AJAX.
Enjoy learning!
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Read Text File Using AJAX</title>
    <script src="fetch-text.js"></script>
</head>
<body>
    <h2>Read File Using AJAX</h2>
    <button onclick="loadTextFile()">Load Text</button>
    <div id="output" style="margin-top: 20px; padding: 10px; border: 1px solid
#ccc;"></div>
</body>
</html>
```



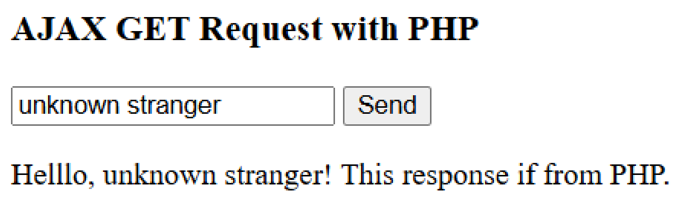
```
This is the content of the text file.
It is being loaded using AJAX.
Enjoy learning!
```

```
// valila Javascript code to fetch text from a URL and display it in a div

function loadTextFile() {
    const xhr = new XMLHttpRequest();
    xhr.open("GET", "file.txt", true);

    xhr.onload = function() {
        if (this.status === 200) {
            document.getElementById("output").innerText = this.responseText;
        }else
            document.getElementById("output").innerText = "Failed to load file: " +
this.status;
    };

    xhr.onerror = function() {
        document.getElementById("output").innerText = "Request Error.";
    };

    xhr.send();
}
```

39.b.html

```
<!-- AJAX with PHp for GET request -->
<!DOCTYPE html>
<html lang="en">
<head>
    <title>AJAX GET Request</title>
    <script>
        function sendData() {
            let name= document.getElementById("name").value;
            let xhr = new XMLHttpRequest();
            xhr.open("GET", '39.b.php?name=' + encodeURIComponent(name), true);
            xhr.onreadystatechange = function() {
                if (xhr.readyState === 4 && xhr.status === 200) {
                    document.getElementById('response').innerHTML = xhr.responseText;
                }
            };
            xhr.send();
        }
    </script>
</head>
<body>
    <h3>AJAX GET Request with PHP</h3>
    <input type="text" id="name" placeholder="Enter your name">
    <button onclick="sendData()">Send</button>
    <p id="response"></p>
</body>
</html>
```

```
<?php
if (isset($_GET['name'])) {
    $name = htmlspecialchars($_GET['name']);    // sanitize input
    echo "Helllo, " . $name . "! This response if from PHP.";
} else {
    echo "No name received.";
}
?>
```



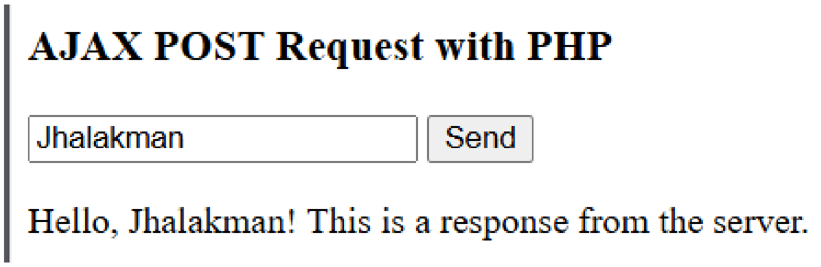
AJAX GET Request with PHP

Helllo, unknown stranger! This response if from PHP.

39.c. html

```
<!DOCTYPE html>
<html>
<head>
    <title>AJAX POST Request</title>
    <script>
        function sendData() {
            let name = document.getElementById("name").value;
            let xhr = new XMLHttpRequest();
            xhr.open("POST", "39.c.php", true);
            xhr.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
            xhr.onreadystatechange = function() {
                if (xhr.readyState === 4 && xhr.status === 200)
                    document.getElementById("response").innerHTML =
xhr.responseText;
            };
            xhr.send('name=' + encodeURIComponent(name));
        }
        </script>
</head>
<body>
    <h3>AJAX POST Request with PHP</h3>
    <input type="text" id="name" placeholder="Enter your name">
    <button onclick="sendData()">Send</button>
    <p id="response"></p>
</body></html>
```

