

THỰC HÀNH CÔNG NGHỆ WEB

YÊU CẦU CHUNG:

- **Sinh viên tạo một báo cáo duy nhất cho TẤT CẢ các buổi thực hành (4 buổi).** Sau mỗi buổi thực hành sinh viên nộp báo cáo (chứa bài làm của buổi hiện thời và các buổi trước đó) dưới dạng tập tin PDF xem như báo cáo tiến độ, không tính điểm. **Điểm thực hành sẽ là điểm của bài báo cáo PDF sinh viên nộp trong buổi thực hành cuối cùng (chứa tất cả bài làm của tất cả các buổi).** Không nộp bài báo cáo tiến độ đúng hạn buổi nào sẽ không tính điểm buổi đó.
- Trong báo cáo ghi thông tin MSSV, Họ tên, Nhóm học phần.
- Các sinh viên bị phát hiện sao chép bài làm của nhau *sẽ bị điểm 0 toàn bộ phần thực hành.*
- Điểm tối đa cho mỗi buổi thực hành là 10 điểm (tương đương 1 điểm của học phần).

BUỔI 4: LẬP TRÌNH WEB ĐỘNG VỚI PHP VÀ MYSQL

Nội dung thực hành

- Sử dụng phong cách lập trình HDT và chuẩn tự động nạp lớp PSR-4 để xây dựng script PHP.
- Sử dụng giao diện PDO để truy xuất (đọc, thêm, sửa, xóa) dữ liệu trong CSDL MySQL.

Báo cáo của buổi 4 là các hình ảnh minh họa kết quả thực thi các script PHP hoặc các chức năng được yêu cầu cài đặt. Mỗi chức năng cần phải có đầy đủ các hình minh họa, ví dụ:

- Giao diện trang khi đang sử dụng chức năng
- Các giao diện các trang thông báo thành công, thông báo lỗi (nếu có)
- Giao diện cho thấy kết quả sau thực hiện chức năng thành công, ...

Cấu hình MySQL và Apache

Chú ý: Các chỉ dẫn dưới đây giả định rằng XAMPP được cài đặt tại thư mục *C:/xampp*, sinh viên tự thực hiện những điều chỉnh thích hợp nếu XAMPP được cài đặt ở vị trí khác.

1. Dùng một MySQL client (phpMyadmin, HeidiSQL, ...) tạo CSDL *ct275_lab4* và tạo một bảng dữ liệu tên *contacts* trong *ct275_lab4* với các trường thông tin sau: *id*, *name*, *phone*, *notes*, *created_at*, *updated_at*.

```
CREATE TABLE `contacts` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  `phone` varchar(15) NOT NULL,  
  `notes` varchar(255) NOT NULL,  
  `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB;
```

Chèn dữ liệu ví dụ vào bảng vừa tạo:

```
INSERT INTO `contacts`(`name`, `phone`, `notes`) VALUES
('Bùi Võ Quốc Bảo', '0962256986', 'Đại học Cần Thơ'),
('Nguyễn Văn Kham', '0973856968', 'Cao đẳng Cần Thơ');
```

2. Tạo thư mục `C:/xampp/apps/lab4` và sao chép **nội dung** thư mục `lab4/start` đã cho vào thư mục này. (Sinh viên có thể sử dụng thư mục khác. Trong trường hợp đó, khi thực hiện theo các chỉ dẫn bên dưới thì cần chú ý điều chỉnh vị trí thư mục cho thích hợp).

3. Cấu hình host ảo cho thư mục `C:/xampp/apps/lab4/public`:

3.1 Hiệu chỉnh tập tin `C:/xampp/apache/conf/extra/httpd-vhosts.conf` (có thể sử dụng nội dung file `vhost.txt` đính kèm):

```
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs"
    ServerName localhost
    # Set access permission
    <Directory "C:/xampp/htdocs">
        Options Indexes FollowSymLinks Includes ExecCGI
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "C:/xampp/apps/lab4/public"
    ServerName ct275-lab4.localhost
    # Set access permission
    <Directory "C:/xampp/apps/lab4/public">
        Options Indexes FollowSymLinks Includes ExecCGI
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

3.2 Restart lại server Apache.

4. Mở trình duyệt bất kỳ, kiểm tra đường dẫn <http://ct275-lab4.localhost/> hoạt động. **TỪ ĐÂY, KHI CHỤP HÌNH MINH HỌA CẦN THỂ HIỆN ĐƯỜNG DẪN URL TRÊN THANH ĐỊA CHỈ.**

Tìm hiểu mã nguồn dự án

Cấu trúc mã nguồn đã cho:

