Ministerul Educaţiei al Republicii Moldova
Universitatea de Stat din Moldova
Facultatea de Matematică şi Informatică Departamentul
“Informatica »

Веб-программирование с использованием PHP

Индивидуальная работа.

IAFR2202ru

David Raisa

Nichita Nartea

- Chişinău 2025 -

Задание:

Требования к выполнению

Основные условия:

1. Разработайте веб-приложение (традиционное веб-приложение, веб-сервис, REST API, мессенджер-бот) средней сложности, содержащее функционал, реализованный на стороне сервера.
2. Для реализации можно использовать любые frontend и backend веб-технологии.
3. Разрешается работа в командах по два человека.
4. Работа должна быть загружена на GitHub, а ссылка прикреплена в Moodle.
5. Индивидуальная работа должна быть представлена преподавателю и коллегам.

Практическая часть

1. Веб-приложение должно содержать:
2. Компонент (страницы) с общедоступным доступом для всех пользователей:
    a. Минимум 2-3 элемента контента должны генерироваться динамически с использованием серверных скриптов.
3. Данные для динамического контента должны быть извлечены из файлов и/или баз данных.
4. Реализуйте как минимум одну форму для сбора данных и одну форму для поиска в базе данных.
5. Возможность разработки собственного REST API, взаимодействующего с базой данных для обработки запросов и предоставления данных.
6. Компонент (страницы) с защищенным доступом (только для авторизованных пользователей):
    a. Для этого компонента создайте роль "администратор".
    b. Пользователь с ролью "администратор" должен иметь доступ к 3-7 дополнительным функциям, включая:
        i. Создание новой учетной записи с ролью "администратор".
        ii. Управление данными из базы данных (просмотр, добавление, изменение и удаление данных).

Требования к безопасности

1. Валидируйте все данные, введенные в формы, чтобы предотвратить внедрение вредоносного кода.
2. Доступ к закрытым частям приложения должен быть защищён аутентификацией (например, пароль). Дополнительно можно использовать CAPTCHA.
3. Пароли должны храниться в базе данных с использованием безопасных хэш-функций и/или "соли".
4. Используйте сессии и переменные сессии (или токены) для управления доступом, чтобы предотвратить обход аутентификации.

Архитектура приложения

1. Постройте приложение на модульной архитектуре для удобства расширения и поддержки.
2. Опционально используйте архитектуру MVC (Model-View-Controller), чтобы улучшить структуру приложения.

Структура:

/project-root

1. data/
    a. books.json - Здесь хранятся книжонки
2. public/
    a. index.php - Списочек книг
    b. search.php - Поиск по книгам
    c. login.php - Вход в админку
    d. dashboard.php - Админ-панель
    e. logout.php - Выход
3. admin/
    a. add_book.php - Добавить книгу
    b. edit_book.php - Изменить книгу
    c. delete_book.php - Удалить книгу
4. includes/
    a. auth.php - Проверка сессии
    b. functions.php - Работа с JSON-файлом, вместо БД (мне лень еще и бд консруировать)
5. /api/
    a. books.php

Код:

books.json (начало)

```
[
  {
    "id": 1,
    "title": "Цой жив?",
    "author": "Цой",
    "year": 1990
  },
  {
    "id": 2,
    "title": "Навальный жив?",
    "author": "Навальный",
    "year": 2024
  }
]
```

functions.php для json

```php
<?php
function loadBooks() {
    $json = file_get_contents(__DIR__ . '/../data/books.json');
    return json_decode($json, true);
}

function saveBooks($books) {
    $json = json_encode($books, JSON_PRETTY_PRINT | JSON_UNESCAPED_UNICODE);
    file_put_contents(__DIR__ . '/../data/books.json', $json);
}
```

index.php – список книг

```php
<?php
require_once '../includes/functions.php';

$books = loadBooks();
?>

<h1>Список книг</h1>

<ul>
<?php foreach ($books as $book): ?>
    <li>
        <strong><?php echo htmlspecialchars($book['title']); ?></strong>
        — <?php echo htmlspecialchars($book['author']); ?> (<?php echo $book['year']; ?>)
    </li>
<?php endforeach; ?>
</ul>
```

