

Web Application Developer - Frontend - Level 1

A. Logical Test

A-Q1:

Sebuah perjalanan antar kota via gurun gersang membutuhkan waktu 6 hari. Satu orang hanya mampu membawa maksimal bekal makanan untuk 4 hari. Berapa orang tambahan yang paling sedikit yang diperlukan agar semua orang tersebut tidak ada yang meninggal kelaparan dan ada yang berhasil melewati gurun tersebut ? Jelaskan secara singkat.

A-A1:

Diperlukan total tiga orang untuk melewati gurun tersebut. Berarti, dua orang tambahan untuk melewati perjalanan ini agar satu orang dapat sampai ke tujuan. Sebutlah orang tersebut A, B, dan C memulai perjalanan. Setelah hari pertama, A memberikan makanan untuk B dan C, lalu pulang (A memiliki sisa bekal 1 hari untuk 1 hari perjalanan pulang). Sehingga B dan C memiliki bekal untuk 4 hari, dengan sisa perjalanan 5 hari. Lalu, setelah hari ke dua, B memberikan C makanan untuk satu hari, lalu pulang (B memiliki sisa bekal 2 hari untuk 2 hari perjalanan pulang). Sehingga, C memiliki bekal 4 hari, dengan sisa waktu perjalanan 4 hari.

A-Q2:

Anang dan Esa adalah saudara kembar dimana Anang lebih tua 3 menit dari Esa. Ketika Anang merayakan hari ulang tahun pada tanggal 10 Agustus 2017, ternyata Esa merayakan ulang tahun pada tanggal 9 Agustus 2017. Jelaskan secara singkat kira-kira kenapa hal tersebut bisa terjadi menurut Anda.

A-A2:

Hal tersebut bisa jadi karena Anang dan Esa lahir dalam perjalanan yang melewati suatu batas zona waktu. Anang lahir sebelum mereka melewati batas tersebut, sedangkan Esa lahir setelah melewati batasnya. Sehingga, ada kemungkinan Esa berada di Zona yang 1 hari lebih lambat daripada Zona tempat Anang lahir. Maka dari itu, Esa merayakan ulang tahunnya sehari sebelum Anang, walaupun Anang lahir 3 menit sebelumnya.

B. Theoretical Test

B-Q1:

Apakah yang dimaksud dengan AJAX, tolong dijelaskan secara detail menurut pemahaman Anda.

B-A1:

AJAX adalah *Asynchronous JavaScript and XML*. Ketika menggunakan AJAX, kita tidak perlu me-*refresh* page untuk mendapat respon dari *page*. Saat *client* melakukan *request* ke *server*, dilakukan dengan JavaScript. *Server* akan merespon dengan bentuk XML yang akan diolah oleh JavaScript, lalu akan ditampilkan ke halaman. Sehingga, akses menjadi lebih cepat dan tanpa *refresh*.

B-Q2:

Apa perbedaan antara responsive dan adaptive design? Dari keduanya, manakah yang lebih baik? Silahkan di elaborasi.

B-A2:

Responsive design merespon perubahan, sedangkan *adaptive design* mengadaptasi terhadap perubahan. Contohnya, perubahan ukuran *browser*. *Responsive Design* akan mengikuti perubahan ukuran sebagai responnya, sedangkan *adaptive design* akan mencari satu titik pada perubahan ukuran tersebut untuk beradaptasi dan merubah ukurannya. Sehingga, *responsive design* menjadi lebih fleksibel dan perubahannya lebih halus. Karena itu, *responsive design* menjadi pilihan yang lebih baik karena dapat mengikuti perbedaan-perbedaan yang ada pada *device* yang berbeda-beda dengan lebih fleksibel dan halus.

B-Q3:

Jelaskan proses yang terjadi dari sejak user mengetik URL di browser sampai dengan halaman muncul secara sempurna di layar (Web Page Lifecycle)

B-A3:

Dimulai dengan *request* terlebih dahulu dari *user*, lalu *server* akan menentukan apakah halaman harus di *parsing* atau *compile*. Lalu, dilakukan *Start* yang menetapkan properti-properti *request* dan *response*. Lalu, dilakukan inisialisasi halaman dimana *control properties* ditetapkan. Lalu dilakukan *load*, jika halaman merupakan *postback*, *control properties* akan diisi dengan informasi yang dibutuhkan. Lalu dilakukan *Event Handling*, jika *postback*, maka akan dilakukan validasi. Lalu dilakukan *rendering*, menuliskan output dari setiap objek pada *response*, dan *unload* ketika seluruh *page* sudah di *render*, dan *response* serta *request* dikosongkan.

B-Q4:

Technology Stack apakah yang biasa digunakan oleh Anda? Silahkan dijelaskan secara detail masing - masing tools dan kegunaannya.

B-A4:

HTML untuk struktur *Web Page* yang memegang kontennya, CSS untuk *styling* dan *design*, JavaScript untuk *control* interaksi pada *page*. Digunakan juga MySQL untuk *database page* tersebut. Saya juga pernah menggunakan ReactJS sebagai *library* untuk *user interface* dan Laravel sebagai framework sebuah website. Selain itu, saya sering menggunakan Apache sebagai *web server*. Saya juga pernah menggunakan Java, REST *architecture*, dan SOAP sebagai protokol sebuah web.

C. Case Study Test

Buatkan sebuah aplikasi todo list sederhana dengan spesifikasi sebagai berikut

1. Aplikasi harus ditulis hanya menggunakan Javascript, HTML, CSS atau varian nya (Typescript, Coffeescript, SASS, LESS atau varian lainnya)
2. Tidak ada data yang disimpan di server, semua data dikelola di browser (client-side)
3. Di dalam list harus mencakup
 - a. Checkbox untuk menandakan task sudah / belum dilakukan
 - b. Judul atau label task
 - c. Deskripsi
 - d. Tanggal mulai dan tanggal selesai
 - e. Prioritas (High, Medium, Low)
 - f. Tombol untuk menghapus task
4. Semua di point 3 harus editable
5. CRUD harus berfungsi dengan baik
6. Apps Responsive.

Anda diberikan kebebasan untuk memilih library dan framework akan digunakan. Silahkan buat dan share repo dari hasil test ini ke kami untuk dievaluasi, berikut dengan cara instalasinya (di dalam README.md)