

## 10 سوال و تمرین جلسه ۳۸ با پاسخ کامل

### سوال ۱ – مفهومی (آسان)

**سوال:** چرا باید از Session استفاده کنیم و چه مشکلی را حل می‌کند؟

**پاسخ:** Session ها برای حل مشکل stateless بودن HTTP استفاده می‌شوند. به طور پیش فرض هیچ اطلاعاتی از درخواست‌های قبلی نگه نمی‌دارد Session. ها این امکان را فراهم می‌کنند:

- ردیابی وضعیت ورود کاربر بین صفحات مختلف
- نگهداری اطلاعات موقت کاربر
- مدیریت سبد خرید
- ذخیره تنظیمات کاربر

---

### سوال ۲ – کدنویسی (متوسط)

**سوال:** کد مربوط به بخش اعتبارسنجی در فایل store.php را کامل کنید:

```
<?php
$email = $_POST['email'];
$password = $_POST['password'];
$errors = [];

// کد اعتبارسنجی را کامل کنید
// ... کد شما ...

if (!empty($errors)) {
    // کد برگشت به فرم را کامل کنید
    // ... کد شما ...
}
```

**پاسخ:**

```

<?php
$email = $_POST['email'];
$password = $_POST['password'];
$errors = [];

// اعتبارسنجی ایمیل
if (!Validator::email($email)) {
    $errors['email'] = 'Please provide a valid email address';
}

// اعتبارسنجی رمز عبور (حداقل ۷ کاراکتر)
if (!Validator::string($password, 7, 255)) {
    $errors['password'] = 'Please provide a password of at least 7 characters';
}

if (!empty($errors)) {
    view('registration/create.view.php', ['errors' => $errors]);
    return;
}

```

---

**سوال ۳ – پایگاه داده (متوسط)**

**سوال:** کوئری SQL برای بررسی وجود کاربر با ایمیل مشخص چگونه نوشته می‌شود و چرا از prepared statement استفاده می‌کنیم؟

**پاسخ:**

```

$user = $db->query("SELECT * FROM users WHERE email = :email", [
    ':email' => $email
])->find();

```

**دلایل استفاده از Prepared Statement:**

۱. امنیت: جلوگیری از SQL Injection
  ۲. کارایی: کوئری compile شده قابل استفاده مجدد است
  ۳. خوانایی: کد تمیزتر و قابل فهم‌تر
  ۴. مدیریت داده: escape کردن خودکار کاراکترهای خاص
- 

## سوال (۴) Session Management - سخت)

**سوال:** کد کاملی بنویسید که پس از ثبت نام موفق، کاربر را وارد سیستم کند و به صفحه dashboard هدایت کند. همچنین session regeneration را نیز اعمال کنید.

**پاسخ:**

```
// ایجاد کاربر جدید در دیتابیس
$db->query("INSERT INTO users (email, password) VALUES
(:email, :password)", [
    ':email' => $email,
    ':password' => password_hash($password,
PASSWORD_DEFAULT) // هش کردن رمز عبور
]);

// کاربر تازه ایجاد شده ID بازیابی
$userId = $db->lastInsertId();

// برای امنیت session تجدید شناسه
session_regenerate_id(true);

// تنظیم اطلاعات session
$_SESSION['user'] = [
    'id' => $userId,
    'email' => $email,
    'logged_in' => true,
    'login_time' => time()
];
```

```
// هدایت به داشبورد
header("Location: /dashboard");
exit();
```

---

## سوال ۵ - امنیت (سخت)

سوال: چه مشکلات امنیتی در کد ارائه شده وجود دارد و چگونه آن‌ها را حل می‌کنید؟

پاسخ:

مشکلات امنیتی موجود:

