

## Laravel Question

Composer কি? :

Composer এটি একটি ডেপেন্ডেন্সি ম্যানেজমেন্ট টুল, এবং এটি প্রোজেক্ট ম্যানেজমেন্ট টুল যা লারাভেল এবং অন্যান্য PHP প্রজেক্টের জন্য ব্যবহার করা হয়। Composer এর সাহায্যে আপনি PHP প্যাকেজ, লাইব্রেরি, ফ্রেমওয়ার্ক ইত্যাদি ইন্সটল করতে পারেন এবং প্রজেক্টের ডিপেন্ডেন্সিগুলি সহজেই ম্যানেজ করতে পারেন।

Composer একটি JSON ফাইল যা প্রজেক্ট ডিপেন্ডেন্সিগুলির তথ্য সংরক্ষণ করে এবং এই ডিপেন্ডেন্সিগুলি ইন্সটল করার জন্য ব্যবহৃত হয়। যখন আপনি একটি লারাভেল প্রজেক্ট শুরু করতে থাকেন বা নতুন ডিপেন্ডেন্সি যোগ করতে চান, তখন Composer ব্যবহার করে আপনি সহজেই প্যাকেজ ইন্সটল করতে পারেন। এটি একটি প্রজেক্টের ডিপেন্ডেন্সি রেজুম তৈরি করে যা `composer.json` নামে থাকে এবং প্যাকেজ ইন্সটল করার সময় ব্যবহৃত হয়।

সংক্ষেপে বলা যায়, Composer হলো একটি ডেপেন্ডেন্সি ম্যানেজমেন্ট টুল যা লারাভেল এবং অন্যান্য PHP প্রজেক্টের জন্য ব্যবহার করা হয় যাতে প্যাকেজ ইন্সটল করতে এবং ডিপেন্ডেন্সিগুলি ম্যানেজ করতে সহজ হয়।

## মিডলওয়্যার

মিডলওয়্যার লারাভেলের খুবই গুরুত্বপূর্ণ ও মজার একটি ফিচার। সহজ কথায় বলতে গেলে এটি একটি ফিল্টার মেকানিজম। যেমন: লারাভেলের (ইউজার অথেনটিকেশন)।

### মিডলওয়্যার আসলে কি ?

মিডলওয়্যার কি তা তার নামের মধ্যেই আছে। মিডলওয়্যার রিকুয়েস্ট ও রেস্পন্সের মধ্যে সম্পর্ক স্থাপনকারী। ইউজার এর HTTP রিকুয়েস্ট ফিল্টার করে রিকুয়েস্টের পরে রেস্পন্স কি হবে তা মিডলওয়্যারে লেখা থাকে।

### কখন মিডলওয়্যার ব্যবহার করব ?

১. যেকোনো রকমের ইউজার অথেনটিকেশন।
  ২. রাউটের মাধ্যমে আসা কোন API KEY হ্যান্ডেল করতে।
  ৩. আপনার অ্যাপের জন্য এক্সেস মডিফায়ার তৈরি করতে (যেমন: ইউজার গেট হলে কি দেখবে অথবা ইউজার অ্যাডমিন হলে কোন পেজ দেখাবে)।
- এতক্ষণে আশা করছি বুঝেই গেছেন মিডলওয়্যার এর গুরুত্ব। আসুন এবার মিডলওয়্যার বানিয়েই ফেলি।

### ১. মিডলওয়্যার তৈরি:

Artisan এর মাধ্যমে লারাভেলে মিডলওয়্যার তৈরি করা সত্যিই পান্তাভাত (ইলিশ মাছ ছাড়া :p) Terminal ওপেন করুন এবং লিখুনঃ

```
php artisan make:middleware <middleware Name>
```

## ২. মিডলওয়্যার রেজিস্টার করাঃ

আমরা একটা মিডলওয়্যার নাহয় বানিয়েই ফেললাম। কিন্তু দেখেন এখন যে যুগ আসছে সবাইকেই ভেরিফাই করা লাগে !! তাহলে মিডলওয়্যারকে কেন নয় ??