```
<?php
if ($_SERVER['REQUEST_METHOD'] === "POST") {
    $name = htmlspecialchars($_POST['name']);
    echo "Hello, " . $name . "! This is a response from the server.";
}else
    echo "No name received via POST.";
?>
```



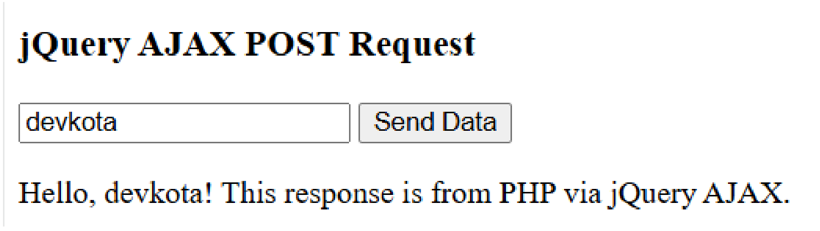
AJAX POST Request with PHP

Hello, Jhalakman! This is a response from the server.

39.d. html

```
<!-- Frontend using jQuery -->
 <!DOCTYPE html>
<html>
<head>
   <title>jQuery AJAX Example</title>
   <!-- jQuery CDN -->
   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
   <script>
      function sendData() {
         var name= $('#name').val();
         $.ajax({
           url: '39.d.php',     // Target PHP script
            type: 'POST',          // HTTP method
            data: { name: name },  // Data to send
            success: function(response){
               $('#response').html(response); // Display response
            },
            error: function() {
               $('#response').html('Error contacting server.');
            }
         });
      }
   </script>
</head>
<body>
   <h3>jQuery AJAX POST Request</h3>
   <input type="text" id="name" placeholder="Enter your name">
   <button onclick="sendData()">Send Data</button>
   <p id="response"></p>
</body></html>
```

```
<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
   $name = htmlspecialchars($_POST['name']);
   echo "Hello, " . $name . "! This response is from PHP via jQuery AJAX.";
} else
   echo "No name received.";
?>
```


jQuery AJAX POST Request

devkota

Send Data

Hello, devkota! This response is from PHP via jQuery AJAX.

39.e (database setup)

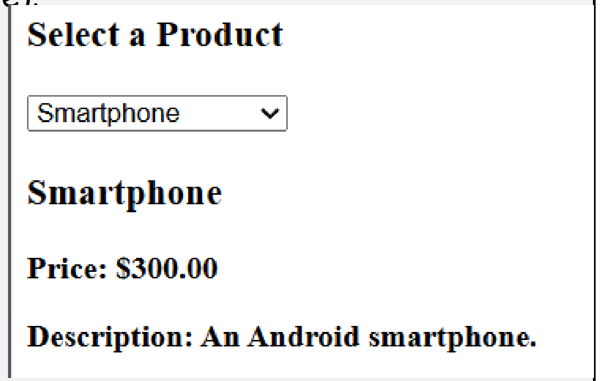
```
CREATE DATABASE productdb;
USE productdb;

CREATE TABLE products (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    price DECIMAL(10,2),
    description TEXT
);

INSERT INTO products (name, price, description) VALUES
('Laptop', 700.00, 'A high-performance laptop.'),
('Smartphone', 300.00, 'An Android smartphone.'),
('Headphones', 50.00, 'Noise-cancelling headphones.');
```

get_product.php

```
<?php
if (isset($_POST['product_id'])) {
    $conn= new mysqli('localhost', 'root', '', 'productdb');

    $id =(int) $_POST['product_id'];
    $stmt = $conn->prepare("SELECT * FROM products WHERE id = ?");
    $stmt->bind_param("i", $id);
    $stmt->execute();

    $result= $stmt->get_result();
    if ($row = $result->fetch_assoc()) {
        echo "<h3>" . htmlspecialchars($row['name']) . "</h3>";
        echo "<p><strong>Price:</.strong> $" . htmlspecialchars($row['price']) . "</p>";
        echo "<p><strong>Description:</strong> " . htmlspecialchars($row['description'])
. "</p>";
    }else {
        echo "<p>No product found.</p>";
    }

    $stmt->close();
    $conn->close();
}
?>
```

Index.php

```
<!-- html + ajax -->
<!DOCTYPE html>
<html>
<head>
    <title>AJAX Dropdown with PHP and MySQL</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script>
        function fetchDetails(id) {
            $.ajax({
                url: 'get_product.php',
                type: 'POST',
                data: {product_id: id},
                success: function(response) {
                    $('#productDetails').html(response);
                }
            });
        }
    </script>
</head>
<body>
    <h3>Select a Product</h3>
    <select onchange='fetchDetails(this.value)'>
        <option value="">-- Select product --</option>
        <?php
            // Database connection
            $conn = new mysqli('localhost', 'root', '', 'productdb');