۱. رمز عبور Plain Text: رمز عبور بدون هش ذخیره می‌شود
۲. عدم وجود CSRF Token: فرم در برابر حملات CSRF آسیب‌پذیر است
۳. Session Fixation: session\_regenerate\_id استفاده نشده
۴. عدم محدودیت تعداد تلاش: امکان brute force attack وجود دارد

راه‌حل‌ها:

۱. // هش کردن رمز عبور

```
$hashedPassword = password_hash($password,
PASSWORD_DEFAULT);
```

۲. // اضافه کردن CSRF Token به فرم

// در: view

```
<input type="hidden" name="csrf_token" value="<?=
generate_csrf_token() ?>">
```

// در: controller

```
if (!verify_csrf_token($_POST['csrf_token'])) {
    die('Invalid CSRF token');
}
```

۳. // تجدید شناسه session

```
session_regenerate_id(true);
```

```

4. // محدودیت تعداد تلاش (rate limiting)
if (check_rate_limit($email)) {
    $errors['general'] = 'Too many attempts. Try again later.';
}

```

---

## سوال ۶ - Debugging - متوسط)

**سوال:** کاربر پس از ثبت نام هدایت نمی‌شود و صفحه خالی نمایش داده می‌شود. علت احتمالی چیست و چگونه debug کنید؟

**پاسخ:**

**علل احتمالی:**

۱. عدم وجود `exit()` پس از `header()`

۲. خروجی قبل از `header` حتی یک (space)

۳. خطا در کوئری دیتابیس

۴. مشکل در تنظیمات session

**راه‌حل‌های Debug:**

1. // چک کردن خطاهای PHP

```

error_reporting(E_ALL);
ini_set('display_errors', 1);

```

2. // اضافه کردن `exit()` پس از `header`

```

header("Location: /dashboard");
exit(); // حتماً اضافه کنید

```

3. // چک کردن موفقیت کوئری

```

$result = $db->query("INSERT INTO users...");
if (!$result) {
    die("Database error: " . $db->errorInfo());
}

```

4. // نمایش session برای debug

```
var_dump($_SESSION);
```

5. // چک کردن header ارسالی

```
if (headers_sent($file, $line)) {  
    echo "Headers already sent in $file:$line";  
}
```

---

### سوال (۷) UI/UX - آسان)

**سوال:** کد HTML و CSS برای نمایش پیام‌های خطا زیر فیلدهای فرم را بنویسید که user-friendly و responsive باشد.

**پاسخ:**

--> فیلد ایمیل <!--

```
<div class="mb-4">  
    <label for="email" class="block text-sm font-medium  
text-gray-700 mb-2">  
        Email Address  
    </label>  
    <input type="email"  
        name="email"  
        id="email"  
        value="<?= htmlspecialchars($email ?? '')  
?>"  
        class="w-full px-3 py-2 border rounded-lg  
focus:outline-none focus:ring-2  
        <?= isset($errors['email']) ?  
'border-red-500 focus:ring-red-500' : 'border-gray-300  
focus:ring-blue-500' ?>">  
  
    <!-- نمایش خطا -->  
    <?php if(isset($errors['email'])): ?>  
        <div class="mt-1 flex items-center">
```

```

        <svg class="w-4 h-4 text-red-500 mr-1"
fill="currentColor" viewBox="0 0 20 20">
            <path fill-rule="evenodd" d="M18 10a8 8
0 11-16 0 8 8 0 0116 0zm-7 4a1 1 0 11-2 0 1 1 0 012
0zm-1-9a1 1 0 00-1 1v4a1 1 0 102 0V6a1 1 0 00-1-1z"
clip-rule="evenodd"></path>
        </svg>
        <p class="text-sm text-red-600"><?=
htmlspecialchars($errors['email']) ?></p>
    </div>
    <?php endif; ?>
</div>

```

---

## سوال ۸ – تمرین عملی (سخت)

**سوال:** یک میدل ویر (Middleware) بنویسید که بررسی کند کاربر وارد شده یا نه و در صورت عدم ورود، او را به صفحه login هدایت کند.