চলুন দেখে নেই কিভাবে মিডলওয়্যার ভেরিফাই করা যায়।

আপনি যদি আপনার মিডলওয়্যারটিকে সব রিকুয়েস্টের ক্ষেত্রেই একটিভ দেখতে চান তাহলেঃ

app/Http/kernel.php এই ফাইলটি ওপেন করুন। এবং \$middleware অ্যারেতে এভাবে অ্যাড করুনঃ

```
protected $middleware = [
```

```
...
```

```
\App\Http\Middleware\<middleware Name>::class
```

```
]
```

## ৩. রাউটারে মিডলওয়্যার রেজিস্টারঃ

এখন আপনি যদি মিডলওয়্যার ইউজারকে আপনার অ্যাপের নির্দিষ্ট কোন লিঙ্কে যাওয়ার সময় চেক করতে চান তাহলে মিডলওয়্যারটিকে রাউটের সাথে বেধে দিতে পারেন। যেমনঃ

```
Route::get('welcome', 'WelcomeController@profile_info');
```

এখানে লক্ষ করুন ইউজার <appURL>/welcome লিখলে রাউট তাকে WelcomeController এর profile\_info function এ নিয়ে যাবে।

এখন আপনার ভাবনায় আসলো যে ...না এভাবে কোন ইউজার এসে প্রফাইল ইনফো নিয়ে জেতে পারবেন না। তাকে আগে দেখাতে হবে যে সে আসলেই আমার ইউজার। আপনার এই সমস্যার সমাধান নিয়েই হাজির মিডলওয়্যার!!

আপনি ইচ্ছা করলেই রাউটএর সাথে বেধে দিতে পারেন আপনার মিডলওয়্যারকে ঠিক আমি জেভাবে লিখলাম সেভাবে। ধরে নেই আপনার একটি auth মিডলওয়্যার আসে যেটা আপনার অ্যাপ ইউজার কিনা চেক করে। তাহলে লিখে ফেলুন এভাবেঃ

```
Route::get('welcome', 'WelcomeController@profile_info')->middleware('Auth');
```

আবার ইচ্ছা করলে একাধিক মিডলওয়্যার ও ব্যবহার করতে পারেন এভাবেঃ

```
Route::get('welcome', 'WelcomeController@profile_info')->middleware('Auth', 'admin');
```

## ৪. মিডলওয়্যার গ্রুপঃ

লারাভেল আপনার কাজকে সহজ করার জন্য অতিমানবীয় আইডিয়া এনেছে এখানে। আপনি ভাবছেন আপনার অ্যাপে ওয়েব ও এপিআই রাউট দুইভাবে ইউজারকে ডাটা দিবেন। কিন্তু এত নিখুত ভাবে কত মিডলওয়্যার লিখা যায় ?? এই সমস্যা সমাধান এর জন্য লারাভেল এনেছে গ্রুপ মিডলওয়্যার। app/Http/kernel.php এই ফাইলে দেখবেন এভাবে লেখা আছে।

```
protected $middlewareGroups = [
```

```
'web' => [
```

```
\App\Http\Middleware\EncryptCookies::class,
```

```

    \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
    \Illuminate\Session\Middleware\StartSession::class,
    \Illuminate\View\Middleware\ShareErrorsFromSession::class,
    \App\Http\Middleware\VerifyCsrfToken::class,
    \Illuminate\Routing\Middleware\SubstituteBindings::class,
],
'api' => [
    'throttle:60,1',
    'auth:api',
],
];

```

এই দুই অ্যারেতে আপনার ইচ্ছানুযায়ী মিডলওয়্যার অ্যাড করে দিতে পারেন।

## 5. মিডলওয়্যার প্যারামিটার:

আমরা ইচ্ছা করলে মিডলওয়্যারে প্যারামিটার পাস করতে পারি। ধরি আমাদের মিডলওয়্যারটি ইউজারের রোল চেক করবে। তো মিডলওয়্যার ফাংশনে আমরা Closure প্যারামিটার এর মধ্যে দিয়ে \$request পাস করতে পারি। যেমনঃ

```

public function handle($request, Closure $next, $role)
{
    if (! $request->user()->hasRole($role)) {
        // Redirect...
    }
    return $next($request);
}

```

এখন আমরা রাউটে মিডলওয়্যার অ্যাড করে দিব।

### app

এই ডিরেক্টরিতে সব এপ্লিকেশন লজিক থাকে, যেগুলো আমরাই বানাবো। যেমনঃ মডেল, কন্ট্রোলার, রাউট

