Laravel

- Membangunkan Authentication Server menggunakan Laravel/Passport
- Selected - paparkan default value pada dropdown

Membangunkan Authentication Server menggunakan Laravel/Passport

Install aplikasi Laravel seperti biasa. Kemudian, rujuk dokumentasi Laravel/Passport di https://laravel.com/docs/11.x/passport

1. Install menggunakan arahan di bawah:

   ```
   php artisan install:api --passport
   ```

   Anda akan ditanya sama ada ingin menggunakan UUID sebagai primary key. pilih 'No' jika anda ingin menggunakan auto increment sebagai primary key.
2. Buka model User di App\Models\User.php, pada ruang "import", tambah baris "use Laravel\Passport\HasApiTokens;" manakala dalam class, tambah "HasApiTokens" seperti di bawah:

```php
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Passport\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;
```

3. Akhirnya, tambah guards di dalam file config/auth.php seperti contoh di bawah:

```
'guards' => [
    'web' => [
        'driver' => 'session',
        'provider' => 'users',
    ],

    'api' => [
        'driver' => 'passport',
        'provider' => 'users',
    ],
],
```

Sekiranya encryption keys belum dibuat, laksanakan arahan "php artisan passport:keys". Jika terdapat error "Encryption keys already exist. Use the --force option to overwrite them.", abaikan langkah ini. Encryption key telah dijana semasa proses instalasi.

Untuk menambah aplikasi yang akan menjadi "client" atau menggunakan User dalam sistem ini, laksanakan arahan "php artisan passport:client". Simpan Client ID dan Client Secret di dalam env aplikasi client.

bangunkan laman Daftar dan Login di dalam sistem ini.

Selected - paparkan default value pada dropdown

Untuk memaparkan default value dropdown pada borang di dalam view, gunakan arahan

@selected($borang->nilai == $dropdown->nilai)

sebagai contoh, katakan kita mempunyai table **Lokasi** yang menyimpan lokasi fail seperti di bawah:

Skema	Data
Lokasi id (int) nama (string)	

id	nama
1	Rak A, Bilik Fail
2	Rak B, Bilik Fail
3	Rak A, Stor Fail

Di bawah pula table **Fail** yang menyimpan senarai fail beserta lokasinya:

Skema	Data
Fail id (int) no_fail (string) tajuk_fail(string) lokasi_id(int)	

id	no_fail	tajuk_fail	lokasi_id
1	ABC/2023/001	Projek Simulasi	2
2	ABC/2023/002	Pembelian Alatganti	1
3	ABC/2024/001	Sistem POS	2

Katakan kita ingin mengemaskini data id=2 untuk table Fail. Berikut adalah contoh kod blade untuk borang kemaskini tersebut:

```
<form action="{{ route('simpan_fail') }}" method="post">
    @csrf
    <input type="hidden" value="{{ $fail-&gt;id }}">
    <label for="no_fail">No Fail</label>
        <input name="no_fail" type="text" value="{{ $fail-&gt;no_fail }}">
    <label for="nama_fail">Tajuk Fail</label>
        <input name="tajuk_fail" type="text" value="{{ $fail-&gt;tajuk_fail }}">
    <label for="lokasi">Lokasi</label>
        <select name="lokasi_id">
            @foreach($lokasis as $lokasi)
                <option value="{{ $lokasi->id}}"> @selected($lokasi->id == $fail->lokasi_id) >{{ $lokasi-&gt;nama }}</option>
            @endforeach
        </select>
</form>
```