|  |  |  |
| --- | --- | --- |
| **1** | **1. Tổng quan ứng dụng Web PHP** | |
| 1 | Website là gì? | • Website là tập hợp các trang tài liệu và tài nguyên có liên quan đến nhau, thường được truy cập thông qua một tên miền  • Các trang tài liệu có thể chứa văn bản, hình ảnh, âm thanh, video... • Mỗi website có địa chỉ cụ thể trên internet gọi là URL  • Website được lưu trữ trên một hoặc nhiều máy chủ (Server)  • Website được quản lý bởi cá nhân, công ty hoặc một tổ chức  • Có hai loại website cơ bản là web động (dynamic web) và web tính (static web) |
| 2 | Mô tả hoạt động của Web | Người dùng gửi yêu cầu -> máy chủ; máy chủ->phản hồi về máy người dùng |
| 3 | So sánh website tĩnh và động | website tĩnh: nội dung không thay đổi  website động: nội dung có thể thay đổi |
| 4 | Mô hình client server | Người dùng gửi http request->sever; sever gửi htt presponse-> người dùng; |
| 5 | HTTP là gì | • HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản |
| 6 | Các đặc điểm cơ bản của HTTP | * Là giao thức (protocol) cho phép máy chủ web (web server) và máy khách (trình duyệt, ứng dụng) trao đổi dữ liệu qua mạng máy tính. * Là giao thức request (yêu cầu gửi từ client lên server), response (phản hồi của server về client). * Sử dụng các kết nối tin cậy (TCP - Transmission Control Protocol), mặc định trên cổng 80. * Là giao thức không lưu giữ trạng thái (Stateless) nghĩa là mỗi yêu cầu được coi là một yêu cầu mới. Nói cách khác, máy chủ mặc định không nhận dạng được máy khách\client. |
| 7 | Các tính năng cơ bản của HTTP | * HTTP không bị phụ thuộc bởi thông điệp: Bất kỳ dạng dữ liệu nào cũng có thể được gửi qua HTTP miễn là máy chủ và máy khách biết cách kiểm soát nội dung thông điệp. * HTTP không duy trì kết nối liên tục: Giả sử máy khách là một trình duyệt, thì sau khi tạo ra và gửi đi một yêu cầu (request), trình duyệt sẽ ngắt kết nối và đợi phản hồi từ phía máy chủ. Hay một cách hiểu khác là Server và Client giao tiếp với nhau qua nhiều bước ngắt/mở kết nối: Một trình duyệt khởi tạo một yêu cầu HTTP và sau đó một yêu cầu được tạo ra, Client ngắt kết nối từ Server và đợi cho một phản hồi. Server xử lý yêu cầu và thiết lập lại sự kết nối với Client để gửi phản hồi trở lại. * HTTP là giao thức Stateless: Server và Client nhận biết nhau qua thông điệp được kèm theo trong yêu cầu hiện tại, bản chất giao thức không giúp chúng lưu giữ bất kỳ thông tin gì về sự nhận diện đó |
| 8 | Các phương thức HTTP request | 1.Get:  2Post:  2.1Put: được sử dụng để cập nhật dữ liệu đã có trong database.  2.2Patch: được sử dụng khi phải cập nhật một phần dữ liệu của đối tượng.  2.3Delete: xoá các dữ liệu của server về tài nguyên thông qua URI.  2.4head:  2.5Option:cấu hình cho phương thức |
