ABC Online Book Store

v1.0.0

Daksh SCRA – Source Code Analysis Report

Jun 26, 2023

Scan Summary

This section provides a summary of the selected inputs and essential metrics collected during the scanning process. It offers an overview of the key information, allowing users to grasp the important aspects of the scan at a glance. For more detailed information about the identified areas of interest, please refer to the respective sections.

```
[+] Inputs Selected:
    [-] Target Directory: ../../Code/Platform/PHP/online-book-store/
    [-] Rule Selected: php
    [-] Total Rules Loaded: 39
        [-] Platform Specific Rules: 18
        [-] Common Rules: 21
    [-] File Types Selected: php
    [-] File Extensions Selected: *.php, *.ini, *.txt, *.config,
                                  *.xml, *.lock, .htaccess, *.log

[+] Detection Summary:
    [-] Total Project Files Identified: 72
    [-] Total Files Identified (Based on Selected Rule): 44
    [-] Total Files Scanned (Based on Selected Rule): 44
    [-] File Extensions Identified (Based on Selected Rule): .php, .txt
    [-] Code Files - Areas-of-Interests (Rules Matched): 13
    [-] File Paths - Areas-of-Interests (Rules Matched): 4

[+] Scanning Timeline:
    [-] Scan start time: 2023-06-26 17:09:52
    [-] Scan end time: 2023-06-26 17:10:33
    [-] Scan completed in: 00Hr:00Min:40s:936ms
```

Security - Areas of Interest

This section lists the key areas within the source code that need to be examined for identifying potential security weaknesses.

The code reviewer should carefully examine the identified areas and review any reported code snippets and file paths to validate the presence of any potential vulnerabilities. The validation process should involve a thorough analysis of the code and its associated components to determine the extent of the potential security risk. Any issues identified during the validation process should be documented.

1. Rule Title : Tainted Inputs: $_GET

Rule Description : Identifies the usage of the $_GET superglobal, which may lead to tainted inputs being used in the code.

Issue Description : If this rule matches, it indicates the potential vulnerability of using user input from the $_GET superglobal without proper validation and sanitization. Attackers can manipulate URL parameters to provide malicious or unexpected values, potentially leading to security vulnerabilities such as injection attacks (SQL, OS, or LDAP), cross-site scripting (XSS), or unauthorized access to sensitive information.

Developer Note : Developers should carefully validate and sanitize user input received from the $_GET superglobal before using it in any sensitive operations. They should implement strong input validation, ensuring that only expected and sanitized values are accepted from URL parameters. Additionally, developers should be aware of the risks associated with using user-supplied data and follow secure coding practices, such as parameterized queries, output encoding, and appropriate input filtering techniques, depending on the context in which the input is used.

Reviewer Note : Reviewers should check for appropriate input handling for the $_GET superglobal. They should assess if developers have implemented proper input validation and sanitization techniques to prevent security vulnerabilities related to tainted inputs. Reviewers should also verify the usage of output encoding and protection against injection attacks to ensure the secure handling of user input received through URL parameters.

- Source File : online-book-store/secure/admin_delete.php

```
[2]         $book_isbn = $_GET['bookisbn'];
```

- Source File : online-book-store/secure/admin_edit.php

```
[9]         if(isset($_GET['bookisbn'])){
[10]                $book_isbn = $_GET['bookisbn'];
```

- Source File : online-book-store/secure/book.php

```
[3]  $book_isbn = $_GET['bookisbn'];
```

- Source File : online-book-store/secure/bookPerPub.php

```
[5]         if(isset($_GET['pubid'])){
[6]                 $pubid = $_GET['pubid'];
```

- Source File : online-book-store/secure/checkout.php

```
[6]                                 book_isbn (get from $_GET['book_isbn']) => number of books
```

2. Rule Title : Insecure Method Call: header()

Rule Description : Detects insecure usage of the header() function, specifically related to the "Location" parameter.

Issue Description : If this rule matches, it indicates the potential vulnerability of insecure header redirection. Attackers can manipulate the "Location" parameter in header() calls to redirect users to malicious websites or perform phishing attacks.

Developer Note : Developers should ensure that any user input used in the "Location" parameter of header() calls is properly validated, sanitized, and restricted to trusted values. They should also consider using security libraries or frameworks that provide safer methods for handling redirects.

