**UNIVERSITY OF TECHNOLOGY, MAURITIUS**

**SCHOOL OF INNOVATIVE TECHNOLOGIES AND ENGINEERING**

**DEPARTMENT OF INDUSTRIAL SYSTEM ENGINEERING**

**Module Name: Web Services**

**Module Code: WAT2124C**

**Cohort: BCNS/ 19A/ FT**

**Student Name: Arouven POOLIAN**

**Student ID: 1903\_16737**

Table of Contents

[Chapter 1: Introduction 1](#_Toc78381336)

[Chapter 2: Design and implementation 2](#_Toc78381337)

[2.1: Core functionalities 2](#_Toc78381338)

[2.1.1: Functionalities implemented 2](#_Toc78381339)

[2.1.2: Structure of files on servers 2](#_Toc78381340)

[2.2: Additional functionalities 3](#_Toc78381341)

[2.2.1: Headers 3](#_Toc78381342)

[2.2.2: How to use 3](#_Toc78381343)

[2.2.3: Scalable API 4](#_Toc78381344)

[2.2.4: XML and JSON format 6](#_Toc78381345)

[2.3: Client website 8](#_Toc78381346)

[2.3.1: JSON format 9](#_Toc78381347)

[2.3.2: XML format 11](#_Toc78381348)

[2.3.3: JavaScript 12](#_Toc78381349)

[Chapter 3: Testing 14](#_Toc78381350)

[3.1: API Testing 14](#_Toc78381351)

[3.1.1: How to use page 14](#_Toc78381352)

[3.1.2: Book not found 14](#_Toc78381353)

[3.1.3: Bad request to the API page 15](#_Toc78381354)

[3.1.4: Request XML format 15](#_Toc78381355)

[3.1.5: Request JSON format 16](#_Toc78381356)

[3.1.6: Request for part of a word and several queries at once 16](#_Toc78381357)

[3.2: Website Testing 17](#_Toc78381358)

[3.2.1: Link without query 17](#_Toc78381359)

[3.2.2: Build URL as per query and fill textboxes 17](#_Toc78381360)

[3.2.3: Multiple queries & part of a word in column 18](#_Toc78381361)

[3.2.4: Detail page 19](#_Toc78381362)

[3.2.5: XML and JSON supports 19](#_Toc78381363)

[3.2.6: Error message 20](#_Toc78381364)

[Chapter 4: Conclusion 21](#_Toc78381365)

[Annex A: myjs.js 22](#_Toc78381366)

[Annex B: display.php 25](#_Toc78381367)

[Annex C: database.php 29](#_Toc78381368)

[Annex D: credentials.php 30](#_Toc78381369)

[Annex E: queryAPI.php 30](#_Toc78381370)

[Annex F: myapi.php 31](#_Toc78381371)

[Annex G: index.php 36](#_Toc78381372)

[Annex H: detail.php 43](#_Toc78381373)

[Annex I: bookstore.sql 50](#_Toc78381374)

**List of Figures**

[Figure 2.1: Function to add headers. 3](#_Toc78381375)

[Figure 2.2: Code for how to use. 3](#_Toc78381376)

[Figure 2.3: Push to array. 4](#_Toc78381377)

[Figure 2.4: WHERE and AND added to subquery string. 5](#_Toc78381378)

[Figure 2.5: Full substring output. 5](#_Toc78381379)

[Figure 2.6: Code to output in JSON format. 6](#_Toc78381380)

[Figure 2.7: Code to output in XML format. 7](#_Toc78381381)

[Figure 2.8: User-defined function xml\_encode 7](#_Toc78381382)

[Figure 2.9: Add php file to index page. 8](#_Toc78381383)

[Figure 2.10: Default constructor of display class. 8](#_Toc78381384)

[Figure 2.11: JSON display function. 9](#_Toc78381385)

[Figure 2.12: Filler function. 10](#_Toc78381386)

[Figure 2.13: XML Display function. 11](#_Toc78381387)

[Figure 2.14: Verify format. 12](#_Toc78381388)

[Figure 2.15: Build URL method. 12](#_Toc78381389)

[Figure 2.16: Fill elements. 13](#_Toc78381390)

[Figure 3.1: How to use page. 14](#_Toc78381391)

[Figure 3.2: No book found. 14](#_Toc78381392)

[Figure 3.3: Bad request. 15](#_Toc78381393)

[Figure 3.4: Request for XML format. 15](#_Toc78381394)

[Figure 3.5: Request for JSON format 16](#_Toc78381395)

[Figure 3.6: Request for part of a word and several queries at once. 16](#_Toc78381396)

[Figure 3.7: Link without query. 17](#_Toc78381397)

[Figure 3.8: Build URL. 17](#_Toc78381398)

[Figure 3.9: Fill textboxes as per link. 18](#_Toc78381399)

[Figure 3.10: Multiple queries & part of a word in column 18](#_Toc78381400)

[Figure 3.11: Detailed page. 19](#_Toc78381401)

[Figure 3.12: JSON support. 19](#_Toc78381402)

[Figure 3.13: XML support. 19](#_Toc78381403)

[Figure 3.14: Error message. 20](#_Toc78381404)

List of Acronyms

API - Application Program Interface 1

HTML - HyperText Markup Language 10

ISBN - International Standard Book Number 6

JSON - JavaScript Object Notation 1

OOP - Object-Oriented Programming 6

PHP - Hypertext Preprocessor 1

SQL - Structured Query Language 5

URL - Uniform Resource Locator 4

XML - eXtensible Markup Language 1

# Introduction

To have some functionalities on the same page, a website is built to query a server and receive the JavaScript Object Notation (JSON) or extensible markup language (XML) file and display the required data. There will be a rest web service application program interface (API) using Hypertext Preprocessor (PHP).

# Design and implementation

## Core functionalities

### Functionalities implemented

The core functionalities developed on the API are also implemented in the client’s website. They are as follows:

* Get books information by title
* Get books information by author
* Get books information by publication year
* Get books information by author and category
* Get books information by author and publication year

Rather than the traditional approach (SELECT \* FROM bookstore.books WHERE title=’harry potter’ ;), another type of approach has been used which will make the API more scalable and work better with multiple query strings. This will be explained further in section 2.2.3: Scalable API.

### Structure of files on servers

* API Server
  + Credentials.php
  + Database.php
* Client’s Server
  + Index.php
  + Detail.php
  + Display.php
  + queryAPI.php

## Additional functionalities

### Headers

In the case of the bookstore, either the books are found or not. But there is another one that can arise; bad request. To help programmability, status code, message, and total book in the database are prepended to the response. A function addStatusToData ($status, $statusMessage, $count, $data) was created (as shown in Figure 2.1) to add the above before the data/response.

The status code and messages are as follow:

* 200 – Successful (represent book found in the bookstore)
* 400 – Bad Request (represent wrong URL query)
* 404 – Not found (represent book not found in the bookstore)

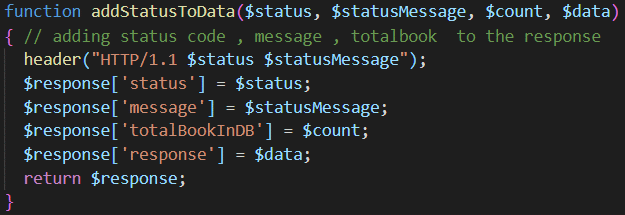


Figure .: Function to add headers.

### How to use

If there is no query in the URL, it is assumed that the client doesn’t know how to use the API. Figure 2.2 shows part of the code to display the how to use page to the user. This will be further explained in Chapter 3: Testing.

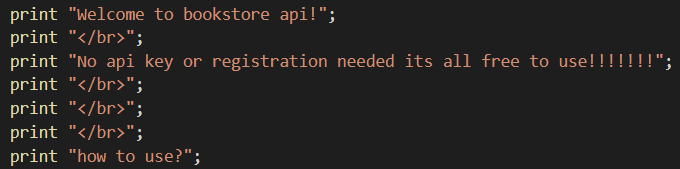


Figure .: Code for how to use.

### Scalable API

As mentioned before, this will make the API more dynamic. The API will be programmed to allow any of its users to use it without any restriction in the number of queries passed as requests. Annex F: myapi.php has the full code for any reference.

The approach is as follows:

1. Get the main query – “SELECT \* FROM books”
2. Combine/concatenate it with the subquery (if any). A function was built to perform this task. It requires all the parameters of the Uniform Resource Locator (URL) to perform its task and return the subquery.

 $this->query = "SELECT \* FROM books" . $this->subquery($params);

**Exploring subquery ($params) method/function.**

The principle is that every query parameter should be prepended with the word “AND”. To achieve this, an array is created. All the queries that required the keyword will be pushed to that array. Figure 2.3 shows two examples of parameters that should be prepended.

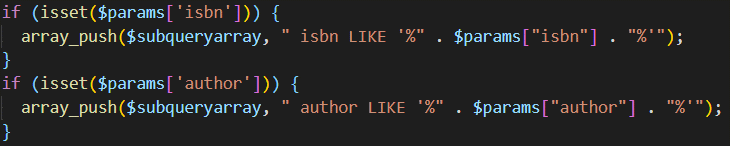


Figure .: Push to array.

Then the next step is to add the “WHERE” keyword. This will be done only if there is something in the array. After that, a Foreach loop will do the job of adding the “AND” in front of each string in the array. The result of $subquery in Figure 2.4 with input author=harry&isbn=001 will be “ WHERE AND author LIKE '%harry%' AND isbn LIKE %001%”. The keyword LIKE will allow the user to insert part of the word to query.

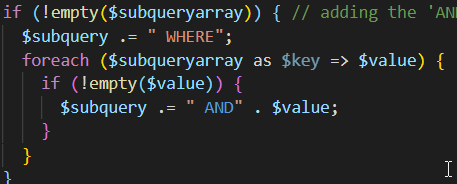


Figure .: WHERE and AND added to subquery string.

As the MYSQL syntax does not require any “AND” in front of “ORDER BY” and “LIMIT”, they (order by and limit) were not pushed to the array. For example, author=harry&isbn=001&orderby=rating&limit=5 will output structured query language (SQL) $subquery = “ WHERE AND author LIKE ‘%harry%’ AND isbn LIKE %001% ORDER BY rating LIMIT 5”. The last line of that function will solve the problem of the “WHERE AND” in front of the subquery by replacing it with “WHERE” and add the ending of the SQL query as shown in Figure 2.5. This will lead to output “ WHERE author LIKE ‘%harry%’ AND isbn LIKE %001% ORDER BY rating LIMIT 5;”

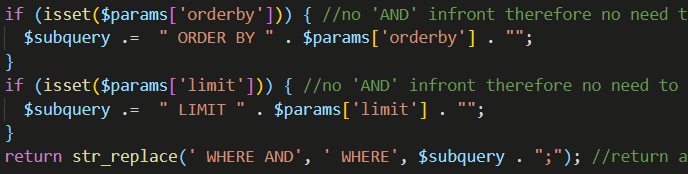


Figure .: Full substring output.

