

# **Library Book Borrowing Tracker - Project Report**

## **Project Overview**

**This PHP-based Library Book Borrowing Tracker implements comprehensive CRUD (Create, Read, Update, Delete) operations to manage book records and borrowing transactions. The system provides a user-friendly menu interface for librarians to efficiently track their book inventory and borrowing status.**

## **Technical Implementation**

### **### 1. CRUD Operations Implementation**

- \*\*Create\*\*:** Add new books with unique ID, title, and author validation
- \*\*Read\*\*:** View all books with real-time status display and filtering
- \*\*Update\*\*:** Borrow and return operations that modify book status and timestamps
- \*\*Delete\*\*:** Clear all books functionality (bulk delete operation)

### **### 2. Core Features Delivered**

#### #### Menu System

The application uses a GET parameter-based navigation system with four main operations:

- **\*\*Add New Book\*\*** (`?action=add`): Form-based book creation with validation
- **\*\*View All Books\*\*** (`?action=view`): Tabular display of entire library collection
- **\*\*Borrow Book\*\*** (`?action=borrow`): Status change from "Available" to "Borrowed"
- **\*\*Return Book\*\*** (`?action=return`): Status change from "Borrowed" to "Available"

#### #### Data Management

- **\*\*Session Storage\*\***: Primary data storage using `\$\_SESSION['books']` array
- **\*\*File Backup\*\***: Automatic saving to "library\_books.txt" for persistence
- **\*\*Data Structure\*\***: Each book stored as associative array with ID, title, author, status, and borrowed\_date

### ### 3. Condition Checks and Loops

#### #### Validation Logic

```php

```
// Prevent duplicate book IDs
if (bookIdExists($_SESSION['books'], $book_id)) {
    $errors[] = "Book ID already exists";
}

// Prevent borrowing already borrowed books
if ($_SESSION['books'][$book_index]['status'] == 'Available')
{
    // Allow borrowing
} else {
    $error_message = "Book is already borrowed!";
}
```

## Loop Usage

- **\*\*foreach loops\*\*** for displaying book collections and calculating statistics

**Data processing loops\*\*** for file operations and search functionality

**Validation loops:** to check duplicate IDs and find books by ID

## 4. Advanced Features

**Real-time Statistics:** Dynamic calculation of total, available, and borrowed books

- **\*\*Status Tracking\*\***: Timestamp recording for borrowing transactions
- **\*\*Input Validation\*\***: Comprehensive server-side validation with error handling
- **\*\*Export Functionality\*\***: CSV download capability for record keeping
- **\*\*Responsive Design\*\***: Bootstrap-based UI that works on all devices

## **## Code Structure Benefits**

- **\*\*Modular Functions\*\***: Separate functions for file operations, book searches, and validation
- **\*\*Error Handling\*\***: Comprehensive error checking with user-friendly messages
- **\*\*Data Persistence\*\***: Dual storage (session + file) ensures data reliability
- **\*\*Security\*\***: Input sanitization and validation prevent data corruption
- **\*\*Scalability\*\***: Array-based structure easily accommodates growing book collections

## **## Key Learning Outcomes**

This implementation demonstrates mastery of PHP fundamentals including array manipulation, file I/O operations, form processing, session management, and

**conditional logic. The project successfully combines CRUD operations with practical library management requirements while maintaining clean, maintainable code structure.**