**پاسخ:**

**فایل: middleware/Auth.php:**

```

<?php

class Auth {
    public static function requireLogin() {
        // بررسی وجود session
        if (session_status() == PHP_SESSION_NONE) {
            session_start();
        }

        // بررسی ورود کاربر
        if (!isset($_SESSION['user']) ||
!$_SESSION['user']) {
            // فعلی برای بازگشت پس از ورود ذخیره
            $_SESSION['intended_url'] =
$_SERVER['REQUEST_URI'];

```

```

        // هدایت به صفحه ورود
        header("Location: /login");
        exit();
    }

    // اختیاری session بررسی انقضای
    if (isset($_SESSION['user']['login_time'])) {
        $sessionLifetime = 3600; // 1 ساعت
        if (time() -
$_SESSION['user']['login_time'] > $sessionLifetime) {
            self::logout();
            header("Location: /login?expired=1");
            exit();
        }
    }

    return true;
}

public static function logout() {
    if (session_status() == PHP_SESSION_NONE) {
        session_start();
    }

    // پاک کردن تمام داده‌های session
    $_SESSION = [];

    // حذف کوکی session
    if (ini_get("session.use_cookies")) {
        $params = session_get_cookie_params();
        setcookie(session_name(), '', time() -
42000,
            $params["path"], $params["domain"],
            $params["secure"], $params["httponly"]
        );
    }
}

```



```

        // نابودی session
        session_destroy();
    }

    public static function user() {
        return $_SESSION['user'] ?? null;
    }

    public static function isLoggedIn() {
        return isset($_SESSION['user']) &&
$_SESSION['user'];
    }
}

```

**استفاده از میدل‌ویر:**

```

// در controller های محافظت شده
<?php
require_once 'middleware/Auth.php';

Auth::requireLogin(); // بررسی ورود کاربر

// باقی کد controller...

```

---

**سوال ۹ – بهینه‌سازی (متوسط)**

**سوال:** کد فعلی store.php بسیار طولانی شده. آن را به چندین کلاس و متد تقسیم کنید تا maintainable باشد.

**پاسخ:**

**کلاس UserRegistration.php:**

```

<?php

class UserRegistration {
    private $db;

```

```

private $validator;

public function __construct(Database $db, Validator
$validator) {
    $this->db = $db;
    $this->validator = $validator;
}

public function register($email, $password) {
    // اعتبارسنجی
    $errors = $this->validateInput($email,
$password);
    if (!empty($errors)) {
        return ['success' => false, 'errors' =>
$errors];
    }

    // بررسی وجود کاربر
    if ($this->userExists($email)) {
        return ['success' => false, 'message' =>
'User already exists'];
    }

    // ایجاد کاربر
    $userId = $this->createUser($email, $password);

    // ورود خودکار
    $this->loginUser($userId, $email);

    return ['success' => true, 'user_id' =>
$userId];
}

private function validateInput($email, $password) {
    $errors = [];

    if (!$this->validator->email($email)) {
        $errors['email'] = 'Please provide a valid
email address';
    }

```

```

    }

    if (!$this->validator->string($password, 7,
255)) {
        $errors['password'] = 'Password must be at
least 7 characters';
    }

    return $errors;
}

private function userExists($email) {
    $user = $this->db->query("SELECT id FROM users
WHERE email = :email", [
        ':email' => $email
    ]->find());

    return $user !== false;
}

private function createUser($email, $password) {
    $this->db->query("INSERT INTO users (email,
password) VALUES (:email, :password)", [
        ':email' => $email,
        ':password' => password_hash($password,
PASSWORD_DEFAULT)
    ]);

    return $this->db->lastInsertId();
}

private function loginUser($userId, $email) {
    session_regenerate_id(true);
    $_SESSION['user'] = [
        'id' => $userId,
        'email' => $email,
        'login_time' => time()
    ];
}
}