Therefore,

 $this->query = "SELECT \* FROM books" . $this->subquery($params);

Will contain “SELECT \* FROM books WHERE author LIKE ‘%harry%’ AND isbn LIKE %001% ORDER BY rating LIMIT 5;”

Using this type of approach, the following can be accomplished:

* Search for a word that contains some characters in the title, author, International Standard Book Number (ISBN), category, language, and year published.
* Order the result by author, category, etc.
* Limit the result of the search to x number. Where x is a number defined by the user.
* Above all, the user is now unlimited with the column to query. He/she can query more than one for example author and category and much more.

The query is then executed and return in a form of an array. This array will be converted to either XML or JSON and outputted to the user.

### XML and JSON format

By using Object-oriented programming (OOP) and some functions/methods, both XML and JSON can be outputted from the API.

#### JSON format

Figure 2.6 show the code that will output the data in JSON format. The header is added and PHP already has its function to output in JSON (json\_encode). This will take the array outputted from the SQL query and convert it to JSON format.

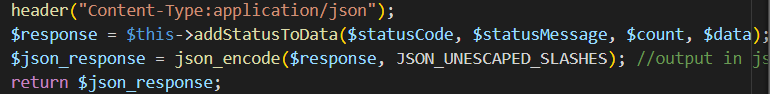


Figure .: Code to output in JSON format.

#### XML format

XML on the other hand is more complicated as there is no inbuilt method to encode in XML format. But the principle stays the same. A user-defined function should be created to perform this task. Figure 2.7 shows how the user-defined will look like.

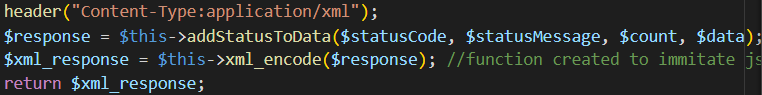


Figure .: Code to output in XML format.

In Figure 2.8, the XML header is added together with the first element. Then the array will be retrieved from the SQL query will be converted to XML with bookX tag(s) (X start from 0 to total record in array) and inserted between the bookstore tags.

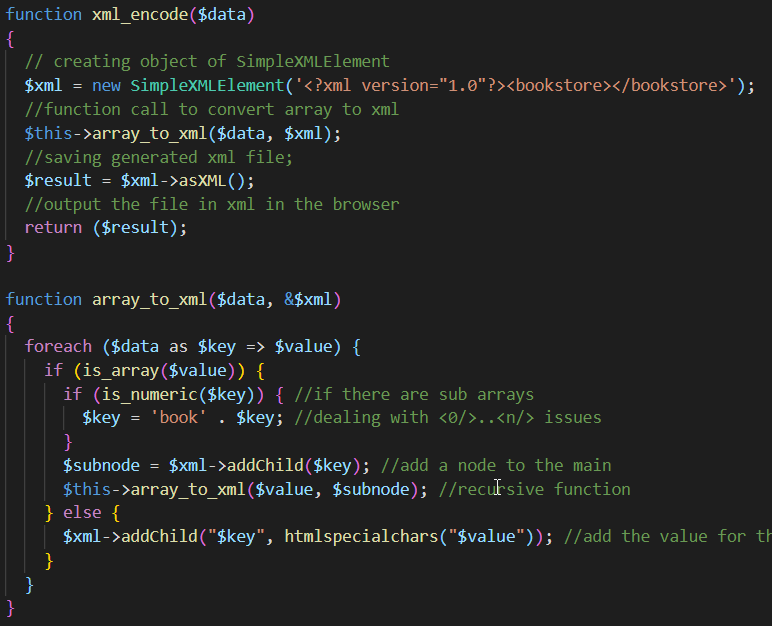


Figure .: User-defined function xml\_encode

## Client website

The client website supports both XML and JSON responses. To use the template efficiently, all codes were put on a separate file and then called in the pages as shown in Figure 2.9. It also indicates that the same query from the client URL will be pass through the API.

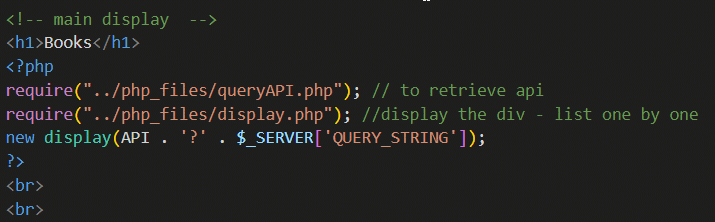


Figure .: Add php file to index page.

Figure 2.10 shows the default constructor of the display class. The constructor will talk the API and detailed page (will be used in books detailed page) as parameters. It will then gather the client’s query in the client’s URL and execute the appropriate function if the format detected is JSON or XML.



Figure .: Default constructor of display class.

### JSON format

This function shown in Figure 2.11 will get the contents from the API and decode the retrieved JSON formatted data. It will then verify if the response has a status code of 200 and then start to display if it is not 200 an error message will be displayed. Then it will call the filler function for each record from the $data (JSON decode function).

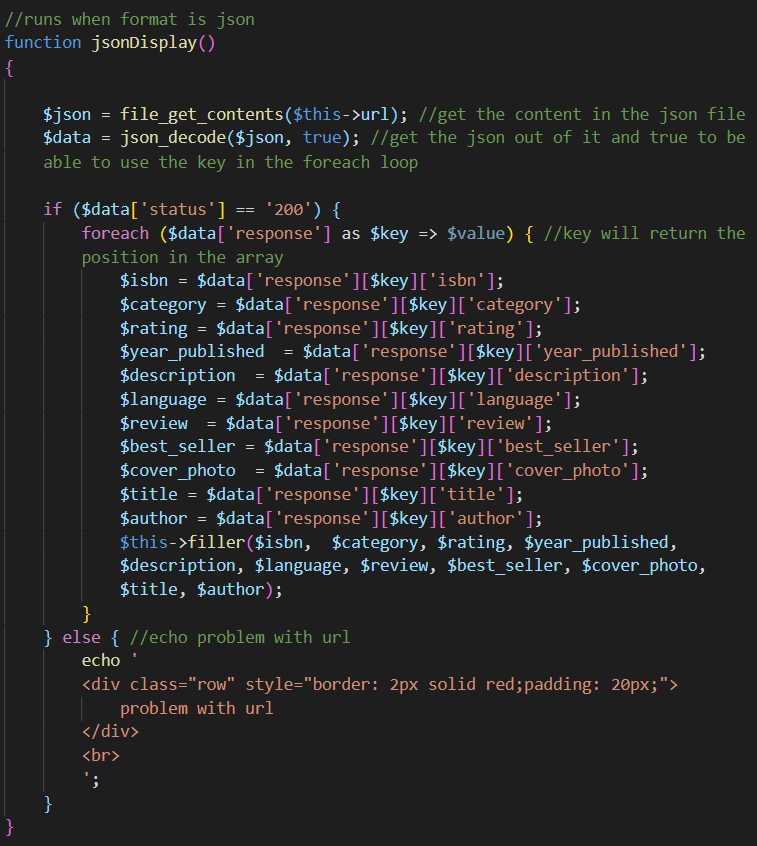


Figure .: JSON display function.

Figure 2.12 shows the filler function. This function will take all previously obtained values and put them in a hypertext markup language (HTML) format. Here the detailed page came into play. There are only two pages on the client website, one to display the list of books and the other to display all the details of the book.

|  |
| --- |
|  |
|  |

Figure .: Filler function.

### XML format

Figure 2.13 shows the XML Display function. It is the same principle as in section 2.3.1: JSON format. It will load the API, verify if its code is 200, and call the filler function. If it is not 200, an error message will be displayed.

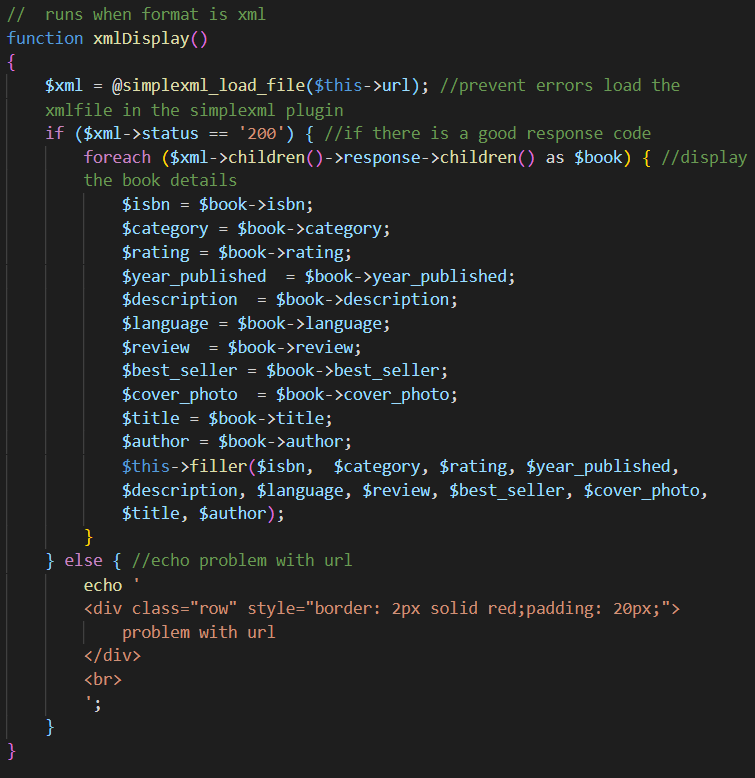


Figure .: XML Display function.

### JavaScript

The piece of code shown in Figure 2.14 is responsible to verify if the client website has format in its query. This is vital as without this no book will be displayed. If there is no format in the query, javascript will add a default format to it (the chosen default format is JSON). Then the client website will be loaded with the new URL.

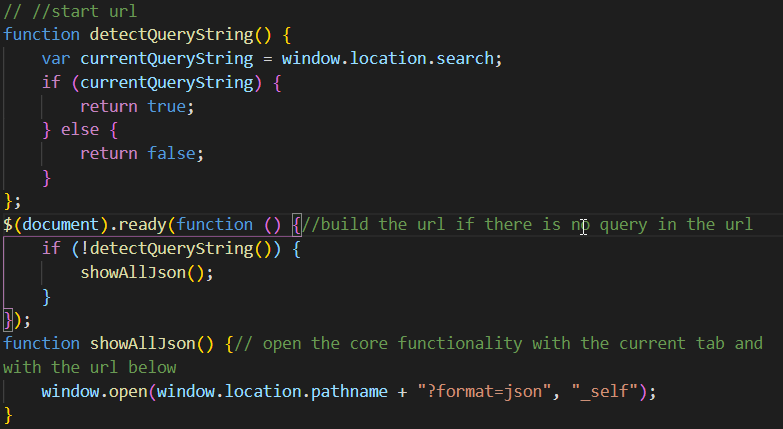


Figure .: Verify format.

Figure 2.15 shows the search system of the client website. On clicking on the search button, the query will be built and then opens. While opening, it will be the new link that will be in the URL (built URL). This will facilitate the PHP API query and also reduce the server-side processing time.



Figure .: Build URL method.

Another function of javascript is the fill elements (see Figure 2.16). This snippet will take the parameters in the website URL and retrieve the queries. Each query will be associated with its respective textboxes or dropdown menu.



Figure .: Fill elements.