```
lab4/
+---partials
+----footer.php      // Định nghĩa footer chung cho các trang web
+----navbar.php      // Định nghĩa thanh điều hướng chung cho các trang web
+---public
+----css              // Thư mục chứa css cho các trang web
+----fonts            // Thư mục chứa fonts cho các trang web
+----js               // Thư mục chứa js cho các trang web
+----add.php          // Trang thêm liên hệ
+----edit.php         // Trang hiệu chỉnh liên hệ
+----index.php        // Trang chủ, hiển thị các liên hệ
```

```
+---src
  +---Contact.php           // Định nghĩa lớp Contact
  +---library.php           // Định nghĩa các hàm trợ giúp
+---vendor
  +---Psr4AutoloaderClass.php // Lớp cài đặt chuẩn PSR-4
+---bootstrap.php           // Định nghĩa các biến môi trường, nạp các phụ thuộc, ...
```

Script *bootstrap.php* có nội dung như sau:

```
<?php

define('BASE_URL_PATH', '/');

require_once __DIR__ . '/src/library.php';
```

Hàm *define(name, value)* dùng để định nghĩa hằng số. Hằng số **BASE_URL_PATH** lưu đường dẫn cơ sở từ Document Root (/) đến các tập tin trong thư mục *public*. Nếu thư mục *public* là Document Root như cấu hình host ảo ở trên thì gán **BASE_URL_PATH= '/'**. Trường hợp *lab4/public* là thư mục con trong Document Root (ví dụ *C:/xampp/htdocs/lab4/public*) thì gán **BASE_URL_PATH= '/lab4/public'**. Ta định nghĩa **BASE_URL_PATH** để nếu có di chuyển vị trí thư mục dự án thì dễ dàng cập nhật lại các đường dẫn URL đến các trang, các tài nguyên tĩnh (css, javascript, ...).

Yêu cầu cài đặt

Sau khi tìm hiểu mã nguồn đã cho, hãy cài đặt các yêu cầu sau đây:

- Dùng lớp *Psr4AutoloaderClass* (*vendor/Psr4Autoloader.php*) để tùy biến tự động nạp lớp theo chuẩn PSR-4 với tiền tố (prefix) không gian tên là CT275\Labs và thư mục cơ sở (base directory) là thư mục *src*.
- Dùng thư viện PDO, hiệu chỉnh lớp Contact (*src/Contact.php*) thêm các phương thức sau đây:
 - **all()**: lấy về tất cả các liên hệ (contact)
 - **find(\$id)**: lấy về liên hệ với một id cụ thể
 - **save()**: lưu liên hệ
 - **update(array \$data)**: cập nhật liên hệ với mảng dữ liệu data
 - **delete()**: xóa liên hệ
- Dùng lớp Contact đã định nghĩa, hiệu chỉnh hoặc thêm mới các script cần thiết để cài đặt các chức năng thêm, sửa và xóa các liên hệ cho ứng dụng web.

Hướng dẫn cài đặt

(Sinh viên có thể có các cài đặt khác với hướng dẫn bên dưới, chỉ cần đảm bảo đúng yêu cầu).

Sử dụng chuẩn autoload PSR-4 và tạo lớp sinh kết nối đến CSDL

1. Sử dụng lớp *Psr4AutoloaderClass* (được lấy từ [GitHub](#)) để tùy biến tự động nạp lớp theo chuẩn PSR-4. Hiệu chỉnh tập tin *bootstrap.php* như sau:

```
<?php

define('BASE_URL_PATH', '/');

require_once __DIR__ . '/src/library.php';
require_once __DIR__ . '/vendor/Psr4AutoloaderClass.php';

$loader = new Psr4AutoloaderClass;
$loader->register();

// Các lớp có không gian tên bắt đầu với CT275\Labs nằm ở src
$loader->addNamespace('CT275\Labs', __DIR__ . '/src');
```

2. Tiếp tục, hiệu chỉnh tập tin *lab4/bootstrap.php* như sau (chú ý **thay đổi các thông số kết nối MySQL** cho phù hợp với môi trường của bạn. **Trong các phòng thực hành của khoa, tài khoản kết nối đến MySQL là người dùng root, mật khẩu trống**):

```
<?php

define('BASE_URL_PATH', '/');

...

try {
    $PDO = (new CT275\Labs\PDOFactory)->create([
        'dbhost' => 'localhost',
        'dbname' => 'ct275_lab4',
        'dbuser' => 'root',
        'dbpass' => 'root'
    ]);
} catch (Exception $ex) {
    echo 'Không thể kết nối đến MySQL,  
kiểm tra lại username/password đến MySQL.<br>';
    exit("<pre>{$ex}</pre>");
}
```

3. Trong thư mục *lab4/src*, tạo tập tin *PDOFactory.php* có nội dung như sau:

```
<?php

namespace CT275\Labs;

use PDO;

class PDOFactory
{
    public function create(array $config)
    {
        [
            'dbhost' => $dbhost,
            'dbname' => $dbname,
            'dbuser' => $dbuser,
            'dbpass' => $dbpass
        ] = $config;
```

