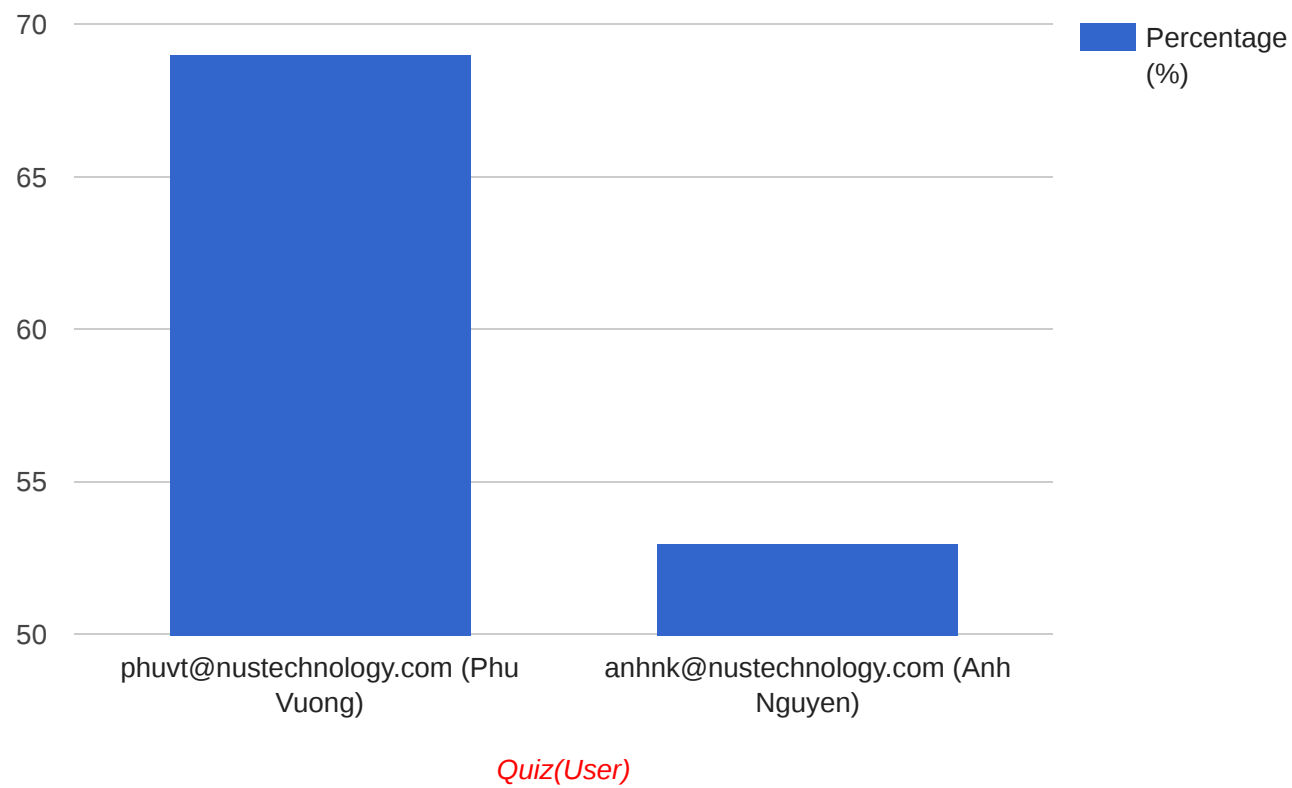


First Name	Anh
Last Name	Nguyen
Email	anhnk@nustechology.com
Quiz Name	[Training Test] Performances
Attempt Time	2018-05-11 01:01:57
Time Spent (Approx.)	30:40
Percentage	53.8462%
Percentile	50%
Score	7
Status	Fail