Reviewer Note : Reviewers should identify the usage of header() with the "Location" parameter and verify that input validation and sanitization are applied to prevent header injection vulnerabilities. They should also check if a secure redirect mechanism is used to mitigate the risk of unauthorized redirects.

- Source File : online-book-store/secure/admin_add.php

```
1 | [62]                    header("Location: admin_book.php");
```

- Source File : online-book-store/secure/admin_delete.php

```
1 | [13]        header("Location: admin_book.php");
```

- Source File : online-book-store/secure/admin_signout.php

```
1 | [4]        header("Location: index.php");
```

- Source File : online-book-store/secure/admin_verify.php

```
1 | [39]        header("Location: admin_book.php");
```

- Source File : online-book-store/secure/edit_book.php

```
1 | [56]                header("Location: admin_edit.php?bookisbn=$isbn");
```

- Source File : online-book-store/secure/process.php

```
1 | [13]                header("Location: purchase.php");
```

- Source File : online-book-store/secure/purchase.php

```
1 | [12]                header("Location: checkout.php");
```

- Source File : online-book-store/secure/functions/admin.php

```
1 | [3]               header("Location: index.php");
```

3. Rule Title : Deprecated MySQL Query Functions

Rule Description : Detects the usage of deprecated MySQL query functions, which can lead to security vulnerabilities.

Issue Description : If this rule matches, it indicates the potential vulnerability of SQL injection or other security issues. Attackers can exploit insecure MySQL queries to manipulate database operations, retrieve unauthorized data, or perform malicious actions.

Developer Note : Developers should migrate to modern and secure database APIs, such as PDO or MySQLi, and use prepared statements or parameterized queries to prevent SQL injection vulnerabilities. It is important to sanitize and validate user input before incorporating it into database queries.

Reviewer Note : Reviewers should ensure that deprecated MySQL query functions, such as mysql_query() or mysqli_query(), are replaced with secure alternatives. They should also verify the presence of proper input validation, query parameterization, and adherence to secure coding practices to mitigate SQL injection vulnerabilities.

- Source File : online-book-store/secure/admin_add.php

```
1 | [40]                $findResult = mysqli_query($conn, $findPub);
2 | [44]                        $insertResult = mysqli_query($conn, $insertPub);
3 | [57]                $result = mysqli_query($conn, $query);
```

- Source File : online-book-store/secure/admin_delete.php

```
1 | [8]        $result = mysqli_query($conn, $query);
```

- Source File : online-book-store/secure/admin_edit.php

```
1 | [23]         $result = mysqli_query($conn, $query);
```

- Source File : online-book-store/secure/admin_verify.php

```
[24]        $result = mysqli_query($conn, $query);
```

- Source File : online-book-store/secure/book.php

```
[9]  $result = mysqli_query($conn, $query);
```

- Source File : online-book-store/secure/bookPerPub.php

```
[17]        $result = mysqli_query($conn, $query);
```

- Source File : online-book-store/secure/books.php

```
[9]  $result = mysqli_query($conn, $query);
```

- Source File : online-book-store/secure/edit_book.php

```
[28]        $findResult = mysqli_query($conn, $findPub);
[32]                $insertResult = mysqli_query($conn, $insertPub);
[51]        $result = mysqli_query($conn, $query);
```

- Source File : online-book-store/secure/process.php

```
[48]                $result = mysqli_query($conn, $query);
```

- Source File : online-book-store/secure/publisher_list.php

```
[7]        $result = mysqli_query($conn, $query);
[26]                        $result2 = mysqli_query($conn, $query);
```

- Source File : online-book-store/secure/verify.php

```
[12]         $result = mysqli_query($conn, $query);
```

- Source File : online-book-store/secure/functions/database_functions.php

```
[14]                $result = mysqli_query($conn, $query);
[27]                $result = mysqli_query($conn, $query);
[37]                $result = mysqli_query($conn, $query);
[49]                $result = mysqli_query($conn, $query);
[59]                $result = mysqli_query($conn, $query);
[76]                $result = mysqli_query($conn, $query);
[91]                $result = mysqli_query($conn, $query);
[102]                $result = mysqli_query($conn, $query);
[118]                $result = mysqli_query($conn, $query);
```

4. Rule Title : Cross-Site Scripting (XSS) via PHP_SELF

Rule Description : Detects the usage of the PHP_SELF variable in the context of cross-site scripting (XSS) vulnerabilities.