# Testing

## API Testing

### How to use page

The how to use page appears when the link does not have any query as shown in Figure 3.1.

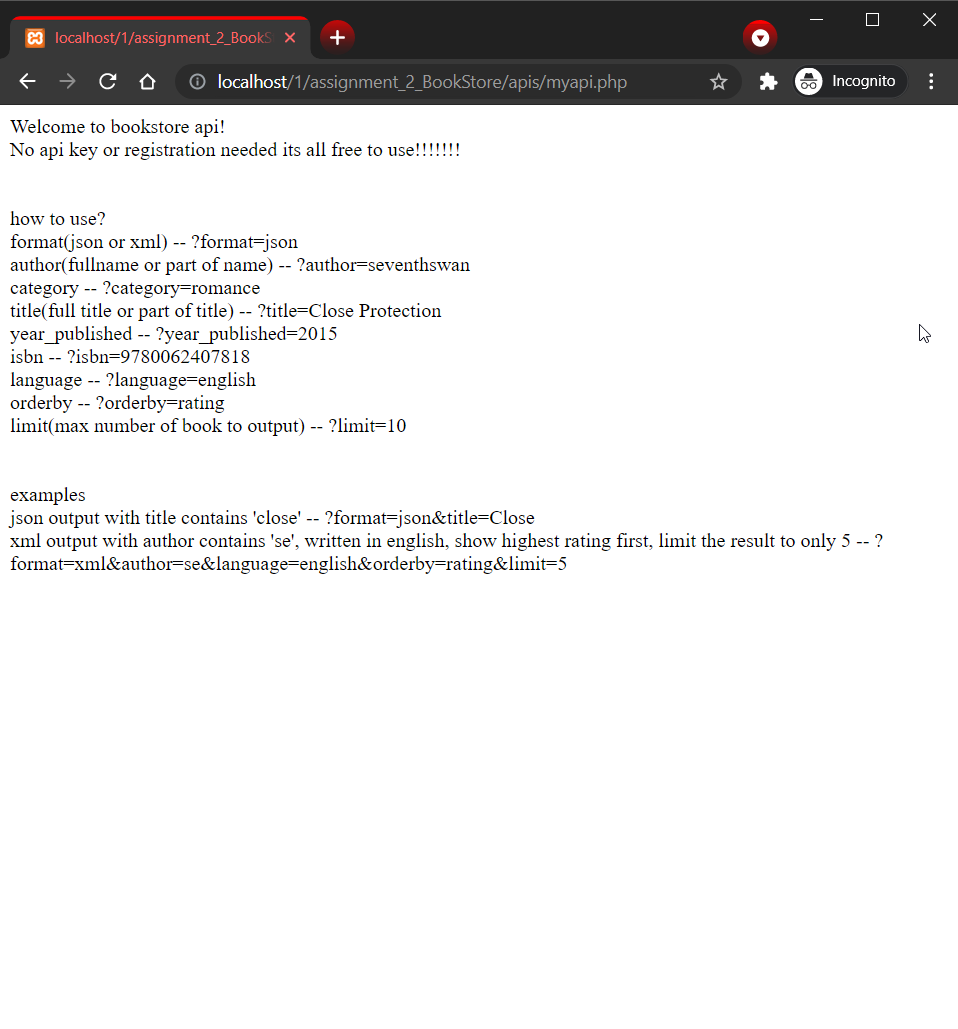


Figure .: How to use page.

### Book not found

Figure 3.2 shows the error code 404 for no book found. This is because no author has “thorn” in their name in the database.

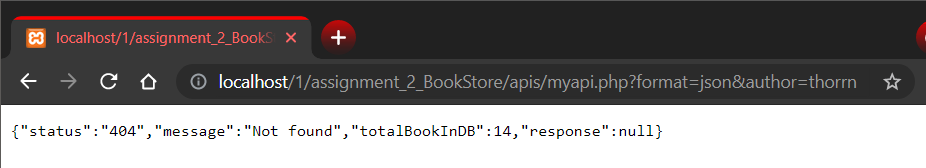


Figure .: No book found.

### Bad request to the API page

Figure 3.3 shows what happened if there is a bad request. In this case a wrong format as a query.

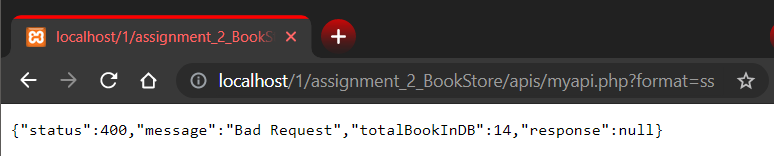


Figure .: Bad request.

### Request XML format

Figure 3.4 shows the output when requesting for the XML format.

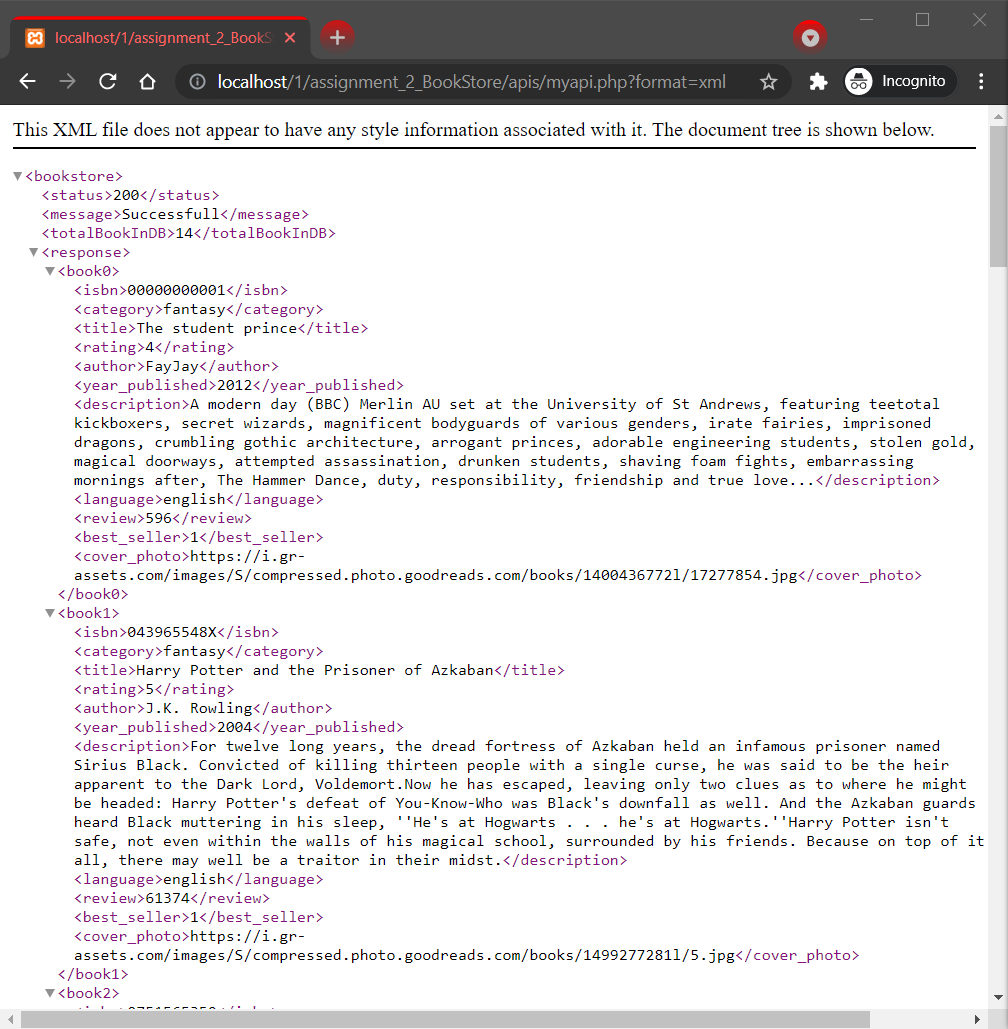


Figure .: Request for XML format.

### Request JSON format

Figure 3.5 shows the output when requesting the JSON format.

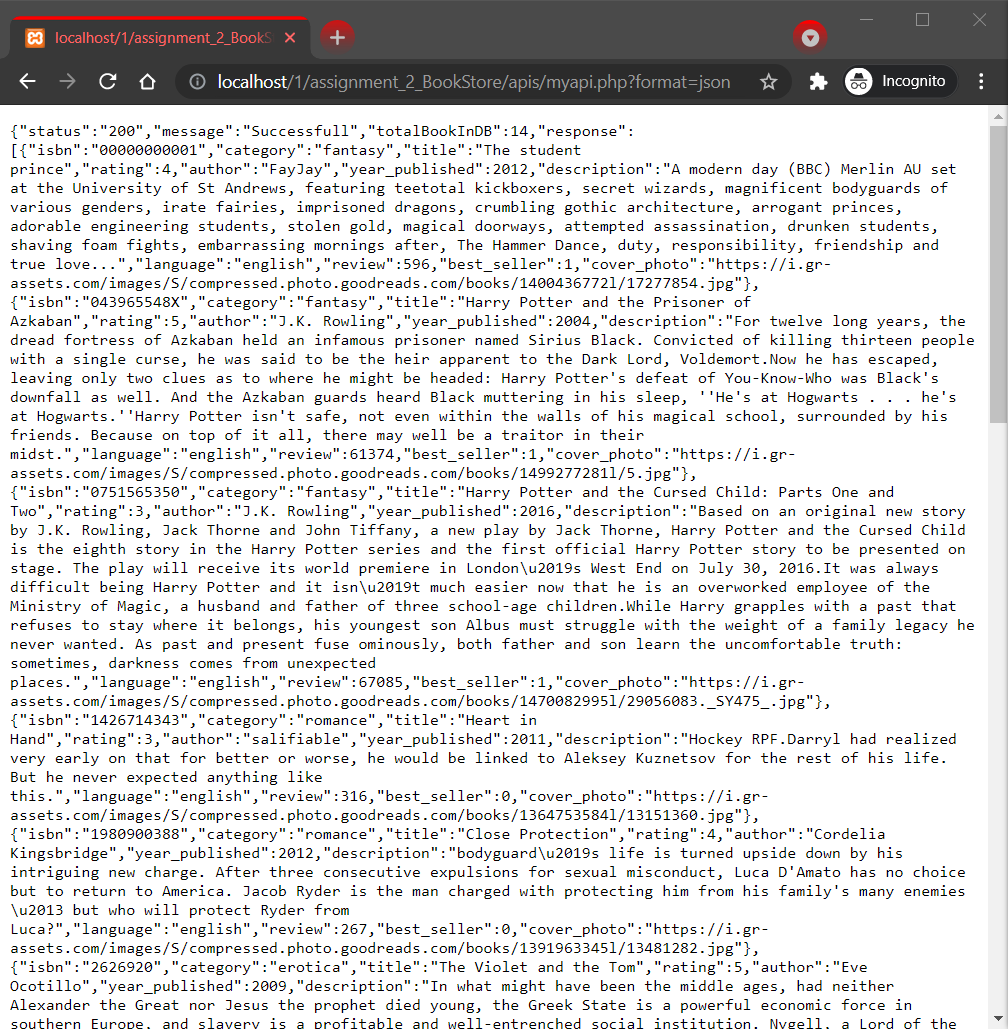


Figure .: Request for JSON format

### Request for part of a word and several queries at once

The status code 200 with the message Successful and the total book in the database are displayed in every request. Figure 3.6 illustrates the query for part of the title “harry” and multiple queries such as language, limit, and order by. In this database, the title of the books that have “harry” in them are the “Harry Potter” s books.