search.php

```php
<?php
require_once '../includes/functions.php';
$books = loadBooks();
$result = [];
if (isset($_GET['q']) && $_GET['q'] !== '') {
    $query = strtolower($_GET['q']);
    foreach ($books as $book) {
            if (str_contains(strtolower($book['title']), $query)) {
```

```
            $result[] = $book;
        }
    }
}
?>

<form method="get">
    <input type="text" name="q" placeholder="Поиск по названию">
    <button type="submit">Найти</button>
</form>

<?php if (!empty($result)): ?>
    <h2>Результаты:</h2>
    <ul>
        <?php foreach ($result as $book): ?>
            <li>
                <strong><?php echo htmlspecialchars($book['title']); ?></strong>
                — <?php echo htmlspecialchars($book['author']); ?> (<?php echo $book['year']; ?>)
            </li>
        <?php endforeach; ?>
    </ul>
<?php elseif (isset($_GET['q'])): ?>
    <p>Ничего не найдено</p>
<?php endif; ?>
```

login.php

```
<?php
session_start();

// пример
$admin_login = 'admin';
$admin_password = '1234';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $login = $_POST['login'] ?? '';
    $password = $_POST['password'] ?? '';
```

```
    if ($login === $admin_login && $password === $admin_password) {
        $_SESSION['admin'] = true;
        header('Location: dashboard.php');
        exit();
    } else {
        $error = "Неверный логин или пароль";
    }
}
?>

<h2>Вход для администратора</h2>
<form method="post">
    <input type="text" name="login" placeholder="Логин">
    <input type="password" name="password" placeholder="Пароль">
    <button type="submit">Войти</button>
</form>
<?php if (isset($error)) echo "<p>$error</p>"; ?>
```

dashboard.php (панелька админа)

```
<?php
session_start();
require_once '../includes/functions.php';

if (!isset($_SESSION['admin'])) {
    header('Location: login.php');
    exit();
}
$books = loadBooks();
?>
```

```
<h1>Панель администратора</h1>
<p><a href="add_book.php">Добавить книгу</a> | <a href="logout.php">Выйти</a></p>
<table border="1" cellpadding="5">
    <tr>
        <th>ID</th><th>Название</th><th>Автор</th><th>Год</th><th>Действия</th>
    </tr>
    <?php foreach ($books as $book): ?>
        <tr>
            <td><?= $book['id'] ?></td>
            <td><?= htmlspecialchars($book['title']) ?></td>
            <td><?= htmlspecialchars($book['author']) ?></td>
            <td><?= $book['year'] ?></td>
            <td>
                <a href="edit_book.php?id=<?= $book['id'] ?>">Редактировать</a> |
                <a href="delete_book.php?id=<?= $book['id'] ?>" onclick="return confirm('Точно удалить?');">Удалить</a>
            </td>
        </tr>
    <?php endforeach; ?>
</table>
```

logout.php

```
<?php
session_start();
session_destroy();
header('Location: login.php');
exit();
```

add_book.php

```php
<?php
session_start();
require_once '../includes/functions.php';

if (!isset($_SESSION['admin'])) {
    header('Location: ../public/login.php');
    exit();
}

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $title = $_POST['title'] ?? '';
    $author = $_POST['author'] ?? '';
    $year = $_POST['year'] ?? '';
    $books = loadBooks();
    $newId = end($books)['id'] + 1;
    $newBook = [
        'id' => $newId,
        'title' => $title,
        'author' => $author,
        'year' => $year
    ];
    $books[] = $newBook;
    saveBooks($books);
    header('Location: dashboard.php');
    exit();
}
?>
<h2>Добавить новую книгу</h2>
<form method="post">
    <input type="text" name="title" placeholder="Название книги" required><br>
```

```
    <input type="text" name="author" placeholder="Автор" required><br>
    <input type="number" name="year" placeholder="Год" required><br>
    <button type="submit">Добавить</button>
</form>
<p><a href="dashboard.php">Назад в (будущее) админку</a></p>
```