Issue Description : If this rule matches, it indicates the potential vulnerability of cross-site scripting (XSS). Attackers can exploit the PHP_SELF variable to inject malicious scripts into web pages and potentially steal sensitive user information or perform unauthorized actions on behalf of the user.

Developer Note : Developers should avoid directly using the PHP_SELF variable to generate URLs or include it in output without proper sanitization and encoding to mitigate cross-site scripting (XSS) vulnerabilities. It is recommended to use alternative methods, such as carefully constructing URLs or relying on server-side frameworks that handle URL generation securely.

Reviewer Note : Reviewers should verify that the PHP_SELF variable is not directly used in generating URLs or output without proper sanitization and encoding. They should also ensure that other security measures, such as input validation and output escaping, are implemented to prevent XSS vulnerabilities in the codebase.

- Source File : online-book-store/secure/admin_add.php

```
[31]                        $directory_self = str_replace(basename($_SERVER['PHP_SELF']), '',
$_SERVER['PHP_SELF']);
```

- Source File : online-book-store/secure/edit_book.php

```
[17]                $directory_self = str_replace(basename($_SERVER['PHP_SELF']), '', $_SERVER['PHP_SELF']);
```

5. Rule Title : SELECT [*] FROM [anytable] WHERE

Rule Description : Detects instances of SELECT statements in SQL queries that retrieve data from a table with a WHERE clause.

Issue Description : If this rule matches, it indicates the potential vulnerability of SQL injection. Attackers can exploit this vulnerability to manipulate the WHERE clause and perform unauthorized data retrieval or modification.

Developer Note : Developers should ensure that SQL queries using SELECT statements with a WHERE clause are properly parameterized or sanitized to prevent SQL injection attacks.

Reviewer Note : Reviewers should verify the presence of secure coding practices, such as parameterized queries or proper input validation and sanitization, to mitigate the risk of SQL injection vulnerabilities in SELECT statements with a WHERE clause.

- Source File : online-book-store/secure/admin_add.php

```
[39]                $findPub = "SELECT * FROM publisher WHERE publisher_name = '$publisher'";
```

- Source File : online-book-store/secure/admin_edit.php

```
[22]        $query = "SELECT * FROM books WHERE book_isbn = '$book_isbn'";
```

- Source File : online-book-store/secure/book.php

```
[8]  $query = "SELECT * FROM books WHERE book_isbn = '$book_isbn'";
```

- Source File : online-book-store/secure/edit_book.php

```
[27]        $findPub = "SELECT * FROM publisher WHERE publisher_name = '$publisher'";
```

- Source File : online-book-store/secure/controllers/DatabaseTalking/Take.php

```
[202]           $this->stat=$this->conn->prepare("SELECT image FROM book WHERE book_id=?");
```

- Source File : online-book-store/secure/controllers/DatabaseTalking/Talking.php

```
[514]           SELECT * FROM admin WHERE user_name=?;
[562]           SELECT * FROM author  WHERE name=?;
[577]           SELECT * FROM genre  WHERE genre_desc=?;
```

- Source File : online-book-store/secure/functions/database_functions.php

```
[36]                $query = "SELECT orderid FROM orders WHERE customerid = '$customerid'";
[58]                $query = "SELECT book_price FROM books WHERE book_isbn = '$isbn'";
[101]                $query = "SELECT publisher_name FROM publisher WHERE publisherid = '$pubid'";
```

6. Rule Title : SELECT ORDERBY

Rule Description : Detects instances of SELECT statements in SQL queries that include an ORDER BY clause.

Issue Description : If this rule matches, it indicates the potential vulnerability of SQL injection. Attackers can exploit this vulnerability to manipulate the ORDER BY clause and potentially alter the query results or introduce additional malicious functionality.

Developer Note : Developers should ensure that SQL queries using SELECT statements with an ORDER BY clause are properly parameterized or sanitized to prevent SQL injection attacks.

Reviewer Note : Reviewers should verify the presence of secure coding practices, such as parameterized queries or proper input validation and sanitization, specifically for SELECT statements that include an ORDER BY clause, to mitigate the risk of SQL injection vulnerabilities.

