

# PHP assignment (home project)


**Due** Jun 12 by 11:59pm **Points** 20 **Submitting** a file upload **File Types** zip  
**Available** until Jun 12 at 11:59pm

This assignment was locked Jun 12 at 11:59pm.

## IK-Library - PHP Home Assignment


Daniel Jackson wrote down all the essential information during his visits to different worlds and collected many written memories. General Hammond and the General Staff decided to create a website to catalogue the books collected so far. They entrust you with this task.

**IK-Library > Home**




**Julie Fortune -  
Sacrifice Moon**

Edit



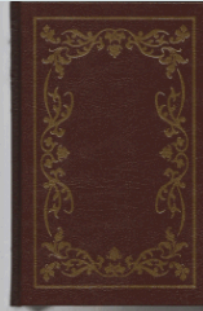
**Sonny Whitelaw -  
City of the Gods**

Edit




**Suzanne Wood -  
Barque of Heaven**

Edit




**Tim Waggoner -  
Valhalla**

Edit



**Amy Griswold -  
Heart's Desire**

Edit



**Geonn Cannon -  
Female of the  
Species**

Edit

IK-Library | ELTE IK Webprogramming

## Main functions

Your task is to create a server-side application written in PHP where users can give feedback on books and mark them as read. The help section below describes the possible storage form of these two data structures (book and user), but of course, you can also structure your data differently.

- Books must be added to the data in advance.
- An admin user whose login data has been recorded must be included among the users. See details at Admin functions.

## Main page / List page

- A title and a short description should be displayed with static text on the list page or the main page.
- The main page is available to unidentified users, who can freely browse the content. The books in the system should be listed here.
- The books should have a link (this can even be on a picture) that leads to the book's detailed page.
- If the user is logged in, the "User profile" page should be accessible via a link.

## Book details

- On the book details page, the title, author, description, cover image, year of publication, source planet, average rating and ratings should be displayed.
- A rating button should be available on the books page if the user is logged in. The user can write a review for any book or mark them as read.
- You can also access the main page and other menu items.

## User details

- The page lists the user's data, the reviews written by him and the books he has read.

## Authentication pages

- It should be possible to access the login and registration page from the main page.
- You must enter your username, email address, and password twice during registration. All are required: the username must be unique, the email address format must be correct, and the passwords must match.
- Error messages should be displayed in the event of a registration error! The form should be stateful! After successful registration, the user should be logged in to the main page.
- During the login, we can identify with the username and password.
- Report errors that occur during login on the form page! After successfully logging in, go to the main page!

## Admin functions

- Create a special user, admin (username: admin, password: admin).
- The admin user should be able to create a new book.

## Extra functions

See the scoring!

## More expectations

- A sophisticated appearance is important. This does not necessarily mean creating a fancy page, but it does mean that the page displays well in 1024x768 resolution. For this, you can also use a minimalist design, create your own CSS turbocharged with different wallpapers and graphic elements, or use any CSS framework.
- The entered data must be checked on the server during the final request processing. To turn off browser validation, add the 'novalidate' attribute to the attributes of your form elements!

- The completed assignment must be compressed and uploaded to the Canvas system by the deadline, together with all the necessary documents and the README.md file located in the program's root folder.
- No framework or external PHP function library can be used for its implementation. A maximum of CSS frameworks can be used.
- If you'd like, you can use this skeleton as a starting point: [skeleton.zip](#)
- **The README.md file must be filled out completely (see under scoring section).**

# Help

## Planning

We would also like to help those who find it more challenging to see and plan a more significant task. The entire task can be planned and developed afterwards, but the following steps can also be used to solve smaller sub-tasks:

1. Create a static HTML prototype of the application to be developed! In the first step, design the list page, the page detailing the book, etc., using only HTML and CSS. There are pages where there is conditional information, plan these as well, and then remove them later with PHP. You can also test CSS statically, e.g. on the list page, how the books should be displayed, you can enter this statically. You can connect the individual pages with links.
2. Think about what data you will need. What must be stored, in which fields?
  - Where do you store users?
  - Where do you store the books?
  - How do you store the evaluations?
  - How do you connect users with the books they read?
3. How can you extract the correct data for your pages from the previously designed data structures? Create a function for each of these, but it is much better if you implement them as methods of the given Storage class.
  - How do you get a user's ratings back?
4. There are two ways to use forms:
  - What should happen in case of success?
  - How do I detect and display an error, and how does the form become stateful?