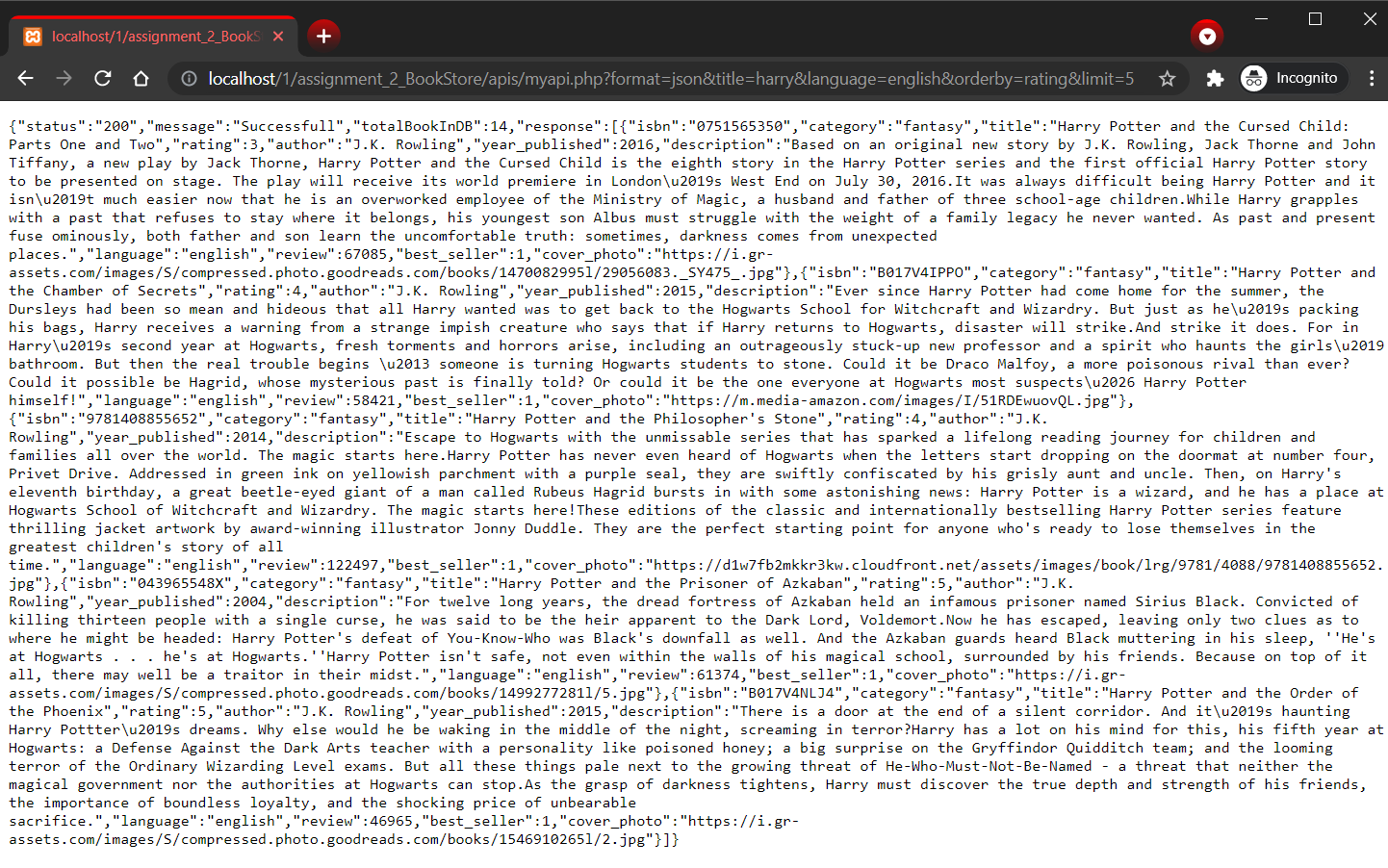


Figure .: Request for part of a word and several queries at once.

## Website Testing

### Link without query

As intended, the “?format=json” was added automatically by javascript as shown in Figure 3.7.

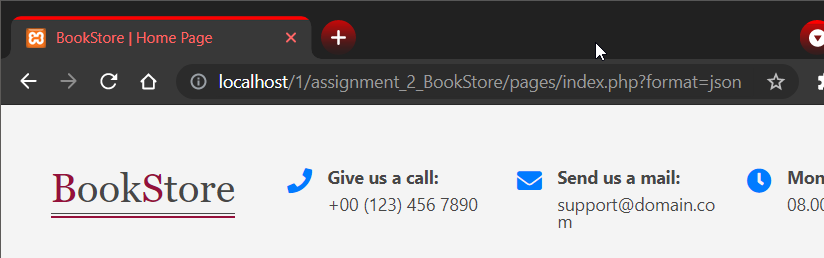


Figure .: Link without query.

### Build URL as per query and fill textboxes

The queries will be built by clicking on the search button. Figure 3.8 shows the browser’s link changed after and before clicking on the search button.

|  |  |
| --- | --- |
|  |  |

Figure .: Build URL.

The same thing will apply to querying directly the URL. The textboxes will be filled automatically as shown in Figure 3.9.

|  |  |
| --- | --- |
|  |  |

Figure .: Fill textboxes as per link.

### Multiple queries & part of a word in column

Figure 3.10 shows an example of multiple queries. The queries are; title=harry, language=English, orderby=rating, limit=2, and format=json. The part of the title harry can be found in the database. They are “harry potters” books.

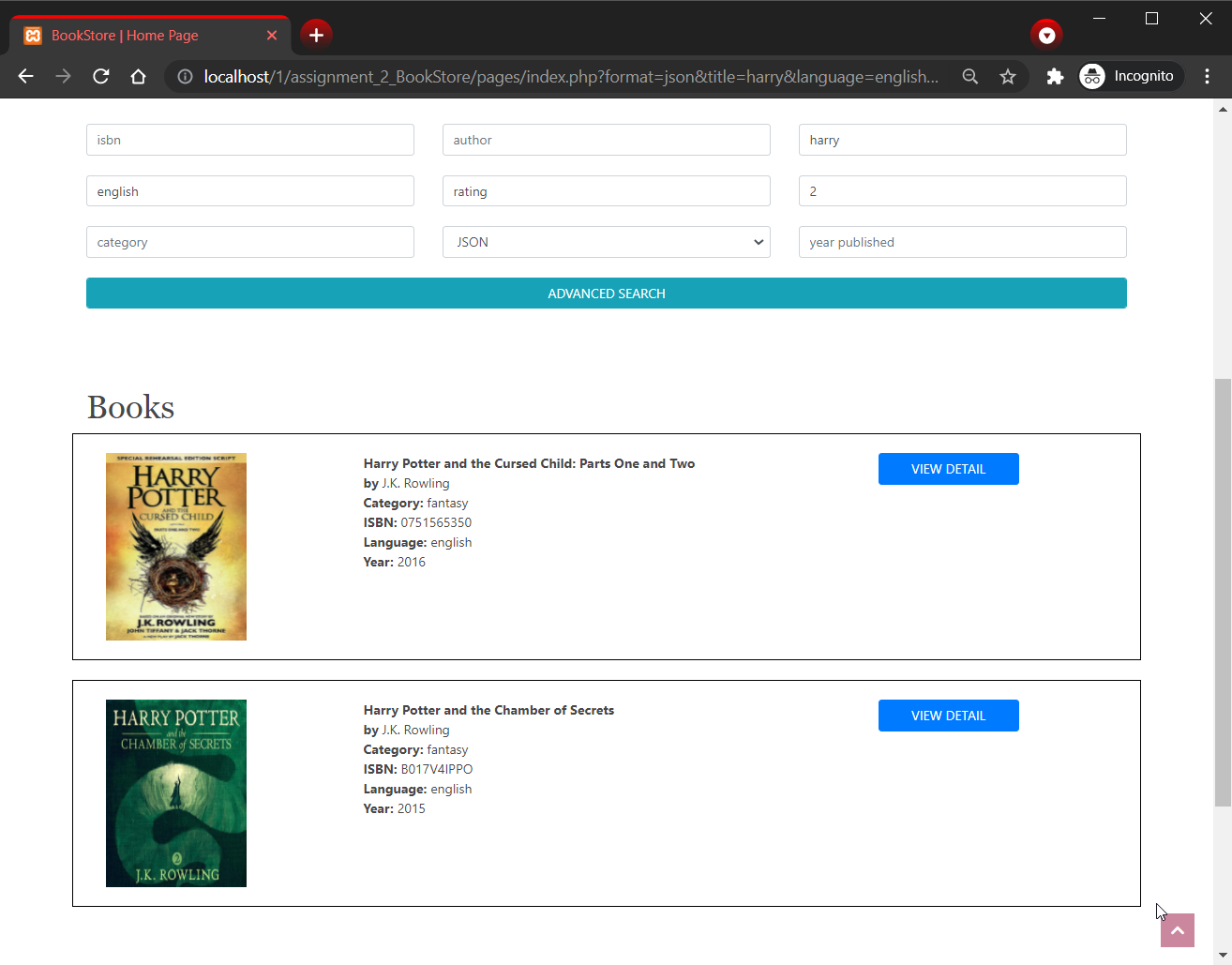


Figure .: Multiple queries & part of a word in column

### Detail page

Figure 3.11 shows the details (Description, rating, review, best seller) of the book searched for that appears once the View detail button was clicked.

