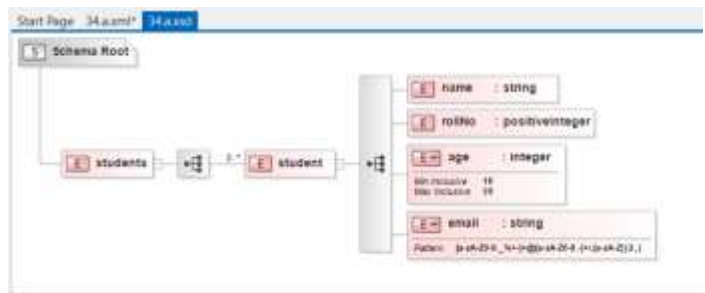
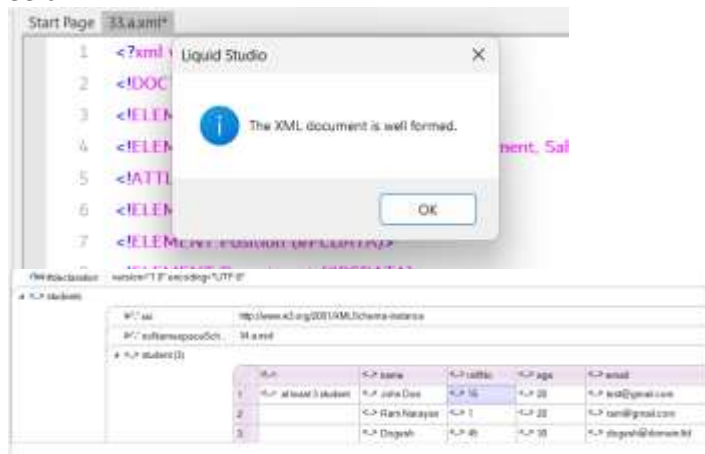
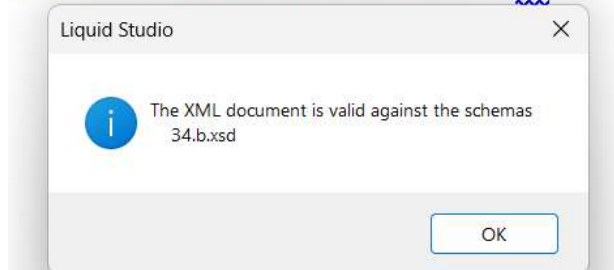


33.a

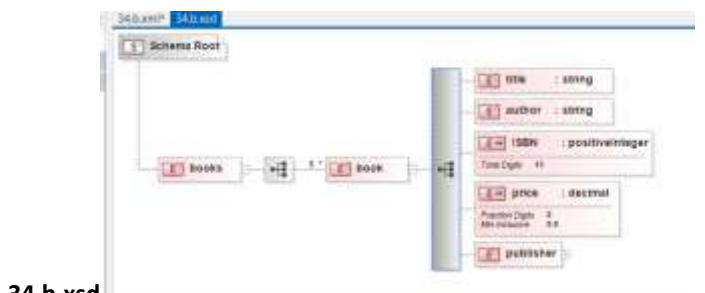


34.b.xml

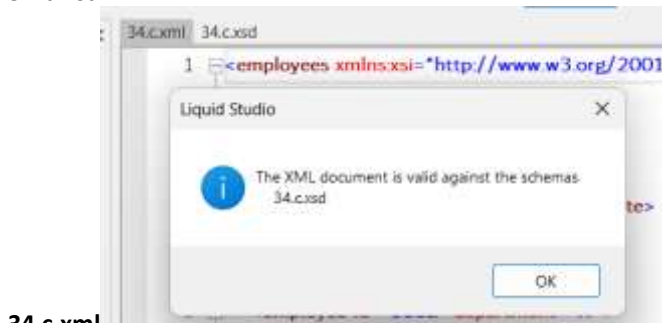
instance" xsi:noNamespaceSchemaLocation="34.b.xsd">



The screenshot displays the XML Schema (XSD) for 34.b.xml. The schema defines a root element 'books' containing a sequence of 'book' elements. Each 'book' element has attributes 'title', 'author', 'ISBN', 'price', and 'publisher', all of which are required and have specific data types (string, string, positiveInteger, decimal, and string respectively).