| 9 | HTTP Status Code của HTTP response  Error: lỗi | • Mã trạng thái HTTP được server phản hồi lại mỗi khi nhận được http resquest  • Status-Code là một số nguyên bao gồm 3 ký tự  • Ký tự đầu tiên phân loại trạng thái  •Ký tự đầu gồm có: 1,2,3,4,5  •1xx: Thông tin  •2xx: Thành công  •3xx: Chuyển hướng  •4xx: Lỗi ở phía client  •5xx: Lỗi ở phía server |
| 10 | So sánh GET và POST | * POST: Bảo mật hơn GET vì dữ liệu được gửi ngầm, không xuất hiện trên URL * GET: Dữ liệu được gửi tường minh, chúng ta có thể nhìn thấy trên URL, đây là lý do khiến nó không bảo mật so với POST. * GET thực thi nhanh hơn POST vì những dữ liệu gủi đi luôn được webbrowser cached lại. |
| 11 | Các kiểu dữ liệu của PHP | * String * Integer * Float (floating point numbers - also called double) * Boolean * Array * Object * NULL * Resource |
| 12 | Khai báo biến chứa các kiểu dữ liệu trong PHP | $VariableName; |
| 13 | Khai báo hằng số trong PHP | C1:define(*name*, *value*, *case-insensitive*)  C2: |
| 14 | Giải thích echo, print, print\_r, var\_dump | * In mảng object dùng var\_dump hoặc print\_r * Const variableName=..; |
| 15 | Các Biến toàn cục trong PHP và ý nghĩa của các biến | $\_GLOBAL  $\_SEVER  $\_REQUEST  $\_POST  $\_GET |
| 16 | Các hàm thao tác với chuỗi ( Liệt kê và giải thích ) | strlen()trả về độ dài của một chuỗi.  str\_word\_count()đếm số từ trong một chuỗi. strrev () - Đảo ngược một chuỗistrpos () - Tìm kiếm văn bản trong chuỗistr\_replace () - Thay thế văn bản trong một chuỗi |
| 17 | Các Toán tử trong PHP |  |
| 18 | Câu lệnh if, if else, if elseif else |  |
| 19 | Câu lệnh switch |  |
| 20 | Vòng lặp while |  |
| 21 | Vòng lặp do…while |  |
| 22 | Vòng lặp for |  |
| 23 | Vòng lặp foreach | *Foreach*($nameArray *as* $*key => $value*) |
| 24 | Tìm hiểu hàm date của php | Date(“d/m/y”); |
| 25 | PHP có mấy cách khai báo? | <?php…?>  <script language=”PHP”>…</script> |
| 26 | Theo bạn, sự khác nhau của toán tử & và && trong PHP là gì? | Toán từ & và && trong PHP đều là phép toán AND, tuy nhiên toán tử một dấu & áp dụng theo kiểu bit, nói dễ hiểu hơn một dấu & là phép AND thao tác trên các bit ví dụ 1 & 0 thì ra 0. Phép toán hai dấu & thì chỉ áp dụng cho kiểu boolean True và False. |
| 27 | Hãy cho biết $a++ và ++$a khác nhau ở đâu? |  |
| 28 | $$ có nghĩa là gì | $$var (đô la kép) là một biến tham chiếu lưu trữ giá trị của biến $ bên trong nó. |
| \* | Cách nhận dữ liệu qua url  Ví dụ: index.php?id=1&name=Nguyen Van A | $x=GET||POST(‘Name’) |
| \* | Cách nhận dữ liệu qua phương thức POST |  |
| \* | Hoàn thành level 1 trong hệ thống bob |  |
| \* | Luyện tập cách xử lý dữ liệu từ form POST và GET |  |
| \* | Luyện viết thành thạo cú pháp khai báo biến, câu điều kiện, vòng lặp, switch case ( mỗi phần 10 lần ) |  |