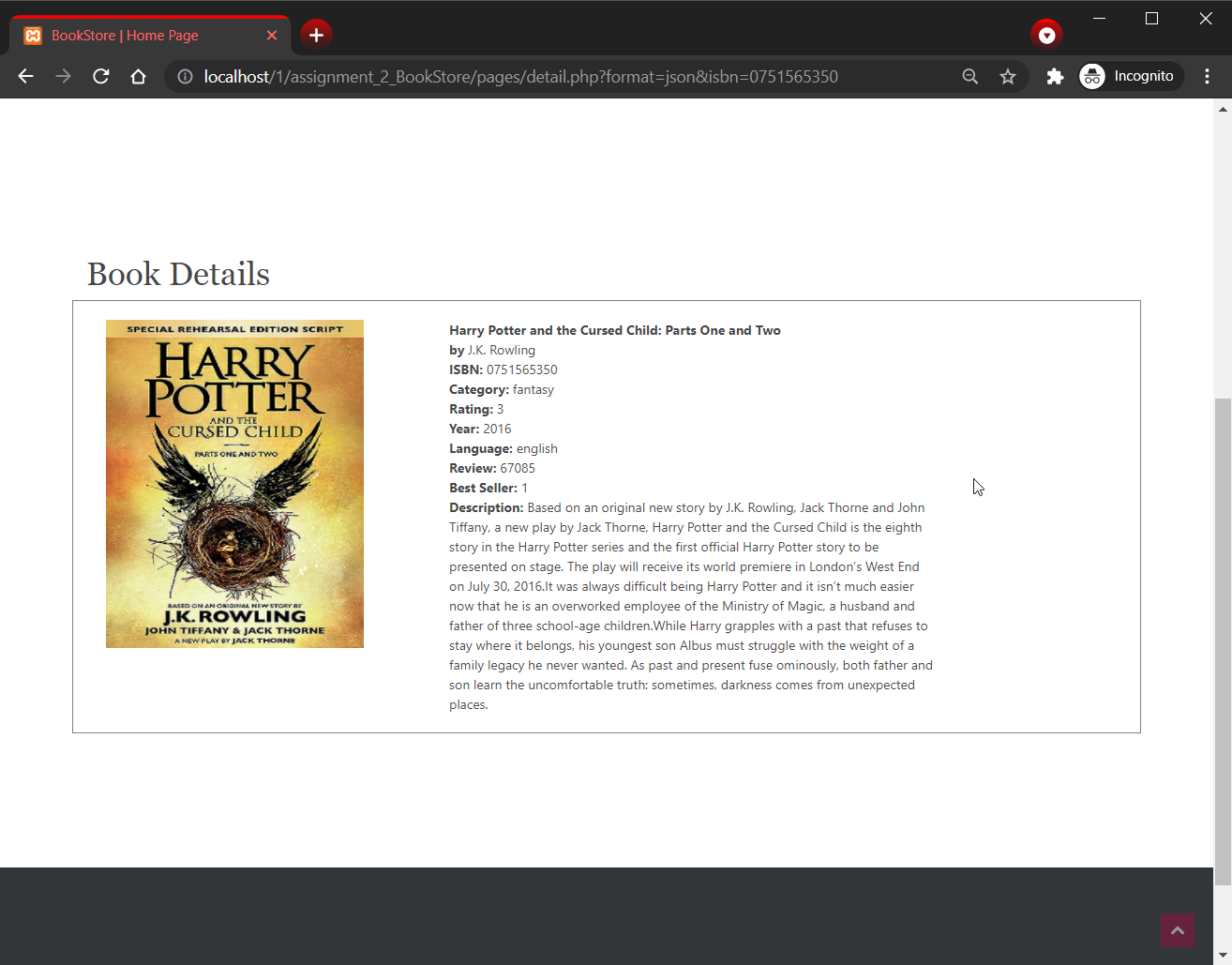


Figure .: Detailed page.

### XML and JSON supports

Figure 3.12 shows that the client website support JSON responses from the API and Figure 3.13 shows that it also supports XML responses from the API.

|  |  |
| --- | --- |
| Figure .: JSON support. | Figure .: XML support. |

### Error message

Figure 3.14 shows the error message that will appear when no book is found or problem with the URL.

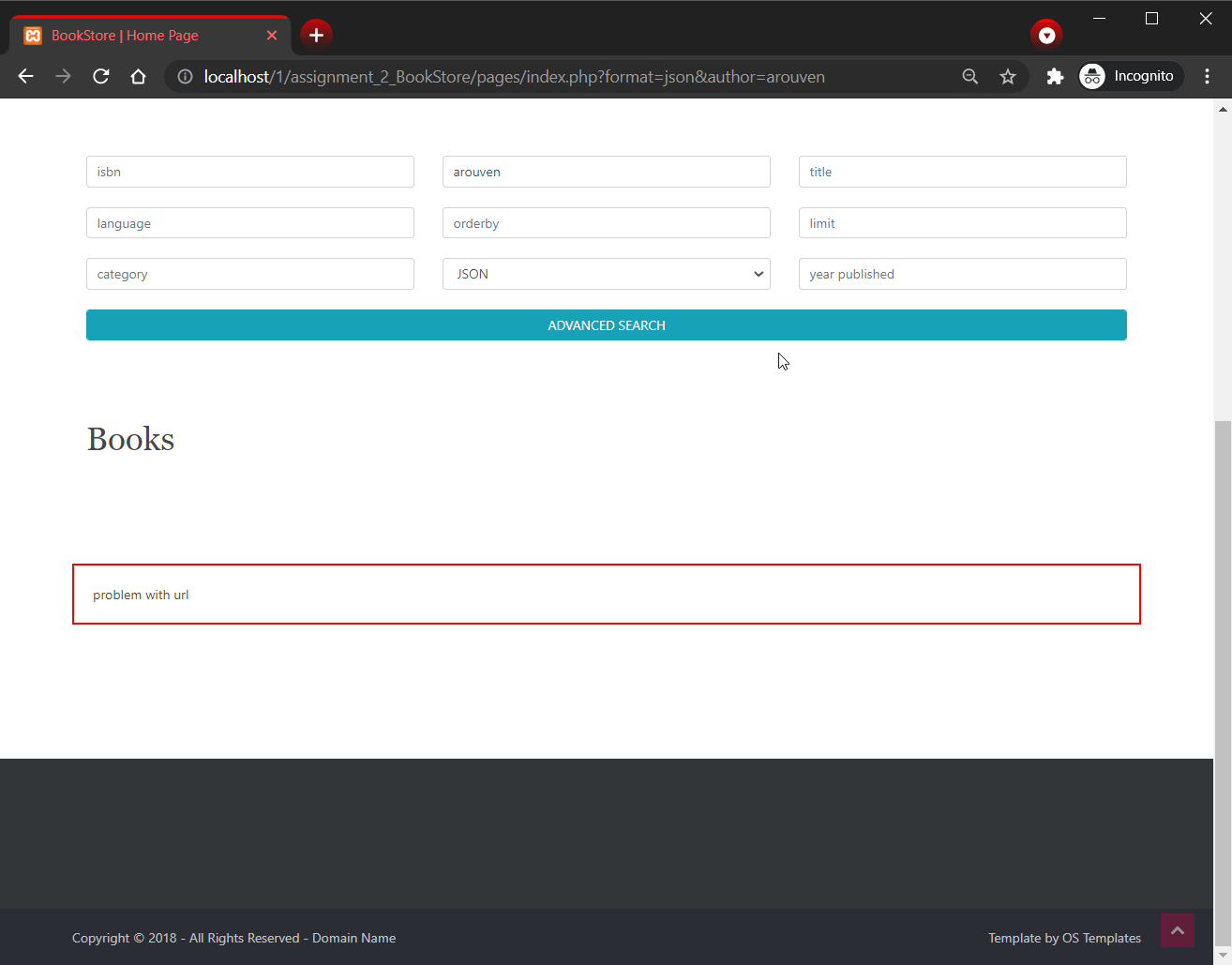


Figure .: Error message.

# Conclusion

As seen in the Chapter 3: Testing, the whole project works well. As an improvement, a user basket could be added. This will help the book buyers to know how much they will buy or improve checkout. Another improvement will be a notification that will tell the buyer when the stock is going down or in case of promotion.

The link to the GitHub file:

<https://github.com/Arouven/webservices>

# Annex A: myjs.js

// //start url

function detectQueryString() {

    var currentQueryString = window.location.search;

    if (currentQueryString) {

        return true;

    } else {

        return false;

    }

};

$(document).ready(function () {//build the url if there is no query in the url

    if (!detectQueryString()) {

        showAllJson();

    }

});

function showAllJson() {// open the core functionality with the current tab and with the url below

    window.open(window.location.pathname + "?format=json", "\_self");

}

function buildURL() {

    var output = window.location.pathname + "?";//build the initial url with the file.php followed by the query

    // get appropriate values from elements

    var format = $('#format').val();

    var isbn = $("#isbn").val();

    var author = $("#author").val();

    var category = $("#category").val();

    var title = $("#title").val();

    var language = $("#language").val();

    var orderby = $("#orderby").val();

    var limit = $('#limit').val();

    var year\_published = $('#year\_published').val();

    // build the url

    output += "format=" + format;

    if (isbn) {

        output += "&isbn=" + isbn;

    }

    if (author) {

        output += "&author=" + author;

    }

    if (category) {

        output += "&category=" + category;

    }

    if (title) {

        output += "&title=" + title;

    }

    if (year\_published) {

        output += "&year\_published=" + year\_published;

    }

    if (language) {

        output += "&language=" + language;

    }

    if (orderby) {

        output += "&orderby=" + orderby;

    }

    if (limit) {

        output += "&limit=" + limit;

    }

    window.open(output, "\_self");//open the url in the current tab

}

//end url

//start fill html elements

function fillElements() {

    var url = window.location.href;// get current url

    url = new URL(url);//change text to url

    // get appropriate queries from url

    var format = url.searchParams.get("format");

    var isbn = url.searchParams.get("isbn");

    var author = url.searchParams.get("author");

    var category = url.searchParams.get("category");

    var title = url.searchParams.get("title");

    var language = url.searchParams.get("language");

    var orderby = url.searchParams.get("orderby");

    var limit = url.searchParams.get("limit");

    var year\_published = url.searchParams.get("year\_published");

    //set the elements with the appropriate values from the query

    $('#format').val(format);

    if (isbn) {

        $("#isbn").val(isbn);

    }

    if (author) {

        $("#author").val(author);

    }

    if (category) {

        $("#category").val(category);

    }

    if (title) {

        $("#title").val(title);

    }

    if (language) {

        $("#language").val(language);

    }

    if (orderby) {

        $("#orderby").val(orderby);

    }

    if (limit) {

        $('#limit').val(limit);

    }

    if (year\_published) {

        $('#year\_published').val(year\_published);

    }

}

//end filling html elements

# Annex B: display.php

<?php

class display

{

    private $url;

    private $detailedPage;

    private $format;

    //default constructor

    function \_\_construct($url, $detailedPage = false)

    {

        $this->url = $url;

        $this->detailedPage = $detailedPage;

        parse\_str(strtolower($\_SERVER["QUERY\_STRING"]), $params); // store all queries in $params

        if (isset($params['format'])) {

            if ($params['format'] == 'json') {

                $this->format = 'json';

                $this->jsonDisplay();

            }

            if ($params["format"] == "xml") {

                $this->format = 'xml';

                $this->xmlDisplay();

            }

        }

    }

    //  runs when format is xml

    function xmlDisplay()

    {

        $xml = @simplexml\_load\_file($this->url); //prevent errors load the xmlfile in the simplexml plugin

        if ($xml->status == '200') { //if there is a good response code

            foreach ($xml->children()->response->children() as $book) { //display the book details

                $isbn = $book->isbn;

                $category = $book->category;

                $rating = $book->rating;

                $year\_published  = $book->year\_published;

                $description  = $book->description;

                $language = $book->language;

                $review  = $book->review;

                $best\_seller = $book->best\_seller;

                $cover\_photo  = $book->cover\_photo;

                $title = $book->title;

                $author = $book->author;

                $this->filler($isbn,  $category, $rating, $year\_published, $description, $language, $review, $best\_seller, $cover\_photo, $title, $author);

            }

        } else { //echo problem with url

            echo '

            <div class="row" style="border: 2px solid red;padding: 20px;">

                problem with url

            </div>

            <br>

            ';

        }

    }

    //runs when format is json

    function jsonDisplay()