            // fetch product list
            $result= $conn->query("SELECT id, name FROM products");
            while ($row = $result->fetch_assoc()) {
                echo "<option value = '{$row["id"]}'>{$row['name']}</option>";
            }
        ?>
    </select>
    <div id="productDetails" style="margin-top:20px;"></div>
</body>
</html>
```



Select a Product

Smartphone

Smartphone

Price: $300.00

Description: An Android smartphone.

39.f

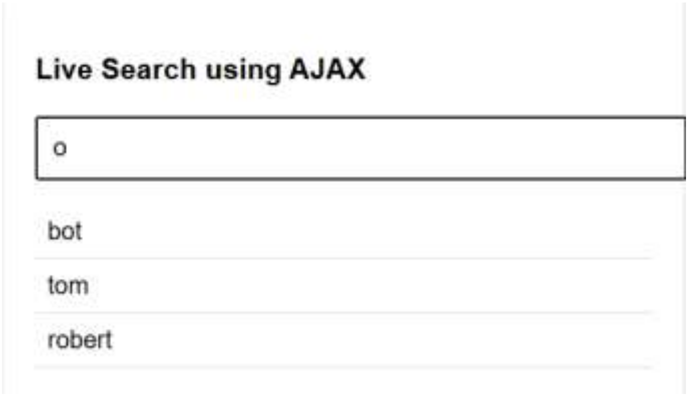
db.php

```
<?php
$host= 'localhost';
$username = 'root';
$password = '';
$dbname= 'crud_php';

// create connnection
$conn= new mysqli($host, $username, $password, $dbname);

// check for connection
if ($conn->connect_error) {
    die('Connection failed: ' . $conn->connect_error)    ;
}
?>
```

search.php

```
<?php
require 'db.php';
if (isset($_POST['query'])) {
    $search = $conn->real_escape_string($_POST['query']);

    $sql ="SELECT * FROM users WHERE name LIKE '%$search%' LIMIT 10";
    $result = $conn->query($sql);

    if ($result->num_rows > 0){
        echo '<ul>';
        while ($row = $result->fetch_assoc()) {
            echo '<li>' .htmlspecialchars($row['name']) . '</li>';
        }
        echo "</ul>";
    }else
        echo '<p>No results found</p>';
}
?>
```

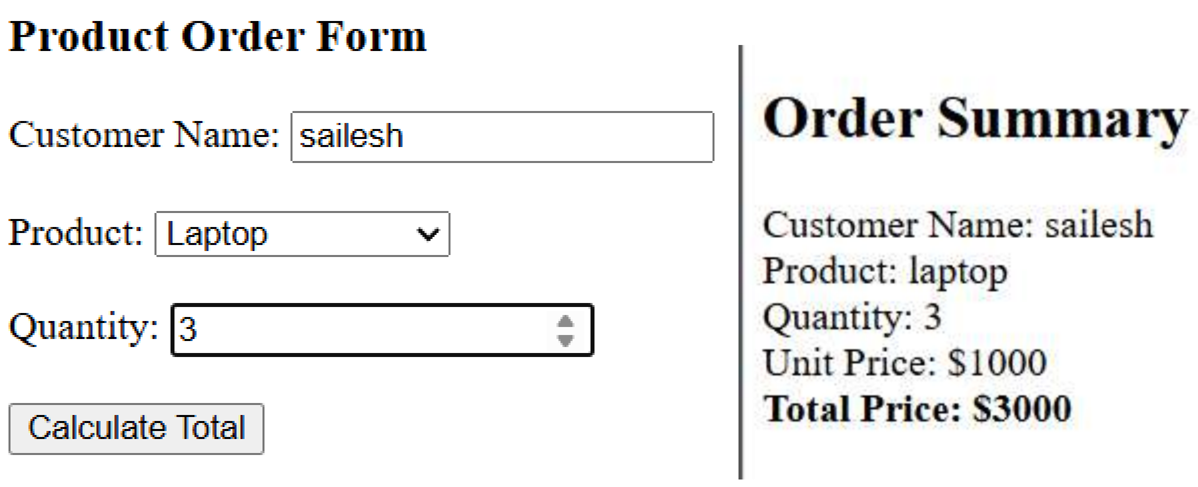
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Live Search using AJAX</title>
   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <div class ='search-box'>
      <h3>Live Search using AJAX</h3>
      <input type='text' id='search' placeholder='Search by name...'>
      <div id='result'></div>
   </div>
   <script>
      $(document).ready(function (){
         $('#search').on('keyup', function () {
            let query = $(this).val();
            if ( query.length > 0) {
               $.ajax({
                  url: 'search.php',
                  type: 'POST',
                  data: {query: query},
                  success: function (data) {
                     $("#result").html(data);
                  }
               });
            }else
               $('#result').html('');
         });
      });
   </script>
</body></html>
```