| \* | Hoàn thành bài kiểm tra |  |
| **2** | **2. Mảng và hàm trong PHP** | |
| 1 | Mảng là gì ? |  |
| 2 | Trong PHP, có mấy kiểu mảng |  |
| 3 | Hàm là gì ? |  |
| 4 | Tham số mặc định của hàm có tác dụng gì |  |
| 5 | Các hàm của mảng ? |  |
| 6 | Các hàm thông dụng của mảng trong PHP ? |  |
| 7 | Mảng đa chiều là gì ? |  |
| 8 | Sử dụng include và require |  |
| 9 | Sử dụng include\_once và require\_once |  |
| 10 | Json là gì ? |  |
| 11 | Cách tạo ra một chuỗi json |  |
| 12 | Hàm json\_encode |  |
| 13 | Hàm json\_decode, json\_decode true |  |
| 14 | Hàm empty làm công việc gì ? |  |
| 15 | Hàm isset làm công việc gì ? |  |
| 16 | empty(null) và isset(null) |  |
| 17 | declare(strict\_types=1) là gì ? |  |
| 18 | Tìm hiểu try-catch |  |
| \* | Luyện tập tạo hàm |  |
| \* | Luyện tập tạo hàm có sự trả về |  |
| \* | Luyện tập tạo mảng Index arrays |  |
| \* | Luyện tập tạo mảng Associative arrays |  |
| \* | Luyện tập tạo mảng Multidimensional arrays |  |
| \* | Sử dụng vòng lặp thông qua một mảng index |  |
| \* | Sử dụng vòng lặp foreach với Associative Array |  |
| \* | Luyện tập các hàm sắp xếp mảng |  |
| \* | Hoàn thành level 2 hệ thống BOB |  |
| \* | Hoàn thành level 3 hệ thống BOB |  |
| **3** | **3. Lớp và đối tượng trong PHP** | |
| 1 | Trình bày khái niệm OOP?  Tại sao lại sử dụng OOP | tạo các đối tượng chứa cả dữ liệu và hàm   * OOP cung cấp một cấu trúc rõ ràng cho các chương trình * OOP giúp giữ cho mã PHP KHÔ "Không lặp lại chính bạn" và làm cho mã dễ bảo trì, sửa đổi và gỡ lỗi hơn * OOP giúp bạn có thể tạo các ứng dụng có thể tái sử dụng đầy đủ với ít mã hơn và thời gian phát triển ngắn hơn |
| 2 | Lớp là gì | Khuôn mẫu của các đối tượng |
| 3 | Đối tượng là gì | Đối tượng bao gồm các thuộc tính và phương thức |
| 4 | So sánh lớp và đối tượng |  |
| 5 | Các tính chất đặc thù của OOP | * Đóng gói * Kế thừa * Đa hình * Trừu tượng |
| 6 | Quy ước đặt tên cho phương thức setName, getName sẽ làm việc với thuộc tính nào | $name; |
| 7 | Biến $this dùng để làm gì | Đại diện cho đối tượng |
| 8 | Phương thức khởi tạo (constructor)\hàm tạo dùng để làm gì ? | Dùng để tạo và khởi tạo các trạng thái ban đầu của đối tượng  Thường dùng để truyền giá trị ngay khi khởi tạo đối tượng |
| 9 | Làm thế nào để khởi tạo đối tượng từ lớp A | $ObjName = new ClassName(); |
| 10 | Các mức truy xuất (access modifiers) | Pubplic;  Private;  protected |
| 11 | Cấu trúc sơ đồ mô tả lớp |  |
| 12 | Khởi tạo đối tượng |  |
| 13 | Vòng đời của một đối tượng |  |
| 14 | Khai báo thuộc tính |  |
| 15 | Khai báo phương thức |  |
| 16 | Cách truy xuất thuộc tính |  |
| 17 | Cách truy xuất  phương thức |  |