    {

        $json = file\_get\_contents($this->url); //get the content in the json file

        $data = json\_decode($json, true); //get the json out of it and true to be able to use the key in the foreach loop

        if ($data['status'] == '200') {

            foreach ($data['response'] as $key => $value) { //key will return the position in the array

                $isbn = $data['response'][$key]['isbn'];

                $category = $data['response'][$key]['category'];

                $rating = $data['response'][$key]['rating'];

                $year\_published  = $data['response'][$key]['year\_published'];

                $description  = $data['response'][$key]['description'];

                $language = $data['response'][$key]['language'];

                $review  = $data['response'][$key]['review'];

                $best\_seller = $data['response'][$key]['best\_seller'];

                $cover\_photo  = $data['response'][$key]['cover\_photo'];

                $title = $data['response'][$key]['title'];

                $author = $data['response'][$key]['author'];

                $this->filler($isbn,  $category, $rating, $year\_published, $description, $language, $review, $best\_seller, $cover\_photo, $title, $author);

            }

        } else { //echo problem with url

            echo '

            <div class="row" style="border: 2px solid red;padding: 20px;">

                problem with url

            </div>

            <br>

            ';

        }

    }

    function filler($isbn,  $category, $rating, $year\_published, $description, $language, $review, $best\_seller, $cover\_photo, $title, $author)

    {

        if ($this->detailedPage) {

            echo '

            <div class="row" style="border: 1px solid grey;padding: 20px;">

                <div class="col-sm-4"><img src="' . $cover\_photo . '" style="width:275px;height:350px;" /></div>

                <div class="col-sm-6">

                    <b>' . $title . '</b>

                    <br>

                    <b>by</b> ' . $author . '

                    <br>

                    <b>ISBN: </b>' . $isbn . '

                    <br>

                    <b>Category: </b>' . $category . '

                    <br>

                    <b>Rating: </b>' . $rating . '

                    <br>

                    <b>Year: </b>' . $year\_published . '

                    <br>

                    <b>Language: </b>' . $language . '

                    <br>

                    <b>Review: </b>' . $review . '

                    <br>

                    <b>Best Seller: </b>' . $best\_seller . '

                    <br>

                    <b>Description: </b>' . $description . '

                    <br>

                </div>

            </div>

            <br>

            ';

        } else {

            echo '

            <div class="row" style="border: 1px solid black;padding: 20px;">

                <div class="col-sm-3"><img src="' . $cover\_photo . '" style="width:150px;height:200px;" /></div>

                <div class="col-sm-6">

                    <b>' . $title . '

                    <br>by </b>' . $author . '

                    <br><b>Category: </b>' . $category . '

                    <br><b>ISBN: </b>' . $isbn . '

                    <br><b>Language: </b>' . $language . '

                    <br><b>Year: </b>' . $year\_published . '

                </div>

                <div class="col-sm-3">

                    <button type="button"

                    class="btn btn-primary"

                    style="width: 150px;"

                    title="Open details"

                    onclick="location.href = \'http://localhost/1/assignment\_2\_BookStore/pages/detail.php?format=' . $this->format . '&isbn=' . $isbn . '\';">View Detail</button>

                </div>

            </div>

            <br>

            ';

        }

    }

}

# Annex C: database.php

<?php

require 'credentials.php';

class database

{

    private $conn;

    function \_\_construct()

    {

        $this->conn = $this->getConnection();

    }

    public function getConnection()

    {

        $conn = new mysqli(HOST, USERNAME, PASSWORD, DATABASENAME);

        if (mysqli\_connect\_errno()) {

            trigger\_error("Problem with connecting to database.");

        }

        $conn->set\_charset("utf8");

        return $conn;

    }

    /\*\*

     \* To get database results

     \*

     \* @param string $query

     \* @param string $paramType

     \* @param array $paramArray

     \* @return array

     \*/

    public function select($query)

    {

        $stmt = $this->conn->prepare($query);

        $stmt->execute();

        $result = $stmt->get\_result();

        if ($result->num\_rows > 0) {

            while ($row = $result->fetch\_assoc()) {

                $resultset[] = $row;

            }

        }

        if (!empty($resultset)) {

            return $resultset;

        }

    }

}

# Annex D: credentials.php

<?php

const HOST = 'localhost';

const USERNAME = 'root';

const PASSWORD = '';

const DATABASENAME = 'bookstore';

# Annex E: queryAPI.php

<?php

const API = 'http://localhost/1/assignment\_2\_BookStore/apis/myapi.php';

# Annex F: myapi.php

<?php

require 'database.php';

new api();

class api

{

  private $format;

  private $query;

  private $db;

  function \_\_construct()

  {

    // Use parse\_str() function to parse the

    // string passed via URL

    parse\_str(strtolower($\_SERVER["QUERY\_STRING"]), $params); // store all queries in $params

    if (isset($params['format'])) {

      if ($params['format'] == 'json') {

        $this->format = 'json'; //set format var to json

      }

      if ($params["format"] == "xml") {

        $this->format = 'xml'; //set format var to xml

      }

      $this->query = "SELECT \* FROM books" . $this->subquery($params); //generate the sql query and set the var query

      $this->db = new database(); //create the database class

      print $this->select($this->query); //retrieve the output from the query

    } else { //if there is a wrong query the page will display how to use to the user

      print "Welcome to bookstore api!";

      print "</br>";

      print "No api key or registration needed its all free to use!!!!!!!";

      print "</br>";

      print "</br>";

      print "</br>";

      print "how to use?";

      print "</br>";

      print "format(json or xml) -- ?format=json";

      print "</br>";

      print "author(fullname or part of name) -- ?author=seventhswan";

      print "</br>";

      print "category -- ?category=romance";

      print "</br>";

      print "title(full title or part of title) -- ?title=Close Protection";

      print "</br>";

      print "year\_published -- ?year\_published=2015";

      print "</br>";

      print "isbn -- ?isbn=9780062407818";

      print "</br>";

      print "language -- ?language=english";

      print "</br>";

      print "orderby -- ?orderby=rating";

      print "</br>";

      print "limit(max number of book to output) -- ?limit=10";

      print "</br>";

      print "</br>";

      print "</br>";

      print "examples";

      print "</br>";

      print "json output with title contains 'close' -- ?format=json&title=Close";

      print "</br>";

      print "xml output with author contains 'se', written in english, show highest rating first, limit the result to only 5 -- ?format=xml&author=se&language=english&orderby=rating&limit=5";

    }

  }

  function subquery($params)

  { // generate sql sub query -- starting from 'WHERE'

    $subquery = "";

    $subqueryarray = array(); //everything that will have 'AND' in front will be push to array

    if (isset($params['isbn'])) {

      array\_push($subqueryarray, " isbn LIKE '%" . $params["isbn"] . "%'");

    }

    if (isset($params['author'])) {

      array\_push($subqueryarray, " author LIKE '%" . $params["author"] . "%'");

    }

    if (isset($params['category'])) {

      array\_push($subqueryarray, " category LIKE '%" . $params["category"] . "%'");

    }

    if (isset($params['title'])) {

      array\_push($subqueryarray,  " title LIKE '%" . $params['title'] . "%'");

    }

    if (isset($params['language'])) {

      array\_push($subqueryarray, " language LIKE '%" . $params['language'] . "%'");

    }

    if (isset($params['year\_published'])) {

      array\_push($subqueryarray, " year\_published LIKE '%" . $params['year\_published'] . "%'");

    }

    if (!empty($subqueryarray)) { // adding the 'AND' word infront of each value in the array

      $subquery .= " WHERE";

      foreach ($subqueryarray as $key => $value) {

        if (!empty($value)) {

          $subquery .= " AND" . $value;

        }

      }

    }

    if (isset($params['orderby'])) { //no 'AND' infront therefore no need to push to array

      $subquery .=  " ORDER BY " . $params['orderby'] . "";

    }

    if (isset($params['limit'])) { //no 'AND' infront therefore no need to push to array

      $subquery .=  " LIMIT " . $params['limit'] . "";

    }

    return str\_replace(' WHERE AND', ' WHERE', $subquery . ";"); //return a usable sql select query statement

  }

  function select($selectquery = "SELECT \* FROM books", $countquery = "SELECT COUNT(\*) FROM books")

  {

    $count = $this->db->select($countquery)[0]["COUNT(\*)"]; //get number of rows in books table

    $data = $this->db->select($selectquery); // get all details from books table

    $statusCode = "";

    $statusMessage = "";

    if (!empty($data)) { //executed when there is format query

      $statusCode = "200";

      $statusMessage = "Successfull";

    } else {

      $statusCode = "404";

      $statusMessage = "Not found";

    }

    if ($this->format == 'json') {

      header("Content-Type:application/json");

      $response = $this->addStatusToData($statusCode, $statusMessage, $count, $data); //call funtion to add status code , message , totalbook  to the response

      $json\_response = json\_encode($response, JSON\_UNESCAPED\_SLASHES); //output in json format

      return $json\_response;

    } elseif ($this->format == 'xml') {

      header("Content-Type:application/xml");

      $response = $this->addStatusToData($statusCode, $statusMessage, $count, $data); //call funtion to add status code , message , totalbook  to the response

      $xml\_response = $this->xml\_encode($response); //function created to immitate json\_encode

      return $xml\_response;

    } else { //bad request

      header("Content-Type:application/json");

      $response = $this->addStatusToData(400, 'Bad Request', $count, null); //call funtion to add status code , message , totalbook  to the response

      $json\_response = json\_encode($response, JSON\_UNESCAPED\_SLASHES); //output in json format

      return $json\_response;

    }

  }

  function addStatusToData($status, $statusMessage, $count, $data)

  { // adding status code , message , totalbook  to the response

    header("HTTP/1.1 $status $statusMessage");

    $response['status'] = $status;

    $response['message'] = $statusMessage;

    $response['totalBookInDB'] = $count;

    $response['response'] = $data;

    return $response;

  }

  function xml\_encode($data)

  {

    // creating object of SimpleXMLElement

    $xml = new SimpleXMLElement('<?xml version="1.0"?><bookstore></bookstore>');

    //function call to convert array to xml

    $this->array\_to\_xml($data, $xml);

    //saving generated xml file;

    $result = $xml->asXML();

    //output the file in xml in the browser

    return ($result);

  }

  function array\_to\_xml($data, &$xml)

  {

    foreach ($data as $key => $value) {

      if (is\_array($value)) {

        if (is\_numeric($key)) { //if there are sub arrays

          $key = 'book' . $key; //dealing with <0/>..<n/> issues

        }

        $subnode = $xml->addChild($key); //add a node to the main

        $this->array\_to\_xml($value, $subnode); //recursive function

      } else {

        $xml->addChild("$key", htmlspecialchars("$value")); //add the value for the node

      }

    }

  }

}

# Annex G: index.php

<!DOCTYPE html>

<!--

Template Name: Wavefire

Author: <a href="https://www.os-templates.com/">OS Templates</a>

Author URI: https://www.os-templates.com/

Copyright: OS-Templates.com

Licence: Free to use under our free template licence terms

Licence URI: https://www.os-templates.com/template-terms

-->

<html lang="en">

<!-- To declare your language - read more here: https://www.w3.org/International/questions/qa-html-language-declarations -->