```

        $dsn = "mysql:host={$dbhost};dbname={$dbname};charset=utf8mb4";
        $options = [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION];
        return new PDO($dsn, $dbuser, $dbpass, $options);
    }
}

```

Lớp *PDOFactory* định nghĩa phương thức *create(config)* giúp tạo một đối tượng PDO từ một mảng chứa các thông tin cần thiết để thiết lập kết nối đến CSDL. Trong script *bootstrap.php*, chúng ta tạo một đối tượng *PDOFactory* và gọi phương thức *create(config)* như đã trình bày ở trên.

Liệt kê các contact

1. Trong lớp *Contact* (*lab4/src/Contact.php*), thêm vào các hàm sau:

```

public function all()
{
    $contacts = [];

    $stmt = $this->db->prepare('select * from contacts');
    $stmt->execute();
    while ($row = $stmt->fetch()) {
        $contact = new Contact($this->db);
        $contact->fillFromDB($row);
        $contacts[] = $contact;
    }

    return $contacts;
}

protected function fillFromDB(array $row)
{
    [
        'id' => $this->id,
        'name' => $this->name,
        'phone' => $this->phone,
        'notes' => $this->notes,
        'created_at' => $this->created_at,
        'updated_at' => $this->updated_at
    ] = $row;
    return $this;
}

```

2. Hiệu chỉnh script *index.php* (*lab4/public/index.php*):

```

<?php
require_once '../bootstrap.php';

use CT275\Labs\Contact;

$contact = new Contact($PDO);
$contacts = $contact->all();
?>
<!DOCTYPE html>
<html lang="en">

```

```

<head>
...
<tbody>
  <?php foreach($contacts as $contact): ?>
    <tr>
      <td><?=htmlspecialchars($contact->name)?></td>
      <td><?=htmlspecialchars($contact->phone)?></td>
      <td><?=date("d-m-Y", strtotime($contact->created_at))?></td>
      <td><?=htmlspecialchars($contact->notes)?></td>
      <td>
        <a href="<?=BASE_URL_PATH . 'edit.php'?>"
          class="btn btn-xs btn-warning">
          <i alt="Edit" class="fa fa-pencil"> Edit</i></a>
        <a href="#" class="btn btn-xs btn-danger">
          <i alt="Delete" class="fa fa-trash"> Delete</i></a>
      </td>
    </tr>
  </tr>
  <?php endforeach ?>
</tbody>
...

```

3. Mở trình duyệt bất kì, truy cập đến <http://ct275-lab4.localhost/> để kiểm tra kết quả.

Thêm mới contact

1. Thêm hàm save vào lớp Contact như sau:

```

public function save()
{
    $result = false;

    if ($this->id >= 0) {
        $stmt = $this->db->prepare('update contacts set name = :name,
            phone = :phone, notes = :notes, updated_at = now()
            where id = :id');
        $result = $stmt->execute([
            'name' => $this->name,
            'phone' => $this->phone,
            'notes' => $this->notes,
            'id' => $this->id]);
    } else {
        $stmt = $this->db->prepare(
            'insert into contacts (name, phone, notes, created_at, updated_at)
            values (:name, :phone, :notes, now(), now())');
        $result = $stmt->execute([
            'name' => $this->name,
            'phone' => $this->phone,
            'notes' => $this->notes]);
        if ($result) {
            $this->id = $this->db->lastInsertId();
        }
    }

    return $result;
}

```

2. Thêm đoạn mã xử lý thêm contact vào script *add.php* (*lab4/public/add.php*):

```
<?php
require_once '../bootstrap.php';

use CT275\Labs\Contact;

$errors = [];

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $contact = new Contact($PDO);
    $contact->fill($_POST);
    if ($contact->validate()) {
        $contact->save() && redirect(BASE_URL_PATH);
    }

    $errors = $contact->getValidationErrors();
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
...
```

Hàm *redirect(url)* được định nghĩa trong *src/library.php* giúp chuyển hướng sang một trang khác. Trong khối lệnh trên, *redirect(BASE_URL_PATH)* sẽ thực hiện chuyển hướng trang về trang chủ.

3. Mở trình duyệt bất kì, truy cập đến <http://ct275-lab4.localhost/add.php> và thử thêm một contact để kiểm tra.

Hiệu chỉnh contact

1. Thêm hàm *find* và *update* vào lớp *Contact* như sau:

```
public function find($id)
{
    $stmt = $this->db->prepare('select * from contacts where id = :id');
    $stmt->execute(['id' => $id]);

    if ($row = $stmt->fetch()) {
        $this->fillFromDB($row);
        return $this;
    }

    return null;
}

public function update(array $data)
{
    $this->fill($data);
    if ($this->validate()) {
        return $this->save();
    }
    return false;
}
```

2. Hiệu chỉnh script *edit.php* (*lab4/public/edit.php*):