| \* | Tính bao gói thể hiện ntn | Là cách để che dấu những tính chất xử lý bên trong của đối tượng, những đối tượng khác không thể tác động trực tiếp làm thay đổi trạng thái  chỉ có thể tác động thông qua các method public của đối tượng đó |
| \* | Tính kế thừa thể hiện ntn | Kế thừa trong OOP = Khi một lớp dẫn xuất từ ​​một lớp khác.  Lớp con sẽ kế thừa tất cả các thuộc tính và phương thức public và protected từ lớp cha. Ngoài ra, nó có thể có các thuộc tính và phương thức riêng.  Một lớp kế thừa được xác định bằng cách sử dụng extends từ khóa. |
| \* | Tính đa hình thể hiện ntn | Các đối tượng của các lớp khác nhau có chung một hành vi nhưng khi thực thi thì sẽ đạt được kết quả khác nhau |
| \* | Tính trừu tượng thể hiện ntn | Chương trình có thể bỏ qua những đặc điểm và hành vi không cần thiết của các đối tượng, chỉ lựa chọn những đặc điểm và hành vi có liên quan đến bài toán cần giải quyết. |
| **4** |  | |
| 1 | Liệt kê access modifier? |  |
| 3 | Phân biệt this và self |  |
| 4 | Các thành phần của một lớp? |  |
| 5 | Mức truy xuất dùng để làm gì |  |
| 6 | Truy cập vào một thuộc tính/biến được khai báo là private từ bên ngoài lớp sẽ bị lỗi gì ? |  |
| 7 | Vẽ sơ đồ lớp của lớp sau: tên lớp **Customer**:  property:  **name:**string **,**  **birthday:**string;  **age**: integer  method:  **getInfo**() :string |  |
| 8 | Namespace là gì ? |  |
| 9 | Từ khóa static |  |
| 10 | Từ khóa self |  |
| 11 | Self có thể đi cùng với $this |  |
| 12 | Static property ( Thuộc tính static ) |  |
| 13 | Static method ( Phương thức static ) |  |
| **5** | **5. Inheritance** | |
| 1 | Kế thừa là gì | Lớp con kế thừa thuộc tính và phương thức có access modifier là public hoặc protected |
| 2 | Kế thừa từ lớp A tới lớp B  Extends: kế thừa. |  |
| 3 | PHP có đa kế thừa không? |  |
| 4 | Từ khóa parent làm nhiệm vụ gì | gọi một thuộc tính tĩnh từ một lớp con |
| 5 | Từ khóa final | ngăn kế thừa lớp hoặc ngăn ghi đè phương thức |
| 6 | Ghi đè phương thức là gì? | Các phương thức kế thừa có thể được ghi đè bằng cách định nghĩa lại các phương thức (sử dụng cùng một tên) trong lớp con. |
| 7 | Mở rộng |  |
| 8 | Ghi đè phương thức được sử dụng để làm gì? |  |
| **6** | **6. Abstract Class & Interface** | |
| 1 | Interface là gì ? |  |
| 2 | Phương thức trừu tượng là gì |  |
| 3 | Đặc điểm khi 1 lớp triển khai từ interface |  |
| 4 | Có thể khai báo 1 phương thức bình thường trong 1 interface được hay ko ? |  |
| 5 | Một lớp có thể triển khai từ nhiều interface hay không ? |  |
| 6 | Một interface có thể kế thừa từ những **interface** khác |  |
| 7 | Abstract Class là gì ? |  |
| 8 | Dùng từ khóa gì để khai báo phương thức trừu tượng trong lớp TT |  |
| 9 | Đặc điểm khi kế thừa 1 lớp từ abstract |  |
| 10 | Có thể khởi tạo đối tượng từ 1 lớp trừu tượng hay không |  |