### bootstrap

এটা লারাভেল এর কিছু ফাইল থাকে যা দিয়ে ফ্রেমওয়ার্ক তাকে এক সাথে কাজ করতে সাহায্য করে, এখানে cache ফাইল গুলোও থাকে।

### config

এই ডিরেক্টরির ফাইল গুলো ব্যবহার করে আমাদের অ্যাপ্লিকেশান কনফিগার করি, যেমন ডাটাবেস, মেইল, সেশন ইত্যাদি।

### database

এখানে সব ডাটাবেস মাইগ্রেসান, সীড থাকে। এমনকি SQLite database ফাইলটাও এখানে থাকে।

## public

সব ধরনের আসেট এখানে রাখে, যেমন CSS, JS, fonts, ছবি ইত্যাদি।

## resources

আমাদের ফ্রন্ট এন্ড ফাইল যাকে ভিউ বলি সেগুলো এখানেই রাখব। তা ছাড়া LESS, SASS, CoffeeScript ও ভাষার ফাইলগুলোও এখানে থাকে।

## routes

এখানে সব রাউটের ডিফাইনেশন গুলো থাকে। যেমনঃ web.php, api.php, ও console.php

## storage

লারাভেল ব্যবহার করে, সব কম্প্লাইড টেম্পলেট, সেশন, cache ফাইল রাখার জন্য।

## tests

টেস্ট ফাইল গুলো থাকে, যেমনঃ PHPUnit ফাইল।

## vendor

সব Composer dependency ফাইলগুলো থাকে।

আসলে লারাভেল নিয়ে কাজ করতে থাকলে এগুলো এমনিতেই পরিচিত হয়ে যাবে।

তবুও আরও জানতে চাইলে [Laravel Application structure](#) দেখুন।

পরবর্তী চ্যাপ্টারে **বেসিক রাউটিং** নিয়ে আলোচনা করা হবে।

## \*. কন্ট্রোলার আর নেমস্পেস:

অ্যাপ্লিকেশন ডেভেলপ করার সময় কিছু কন্ট্রোলারকে পৃথক কোন ফোল্ডার কিংবা ডিরেক্টরিতে রাখতে হয়। আর এইক্ষেত্রে আমরা কাস্টম নেমস্পেস ব্যবহার করে তা করতে পারি। ধরুন লারাভেলের কন্ট্রোলার ডিরেক্টরিতে **Photos** নামে একটি সাব-ডিরেক্টরি তৈরি করব আর ওই সাব-ডিরেক্টরিতে **AdminController** রাখব তাহলে কন্ট্রোলার ফাইলটি নিচের মত হবে।

```
<?php namespace App\Http\Controllers\Photos;
use App\Http\Controllers\Controller;
class AdminController extends Controller {
    public function index() {
        return "You've arrived at Admin Controller.";
    }
}
```

## \*Tinker:

Laravel Tinker হলো Laravel এর একটি REPL (Read-Eval-Print Loop) টুল, যা Laravel

অ্যাপ্লিকেশনের সাথে ইন্টারয়েক্ট করতে একটি প্রাক্টিকাল মাধ্যম প্রদান করে। Tinker ব্যবহার করে, আপনি আপনার অ্যাপ্লিকেশনের মডেল, ডেটাবেস সম্পর্কিত কাজ, কাস্টম কোয়েরি, এবং অন্যান্য সম্পর্কিত কাজ সহ বিভিন্ন কাজ করতে পারেন।

Tinker ব্যবহারের জন্য নিম্নলিখিত ধাপগুলি অনুসরণ করুন:

1. টার্মিনাল প্রোজেক্ট ডিরেক্টরিতে যান এবং নিম্নলিখিত কমান্ড চালান:

Copy code

php artisan tinker

2. Tinker এ প্রবেশ করলে, আপনি Laravel এপ্লিকেশনের বাচ্চা সব কিছুই এখানে ব্যবহার করতে পারেন, যেমন:

- মডেলের ডেটা প্রদান করা এবং ডেটা সংরক্ষণ করা.
- ডেটাবেস তালিকা বা ডেটাবেস কর্মক্ষেত্রে কাস্টম কোয়েরি চালানো.
- ব্যবহারকারী তথ্য প্রবেশ করা এবং তথ্য সম্পর্কিত পরিবর্তন সাবধানে চেষ্টা করা.
- সব ধরনের লারাভেল মডুল এবং সেটিংস ব্যবহার করা।