- Source File : online-book-store/secure/publisher_list.php

```
1 | [6]        $query = "SELECT * FROM publisher ORDER BY publisherid";
```

- Source File : online-book-store/secure/controllers/DatabaseTalking/Talking.php

```
1 | [467]            SELECT publisher_id,name FROM publisher ORDER BY  2 ");
2 | [535]            SELECT * FROM admin ORDER BY admin_id;
3 | [549]            SELECT * FROM author  ORDER BY name;
4 | [593]            SELECT * FROM genre  ORDER BY genre_desc;
```

- Source File : online-book-store/secure/functions/database_functions.php

```
1 | [13]               $query = "SELECT book_isbn, book_image FROM books ORDER BY book_isbn DESC";
2 | [117]               $query = "SELECT * from books ORDER BY book_isbn DESC";
```

7. Rule Title : Password

Rule Description : Detects potential password-related strings in the code.

Issue Description : If this rule matches, it indicates the potential presence of password-related strings in the code, which can lead to security risks if not handled properly.

Developer Note : Developers should follow best practices for password handling, including strong encryption, salted hashing, and enforcing secure password policies.

Reviewer Note : Reviewers should assess the password handling mechanisms and verify if proper security measures are in place.

- Source File : online-book-store/secure/database/readme.txt.txt

```
1 | [6]To change the localhost, username, password for connecting to database, change it only one
  | time in
```

8. Rule Title : Authentication Modules (Login|Authentication|Authenticated|Oauth|JWT)

Rule Description : Detects the presence of common module names related to authentication.

Issue Description : If this rule matches, it indicates the potential presence of module names associated with authentication, which can provide attackers with valuable information about the system's functionality and increase the attack surface.

Developer Note : Developers should ensure that the authentication processes are implemented securely.

Reviewer Note : Reviewers should verify if proper security measures are in place related to authentication.

- Source File : online-book-store/secure/template/footer.php

```
1 | [8]                        <a href="admin.php">Admin Login</a> 2017
```

9. Rule Title : Admin Modules/Section (Admin|Administrator)

Rule Description : Detects the presence of common module names related to administration functionalities or admin login pages.

Issue Description : If this rule matches, it indicates the potential presence of module names associated with administration functionalities or admin login pages, which can provide attackers with valuable information about the system's administration interfaces and increase the attack surface.

Developer Note : Developers should avoid using generic or predictable module names for admin functionalities or admin login pages and ensure that access control mechanisms and authentication processes are implemented securely.

Reviewer Note : Reviewers should assess the usage of module names related to administration functionalities or admin login pages and verify if proper security measures are in place to protect these critical areas of the system.

- Source File : online-book-store/secure/admin_add.php

```
[62]                        header("Location: admin_book.php");
```

- Source File : online-book-store/secure/admin_delete.php

```
[13]        header("Location: admin_book.php");
```

- Source File : online-book-store/secure/admin_verify.php

```
[23]        $query = "SELECT name, pass from admin";
[39]        header("Location: admin_book.php");
```

- Source File : online-book-store/secure/edit_book.php

```
[56]                header("Location: admin_edit.php?bookisbn=$isbn");
```

- Source File : online-book-store/secure/verify.php

```
[11]        $query = "SELECT username, password FROM admin";
[20]                        echo "Welcome admin! Long time no see";
```

- Source File : online-book-store/secure/controllers/DatabaseTalking/Talking.php

```
[507]     * @param $name : the name of admin
[508]     * @return Admin or null if any exception happened or user not found
[510]    public function getadmin ($name): Admin
[514]            SELECT * FROM admin WHERE user_name=?;
[520]            return new Admin($row['admin_id'], $row['user_name'], $row['pass']);
[535]            SELECT * FROM admin ORDER BY admin_id;
```

- Source File : online-book-store/secure/database/readme.txt.txt

```
[10]to connect the admin section, click the name Nghi Le Thanh at the bottom.
[11]the name and pass for log in is admin , admin. Just to make it simple.
```

- Source File : online-book-store/secure/models/admin/Admin.php

```
[10]class Admin
[18]     * Admin constructor.
[28]        Admin::$admins;
```

10. Rule Title : File Upload Functionality

Rule Description : Detects the presence of file upload functionality.

Issue Description : If this rule matches, it indicates the potential presence of file upload functionality that may allow unauthorized or malicious files to be uploaded, posing security risks such as arbitrary code execution or unauthorized access.