edit_book.php

```
<?php
session_start();
require_once '../includes/functions.php';
if (!isset($_SESSION['admin'])) {
    header('Location: login.php');
    exit();
}
$books = loadBooks();
$id = $_GET['id'] ?? null;
if (!$id) {
    echo "Нет ID книги.";
    exit();
}
$book = null;
foreach ($books as $b) {
    if ($b['id'] == $id) {
        $book = $b;
        break;
    }
}

if (!$book) {
```

```
    echo "Книга не найдена.";
    exit();
}
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    foreach ($books as &$b) {
        if ($b['id'] == $id) {
            $b['title'] = $_POST['title'];
            $b['author'] = $_POST['author'];
            $b['year'] = $_POST['year'];
            break;
        }
    }
    saveBooks($books);
    header('Location: dashboard.php');
    exit();
}
?>

<h2>Редактировать книгу</h2>
<form method="post">
    <input type="text" name="title" value="<?= htmlspecialchars($book['title']) ?>" required><br>
    <input type="text" name="author" value="<?= htmlspecialchars($book['author']) ?>" required><br>
    <input type="number" name="year" value="<?= $book['year'] ?>" required><br>
    <button type="submit">Сохранить</button>
</form>
<p><a href="dashboard.php">Назад</a></p>
```

delete_book.php

```
<?php
```

```
session_start();
require_once '../includes/functions.php';

if (!isset($_SESSION['admin'])) {
    header('Location: login.php');
    exit();
}

$id = $_GET['id'] ?? null;
if (!$id) {
    echo "Нет ID.";
    exit();
}

$books = loadBooks();
$books = array_filter($books, fn($b) => $b['id'] != $id);
$books = array_values($books);
saveBooks($books);
header('Location: dashboard.php');
exit();
```

api/books.php

```
<?php
require_once '../includes/functions.php';
header("Content-Type: application/json");
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: GET, POST, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type");
$books = loadBooks();
$method = $_SERVER['REQUEST_METHOD'];
```

```
switch ($method) {
    case 'GET':
        echo json_encode($books, JSON_UNESCAPED_UNICODE);
        break;

    case 'POST':
        $data = json_decode(file_get_contents("php://input"), true);
        $newId = end($books)['id'] + 1;
        $newBook = [
            'id' => $newId,
            'title' => $data['title'] ?? '',
            'author' => $data['author'] ?? '',
            'year' => $data['year'] ?? ''
        ];
        $books[] = $newBook;
        saveBooks($books);
        echo json_encode(['message' => 'Книга добавлена']);
        break;

    case 'PUT':
        $data = json_decode(file_get_contents("php://input"), true);
        $id = $data['id'] ?? null;
        if (!$id) {
            http_response_code(400);
            echo json_encode(['error' => 'ID не указан']);
            exit();
        }
        foreach ($books as &$book) {
            if ($book['id'] == $id) {
                $book['title'] = $data['title'] ?? $book['title'];
```

```
            $book['author'] = $data['author'] ?? $book['author'];
            $book['year'] = $data['year'] ?? $book['year'];
            break;
        }
    }
    saveBooks($books);
    echo json_encode(['message' => 'Книга обновлена']);
    break;

case 'DELETE':
    $data = json_decode(file_get_contents("php://input"), true);
    $id = $data['id'] ?? null;
    if (!$id) {
        http_response_code(400);
        echo json_encode(['error' => 'ID не указан']);
        exit();
    }
    $books = array_filter($books, fn($b) => $b['id'] != $id);
    $books = array_values($books);
    saveBooks($books);
    echo json_encode(['message' => 'Книга удалена']);
    break;

default:
    http_response_code(405);
    echo json_encode(['error' => 'Метод не поддерживается']);
}
```

Получить книги:

curl http://localhost/api/books.php

Добавить книгу:

curl -X POST -H "Content-Type: application/json" -d '{"title":"Поставьте 5 пожалуйста","author":"Я Давид Раиса","year":2025}' http://localhost/api/books.php

Обновить:

curl -X PUT -H "Content-Type: application/json" -d '{"id":2,"title":"Обновлено"}' http://localhost/api/books.php

Удалить:

curl -X DELETE -H "Content-Type: application/json" -d '{"id":2}' http://localhost/api/books.php

Тестовые данные

- Логин: admin

- Пароль: 1234

##Возможности

- Публичный список книг

- Поиск по названию

- Авторизация сокамерника, ой, администратора

- Панель управления:

  - Добавление книг

  - Редактирование

  - Удаление

- REST API (`api/books.php`) с методами GET / POST / PUT / DELETE

Простое приложение для просмотра и управления списком книг.

Хранение данных в .json файле, без базы данных.

Список использованных источников

W3Schools Tryit Editor

433zuntjv - PHP - OneCompiler

HTML Online Editor (Compiler, Interpreter & Runner)