এই উপায়ে, আপনি আপনার Laravel এপ্লিকেশনের নিচের স্তরে গিয়ে ডেটা প্রদান, পরিবর্তন এবং চেক করতে পারেন সহজভাবে এবং ডেভেলপমেন্ট প্রসেস সময় সাথে সাথে দেখতে পারেন।

Tinker একটি মহান উপায় যেটি ডেভেলপমেন্ট এবং ডেবাগিং প্রসেস সহজ করে তুলে ধরে Laravel প্রজেক্ট ডেভেলপমেন্ট করার জন্য।

**সার্ভিস প্রোভাইডার (Service Provider):** সার্ভিস প্রোভাইডার হলো একটি লারাভেল কম্পোনেন্ট যা সেই কম্পোনেন্টের সেবা বা প্যাকেজ প্রদান করে। এটি সার্ভিস কন্টেইনারে কোন সেবা প্রদান করতে পারে এবং সেটি সার্ভিস কন্টেইনারের মধ্যে নিবন্ধিত করে থাকে।

একটি সার্ভিস প্রোভাইডার তৈরি করতে, আপনি নিম্নলিখিত ধাপগুলি অনুসরণ করতে পারেন:

1. নতুন সার্ভিস প্রোভাইডার তৈরি করতে, নিম্নলিখিত কমান্ড ব্যবহার করুন:
- 2.

```
php artisan make:provider YourServiceProvider
```

এই কমান্ডটি একটি নতুন সার্ভিস প্রোভাইডার ক্লাস তৈরি করবে যেটি `app/Providers` ফোল্ডারে সংরক্ষিত হবে।

3. নতুন সার্ভিস প্রোভাইডারের `register` মেথডে সেবাটির নিবন্ধন করুন। এই মেথডে আপনি সার্ভিস কন্টেইনারে সেবাটি নিবন্ধন করতে পারেন এবং সেটির ক্রিয়াকলাপ নির্ধারণ করতে পারেন।

- 4.

```
public function register()  
{  
    $this->app->bind('your-service', function () {  
        return new YourService();  
    });  
}
```

10. }

- 11.

12. উপরের উদাহরণে, আমরা 'your-service' নামের একটি সেবা নিবন্ধন করেছি এবং এটির জন্য একটি ফাংশন সংরক্ষিত করেছি যা সেবাটি নির্মাণ করতে ব্যবহার হবে।

**সার্ভিস কন্টেইনার (Service Container):** সার্ভিস কন্টেইনার হলো একটি সংরক্ষণ সিস্টেম যা লারাভেল এপ্লিকেশনে সেবা প্রদান এবং সেবা গ্রহণের জন্য ব্যবহার করা হয়। এটি একটি ডিপেন্ডেন্সি ইনজেকশন কন্টেইনার হিসেবে কাজ করে, যাতে কোনও সেবা অপর সেবা গ্রহণ করতে পারে এবং এটির মাধ্যমে ডিপেন্ডেন্সি ইনজেকশন করা যায়।

একটি সেবা সেবা কন্টেইনার থেকে গ্রহণ করতে, আপনি নিম্নলিখিত ধাপগুলি অনুসরণ করতে পারেন:

1. আপনি কোন ক্লাস বা ফাংশন এক্সেস করতে চান যা সার্ভিস কন্টেইনার থেকে সেবা নিয়ে। তারপর, সার্ভিস কন্টেইনার থেকে সেবা নিতে, `app()` ফাংশন ব্যবহার করুন এবং সেবার নাম পাস করুন:

2.

```
$yourService = app('your-service');
```

উপরের উদাহরণে, 'your-service' সেবা নামটির সাথে সেবা সংশ্লিষ্ট ক্লাসটি ইন্সট্যান্টিয়েট করার জন্য `app()` ফাংশন ব্যবহার করা হয়েছে।

3. এটির পর, আপনি সেবার মধ্যে সম্পাদনা করতে পারেন এবং সেবা ব্যবহার করতে পারেন:

4.

```
$result = $yourService->someMethod();
```

উপরের উদাহরণে, `$yourService` এখানে 'your-service' সেবার ইন্সট্যান্ট এবং `someMethod()` মেথড ব্যবহার করতে পারেন।