bot
tom
robert

40 index.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Order Form</title>
    <script>
        function validateCustomerName() {
            const name = document.getElementById("customer").value.trim();
            if (name === ''){
                alert ("Please enter a customer name.");
                return false;
            }
            return true;
        }

        function validateQuuantity() {
            const quantity= document.getElementById("quantity").value;
            if (isNaN(quantity) || quantity <= 0){
                alert ("Quantity must be a +ve number.");
                return false;
            }
            return true;
        }

        function validateForm() {
            return validateCustomerName() && validateQuuantity();
        }
    </script>
</head>
<body>
    <h3>Product Order Form</h3>
    <form method ='post' action ='' onsubmit='return validateForm()'>
        Customer Name:
        <input type='text' id='customer' name='customer'
onblur='validateCustomerName()'><br><br>
        Product: <select name='product'>
            <option value=''>Select a product</option>
            <option value='laptop'>Laptop</option>
            <option value='phone'>Phone</option>
            <option value='tablet'>Tablet</option>
            </select><br><br>
            Quantity:
            <input type='number' id='quantity' name='quantity' min='1'
onblur='validateQuuantity()'><br><br>
        <input type='submit' value='Calculate Total'>
    </form>
</body>
</html>

<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $customer = htmlspecialchars($_POST['customer']);
    $product = $_POST['product'];
    $quantity = (int)$_POST['quantity'];

    // prices
    $prices =[
        'laptop' => 1000,
        'phone' => 500,
        'tablet' => 300
    ];
    $unit_price = $prices[$product];
    $total = $unit_price * $quantity;

    echo "<h2>Order Summary</h2>
        Customer Name: $customer<br>
        Product: $product<br>
        Quantity: $quantity<br>
        Unit Price: $$unit_price<br>
        <strong>Total Price: $$total</strong>";
}
?>
```



Product Order Form

Customer Name: sailesh

Product: Laptop

Quantity: 3

Calculate Total

Order Summary

Customer Name: sailesh
Product: laptop
Quantity: 3
Unit Price: $1000
Total Price: $3000

41. PHP framework

codeignitor

1. **Create Database and Table**

```
CREATE DATABASE ci4_crud_db;

USE ci4_crud_db;

CREATE TABLE products (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    price DECIMAL(10,2) NOT NULL
);
```

Update .env or app/Config/Database.php with:

```
database.default.hostname = localhost
database.default.database = ci4_crud_db
database.default.username = root
database.default.password =
database.default.DBDriver = MySQLi
```

2. **Create Model (app/Models/ProductModel.php)**

```
<?php
namespace App\Models;

use CodeIgniter\Model;

class ProductModel extends Model{
    protected $table ='products';
    protected $primaryKey= 'id';
    protected $allowedFields = ['name', 'price'];
}
```

3. **Create Controller (app/Controllers/Product.php)**

```
<?php
namespace App\Controllers;

use App\Models\ProductModel;

class Product extends BaseController{
    public function index() {
        $model = new ProductModel();
        $data['products'] = $model->findAll();
        return view ('product_list', $data);
    }

    public function create() {
        return view('product_create');
    }

    public function store() {
        $model = new ProductModel();
        $model->save([
            'name' => $this->request->getPost('name'),
            'price' => $this->request->getPost('price'),
        ]);
        return redirect()->to(base_url('product'));
    }
}
```

4. Create Views (app/Views/product_list.php)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Product List</title>
    <style>
        table{
            width: 25%;
            border-collapse: collapse;
        }
      th{
            background-color: green;
            color:white;
        }
  </style>
</head>

<body>
    <h3>Product List</h3>
    <a href= 'product/create'>Add New Product</a>
    <table border='1'>
        <tr><th>ID</th><th>Name</th><th>Price</th></tr>
        <?php foreach ($products as $product): ?>
        <tr>
            <td><?= $product['id'] ?></td>
            <td><?= $product['name'] ?></td>
            <td><?= $product['price'] ?></td>
        </tr>
        <?php endforeach; ?>
    </table>
</body>
</html>
```