```

## store.php: جدید Controller

```
<?php

$email = $_POST['email'] ?? '';
$password = $_POST['password'] ?? '';

$db = App::resolve(Database::class);
$validator = new Validator();
$registration = new UserRegistration($db, $validator);

$result = $registration->register($email, $password);

if ($result['success']) {
    header("Location: /dashboard");
    exit();
} else {
    if (isset($result['errors'])) {
        view('registration/create.view.php', ['errors'
=> $result['errors']]);
    } else {
        header("Location: /login");
        exit();
    }
}
```

---

### سوال ۱۰ - تمرین جامع (خیلی سخت)

سوال: یک سیستم کامل Remember Me پیاده‌سازی کنید که شامل موارد زیر باشد:

- کوکی امن برای Remember Me
- جدول remember\_tokens در دیتابیس
- خروج خودکار پس از مدت زمان مشخص
- امکان logout از همه دستگاه‌ها

پاسخ:

## 1. جدول دیتابیس:

```
CREATE TABLE remember_tokens (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT NOT NULL,  
    token VARCHAR(255) NOT NULL UNIQUE,  
    expires_at TIMESTAMP NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (user_id) REFERENCES users(id) ON  
DELETE CASCADE  
);
```

## 2. کلاس RememberMe.php:

```
<?php  
  
class RememberMe {  
    private $db;  
    private $cookieName = 'remember_me';  
    private $tokenLifetime = 2592000; // 30 روز  
  
    public function __construct(Database $db) {  
        $this->db = $db;  
    }  
  
    public function createRememberToken($userId) {  
        // تولید توکن امن  
        $token = $this->generateSecureToken();  
        $hashedToken = password_hash($token,  
PASSWORD_DEFAULT);  
        $expiresAt = date('Y-m-d H:i:s', time() +  
$this->tokenLifetime);  
  
        // ذخیره در دیتابیس  
        $this->db->query("  
            INSERT INTO remember_tokens (user_id,  
token, expires_at)  
            VALUES (:user_id, :token, :expires_at)  
            ", [
```

```

        ':user_id' => $userId,
        ':token' => $hashedToken,
        ':expires_at' => $expiresAt
    ]);

    // تنظیم کوکی
    $cookieValue = $userId . ':' . $token;
    setcookie($this->cookieName, $cookieValue,
time() + $this->tokenLifetime,
        '/', '', true, true); // secure 9
httponly

    return $token;
}

public function checkRememberToken() {
    if (!isset($_COOKIE[$this->cookieName])) {
        return false;
    }

    $cookieData = explode(':', $_COOKIE[$this->cookieName], 2);
    if (count($cookieData) !== 2) {
        return false;
    }

    [$userId, $token] = $cookieData;

    // یافتن توکن در دیتابیس
    $dbToken = $this->db->query("
        SELECT * FROM remember_tokens
        WHERE user_id = :user_id AND expires_at >
NOW()
        ", [':user_id' => $userId])->find();

    if (!$dbToken || !password_verify($token,
    $dbToken['token'])) {
        $this->clearRememberToken();
        return false;
    }
}

```

```

    }

    // ورود خودکار کاربر
    $user = $this->db->query("SELECT * FROM users
WHERE id = :id",
                                [':id' => $userId])-
>find();

    if ($user) {
        session_regenerate_id(true);
        $_SESSION['user'] = [
            'id' => $user['id'],
            'email' => $user['email'],
            'login_time' => time(),
            'remembered' => true
        ];

        return true;
    }

    return false;
}

public function clearRememberToken($userId = null)
{
    // حذف کوکی
    setcookie($this->cookieName, '', time() - 3600,
    '/', '', true, true);

    // حذف از دیتابیس
    if ($userId) {
        $this->db->query("DELETE FROM
remember_tokens WHERE user_id = :user_id",
                                [':user_id' => $userId]);
    }
}

public function clearAllRememberTokens($userId) {