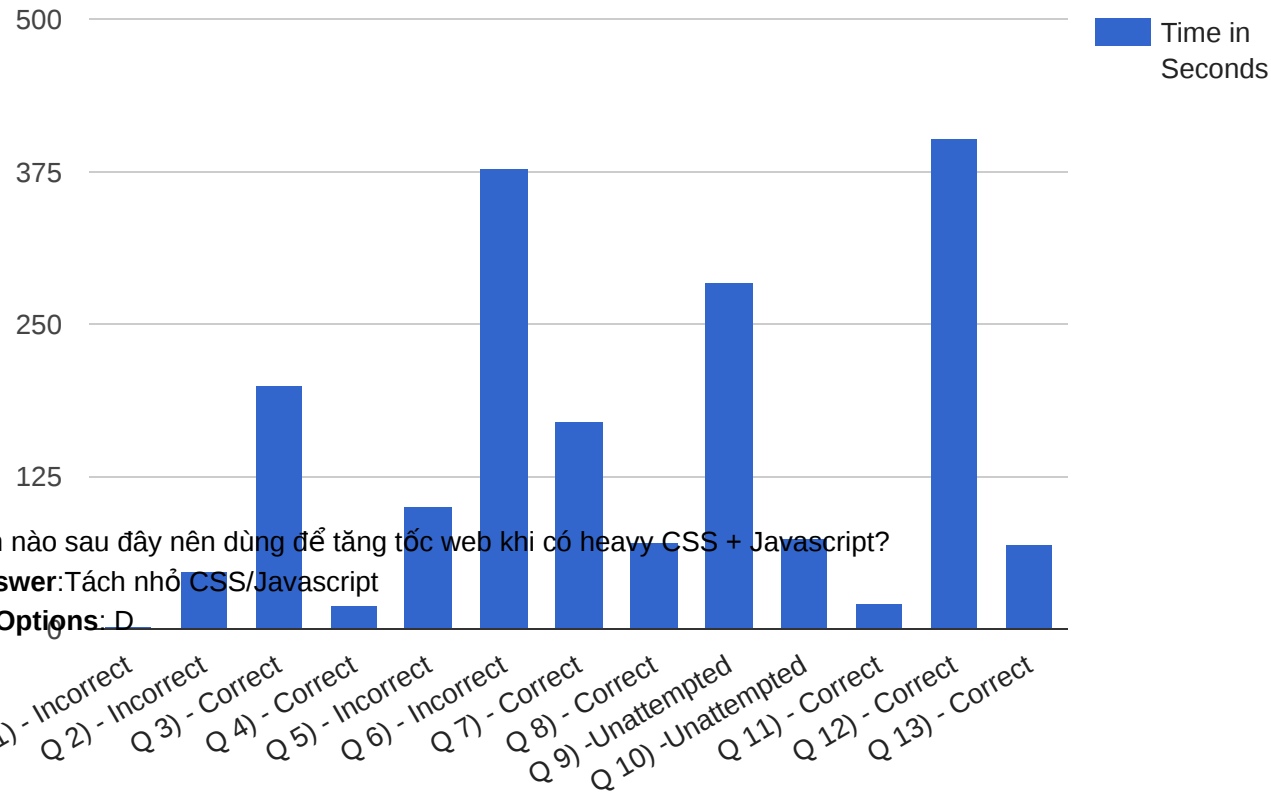
Categorywise Analysis

Category Name	Score	Time Spent (Approx.)	Correct	Incorrect	
[Training Test] Performances	7	30:40 Min.	7	4	2
Total	7	30:40 Min. (Approx.)	7	4	2

Top 10 results for Quiz: [Training Test] Performances



Time spent on individual question (in seconds)



1

Các cách nào sau đây nên dùng để tăng tốc web khi có heavy CSS + Javascript?