Developer Note : Developers should ensure that file upload functionality is implemented securely by validating file types, limiting file size, and applying appropriate access controls. It is recommended to use well-established file upload libraries or frameworks that have built-in security features.

Reviewer Note : Reviewers should verify the implementation of secure file upload practices, including proper file type validation, size restriction, and access control mechanisms.

- Source File : online-book-store/secure/models/serve/Image.php

```
[8]    function upload($newname):bool {
```

11. Rule Title : Potential SQL Injection: SELECT Statement with ORDER BY Clause

Rule Description : Detects the usage of SELECT statements with ORDER BY clauses.

Issue Description : If this rule matches, it indicates the potential use of SELECT statements with ORDER BY clauses, which can introduce the risk of SQL injection or unintended data exposure if not properly handled.

Developer Note : Developers should implement proper input validation and parameterization techniques to mitigate the risk of SQL injection when using SELECT statements with ORDER BY clauses.

Reviewer Note : Reviewers should verify the implementation of secure coding practices and assess if additional security measures are in place to prevent SQL injection.

- Source File : online-book-store/secure/publisher_list.php

```
1 | [6]         $query = "SELECT * FROM publisher ORDER BY publisherid";
```

- Source File : online-book-store/secure/controllers/DatabaseTalking/Talking.php

```
1 | [467]             SELECT publisher_id,name FROM publisher ORDER BY  2 ");
2 | [535]             SELECT * FROM admin ORDER BY admin_id;
3 | [549]             SELECT * FROM author  ORDER BY name;
4 | [593]             SELECT * FROM genre  ORDER BY genre_desc;
```

- Source File : online-book-store/secure/functions/database_functions.php

```
1 | [13]                $query = "SELECT book_isbn, book_image FROM books ORDER BY book_isbn DESC";
2 | [117]                $query = "SELECT * from books ORDER BY book_isbn DESC";
```

12. Rule Title : Trusted Or Untrusted URLs

Rule Description : Detects the presence of URLs in the application that may pose security risks or indicate potential trust issues with third-party URLs.

Issue Description : If this rule matches, it indicates the presence of URLs starting with http:// or https://, which could potentially pose security risks if not properly handled or validated. These URLs might involve data exfiltration or raise concerns about trusting third-party resources without thorough validation. Developers and reviewers should investigate these URLs to ensure their purpose, legitimacy, and adherence to security practices.

Developer Note : Developers should list and carefully review all URLs used in the application. They should implement proper URL validation and sanitization techniques to ensure the security and integrity of the application when dealing with user-provided or third-party URLs. It is important to prevent any unintentional exposure of sensitive information and perform thorough taint validation on third-party URLs to mitigate trust-related security risks.

Reviewer Note : Reviewers should verify the implementation of secure URL handling practices and assess if additional security measures are necessary. They should thoroughly investigate the presence of these URLs, determine their purpose, and ensure they undergo appropriate taint validation. Reviewers should pay special attention to URLs associated with data exfiltration concerns and assess the trustworthiness of third-party URLs to mitigate potential security risks.

- Source File : online-book-store/secure/template/footer.php

```
1 | [5]                 <a href="http://projectworlds.in" target="_blank"> projectworlds </a>
```

Parsed Paths – Areas of Interest

This section contains a list of file paths that have been identified by matching them with a predefined set of keywords. These files are typically of interest to a code reviewer, and should be examined for possible security vulnerabilities or insecure implementations.

1. Rule Title : Admin Section/Areas

- Source File : online-book-store/secure/admin.php
- Source File : online-book-store/secure/admin_add.php
- Source File : online-book-store/secure/admin_book.php
- Source File : online-book-store/secure/admin_delete.php
- Source File : online-book-store/secure/admin_edit.php
- Source File : online-book-store/secure/admin_signout.php
- Source File : online-book-store/secure/admin_verify.php
- Source File : online-book-store/secure/functions/admin.php
- Source File : online-book-store/secure/models/admin/Admin.php

2. Rule Title : Purchases | Ordering

- Source File : online-book-store/secure/cart.php
- Source File : online-book-store/secure/purchase.php
- Source File : online-book-store/secure/functions/cart_functions.php
- Source File : online-book-store/secure/models/orders/OrderDetails.php
- Source File : online-book-store/secure/models/orders/Orders.php
- Source File : online-book-store/secure/models/orders/OrdersTest.php
- Source File : online-book-store/secure/models/orders/UnAvailableOrderDetails.php