```

```

        $this->db->query("DELETE FROM remember_tokens
WHERE user_id = :user_id",
                        [':user_id' => $userId]);
        $this->clearRememberToken();
    }

    private function generateSecureToken() {
        return bin2hex(random_bytes(32));
    }
}

```

### 3. استفاده در: Login

```

// در فرم login
if (isset($_POST['remember_me']) &&
$_POST['remember_me'] === '1') {
    $rememberMe = new RememberMe($db);
    $rememberMe->createRememberToken($user['id']);
}

```

### 4. بررسی در هر درخواست:

```

// در ابتدای هر صفحه
session_start();

if (!isset($_SESSION['user'])) {
    $rememberMe = new RememberMe($db);
    $rememberMe->checkRememberToken();
}

```

### 5. Logout از همه دستگاهها:

```

// در صفحه logout
$rememberMe = new RememberMe($db);
if (isset($_SESSION['user']['id'])) {
    $rememberMe->clearAllRememberTokens($_SESSION['user']['id']);
}

```



// پاک کردن session

```
session_destroy();  
header("Location: /login");  
exit();
```

---

### خلاصه امتیازدهی:

- (امتیاز 2) آسان: سوالات ۱، ۷
- (امتیاز 3) متوسط: سوالات ۲، ۳، ۶، ۹
- (امتیاز 4) سخت: سوالات ۴، ۵، ۸
- (امتیاز 5) خیلی سخت: سوال ۱۰

مجموع امتیاز 32: امتیاز

۱. چرا برای نمایش فرم ثبت نام یک view جدا می سازیم؟

پاسخ:

برای جدا کردن لایه نمایش از منطق برنامه (MVC) این کار باعث خوانایی، قابلیت مدیریت و توسعه بهتر پروژه می شود و اجزای هر بخش (کنترلر و ویو) جداگانه قابل تغییر هستند.

---

۲. نقش SESSION در ثبت نام کاربر چیست؟

پاسخ:

SESSION این امکان را می دهد تا پس از ثبت نام یا ورود موفق، کاربر در کل سایت شناسایی و اطلاعات او به طور امن در سرور ذخیره شود تا وضعیت ورودش حفظ شود.

---

۳. چرا باید فیلد ایمیل و پسورد را اعتبارسنجی (validation) کنیم؟

پاسخ:

واتربخشی و امنیت! اگر داده‌ها بررسی نشود، داده اشتباه یا مخرب وارد دیتابیس می‌شود و میتواند باگ منطقی یا آسیب امنیتی ایجاد کند.

---

۴. یک کد ساده PHP بنویسید تا تأیید کند ایمیل معتبر است یا نه.

پاسخ:

php

```
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {  
    echo "ایمیل معتبر نیست";  
}
```

---

۵. چگونه جلوی ثبت‌نام با ایمیل تکراری را می‌گیریم؟

پاسخ:

قبل از درج کاربر جدید، باید یک کوئری بررسی کنید که ایمیل در جدول users وجود دارد یا نه؛ اگر وجود داشت ثبت‌نام متوقف و معمولاً به کاربر پیام مناسب داده یا ریدایرکت می‌شود.

---

۶. اگر اعتبارسنجی ورودی‌ها شکست خورد، چه باید کرد؟

پاسخ:

آرایه‌ای از خطاها ذخیره می‌کنیم، همان view فرم ثبت‌نام را مجدد نمایش می‌دهیم و آرایه خطا را به ویو پاس می‌دهیم تا پیام خطا نمایش داده شود.

