**ĐÁP ÁN ÔN TẬP 2**

**Câu 1:**

1. Tổng điểm: 70 điểm
2. b. Các yêu cầu cần đạt được:

- Cơ sở dữ liệu, framework lập trình được phép tự chọn.

- Cho phép sử dụng một số hàm thư viên có sẵn để xử lý kết nối database, không cần phải lập trình các lớp xử lý kết nối.

- Không cần xây dựng các chức năng liên quan đến tài khoản như đăng nhập, đăng ký, phân quyền…

- Về giao diện phía người xem: Giao diện người xem cần dễ nhìn, có tính thẩm mỹ, hiển thị responsive tương thích tốt với điện thoại.

- Về chức năng: đáp ứng đầy đủ các tính năng phục vụ cho việc quản trị tin tức.

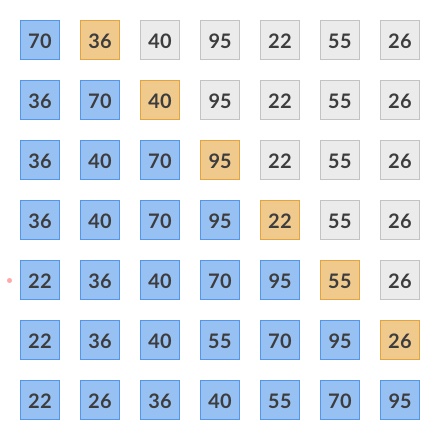
1. Thang điểm:

|  |  |
| --- | --- |
| **Yêu cầu** | **Điểm tối đa** |
| Quản lý danh sách tin tức hiển thị đủ | 10 |
| Thêm mới tin tức thành công | 10 |
| Chỉnh sửa tin tức thành công | 10 |
| Xóa tin tức thành công | 10 |
| Danh sách tin tức cho người xem được hiển thị đủ thông tin | 10 |
| Trang chi tiết tin tức cho người hiển thị hiển thị đủ thông tin | 10 |
| Giao diện cho người xem có responsive, có tính thẩm mỹ | 10 |

**Câu 2:**

1. Tổng điểm: 30 điểm.
2. Yêu cầu:

* Dữ liệu đầu vào kiểm thử: Dãy số 70 – 36 – 40 – 95 – 22 – 55 – 26.
* Dữ liệu đầu ra: Dãy số đã sắp xếp 22 – 26 – 36 – 40 – 55 – 70 – 05.
* In được các bước sắp xếp theo thứ tự từng lần hoán vị.



* Lập trình được thuật toán theo insertion sort như sau:

1. Thang điểm:

|  |  |
| --- | --- |
| **Thang điểm** | **Điểm tối đa** |
| In được kết quả đã sắp xếp cuối cùng | 10đ |
| In được danh sách tất các lần sắp xếp | 10đ |
| Thuật toán chính xác | 10đ |

void insertionSort(int a[], int array\_size) {

int i, j, last;

for (i=1; i < array\_size; i++) {

last = a[i];

j = i;

while ((j > 0) && (a[j-1] > last)) {

a[j] = a[j-1];

j = j - 1; }

a[j] = last;

} // end for

} // end of isort