```
<?php
require_once '../bootstrap.php';

use CT275\Labs\Contact;

$contact = new Contact($PDO);

$id = isset($_REQUEST['id']) ?
    filter_var($_REQUEST['id'], FILTER_SANITIZE_NUMBER_INT) : -1;

if ($id < 0 || !($contact->find($id))) {
    redirect(BASE_URL_PATH);
}

$errors = [];
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if ($contact->update($_POST)) {
        // Cập nhật dữ liệu thành công
        redirect(BASE_URL_PATH);
    }
    // Cập nhật dữ liệu không thành công
    $errors = $contact->getValidationErrors();
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
...
```

Mảng **\$_REQUEST** là một biến toàn cục (superglobal) được định nghĩa sẵn (tương tự \$_GET, \$_POST, \$_SERVER,...) chứa nội dung tổng hợp từ 3 mảng khác là \$_GET, \$_POST và \$_COOKIE.

3. Trong script *index.php*, thêm chuỗi truy vấn vào URL đến script *edit.php* (thuộc tính href):

```
...
<tbody>
<?php foreach($contacts as $contact): ?>
    <tr>
        ...
        <td>
            <a href="<?=BASE_URL_PATH . 'edit.php?id=' . $contact->getId()?>"
                class="btn btn-xs btn-warning">
                <i alt="Edit" class="fa fa-pencil"> Edit</i></a>
            <a href="#" class="btn btn-xs btn-danger">
                <i alt="Delete" class="fa fa-trash"> Delete</i></a>
        </td>
    </tr>
<?php endforeach ?>
</tbody>
...
```

4. Mở trình duyệt web bất kì và kiểm tra cập nhật contact.

Xóa contact

1. Thêm hàm delete vào lớp Contact:

```
public function delete()
{
    $stmt = $this->db->prepare('delete from contacts where id = :id');
    return $stmt->execute(['id' => $this->id]);
}
```

2. Trong thư mục *lab4/public*, tạo script *delete.php*:

```
<?php
require_once '../bootstrap.php';

use CT275\Labs\Contact;

$contact = new Contact($PDO);

if (
    $_SERVER['REQUEST_METHOD'] === 'POST'
    && isset($_POST['id'])
    && ($contact->find($_POST['id'])) !== null
) {
    $contact->delete();
}

redirect(BASE_URL_PATH);
```

3. Hiệu chỉnh script *index.php*:

```
...
<tbody>
    <?php foreach($contacts as $contact): ?>
        <tr>
            ...
            <td>
                <a href="<?=BASE_URL_PATH . 'edit.php?id=' . $contact->getId()?>"
                    class="btn btn-xs btn-warning">
                    <i alt="Edit" class="fa fa-pencil"> Edit</i></a>
                <form class="delete" action="<?=BASE_URL_PATH . 'delete.php'?>"
                    method="POST" style="display: inline;">
                    <input type="hidden" name="id"
                        value="<?=$contact->getId()?>">
                    <button type="submit" class="btn btn-xs btn-danger"
                        name="delete-contact">
                        <i alt="Delete" class="fa fa-trash"> Delete</i></button>
                </form>
            </td>
        </tr>
    <?php endforeach ?>
</tbody>
...
<div id="delete-confirm" class="modal fade" role="dialog">"
```

```

<div class="modal-dialog">
  <div class="modal-content">
    <div class="modal-header">
      <button type="button" class="close"
        data-dismiss="modal">&times;</button>
      <h4 class="modal-title">Confirmation</h4>
    </div>
    <div class="modal-body">Do you want to delete this contact?</div>
    <div class="modal-footer">
      <button type="button" data-dismiss="modal"
        class="btn btn-danger" id="delete">Delete</button>
      <button type="button" data-dismiss="modal"
        class="btn btn-default">Cancel</button>
    </div>
  </div>
</div>
</div>
</div>

<?php include('../partials/footer.php') ?>
...
<script>
  $(document).ready(function(){
    new WOW().init();
    $('#contacts').DataTable();

    $('#button[name="delete-contact"]').on('click', function(e){
      e.preventDefault();

      const form = $(this).closest('form');
      const nameTd = $(this).closest('tr').find('td:first');
      if (nameTd.length > 0) {
        $('#modal-body').html(
          `Do you want to delete "${nameTd.text()}"?`
        );
      }
      $('#delete-confirm').modal({
        backdrop: 'static', keyboard: false
      })
      .one('click', '#delete', function() {
        form.trigger('submit');
      });
    });
  });
</script>

```

4. Mở trình duyệt web bất kì và kiểm tra xóa contact.