এই উদাহরণে, "সার্ভিস প্রোভাইডার" সার্ভিসটি নিবন্ধন করে এবং "সার্ভিস কন্টেইনার" সেবা নিয়ে ব্যবহার করে এটি অ্যাপ্লিকেশনে ব্যবহার করতে সাহায্য করে। এই প্রসেসটি একটি লারাভেল এ্যাপ্লিকেশনে কাস্টম সার্ভিস প্রোভাইড এবং উপযুক্ত সার্ভিস নিয়ে কাজ করতে সাহায্য করে।

•

1. লারাভেল কি, এটি কেন ব্যবহার করা হয়?

**উত্তর:** লারাভেল একটি ওপেন সোর্স পিএইচপি ফ্রেমওয়ার্ক, যা ওয়েব অ্যাপ্লিকেশন ডেভেলপমেন্ট জন্য ব্যবহার করা হয়। এটি প্রযুক্তিগত এবং পার্যাপ্ত ডকুমেন্টেশন সহ অনেক পরিস্থিতির ফিচার সেট সরবরাহ করে, যা উন্নত এবং স্কেলেবল ওয়েব অ্যাপ্লিকেশন তৈরি করতে সাহায্য করে। এটি এলাকাপ্রদান করে MVC (Model-View-Controller) প্যাটার্নে ডেভেলপমেন্ট সাপোর্ট করে, যা ডেভেলপারদের অ্যাপ্লিকেশনের ম্যানেজমেন্ট এবং পরিচালনা সহায়ক করে।

## 2. লারাভেলের মুখ্য ফিচারগুলি কী?

**উত্তর:** লারাভেলের মুখ্য ফিচারগুলি নিম্নলিখিত:

- **Eloquent ORM:** Eloquent ডেটাবেস টেবিলগুলির সাথে সহজে ইন্টারফেস করতে সাহায্য করে এবং এলাকাপ্রদান করে PHP মডেলের সাথে।
- **ডেটা ব্যাসেড রাউটিং:** লারাভেল পাথের মাধ্যমে ডেটাবেস থেকে রাউটিং কনফিগার করে রাউট প্রাপ্ত করতে সাহায্য করে।
- **এটিস্যন কনসোল:** পার্টিয়াল কমান্ড লাইন টুল এটিস্যন ডেভেলপারদের প্রজেক্ট ম্যানেজমেন্ট এবং কোড জেনারেট করতে সাহায্য করে।
- **সেশন ম্যানেজমেন্ট:** লারাভেল সেশন ম্যানেজ করতে সাহায্য করে, যা অ্যাপ্লিকেশনের স্টেট সংরক্ষণ করে এবং ব্যবহারকারী সেশন ম্যানেজমেন্ট এবং সিকিউরিটি সাপোর্ট করে।
- **স্যাকশন:** একটি কোড ক্লিনিং টুল, যা অ্যাপ্লিকেশনের সিকিউরিটি সমর্থন করে এবং এরর হ্যান্ডলিং সুসংগত করে।

\*.লারাভেল রুটিং কি এবং কীভাবে এটি কনফিগার করা যায়?

**উত্তর:** লারাভেল রুটিং হলো URL থেকে বিভিন্ন এক্সপ্রেশনের অনুসরণ করে বিশেষ কাজ করার ধাপ। রুটিং ব্যবহার করে আপনি ব্যবহারকারীদের বিভিন্ন URL প্যাটার্নের অনুসরণ করে নির্দিষ্ট হ্যান্ডলার ফাংশনে পৌঁছাতে পারেন।

লারাভেলে রুটিং কনফিগার করার জন্য, নিম্নলিখিত ধাপগুলি অনুসরণ করতে হয়:

- **routes/web.php** ফাইলে আপনার ওয়েব অ্যাপ্লিকেশনের রুট ডিফাইন করুন, উদাহরণস্বরূপ:

phpCopy code

```
Route::get('/', 'HomeController@index');
```

- উপরের উদাহরণে, / প্যাথে একটি **HomeController** ক্লাসের **index** মেথড কে হ্যান্ডল করতে বলা হচ্ছে।
- আপনি এই রুটিং প্রোজেক্টের অন্যান্য বিভিন্ন URL প্যাটার্নে ব্যবহার করতে পারেন, এবং এগুলি বিভিন্ন হ্যান্ডলার ফাংশনে নির্দিষ্ট করতে পারেন।