| 11 | So sánh Abstract Class & Interface | |  |  | | --- | --- | | Abstract Class | Interface | |  |  | |  |  | |  |  | |  |  | |  |  | |
| 12 | Khi nào dùng Abstract Class |  |
| 13 | Khi nào dùng Interface |  |
| **7** | **7. Clean Code & Refactoring** | |
| 1 | Clean Code là gì ? |  |
| 2 | Trình bày được tầm quan trọng của Clean Code |  |
| 3 | Kể tên một số Coding Convention trong PHP |  |
| 4 | Refactor là gì? Em đã sử dụng các kỹ thuật refactor nào? |  |
| 5 | Đặt tên tốt là như thế nào  Các quy tắc đặt tên biến |  |
| 6 | Các quy tắc đặt tên lớp |  |
| 7 | Các quy tắc đặt tên phương thức |  |
| **8** | **8. DSA: Array & LinkedList** | |
| 1 | Cấu trúc dữ liệu |  |
| 2 | Các cấu trúc dữ liệu thông dụng |  |
| 3 | ArrayList |  |
| 4 | Các thao tác cơ bản của ArrayList |  |
| 5 | Cài đặt lớp ArrayList |  |
| 6 | LinkedList |  |
| 7 | Cấu trúc của LinkedList |  |
| 8 | Lớp MyLinkedList |  |
| 9 | Cài đặt lớp LinkedList |  |
| 10 | Singly Linked List |  |
| 11 | Doubly Linked List |  |
| 12 | Circular Doubly Linked List |  |
| 13 | Lựa chọn ArrayList hay LinkedList |  |
| 14 | Tóm tắt bài học |  |
| **9** | **9. DSA: Stack, Queue, Tree** | |
| 1 | Stack và cấu trúc của Stack |  |
| 2 | Các thao tác cơ bản của Stack |  |
| 3 | Cài đặt lớp Stack |  |
| 4 | Queue, cấu trúc của Queue |  |
| 5 | Các thao tác cơ bản của Queue |  |
| 6 | Cài đặt lớp Queue |  |
| 7 | Tree |  |
| 8 | Các khái niệm cơ bản về cây(tree) |  |
| 9 | Cây nhị phân - Binary Tree  Cài đặt cây nhị phân - Binary Tree |  |
| 10 | Cây nhị phân tìm kiếm - Binary Search Tree |  |
| 11 | Hoạt động cơ bản trên cây tìm kiếm nhị phân |  |
| 12 | Tóm tắt bài học |  |
| **11** | **11. Thuật toán tìm kiếm** | |
| 1 | Thuật toán tìm kiếm tuyến tính |  |
| 2 | Thuật toán tìm kiếm nhị phân |  |
| 3 | **So sánh tìm kiếm tuyến tính và tìm kiếm nhị phân** | |
|  | | |
| 4 | Tóm tắt bài học |  |
| **12** | **12. Thuật toán sắp xếp** | |
|  |  |  |
| **13** | **13. Xử lý ngoại lệ** | |
| 1 | Ngoại lệ là gì? Tại sao phải xử lý ngoại lệ? |  |
| 2 | Các kiểu lỗi thường xảy ra |  |
| 3 | Xử lý ngoại lệ trong PHP |  |
| 4 | Sử dụng khối try-catch |  |
| 5 | Khối finally  Finally: luôn thực thi |  |
| 6 | Ném ngoại lệ (throw new)  Nằm trong function vẫn thực thi, khi nằm trong try. |  |
| 7 | Ngoại lệ do người dùng tự tạo |  |
| 8 | Tóm tắt bài học |  |
| **14** | **14. String & Regex** | |
|  |  |  |
|  |  |  |
| **15** | **15. Cơ sở dữ liệu quan hệ** | |
| 1 | Khái niệm cơ sở dữ liệu |  |
| 2 | Phân biệt được cơ sở dữ liệu và file |  |
| 3 | Khái niệm cơ sở dữ liệu quan hệ |  |
| 4 | Khái niệm cơ bản trong cơ sở dữ liệu quan hệ |  |
| 5 | Hệ quản trị CSDL là gì ? |  |
| 6 | Các hệ QTCSDL mà bạn biết |  |
