

## LoginListener.php

```
class LoginListener
{
    /**
     * Create the event listener.
     *
     * @return void
     */
    public function __construct()
    {
        //
    }

    /**
     * Handle the event.
     *
     * @param object $event
     * @return void
     */
    public function handle($event)
    {
        $event->user->update([
            'last_login' => now(),
            'login_count' => $event->user->login_count + 1,
        ]);
    }
}
```

- Listener for **Login Event** to update “last\_login and login\_count” upon logging in.

## EventServiceProvider.php

```
<?php

namespace App\Providers;

use App\Listeners\LoginListener;
use Illuminate\Auth\Events\Login;
use Illuminate\Auth\Events\Registered;
use Illuminate\Auth\Listeners\SendEmailVerificationNotification;
use Illuminate\Foundation\Support\Providers\EventServiceProvider as ServiceProvider;
use Illuminate\Support\Facades\Event;

class EventServiceProvider extends ServiceProvider
{
    /**
     * The event listener mappings for the application.
     *
     * @var array<class-string, array<int, class-string>>
     */
    protected $listen = [
        Registered::class => [
            SendEmailVerificationNotification::class,
        ],
        Login::class => [
            LoginListener::class,
        ],
    ];
}
```

- Declaration of **Login Event**

## UpdateUserPassword.php

```
/**
 * Validate and update the user's password.
 *
 * @param mixed $user
 * @param array $input
 * @return void
 */
public function update($user, array $input)
{
    Validator::make($input, [
        'current_password' => ['required', 'string'],
        'password' => $this->passwordRules(),
    ])->after(function ($validator) use ($user, $input) {
        if (! isset($input['current_password']) || ! Hash::check($input['current_password'], $user->password)) {
            $validator->errors()->add('current_password', __('The provided password does not match your current password'));
        }
    })->validateWithBag('updatePassword');

    $user->forceFill([
        'password' => Hash::make($input['password']),
        'last_pass_change' => now()
    ])->save();
}
```

- Updating “last\_pass\_change” to current date upon successfully updating the password.

## DATABASE

Type	active	date_of_birth	schoolName	last_login	login_count	last_pass_change
her	1	1998-12-22	HARVARD	NULL	0	NULL
ent	1	2022-03-15	NDKC23	2022-03-15 14:57:06	9	NULL
ent	1	2022-03-15	NDKC	2022-03-15 16:07:43	8	2022-03-15 16:09:31