## 3. লারাভেলে Middleware কি এবং এটি কীভাবে কাজ করে?

**উত্তর:** Middleware হলো লারাভেলের একটি কনসেপ্ট, যা HTTP রিকুয়েস্টের মধ্যে মধ্যে কাস্টম প্রক্রিয়া সমর্থন করে। Middleware দ্বারা একটি রিকুয়েস্টের পূর্ব এবং পরের প্রক্রিয়া সাজানো যায়, এবং এটি বিভিন্ন ধরনের প্রক্রিয়া সমর্থন করতে সাহায্য করে, উদাহরণস্বরূপ:

- **Authentication Middleware:** ব্যবহারকারীর প্রমাণীকরণের জন্য ব্যবহৃত হয়, যাতে যে ব্যবহারকারী লগইন আছে তা নির্ধারণ করা যায়।

- **Authorization Middleware:** এটি ব্যবহারকারীদের কি ধরনের অ্যাক্সেস অনুমতি আছে তা নির্ধারণ করে, উদাহরণস্বরূপ কোন পোস্ট বা পৃষ্ঠা একটি ব্যবহারকারী একবারে একবারে একটি বা একাধিক বার সম্পর্ক করতে পারে কিনা সেটি নির্ধারণ করতে সাহায্য করে।
- **Logging Middleware:** এটি রিকুয়েস্টের ডেটা প্রটোকল বা ট্র্যাশ রিপোর্ট তৈরি করে এবং এগুলি লগ ফাইলে লেখা যায়, যা ব্যবহারকারীর সংক্রান্ত মানদণ্ড যাচাই করতে সাহায্য করে।

Middleware মোডিউলার এবং পুনর্ব্যবহার যাচাইয়ে সহজে কাস্টমাইজ করা যায়, এবং এটি রিকুয়েস্টের বিভিন্ন ধরনের প্রক্রিয়া সমর্থন করে আপনার অ্যাপ্লিকেশনের কাস্টম লজিক যোগ করতে সাহায্য করে।

4. লারাভেলে কন্ট্রোলার কি এবং কীভাবে এটি তৈরি করা হয়?

**উত্তর:** কন্ট্রোলার হলো লারাভেলের একটি ক্লাস, যা ওয়েব অ্যাপ্লিকেশনের ব্যবহারকারী রিকুয়েস্ট সমর্থন করে এবং সেই রিকুয়েস্টের উপর বিভিন্ন অ্যাকশনের প্রক্রিয়া সমর্থন করে। কন্ট্রোলারের মধ্যে ব্যবহারকারী রিকুয়েস্টের জন্য নির্ধারণ করা লজিক থাকে, উদাহরণস্বরূপ ডেটা সংগ্রহণ এবং ডেটা প্রস্তুত করতে এবং ডেটা মডেলে সংরক্ষণ করতে।

একটি কন্ট্রোলার তৈরি করতে, নিম্নলিখিত কমান্ড ব্যবহার করুন:

goCopy code

```
php artisan make:controller YourControllerName
```

এই কমান্ডটি একটি নতুন কন্ট্রোলার ফাইল তৈরি করবে, যেটি **app/Http/Controllers** ফোল্ডারে সংরক্ষিত হবে। এরপর, আপনি আপনার কন্ট্রোলারের মধ্যে বিভিন্ন পাবলিক মেথড তৈরি করতে পারেন, যেগুলি আপনার ব্যবহারকারী রিকুয়েস্টের সমর্থন করতে সাহায্য করবে।

উদাহরণস্বরূপ, নিম্নলিখিত কন্ট্রোলারে একটি স্বাগতমেসেজ প্রদান করার উদাহরণ:

phpCopy code

```
// app/Http/Controllers/YourControllerName.php namespace App\Http\Controllers; use Illuminate\Http\Request; class YourControllerName extends Controller { public function welcome() { return "স্বাগতম লারাভেল!"; } }
```

এই কন্ট্রোলারের **welcome** মেথড রিকুয়েস্টের সাথে সংরক্ষণ করা এবং একটি স্বাগতমেসেজ প্রদান করবে।