34.b.xsd

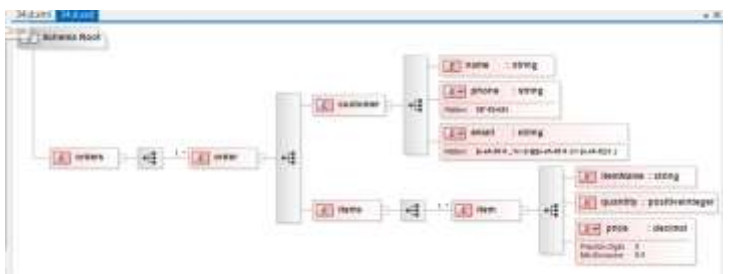
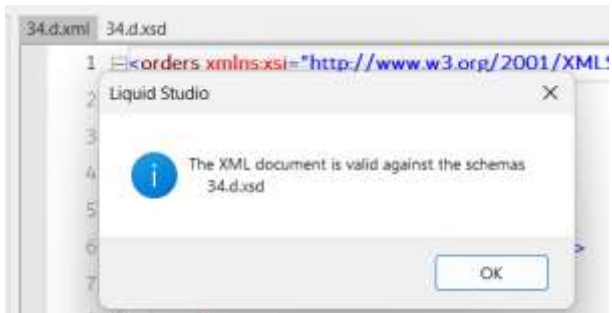


34.c.xml



The screenshot displays the XML Schema (XSD) for 34.c.xml. The schema defines a root element 'employees' containing a sequence of 'employee' elements. Each 'employee' element has attributes 'id', 'department', 'name', 'salary', and 'joiningDate', all of which are required and have specific data types (positiveInteger, string, string, decimal, and date respectively).

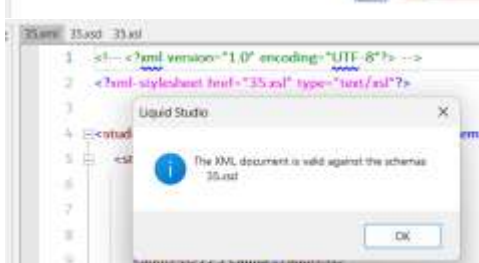
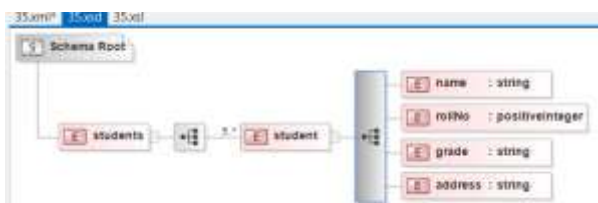
34.d.xsd



35.xml

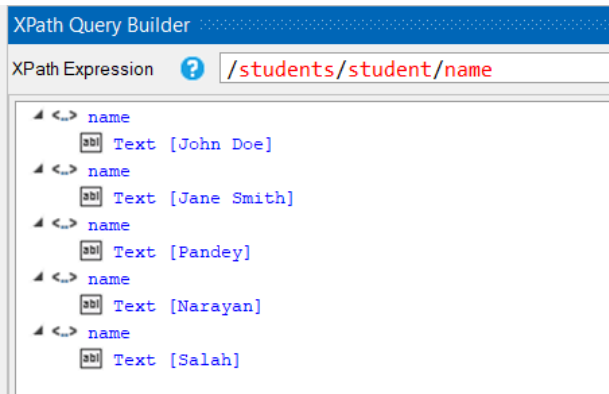
The screenshot displays the XML document for 35.xml. The document is a table with 4 columns: 'name', 'rollNo', 'grade', and 'address'. The table contains 5 rows of student data.

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



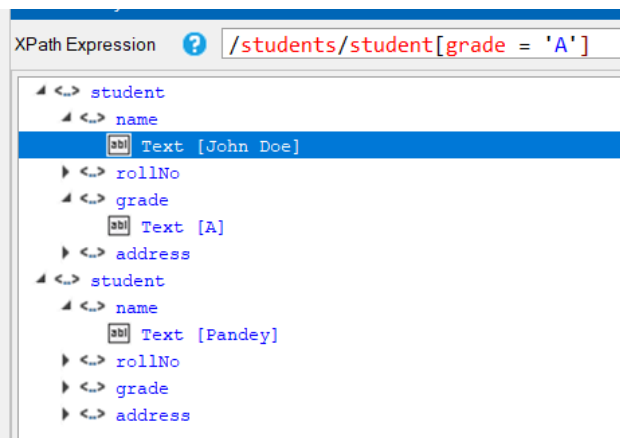
36.xsl (Xpath)