<head>

  <title>BookStore | Home Page</title>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">

  <link href="../layout/styles/layout.css" rel="stylesheet" type="text/css" media="all">

  <link href="../fontawesome-free-5.15.3-web/css/all.css" rel="stylesheet">

  <link href="../bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet">

  <link href="../layout/styles/mycss.css" rel="stylesheet">

</head>

<!-- when the body load fill the elements with the appropriate details such as date time alertlevel format... -->

<body id="top" onload="fillElements()">

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <div class="wrapper row0">

    <header id="header" class="hoc clear">

      <!-- ################################################################################################ -->

      <div id="logo" class="one\_quarter first">

        <h1><a href="index.php"><span>B</span>ook<span>S</span>tore</a></h1>

      </div>

      <div class="three\_quarter">

        <ul class="nospace clear">

          <li class="one\_third first">

            <div class="block clear"><a href="#"><i class="fas fa-phone"></i></a> <span><strong>Give us a call:</strong> +00 (123) 456 7890</span></div>

          </li>

          <li class="one\_third">

            <div class="block clear"><a href="#"><i class="fas fa-envelope"></i></a> <span><strong>Send us a mail:</strong> support@domain.com</span></div>

          </li>

          <li class="one\_third">

            <div class="block clear"><a href="#"><i class="fas fa-clock"></i></a> <span><strong> Mon. - Sat.:</strong> 08.00am - 18.00pm</span></div>

          </li>

        </ul>

      </div>

      <!-- ################################################################################################ -->

    </header>

  </div>

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <div class="wrapper row1">

    <section class="hoc clear">

      <!-- ################################################################################################ -->

      <nav id="mainav">

        <ul class="clear">

          <li><a href="index.php">Home Page</a></li>

          <li class="active"><a class="drop" href="#">Pages</a>

            <ul>

              <li class="active"><a href="index.php">Home Page</a></li>

            </ul>

          </li>

        </ul>

      </nav>

      <!-- ################################################################################################ -->

      <div id="searchform">

        <div>

        </div>

      </div>

      <!-- ################################################################################################ -->

    </section>

  </div>

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <div class="wrapper bgded overlay" style="background-image:url('../images/demo/backgrounds/01.png');">

    <div id="breadcrumb" class="hoc clear">

      <!-- ################################################################################################ -->

      <h6 class="heading">BookStore</h6>

      <ul>

        <li><a href="#">Home Page</a></li>

      </ul>

      <!-- ################################################################################################ -->

    </div>

  </div>

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <div class="wrapper row3">

    <main class="hoc container clear">

      <!-- main body -->

      <!-- ################################################################################################ -->

      <div class="content">

        <!-- ################################################################################################ -->

        <br>

        <br>

        <br>

        <br>

        <div class="row">

          <div class="col-sm-4">

            <input type="text" value="" name="isbn" id="isbn" class="form-control" placeholder="isbn">

          </div>

          <div class="col-sm-4">

            <input type="text" value="" name="author" id="author" class="form-control" placeholder="author">

          </div>

          <div class="col-sm-4">

            <input type="text" value="" name="title" id="title" class="form-control" placeholder="title">

          </div>

        </div>

        <br>

        <div class="row">

          <div class="col-sm-4">

            <input type="text" value="" name="language" id="language" class="form-control" placeholder="language">

          </div>

          <div class="col-sm-4">

            <input type="text" value="" name="orderby" id="orderby" class="form-control" placeholder="orderby">

          </div>

          <div class="col-sm-4">

            <input type="text" value="" name="limit" id="limit" class="form-control" placeholder="limit">

          </div>

        </div>

        <br>

        <div class="row">

          <div class="col-sm-4">

            <input type="text" value="" name="category" id="category" class="form-control" placeholder="category">

          </div>

          <div class="col-sm-4">

            <select name="format" id="format" class="form-control">

              <option value="xml">XML</option>

              <option value="json">JSON</option>

            </select>

          </div>

          <div class="col-sm-4">

            <input type="text" value="" name="year\_published" id="year\_published" class="form-control" placeholder="year published">

          </div>

        </div>

        <br>

        <div class="row">

          <div class="col-sm-12">

            <!--on click the js will build the url and then request the server -->

            <input type="button" name="search" id="search" value="Advanced Search" class="btn btn-info form-control" onclick="buildURL();" />

          </div>

        </div>

        <br>

        <br>

        <br>

        <br>

        <!-- main display  -->

        <h1>Books</h1>

        <?php

        require("../php\_files/queryAPI.php"); // to retrieve api

        require("../php\_files/display.php"); //display the div - list one by one

        new display(API . '?' . $\_SERVER['QUERY\_STRING']);

        ?>

        <br>

        <br>

        <!-- ################################################################################################ -->

      </div>

      <!-- ################################################################################################ -->

      <!-- / main body -->

      <div class="clear"></div>

    </main>

  </div>

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <div class="wrapper row4">

    <footer id="footer" class="hoc clear">

      <!-- ################################################################################################ -->

      <!-- ################################################################################################ -->

    </footer>

  </div>

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <div class="wrapper row5">

    <div id="copyright" class="hoc clear">

      <!-- ################################################################################################ -->

      <p class="fl\_left">Copyright &copy; 2018 - All Rights Reserved - <a href="#">Domain Name</a></p>

      <p class="fl\_right">Template by <a target="\_blank" href="https://www.os-templates.com/" title="Free Website Templates">OS Templates</a></p>

      <!-- ################################################################################################ -->

    </div>

  </div>

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <!-- ################################################################################################ -->

  <a id="backtotop" href="#top"><i class="fas fa-chevron-up"></i></a>

  <!-- JAVASCRIPTS -->

  <script src="../layout/scripts/jquery-3.5.1.min.js"></script>

  <!-- <script src="../layout/scripts/jquery.min.js"></script> -->

  <script src="../layout/scripts/jquery.backtotop.js"></script>

  <script src="../layout/scripts/jquery.mobilemenu.js"></script>

  <script src="../bootstrap/4.3.1/js/bootstrap.min.js"></script>

  <script src="../layout/scripts/myjs.js"></script>

</body>

</html>

# Annex H: detail.php

<!DOCTYPE html>

<!--

Template Name: Wavefire

Author: <a href="https://www.os-templates.com/">OS Templates</a>

Author URI: https://www.os-templates.com/

Copyright: OS-Templates.com

Licence: Free to use under our free template licence terms

Licence URI: https://www.os-templates.com/template-terms

-->

<html lang="en">

<!-- To declare your language - read more here: https://www.w3.org/International/questions/qa-html-language-declarations -->

<head>

    <title>BookStore | Home Page</title>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">

    <link href="../layout/styles/layout.css" rel="stylesheet" type="text/css" media="all">

    <link href="../fontawesome-free-5.15.3-web/css/all.css" rel="stylesheet">

    <link href="../bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet">

    <link href="../layout/styles/mycss.css" rel="stylesheet">

</head>

<!-- when the body load fill the elements with the appropriate details such as date time alertlevel format... -->

<body id="top">

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <div class="wrapper row0">

        <header id="header" class="hoc clear">

            <!-- ################################################################################################ -->

            <div id="logo" class="one\_quarter first">

                <h1><a href="index.php"><span>B</span>ook<span>S</span>tore</a></h1>

            </div>

            <div class="three\_quarter">

                <ul class="nospace clear">

                    <li class="one\_third first">

                        <div class="block clear"><a href="#"><i class="fas fa-phone"></i></a> <span><strong>Give us a call:</strong> +00 (123) 456 7890</span></div>

                    </li>

                    <li class="one\_third">

                        <div class="block clear"><a href="#"><i class="fas fa-envelope"></i></a> <span><strong>Send us a mail:</strong> support@domain.com</span></div>

                    </li>

                    <li class="one\_third">

                        <div class="block clear"><a href="#"><i class="fas fa-clock"></i></a> <span><strong> Mon. - Sat.:</strong> 08.00am - 18.00pm</span></div>

                    </li>

                </ul>

            </div>

            <!-- ################################################################################################ -->

        </header>

    </div>

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <div class="wrapper row1">

        <section class="hoc clear">

            <!-- ################################################################################################ -->

            <nav id="mainav">

                <ul class="clear">

                    <li><a href="index.php">Home Page</a></li>

                    <li class="active"><a class="drop" href="#">Pages</a>

                        <ul>

                            <li class="active"><a href="index.php">Home Page</a></li>

                        </ul>

                    </li>

                </ul>

            </nav>

            <!-- ################################################################################################ -->

            <div id="searchform">

                <div>

                    <form action="#" method="post">

                        <fieldset>

                            <legend>Quick Search:</legend>

                            <input type="text" placeholder="Enter search term&hellip;">

                            <button type="submit"><i class="fas fa-search"></i></button>

                        </fieldset>

                    </form>

                </div>

            </div>

            <!-- ################################################################################################ -->

        </section>

    </div>

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <div class="wrapper bgded overlay" style="background-image:url('../images/demo/backgrounds/01.png');">

        <div id="breadcrumb" class="hoc clear">

            <!-- ################################################################################################ -->

            <h6 class="heading">BookStore</h6>

            <ul>

                <li><a href="#">Home Page</a></li>

            </ul>

            <!-- ################################################################################################ -->

        </div>

    </div>

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <div class="wrapper row3">

        <main class="hoc container clear">

            <!-- main body -->

            <!-- ################################################################################################ -->

            <div class="content">

                <!-- ################################################################################################ -->

                <br>

                <br>

                <br>

                <br>

                <br>

                <br>

                <br>

                <br>

                <br>

                <!-- main display  -->

                <h1>Book Details</h1>

                <?php

                require("../php\_files/queryAPI.php"); // to retrieve api

                require("../php\_files/display.php"); //display the div - list one by one

                new display(API . '?' . $\_SERVER['QUERY\_STRING'], true);

                ?>

                <br>

                <br>

                <!-- ################################################################################################ -->

            </div>

            <!-- ################################################################################################ -->

            <!-- / main body -->

            <div class="clear"></div>

        </main>

    </div>

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <div class="wrapper row4">

        <footer id="footer" class="hoc clear">

            <!-- ################################################################################################ -->

            <!-- ################################################################################################ -->

        </footer>

    </div>

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <div class="wrapper row5">

        <div id="copyright" class="hoc clear">

            <!-- ################################################################################################ -->

            <p class="fl\_left">Copyright &copy; 2018 - All Rights Reserved - <a href="#">Domain Name</a></p>

            <p class="fl\_right">Template by <a target="\_blank" href="https://www.os-templates.com/" title="Free Website Templates">OS Templates</a></p>

            <!-- ################################################################################################ -->

        </div>

    </div>

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <!-- ################################################################################################ -->

    <a id="backtotop" href="#top"><i class="fas fa-chevron-up"></i></a>

    <!-- JAVASCRIPTS -->

    <script src="../layout/scripts/jquery-3.5.1.min.js"></script>

    <!-- <script src="../layout/scripts/jquery.min.js"></script> -->

    <script src="../layout/scripts/jquery.backtotop.js"></script>

    <script src="../layout/scripts/jquery.mobilemenu.js"></script>

    <script src="../bootstrap/4.3.1/js/bootstrap.min.js"></script>

    <script src="../layout/scripts/myjs.js"></script>

</body>

</html>

# Annex I: bookstore.sql

CREATE DATABASE `bookstore`;