---

۷. ثبت کاربر جدید با پسورد ساده چه مشکلی دارد؟

پاسخ:

اگر پسورد را به طور ساده (plain text) ذخیره کنیم، امنیت پایین می‌آید و در صورت لو رفتن دیتابیس، اطلاعات همه کاربران افشا می‌شود. باید پسورد را هش کرد. (hash)

---

۸. کد شرطی که نمایش عکس پروفایل یا لینک ثبت‌نام را در nav.php می‌سازد چیست؟

پاسخ:

php

```
<?php if($_SESSION['user'] ?? false): ?>
  
<?php else: ?>
  <a href='/register'>Register</a>
<?php endif; ?>
```

---

۹. برای اینکه بعد از ثبت‌نام، کاربر به صورت خودکار وارد سایت شود چه می‌کنیم؟

پاسخ:

پس از درج در دیتابیس، مقدار ایمیل (یا اطلاعات کاربر) را در SESSION ذخیره می‌کنیم، مثلاً `$_SESSION['user'] = ['email'=>$email]` و سپس Redirect به صفحه اصلی انجام می‌شود.

---

۱۰. یکی از اقدامات حرفه‌ای برای امنیت بیشتر Session چیست؟

پاسخ:

پس از login/ ثبت‌نام موفق، بهتر است session\_id را بازتولید کنیم (`session_regenerate_id()`) تا جلوی حمله session fixation گرفته شود و زمان انقضا منطقی هم برای session تعیین کنیم.

---

1.

**سوال:** دستور زیر در PHP چه کاری انجام می‌دهد؟

```
php -S localhost:8888 -t public
```

**جواب:** یک وب‌سرور داخلی PHP را روی پورت ۸۸۸۸ اجرا می‌کند و پوشه‌ی public را به عنوان ریشه (Document Root) در نظر می‌گیرد.

---

2.

**تمرین:** اگر فایل index.php در ریشه‌ی پروژه باشد ولی سرور روی public اجرا شود، چه اتفاقی می‌افتد؟

**جواب:** مرورگر آن فایل را پیدا نمی‌کند (۴۰۴) چون ریشه‌ی اصلی شده. پس باید فایل index.php را به پوشه public منتقل کنیم یا داخل public/index.php قرار دهیم.

---

3.

**سوال:** ثابت \_\_DIR\_\_ در PHP چه کاری انجام می‌دهد؟

**جواب:** مسیر دایرکتوری فایل جاری را برمی‌گرداند. مثلاً اگر فایل در /var/www/laracast-php/public/index.php باشد، مقدار \_\_DIR\_\_ برابر /var/www/laracast-php/public است.

---

4.

**تمرین:** اگر در فایل index.php کدی مثل زیر بنویسیم:

```
require __DIR__ . '/../functions.php';
```

این چه فایلی را لود می‌کند؟

**جواب:** فایل `functions.php` که یک سطح بالاتر از پوشه `public` قرار دارد.

---

5.

**سوال:** چرا معمولاً فایل‌های اصلی پروژه (مثل کنترلرها، `view`ها، `config` را در پوشه `public` قرار نمی‌دهیم؟

**جواب:** چون `public` مستقیم در دسترس مرورگر است. اگر فایل‌های حساس داخل آن باشند، کاربران می‌توانند به آن‌ها دسترسی پیدا کنند.

---

6.

**تمرین:** یک فایل به نام `routes.php` بساز و آن را داخل پروژه قرار بده و از `index.php` طوری ایمپورت کن که حتی اگر پروژه را در جای دیگری کپی کنی، باز هم درست کار کند.

**جواب:**

```
// public/index.php
require __DIR__ . '/../routes.php';
```

---

7.

**سوال:** اگر در مسیر اشتباه یک فایل را `require` کنیم چه خطایی رخ می‌دهد؟

**جواب:** خطای **Fatal error: Failed opening required** نمایش داده می‌شود و اجرای اسکریپت متوقف می‌شود.

---

8.

**تمرین:** فرض کن داخل پوشه‌ی `core` یک فایل به نام `bootstrap.php` داری. چطور باید آن را داخل `index.php` ایمپورت کنی؟

**جواب:**

```
require __DIR__ . '/../core/bootstrap.php';
```

---

9.