| 7 | Trong một CSDL gồm các thành phần gì ? |  |
| 8 | Trong một bảng trong CSDL gồm các thành phần gì ? |  |
| 9 | SQL là gì |  |
| 10 | SQL dùng để làm gì ? |  |
| 11 | MySQL là gì |  |
| 12 | Các câu lệnh làm việc với dữ liệu: |  |
| 13 | Cú pháp tạo CSDL |  |
| 14 | Cú pháp xóa CSDL |  |
| 15 | Cú pháp Tạo bảng |  |
|  | Các datatype trong bảng |  |
| 16 | Cú pháp Xóa bảng |  |
| 17 | Cú pháp SELECT |  |
| 18 | SELECT \* FROM customers có ý nghĩa gì ? |  |
| 19 | SELECT customerName, phone, city, country FROM customers có ý nghĩa gì ? |  |
| 20 | SELECT \* FROM customers WHERE customerName = 'Atelier Graphique' có ý nghĩa gì ? |  |
| 21 | SELECT \* FROM customers WHERE customername like '%A%' có ý nghĩa gì ? |  |
| 22 | SELECT \* FROM customers WHERE city IN ('Nantes',' Las Vegas',' Warszawa','NYC') có ý nghĩa gì ? |  |
| **16** | **Thiết kế và tạo CSDL** | |
| 1 | Phân tích và Thiết kế CSDL |  |
| 2 | Các bước phân tích và thiết kế CSDL Quan hệ |  |
| 3 | Tập hợp Dữ liệu, Tổ chức các bảng |  |
| 4 | Tạo mối quan hệ giữa các bảng |  |
| 5 | Tinh chỉnh và chuẩn hóa thiết kế |  |
|  | Đọc thêm | https://www.codelean.vn/2019/12/binh-thuong-hoa-la-gi-1nf-2nf-3nf-bcnf.html |
| 6 | Chưa chuẩn hóa |  |
| 7 | Thế nào là chuẩn 1NF |  |
| 8 | Thế nào là chuẩn 2NF |  |
| 9 | Thế nào là chuẩn 3NF |  |
| 10 | Phụ thuộc chức năng bắc cầu là gì? |  |
| 11 | ERD Website bán hàng |  |
| 12 | Constraint (ràng buộc) |  |
| 13 | Một số ràng buộc thông dụng |  |
| 14 | NOT NULL |  |
| 15 | UNIQUE |  |
| 16 | CHECK |  |
| 17 | DEFAULT |  |
| 18 | Primary Key (Khóa chính) |  |
| 19 | Đặc điểm của khóa chính |  |
| 20 | Tạo khóa chính phức hợp |  |
| 21 | Tạo khóa chính sau khi tạo bảng |  |
| 22 | Khóa ngoại (Foreign Key) |  |
| 23 | Đặc điểm của khóa ngoại |  |
| 24 | Khóa ngoại: Ví dụ |  |
| 25 | Các kiểu dữ liệu thông dụng |  |
| 26 | Tóm tắt bài học |  |
| **17** |  | |
| 1 | Toán tử so sánh |  |
| 2 | Toán tử logic |  |
| 3 | Sử dụng WHERE |  |
| 4 | Câu lệnh AND |  |
| 5 | Câu lệnh OR |  |
| 6 | Câu lệnh NOT |  |
| 7 | Câu lệnh GROUP BY |  |
| 8 | Câu lệnh JOIN được sử dụng để truy vấn dữ liệu kết hợp từ  nhiều bảng.  Liên kết tham chiếu từ bảng này sang khóa chính bảng khác.  Câu lệnh INNER JOIN |  |
| 9 | Cậu lệnh LEFT JOIN |  |
| 10 | Câu lệnh RIGHT JOIN |  |
| 11 | Câu lệnh FULL JOIN |  |
| 12 | Cách join 2 bảng |  |
| 13 | Cách join 3 bảng |  |
| **18** | **18. Các hàm thông dụng trong SQL :** | |
| 1 | UPPER |  |
| 2 | LOWER |  |
| 3 | SQRT |  |
| 4 | AVG |  |
| 5 | SUM |  |
| 6 | ABS |  |
| 7 | MIN / MAX |  |
| 8 | COUNT |  |
| **19** |  | |
| 1 | View trong MySQL là gì? |  |
| 2 | Ưu điểm khi sử dụng View trong MySQL |  |
| 3 | Nhược điểm khi sử dụng View trong MYSQL |  |