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

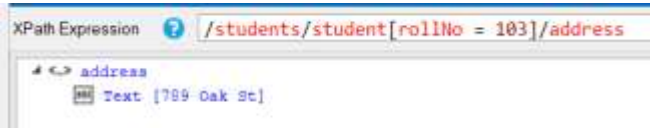


2. Students who have a grade 'A'.

```
<?xml version='1.0' encoding='UTF-8'>
<students xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <student>
    <name>John Doe</name>
    <rollNo>101</rollNo>
    <grade>A</grade>
    <address>123 Lalitpur</address>
  </student>
  <student>
    <name>Pandey</name>
    <rollNo>103</rollNo>
    <grade>A</grade>
    <address>789 Oak St</address>
  </student>
</students>
```



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



37.a.xquery Output: design

Book ID	Book Title	Author	Price	Year
1	Learning XML	Erik T. Ray	\$49.99	2006
2	Programming Python	Mark Lutz	\$49.99	2016
3	Clean Code	Robert C. Martin	\$49.99	2018
4	Fluent Python	Luciano Ramalho	\$49.99	2015
5	Introduction to Algorithms	Thomas H. Cormen	\$49.99	2009
6	Artificial Intelligence: A Modern Approach	Stuart Russell	\$49.99	2020

37.b.xquery

Book ID	Book Title	Author	Price	Year
1	Learning XML	Erik T. Ray	\$49.99	2006
2	Programming Python	Mark Lutz	\$49.99	2016
3	Clean Code	Robert C. Martin	\$49.99	2018
4	Fluent Python	Luciano Ramalho	\$49.99	2015
5	Introduction to Algorithms	Thomas H. Cormen	\$49.99	2009
6	Artificial Intelligence: A Modern Approach	Stuart Russell	\$49.99	2020

37.c.xquery

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

```
<?xml version='1.0' encoding='UTF-8'>
<books>
  <book>
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <price>$49.99</price>
    <year>2006</year>
  </book>
  <book>
    <title>Programming Python</title>
    <author>Mark Lutz</author>
    <price>$49.99</price>
    <year>2016</year>
  </book>
  <book>
    <title>Clean Code</title>
    <author>Robert C. Martin</author>
    <price>$49.99</price>
    <year>2018</year>
  </book>
  <book>
    <title>Fluent Python</title>
    <author>Luciano Ramalho</author>
    <price>$49.99</price>
    <year>2015</year>
  </book>
  <book>
    <title>Introduction to Algorithms</title>
    <author>Thomas H. Cormen</author>
    <price>$49.99</price>
    <year>2009</year>
  </book>
  <book>
    <title>Artificial Intelligence: A Modern Approach</title>
    <author>Stuart Russell</author>
    <price>$49.99</price>
    <year>2020</year>
  </book>
</books>
```

38.a.php

PHP Array

Indexed Array

- 2
- 4
- 6
- 10

Associative Array

- name : joy
- age : 22
- gender : male
- hobby : sports

38.b.php

Name:

Age:

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.e.php

Car 1 Details:

Brand: Toyota
 Color: Red

Car 2 Details:

Brand: Honda
 Color: Blue

38.c.php

User Registration Form

Full Name:

Email:

Username:

Password:

Confirm Password:

User Registration Form

Registration successful! Welcome, Mahadev.

Full Name:

Email:

Username:

Password:

Confirm Password:

38.d

Login Page

Username:

Password:

☒ Remember Me

Welcome, admin!

You have successfully logged in.

[Logout](#)

38.f (crud app)

Create New User

Name:

Email:

[View Users](#)

Update User

Name:

Email:

[Back to List](#)

User List

[Add New User](#)

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

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

40 index.php

Product Order Form

Customer Name:

Product:

Quantity:

Order Summary

Customer Name: sailesh

Product: laptop

Quantity: 3

Unit Price: \$1000

Total Price: \$3000

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

Read File Using AJAX

Load Text

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

39.b.html

AJAX GET Request with PHP

unknown stranger

Send

Hello, unknown stranger! This response is from PHP.

39.c. html

AJAX POST Request with PHP

Jhalakman

Send

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

39. e(database setup)

Select a Product

Smartphone ▼

Smartphone

Price: \$300.00

Description: An Android smartphone.

39.f

Live Search using AJAX

o

bot

tom

robert

39.d. html

jQuery AJAX POST Request

devkota

Send Data

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

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>
```

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:

Price:

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](#)