**سوال:** تفاوت `__DIR__` و `getcwd()` چیست؟

**جواب:**

- `__DIR__` مسیر پوشه‌ی فایل را برمی‌گرداند که داخلش قرار دارد.
- `getcwd()` مسیر پوشه‌ی جاری (Current Working Directory) را برمی‌گرداند، که ممکن است با محل فایل فرق کند.

---

10.

**تمرین:** یک تابع بنویس که مسیر کامل فایل‌های داخل پروژه را بر اساس `__DIR__` بسازد، به طوری که همیشه بدون مشکل کار کند.

**جواب:**

```
function base_path($path) {  
    return __DIR__ . '/../' . $path;  
}  
  
// استفاده :  
require base_path('core/bootstrap.php');
```

**تمرین ۱: اعتبارسنجی ایمیل و نمایش خطا**

**صورت تمرین:**

کدی بنویس که یک مقدار ایمیل را گرفته و اگر معتبر نبود، پیام خطا نمایش دهد.

پاسخ:

php

```
$email = "test@site"; // مقدار تستی
```

```
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {  
    echo "ایمیل نامعتبر است";  
} else {  
    echo "ایمیل معتبر است";  
}
```

---

تمرین ۲: بررسی تکراری نبودن ایمیل در آرایه کاربران

صورت تمرین:

فرض کنید لیستی از ایمیل کاربران داریم. کدی بنویس که بررسی کند آیا ایمیل وارد شده قبلاً ثبت شده یا نه.

پاسخ:

php

```
$users = ["a@test.com", "b@test.com"];  
$newEmail = "a@test.com";
```

```
if (in_array($newEmail, $users)) {  
    echo "این ایمیل قبلاً ثبت شده";  
} else {  
    echo "ایمیل جدید است";  
}
```

---

تمرین ۳: درج کاربر جدید در دیتابیس فرضی (آرایه)

صورت تمرین:

اگر ایمیل تکراری نبود، نام و ایمیل کاربر جدید را به آرایه اضافه کن.

پاسخ:

php

```
$users = [{"name"=>"Ali", "email"=>"a@test.com"}];  
$newUser = [{"name"=>"Sara", "email"=>"sara@test.com"}];
```

```
$exist = false;  
foreach ($users as $user) {  
    if ($user["email"] === $newUser["email"]) {  
        $exist = true;  
        break;  
    }  
}
```

```
if (!$exist) {  
    $users[] = $newUser;  
    echo "کاربر اضافه شد";  
} else {  
    echo "این ایمیل قبلاً ثبت شده";  
}
```

---

تمرین ۴: استفاده از SESSION برای ذخیره وضعیت ورود

صورت تمرین:

کدی بنویس که اگر کاربر با موفقیت ثبت نام کرد، اطلاعات او را در SESSION ذخیره کند.

پاسخ:

php

```
session_start();
```

```
$_SESSION['user'] = []
```



```
"email" => "sara@test.com",  
"name" => "Sara"  
];
```

```
echo $_SESSION['user']['name'];
```

---

**تمرین ۵: نمایش نام کاربر در نوبری فقط در صورت ورود**

**صورت تمرین:**

فرض کنید SESSION کاربر ست شده است. کدی بنویس که اگر کاربر لاگین بود نام او را نمایش دهد، وگرنه دکمه ورود یا ثبت نام نمایش دهد.

**پاسخ:**

php

```
session_start();  
if (isset($_SESSION['user'])) {  
    echo $_SESSION['user']['name'] . "، سلام";  
} else {  
    echo "<a href='/register'>ثبت نام</a> | <a href='/login'>ورود</a>";  
}
```

---

هر تمرین نمایندگی یکی از بخش‌ها یا مفاهیم اصلی این جلسه را دارد و کدها کاملاً عملی، تست شده و قابل گسترش هستند.