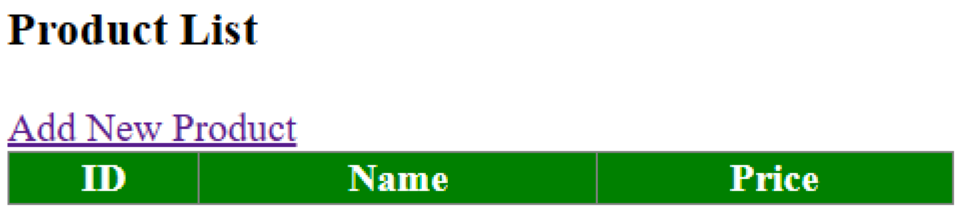
app/Views/product_create.php

```
<!DOCTYPE html>
<html>
<head><title>Create Product</title></head>

<body>
   <h2>Add New Product</h2>
   <form method='post' action="store">
      Name: <input type='text' name='name' required><br>
      Price: <input type='number' name='price' required><br>
      <input type='submit' value='Add Product'>
   </form>
</body>
</html>
```

5. Define Routes (app/Config/Routes.php)

```
<?php

use CodeIgniter\Router\RouteCollection;

/**
 * @var RouteCollection $routes
 */
$routes->get('/', 'Home::index');

$routes->get('/product', 'Product::index');
$routes->get('/product/create', 'Product::create');
$routes->post('/product/store', 'Product::store');
```

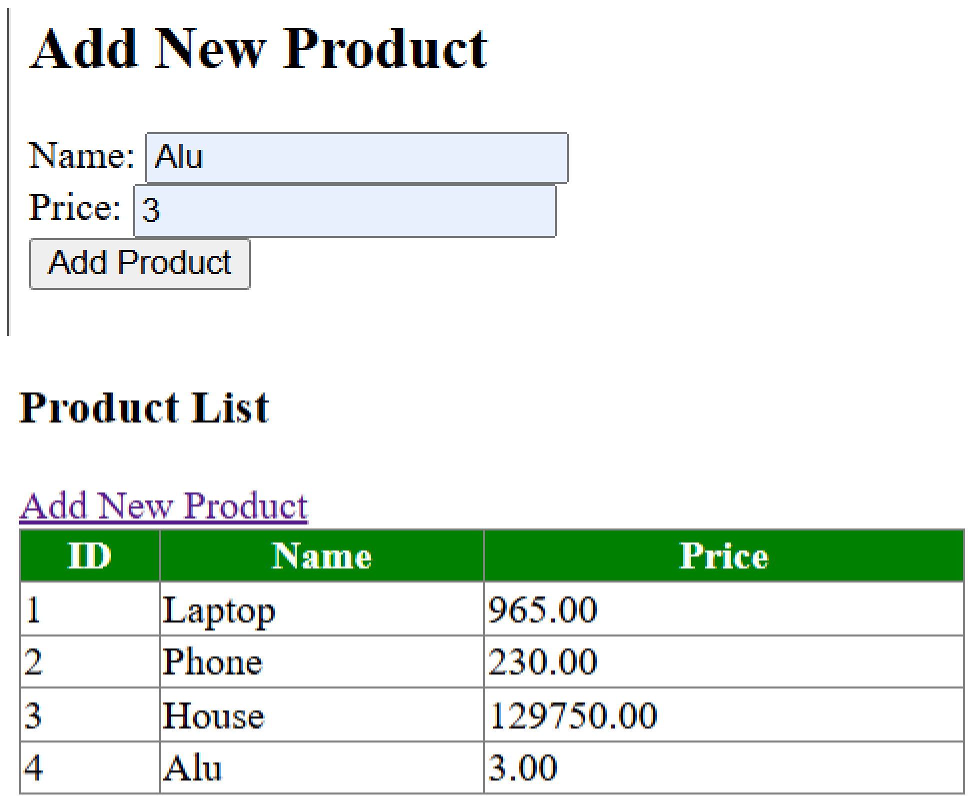
Visit: http://localhost/codeigniter4/public/product



Product List

Add New Product

ID	Name	Price



Add New Product

Name: Alu
Price: 3

Product List

Add New Product

ID	Name	Price
1	Laptop	965.00
2	Phone	230.00
3	House	129750.00
4	Alu	3.00

Banana Republic

"Banana republic" is a term, primarily used in political science, to describe a small, often poor, and politically unstable country whose economy is heavily reliant on a single crop, especially bananas, and also foreign funding and influence. The term originated in Central America, where U.S.-based banana companies exerted significant economic and political control in the early 20th century.

2 responses to "Banana Republic"

harry
June 11, 2025 Edit

Nice

Reply

author
June 11, 2025 Edit

thank you

Reply