## Data storage

- Data: There are two types of data in the task: Book User.
  - The following data of the books must be stored:
    - The book's title
    - The author of the book
    - The year the book was written
    - Description of the book (optional)
    - Image from the cover - this can be an image file path or image title (optional)
    - The name of the planet where the tear comes from (optional)
      - Starts with P, continues with two arbitrary characters, followed by a dash and 3 or 4 digits (see sample data above)
  - User data:
    - User name
    - User's e-mail address
    - User Password
    - User's last login
    - Admin authority

- An example of how to store books. Using the Storage class shown in class is highly recommended, rather than writing it from scratch! This example is only helpful; it must be converted to work with class!

```
{
  "book0": {
    "title": "City of the Gods",
    "author": "Sonny Whitelaw",
    "description": "Notes about Argos, the planet originally under the control of the Goa'uld Pelops. A definitive Greek ancestry decorates the structures, as well as the clothing of the Argosians, the planets occupants who preferred to call themselves the Chosen. For some time Pelops operated on the planet, experimenting on the human population, much like Nirrti has done, attempting to expand their lifespan.",
    "year": 2005,
    "image": "cityOfTheGods.jpg",
    "planet": "P3X-8596"
  },
  "book1": {
    "title": "The Power Behind The Throne",
    "author": "Steven Savile",
    "description": "P3X-797 is a world divided into a dark and light side. The Stargate resides on the dark side of the planet where the \"Touched\" exist. The light side is inhabited by the \"Untouched\". A histaminolytic virus inhabits the world, forcing anyone with histamine in their bodies to succumb to primeval tendencies. The leader of the light side is Tuplo.",
    "year": 2007,
    "image": "thePowerBehingTheThrone.jpg",
    "planet": "P3X-797"
  }
}
```

## Scoring

Solving the task can earn you 20 points. There are minimum requirements without which the application cannot be accepted. You can get a maximum of 5 additional points for extra tasks, but all extra tasks are worth more, so you can choose from them. If someone does everything, they can get a maximum of 25 points for the assignment.

### Must be completed as a minimum (we will not accept without it, 6 points)

- **0.0 points** Readme.md file: completed, uploaded
- **0.0 points** Main page: appears
- **1.0 point** Main page: list of all books, e.g. with pictures
- **1.0 point** Main page: clicking on the name of the book takes you to the corresponding book details page
- **1.0 point** Book details: The book details are displayed.
- **1.0 point** Book details: The book cover is displayed.
- **2.0 points** Admin: Create a new book: error handling, successful save

### The basic tasks (14 points)

- **1.0 point** Book details: The book's average rating is displayed.
- **1.0 point** Book details: The evaluations for the book are displayed.
- **0.5 points** Registration form: Contains appropriate elements
- **1.0 point** Registration form: Handling of error cases, error message, status maintenance
- **0.5 points** Registration form: Successful registration
- **1.0 point** Login: Handling faulty cases
- **0.5 points** Login: Successful login
- **0.5 points** Logout
- **0.5 points** Main page: The user's name is displayed
- **0.5 points** Main page: Clicking on the user name will take you to the user's details page.
- **1.0 point** Main page: Books can be filtered by genre.
- **1.0 point** User details: The user's name, e-mail address, and last login date are displayed.