Your Answer: Tách nhỏ CSS/Javascript

Correct Options: D



2

Có nên dùng includes option của association để tránh N + 1

Answer

Word counts: 1

Có

"Không, option này làm association luôn luôn load khi resource load, nhiều lúc sẽ bị dư thừa.
Chỉ dùng option này khi ta thường xuyên load association với resource"



3

Khi ta config asset_host với 1 CDN, ta sẽ có lợi ích gì?

Answer

Word counts: 15

Giảm tải và tiết kiệm tài nguyên cho website



4

Ta nên dùng Gem gì để optimize JSON

Your Answer: oj

Correct Options: oj



5

Cho đoạn code sau:

```
@comments = Comment.all
@comments.each do |comment|
  p User.find_by_id(comment.user_id).try(:name)
end
```

Đoạn code trên bị dính lỗi N+1 query, nhưng model User và Comment không khai báo association với nhau, vậy làm cách nào để tránh được lỗi N+1 query trong trường hợp này?

Answer

Word counts: 20

```
@comments = Comment.includes(:user).all
@comments.each do |comment|
  p User.find_by_id(comment.user_id).try(:name)
end
```

Do User và Comment không khai báo association với nhau:

```
"@comments = Comment.all
users = User.where(id: @comments.map(&:user_id))
@comments.each do |comment|
  p users.find{|u| u.id == comment.user_id}.try(:name)
end"
```



6

Ta có nên đánh index lên tất cả các field để truy vấn có điều kiện cho nhanh ko? Thích where field nào thì where thoải mái, cho khỏe? Vì sao?

Answer

Word counts: 33

Không nên đánh index lên tất cả các field, chỉ nên áp dụng trên những table có lượng dữ liệu lớn

"Không, index sẽ làm chậm các câu DML (update/insert/delete)"



7

Các phát biểu nào sau đây đúng?

Your Answer: Field khóa ngoại nên đánh index, Field hay được sử dụng để filter nên đánh index,

Correct Options: Field khóa ngoại nên đánh index, Field hay được sử dụng để filter nên đánh index



8

Nên dùng Background Job cho các trường hợp nào sau đây:

Your Answer: Job chạy lâu, Job không cần chờ response (Gửi email),
Correct Options: Job chạy lâu, Job không cần chờ response (Gửi email)



9

Đoạn code sau được đặt trong Transaction có tác dụng gì?

```
Person.transaction do  
  my_collection.each do |name|  
    Person.create({:name => name})  
  end  
end
```

Answer

Word counts: 0

"- Chỉ cần tạo 1 connection để làm các công việc trên, tăng tốc độ
- Nếu rollback thì sẽ rollback tất cả các record (ý phụ, có thể bỏ qua)
"

10

Đoạn code sau được đặt trong Transaction có tác dụng gì?

```
Person.transaction do  
  active_people = Person.active.pluck(:email)  
  disabled_people = Person.disabled.pluck(:email)  
end
```

Answer

Word counts: 0

"Không có tác dụng, chỉ là câu truy vấn, ko cần dùng Transaction
"

11

Trong document về performance thì tool checking performance nào sau đây KHÔNG được đề cập đến

Your Answer: WebPagetest

Correct Options: WebPagetest



12

Cho 2 đoạn snippet sau:

a)

```
for (var i=0; i<1000; i++){  
$("#test").append("" + i + "");  
}
```

b)

```
html = "";  
for (var i=0; i<1000; i++){  
html += "" + i + "";  
}
```


`$("#test").append(html);`

Đoạn snippet nào tốt hơn? Tại sao? Giải thích chi tiết

Answer

Word counts: 72

Đoạn snippet b tốt hơn vì sẽ thực thi nhanh hơn, giảm số lần thực hiện. Với snippet a mỗi lần lặp là mỗi lần insert một HTML element. Còn với snippet b sẽ gộp hết kết quả của vòng lặp rồi insert một lần



13

Tại sao dùng cache lại làm cải thiện performance của app?

Answer

Word counts: 25

Những phần đã được cache sẽ không cần phải load nữa => giảm request tới server => tăng performance