| 4 | Cách tạo view |  |
| 5 | Cách tạo view với điều kiện ( WITH CHECK OPTION) |  |
| 6 | Select View ( Lấy kết quả trả về ) |  |
| 7 | Update View ( cập nhật kết quả trong View ) |  |
| 8 | Quy tắc khi cập nhật VIEW |  |
| 9 | Chèn dữ liệu vào VIEW |  |
| 10 | Xóa dữ liệu khỏi VIEW |  |
| 11 | Exercise 1 |  |
| 12 | Exercise 2 |  |
| 13 | MySQL Index là gì? |  |
| 14 | Các loại index trong MySQL |  |
| 15 | Chỉ mục index có tác dụng gì |  |
| 16 | Để xem tất cả những index thuộc một table |  |
| 17 | Tạo index cho cột |  |
|  | Store Procedure trong MySQL |  |
|  | Ưu điểm của Stored Procedure trong Mysql  Store Procedure: là thủ tục lưu trữ. |  |
|  | Nhược điểm của Stored Procedure trong Mysql |  |
|  | Tạo Stored Procedure trong MySql |  |
|  | DELIMITER $$ |  |
|  | CREATE PROCEDURE procedureName() |  |
|  | BEGIN và END; $$ |  |
|  | DELIMITER ; |  |
|  | Xem tất cả các Procedure |  |
|  | Exercise 1 |  |
|  | Exercise 2 |  |
|  | Các loại tham số trong Mysql Stored Procedure |  |
|  | Tham số loại IN trong Mysql Stored Procedure |  |
|  | Exercise 1 |  |
|  | Exercise 2 |  |
|  | Tham số loại **OUT** trong Mysql Stored Procedure |  |
|  | Exercise 1 |  |
|  | Exercise 2 |  |
|  | Tham số dạng **INOUT** trong Mysql Stored Procedure |  |
|  | Exercise 1 |  |
|  | Exercise 2 |  |
|  | Sửa Stored Procedure đã tạo |  |
| **20** | **20. Database & CRUD:** | |
| 1 | CRUD |  |
| 2 | Ứng dụng quản lý cơ bản |  |
| 3 | Các câu lệnh CRUD trong SQL |  |
| 4 | Câu lệnh INSERT/INTO  Chèn/thêm dữ liệu |  |
| 5 | Câu lệnh SELECT/FROM  Chọn dữ liệu |  |
| 6 | Câu lệnh UPDATE/SET  Cập nhật |  |
| 7 | Câu lệnh DELETE  Xóa |  |
| 8 | Câu lệnh WHERE |  |
| 9 | Các toán tử trong câu lệnh WHERE |  |
| 10 | Kết nối đến CSDL |  |
| 11 | Xem kỹ | https://james.codegym.vn/mod/page/view.php?id=13559 |
| **21** | **21. MVC Model** | |
| 1 | Kiến trúc phân tầng: trình bày mô hình MVC |  |
| 2 | Kiến trúc MVC |  |
| 3 | MVC là gì |  |
| 4 | Lợi ích của kiến trúc MVC |  |
| 5 |  |  |
| **22** |  | |
| 1 | CSS là gì ? |  |
| 2 | Cách nhúng CSS vào web |  |
| 3 | Cú pháp của CSS |  |
| 4 | Độ ưu tiên nhúng CSS |  |
| 5 | Độ ưu tiên selector |  |
| **23** | **23. CSS Layout** | |
| 1 | Layout web là gì? |  |
| 2 | Template là gì? |  |
| 3 | Mô hình hộp là gì?  Kích thước của Box model? |  |
|  | Các thành phần trong mô hình hộp? |  |
| 4 | Kể tên các loại thuộc tính position? |  |
| 5 | Trình bày position 1 |  |
| 6 | Trình bày position 2 |  |
| 7 | Trình bày position 3 |  |
| 8 | Trình bày position 4 |  |
| 9 | Responsive Web Design là gì |  |
| 10 | Grid-view là gì? |  |
| **24** | **24. Bootstrap cơ bản** | |
|  | Bootstrap là gì |  |
|  | Nhúng bootstrap vào trang web |  |
|  |  |  |