- **0.5 points** User details: The evaluations written by the user are displayed.
- **0.5 points** Admin: You can log in with the data of the admin user
- **0.5 points** Admin: Creating a new book is only available with the Admin user
- **0.5 points** Evaluation: The evaluation form appears.
- **0.5 points** Evaluation: The form contains the correct elements.
- **1.0 point** Evaluation: Handling faulty cases, error messages, and status maintenance.
- **0.5 points** Admin: Modification of book data: available with admin user
- **1.0 point** Demanding design

## Extra tasks (max plus 5 points)

- **1.0 point** Main page: The books are sorted by average rating.
- **1.0 point** Admin: Modification of book data: error handling, status maintenance, successful saving
- **1.0 point** Book details: Logged-in users should be able to mark books as read.
- **1.0 point** Book details: The users who have read the given book are displayed, with the time of reading
- **1.0 point** User details: The books read by the user are displayed, in descending order according to the time of reading.

## README.md file

Without a properly filled README.md file, the solution will not be accepted! The README.md file must contain the following statement (<> signs are not required):

```
<Student's name>
<Neptun code>
Web programming - assignment
This solution was submitted and created by the student above for the Web Programming course.
I declare that this solution is my own work. I did not copy or use it from a third party
solutions from third parties. I did not forward my solution to my fellow students, nor did I publish it.
Eötvös Loránd University Student Requirements System
(Organizational and Operational Regulations of ELTE, Volume II, § 74/C) states that as long as,
as long as a student has been working on the work - or at least a significant part of it - of another student
of another student's work as his or her own, it is a disciplinary offence.
The most serious consequence of a disciplinary offence is dismissal from the university.

### Must be completed as a minimum (we will not accept without it, 6 points)

[ ] 0.0 points Readme.md file: completed, uploaded
[ ] 0.0 points Main page: appears
[ ] 1.0 point Main page: list of all books, e.g. with pictures
[ ] 1.0 point Main page: clicking on the name of the book takes you to the corresponding book details page
[ ] 1.0 point Book details: The book details are displayed.
[ ] 1.0 point Book details: The book cover is displayed.
[ ] 2.0 points Admin: Create a new book: error handling, successful save

### The basic tasks (14 points)
[ ] 1.0 point Book details: The book's average rating is displayed.
[ ] 1.0 point Book details: The evaluations for the book are displayed.
[ ] 0.5 points Registration form: Contains appropriate elements
[ ] 1.0 point Registration form: Handling of error cases, error message, status maintenance
[ ] 0.5 points Registration form: Successful registration
[ ] 1.0 point Login: Handling faulty cases
[ ] 0.5 points Login: Successful login
[ ] 0.5 points Logout
[ ] 0.5 points Main page: The user's name is displayed
[ ] 0.5 points Main page: Clicking on the user name will take you to the user's details page.
[ ] 1.0 point Main page: Books can be filtered by genre.
[ ] 1.0 point User details: The user's name, e-mail address, and last login date are displayed.
[ ] 0.5 points User details: The evaluations written by the user are displayed.
[ ] 0.5 points Admin: You can log in with the data of the admin user
[ ] 0.5 points Admin: Creating a new book is only available with the Admin user
[ ] 0.5 points Evaluation: The evaluation form appears.
```

[ ] 0.5 points Evaluation: The form contains the correct elements.  
[ ] 1.0 point Evaluation: Handling faulty cases, error messages, and status maintenance.  
[ ] 0.5 points Admin: Modification of book data: available with admin user  
[ ] 1.0 point Demanding design

### Extra tasks (max plus 5 points)

[ ] 1.0 point Main page: The books are sorted by average rating.  
[ ] 1.0 point Admin: Modification of book data: error handling, status maintenance, successful saving  
[ ] 1.0 point Book details: Logged-in users should be able to mark books as read.  
[ ] 1.0 point Book details: The users who have read the given book are displayed, with the time of reading.  
[ ] 1.0 point User details: The books read by the user are displayed, in descending order according to the time of reading.

**We wish everyone good work for the solution!**