CREATE TABLE bookstore.books(isbn Varchar(255) PRIMARY KEY NOT NULL,category varchar(255) NOT NULL,title Varchar(255) NOT NULL,rating int(1) NOT NULL,author varchar(255) not null,year\_published int(4) NOT NULL,description LONGTEXT NOT NULL,language Varchar(255) NOT NULL,review INT(5) NOT NULL,best\_seller boolean NOT NULL,cover\_photo varchar(255) NOT NULL);

use bookstore;

INSERT INTO `books` (`isbn`,`category`,`title`,`rating`,`author`,`year\_published`,`description`,`language`,`review`,`best\_seller`,`cover\_photo`) VALUES

    ('9780062407818','All category','Never Split the Difference: Negotiating As If Your Life Depended On It',5,'Chris Voss',2016,"From Scribd: About the Book.Hailed as one of the best modern books on the art of negotiation, Never Split the Difference is written by Chris Voss, a renowned negotiator.After serving on the police force in Kansas City, Missouri, and experiencing what the rough streets there had to offer, Voss joined the FBI to further his career. He eventually became a hostage negotiator, and in this line of work he often came face-to-face with a range of criminals, including bank robbers and terrorists.Only by becoming the FBI' s lead international kidnapping negotiator did Voss reach the pinnacle of his career,and Never Split the Difference will take you inside the life - or - death stakes negotiations that Voss managed.This book gets inside Voss 's head, revealing his skills and methods that allowed him and his colleagues to be so successful when so much was on the line.Broken down into nine effective principles, this practical guide will teach you the methods and strategies you can use to become more persuasive in both your personal and professional life.As Voss says, life is a series of negotiations that you should always be prepared for. Everything from buying a car to negotiating a salary, and even discussing things with your partner. With the knowledge contained in Never Split the Difference you can take your emotional intelligence to the next level.",'english',110,'1','https://imgv2-1-f.scribdassets.com/img/word\_document/310560108/original/216x287/7b8f89909b/1627012728?v=1'),

    ('2626920','erotica','The Violet and the Tom',5,'Eve Ocotillo',2009,'In what might have been the middle ages, had neither Alexander the Great nor Jesus the prophet died young, the Greek State is a powerful economic force in southern Europe, and slavery is a profitable and well-entrenched social institution. Nygell, a Lord of the Northern Isles, is given the gift of a Grecian slave by the King. Nygell wants no such responsibility.A homoerotic romance. Set in an alternate universe with institutionalized slavery, thus consent is by nature dubious at best. Elements of BDSM.','english',343,'1','https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1295750108l/8164754.jpg'),

    ('00000000001','fantasy','The student prince',4,'FayJay',2012,'A modern day (BBC) Merlin AU set at the University of St Andrews, featuring teetotal kickboxers, secret wizards, magnificent bodyguards of various genders, irate fairies, imprisoned dragons, crumbling gothic architecture, arrogant princes, adorable engineering students, stolen gold, magical doorways, attempted assassination, drunken students, shaving foam fights, embarrassing mornings after, The Hammer Dance, duty, responsibility, friendship and true love...','english',596,'1','https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1400436772l/17277854.jpg'),

    ('1426714343','romance','Heart in Hand',3,'salifiable',2011,'Hockey RPF.Darryl had realized very early on that for better or worse, he would be linked to Aleksey Kuznetsov for the rest of his life. But he never expected anything like this.','english',316,'0','https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1364753584l/13151360.jpg'),

    ('1980900388','romance','Close Protection',4,'Cordelia Kingsbridge',2012,"bodyguard’s life is turned upside down by his intriguing new charge. After three consecutive expulsions for sexual misconduct, Luca D'Amato has no choice but to return to America. Jacob Ryder is the man charged with protecting him from his family's many enemies – but who will protect Ryder from Luca?",'english',267,'0','https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1391963345l/13481282.jpg'),

    ('979-8679122673','romance','Drunk Text',3,'seventhswan',2018,"Gunnar wakes up in his dorm room on a Friday in April and notices three things in rapid succession. The most pressing is a text message that says: 'God, my ass hurts. I can't even sit down properly. Cheers to you. Coffee @ 1?'",'english',237,'1','https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1395690840l/13494082.jpg'),

    ('9781408855652','fantasy',"Harry Potter and the Philosopher's Stone",4,'J.K. Rowling',2014,"Escape to Hogwarts with the unmissable series that has sparked a lifelong reading journey for children and families all over the world. The magic starts here.Harry Potter has never even heard of Hogwarts when the letters start dropping on the doormat at number four, Privet Drive. Addressed in green ink on yellowish parchment with a purple seal, they are swiftly confiscated by his grisly aunt and uncle. Then, on Harry's eleventh birthday, a great beetle-eyed giant of a man called Rubeus Hagrid bursts in with some astonishing news: Harry Potter is a wizard, and he has a place at Hogwarts School of Witchcraft and Wizardry. The magic starts here!These editions of the classic and internationally bestselling Harry Potter series feature thrilling jacket artwork by award-winning illustrator Jonny Duddle. They are the perfect starting point for anyone who's ready to lose themselves in the greatest children's story of all time.",'english',122497,'1','https://d1w7fb2mkkr3kw.cloudfront.net/assets/images/book/lrg/9781/4088/9781408855652.jpg'),

    ('B017V4IPPO','fantasy',"Harry Potter and the Chamber of Secrets",4,'J.K. Rowling',2015,"Ever since Harry Potter had come home for the summer, the Dursleys had been so mean and hideous that all Harry wanted was to get back to the Hogwarts School for Witchcraft and Wizardry. But just as he’s packing his bags, Harry receives a warning from a strange impish creature who says that if Harry returns to Hogwarts, disaster will strike.And strike it does. For in Harry’s second year at Hogwarts, fresh torments and horrors arise, including an outrageously stuck-up new professor and a spirit who haunts the girls’ bathroom. But then the real trouble begins – someone is turning Hogwarts students to stone. Could it be Draco Malfoy, a more poisonous rival than ever? Could it possible be Hagrid, whose mysterious past is finally told? Or could it be the one everyone at Hogwarts most suspects… Harry Potter himself!",'english',58421,'1','https://m.media-amazon.com/images/I/51RDEwuovQL.jpg'),

    ('043965548X','fantasy',"Harry Potter and the Prisoner of Azkaban",5,'J.K. Rowling',2004,"For twelve long years, the dread fortress of Azkaban held an infamous prisoner named Sirius Black. Convicted of killing thirteen people with a single curse, he was said to be the heir apparent to the Dark Lord, Voldemort.Now he has escaped, leaving only two clues as to where he might be headed: Harry Potter's defeat of You-Know-Who was Black's downfall as well. And the Azkaban guards heard Black muttering in his sleep, ''He's at Hogwarts . . . he's at Hogwarts.''Harry Potter isn't safe, not even within the walls of his magical school, surrounded by his friends. Because on top of it all, there may well be a traitor in their midst.",'english',61374,'1','https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1499277281l/5.jpg'),

    ('B017V4NQGM','fantasy',"Harry Potter and the Goblet of Fire",5,'J.K. Rowling',2015,"Harry Potter is midway through his training as a wizard and his coming of age. Harry wants to get away from the pernicious Dursleys and go to the International Quidditch Cup with Hermione, Ron, and the Weasleys. He wants to dream about Cho Chang, his crush (and maybe do more than dream). He wants to find out about the mysterious event that's supposed to take place at Hogwarts this year, an event involving two other rival schools of magic, and a competition that hasn't happened for hundreds of years. He wants to be a normal, fourteen-year-old wizard. But unfortunately for Harry Potter, he's not normal - even by wizarding standards.",'english',54304,'1','https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1554006152l/6.jpg'),

    ('B017V4NLJ4','fantasy',"Harry Potter and the Order of the Phoenix",5,'J.K. Rowling',2015,"There is a door at the end of a silent corridor. And it’s haunting Harry Pottter’s dreams. Why else would he be waking in the middle of the night, screaming in terror?Harry has a lot on his mind for this, his fifth year at Hogwarts: a Defense Against the Dark Arts teacher with a personality like poisoned honey; a big surprise on the Gryffindor Quidditch team; and the looming terror of the Ordinary Wizarding Level exams. But all these things pale next to the growing threat of He-Who-Must-Not-Be-Named - a threat that neither the magical government nor the authorities at Hogwarts can stop.As the grasp of darkness tightens, Harry must discover the true depth and strength of his friends, the importance of boundless loyalty, and the shocking price of unbearable sacrifice.",'english',46965,'1','https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1546910265l/2.jpg'),

    ('B017V4NOEG','fantasy',"Harry Potter and the Half-Blood Prince",5,'J.K. Rowling',2015,"The war against Voldemort is not going well; even Muggle governments are noticing. Ron scans the obituary pages of the Daily Prophet, looking for familiar names. Dumbledore is absent from Hogwarts for long stretches of time, and the Order of the Phoenix has already suffered losses.And yet . . .As in all wars, life goes on. The Weasley twins expand their business. Sixth-year students learn to Apparate - and lose a few eyebrows in the process. Teenagers flirt and fight and fall in love. Classes are never straightforward, through Harry receives some extraordinary help from the mysterious Half-Blood Prince.So it's the home front that takes center stage in the multilayered sixth installment of the story of Harry Potter. Here at Hogwarts, Harry will search for the full and complete story of the boy who became Lord Voldemort - and thereby find what may be his only vulnerability.",'english',43364,'1','https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1587697303l/1.\_SX318\_.jpg'),

    ('B017WJ5PR4','fantasy',"Harry Potter and the Deathly Hallows",5,'J.K. Rowling',2015,"Harry Potter is leaving Privet Drive for the last time. But as he climbs into the sidecar of Hagrid’s motorbike and they take to the skies, he knows Lord Voldemort and the Death Eaters will not be far behind.The protective charm that has kept him safe until now is broken. But the Dark Lord is breathing fear into everything he loves. And he knows he can’t keep hiding.To stop Voldemort, Harry knows he must find the remaining Horcruxes and destroy them.He will have to face his enemy in one final battle.",'english',69958,'1','https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1474171184l/136251.\_SY475\_.jpg'),

    ('0751565350','fantasy',"Harry Potter and the Cursed Child: Parts One and Two",3,'J.K. Rowling',2016,"Based on an original new story by J.K. Rowling, Jack Thorne and John Tiffany, a new play by Jack Thorne, Harry Potter and the Cursed Child is the eighth story in the Harry Potter series and the first official Harry Potter story to be presented on stage. The play will receive its world premiere in London’s West End on July 30, 2016.It was always difficult being Harry Potter and it isn’t much easier now that he is an overworked employee of the Ministry of Magic, a husband and father of three school-age children.While Harry grapples with a past that refuses to stay where it belongs, his youngest son Albus must struggle with the weight of a family legacy he never wanted. As past and present fuse ominously, both father and son learn the uncomfortable truth: sometimes, darkness comes from unexpected places.",'english',67085,'1','https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1470082995l/29056083.\_SY475\_.jpg');