3. Rule Title : Models

- Source File : online-book-store/secure/models/admin/Admin.php
- Source File : online-book-store/secure/models/customer/Customers.php
- Source File : online-book-store/secure/models/goods/Address.php
- Source File : online-book-store/secure/models/goods/Author.php
- Source File : online-book-store/secure/models/goods/Book.php
- Source File : online-book-store/secure/models/goods/BookTest.php
- Source File : online-book-store/secure/models/goods/Genre.php
- Source File : online-book-store/secure/models/goods/Publishers.php
- Source File : online-book-store/secure/models/interfaces/Damage.php
- Source File : online-book-store/secure/models/orders/OrderDetails.php
- Source File : online-book-store/secure/models/orders/Orders.php
- Source File : online-book-store/secure/models/orders/OrdersTest.php
- Source File : online-book-store/secure/models/orders/UnAvailableOrderDetails.php
- Source File : online-book-store/secure/models/reviews/Review.php
- Source File : online-book-store/secure/models/serve/FileInterface.php
- Source File : online-book-store/secure/models/serve/Image.php

4. Rule Title : Controllers

- Source File : online-book-store/secure/controllers/DatabaseTalking/Take.php
- Source File : online-book-store/secure/controllers/DatabaseTalking/Talking.php

Identified Files Path

In this section, you'll find a comprehensive list of project file paths that were parsed during the analysis. This list was generated based on the selected file types during the scanning process and was used to identify the areas of interest listed in the previous sections. It's important to note that this list includes all the files, making it a superset of the list provided in the "Parsed Paths - Areas of Interest" section.

It is still recommended for code reviewers to review this list for any potentially interesting files that may have been overlooked in the "Parsed Paths - Areas of Interest" section.

- Source File : online-book-store/secure/admin.php
- Source File : online-book-store/secure/admin_add.php
- Source File : online-book-store/secure/admin_book.php
- Source File : online-book-store/secure/admin_delete.php
- Source File : online-book-store/secure/admin_edit.php
- Source File : online-book-store/secure/admin_signout.php
- Source File : online-book-store/secure/admin_verify.php
- Source File : online-book-store/secure/book.php
- Source File : online-book-store/secure/bookPerPub.php
- Source File : online-book-store/secure/books.php
- Source File : online-book-store/secure/cart.php
- Source File : online-book-store/secure/checkout.php
- Source File : online-book-store/secure/contact.php
- Source File : online-book-store/secure/edit_book.php
- Source File : online-book-store/secure/empty_session.php
- Source File : online-book-store/secure/index.php
- Source File : online-book-store/secure/process.php
- Source File : online-book-store/secure/publisher_list.php
- Source File : online-book-store/secure/purchase.php
- Source File : online-book-store/secure/verify.php
- Source File : online-book-store/secure/controllers/DatabaseTalking/Take.php
- Source File : online-book-store/secure/controllers/DatabaseTalking/Talking.php
- Source File : online-book-store/secure/database/readme.txt.txt
- Source File : online-book-store/secure/functions/admin.php
- Source File : online-book-store/secure/functions/cart_functions.php
- Source File : online-book-store/secure/functions/database_functions.php
- Source File : online-book-store/secure/models/admin/Admin.php
- Source File : online-book-store/secure/models/customer/Customers.php
- Source File : online-book-store/secure/models/goods/Address.php
- Source File : online-book-store/secure/models/goods/Author.php
- Source File : online-book-store/secure/models/goods/Book.php
- Source File : online-book-store/secure/models/goods/BookTest.php
- Source File : online-book-store/secure/models/goods/Genre.php
- Source File : online-book-store/secure/models/goods/Publishers.php
- Source File : online-book-store/secure/models/interfaces/Damage.php
- Source File : online-book-store/secure/models/orders/OrderDetails.php
- Source File : online-book-store/secure/models/orders/Orders.php
- Source File : online-book-store/secure/models/orders/OrdersTest.php
- Source File : online-book-store/secure/models/orders/UnAvailableOrderDetails.php

- Source File : online-book-store/secure/models/reviews/Review.php
- Source File : online-book-store/secure/models/serve/FileInterface.php
- Source File : online-book-store/secure/models/serve/Image.php
- Source File : online-book-store/secure/template/footer.php
- Source File : online-book-store/secure/template/header.php